

POCHE COMPUTERS

POCHE
COMPUTERS

Le magazine des passionnés
de micros de poche
et de calculatrices

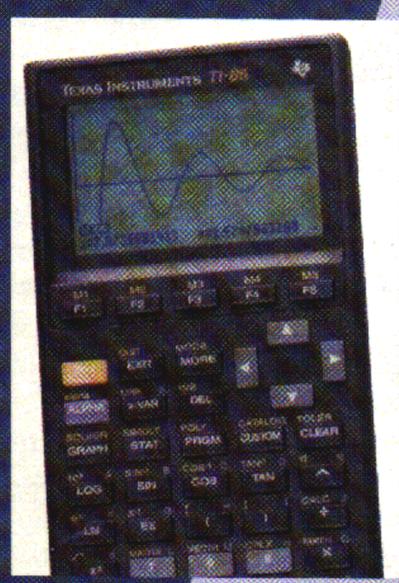
SHARP : PC3100
PCE500

N°2

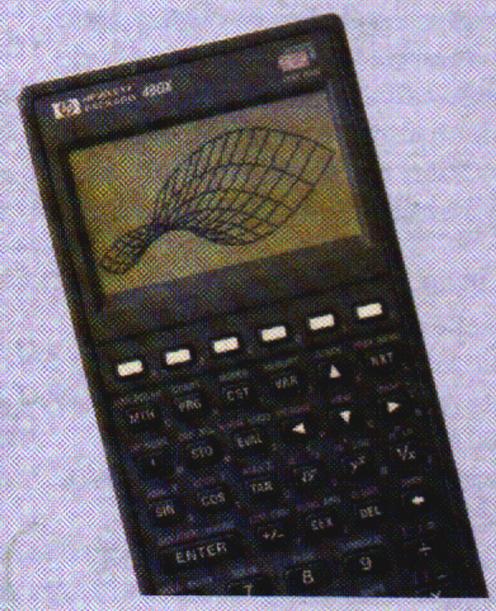


Des programmes
décoiffants
sur Casio graphiques
et basic!

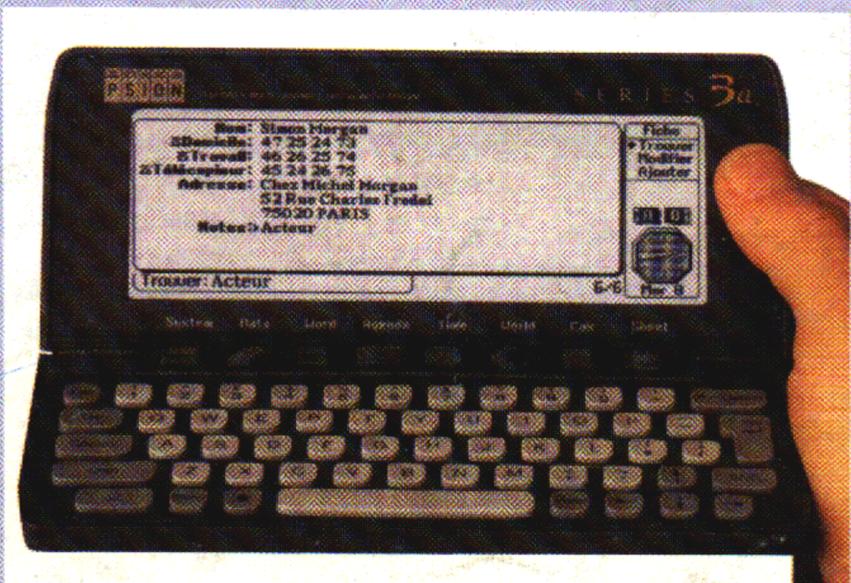
Ne jetez pas votre
Casio PB1000 !
Il a encore tant de
choses à dire !



TI 85 :
résolution d'équations



INITIATION A
L'ASSEMBLEUR
sur HP 48



NOUVEAU !!!
Psion series 3a
une révolution à l'anglaise ?

Dom-tom 39 F
Belgique 256 FB
Canada 9,50 \$ can
Luxembourg 245 lux
Suisse 10,8 Fs
Italie 8500 L
Espagne 1000 pta
USA 7,5 \$

35 F

L4008 - 2 - 35,00 F -RD





ATARI PORTFOLIO

MICROCARDS - 42220 - BURDIGNES cedex - FRANCE - tél (33) 77.39.68.13 - fax (33) 77.39.19.60
Tarif TTC au 15/12/1993 (ce tarif remplace et annule les précédents)

HARDWARE

Cartes mémoire RAM (format carte de crédit)

Cardram CCR 128 495,00 F

Cardram CCR 256 690,00 F

Cardram CCR 512 960,00 F

Cartes mémoire ROM (format carte de crédit)

Cardrom CCR 128 430,00 F

Cardrom CCR 256 580,00 F

Cardrom CCR 512 810,00 F

(nota: les cardroms sont programmables uniquement avec un programmeur spécial)

Portfolios neufs

Portfolio 128K 1690,00 F

Portfolio 256K 1990,00 F

Portfolio 384K 2260,00 F

Portfolio 512K 2510,00 F

Extensions mémoire interne (augmentent la mémoire interne du Portfolio ; nécessite l'envoi de votre appareil)

EMI 256K (ajoute 128K au Portfolio) 520,00 F

EMI 384K (ajoute 256K au Portfolio) 790,00 F

EMI 512K (ajoute 384K au Portfolio) 980,00 F

Packalim Système remplaçant les piles par des accus très puissants (850 ou 1200 mA) et se rechargeant dans le Portfolio par un bloc secteur. Ce système comprend la fourniture des 3 éléments : chargeur interne (nécessite donc l'envoi du Portfolio) + accus + alimentation : Très fiable. Amortissement du système en quelques mois. Nécessite l'envoi du Portfolio.

Le packalim avec accus de 850 mA 320,00 F

Le packalim avec accus de 1200 mA 420,00 F

Alimentation de sauvegarde Complémentaire du Packalim, cette alim permet de sauvegarder les données du Portfolio en cas de problème d'alimentation générale (plantage, micro coupures etc...) Un micro accumulateur vient prendre immédiatement le relais des piles ou accus (conservation des données durant plusieurs mois sans aucune source d'alimentation ; nécessite l'envoi du Portfolio).

Alimentation de sauvegarde 245,00 F

Interface parallèle La liaison facile entre un Portfolio et un PC ou une imprimante. 460,00 F

Câble de liaison PC 80,00 F

Câble de liaison imprimante 80,00 F

Interface série Permet la communication également avec d'autres périphériques et notamment le Macintosh

Interface série 520,00 F

Changement de Rom interne Remplacement des versions 1.052 ou 1.072 du bios par une version récente 1.130 ; permet d'éviter les plantages et autres erreurs division par zéro ! BIOS 370,00 F
Accumulateurs surpuissants ! Ces nouveaux accumulateurs possèdent une puissance de 1200 mA avec une tension maxi de 1,55 volt, et pour une taille standard de type R6. Remplacent les piles du Portfolio et en augmentent son autonomie de 40%. Ils sont utilisables également dans d'autres appareils, ainsi qu'en photo et en modélisme.

Le 850mA prix u. : 23 F - le 1200mA prix u. : 46 F

SERVICE APRES-VENTE ET

REPARATION DES ORDINATEURS DE POCHE.

Devis gratuit sur simple demande

Tarif forfaitaire pour : changement liaison écran-clavier : 320F
le Portfolio : changement d'écran : 650F

: changement d'alimentation : 160F

SOFTWARE

PBASIC Dernière version 4.91 de B.J.GLEASON avec de nombreux utilitaires Dos. Compatible GWBASIC, ce langage est particulièrement adapté au Portfolio. Il est fourni avec une documentation pour débutants avec des exemples simples. N'occupant que 40K, il fonctionne sans problème avec un Portfolio standard.

prix : 470,00 F (sur carte rom)

ASTROPLAN Superbe logiciel d'Astrologie pour amateur ou professionnel. Fonctionne sur Portfolio standard. prix: 990,00 F (sur carte rom)

PFSOLVER de G.LEGRAND et R.NOMI. BEGUIN

Avec ce module, vous n'aurez plus rien à envier à H P !

Calculatrice en notation polonaise inversée ; graphisme et tracés de courbes ; fonctions exponentielles et polaires etc... Ces softs de qualité ont été élaborés par des ingénieurs. Fournis sur carte Rom et fonctionnent sur Portfolio standard. Documentation explicite.

Prix : 380,00 F

PBASE Superbe base de données pour Portfolio. Gère jusqu'à 30 champs différents. Indispensable ! Fonctionne sur Portfolio standard et fournie sur carte Rom. Documentation détaillée et d'apprentissage.

Prix : 690,00 F

GESTION FAMILIALE Pour gérer votre budget - cartes de crédit - chèque avec établissement des comptes et sortie sur imprimante. Fonctionne sur Portfolio standard. Fournie sur carte Rom. Prix : 490,00 F

LE CONTRÔLEUR Puissant logiciel de gestion d'emploi du temps ; avec visites clients, horaires, relances, factures etc... Sur Portfolio standard et fournie sur carte Rom. Prix : 590,00 F

JEUX 1 et 2 Voilà 2 cartes mémoire ROM ludiques exploitant pleinement les possibilités du Portfolio. Jeux de réflexion, d'adresse etc... Fonctionnent sur Portfolio standard. Fournies sur carte Rom.

Prix JEUX 1 : 380,00 F

Prix JEUX 2 : 380,00 F

UTILS 1,2,3 Voici 3 trois cartes Rom bourrées d'utilitaires du dos pour Portfolio, ainsi que de programmes divers. Elles fonctionnent toutes sur Portfolio standard.

Prix UTILS 1 : 380,00 F

Prix UTILS 2 : 380,00 F

Prix UTILS 3 : 380,00 F

MACFOLIO de R.THIBERT Avec cet ensemble, soft + câble, la communication entre le Portfolio et un Macintosh devient possible. Fonctionne avec tous les modèles de Mac. Prix : 380,00 F

FOLIOLINK Ensemble câble + soft permettant la liaison entre le Portfolio et un Atari ST. Prix : 380,00 F

ETUI de protection Magnifique étui en toile matelassée spécialement étudié pour le Portfolio: 295,00F

Garantie totale pendant une année sur tous nos produits. Frais de port colissimo+ assurance = 45F par tranche de 1500F de matériel et 60F par tranche de 4000F. Règlement par chèque/ mandat carte ou 45,00F de supplément de contre remboursement et + 110,00F pour un chronopost.

CATALOGUE SUR DEMANDE

VENTE PAR CORRESPONDANCE EXCLUSIVEMENT
(ouverture du lundi au vendredi inclus : de 9h à 13h et de 15h 30 à 19h)

Copyright 1993 MICROCARDS

Document non contractuel : les prix peuvent être modifiés sans préavis



SOMMAIRE du N°2

Janvier - Février 1994

POCHE COMPUTERS

adresse postale :

42220 BURDIGNES cedex
Tél 77.39.16.33 Fax 77.39.19.60

Rédaction : P. MARTIN

Ont participé à ce numéro :

David WINTER

Régis DUCHESNE

F.P. LANGLOIS

Dr Herbert A. KELLNER

Daniel PRATLONG

Thierry PEYCRU

Eric QUAGLIOZZI

A. MARKOUZI

Damien MARMISOLE

Loïc FIEUX

Vincent LEGROS

Pierre Philippe COUPARD

M.MANSEL

M.VIESCOU

M.THECKINIAN

Anthony TARDIVEL

J;BELIN

Noël BONNET

G.C.

JL LECOMTE

Les illustrations sont de :

Jluc MORCEAU et de

Agmeric LABURTHE

Publicité : au journal

Graphisme et mise en pages :

au journal

Imprimerie ALPHA

07100 ANNONAY

Directeur de la publication :

JL LECOMTE

Commission paritaire N° 74638

Dépôt légal : 4ème trimestre 1993

Editeur :

MICROCARDS s.a.r.l. 63520 St Dier

Copyright 1993 MICROCARDS

Toute reproduction ou représenta-

tion intégrale ou partielle, par quel-

que procédé que ce soit, des pages

publiées dans ce magazine sans

l'autorisation écrite de l'éditeur est

illicite et constitue une contrefaçon

(Loi du 11 mars 1957 art. 40 et 41 et

Code pénal art. 425).

Tous droits réservés pour tous

pays.

Les documents reçus ne sont pas

rendus et leur envoi implique l'ac-

cord tacite de l'auteur pour leur li-

bre publication.

Les indications de marques et les

adresses qui figurent dans les pa-

ges rédactionnelles de ce numéro

sont données à titre d'information,

sans but publicitaire.

Tous les noms de marque cités dans

ce numéro sont déposés.

EDITORIAL	02
LES NEWS	03
Assembleur-désassembleur sur Z88	05
Le Psion Series 3a : une révolution à l'anglaise !	08
Soft de calcul de devis sur PSION SERIE 3	10
Biorythmes sur PSION series 3	14
Le train et le series 3	15
Animez vos machines graphiques CASIO	16
Connaitre : les accumulateurs	18
HP48SX contre HP48GX	20
Initiation à l'assembleur sur HP48	21
Jeux de tetris sur HP28s	26
Résolution numérique d'équations différentielles sur HP42S.....	28
Décomposition d'un nombre en ses facteurs CASIO PB-2000	33
Programmes sur CASIO PB1000	34
Routines sonores pour CASIO PB1000	38
Trucs et astuces : jeux	41
Résolution d'équation sur TI85	45
Electronique :	
Interface 32 entrées-sorties sur port parallèle (suite)	47
(explications et schéma routé d'application)	
PORTFOLIO	
* Récupération des données du Portfolio vers Le Rédacteur 4	51
* TRUCS et ASTUCES	53
* Le Turbo Forth	56
SHARP PC 3100	
* Transmission Laplink Sharp <> PC	58
* Le connecteur du bus de sortie	59
SHARP PC E500	
* La transmission série	60
Les livres	61
Graphisme sur HP95 LX (suite)	62
Les petites annonces gratuites	64

PROCHAINE PARUTION : début AVRIL 1994



EDITORIAL

La FRANCE, malade du complexe du programmeur ?

Chers amis lecteurs,

Tout d'abord, nous tenons à vous remercier pour votre soutien enthousiaste et immédiat après la sortie du premier numéro de POCHÉ COMPUTERS. Cela prouve qu'un tel magazine manquait cruellement.

Vous avez été des centaines à nous écrire, à nous téléphoner, à nous faxer votre sympathie et vos encouragements. Des centaines également à nous faire part de vos desideratas et remarques positives concernant ce magazine. Des centaines à nous expédier spontanément vos programmes et idées, au point qu'on ne pourra pas publier tout le monde dans ce numéro. Excusez-nous de ne pas pouvoir répondre à tous, mais soyez assurés qu'on tiendra compte de vous et que les meilleurs softs seront imprimés dans les prochains numéros. Ceci nous fait chaud au coeur et nous renforce pour persévérer dans cette entreprise de presse. Continuez à nous écrire, nous en avons besoin. Mille excuses aussi pour avoir tardé à sortir ce numéro 2, mais nous sommes obligés quelquefois de passer beaucoup plus de temps que prévu pour la rédaction ; nous refusons de faire des concessions quant à la qualité du contenu du journal. Je pense que vous serez compréhensifs.

Je ne manquerai pas de répondre à certaines interrogations concernant le magazine.

Il faut savoir qu'un journal, surtout s'il est relativement marginalisé, ne fait pas gagner d'argent. Les annonceurs ne sont pas nombreux. Sur les 35F du prix du numéro, les intermédiaires (messageries de presse, distributeurs, diffuseurs, frais d'invendus) s'octroient plus de 40% au passage (ce qui est normal ; il faut bien qu'ils vivent aussi !). Il reste que l'équipe rédactionnelle travaille quasiment bénévolement, uniquement par passion. Nous vous encourageons donc à vous abonner pour nous aider le plus possible.

POCHÉ COMPUTERS est aussi votre journal.

Si j'en juge par la qualité et la quantité du courrier reçu après la parution de POCHÉ COMPUTERS, je comprends pourquoi la FRANCE est le premier (ou le deuxième : devant ou derrière les USA selon les avis) pays au monde à posséder une telle pluralité et une telle quantité de programmeurs.

Eh alors, me diriez-vous ? Et bien, ça me rend malade... Malade qu'on les ignore... Malade qu'on ne connaisse ou reconnaisse pas cet immense réservoir de matière grise ; là, à votre porte.

Depuis le programmeur de très haut niveau jusqu'au plus humble ; génial inventeur-imaginateur programmant en Basic, je leur rends hommage. L'informatique c'est aussi la vie.

Cette vie informatique française me fait quelquefois penser à cet abrutissement systématique que nous font ingurgiter de force la plupart des médias ; à savoir une pseudo culture anglo-saxonne. Cela commence par des chansons (ne retenez que le rythme et encore ! où sont les Beatles !), ça continue par des séries télévisées américaines (d'une débilité affligeante : dites-moi, paraît-il qu'il n'est plus de bon goût de déclamer qu'ARTE est très ennuyeux ; il va falloir en dire du bien) et ça se termine par un matraquage style WINDOWS.

Et l'homme dans tout ça !

La FRANCE sait fabriquer ARIANE, le TGV, l'AIRBUS, mais ne sait pas programmer... Elle ne se donne pas les moyens de permettre l'éclosion de multiples vocations de programmeurs ou d'électroniciens. Elle subit en se lamentant l'omniprésence anglo-américaine et asiatique. Elle ne sait générer dans ce domaine que des sociétés de «prestige» complètement anachroniques, inopérantes et incapables, dont la tête n'est maintenue hors de l'eau qu'à coups de milliards de francs (payés par le contribuable). Elle n'a pas compris que l'informatique et l'électronique sont des affaires quasi personnelles ou à petite échelle, exploitant au mieux les ingéniosités de chacun. Mieux vaut mille sociétés de 5 personnes que 5 sociétés de mille personnes.

Alors la FRANCE malade du complexe du programmeur ?

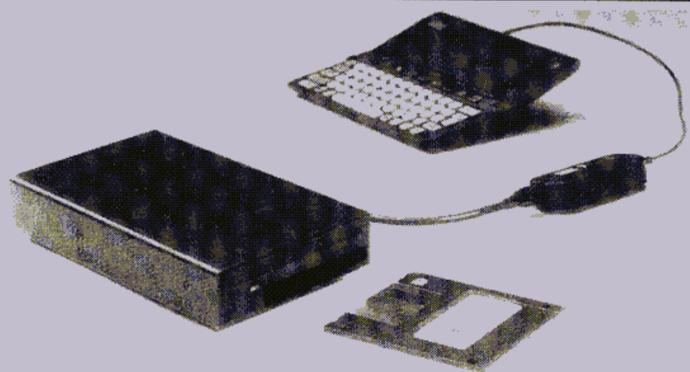
Je dirais plutôt malade du complexe d'agir. Comme toujours en France, on parle, on se gargarise de bonnes paroles sulfureuses, on politise ... mais c'est tout ! prudence est mère de sûreté...

Tenez, vous êtes jeune, compétent, plein d'idées et vous demandez un prêt pour créer une entreprise (par exemple d'informatique ou d'électronique). On vous écoute, vous comprend, vous demande de vous expliquer sur votre projet, avec force détails et papiers concernant votre gestion à venir. Puis pour terminer, on vous informe que si votre papa possédait quelque château en Espagne pour cautionner votre emprunt, votre projet de création serait sans doute génial !!!

Voilà le niveau de relation et d'aide que procure l'état français à ce merveilleux potentiel de créativité existant dans notre pays. Ne demandez pas pourquoi tant de «cerveaux» se réfugient dans l'éducation nationale ! Pour eux, une attitude différente serait suicidaire.. Allez, passez de bonnes fêtes quand même et tous à vos machines !!!

Pierre MARTIN

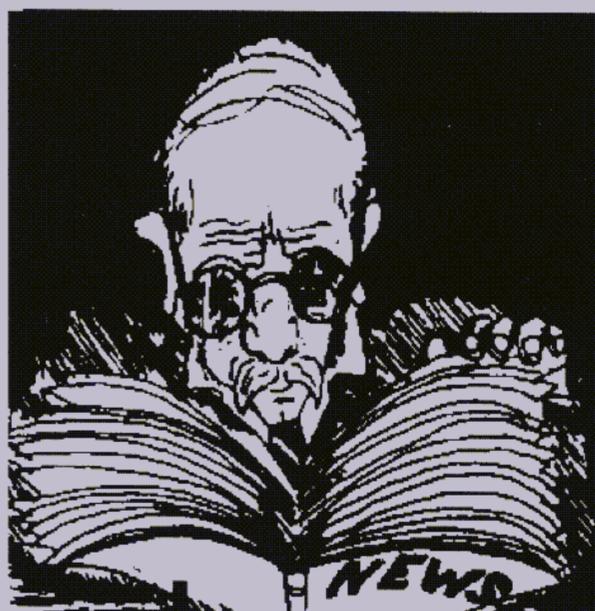
LES NEWS



NOUVEAU !

pour le PSION series 3 ou 3a

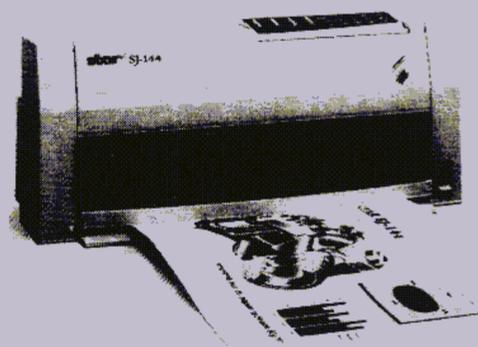
La société MICROCARDS annonce la disponibilité d'un lecteur de disquettes au format 3 1/2 pouces. Il permet sans problème de lire et d'écrire les disquettes 720KB ou 1,44MB utilisées par les PC. Il possède sa propre alimentation. Il utilise la serial link pour se connecter au series 3 et son soft interne pour transmettre programmes et fichiers. Toutefois, pour faire tourner ces programmes, il faut au préalable les charger sur une SSD. Son prix : 3027F ht



La STAR SJ-144

Une imprimante couleur et noir et blanc, à transfert thermique au prix d'une imprimante jet d'encre ?

En effet, et d'une qualité laser (360 x 360 dpi), ainsi que des couleurs irréprochables, très proches de la qualité photo.



L'impression est possible sur tous types de papiers, enveloppes et transparents. Son coût d'exploitation à la page n'est pas plus onéreux qu'avec une jet d'encre.

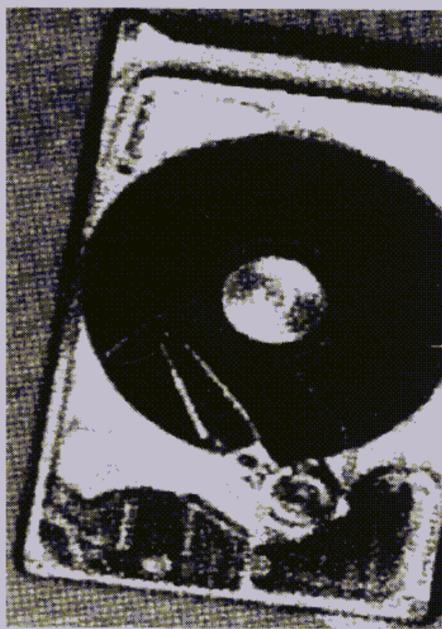
Son prix : 3940F ht.

EMULATEUR MINITEL pour PSION series 3/3a

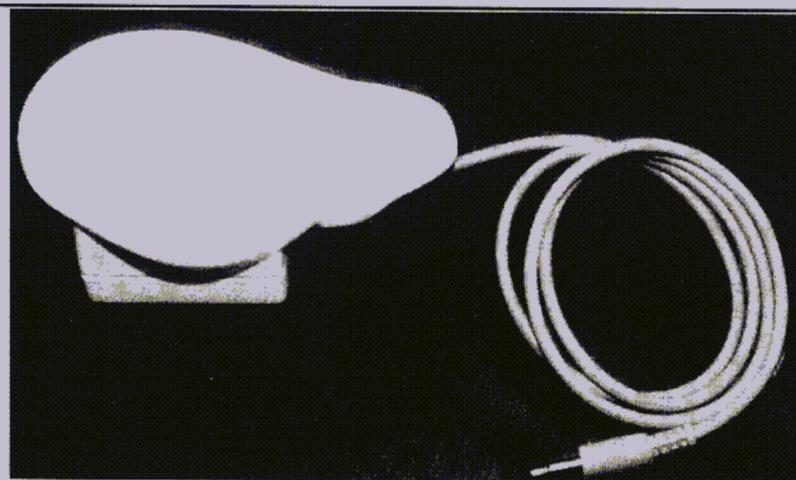
La société NOGEMA vient de mettre au point un émulateur minitel pour le series 3/3a, qui peut être accompagné d'un câble de liaison à un minitel ou d'un modem V22 et V23. Il permet l'appel de serveurs télématiques et autres serveurs dédiés. Il récupère en outre les écrans du minitel par la ligne téléphonique. Il est livré sur disquette PC ou sur SSD flash. La configuration nécessaire est d'un PSION series 3/3a et d'un serial link. Disponible chez MICROCARDS. prix à partir de 950 Fht

VOUS AVEZ DIT : MINIATURISATION

Maxtor (US) vient de réaliser l'exploit d'un disque dur de 105 MB au format d'une carte PCMCIA type III. Sous un format 1,8 pouces et de technologie Winchester, ce disque se trouvera en France d'ici 6 mois pour un prix avoisinant les 2500F. Que deviendront les



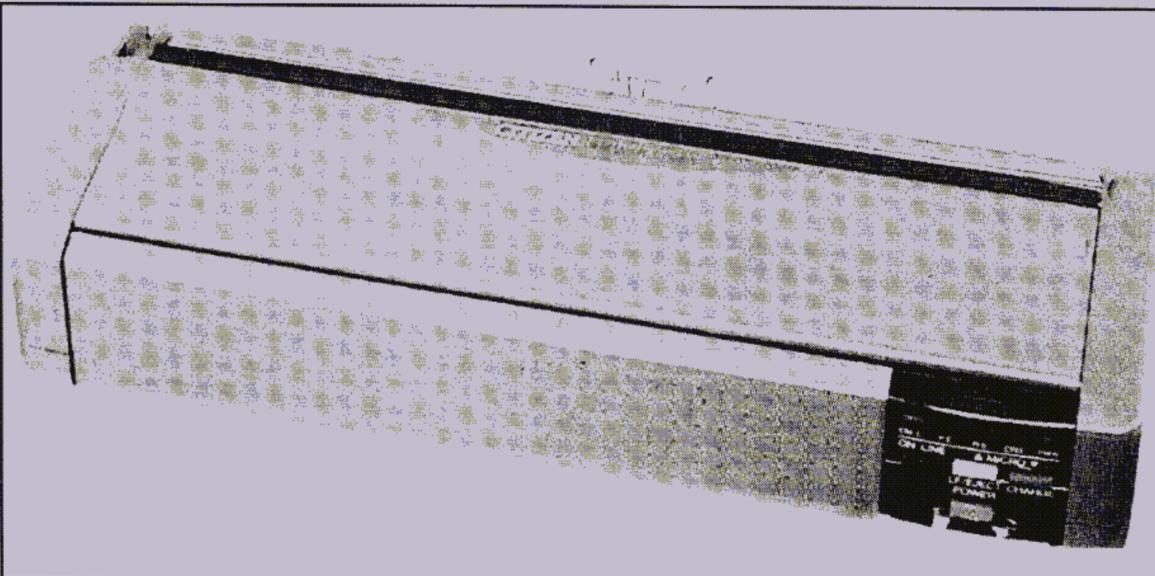
possesseurs d'appareils de type PCMCIA II ou même I ? Qu'ils se rassurent, le marché étant très important, MAXTOR prévoit de sortir un 100MB sur type II d'ici fin 1994 et un 100MB de type I d'ici fin 1995. La course avec INTEL et ses cartes Flash de 36 MB est engagée. Pour le plus grand bénéfice du consommateur final !



Votre poche a-t-il de l'oreille !

La société Blue Sky Technology, installée en Californie fabrique une oreille électronique "EarMic" en forme d'oreille plastique. Vous la collez sur le buzzer de votre écran avec un velcro et obtenez ainsi un microphone dont le gain d'amplification est très important. Associée à une carte son (il en existe sur port parallèle), vous pourrez capter et numériser tous sons. Prix : environ 1500F (hic !!!)

LES NEWS... LES NEWS... LES NEWS... LES NEWS... LES NEWS..



CITIZEN vient de sortir une nouvelle imprimante couleur portable à transfert thermique, fonctionnant sur secteur ou batteries. La Notebook printer II est idéale pour les pockets, car son poids et son encombrement est minime (de la taille d'une cartouche de cigarettes). Sa définition atteint 360 x 360 dpi. Son prix : 3000F ht.

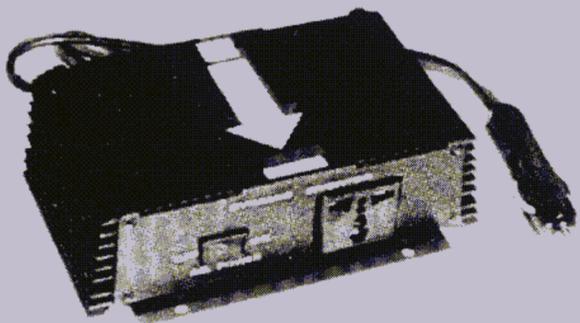
UNE INNOVATION ?

On vous les avait annoncés dans le premier numéro de POCHE COMPUTERS, et bien les voici en chair et en os !

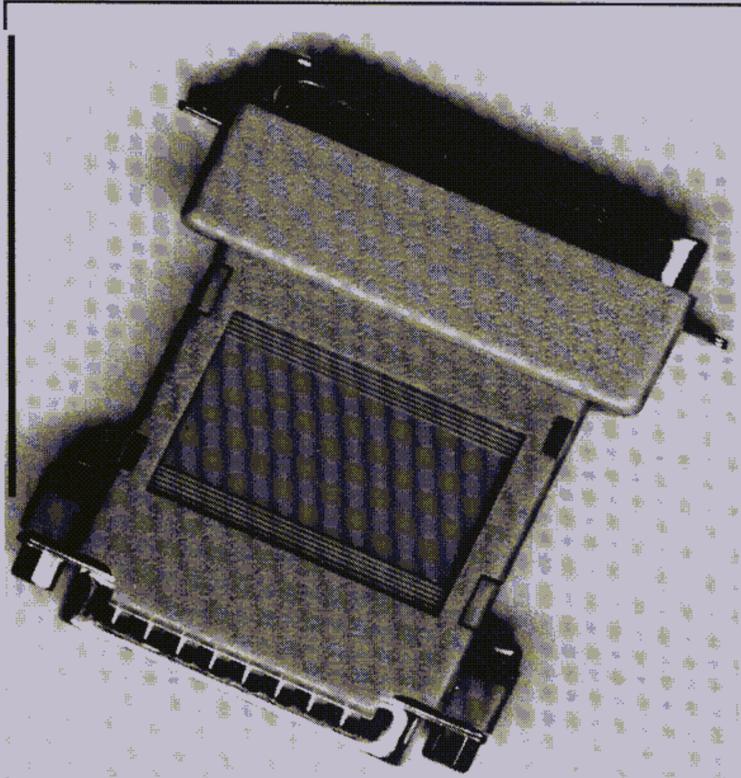
Le SHARP PC 1490 avec 64KB de mémoire (extensible à 512KB). L'affichage est de 4 lignes par 40 caractères ou de 240 par 32 dots. Les fonctions internes ont été largement améliorées par rapport aux autres modèles.

Le CASIO FX-890P marque également un réel progrès par rapport au FX-880P, très connu en France, tant au niveau des possibilités mémoire que de la rapidité et de l'affichage.

La société MICROCARDS est en pourparler pour importer ces deux produits. Vous serez tenus au courant dans les prochaines semaines.



AUTONOMIE, tel semble est le maître mot de la société GP COMM qui propose ce convertisseur qui transforme la tension 12v de la batterie de votre voiture, en 220v. Très utile pour les personnes utilisant une imprimante portable durant leurs déplacements (mais également toutes sortes d'objets électriques et bien entendu des micros de poche dont l'alimentation demande à être rechargée).



Fin d'un problème chez HP et chez PSION?

MICROCARDS propose le PARALINK qui va résoudre beaucoup de problèmes chez les utilisateurs des HP 95LX et HP 100LX, ainsi que chez les fans du PSION series 3/3a.

En effet, ces trois appareils ont un handicap majeur, à savoir qu'ils ne possèdent pas d'interface parallèle intégrée. Or, lorsqu'il faut sortir sur imprimante, ce n'est pas aisé. En effet, l'immense majorité des imprimantes ne possèdent qu'une sortie parallèle. Le convertisseur proposé ici, permet d'obtenir un port parallèle à partir du port série de votre machine. Directement accessible sans manipulation aucune, il est, en outre, auto alimenté par une pile. Il suffit de brancher ce mini boîtier sur la prise 25 points série, quelques paramètres à définir sur votre ordinateur, et le tour est joué ! prix 557Fht

MONITEUR ET DESASSEMBLEUR POUR Z88

par Thierry PEYCRU

Voici un moniteur et désassembleur pour le fameux Z88. Entièrement écrit en basic et en assembleur, il vous permettra d'accéder au magique et quelque peu ésotérique LANGAGE MACHINE

Utilisation de l'assembleur Z80 dans le BBC Basic :

. La variable p% contient le départ du programme (équivalent de ORG)

. La commande OPT permet de choisir le mode d'assemblage (cf :Manuel)

ex: 15 [OPT 3:]

. Le caractère "N" permet d'afficher une variable en hexadécimal.

ex: 80 PR. NA%

. Ouvrir un programme par un crochet (exemple) :

10 P%=&7000

20 |

30 LD HL, (&2000)

40 LD A, (HL)

50 JR Z, &7000

60 |

10 REM Moniteur Desassembleur

20 REM (C) Thierry Peycru - ESN Bordeaux

30 REM MD - V1.2 (23/10/93)

40 *NAME md

50 CLS

60 PRINT "Initialisation en cours..."

70 REM *variables

80 DIM M1\$(255)

90 DIM M2\$(255)

100 DIM M3\$(191)

110 INDEX=FALSE

120 RETORE 850

130 FOR I%=0 TO 255

140 READ MN\$

150 M1\$(I%)=MN\$

160 NEXT

170 RESTORE 1180

180 FOR I%=0 TO 63

190 READ MN\$

200 M2\$(I%)=MN\$

210 NEXT

220 DATA 064,127,"BIT "

230 DATA 128,191,"RES "

240 DATA 192,255,"SET "

250 DATA "B","C","D","E","H","L","(HL)","A"

260 RESTORE 220:GOSUB 300

270 RESTORE 230:GOSUB 300

280 RESTORE 240:GOSSUB 300

290 GOTO 390

300 READ D%,F%,MNE\$

310 FOR J%=D% TO F% STEP 8

320 RESTORE 250

330 FOR K%=0 TO 7

340 READ REG\$

350 M2\$(J%+K%)= MNE\$+STR\$((J%-D%)/8)+","+ REG\$

360 NEXT

370 NEXT

380 RETURN

390 FOR I%=0 TO 63

400 M3\$(I%)=""

410 NEXT I%

420 RESTORE 1270

430 FOR I%=64 TO 191

440 READ MN\$450 M3\$(I%)=MN\$

Quelques adresses :

0000-01A4 Zone des RST, gestion des pages de mémoire.

CC01 Tables des codes de contrôle ASCII.

D200 Routine de gestion de la mémoire...

D980 Départ du Basic BBC

DAD7 Tables des mots-clefs du basic avec leur token.

DD7D Table des messages d'erreur.

F4FA Table des mnémoniques de l'assembleur Z80.

FE45 Dédicace du Basic

FE70 Autre départ du Basic.



460 NEXT I%

470 O%=&7000

480 P%=0%

490 [OPT 3:]

500 DIM H%(15)

510 FOR I%=0 TO 15

520 CH%=48

530 IF I%>9 THEN CH%=55

540 H%(I%)=CH%+I%

550 NEXT

560 REM *fonctions

570 DEF FND2H8\$(D%)= CHR\$(H%(D%DIV&0010))+
CHR\$(D%MOD&0010);

580 DEF FND2HWS(A%)=

CHR\$(H%(A%DIV&1000))+CHR\$(H%((A%MOD&1000) DIV&0100)) +

CHR\$(H%(((A%MOD&1000) MOD&0100)DIV&0010)) +

CHR\$(H%(((A%MOD&1000) MOD&0100)MOD&0010));

590 REM *procedures

600 GOTO 1430

610 DEF PROCE(B)



```
620 IF B=0 THEN ENDPROC
630 A=OPENOUT":RAM.0/EE.CLI"
640 B$=":RAM.0/E.CLI"
650 PRINT #A,".">"+B$
660 PRINT #A,".J","LIST"+ STR$(B),"PROCF"
670 CLOSE #A
680 *CLI *:RAM.0/EE.CLI
690 ENDPROC
700 DEF PROCF
710 A=INKEY(0)
720 A=OPENIN B$
730 INPUT #A,A$,A$
740 CLOSE #A
750 A=OPENOUT B$
760 PRINT #A,".J",A$
770 PTR#A=PTR#A-1
780 BPUT#A,0
790 CLOSE #A
800 VDU 8
810 OSCLI"*CLI .<" +B$
820 ENDPROC
830 REM *data
840 REM instructions sans prefixe
850 DATA "1NOP", "3LD BC,&", "1LD (BC),A", "1INC BC", "1INC B",
"1DEC B", "2LD B,&", "1RLCA"
860 DATA "1EX AF,AF", "1ADD HL,BC", "1LD A,(BC)", "1DEC BC",
"1INC C", "1DEC C", "2LD C,&", "1RRCA"
870 DATA "2DJNZ #", "3LD DE,&", "1LD (DE),A", "1INC DE", "1INC
D", "1DEC D", "2LD D,&", "1RLA"
880 DATA "2JR #", "1ADD HL,DE", "1LD A,(DE)", "1DEC DE", "1INC
E", "1DEC E", "2LD E,&", "1RRA"
890 DATA "2JR NZ,#", "3LD HL,&", "3LD (&),HL", "1INC HL", "1INC H",
"1DEC H", "2LD H,&", "1DAA"
900 DATA "2JR Z,#", "1ADD HL,HL", "3LD HL,(&)", "1DEC HL", "1INC
L", "1DEC L", "2LD L,&", "1CPL"
910 DATA "2JR NC,#", "LD SP,&", "3LD (&),A", "1INC SP", "1INC (HL)",
"1DEC (HL)", "2LD (HL),&", "1SCF"
920 DATA "2JR C,#", "ADD HL,SP", "3LD A,(&)", "1DEC SP", "1INC A",
"1DEC A", "2LD A,&", "1CCF"
930 DATA "1LD B,B", "1LD B,C", "1LD B,D", "1LD B,E", "1LD B,H",
"1LD B,L", "1LD B,(HL)", "1LD B,A"
940 DATA "1LD C,B", "1LD C,C", "1LD C,D", "1LD C,E", "1LD C,H",
"1LD C,L", "1LD C,(HL)", "1LD C,A"
950 DATA "1LD D,B", "1LD D,C", "1LD D,D", "1LDD,E", "1LD D,H",
"1LD D,L", "1LD D,(HL)", "1LD D,A"
960 DATA "1LD E,B", "1LD E,C", "1LD E,D", "1LD E,E", "1LD E,H",
"1LD E,L", "1LD E,(HL)", "1LD E,A"
970 DATA "1LD H,B", "1LD H,C", "1LD H,D", "1LD H,E", "1LD H,H",
"1LD H,L", "1LD H,(HL)", "1LD H,A"
980 DATA "1LD L,B", "1LD L,C", "1LD L,E", "1LD L,H", "1LD L,L", "1LD
L,(HL)", "1LD L,A"
990 DATA "1LD (HL),B", "1LD (HL),C", "1LD (HL),D", "1LD (HL),E",
"1LD (HL),H", "1LD (HL),L", "1HALT", "1LD (HL),A"
1000 DATA "1LD A,B", "1LD A,C", "1LD A,D", "1LD A,E", "1LD A,H",
"1LD A,L", "1LD A,(HL)", "1LD A,A"
1010 DATA "1ADD A,B", "1ADD A,C", "1ADD A,D", "1ADD A,E", "1ADD
A,H", "1ADD ,L", "1ADD A,(HL)", "1ADD A,A"
1020 DATA "1ADC A,B", "1ADC A,C", "1ADC A,D", "1ADC A,E", "1ADC
A,H", "1ADC A,L", "1ADC ,(HL)", "1ADC A,A"
1030 DATA "1SUB B", "1SUB C", "SUB D", "1SUB E", "1SUB H", "1SUB
L", "1SUB (HL)", "1SUB A"
1040 DATA "1SBC A,B", "1SBC A,C", "1SBC A,D", "1SBC A,E", "1SBC
A,H", "1SBC A,L", "1SBC A,(HL)", "1SBC A,A"
1050 DATA "1AND B", "1AND C", "1AND D", "1AND E", "1AND H",
"1AND L", "1AND (HL)", "1AND A"
1060 DATA "1XOR B", "1XOR C", "1XOR D", "1XOR E", "1XOR H",
"1XOR L", "1XOR (HL)", "1XOR A"
1070 DATA "1OR B", "1OR C", "1OR D", "1OR E", "1OR H", "1OR L",
"1OR (HL)", "1OR A"
1080 DATA "1CP B", "1CP C", "1CP D", "1CP E", "1CP H", "1CP L",
"1CP (HL)", "1CP A"
1090 DATA "1RET NZ", "1POP BC", "3JP NZ,&", "3JP &", "3CALL
```

```
NZ,&","1PUSH BC","2ADD A,&","1RST 0H"
1100 DATA "1RET Z", "1RET", "3JP Z,&", "1@CB", "3CALL Z,&",
"3CALL &", "2ADC A,&", "1RST 8H"
1110 DATA "1RET NC3", "1POP DE", "3JP NC,&", "2OUT (&),A",
"3CALL NC,&", "1PUSH DE", "2SUB &", "1RST 10H"
1120 DATA "1RET C", "1EXX", "3JP C,&", "2IN A,(&)", "3CALL C,&",
"1@IX", "2SBC A,&", "1RST 18H"
1130 DATA "1RET PO", "1POP HL", "3JP PO,&", "1EX (SP),HL",
"3CALL PO,&", "1PUSH HL", "2AND &", "1RST 20H"
1140 DATA "1RET DE", "1JP (HL)", "3JP PE,&", "1EX DE,HL", "3CALL
PE,&", "1@ED", "2XOR &", "1RST 28H"
1150 DATA "1RET P", "1POP AF", "3JP P,&", "1DI", "3CALLP,&",
"1PUSH AF", "2OR &", "1RST 30H"
1160 DATA "1RET M", "1LD SP,HL", "3JP M,&", "1EI", "3CALL M,&",
"1@IY", "2CP &", "1RST 38H"
1170 REM instructions avec prefixe CB
1180 DATA "RLC B", "RLC C", "RLC D", "RLC E", "RLC H", "RLC L",
"RLC (HL)", "RLC A"
1190 DATA "RRC B", "RRC C", "RRC D", "RRC E", "RRC H", "RRC L",
"RRC (HL)", "RRC A"
1200 DATA "RL B", "RL C", "RL D", "RL E", "RL H", "RL L", "RL (HL)",
"RL A"
1210 DATA "RR B", "RR C", "RR D", "RR E", "RR H", "RR L", "RR (HL)",
"RR A"
1220 DATA "SLA B", "SLA C", "SLA D", "SLA E", "SLA H", "SLA L",
"SLA (HL)", "SLA A"
1230 DATA "SRA B", "SRA C", "SRA D", "SRA E", "SRA H", "SRA L",
"SRA (HL)", "SRA A"
1240 "" , "" , "" , "" , "" , "" , "" , ""
1250 DATA "SRL B", "SRL C", "SRL D", "SRL E", "SRL H", "SRL L",
"SRL (HL)", "SRL A"
1260 REM instructions avec prefixe ED
1270 DATA "1IN B,(C)", "1OUT (C),B", "1SBC HL,BC", "3LD (&),BC",
"1NEG", "1RET M", "1IM 0", "1LD I,A"
1280 DATA "1IN C,(C)", "1OUT (C),C", "1ADC HL,BC", "3LD BC,(&)", "",
"1RETI", "", "1LD R,A"
1290 DATA "1IN D,(C)", "1OUT (C),D", "1SBH HL,DE", "3LD (&),DE", "",
"", "1IM 1", "1LD A,I"
1300 DATA "1IN E,(C)", "1OUT (C),E", "1ADC HL,DE", "3LD DE,(&)", "",
"", "1IM 2", "1LD A,R"
1310 DATA "1IN H,(C)", "1OUT (C),H", "1SBC HL,HL", "3LD (&),HL", "",
"", "", "1RRD"
1320 DATA "1IN L,(C)", "1OUT (C),L", "1ADC HL,HL", "3LD HL,(&)", "",
"", "", "1RLD"
1330 DATA "1IN R,(C)", "", "1SBC HL,SP", "3LD (&),SP", "", "", "", ""
1340 DATA "1IN A,(C)", "1OUT (C),A", "1ADC HL,SP", "3LD SP,(&)", "",
"", "" , ""
1350 DATA "" , "" , "" , "" , "" , "" , "" , ""
1360 DATA "" , "" , "" , "" , "" , "" , "" , ""
1370 DATA "" , "" , "" , "" , "" , "" , "" , ""
1380 DATA "" , "" , "" , "" , "" , "" , "" , ""
1390 DATA "1LDI", "1CPI", "1INI", "1OUTI", "", "", "", ""
1400 DATA "1LDD", "1CPD", "1IND", "1OUTD", "", "", "", ""
1410 DATA "1LDIR", "1CPIR", "1INIR", "1OTIR", "", "", "", ""
1420 DATA "1LDDR", "1CPDR", "1INDR", "1OTDR", "", "", "", ""
1430 REM *principal
1440 CLS
1450 PRINT
1460 PRINT
1470 PRINT "(M)oniteur"
1480 PRINT "(D)esassembleur"
1490 K$=INKEY$(1)
1500 IF K$="" THEN GOTO 1490
1510 IF K$="M" THEN GOSUB 1540
1520 IF K$="D" THEN GOSUB 2160
1530 REM *moniteur
1550 CLS
1560 AD%=0
1570 FOR L%=1 TO 4
1580 PRINT
1590 PRINT FND2HWS$(AD%);" - ";
1600 FOR N%=AD% TO AD%+15
```



```
1610 PRINT FND2H8$(?N%) + " ";
1620 NEXT
1630 FOR N%=AD% TO AD%+15
1640 IF ?N%>31 THEN PRINT CHR$(?N%); ELSE PRINT " ";
1650 NEXT
1660 AD%=AD%+16
1670 NEXT L%
1680 K$=INKEY$(1)
1690 IF K$="" THEN 1680
1700 IF K$=" " OR K$="N" THEN 1570
1710 IF K$="P" THEN AD%=AD%&80:GOTO 1570
1720 IF K$="A" THEN GOSSUB 1770:GOTO 1570
1730 IF K$="S" THEN GOSSUB 1830:GOTO 1570
1740 IF K$="W" THEN GOSSUB 2020:GOTO 1570
1750 IF K$="F" THEN RETURN
1760 GOTO 1680
1770 REM *nouvelle adresse
1780 PRINT TAB(73,1);"Adresse"
1790 INPUT TAB(73,2);AD$;
1800 AD%=EVAL("&"+AD$)
1810 PRINT TAB(0,0);"";
1820 RETURN
1830 REM *recherche chaine
1840 PRINT TAB(73,1);"Chaîne"
1850 INPUT TAB(73,2);ST$
1860 PRINT TAB(73,3);"Adresse"
1870 INPUT TAB(73,4);AA$
1880 AA%=EVAL("&"+AA$)
1890 L%=LEN(ST$)
1900 PO%=ASC(LEFT$(ST$,1))
1910 LOOP=TRUE
1920 REPEAT
1930 REM XX$=LEFT$(AA%,L%)
1940 IF ?AA%=PO% THEN GOSUB 1990
1950 PRINT TAB(73,5);"&";FND2HW$(AA%)
1960 AA%=AA%+1
1970 UNTIL LOOP=FALSE
1980 RETURN
1990 XX$=LEFT$(AA%,L%)
2000 IF XX$=ST$ THEN AD%=AA%:LOOP=FALSE
2010 RETURN
2020 REM *recherche mot
2030 PRINT TAB(73,2);"Mot"
2040 INPUT TAB(73,3);W$
2050 PRINT TAB(73,4);"Adresse"
2060 INPUT TAB(73,5);A$
2070 W1%=EVAL("&"+LEFT$(W$,2))
2080 W2%=EVAL("&"+RIGHT$(W$,2))
2090 AA%=EVAL("&"+A$)
2100 LOOP = TRUE
2110 REPEAT
2120 IF ?AA%=W1%
AND?(AA%+1)=W2% THEN AD%=AA%:LOOP=FALSE
2130 AA%=AA%+1
2140 UNTIL LOOP=FALSE
2150 RETURN
2160 REM *desassembleur
2170 CLS
2180 INPUT "Adresse";AD$
2190 AD%=EVAL("&"+AD$)
2200 REM boucle desa
2210 PRINT FND2HW$(AD%);" ";
2220 B%=?AD%
2230 MN$=M1$(B%)
2240 IF B%=&DD THEN GOSUB 2640
2250 IF B%=&FD THEN GOSUB 2690
2260 IF B%=&CB AND INDEX=FALSE THEN GOSUB 2520
2270 IF B%=&ED THEN GOSUB 2580
2280 N%=EVAL(LEFT$(MN$,1))
2290 IF N%=1 THEN CC$=""
```

```
2300 IF N%=2 THEN CC$=FND2H8$(?(AD%+1))
2310 IF N%=3 THEN CC$=FND"H8$(?(AD%+2)) +
FND2H8$(?(AD%+1))
2320 IF N%=3 AND INDEX=TRUE THEN CC$=FND2H8$(?(AD%+2))
2330 PN%=INSTR(MN$,"&")
2340 IF PN%=0 THEN PN%=LEN(MN$):CC$=""
2350 FM%=LEN(MN$)-PN%
2360 IF FM%=0 THEN FM$="" ELSE FM$=RIGHT$(MN$,FM%)
2370 PR%=INSTR(MN$,"#")
2380 IF PR%<>0 THEN GOSUB 2950
2390 Z$=MID$(MN$,2,PN%-1)+CC$+FM$
2400 FOR I%=0 TO N%-1
2410 PRINT FND2H8$(?(AD%+I%));" ";
2420 NEXT
2430 PRINT TAB(24);Z$
2440 AD%=AD%+N%
2450 REPEAT
2460 K$=INKEY$(1)
2470 IF K$="F" THEN GOTO 1430
2480 UNTIL K$<> ""
2490 INDEX=FALSE
2500 MN$=""
2510 GOTO 2200
2520 REM ttt prefixe CB
2530 PRINT "CB ";
2540 AD%=AD% +1
2550 B%=?AD%
2560 MN$="1"+M2$(B%)
2570 RETURN
2580 REM ttt prefixe ED
2590 PRINT "ED ";
2600 AD%=AD%+1
2610 B%=?AD%
2620 MN$=M3$(B%)
2630 RETURN
2640 REM ttt index IX
2650 PRINT "DD "
2660 IZ$="IX"
2670 GOSUB 2470
2680 RETURN
2690 REM ttt index IY
2700 PRINT "FD ";
2710 IZ$="IY"
2720 GOSUB 2740
2730 RETURN
2740 REM ttt index IZ
2750 INDEX=TRUE
2760 AD%=AD%+1
2770 B%=?AD%
2780 MT$=M1$(B%)
2790 IF B%=&CB THEN GOSUB 2920
2800 N%=EVAL(LEFT$(MT$,1))
2810 PIZ%=INSTR(MT$,"(HL)")
2820 IF PIZ%=0 THEN GOTO 2880
2830 REM il existe un index IZ+dd
2840 N%=N%+1
2850 REG$="( "+IZ$+" "+FND2H8$(?(AD%+1))+")"
2860 MN$=STR$(N%)+MID$(MT$,2,PIZ%-2)+REG$+RIGHT$(MT$,LEN(MT$)-PIZ%-3)
2870 RETURN
2880 PIZ%=INSTR(MT$,"HL")
2890 MN$=STR$(N%)+MID$(MT$,2,PIZ%2)+IZ$+RIGHT$(MT$,LEN(MT$)-PIZ%-1)
2900 INDEX=FALSE
2910 RETURN
2920 REM tt index IZ suivit de prefixe CB
2930 MT$="2"+M2$(?(AD%+2))
2940 RETURN
2950 REM ttt adressage relatif
2960 AR%=?AD%+1
2970 IF AR%>127 THEN ADR%=AD%+AR%-254
2980 IF AR%<128 THEN ADRE%=AD%+AR%+2
2990 CC$=FND2HW$(ADR%)
3000 RETURN
```

PSION series 3a : SI PROCHE...

ET POURTANT SI DIFFERENT...

par Vincent LEGROS

Dans le premier numéro de Poche Computers, si vous ne le connaissiez pas encore, vous découvriez avec surprise le Psion Series 3. Surpris certes par les fabuleuses possibilités de cette petite machine...

Mais cette fois-ci, vous ne pourrez résister à ce que vous réserve le dernier né de chez PSION, à savoir le petit frère du series 3, le fabuleux **Series 3a**...

Parlons peu, mais parlons bien ! Extérieurement, les deux machines sont rigoureusement identiques si ce n'est le petit «a» placé à côté du nom Psion Series 3 de la face avant de l'appareil. Mais prenons ensemble un risque, et au risque de devoir passer votre ancienne machine (le Series 3 par exemple, mais qui n'est en aucun cas considéré comme du matériel obsolète) à la cuisine à des fins ménagères (déclanchement d'appareils électriques par exemple), entrons à l'intérieur de la bête. MIRACLE, l'écran est plus grand ! Que dis-je ? Il est plus grand, mais en fait il n'a rien à voir avec l'ancien... Mais avant d'entrer sauvagement dans les entrailles de la machine, nous remarquons que la touche Programmation OPL (touche tactile sur le Series 3) a disparue !!! Pas de panique, elle est remplacée par la touche Tableur, ce qui nous permet d'en déduire que désormais ce logiciel se trouve dans la ROM !!! Et les programmeurs peuvent toujours entrer dans l'éditeur OPL mais désormais en se plaçant sur l'icône de l'écran Système. Autre changement fondamental : Une autre touche a été remplacée: Verr MAJ n'est plus, elle a été substituée par une touche Diamant dont l'utilité sera décrite un peu plus loin. La possibilité de bloquer le clavier en Majuscules existe toujours, et pour ce faire, il suffit de presser simultanément les touches Psion + Diamant.

Pour tester le clavier, il semblerait subtil d'allumer cette petite machine... Allons-y ... Les choses prennent une allure inattendue, dès que l'écran système s'allume... Sans vouloir vexer le Series 3, Monsieur Series 3a est encore mieux en étant à la fois semblable... Mieux car son écran est quatre fois plus grand, qu'il existe désormais un niveau



de gris permettant un effet d'ombre sur les icônes (qui sont en 48*48 pixels !), mais identique dans l'utilisation.

Le gestionnaire de programmes est très bien réalisé et d'une facilité d'exploitation exceptionnelle. Le clavier s'est encore un peu amélioré et on peut désormais frapper des textes à une allure tout à fait convenable sans faire trop d'oublis de caractères. De là à n'aller en cours qu'avec son Series 3a...

Mais, que vois-je ? Il est marqué au dessus de l'écran "16 bit CPU" (comme le Series 3), "MULTI-TACHES" (comme le Series 3), "SYSTEME AUDIO NUMERIQUE" (Nouveauté!!!). Et oui, nous proposons de parler peu, mais de parler bien: Le Series 3a digitalise votre voix ou tous sons grâce à un petit micro subtilement caché dans le boîtier, et les restitue d'une manière étonnante. Seul inconvénient, les sons prennent beaucoup de place: 8 Kilo octets par seconde !!! Les utilitaires de compression n'arrivent pas à réduire la place prise par les fichiers son.

C'est donc à ce moment précis que notre intérêt va se porter sur les capacités de stockage de la machine (Disque interne). Le Series 3a a gagné dans cette cure de rajeunissement 256 Ko, c'est à dire que le modèle 128 Ko passe à 256Ko, et comme vous l'aurez deviné avec stupéfaction, le modèle 256 Ko passe donc à 512 Ko !!!

Bien sûr, les deux lecteurs existent toujours, et permettent de stocker des données sur des extensions mémoires du type RAM disque ou FLASH disque (de 128Ko à 2Mo par lecteur). On commence à se demander comment PSION a pu organiser cette machine pour mettre toutes ces caractéristiques dans un si petit boîtier...

Puisque l'on commence à approcher le vif du sujet, parlons un peu des données constructeur relatives à cet appareil :

- * Le processeur ne change pas : toujours un NECV30H mais qui tourne à 7,68 Mhz cette fois ci. (soit le double)

- * La ROM fait près de 1 Méga (normal lorsque l'on sait tous les logiciels qui dorment en cette machine...

- * ECRAN 480*160 pixels avec 1 niveau de gris

- * Poids 275g piles incluses

- * Logiciels Intégrés : Enregistreur numérique, Gestionnaire de fiches, Traitement de textes, Agenda, Heure (6 alarmes différentes), Monde, une calculatrice, le Tableur, et le langage de programmation OPL.

Les logiciels internes à la machine, sont véritablement exceptionnels, simples d'emploi et puissants. Ajoutons à ces éloges la compatibilité des informations avec votre ordinateur de bureau...



LE TABLEUR

Excellent petit tableur compatible directement avec Lotus 123 et Excel sur PC ou Mac. La touche diamant permet de passer du mode tableur au mode graphique. La fonction Zoom existe également.

LE TRAITEMENT DE TEXTES

Compatible avec MS Winword (format RTF). Les fonctions de style sont assez puissantes et améliorées par rapport au Series 3. Un reproche, l'impossibilité de faire du multi colonnage. (Notez que cet article a été tapé dans le métro Parisien, aux heures de pointes... On peut donc bien parler de compatibilité !)

L'AGENDA

6 modes accessibles avec la touche diamant : Jour, semaine, année (calendrier), choses à faire, anniversaires, et une liste des enregistrements. Les alarmes sont des sons digitalisés en ROM ou vos propres sons réalisés avec l'enregistreur de Sons.

MONDE

Planisfère donnant des renseignements utiles sur différentes villes du Monde (indicatif téléphonique, heures lever/coucher du soleil, distance par rapport à une ville que l'utilisateur aura fixée comme base, décalage horaire...) Bien amélioré par rapport à la version Series 3.

CALCULATRICE

Pas mieux qu'avant !!! Les fonctions principales sont présentes, mais vous pouvez charger vos propres fonctions sous forme de modules que vous créez facilement grâce au langage de programmation OPL...

LANGAGE OPL

Langage dont la structure rappelle le Pascal (procédures, fonctions) et les instructions, le Basic avancé. Très riche !!! Enfin, pour terminer, toutes les applications et les périphériques du Series 3 sont directement compatibles avec le Series 3a. Un appareil à découvrir et à adopter absolument...

LE GESTIONNAIRE DE FICHES

Quelques changements par rapport à la version Series 3. On peut désormais faire une recherche sur un libellé. Par exemple, pour rechercher Monsieur PASCAL (nom de famille), vous faites une recherche sur le libellé NOM: et le gestionnaire de fiches ne vous sortira plus les fiches relatives aux gens dont le prénom est Pascal, ou celles où le libellé Notes: comporte la mention «Grand adepte de Pascal», si tenté que certains de vos amis en soient effectivement des adeptes.

La recherche d'une fiche sur un grand fichier a été accélérée. La touche diamant permet par une simple pression de passer du mode Ajouter Fiche à Trouver fiche. De plus Psion+Z permet d'avoir un ZOOM : les caractères changent de taille et vous offrent 4 tailles d'affichage.

ORGANISATION **Deb's**

Société Organisation Deb's
 Contact Thierry Desbruères
 Adresse 139, av Charles de Gaulle
 92200 Neuilly sur Seine
 Téléphone (1) 47 45 20 44
 Fax (1) 46 24 76 81

Trouver: Series 3a 1/1 Jeu 21

Repert(A) Berlitz Nancy(A) Agenda(B)
 Repvide(A) Codes(A) Concours(B)
 Words(A) Debsnews(B)
 Délég(B)

282 utilisés 238 livres

Une interface conviviale pour le 3a. Etre petit, ce n'est pas être étanche

AI	Conversion h/m en h.1/100h	h/m =>	78,57	78,95
	Désignation des pièces	Référence	H. A. mont.	C. A. mont.
	Axe rotor	658723	200,00	400
	Axe rotor arrière	ARA37890	100,00	néant
	Axe rotor principal	ARP456	350,00	600
	Cellule	C345219	100,00	néant
	Démarrateur	D459007	néant	50
	Embrayage rotor	E4319	150,00	300
	Rotor arrière	RA1907	500,00	néant
	Turbine	95F6786	1 200,00	2 400

Dans le Serie 3a, un tableur, un vrai !

Turbine(B) Screen Cardio Change(A)
 Ventil(B) Jackpot(A) Logof(A)
 Mafalda(A) Marilyn(A)

281 utilisés 231 livres

Une interface conviviale pour le 3a. Etre petit, ce n'est pas être étanche.

Fichier Editer Trouver Texte Paragraphes Styles Spécial

Grand Concours Deb's PSION

Tous et toutes à vos PSIONS, le clavier, la tête légèrement penchée, fesses bien calées, le starter a sorti le doigt sur la détente. La carotte est doucement au loin.

Préférences Maj+F
 Zoom avant Maj=Z
 Zoom arrière Maj=Z
 Config Imprimante Maj=C
 Aperçu Maj=U
 Imprimer Maj=I
 Quitter sans sauvegarder Maj=Q
 Quitter en sauvegardant Maj=Q

Ven 5 Novemb

Détails d'alarme

Alarme Oui
 Délai avant 00:30
 Alarme à 12:30
 Nbre de jours avant 8
 Sonorité +Fanfare+

Tester sonorité Confirmer

Menu Entrée

Faites sonner les clairons, il est arrivé, qui ? , le 3a

Société: Organisation Deb's
 Nom: Thierry Desbruères
 Pro: 47 45 20 44, fax: 46 24 76 81
 Ad. Pro: 139, avenue Charles de Gaulle
 92200 Neuilly

Notes: vous propose la prise en dépôt de votre Series 3 pour mise en vente sur le marché d'occasion à travers son réseau et à vos conditions contre l'achat d'un Series 3a.

Trouver: deb's 462/462

Novembre 1993

San 6 Entren bourse mois écoulé > Δ
 Din 7 15:00 Club Psion Noisy Le Grand (Φ19:00) Δ
 Mar 9 20:00 Diner Pipi à confirmer (Φ23:00) Δ
 Mer 10 A Anne (1939, 54 ans) > Δ
 Jeu 11 06:00 Anniversaire 1918 > Δ
 Jeu 18 A Robert (1936, 63 ans) > Δ

Décembre 1993

Mer 22 Dépistage SIDA négatif > Δ
 San 25 06:00 Noël > Δ
 Ven 31 Rentrer derniers cours Bourse de l'année. > Δ

Janvier 1994

San 1 A Férié Nouvel An > Δ
 A Rentrer cours Bourse du premier jour de l'année. > Δ

Si jamais je suis séropositif, le Série 3a n'y est pour rien, juré !

Tous les produits PSION sont en stock chez DEB'S (logiciels, câbles, mémoires et divers) ; seuls les Séries 3a sont parfois sujets à délais.



CALCUL DE DEVIS sur PSION series 3

par Daniel TCHEKINIAN

Écrit pour le travail dans le plastique, ce logiciel permet de calculer un prix de revient au mètre HT et TTC, suivant des épaisseurs et des couleurs différentes du PMMA, PC (Altuglas, Plexiglas), ainsi que le prix de la main d'œuvre pour le polissage manuel ou mécanique, et la fabrication de capots, suivant un taux horaire à la minute près. Il est facile de l'adapter à un autre corps de métier (métaux, bois, tissus etc...) Ce programme peut, de toute façon avoir un but pédagogique, et montrer à quoi peut bien servir, sur un plan pratique, un ordinateur de poche. L'auteur n'a pas la prétention d'avoir créé un logiciel parfait, il engage même les personnes intéressées à le modifier et à l'améliorer. Vous pouvez le contacter au (1) 42.80.38.63 à PARIS.

```
PROC devis:
LOCAL h$(18),a$(5)
GLOBAL k%,q%,lo,la,q,a(30),x
GLOBAL ep,ef,ec,pt,tx,ptr,pf,apr,avr
GLOBAL mt,pd,mt1,mt2,mt3,mt4,mt5
GLOBAL pd1,pd2,pd3,pd4,pd5,ppd(30)
GLOBAL
t$(3),spl(30),pma,pma1,sxt(30),pdx(30),pxt,pxt1
GLOBAL
sch(30),ph(30),pch,pch1,spc(30),pdc(30),ppc,ppc1
GLOBAL
smi(30),pmi(30),pmir,pmir1,pcapot
GLOBAL cl%,fm%,t,g%,qsp%,qs&
GLOBAL
d,ps,eps%,ep1,co%,qs%,psc,pscq
GLOBAL
d&,pm,p4,tr,apr1,temp%,nom$(128)
GLOBAL
nm$(255),dat$(30),piec$(255)
GLOBAL
prix$(30),moul$(30),not$(255),titre$(30)
dat$=LEFT$(DATIM$,15)
CLS
gBORDER 1
gAT 93,40 :gSTYLE 9 :gPRINT «*
DEVIS *»
gAT 37,65 :gSTYLE 0 :gFONT 1
:gPRINT «+ Vers 1.0-93 TCHEKINIAN
Daniel»
h$=»PDXLMCTURWSAOIFEKQ»
DO
BUSY «<Menu>»,3,4
k%=GET
IF k%=$122
BUSY OFF
mINIT
mCARD «Matière»,»Pmma
coulé»,%P,»Pmma choc»,%D,»Pmma
XT»,%X,»Polycarbonate»,%L,»Miroir»,%M
mCARD
«Travail»,»Capot»,%C,»Travail»,%T,»Polissage»,%J
mCARD
«Divers»,»Remise»,%R,»Majoration»,%W,»1/2
sphère»,%S,»Colles & Entretien»,%A
mCARD
«Fichier»,»Fiches»,%O,»Image»,%I
mCARD «Spécial»,»Prix
final»,%F,»Mettre à
<0>»,%E,»Surfaces/
Poids»,%K,»Quitter»,%Q
k%=MENU
IF k%
a$=»p»+CHR$(k%)
@(a$):
ENDIF
ELSEIF k% AND $200
k%=(k%-$200) AND $FFDF
k%=LOC(h$,CHR$(k%))
IF K%
a$=»p»+MID$(h$,k%,1)
@(a$):
ENDIF :ELSE :ENDIF
UNTIL 0
ENDP
PROC pP:
REM * Pmma *
CLS
GIPRINT «PMMA COULE»
dINIT «* PMMA coulé *»
dTEXT «»,»Format
standard(T0)», $402
dTEXT «»,»Format découpé
(TX)», $402
dTEXT «»,»Transformation (TS)», $402
g%=DIALOG
IF g%=2 :t$=»T0" :fm%=0
ELSEIF g%=3 :t$=»TX" :fm%=20
ELSEIF g%=4 :t$=»TS" :fm%=50
ENDIF
dINIT «Sélectionnez une entrée»
dTEXT «»,»Incolore», $402
dTEXT «»,»Blanc Diffusant», $402
dTEXT «»,»Couleur», $402
dTEXT «»,»Fumé», $402
g%=DIALOG
IF g%=2 :cl%=0
ELSEIF g%=3 :cl%=5
ELSEIF g%=4 :cl%=10
ELSEIF g%=5 :cl%=10
ENDIF
dt:
CLS
AT 1,1 :PRINT «Nombre
d'épaisseur(s)?:» :INPUT x
IF x=0
GOTO dt
ENDIF
DO
qt:
ep:
PRINT «Longueur :»: :INPUT lo
PRINT «Largeur :»: :INPUT la
PRINT «Quantité :»: :INPUT q
spl(x)=(lo*la)*q
mt1=spl(x)+1
ppd(x)=(spl(x)*1.2)*ppd(x)
pd1=ppd(x)+1
ec=a(x)*cl%/100 :a(x)=a(x)+ec
ef=a(x)*fm%/100 :a(x)=a(x)+ef
a(x)=spl(x)*a(x)
x=x-1
UNTIL x=0
gAT 133,10 :gBOX 107,29 :gAT 140,22
gPRINT «Surf.=»,FIX$(mt1,2,-
10),»m2"
gAT 145,34
gPRINT «Pds=»,FIX$(pd1,2,-10),»Kg»
pplex:
ENDP
PROC qt:
epais:
AT 1,2 :PRINT «Epaisseur:»: :TRAP
INPUT a(x)
ep=a(x)
IF a(x)<>1.5 AND a(x)<>2 AND
a(x)<>3 AND a(x)<>4 AND a(x)<>5
AND a(x)<>6 AND a(x)<>8 AND
a(x)<>10 AND a(x)<>12 AND a(x)<>15
AND a(x)<>20 AND a(x)<>25
IF a(x)<>30 AND a(x)<>35 AND
a(x)<>40 AND a(x)<>50 AND a(x)<>60
AND a(x)<>70 AND a(x)<>80 AND
a(x)<>100 AND a(x)<>110 AND
a(x)<>120 AND a(x)<>150 AND
a(x)<>180 AND a(x)<>200 AND
a(x)<>250
GOTO epais
ENDIF :ENDIF
ENDP
PROC ep:
IF a(x)=1.5 :ppd(x)=a(x) :a(x)=268.19
ELSEIF a(x)=2 :ppd(x)=a(x)
:a(x)=268.19
ELSEIF a(x)=3 :ppd(x)=a(x)
:a(x)=242.8
ELSEIF a(x)=4 :ppd(x)=a(x)
:a(x)=290.93
ELSEIF a(x)=5 :ppd(x)=a(x)
:a(x)=343.42
ELSEIF a(x)=6 :ppd(x)=a(x)
:a(x)=412.12
ELSEIF a(x)=8 :ppd(x)=a(x)
:a(x)=549.47
ELSEIF a(x)=10 :ppd(x)=a(x)
:a(x)=686.83
ELSEIF a(x)=12 :ppd(x)=a(x)
:a(x)=824.19
ELSEIF a(x)=15 :ppd(x)=a(x)
:a(x)=1030.28
ELSEIF a(x)=20 :ppd(x)=a(x)
:a(x)=1373.67
ELSEIF a(x)=25 :ppd(x)=a(x)
:a(x)=1530
ELSEIF a(x)=30 :ppd(x)=a(x) :GOTO
a30
ELSEIF a(x)=35 :ppd(x)=a(x) :GOTO
a30
ELSEIF a(x)=40 :ppd(x)=a(x) :GOTO
a30
ELSEIF a(x)=50 :ppd(x)=a(x) :GOTO
a50
ELSEIF a(x)=60 :ppd(x)=a(x) :GOTO
a50
ELSEIF a(x)=70 :ppd(x)=a(x) :GOTO
a50
ELSEIF a(x)=80 :ppd(x)=a(x) :GOTO
a50
ELSEIF a(x)=85 :ppd(x)=a(x) :GOTO
a50
ELSEIF a(x)=90 :ppd(x)=a(x) :GOTO
a90
ELSEIF a(x)=100 :ppd(x)=a(x) :GOTO
a90
ELSEIF a(x)=110 :ppd(x)=a(x) :GOTO
a90
ELSEIF a(x)=120 :ppd(x)=a(x) :GOTO
a90
ELSEIF a(x)=150 :ppd(x)=a(x) :GOTO
a90
ELSEIF a(x)=180 :ppd(x)=a(x) :GOTO
a180
ELSEIF a(x)=200 :ppd(x)=a(x) :GOTO
a180
ELSEIF a(x)=250 :ppd(x)=a(x) :GOTO
a180
ENDIF
RETURN
a30:
a(x)=75.35*1.2*a(x) :RETURN
a50:
a(x)=82.19*1.2*a(x) :RETURN
a90:
a(x)=102.83*1.2*a(x) :RETURN
a180:
a(x)=127.38*1.2*a(x) :RETURN
ENDP
PROC pplex:
tx=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)
+a(9)+a(10)+a(11)+a(12)+a(13)+a(14)+a(15)
+a(16)+a(17)+a(18)+a(19)+a(20)+a(21)+a(22)
+a(23)+a(24)+a(25)+a(26)+a(27)+a(28)+a(29)
+a(30)
pma=tx :pma1=pma
AT 1,6 :PRINT «Prix du
pmma»,»(«:t$;»)»,»»,FIX$(pma1,2,-
10)
AT 1,7 :PRINT
«T.V.A._____»,FIX$(pma1*18.6/
100,2,-10)
gAT 0,66 :gLINEBY 280,0
AT 1,9 :PRINT
«T.T.C._____»,FIX$(pma1*1.186,2,-
10)
ENDP
PROC pD:
REM * Pmma choc *
CLS
GIPRINT «PMMA CHOC»
dINIT «* PMMA CHOC *»
dTEXT «»,»Format standard», $402
dTEXT «»,»Format découpé», $402
g%=DIALOG
IF g%=2 :t$=»T0" :fm%=0
ELSEIF g%=3 :t$=»TX" :fm%=20
ENDIF
dt:
AT 1,1 :PRINT «Nombre
d'épaisseur(s)»:» :INPUT x
IF x=0
GOTO dt
ENDIF
DO
qtcho:
epch:
PRINT «Longueur :»: :INPUT lo
PRINT «Largeur :»: :INPUT la
PRINT «Quantité :»: :INPUT q
sch(x)=(lo*la)*q :mt2=sch(x)+1
ph(x)=(sch(x)*1.2)*ph(x)
pd2=ph(x)+1
ef=a(x)*fm%/100
a(x)=a(x)+ef :a(x)=sch(x)*a(x)
x=x-1
UNTIL x=0
gAT 133,10
gBOX 107,29
gAT 140,22
gPRINT «Surf.=»,FIX$(mt2,2,-
10),»m2"
gAT 145,34
gPRINT «Pds=»,FIX$(pd2,2,-10),»Kg»
pchoc:
ENDP
PROC qtcho:
epais:
AT 1,2 :PRINT «Epaisseur:»: :TRAP
INPUT a(x)
ep=a(x)
IF a(x)<>3 AND a(x)<>4 AND a(x)<>5
AND a(x)<>6 AND a(x)<>8 AND
a(x)<>10 AND a(x)<>12 AND a(x)<>15
GOTO epais
ENDIF
ENDP
PROC epch:
IF a(x)=3 :ph(x)=a(x) :a(x)=356.18
ELSEIF a(x)=4 :ph(x)=a(x)
:a(x)=447.17
ELSEIF a(x)=5 :ph(x)=a(x)
:a(x)=526.45
ELSEIF a(x)=6 :ph(x)=a(x)
:a(x)=631.74
ELSEIF a(x)=8 :ph(x)=a(x)
:a(x)=830.19
ELSEIF a(x)=10 :ph(x)=a(x)
:a(x)=1053.04
ELSEIF a(x)=12 :ph(x)=a(x)
:a(x)=1263.45
ELSEIF a(x)=15 :ph(x)=a(x)
:a(x)=1586.46
ENDIF
ENDP
```



CALCUL DE DEVIS sur PSION series 3... CALCUL DE DEVIS sur PSION series3...

```
PROC pchoc:
tx=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+
+a(9)+a(10)+a(11)+a(12)+a(13)+a(14)+a(15)+
a(16)+a(17)+a(18)+a(19)+a(20)+a(21)+a(22)+
+a(23)+a(24)+a(25)+a(26)+a(27)+a(28)+a(29)+
+a(30)
pch=tx :pch1=pch
AT 1,6 :PRINT «Prix du
choc»,»(«:t$;»);»»,FIX$(pch1,2,-10)
AT 1,7 :PRINT
«T.V.A. _____:»,FIX$(pch1*18.6/
100,2,-10)
gAT 0,66 :gLINEBY 280,0
AT 1,9 :PRINT
«T.T.C. _____:»,FIX$(pch1*1.186,2,-
10)
ENDP
PROC pX:
REM * Pmma XT *
CLS
GIPRINT «PMMA EXTRUDE»
dINIT «* PMMA XT *»
dTEXT «»,»Format
standard(T0)», $402
dTEXT «»,»Format découpé
(TX)», $402
dTEXT «»,»Transformation (TS)», $402
g%=DIALOG
IF g%=2 :t$=»T0» :fm%=0
ELSEIF g%=3 :t$=»TX» :fm%=20
ELSEIF g%=4 :t$=»TS» :fm%=50
ENDIF
dINIT «Sélectionnez une entrée»
dTEXT «»,»Incolore», $402
dTEXT «»,»Blanc diffusant &
Opaque», $402
g%=DIALOG
IF g%=2 :cl%=0
ELSEIF g%=3 :cl%=5
ENDIF
dt:
AT 1,1 :PRINT «Nombre d'épaisseur(s)
d'XT ?»:»:»: :INPUT x
IF x=0
GOTO dt
ENDIF
DO
qtXT:
epXT:
PRINT «longueur :»:»: :INPUT lo
PRINT «Largeur :»:»: :INPUT la
PRINT «Quantité :»:»: :INPUT q
sxt(x)=(lo*la)*q
mt3=sxt(x)+1
pdx(x)=(sxt(x)*1.2)*pdx(x)
pd3=pdx(x)+1
ec=a(x)*cl%/100
a(x)=a(x)+ec :ef=a(x)*fm%/100
:a(x)=a(x)+ef
a(x)=sxt(x)*a(x)
x=x-1
UNTIL x=0
gAT 133,10 :gBOX 107,29 :gAT 140,22
gPRINT «Surf.=»,FIX$(mt3,2,-
10),»m2"
gAT 145,34
gPRINT «Pds=»,FIX$(pd3,2,-10),»Kg»
pXT:
ENDP
PROC qtXT:
epais:
AT 1,2 :PRINT «Epaisseur:»:»: :TRAP
INPUT a(x)
ep=a(x)
IF a(x)<>1.5 AND a(x)<>2 AND
a(x)<>2.5 AND a(x)<>3 AND a(x)<>4
AND a(x)<>5 AND a(x)<>6
GOTO epais
ENDIF
ENDP
PROC epXT:
IF a(x)=1.5 :pdx(x)=a(x) :a(x)=113.26
ELSEIF a(x)=2 :pdx(x)=a(x)
:a(x)=138.48
```

```
ELSEIF a(x)=2.5 :pdx(x)=a(x)
:a(x)=165.58
ELSEIF a(x)=3 :pdx(x)=a(x)
:a(x)=188.83
ELSEIF a(x)=4 :pdx(x)=a(x)
:a(x)=244.68
ELSEIF a(x)=5 :pdx(x)=a(x)
:a(x)=305.86
ELSEIF a(x)=6 :pdx(x)=a(x)
:A(x)=367.01
ENDIF
ENDP
PROC pXT:
tx=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+
a(9)+a(10)+a(11)+a(12)+a(13)+a(14)+a(15)+
a(16)+a(17)+a(18)+a(19)+a(20)+a(21)+a(22)+
a(23)+a(24)+a(25)+a(26)+a(27)+a(28)+a(29)+a(30)
pxt=tx :pxt1=pxt
AT 1,6 :PRINT «Prix de
l'XT»,»(«:t$;»);»»,FIX$(pxt1,2,-10)
AT 1,7 :PRINT
«T.V.A. _____:»,FIX$(pxt1*18.6/
100,2,-10)
gAT 0,66 :gLINEBY 280,0
AT 1,9 :PRINT
«T.T.C. _____:»,FIX$(pxt1*1.186,2,-
10)
ENDP
PROC pL:
REM * Polycarbonate *
CLS
GIPRINT «POLYCARBONATE»
dINIT «* POLYCARBONATE *»
dTEXT «»,»Format standard», $402
dTEXT «»,»Format découpé», $402
g%=DIALOG
IF g%=2 :t$=»T0» :fm%=0
ELSEIF g%=3 :t$=»TX» :fm%=20
ENDIF
dINIT «Sélectionnez une entrée»
dTEXT «»,»Incolore», $402
dTEXT «»,»Blanc diffusant», $402
dTEXT «»,»Fume», $402
dTEXT «»,»Granité incolore», $402
g%=DIALOG
IF g%=2 :cl%=0
ELSEIF g%=3 :cl%=5
ELSEIF g%=4 :cl%=10
ELSEIF g%=5 :cl%=12
ENDIF
dt:
AT 1,1 :PRINT «Nombre
d'épaisseur(s):»:»: :INPUT X
IF x=0
GOTO dt
ENDIF
DO
qtpc:
epcc:
PRINT «longueur :»:»: :INPUT lo
PRINT «Largeur :»:»: :INPUT la
PRINT «Quantité :»:»: :INPUT q
spc(x)=(lo*la)*q
mt4=spc(x)+1
pdc(x)=(spc(x)*1.2)*pdc(x)
pd4=pdc(x)+1
ec=a(x)*fm%/100 :a(x)=a(x)+ec
:ef=a(x)*cl%/100
a(x)=a(x)+ef :a(x)=spc(x)*a(x)
x=x-1
UNTIL x=0
gAT 133,10 :gBOX 107,29 :gAT 140,22
gPRINT «Surf.=»,FIX$(mt4,2,-
10),»m2"
gAT 145,34
gPRINT «Pds=»,FIX$(pd4,2,-10),»Kg»
ppc:
ENDP
PROC qtpc:
epais:
AT 1,2 :PRINT «Epaisseur:»:»: :TRAP
INPUT a(x)
ep=a(x)
IF a(x)<>1 AND a(x)<>1.5 AND
a(x)<>2 AND a(x)<>3 AND a(x)<>4
AND a(x)<>5 AND a(x)<>6 AND
a(x)<>8 AND a(x)<>10 AND a(x)<>12
```

```
GOTO epais
ENDIF
ENDP
PROC eppc:
IF a(x)=1 :pdc(x)=a(x) :a(x)=121.95
ELSEIF a(x)=1.5 :pdc(x)=a(x)
:a(x)=176.26
ELSEIF a(x)=2 :pdc(x)=a(x)
:a(x)=240.79
ELSEIF a(x)=3 :pdc(x)=a(x)
:a(x)=355.51
ELSEIF a(x)=4 :pdc(x)=a(x)
:a(x)=479.27
ELSEIF a(x)=5 :pdc(x)=a(x)
:a(x)=596.3
ELSEIF a(x)=6 :pdc(x)=a(x)
:a(x)=713.15
ELSEIF a(x)=8 :pdc(x)=a(x)
:a(x)=946.96
ELSEIF a(x)=10 :pdc(x)=a(x)
:a(x)=1181.11
ELSEIF a(x)=12 :pdc(x)=a(x)
:a(x)=1415
ENDIF
ENDP
PROC ppc:
tx=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+
a(9)+a(10)+a(11)+a(12)+a(13)+a(14)+a(15)+
a(16)+a(17)+a(18)+a(19)+a(20)+a(21)+a(22)+
a(23)+a(24)+a(25)+a(26)+a(27)+a(28)+a(29)+a(30)
ppc=tx :ppc1=ppc
AT 1,6 :PRINT «Prix du
PC»,»(«:t$;»);»»,FIX$(ppc1,2,-10)
AT 1,7 :PRINT
«T.V.A. _____:»,FIX$(ppc1*18.6/
100,2,-10)
gAT 0,66 :gLINEBY 280,0
AT 1,9 :PRINT
«T.T.C. _____:»,FIX$(ppc1*1.186,2,-
10)
ENDP
PROC pm:
REM * Miroir *
CLS
GIPRINT «MIROIR»
dINIT «* MIROIR XT *»
dTEXT «»,»Format standard», $402
dTEXT «»,»Format découpé», $402
g%=DIALOG
IF g%=2 :fm%=0
ELSEIF g%=3 :fm%=25
ENDIF
dt:
AT 1,1 :PRINT «Nombre
d'épaisseur(s):»:»: :INPUT x
IF x=0
GOTO dt
ENDIF
DO
qtmir:
epmir:
lo:
AT 1,3 :PRINT «Longueur :»:»: :TRAP
INPUT lo
IF lo>2.44
BEEP 3,300
GIPRINT «Longueur maxi=2.440»
:GOTO lo
ENDIF
la:
AT 1,4 :PRINT «Largeur :»:»: :TRAP
INPUT la
IF la>1.22
BEEP 3,300
GIPRINT «Largeur maxi=1.220»
:GOTO la
ENDIF
PRINT «Quantité :»:»: :INPUT q
smi(x)=(lo*la)*q
mt5=smi(x)+1
pmi(x)=(smi(x)*1.2)*pmi(x)
pd5=pmi(x)+1
ef=a(x)*fm%/100 :a(x)=a(x)+ef
:a(x)=smi(x)*a(x)
x=x-1
UNTIL x=0
gAT 133,10 :gBOX 107,29 :gAT 140,22
```

```
gPRINT «Surf.=»,FIX$(mt5,2,-
10),»m2"
gAT 145,34
gPRINT «Pds=»,FIX$(pd5,2,-10),»Kg»
pmir:
ENDP
PROC qtmir:
epais:
AT 1,2 :PRINT «Epaisseur:»:»: :TRAP
INPUT a(x)
ep=a(x)
IF a(x)<>2 AND a(x)<>3
GOTO epais
ENDIF
ENDP
PROC epmir:
IF a(x)=2 :pmi(x)=a(x) :a(x)=432.86
ELSEIF a(x)=3 :pmi(x)=a(x)
:a(x)=474.97
ENDIF
ENDP
PROC pmir:
tx=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+
a(9)+a(10)+a(11)+a(12)+a(13)+a(14)+a(15)+
a(16)+a(17)+a(18)+a(19)+a(20)+a(21)+a(22)+
a(23)+a(24)+a(25)+a(26)+a(27)+a(28)+a(29)+a(30)
pmir=tx :pmir1=pmir
AT 1,6 :PRINT «Prix du
miroir:»,FIX$(pmir1,2,-10)
AT 1,7 :PRINT
«T.V.A.:»,FIX$(pmir1*18.6/100,2,-10)
gAT 0,66 :gLINEBY 280,0
AT 1,9 :PRINT
«T.T.C.:»,FIX$(pmir1*1.186,2,-10)
ENDP
PROC pT:
REM * Travail *
CLS
GIPRINT «MAIN D'OEUVRE»
PRINT «Main d'oeuvre en mn:»:»:
:INPUT t :t=t*1 REM ** Prix H.T en
minutes **
AT 1,3 :PRINT «Travail
H.T.:»,FIX$(t,2,-10)
ENDP
PROC pU:
REM * Polissage *
CLS
GIPRINT «POLISSAGE»
dINIT «Choix du polissage»
dTEXT «»,»Mécanique», $402
dTEXT «»,»Manuel», $402
g%=DIALOG
IF g%=2 :qtcot:
GOTO mecan
ELSEIF g%=3 :qtcot:
GOTO manu
ENDIF
mecan:
PRINT «Prix au mètre:»:»: :INPUT pm
p4=p4*pm
PRINT «Prix du polissage
mécanique:»,FIX$(p4,2,-10) :RETURN
manu:
pm=ep*1 REM ** Prix H.T à la minute
**
p4=p4*pm
PRINT «Prix du polissage
manuel:»,FIX$(p4,2,-10)
ENDP
PROC qtcot:
REM * Nombre de côtés *
qt:
dINIT
dLONG d&,»Nombre de c_tés:»,1,4
IF DIALOG=0
GOTO qt
ELSEIF d&=1 :qcot1:
ELSEIF d&=2 :qcot2:
ELSEIF d&=3 :qcot3:
ELSEIF d&=4 :p4=lo+lo+la+la
ENDIF
ENDP
PROC qcot1:
REM * Questions 1 c_tés *
dINIT «Sélectionner un côté»
dTEXT «»,»Longueur», $402
```



CALCUL DE DEVIS sur PSION series 3... CALCUL DE DEVIS sur PSION series3...

```
dTEXT «»,»Largeur», $402
g%=DIALOG
IF g%=2 :p4=lo
ELSEIF g%=3 :p4=la
ENDIF
ENDP
PROC qcot2:
REM * 2 C_tés *
dINIT «Sélectionner les côtés»
dTEXT «»,»Longueur+largeur», $402
dTEXT «»,»Longueur+longueur», $402
dTEXT «»,»Largeur+largeur», $402
g%=DIALOG
IF g%=2 :p4=lo+la
ELSEIF g%=3 :p4=lo+lo
ELSEIF g%=4 :p4=la+la
ENDIF
ENDP
PROC qcot3:
REM * 3 Côtés *
dINIT «Sélectionner un côté»
dTEXT «»,»2 longueurs+1 Largeur», $402
dTEXT «»,»2 largeurs+1 longueur», $402
g%=DIALOG
IF g%=2 :p4=lo+lo+la
ELSEIF g%=3 :p4=la+la+lo
ENDIF
ENDP
PROC pf:
REM * Affichage Final *
LOCAL r%
CLS
GIPRINT «PRIX FINAL»
gAT 0,4 :gLINEBY 280,0
pf=pma+pch+pxt+ppc+pmir
AT 1,2
PRINT
«Matière _____:»,FIX$(pf,2,-10),»frs»
PRINT
«Capot _____:»,FIX$(pcapot,2,-10),»frs»
PRINT «1/2
sphère _____:»,FIX$(ps,2,-10),»frs»
PRINT
«Polissage _____:»,FIX$(p4,2,-10),»frs»
PRINT «Main
d'oeuvre _____:»,FIX$(t,2,-10),»frs»
pf=pf+pcapot+ps+t+p4
gAT 0,58 :gLINEBY 280,0
AT 1,8
PRINT «TOTAL
H.T.V.A. _____:»,FIX$(pf,2,-10),»frs»
gAT 0,75 :gLINEBY 280,0
r%=GET
dINIT «SAUVER DEVIS»
dBUTTONS «Non»,%N,»Oui»,%O
r%=DIALOG
IF r%=0
RETURN
ENDIF
IF r%=111
prix$=FIX$(pf,2,-10)
titre$=»Sauver devis»
OPEN
«M:clients»,A,nm$,dat$,piec$,prix$,moul$,not$
IF affd$:
A.nm$=nm$ :A.dat$=dat$
:A.piec$=piec$
:A.prix$=prix$ :A.moul$=moul$
:A.not$=not$
APPEND
CLOSE
ENDIF :ENDIF
ENDP
PROC pC:
REM * capot *
LOCAL h,sc,slp,slp1,surcapo,d,toc
GIPRINT «CAPOT»
dt:
CLS
```

```
PRINT «Nombre de capot ?»: :INPUT
x
IF x=0
GOTO dt
ENDIF
epais:
DO
qt:
ep:
PRINT «Longueur(dessus):»: :INPUT
lo :lo=lo+.01
PRINT «Largeur (dessus):»: :INPUT la
:la=la+.01
PRINT «Hauteur _____:»: :INPUT h
sc=lo*la :slp=(lo*h)*2 :slp1=(la*h)*2
surcapo=sc+slp+slp1
a(x)=surcapo*a(x) :a(x)=a(x)+20%
PRINT «Matière _____:»,FIX$(a(x),2,-10)
x=x-1
UNTIL x=0
d=((lo+la+h)*2.5)*1 REM ** Prix
horaire H.T **
PRINT «Travail _____:»,FIX$(d,2,-10)
toc=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+
+a(9)+a(10)+a(11)+a(12)+a(13)+a(14)+a(15)+
+a(16)+a(17)+a(18)+a(19)+a(20)+a(21)+a(22)+
+a(23)+a(24)+a(25)+a(26)+a(27)+a(28)+a(29)+a(30)
pcapot=toc+d
PRINT «_____»
PRINT «Total H.T.:»,FIX$(pcapot,2,-10)
ENDP
PROC pR:
REM * Remise *
CLS
GIPRINT «REMISE»
PRINT «Taux de remise:»: :INPUT tr
AT 1,1 :PRINT «Taux de remise
:»,tr,»%»
avr=pma+pch+pxt+ppc
PRINT «Prix avant
remise:»,FIX$(avr,2,-10)
pma1=pma*tr/100 :pma=pma-pma1
pch1=pch*tr/100 :pch=pch-pch1
pxt1=pxt*tr/100 :pxt=pxt-pxt1
ppc1=ppc*tr/100 :ppc=ppc-ppc1
apr=pma1+pch1+pxt1+ppc1
PRINT «Moins la remise
:»,FIX$(apr,2,-10)
PRINT «_____»
apr1=pma+pch+pxt+ppc
PRINT «Prix après
remise:»,FIX$(apr1,2,-10)
ENDP
PROC pW:
REM * Majoration *
CLS
GIPRINT «MAJORATION»
PRINT «Taux de majoration:»: :INPUT
tr
AT 1,1 :PRINT «Taux de majoration
:»,tr,»%»
avr=pma+pch+pxt+ppc
PRINT «Prix avant
majoration:»,FIX$(avr,2,-10)
pma1=pma*tr/100 :pma=pma+pma1
pch1=pch*tr/100 :pch=pch+pch1
pxt1=pxt*tr/100 :pxt=pxt+pxt1
ppc1=ppc*tr/100 :ppc=ppc+ppc1
apr=pma1+pch1+pxt1+ppc1
PRINT «Plus la majoration
:»,FIX$(apr,2,-10)
PRINT «_____»
apr1=pma+pch+pxt+ppc
PRINT «Prix avec majoration
:»,FIX$(apr1,2,-10)
ENDP
PROC pS:
REM * Demi-sphère *
CLS
GIPRINT «DEMI-SPHERE»
dINIT «Sélectionner une entrée»
dTEXT «»,»Chercher un diamè-
tre», $402
dTEXT «»,»Calculer un prix», $402
```

```
dTEXT «»,»Quitter», $402
g%=DIALOG
IF g%=2 :sphV:
ELSEIF g%=3 :GOTO qt
ELSEIF g%=4 :devis:
ENDIF
qt:
qs&=1
dINIT
dTEXT «»,»Nombre de demi-sphère
?»,2
dLONG qs&,»Quantité _____»,1,1000
IF DIALOG=0
GOTO qt
ELSEIF qs&>0 AND qs&<5 :qs%=0
ELSEIF qs&>=5 AND qs&<=10
:qs%=10
ELSEIF qs&>=11 AND qs&<=15
:qs%=20
ELSEIF qs&>=16 :qs%=30
ENDIF
dINIT «Sélectionner une entrée»
dTEXT «»,»Incolore», $402
dTEXT «»,»Blanc», $402
dTEXT «»,»Couleur», $402
dTEXT «»,»Altuglas choc», $402
g%=DIALOG
IF g%=2 :co%=0
ELSEIF g%=3 :co%=5
ELSEIF g%=4 :co%=10
ELSEIF g%=5 :co%=25
ENDIF
dINIT «Sélectionner une entrée»
dTEXT «»,»Epaisseur normale», $402
dTEXT «»,»Epaisseur modifiée», $402
g%=DIALOG
IF g%=2 :sphere:
ELSEIF g%=3 :pNS:
ENDIF
ENDP
PROC sphere:
dt:
AT 1,1 :PRINT «Diamètre en m/m:»,
:INPUT d
IF d<>50 AND d<>60 AND d<>70 AND
d<>80 AND d<>90 AND d<>100 AND
d<>105 AND d<>120 AND d<>125
AND d<>133
IF d<>140 AND d<>150 AND d<>160
AND d<>168 AND d<>170 AND
d<>180 AND d<>195 AND d<>200
AND d<>210 AND d<>215 AND
d<>220 AND d<>230 AND d<>250
AND d<>280 AND d<>285
IF d<>300 AND d<>308 AND d<>310
AND d<>320 AND d<>325 AND
d<>330 AND d<>342 AND d<>346
AND d<>350
IF d<>358 AND d<>360 AND d<>380
AND d<>400 AND d<>405 AND
d<>420 AND d<>425 AND d<>440
AND d<>450 AND d<>457 AND
d<>463 AND d<>470 AND d<>500
IF d<>523 AND d<>536 AND d<>546
AND d<>550 AND d<>556 AND
d<>565 AND d<>580 AND d<>600
AND d<>630 AND d<>640 AND
d<>650 AND d<>665 AND d<>670
AND d<>698 AND d<>700
IF d<>715 AND d<>720 AND d<>725
AND d<>740 AND d<>760 AND
d<>770 AND d<>800 AND d<>850
AND d<>865 AND d<>875 AND
d<>900 AND d<>915 AND d<>950
AND d<>980 AND d<>996
IF d<>1000 AND d<>1005 AND
d<>1015 AND d<>1040 AND d<>1085
AND d<>1100 AND d<>1150 AND
d<>1172 AND d<>1180 AND d<>1200
AND d<>1350 AND d<>1400 AND
d<>1500 AND d<>1630 AND d<>1800
AND d<>2000 AND d<>2150
GOTO dt
ENDIF :ENDIF
ENDIF :ENDIF
ENDIF :ENDIF
ENDIF
```

```
psph:
totsph:
ENDP
PROC psph:
LOCAL
a,b,c,e,f,g,h,i,j,k,l,m,n,o,q,r,s,t,u
a=117.38 :REM 90 à 100
b=120.65 :REM 120 à 125
c=124.26 :REM 133 à 150
e=147.24 :REM 160 à 200
f=165.87 :REM 210 à 250
g=196.17 :REM 280 à 300
h=226.23 :REM 308 à 350
i=233.42 :REM 358 à 400
j=256.44 :REM 405 à 450
k=297.31 :REM 463 à 500
l=343.01 :REM 523 à 550
m=432.61 :REM 556 à 600
n=534.88 :REM 630 à 698 (4m/m)
o=570.07 :REM 700 à 740
q=600.59 :REM 760 à 770
r=749.04 :REM 850 à 875
s=1279.04 :REM 915 à 1005
t=1443.3 :REM 1015 à 1100
u=1929.38 :REM 1150 à 1200
q=749.04 :REM 850 à 875
ep1=ep1
IF d=50 :eps%=3 :ps=114.4
ELSEIF d=60 :eps%=3 :ps=115.02
ELSEIF d=70 :eps%=3 :ps=115.71
ELSEIF d=80 :eps%=3 :ps=116.47
ELSEIF d>=90 AND d<=100 :eps%=3
:ps=a
ELSEIF d=105 :eps%=3 :ps=118.81
ELSEIF d=120 :eps%=3 :ps=b
ELSEIF d=125 :eps%=3 :ps=b
ELSEIF d>=133 AND d<=150 :eps%=3
:ps=c
ELSEIF d>=160 AND d<=200 :eps%=3
:ps=e
ELSEIF d>=210 AND d<=250 :eps%=3
:ps=f
ELSEIF d>=280 AND d<=300 :eps%=3
:ps=g
ELSEIF d>=308 AND d<=350 :eps%=3
:ps=h
ELSEIF d>=358 AND d<=400 :eps%=3
:ps=i
ELSEIF d>=405 AND d<=450 :eps%=3
:ps=j
ELSEIF d>=457 AND d<=500 :eps%=3
:ps=k
ELSEIF d>=523 AND d<=550 :eps%=3
:ps=l
ELSEIF d>=556 AND d<=600 :eps%=3
:ps=m
ELSEIF d>=630 AND d<=698 :eps%=4
:ps=n
ELSEIF d>=700 AND d<=740 :eps%=4
:ps=o
ELSEIF d>=760 AND d<=770 :eps%=4
:ps=q
ELSEIF d=800 :eps%=4 :ps=634.08
ELSEIF d>=850 AND d<=875 :eps%=4
:ps=r
ELSEIF d=900 :eps%=4 :ps=775.08
ELSEIF d>=915 AND d<=1005
:eps%=5 :ps=s
ELSEIF d>=1015 AND d<=1100
:eps%=5 :ps=t
ELSEIF d>=1150 AND d<=1200
:eps%=5 :ps=u
ELSEIF d=1350 :eps%=6 :ps=2337.7
ELSEIF d=1400 :eps%=6 :ps=2505.46
ELSEIF d=1500 :eps%=6 :ps=2970.89
ELSEIF d=1630 :eps%=6 :ps=3329.3
ELSEIF d=1800 :eps%=6 :ps=4313.02
ELSEIF d=2000 :eps%=6 :ps=4852.68
ENDIF
ENDP
PROC totsph:
psc=ps*co%/100 :ps=ps+psc
:pscq=ps*qs%/100 :ps=ps-pscq
CLS
PRINT «Nombre de sphère _____:»,qs&
:REM qsph%
PRINT «Diamètre _____:»,d
```



CALCUL DE DEVIS sur PSION series 3... CALCUL DE DEVIS sur PSION series3...

```

PRINT «Epaisseur _____:»,eps%
PRINT «Unitaire
H.T. _____:»,FIX$(ps,2,-10),»frs»
gAT 0,40 :gLINEBY 280,0
ps=ps*qs&
AT 1,6
PRINT «Total
H.T. _____:»,FIX$(ps,2,-10),»frs»
ENDP
PROC pNS:
REM * MODIF EP. SPH. *
CLS
qt:
AT 1,1 :PRINT «Diamètre à modifier:»,
:TRAP INPUT d
IF d<50
BEEP 3,300
GOTO qt
ENDIF
psph:
PRINT «Nouvelle epaisseur:», :INPUT
ep1
ps=(ps/eps%)*ep1 :eps%=ep1
totsph:
ENDP
REM * FICHER *
PROC pO:
LOCAL g%
GIPRINT «OUVERTURE DU
FICHER:CLIENTS»
DO
IF NOT EXIST(«clients»)
CREATE
«M:clients»,A,nm$,dat$,piec$,prix$,moul$,not$
ELSE
OPEN
«M:clients»,A,nm$,dat$,piec$,prix$,moul$,not$
ENDIF
dINIT «Sélectionnez une action»
dTEXT «>>»,»Ajouter une fiche», $402
dTEXT «>>»,»Editer une fiche», $402
dTEXT «>>»,»Nombre de fiches», $402
dTEXT «>>»,»Effacer une fiche», $402
dTEXT «>>»,»Retour», $402
g%=DIALOG
IF g%=2 :ajout:
ELSEIF g%=3 :edit:
ELSEIF g%=4
CLS
gAT 50,25 :gBORDER 3,145,30
gAT 70,42 :gPRINT «Nombre de fiches
:»,COUNT
GET
CLOSE
ELSEIF g%=5 :eff:
ELSEIF g%=6 :CLOSE :devis:
ENDIF
UNTIL g%=0
ENDP
PROC ajout:
nm$=»» :piec$=»»
prix$=»» :moul$=»» :not$=»»
titre$=»Entrez une fiche»
IF affd%:
A.nm$=nm$ :A.dat$=dat$

```

```

:A.piec$=piec$
A.prix$=prix$ :A.moul$=moul$
:A.not$=not$
APPEND
CLOSE
ENDIF
ENDP
PROC edit:
LOCAL rech$(30),p%
CLS
dINIT «Trouver et éditer une fiche»
dEDIT rech$,»Chercher»,15
IF DIALOG
IF FIND(«*»+rech$+»*)=0
ALERT(«Pas de fiche trouvée»)
CLOSE
RETURN :ENDIF
DO
nm$=A.nm$ :dat$=A.dat$
:piec$=A.piec$
prix$=A.prix$ :moul$=A.moul$
:not$=A.not$
titre$=»Editer fiche»
IF affd%:
UPDATE :BREAK
ELSE :NEXT :ENDIF
FIND(«*»+rech$+»*)
IF EOF
ALERT(«Pas d'autres fiches»)
BREAK :ENDIF
UNTIL 0 :ENDIF
CLOSE
ENDP
PROC eff:
LOCAL rech$(30),p%,d%
CLS
dINIT «Trouver et effacer une fiche»
dEDIT rech$,»Chercher»,15
IF DIALOG
IF FIND(«*»+rech$+»*)=0
ALERT(«Pas de fiche trouvée»)
CLOSE
RETURN :ENDIF
DO
nm$=A.nm$ :dat$=A.dat$
:piec$=A.piec$
prix$=A.prix$ :moul$=A.moul$
:not$=A.not$
titre$=»Effacer fiche»
IF affd%:
dINIT «Effacer fiche ?»
dBUTTONS «Non»,%N,»Oui»,%O
d%=DIALOG
IF d%=0 :RETURN :ENDIF
IF d%=111 :ERASE :BREAK
ELSE :NEXT :ENDIF
FIND(«*»+rech$+»*)
IF EOF
ALERT(«Pas d'autres fiches»)
BREAK :ENDIF :ENDIF
UNTIL 0 :ENDIF
CLOSE
ENDP
PROC affd%:
dINIT titre$
dEDIT nm$,» Nom:»,25
dEDIT dat$,» Date:»,25
dEDIT piec$,»Pièce:»,25
dEDIT prix$,»Prix :»,25

```

```

dEDIT moul$,»Moule:»,25
dEDIT not$,» Notes:»,25
RETURN DIALOG
ENDP
PROC pA:
LOCAL ch%,d%
GIPRINT «COLLES & ENTRETIEN»
ch%=1
dINIT «Colles & Produits d'entretien»
dCHOICE ch%,»Altufix»,»S
500gr:47.03,S 1kg:82.82,P10
500gr:83.46,P12 1kg:155.78"
dCHOICE
ch%,»Altugliss»,»1kg:190.17"
dCHOICE
ch%,»Alunet»,»400gr:25.20,4litres:191.98"
dCHOICE ch%,»Altupol»,»N°1
1kg:44.90,N°2 1kg:37.53"
dCHOICE ch%,»Transfert
3M»,»Largeur 12m/m:285.05.Largeur
19m/m:451.37"
dTEXT «>>»,»<Tab> voir liste - <Esc>
pour quitter»
d%=DIALOG
devis:
ENDP
PROC pE:
LOCAL r%
GIPRINT «EFFACER MEMOIRES»
gFONT 1 :gSTYLE 0
dINIT «Remettre à 0 les variables»
dBUTTONS «Non»,%N,»Oui»,%O
r%=DIALOG
IF r%=0 :RETURN :ENDIF
IF r%=111
GIPRINT «Les variables sont remises
à 0»,1
pause -15
devis:
ENDIF
ENDP
PROC pK:
CLS
GIPRINT «SURFACES & POIDS»
mt=mt1+mt2+mt3+mt4+mt5
:pd=pd1+pd2+pd3+pd4+pd5
gAT 45,20 :gBORDER 3,150,40
gAT 65,38 :gPRINT
«Surface:»,FIX$(mt,2,-10),»m2"
gAT 78,48 :gPRINT
«Poids:»,FIX$(pd,2,-10),»Kg»
ENDP
PROC pQ:
REM * FIN *
LOCAL r%
dINIT «Quitter ce programme ?»
dBUTTONS «Non»,%N,»Oui»,%O
r%=DIALOG
IF r%=0 :RETURN :ENDIF
IF r%=111 :STOP : ENDIF
ENDP
PROC sphV:
LOCAL ch%,d%
ch%=1
dINIT
dCHOICEch%,»Diamètre50à195»,»50mm=
114.4,60mm=115.02,70mm=115.71,80mm=
116.47,90mm=117.38,100mm=117.38,105mm=

```

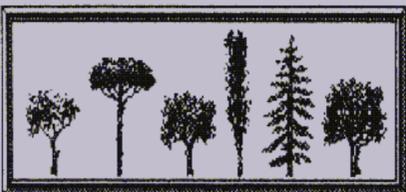
```

118.81,120mm=120.65,125mm=120.65,133mm=
124.26,140mm=124.26,150mm=124.26,160mm=
147.24,168mm=147.24,170mm=147.24,180mm=
147.24,195mm=147.24"
dCHOICEch%,»Diamètre200à346»,»200mm=
147.24,210mm=165.87,215mm=165.87,220mm=
165.87,230mm=165.87,250mm=165.87,280mm=
196.17,285mm=196.17,300mm=196.17,308mm=
226.23,310mm=226.23,320mm=226.23,325mm=
226.23,330mm=226.23,342mm=226.23,346mm=
226.23"
dCHOICEch%,»Diamètre350à536»,»350mm
=226.23,358mm=233.42,360mm=233.42,380mm
=233.42,400mm=233.42,405mm=256.44,420mm
=256.44,425mm=256.44,440mm=256.44,450mm
=256.44,457mm=297.31,463mm=297.31,470mm
=297.31,500mm=297.31,523mm=343.01,536mm
=343.01"
dCHOICEch%,»Diamètre546à725»,»546mm
=343.01,546mm=343.01,565mm=432.61,565mm
=432.61,580mm=432.61,600mm=432.61,630mm
=534.88,640mm=534.88,650mm=534.88,665mm
=534.88,670mm=534.88,698mm=534.88,700mm
=570.07,715mm=570.07,720mm=570.07,725mm
=570.07"
dCHOICEch%,»Diamètre740à1015»,»740mm
=570.07,760mm=570.07,770mm=570.07,800mm
=634.08,850mm=749.04,865mm=749.04,875mm
=749.04,900mm=775.08,915mm=1279.04,950mm
=1279.04,980mm=1279.04,996mm=1279.04,1000mm
=1279.04,1005mm=1279.04,1015mm=1443.3"
dCHOICEch%,»Diamètre1040à2000»,»
1040mm=1443.3,1085mm=1443.3,1100mm
=1443.3,1150mm=1929.33,1172mm=
1929.33,1180mm=1929.33,1200mm=
1929.33,1350mm=2337.7,1400mm=
2505.46,1500mm=2970.89,1630mm=
3329.3,1800mm=4313.02,2000mm=4852.68"
dTEXT «>>»,»<Tab> voir liste
<Esc> pour quitter»
d%=DIALOG
devis:
ENDP
PROC pl:
LOCAL r%
GIPRINT «CHARGER DESSIN»
dINIT «Effacer écran ?»
dBUTTONS «Non»,%N,»Oui»,%O
r%=DIALOG
IF r%=0 :GOTO Charge:: :ENDIF
IF R%=111
CLS
Charge::
dINIT
dTEXT «>>»,»Dessin à charger», $102
dFILE nom$,»Charger:»,16
DIALOG
IF nom$<>»
temp%=gLOADBIT(nom$)
gUSE 1
gAT 0,0
gCOPY temp%,0,0,240,90,2
gCLOSE temp%
GIPRINT
«Dessin»+CHR$(34)+RIGHT$(nom$,12)
+CHR$(34)
ENDIF :ENDIF
ENDP

```

Logiciel «Herbier» pour PSION series 3.

Mettre des arbres ou un herbier dans sa poche, c'est possible aujourd'hui, grâce nos petits compagnons de tous les jours: les ordinateurs de poche. Le PSION series 3 ce prête parfaitement à cette application. Ses possibilités graphiques plus que correctes (encore meilleures sur le nouveau PSION series 3a...), sa gestion des menus, sa mémoire, permettent d'avoir accès toute une bibliothèque de dessins, de définitions, de nomenclatures dans un volume vraiment très réduit (une carte FLASH ou RAM), disponible à tout moment. Ce logiciel permettra aux enfants (et aux adultes...) d'avoir une visualisation d'arbres de feuilles et de fruits, d'imprimer des fiches se rapportant aux feuilles trouvées au cours d'une sortie en famille ou pourquoi pas l'école avec le maître ou la maîtresse, pour se constituer un herbier. Une première version distribuée en diffusion libre, par POICHE COMPUTERS et le CLUB series 3, donnera un aperçu du logiciel qui sera beaucoup plus complet par la suite, avec plus de 100 dessins et textes. D'autres logiciels de ce type peuvent être mis au point. Pour plus de renseignements contacter Mr Daniel Tchekinian au 42 80 38 63 Paris. Programmeur en C ou en Pascal bien venu pour mettre au point des programmes.



cultivé
t(prune),
artout,
p humides
ternées

Nom: TCHEKINIAN
Prenom: Daniel
Tel: 42 80 38 63

Logiciel: © HERBIER
Version: 1.01 Sept. 93

Abricotier, m. : arbre fruitier à fleurs blanches ou roses paraissant avant les feuilles, cultivé pour ses fruits. Feuilles caduques, cordées à la base. Famille des rosacées.



BIORYTHMES POUR SERIES 3

par Gilles JEAN

```
APP Bio
  type 3
  path»\bio»
  ext «bio»
  icon «biorythm.pic»
ENDA
PROC start:
  global file$(128),t$(1)
  global nom$(12),dn&,dj&,d&,p%,e%,i%
  giprint»Gilles JEAN - Shareware diffusable v1.2F»
  dj&=days(day,month,year)
  init:
  gfont 1
  gstyle 0
  do
  lock on
  dinit
  dposition 1,-1
  ddate
  dj&,»Aujourd'hui»,days(1,1,1900),days(31,12,2049)
  ddate dn&,»Date de
  naissance»,days(1,1,1900),days(31,12,2049)
  dedit nom$,»Nom»,12
  if dialog
    if dn&>dj&
      giprint»Date de naissance > date du jour»
    else
      d&=(dj&-dn&)
      affi:
    endif
  else
    if t$=>C»
      a.nom$=nom$
      a.dn&=dn&
      append
    elseif t$=>O»
      if a.nom$<>nom$ or a.dn&<>dn&
        a.nom$=nom$
        a.dn&=dn&
        update
      endif
    endif
  break
endif
until 0
lock off
ENDP
PROC affi:
  gat 100,47 :gpatt -1,100,8,3
  gat 100,57 :gpatt -1,100,8,3
  gat 100,67 :gpatt -1,100,8,3
  p%=int(sin(2*pi*(d&-(d&/23*23))/23)*100)
  e%=int(sin(2*pi*(d&-(d&/28*28))/28)*100)
  i%=int(sin(2*pi*(d&-(d&/33*33))/33)*100)
  gat 100,47 :gfill int((100+p%)/2),8,0
  gat 100,57 :gfill int((100+e%)/2),8,0
  gat 100,67 :gfill int((100+i%)/2),8,0
  gat 205,55 :gprintb num$(p%,4)+»%»,30,1
  gat 205,65 :gprintb num$(e%,4)+»%»,30,1
  gat 205,75 :gprintb num$(i%,4)+»%»,30,1
ENDP
PROC INIT:
  file$=cmd$(2)
  t$=cmd$(3)
```



```
gat 0,0
gborder $204,45,45
gat 0,45
gborder $204,240,35
gat 5,20
gstyle 8
gprintb»Bio»,35,3
gstyle 0
gat 5,35
gprintb»rythme»,35,3
gat 85,44 :gprint»-100"
gat 185,44 :gprint»+100"
gat 5,55 :gprintb
«Physique»,60
gat 5,65 :gprintb
«Emotif»,60
gat 5,75 :gprintb

«C,r,bral»,60
if t$=>C»
  trap mkdir
  left$(file$,12)
  create
  file$,a,nom$,dn&
  nom$=>????????»
  elseif t$=>O»
    open
    file$,a,nom$,dn&
    nom$=a.nom$
    dn&=a.dn&
    d&=(dj&-dn&)
    affi:
  endif
  setname file$
ENDP
```

CLUB PSION

Depuis le mois de mai, les Psionistes peuvent se regrouper sous la bannière du club.

Le but de ce club, indépendant aussi bien d'Aware que de Psion, est de regrouper tous les utilisateurs de cet engin, passionnés de programmation ou simples utilisateurs, et de diffuser la connaissance et les programmes existants pour le Psion.

Les activités du club sont ou seront multiples :

- * Edition d'un bulletin périodique
- * Réunions des membres pour échange d'infos, de programmes...
- * Contacts avec les clubs utilisateurs en Europe. (GB en particulier)

Pour adhérer ou pour tout renseignement :

Club series 3 : C/O Olivier MEDAM -
11 bd Bonne Nouvelle 75002 PARIS



Le train et Le series 3 (conte de Noël)

ou ne perdez plus de temps à consulter vos horaires de train grâce à votre series 3a...

par Vincent LEGROS selon une idée de Jean Rémi LEGROS

Vous connaissez tous le fameux système américain SOCRATE, acheté à prix d'or et employé par la SNCF, qui soit disant est sensé nous apporter du bien être dans nos déplacements par le rail. On voit le résultat ! Eh bien voici la preuve qu'un programmeur français fait mieux et moins cher... Question : pourquoi nos très éclairés dirigeants n'ont-ils pas pris la peine de consulter les frères LEGROS ! on vous le disait, la FRANCE est le centre du monde pour la programmation!!! ouh là que le chauvinisme ne m'étouffe pas !

N'avez vous jamais cherché autour de vous les fiches horaires SNCF pour savoir quel train prendre pour arriver à la réunion de M. Heintel à Paris, sachant que cette réunion commence à 10h30 précises ? A quelle heure prendre votre train en gare de la Part Dieu pour arriver à Paris sans être en retard ???

La seule réponse à ce type de d'interrogation est le système très performant de base de données du PSION series 3a...

En effet, le S3a propose une recherche sur un libellé, donc en utilisant judicieusement une base de données, vous pourrez avoir très rapidement la réponse à votre problème.

Tout d'abord, chargez la gestion de base de données dans le menu fichier, sélectionnez Nouveau.

Donnez un nom à votre fichier, par exemple LYO-PAR pour les horaires Lyon-Paris, sachant qu'il faudra faire une autre base PAR-LYO pour le retour.

Editez les libellés (menu Editer) afin de paramétrer les champs de votre base de la manière suivante :

NUMERO DE TRAIN : SATOLAS TGV : PERRACHE : PART DIEU : VILLEFRANCHE :
MACON VILLE : TOURNUS : CHALON SUR MARNE : CHAGNY : BEAUNE : DIJON
MACON LOCHE TGV : LE CREUSOT : MASSY : PARIS : CIRCULATION :

Saisissez les horaires de la manière suivante : LE CREUSOT 09:44, pour un train passant en gare du Creusot à 9h44 (21:44 s'il s'agit du soir). Une fois votre base de données terminée, l'Astuce arrive.

Pour connaître les trains partant de Lyon Part Dieu pour être en gare de Lyon (Paris) vers 10h00, effectuez une recherche sur le libellé (touche PSION+ R), puis Tab. Déplacez-vous pour sélectionner LYON PART DIEU et validez.

Il vous suffit de taper 08 : et de valider, le Psion cherchera les trains vers 8h00.

Il ne vous reste plus qu'à vérifier si ce train circule bien à la date voulue (Circulation), et à prier que vous n'ayez pas fait d'erreurs de saisie. La base de données PARIS --> LYON et LYON --> PARIS est disponible au club series 3. (sinon, il vous faudra la créer vous-même). Si cela était, merci de nous l'expédier pour en faire profiter tout le monde intéressé.

BEEPON

par Vincent LEGROS

Ce petit utilitaire permet de charger une ou deux images lors de l'allumage du Psion series 3a. Les instructions avant le Call (\$6c8d) seront exécutées lorsque l'on presse (ON).

Taper le programme en OPL - Transcoder S3a - inutile exécuter - Exécuter BEEPON - pour arrêter, se placer sur le nom du fichier et presser Eff -

```

PROC beepon:
Local a%(6),im%,fen%,im2%,i
fen%=gIdentity
im%=gLoadBit("\pic\IMAGE1")
im2%=gLoadBit("\pic\IMAGE2")
guse fen%
gAt 0,0
gcopy im%,0,0,125,150,0
gFont 12
gStyle 9
gAt 2023,0
gBorder 1,277,160
gAt 300,50
gPrint " PSION series 3"
gAt 300,100
gcopy im2%,0,0,129,37,0
pause 90
call ($6c8d):gUpdate
While 1
do
getevent a% 0
ifa%(1)=$404:stop:endif:REMEteint
until a%(1)=$403
call($198d,0,0):gupdate
beep 5,300:pause 10:beep 5,500
Pause 50
call($198d,100,0):gupdate
endwh
ENDP

```

**Avez-vous besoin de mémoire pour votre PSION series 3/3a ?
comparez les prix !!!**

SSD RAM 128K : 790Fttc SSD RAM 512K : 1690Fttc SSD RAM 1MB : 2990Fttc

SSD FLASH 128K : 520Fttc SSD FLASH 512K : 1290Fttc

SSD FLASH 1MB : 2290Fttc SSD FLASH 2MB : 2990Fttc

MICROCARDS - 42220 BURDIGNES - TEL 77.39.68.13 FAX 77.39.19.60

ANIMEZ VOS MACHINES GRAPHIQUES CASIO !!!

par David WINTER

Dans cette première partie, nous allons voir comment afficher la plupart des caractères de la CASIO graphique n'importe où sur l'écran.

1) L'accès à la mémoire vive (RAM):

La zone texte de l'écran est placée à un certain endroit dans la RAM de la machine. Il existe un moyen d'y accéder: les variables à indices, par exemple Z(400). Or, la machine interdit l'accès à ces mémoires au delà d'une valeur précise (qui varie selon la machine et le nombre de mémoires étendues configurées). Il faut donc faire une petite «magouille», simple, mais qui efface tous les programmes. La voici : c'est une succession de touches à presser.

- | | |
|-------------|-----------------------|
| - ? | - <== |
| → | - DEL |
| - EXE | - éteindre la machine |
| - MODE | - MODE |
| - EXP | - 3 |
| - AC | - SHIFT |
| - RANGE x 2 | - DEL |
| - II 7 | |

Ces 4 dernières opérations sont à répéter jusqu'à obtenir 6774 Bytes Free. En cas d'erreur, éteindre puis rallumer la machine. Vérifions l'accès libre aux mémoires : faisons MODE . EXE. On doit avoir:

- Program : 0 - aucun programme en mode 2
 Memory : 0 - toutes les mémoires sont accessibles
 x Bytes Free - nombre de pas libres sans programmes + 208.

(par exemple, sous 8500, on a 6566 + 208 =6774 Bytes free).

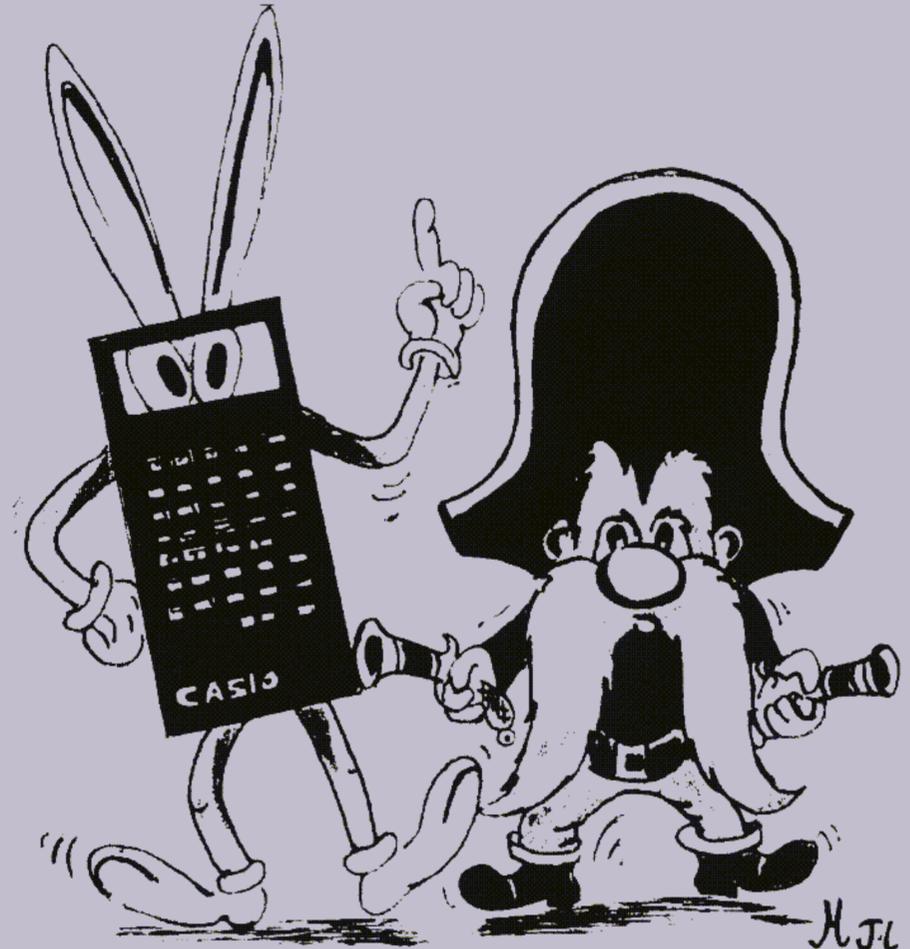
2) La zone de l'écran texte:

Cette zone est délimitée par deux adresses qui varient selon les machines. Les voici :

MACHINE	Début	Fin
7000	Z(183)	Z(198)
7500	Z(631)	Z(646)
8000	Z(311)	Z(326)
8500	Z(951)	Z(966)

Nous nous apercevons que les 8 lignes de 16 caractères de l'écran texte sont codées par 16 mémoires. Autrement dit, chaque mémoire représente 8 caractères. Voici le plan de cette zone texte. La variable n représente l'adresse du début de la zone.

n + 16	n + 14	n + 5	n + 4
n + 13	n + 12	n + 3	n + 2
n + 11	n + 10	n + 1	n
n + 9	n + 8		
n + 7	n + 6		



3) Codage d'une zone de huit caractères.

Comme nous l'avons vu, l'affichage se fait par groupes de 8 caractères. Il n'est donc pas possible d'en afficher moins de 8: il faudra mettre des espaces dans ce cas.

Exemple : pour afficher "ABC" tout à droite du groupe, il faudra en fait afficher "uuuuABC", où les "u" sont les espaces.

Le codage d'une mémoire n'est pas aussi simple. En effet, il y a une petite partie délicate. Tout d'abord, parlons des caractères affichables.

Dans la mémoire, un caractère est codé sur 8 bits, donc son code est représenté par deux chiffres. Or ce code est en mode hexadécimal, c'est-à-dire que chacun des deux chiffres du code d'un caractère varie de 0 à F en hexadécimal, donc de 0 à 15 en décimal. Voici une correspondance :

décimal	hexa
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Chaque code peut donc varier de 00 à FF: il y a 256 codes possibles, donc 256 caractères existants. Or, seuls les chiffres compris entre 0 et 9 sont accessibles avec le clavier de la machine: les caractères dont le code comprend une ou deux lettres (A à F), ne pourront pas être affichés.



ANIMEZ VOS MACHINES GRAPHIQUES CASIO...ANIMEZ VOS MACHINES

Voici à présent le tableau des caractères affichables:

	0	1	2	3	4	5	6	7	8	9
0	□	■	°	×	÷	√	←	→	↑	↓
1	x	y	\bar{x}	\bar{y}	\bar{x}	\bar{y}	Δ	Π	Σ	Ω
2	□	!	"	#	\$	%	&	'	()
3	0	1	2	3	4	5	6	7	8	9
4	@	A	B	C	D	E	F	G	H	I
5	P	Q	R	S	T	U	V	W	X	Y
6	.	a	b	c	d	e	f	g	h	i
7	p	q	r	s	t	u	v	w	x	y
8	A	B	D	E	g	□	O	ã	r	T
9	α	β	γ	δ	ε	τ	θ	λ	μ	ν

Passons à présent au codage d'une mémoire.

Le codage est sous le format d'un nombre décimal multiplié par un exposant. Le mantisse et le signé du nombre changent suivant le code.

Exemple: + 4,424344454647 E + 48.

Première étape : décomposer les 8 caractères et inscrire en dessous de chacun leur code. Nous allons afficher «ABCDEFGH».

On a alors: A B C D E F G H
41 42 43 44 45 46 47 48

Seconde étape: placer la virgule, les deux signes et l'exposant.

A B C D E F G H
41 42 43 44 45 46 47 48

L'exposant se place entre les codes des deux derniers caractères.

On a alors: A B C D E F G H
41 42 43 44 45 46 47 E 48

Placer la virgule entre les codes des deux premiers caractères.

On obtient: A B C D E F G H
41 . 42 43 44 45 46 47 E 48

Remplacer le second chiffre du premier code par le signe + ou - avec la touche +/- et le placer à gauche du premier chiffre du code. Il y a 4 combinaisons de signes suivant les valeurs du chiffre concerné:

chiffre	mantisse	exposant
0	+	-
1	+	-
5	-	-
6	-	+

Dans notre cas, le chiffre est égal à 1 : nous placerons le signe + à gauche du 4 et on retire le 1 du code 41, puis nous plaçons le signe + à gauche de l'exposant.

Nous obtenons alors: + 4 . 42 43 44 45 46 47 E + 48

ATTENTION : si le signe de l'exposant est -, il faut retrancher 100 à l'exposant. Exemple : si l'exposant est E-20, il faut retrancher 100 à 20, donc on obtient: E-80 ...

Le format du groupe exposant est:

EXP. SIGNE VALEUR EXPOSANT
E - ou + 20 par ex.

Or la machine se fixe à 10 chiffres significatifs. Il faut donc faire des opérations sur le code. Dans notre cas, nous avons:

$$4,424344454647E48 = 4,42434445E48 + 4647E36$$

4) Visualisation d'un code de 8 caractères:

Il faut stocker le code final dans une mémoire de la zone texte, puis, à votre choix:

- appuyer deux fois sur G/T

- faire " " (chaîne vide). Ce cas est la seule façon de voir les codes dans un programme.

(Si vous voulez vous servir de la touche £, il faut vérifier que le Range est correct).

Par exemple, tapez:

* 4. 42434445E48 + 4647E36 -> Z (951)

* appuyer deux fois sur G/T

* que voyez-vous en bas à droite ? **HURRAAAH !!!!**

Un bon conseil: ne vous affolez pas, cette méthode nécessite de la patience et de l'habitude.

5) Petit programme pour illustrer.

(87 ÷ 99 + 2 - E - 12) x E 87 -> B

(87 ÷ 99 + Z - E - 12) x E 20 -> C

C - 67 E 8 -> D

D - 67 E 10 -> E

E - 67 E 12 -> F

F - 67 E 14 -> G

G - 67 E 16 -> H

H - 67 E 18 -> I

G -> A

Lbl 0

B -> Z [n+1 + 2A]

A [Int 7 Ran # + 1] -> Z [n+2A]

A-1 -> A

A ≥ 0 ⇒ Goto 0

« « (chaîne vide)

G -> A

goto 0

Dans ce programme, la lettre n désigne l'adresse de départ de la zone texte. Sur la 8500G, elle vaut 951.

6) A propos de la méthode étudiée dans ce chapitre:

Tout ceci a été découvert en branchant le bus de données de la RAM de la 8500, via un convertisseur PARALLELE->SERIE, sur le port série de mon PC que j'avais à l'époque (un vieux TO-16 à 8088 ...)

Grâce à un programme que j'avais conçu, je pouvais voir à l'écran du PC ce qu'il se passait dans la RAM de ma CASIO 8500.

Dans l'article du prochain numéro de POICHE COMPUTERS, nous verrons comment jouer avec la partie graphique, et comment faire de petites animations.

connaître: les Accumulateurs

par M. Viescou

Les technologies évoluant vers la miniaturisation, permettent aux utilisateurs un degré de liberté supplémentaire dès lors que leurs appareils ne sont plus attachés par un fil au réseau électrique, mais alimentés par des piles ou des accumulateurs.

Dans le cas d'applications utilisant peu d'énergie et ayant une grande autonomie (plusieurs mois ou plusieurs années), les piles constituent une source d'énergie idéale.

En revanche, dans le cas d'applications de faible autonomie, (quelques heures ou quelques jours), le coût d'utilisation avec des piles devient prohibitif, et les appareils sont alimentés par des accumulateurs.

Ainsi chacun d'entre vous est confronté à leur utilisation, et souhaiterait une autonomie la plus grande possible afin de limiter le nombre de recharges qui constituent une contrainte. Il est difficile de dire, du point de vue de l'application pratique s'il vaut mieux recommander des piles ou des accumulateurs, car cela dépend toujours du type d'appareil utilisé. Spontanément, une R6 alcaline de bonne qualité a une capacité de plus de 2 AH alors qu'un accumulateur R6 Nickel Cadmium de base a 500 ou 600 mAH soit environ 4 fois moins.

Il faut utiliser les accumulateurs à l'hydrure de nickel de 1200 mAH pour arriver à une autonomie voisine ou supérieure à celle des piles.

En effet, utilisés sur des régimes de décharge élevés et continus, les accumulateurs de 1200 mAH donneront 1000 mAH alors que les piles de 2000 mAH ne donneront que 800 à 900 mAH. De plus, si la tension d'arrêt de l'appareil, c'est-à-dire la tension au dessous de laquelle l'électronique ne fonctionne plus, est de 1,1 volt par élément, la pile alcaline ne fera plus fonctionner l'appareil alors qu'il lui restera encore 40 à 50 % de sa capacité, cependant que l'accumulateur délivre 100% de sa capacité à 1,2 volt par élément.

Il est donc très difficile, dès lors que la capacité d'une pile est donnée pour une tension minimale de 0,9 ou 1 volt par élément, de définir sa durée dans une application où la tension d'arrêt serait de 1,1 volt par élément: sa capacité mesurée à la tension mini de 1,1 volt peut selon le débit n'être que de 50% de sa capacité à 0,9 volt, si son pallier se situe de 1,3 à 0,9 volt.

Enfin, l'avantage de la courbe de décharge à tension à peu près constante de l'accumulateur, constituant un avantage pour le rendement de ce dernier, peut constituer un inconvénient en fin de décharge, car la tension va chuter rapidement, cependant qu'avec une pile la tension varie très régulièrement. Lorsque les appareils sont dotés d'un système d'alerte de tension trop faible, l'utilisateur dispose d'une réserve de temps supérieure avec une pile qu'avec un accumulateur.

Sur les appareils sophistiqués, les accumulateurs sont livrés à l'origine avec leur chargeur, c'est le cas des caméscopes, des ordinateurs et imprimantes portables, des téléphones sans



fil et des téléphones cellulaires.

Dans ce cas, chacun se conforme aux indications du manuel d'utilisation.

En revanche, lorsque l'utilisateur doit charger son accumulateur, il doit veiller à contrôler les caractéristiques techniques afin éventuellement de modifier la durée de charge, si la capacité de son nouvel accumulateur est différente de celle de son ancien.

Si le chargeur est très sophistiqué, il chargera l'accumulateur jusqu'à ce que la température ou la tension atteigne une certaine valeur, puis s'arrêtera pour charger de nouveau lorsque la température ou la tension seront redescendues au-dessous d'un certain seuil caractéristique du type d'accumulateur utilisé, ainsi on pourra laisser l'accumulateur en charge assez longtemps pour que la capacité maximum soit atteinte (et donc l'autonomie maximum) sans risquer de détériorer ce dernier par une surcharge abusive, le chargeur régulant de lui-même le courant fourni.

Dans ces conditions, votre accumulateur pourra fonctionner de 500 à 1000 fois avec une détérioration de l'autonomie inférieure à 50%.

Lors de la surcharge d'un accumulateur, l'électrolyte se décompose en oxygène et hydrogène, l'oxygène est absorbé sur une électrode mais pas l'hydrogène. La pression à l'intérieur de l'accumulateur étanche augmente et pourrait le faire exploser s'il n'y avait pas une soupape de sécurité. En fait cette soupape s'ouvrira, et selon la position de l'accumula-



connaître : les Accumulateurs... connaître : les Accumulateurs... connaître : les Accu

teur, laissera sortir les gaz mais probablement aussi de l'électrolyte. Même si l'électrolyte n'endommage pas l'appareil, l'impédance de l'accumulateur augmentera et sa puissance diminuera.

Ainsi, dans une série cet accumulateur sera-t-il plus faible que les autres (moins de capacité après recharge). De surcroît, une électrode sera passivée et ce phénomène sera accentué. Il pourra arriver à 50% de sa capacité après seulement 10 ou 20 cycles au lieu de 500 ou 1000.

Dans la pratique, selon la tension à laquelle votre appareil s'arrête de fonctionner, la perte d'autonomie peut-être beaucoup plus rapide à cause de l'effet mémoire pour les accumulateurs au Nickel Cadmium.

Dans ce cas, il est recommandé de charger votre accumulateur, de le décharger rapidement et complètement, de le recharger complètement et éventuellement recommencer cette procédure qui aura pour effet de diminuer cet effet mémoire lié à l'électrode de cadmium. Malheureusement, cet effet se reproduira après quelques cycles d'utilisation supplémentaire et il pourra être nécessaire de renouveler ce «nettoyage» régulièrement.

En revanche, avec les accumulateurs au plomb ou les accumulateurs à l'hydrure de nickel, cet effet n'existe pas.

Si vous utilisez un chargeur du commerce, avec des accumulateurs de type R6 par exemple, et si l'appareil n'est pas très sophistiqué, vous serez obligé de contrôler la durée de la charge, et pour connaître cette durée, il faudra connaître le débit de votre chargeur.

En général les chargeurs de R6 débitent 1/10ème de la capacité d'un accumulateur R6 de base de 500 mA, soit 50 mA, et la notice du chargeur vous demande de le charger pendant 13 à 14 heures.

Avec un tel chargeur, si vous achetez des accumulateurs R6 nickel cadmium de 850 mAH, vous devrez les charger $850/50 = 17 \times 1,3$ ou $1,4 = 22$ à 24 heures pour obtenir une charge complète sinon en 14 heures vous ne chargeriez que 60% de la capacité et risqueriez de conclure que les 850 mAH ne donnent pas plus d'autonomie que les 500 mAH.

Raison de plus si vous utilisez des accumulateurs R6 à l'hydrure de nickel de 1200 mAH, vous risquez de ne les charger qu'à 40% si vous respectez les temps de charge du constructeur de chargeur de 14 heures au lieu du temps calculé de 34 heures.

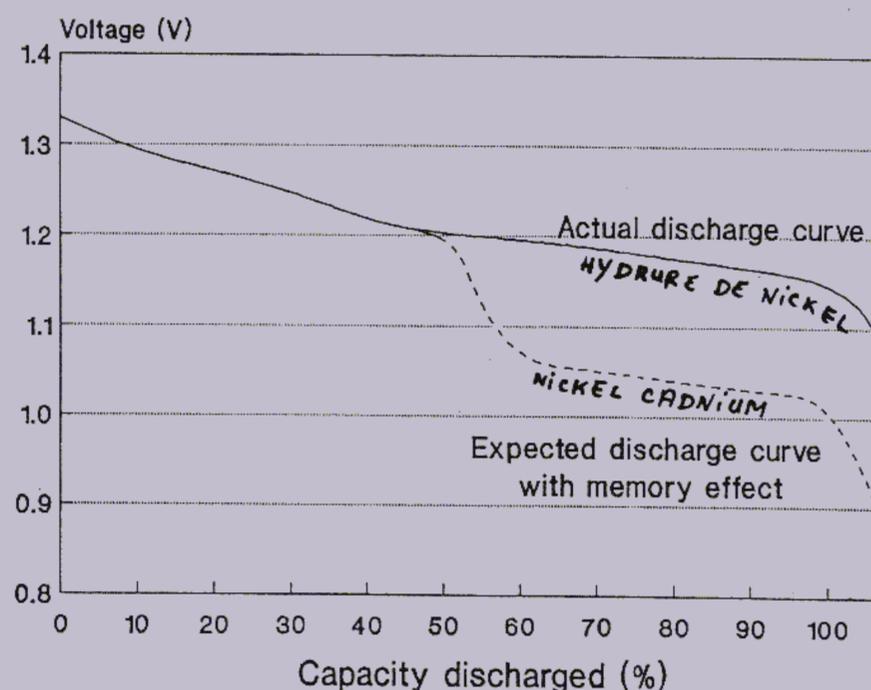
Notamment pour cette raison, des chargeurs spécifiques pour accumulateurs à l'hydrure de nickel vont être mis sur le marché à la fin 1993. Ils seront dotés d'un circuit intégré calculant l'évolution de la tension de charge en fonction du temps et réagiront sur l'intensité de charge afin de charger le plus rapidement tout en évitant la surcharge, toujours préjudiciable à la bonne qualité des accumulateurs et limitant le nombre de cycles charge/décharge.

Ce nouveau système d'accumulateurs nickel/hydrure de nickel constitue sans nul doute le système d'avenir car non pol-

luant et permettant, sans effet mémoire, de stocker dans le même volume le double de l'énergie du système nickel cadmium standard et 30 à 40 % de plus que les dernières et ultimes versions du super NICAD qui commencent à voir le jour et constituent l'étape ultime de ce que l'on peut faire en Nickel Cadmium.

La figure ci-après montre après 190 cycles ce que donne un accumulateur hydrure de nickel par rapport à un accumulateur nickel cadmium qui aurait la même capacité donc un volume supérieur de 30 à 50 %.

Memory Effect Test for NiMH Cylindrical Cell
Discharge curve at 190th cycles



Enfin, en conclusion, il est tout fait possible d'utiliser ces accumulateurs à hydrure de nickel dans la plupart des micros de poche (Sharp-Psion-Portfolio-Casio-HP etc...), puisque leur tension maximum avoisine les 1,5 volt à pleine charge. Pour les calculatrices, le problème résidera dans le fait qu'elles utilisent des piles plus petites de type AAA très souvent, or les accus hydrure de nickel ne sont pas, pour le moment disponibles dans cette catégorie.

Quant à la charge ; elle est tout à fait possible avec un simple chargeur du commerce, à condition de laisser les accus en charge le temps adéquat.

ACCUMULATEURS ULTRA PUISSANTS (1200 mA)

de type R6 AA, tension nominale 1,2V, tension maximale 1,5V.

Rechargeables 1000 fois avec chargeur du commerce

Se mettent en lieu et place des piles - sont écologiques car sans cadmium

Applications : micros de poche - photo - modélisme etc..

Prix : 1 à 2 unités : 46F ttc unitaire

3 à 5 unités : 43F ttc unitaire

6 à 10 unités : 40 F ttc unitaire

au delà : 36 F ttc unitaire

Disponibles sur stock :

MICROCARDS - 42220 BURDIGNES cedex

- tél 77.39.68.13 fax 77.39.19.60 -

HP 48 GX CONTRE... HP 48 SX

par David Winter.

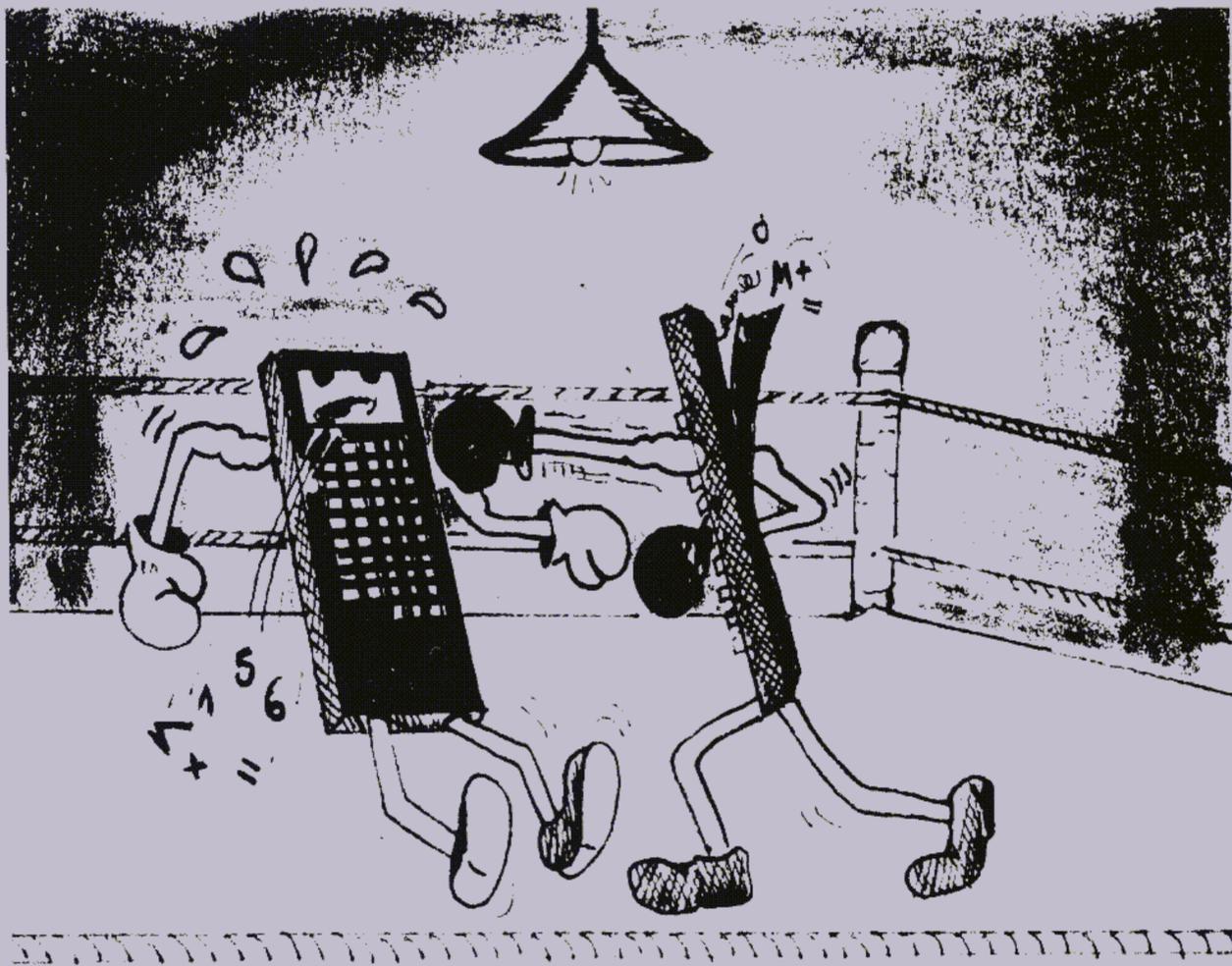
Il y a quelques mois encore, on prononçait HP 48 S ou HP 48 SX. Aujourd'hui, les choses ont changé : HEWLETT PACKARD nous a pondus ses derniers oeufs d'or. Ils s'appellent HP 48 G et HP 48 GX. Ces deux machines sont en train de rendre fous les éternels codeurs.

La HP 48 G : par rapport à la HP 48 S, se distingue par sa ROM de 512 Ko qui comprend le logiciel EQLIB, auparavant réservé à la HP 48 SX, car disponible uniquement sur carte. L'écran a changé : on passe au bleu et au mode hyper-contrasté, ce qui est tout de même plus agréable !

Vous avez dit «communiquer»? Soyez rassurés, il n'y a pas que ce cher KERMIT, HP a pensé au X-MODEM qui est plus rapide. Bien sûr, rien ne vous empêche de communiquer entre HP 48 S et HP 48 G via KERMIT. Passons à la HP 48 GX : il est vrai que 32 Ko de RAM sont assez vite remplis par les programmes d'aujourd'hui (par exemple, PACMANIA prend 90 Ko de mémoire, et il faut un peu plus de 20 Ko de libres pour le faire tourner !!!). Et bien soyez heureux, on a maintenant 128Ko! Bravo HP, c'est assez bien joué. Mais les ports de nos cartes ? Préparez-vous à sauter, le port 1 reste un 128 Ko, mais le 2 peut recevoir jusqu'à ... 4 Mo (oui, 4 mégas!) directement accessibles. Le problème sera de les remplir, ces 4 Mo, car il est déjà assez dur de gonfler une carte 512 Ko.

Notons malgré tout, quelques petites imperfections: en effet, HP a voulu garder une compatibilité entre les HP 48 S (X) et G (X). Cela signifie qu'il faut, pour la G(X), rappeler des objets en ROM, les copier en RAM, puis les évaluer pour les utiliser. En d'autres termes, on perd du temps machine. Mais n'oublions pas que les nouvelles HP 48 GX sont cadencées à environ 4 Mhz, contre environ 2 Mhz chez les 48 SX. De plus ce mode d'accès à la ROM (objets situés plus haut que les 256 premiers Ko directement utilisables) reste valable pour le port 2. Un bon conseil : faites-en un port de sauvegarde !

Mis à part la ROM et les ports, il faut avouer que l'écran, qui est agréable au



regard, est assez lent au niveau vitesse d'affichage. La plupart des jeux déjà programmés doivent être ralentis pour pouvoir admirer les scrollings...

Mais nous avons gardé le must pour la fin : les menus déroulants! Oui vous avez bien lu «menus déroulants»: c'est un superbe coup de maître de la part de HP, grâce à qui l'utilisation de la machine est plus conviviale. Mais souve-

nez-vous de la petite histoire des objets à copier... Dommage que ces menus soient un peu lents. Par exemple, quand on tape vite sur le clavier, la HP n'a pas le temps de tout afficher. Alors, ne soyez pas déçus, soyez patients et amusez-vous bien !

Bonne programmation, et à bientôt sur la HP48GX dans le prochain numéro.

HP MANIAC

BESOIN DE MEMOIRE SUR VOTRE HP48 (SX-GX)?

cartes ram MICROCARDS (fabricant)

CARDRAM 48HP128 (128K) 495F (TTC)

CARDRAM 48HP256 (256K) 830F (TTC)

CARDRAM 48HP512 (512K) 1350F (TTC)

(garantie 1 an - frais de port colissimo 45F)

MICROCARDS - 42220 BURDIGNES cedex - FRANCE

tél 77.39.68.13 fax 77.39.19.60



INITIATION A L'ASSEMBLEUR SUR HP 48

(1ère partie) par Régis DUCHESNE

Vous avez lu, dans le numéro 1 de Poche Computers, l'article de Stéphane DOUAT sur l'assembleur HP48, et ce langage vous intéresse. Malheureusement, vous n'y comprenez pas grand chose. Cet article va vous permettre d'y voir un peu plus clair.

1) Présentation

Soyons lucides dès le départ: ce cours ne s'adresse qu'aux personnes qui ont déjà quelques notions de programmation en général, c'est-à-dire qui savent ce qu'est la ROM, la RAM, une variable, un test, un sous-programme... Si ce n'est pas votre cas, il est vivement conseillé de lire, avant cet article, un des nombreux ouvrages de vulgarisation traitant par exemple de la programmation en langage BASIC ou PASCAL.

Tout d'abord un point de vocabulaire, nous distinguerons ici le langage assembleur et l'assembleur lui-même, la nuance est délicate: le langage assembleur, c'est le langage dans lequel vous allez écrire vos programmes (le listing d'un programme s'appelle le source), comme le BASIC ou le PASCAL, avec des instructions compréhensibles. L'assembleur, c'est le programme qui va assembler votre source, c'est-à-dire le transformer en un objet en langage machine, directement exécutable par la machine (L'objet obtenu s'appelle d'ailleurs un exécutable).

Plusieurs fois dans cet article, il sera fait référence à «Voyage au centre de la HP48», excellent ouvrage de Paul COURBIS et Sébastien LALANDE, aux éditions ANGKOR, considéré comme la bible du programmeur en langage assembleur, et ceci toutes versions de la machine confondues (HP48S, HP48SX, HP48G, et HP48GX). Cet ouvrage est indispensable et devra obligatoirement figurer dans la bibliothèque de tout programmeur en langage assembleur qui se respecte.

Dans la suite de cet article, les personnes citées se verront accompagnées d'un mot entre parenthèses, ce mot correspond à un pseudonyme. En effet, il existe plusieurs serveurs minitels rassemblant la famille sans cesse proliférante des programmeurs HP, et comme chacun sait, le minitel devant conserver l'anonymat, on a recours aux pseudonymes. Sur ces serveurs, on peut discuter en direct ou en différé avec d'autres programmeurs, et l'on peut télécharger des programmes directement sur la HP. Pour plus de détails, je vous conseille vivement d'aller ouvrir un boîte aux lettres électronique sur l'un de ces serveurs, par exemple sur le 3615 RTEL ou même mieux sur TOP JEF, un

serveur privé (la connection ne coûte que le prix de la communication téléphonique) au (1) 47. 95. 28. 10. Mon pseudonyme est HPReg.

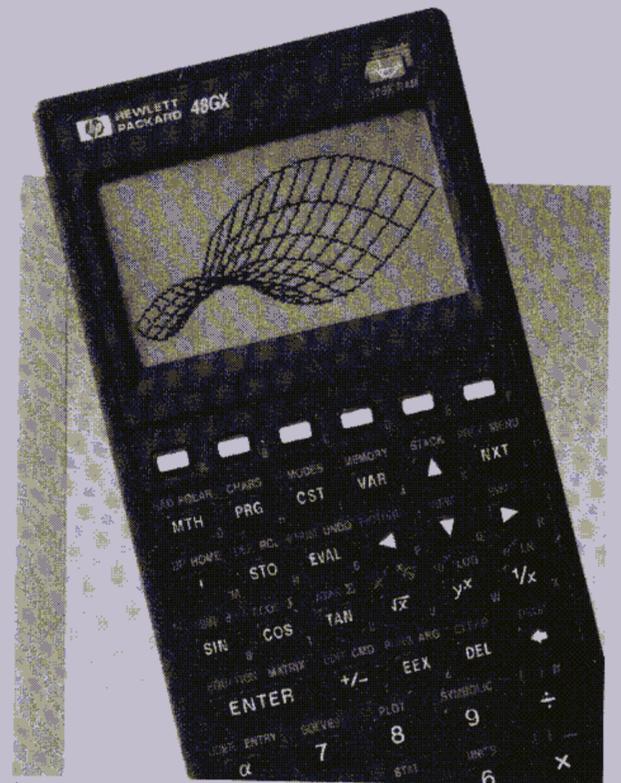
Pour écrire vos sources sur HP48, on utilise une chaîne de caractères. Vous vous apercevrez rapidement que l'éditeur intégré de la machine est très vite dépassé. Pour pallier ce défaut, j'utilise personnellement le très complet STRING WRITER version 2.1 de Jean-Yves AVENARD (de pseudonyme GHERKIN), qui a l'avantage d'être petit, rapide, et de permettre le couper-coller de blocs... Dans cet article, l'assembleur utilisé est ASMFLASH, de Phong N'GUYEN (pseudonyme HPNINJA) dans sa version 6.1 adaptée par Jean-Yves AVENARD (GHERKIN). C'est un assembleur sur HP48, donc toute l'opération se fait sur votre machine. Pour vous procurer ces programmes, allez sur TOP JEF, tout vous sera expliqué.

Vous voilà quasiment prêt pour commencer, mais sachez que le développement de programmes en langage assembleur demande beaucoup de mémoire, il est donc conseillé dès que vous passerez à de gros projets, de vous procurer une carte 128Ko RAM (Je me suis laissé dire que la société MICROCARDS en proposait pour pas cher...)

2) Fonctionnement du microprocesseur

Ce paragraphe apporte des notions nouvelles à certains qui le trouveront donc un peu fastidieux à lire. Ne vous découragez pas, relisez-le plusieurs fois s'il le faut, puis passez à l'exemple...

Le microprocesseur de la HP48, le cerveau de la machine, a été baptisé SATURN par HP. Il s'agit d'un microprocesseur 4 bits, c'est-à-dire que son vocabulaire est limité à seize mots, que l'on pourrait numéroter de 1 à 16, mais que l'on préfère numéroter en hexadécimal (base seize), c'est-à-dire de 0 à F, en employant les symboles A à F pour les 6 derniers mots. L'avantage de la numérotation par bit est qu'elle correspond exactement à ce qui se passe physiquement dans le microprocesseur, on représente un bit par deux états, le courant passe (état 1) ou il ne passe pas (état 0). Lorsque l'on utilise un groupe de 4 bits, on parle de quartet. La conversion binaire (base deux, c'est la



base du bit=binary digit en anglais) en hexadécimal est très simple:

soit à convertir 1100 (binaire) en hexadécimal, on écrit le tableau suivant:

2^3	2^2	2^1	2^0
1	1	0	0

puis on somme:

$1*2^3+1*2^2+0*2^1+0*2^0=1*8+1*4+0*2+0*1=12$ (décimal = C (hexadécimal)

Aux bits 1100 correspond le quartet C. Pour faire la conversion inverse, on procède dans l'autre sens :

Soit à convertir A(hexadécimal) en binaire, on décompose A en puissances de deux, $A(\text{hexa})=10(\text{décimal})=8+2=1*2^3+0*2^2+1*2^1+0*2^0$ donc vaut 1010 en binaire. Au quartet A correspondent donc les bits 1010.

Pour faire ces opérations sur plus de chiffres, on décompose toujours les nombres binaires en tranches de 4 bits et les nombres hexadécimaux en tranches de 1 quartet, exemple :

soit à convertir 0101111100111110(binaire) en hexadécimal, on décompose en tranches en partant de la droite, et on fait comme ci-dessus, 0101 1111 0011 1110 ce qui donne 5 F 3 E, le résultat est donc 5F3E. Retenez bien ceci, on s'en sert constamment, mais que cela ne vous effraie pas, cela vient rapidement avec un peu de pratique!. Pour vous habituer, nous ne parlerons dorénavant plus qu'en hexadécimal...

Le SATURN peut aller lire ou écrire des données en mémoire. Il faut savoir que l'unité de mémoire est le quartet, et pour



savoir ou l'on doit lire ou écrire, il est nécessaire de connaître l'adresse de ce quartet (comment voulez-vous qu'une lettre vous arrive si le facteur ne connaît pas votre adresse?). Pour le SATURN, les adresses vont de 00000 à FFFFF. La seule chose à retenir est que le SATURN, lors d'une opération de lecture ou d'écriture, «retourne» les quartets, mais nous aurons l'occasion d'y revenir...

Le SATURN possède des registres, qui sont un peu comme des variables, dans lesquels on peut stocker quelques données et sur lesquels on peut pratiquer des opérations élémentaires. Voici les plus utiles de ces registres qui peuvent être classés en 6 catégories :

Registres d'entrées/sorties:

Très peu utilisés par le débutant, le registre OUT(12 bits) permet d'envoyer du courant sur des fils du clavier ou du buzzer, et le registre IN(16 bits) permet de recevoir du courant sur des fils du clavier. Ces deux registres servent donc à la gestion du clavier et du son.

Registres drapeaux:

La CARRY(1 bit) est le registre de retenue, après exécution par le microprocesseur d'une instruction correspondant à une opération algébrique sur un registre, la CARRY est mise à 1(on dit qu'elle est armée) si l'opération a donné lieu à une retenue: somme supérieure à ce que peut contenir le registre, ou soustraction donnant un résultat négatif. La CARRY est aussi armée lorsque une instruction de test est vraie.

ST (16 bits) est le registre de statut, il s'agit de drapeaux que l'utilisateur peut armer ou désarmer(mettre à 0).

Registres pointeurs:

D0 et D1(20 bits chacun), contiennent la plupart du temps une adresse mémoire (on dit qu'ils pointent sur cette adresse). Ils sont utilisés pour les opérations de lecture/écriture en mémoire

PC(20 bits), c'est le program counter, il contient l'adresse de la prochaine instruction en langage machine à exécuter.

Registres de sauvegarde:

RSTK(8*20 bits) est un registre un peu spécial car c'est en fait une mini pile de huit étages de 20 bits, utilisée notamment pour la sauvegarde des adresses de retour lors d'appels à des sous-routines, mais nous aurons l'occasion d'y revenir...

R0,R1,R2,R3,R4(64 bits chacun) sont des registres utilisés la plupart du

temps pour la sauvegarde des registres de calcul qu'il me faut donc présenter:

Registres de calcul:

A,B,C,D (64 bits chacun) , où A et C sont spécialement dédiées aux opérations de lecture/écriture en mémoire.

Le pointeur de champ:

P(4 bits): en général, P= 0

Pourquoi pointeur de champ? Parce que le SATURN définit des champs (des subdivisions) sur tous les registres de 64 bits. Voici un tableau du nom et de la disposition des champs (rappel:64 bits=16 quartets):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W															
S	M										XS	B			
												A			
												X			

Comme P contient toujours une valeur de 0 à F, il définit lui deux champs: le champ P qui est en fait disposé sur l'unique quartet de numéro P, et le champ WP qui est disposé sur les quartets 0 à P. Exemple si P=2, on a le cas particulier où le champ P est le champ XS et où le champ WP est le champ X.

Exemple: J'entends d'ici ceux qui disent: «c'est bien beau tout ça, mais maintenant, je fais quoi?». Revenons un peu dans le monde du concret, essayons d'écrire la valeur BA3 contenue dans le registre C champ X (que l'on notera dorénavant C X) à l'adresse mémoire 00125, voilà à quoi ressemblera le source:

LC BA3 %charge les trois premiers quartets de C, donc C X, avec la valeur BA3

D0= 00125 %charge le registre D0 avec l'adresse où l'on veut écrire

DAT0=C X %écrit C X à l'adresse pointée par D0

En fait, en mémoire, puisque le SATURN «retourne» tout, on trouvera à l'adresse 00125 le quartet 3, à l'adresse 00126 le quartet A, et enfin, en 00127 le quartet B. Facile non?

A noter qu'en pratique, on a rarement à se soucier de ce «retournement», car si on essaye de lire sur trois quartets (par exemple dans A X) la valeur contenue à l'adresse 00125, on retrouvera bien A X=BA3). Le double «retournement» remet tout en ordre!

En fait, P agit aussi sur le chargement des constantes dans les registres A et C: Dans l'exemple précédent, P valait 0, mais lorsque l'on écrit l'instruction LC BA3, le champ numéro P de C prends la valeur B, le champ de numéro P+1 la valeur A, et le champ de numéro P+2 la valeur 3. Et ceci de manière cyclique:

Exemple : Soit le source suivant:

P= 15% initialise P à la valeur décimale 15
LC ABC %charge C avec ABC, toujours à partir du champ P

On trouvera dans les 16 quartets de C :

A	?	?	?	?	?	?	?	?	?	?	?	?	?	?	C	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Application : Vous êtes maintenant apte, avec un peu d'aide, à écrire votre premier programme en langage assembleur: Il va s'agir d'effectuer un SWAP des deux premiers niveau de la pile, comme le ferait la fonction du RPL(le langage décrit par HP dans les manuels vendus avec la machine), mais sans aucune vérification du nombre d'arguments présents sur la pile. Pour ce faire, il faut savoir que la machine utilise de manière interne cinq registres:

D0: qui pointe sur le prochain objet à exécuter.

D1: qui pointe sur le premier étage de la pile où ne sont en fait stockées que les adresses des objets que l'on voit s'afficher sur la pile à l'écran, et non les objets eux-mêmes.

B champ A(noté B A)

D champ A(noté D A) qui contient la place mémoire restante divisée par 5
Avant de rendre la main au système, il conviendra donc de redonner à ces registres des valeurs adéquates si on les modifie.
Passons au source:

A=DAT1 A %charge A A avec 5 quartets pointés par D1, comme D1 pointe sur l'objet au niveau 1 de la pile, A A va contenir l'adresse de l'objet au niveau 1 de la pile.
D1=D1+ 5 %D1 pointe maintenant sur une adresse mémoire contenant l'adresse de l'objet au niveau 2 de la pile.

C=DAT1 A %charge C A avec l'adresse de l'objet au niveau 2 de la pile.

DAT1=A A %écrit l'adresse de l'objet qui était au niveau 1 au nouveau niveau 2.

D1=D1- 5 %D1 pointe à nouveau sur le niveau 1 de la pile.

DAT1=C A %écrit l'adresse de l'objet qui était au niveau 2 au nouveau niveau 1.

Remarquons que nous n'avons modifié ni B A, ni D A, ni D0, ni D1(puisque il est revenu à sa valeur initiale). Mais il y a un hic: si on assemble ce programme, il ne marchera pas. Nous avons tout bonnement oublié de rendre la main au système! En fait, comment faut-il faire? Il suffit de se brancher sur l'objet suivant à exécuter.

Tout programme en langage assembleur doit donc se terminer par les instructions suivantes:

A=DAT0 A %charge A avec l'adresse de la prochaine instruction à exécuter.

D0=D0+ 5 %la prochaine instruction à exécuter devient la suivante.

PC=(A) %Il est impossible, sans me lan-

INITIATION A L'ASSEMBLEUR SUR HP 48... INITIATION A L'ASSEMBLEUR SUR

cer dans beaucoup plus d'explications, de vous dire, chers débutants, pourquoi.

3) Utilisation de ASMFLASH

Le menu de la librairie 1217(ASMFLASH version 6.1) possède quatre cases que nous allons voir en détail:

- ABOUT: Sert d'aide mémoire sur le programme, un peu compliqué pour une initiation, reprenez simplement que le symbole % (obtenu par pression sur Alpha, Right Shift, et Tan) sert de séparateur pour dire à l'assembleur que ce qui suit sur la ligne de votre source est un commentaire, que le symbole * sert de déclarateur de label, et que le symbole \$ sert à forcer l'assembleur à mettre une suite de quartets que vous voulez à la place où vous voulez dans l'exécutable. Enfin, reprenez que le symbole @ (obtenu par pression sur Alpha, Right Shift, Enter) sert à dire à l'assembleur où se situe la fin de votre source.)

- OPT: propose un menu d'options de compilation, que l'on change en appuyant dessus, les programmeurs sont maintenant tous d'accord pour toujours choisir CODE, 0-15, PC, JMP, OFF. Pour sortir et valider votre choix, appuyez sur EXIT.

- ED: sert lors de l'opération d'assemblage, lorsque l'assembleur a détecté une erreur, il vous suffit alors de taper ED pour pouvoir directement corriger la ligne où est survenue l'erreur.

- ASM: lance l'opération d'assemblage d'un source mis au niveau 1 de la pile.

Votre source doit finalement se présenter exactement comme ceci, ASMFLASH étant comme tout les assembleurs très pointilleux sur la syntaxe, attention donc à ne pas omettre ou ajouter des espaces ou un saut de ligne(par exemple, on ne met plus rien après le @ final.)

```
«%SWAP sans vérification d'arguments
A=DAT1 A
D1=D1+ 5
C=DAT1 A
DAT1=A A
D1=D1- 5
DAT1=C
%Retour au système
A=DAT0 A
D0=D0+ 5
PC=(A)
@»
```

Voilà, vous avez donc fini de rentrer ce source, stockez-le sous le nom 'SWAP.SRC', et mettez le au niveau 1 de la pile avant de taper ASM pour l'assembler. Si aucune erreur ne s'est produite, stockez l'exécutable dans 'SWAP.EXE',

sinon vérifiez la syntaxe de votre source, la place mémoire disponible, ainsi que le choix de vos options de compilation.

Maintenant placez 2 objets quelconques sur la pile, exécutez SWAP.EXE et, miracle du langage assembleur, les deux objets ont changé de niveau!!!

Remarque pratique: Il vous arrivera immanquablement un jour, au hasard de votre programmation en langage assembleur, d'assembler un source qui bien que syntaxiquement correct, comporte des erreurs de programmation(on appelle ça un bug). Il se peut aussi que vous oubliez de mettre sur la pile les arguments dont se sert votre programme (ici deux objets quelconques). Si vous exécutez votre programme, il en résultera quelque chose d'imprévisible: Essayez immédiatement de faire un arrêt système (en pressant ON et C simultanément). Si cela ne marche pas, et que votre programme semble bloquer la machine, retournez la machine et enlevez lui (avec l'ongle) ses deux patins anti-dérapage les plus éloignés du logement à pile. L'un d'eux découvrira un petit trou marqué d'un petit R(comme RESET). Alors, enfoncez un trombone dans ce trou jusqu'à sentir une résistance élastique, puis une forte résistance solide. Relâchez, appuyez sur ON. Après un double arrêt système, votre machine devrait fonctionner à nouveau. Moralité: le niveau d'un programmeur en langage assembleur est inversement proportionnel au nombre de trombones qu'il possède dans la housse de protection de sa HP!!!

Remarque importante: Rappelez-vous simplement que sauf cas très particulier, il est impossible de casser physiquement sa machine à l'aide d'un simple programme. Le Soft ne détruit jamais le Hard. Je ne connais qu'une seule exception à cette règle: (N'essayez pas!!!) En écrivant le quartet 2 à l'adresse 00102, à l'aide du source suivant par exemple:

```
«%Seul programme dangereux
LC 2 %Affectation du quartet 2 dans C
champ P(P= 0 par défaut...)
AD0EX %Sauvegarde de D0, qui ne doit pas être modifié
D0= 00102 %Affectation de l'adresse où écrire dans D0
DAT0=C P %Ecriture à l'adresse pointée par D0 de C P, soit un quartet...D0=A
%Récupération de la valeur initiale de D0
%Retour au système
A=DAT0 A
D0=D0+ 5
PC=(A)
@»
```

Cette adresse bloque le balayage écran sur la ligne actuellement balayée, ce qui implique une surtension sur cette ligne, surtout si l'écran est très contrasté. Moralité: à part ceci, vous pouvez vous permettre tous les délires avec votre machine. Néanmoins, HP et moi-même déclinons toute responsabilité pour ce qu'il pourrait advenir de votre machine ou de de votre santé, en lisant cet article!!!

4) Le jeu d'instructions du SATURN

Vous allez me dire: «Oui, mais la programmation, ça ne consiste pas qu'à écrire des valeurs dans des registres ou en mémoire». C'est faux. En langage assembleur, on ne fait que ça, et vous allez voir que ces instructions élémentaires (je les ai à peu près toutes mises, il en manque certaines inutiles aux débutants), exécutées en une infime fraction de seconde, vont vous permettre d'élaborer des programmes très complexes. Ce paragraphe n'est pas très intéressant du point de vue de la programmation en elle-même, il ne fait qu'énumérer les instructions que vous devez absolument connaître pour programmer (Ne vous en faites pas, ça vient en les utilisant). Le paragraphe suivant explique comment utiliser ces instructions.

Dans toute la suite du paragraphe III, on désignera par la lettre «f» un champ quelconque d'un registre, parmi P, WP, S, XS, B, X, A, M, W. On désignera de même par la lettre «x» un nombre compris entre 0 et 15 inclus.

4.1) Instructions d'affectation

a) chargement de constantes

LA qqqqqqqqqqqqqqqq %Charge le registre A avec un nombre de quartets inférieur ou égal à 16: par exemple, LA 1BC charge le champ X de A avec la valeur 1BC.

ABIT=0 x %Mise à zéro du bit numéro x du registre A, les bits sont numérotés de droites à gauche.

ABIT=1 x %Mise à un du bit numéro x du registre A

Remarque: Ces trois instructions aussi en remplaçant A par C.

A=0 f %Mise à zéro du champ f du registre A

Remarque: Ces trois instructions existent aussi en remplaçant A par B,C ou D.

P= x %Affecte la valeur x à P.

D0= qqqqq %Charge le registre D0 avec un nombre de quartets donné: 2, 4 ou 5: par exemple, D0= 12345 suivi de D0= AB attribue à D0 la valeur 123AB

Remarque: Cette instruction existe aussi en remplaçant D0 par D1.

ST=0 x %Arme le drapeau numéro x. Il est conseillé de ne pas utiliser les drapeaux



10, 13, 14, 15 qui sont utilisés de manière interne par la machine.

ST=1 x %Désarme le drapeau numéro x.
CLRST %Désarme tous les drapeaux.

b) Chargements de valeurs

A=B f %Attribue au champ f de A la valeur du champ f de B
A=C f %Idem mais avec la valeur du champ f de C
B=A f %Attribue au champ f de B la valeur du champ f de A
B=C f %Idem mais avec la valeur du champ f de C
C=A f %Attribue au champ f de C la valeur du champ f de A
C=B f %Idem mais avec la valeur du champ f de B
C=D f %Idem mais avec la valeur du champ f de D
D=C f %Attribue au champ f de D la valeur du champ f de C

C=ST %Charge les drapeau de numéro 0 à 11 dans les 12 bits de C champ X
C=P x %Charge la valeur de P dans le quartet numéro x de C

P=C x %Charge P avec la valeur du quartet numéro x de C

D0=A %Charge A A dans D0
D0=AS %Charge les quartets de numéro 0 à 3 de A dans ceux de D0

Remarque: Ces deux dernières instructions existent aussi en remplaçant D0 par D1

A=IN %Ecrit les 4 quartets du registre IN dans les quartets 0 à 3 de A

C=IN %Idem avec le registre C

OUT=C %Charge le registre OUT avec les trois quartets de C champ X

OUT=CS %Charge le quartet 0 du registre OUT avec le quartet 0 de C.

c) Sauvegardes et récupérations

A=R0 f %Charge le contenu de R0 champ f dans celui de A champ f

R0=A f %Idem dans l'autre sens.

Remarque: Ces deux dernières instructions existent aussi en remplaçant R0 par R1, R2, R3, R4 et A par B, C, D.

C=RSTK %Sauve la valeur de C champ A sur le premier étage de la pile des retours, l'ancien niveau 8 de cette pile est perdu.

RSTK=C %Dépile le premier étage de la pile des retours et stocke sa valeur dans C champ A, le niveau 8 de cette pile est mis à 0000.

d) Transferts avec la mémoire

A=DAT0 f %Charge A champ f avec les quartes lus en mémoire à l'adresse pointée par D0.

DAT0=A f %Ecrit en mémoire à l'adresse pointée par D0 les quartets du champ f de A.

DAT0=A x+1 %Ecrit x+1 quartets en mémoire à l'adresse pointée par D0.

R : Ces trois dernières instructions existent aussi en remplaçant A par C et D0 par D1.

e) Echange de contenus de registres

ABEX f %Echange les champs f de A et de B

ACEX f %Idem pour A et C

BCEX f %Idem pour B et C

CDEX f %Idem pour C et D

AD0EX %Echange le contenu de A champ A et D0

AD0XS %Echange les quartets 0 à 3 de A et D0

R : Ces deux dernières instructions existent aussi en remplaçant A par C et D0 par D1.

AR0EX f %Echange les champs f de A et de R0.

R: Cette instruction existe aussi en remplaçant R0 par R1, R2, R3, R4 et A par B, C, D.

CPEX x %Echange la valeur du quartet numéro x de C avec la valeur de P

CSTEX %Echange les 12 drapeaux avec les 12 premiers bits de C champ X

4.2) Opérations mathématiques

a) Incrémentations

A=A+x+1 f %Incrémte le champ f de A de x+1 unités

A=A-x-1 f %Idem mais décrémte. Par exemple, A=A-10 S décrémte de 10 le champ S du registre A

Remarque: Ces deux dernières instructions existe aussi en remplaçant A par B, C, D.

P=P+1 %Augmente P d'une unité

P=P-1 %Diminue P d'une unité

D0=D0+ x+1 %Augmente D0 d'une unité

D0=D0- x-1 %Diminue D0 d'une unité

Remarque: Ces deux dernières instructions existent aussi en remplaçant D0 par D1.

b) Additions

A=A+B f %Somme les champs f de A et B, le résultat est mi dans le champ f de A

A=A+C f %Idem pour A et C

B=B+A f %Idem pour B et A

B=B+C f %idem pour B et C

C=C+A f %Idem pour C et A

C=C+B f %Idem pour C et B

C=C+D f %Idem pour C et D

D=D+C f %Idem pour D et C

A=A-C f %Soustrait au champ f de A le champ f de C et place le résultat dans celui de A

A=B-A f %Idem

B=B-A f %Idem

B=B-C f %Idem

B=C-B f %Idem

C=C-A f %Idem

C=A-C f %Idem

C=C-D f %Idem

D=D-C f %Idem

D=C-D f %Idem

C+P+1 %Ajoute la valeur de P plus 1 à C champ A

c) Opération logique AND

C'est une opération prenant deux bits comme argument et renvoyant un bit comme résultat: 0 AND 0=0, 0 AND 1=0, 1 AND 0=0, 1 AND 1=1.

Les instructions s'écrivent pareil que pour les additions(pas les soustractions!) en remplaçant le symbole + par le symbole &.

Par exemple,

C=C&A f %Fait un AND logique bit à bit entre les champs f de C et A, et place le résultat dans le champ f de C.

d) Opération logique OR

C'est une opération prenant deux bits comme argument et renvoyant un bit comme résultat: 0 OR 0=0, 0 OR 1=1, 1 OR 0=1, 1 OR 1=1.

Les instructions s'écrivent pareil que pour l'opération logique AND, en remplaçant le symbole & par le symbole !.

Par exemple,

C=C!A f %Fait un OR logique bit à bit entre les champs f de C et A, et place le résultat dans le champ f de C.

e) Opération logique NOT

C'est une opération prenant un bit comme argument et renvoyant un bit comme résultat: NOT 0=1, NOT 1=0.

A=-A-1 f %Effectue un NOT logique bit à bit sur le champ f de A

A=-A f %Calcule l'opposé de A champ f en base 2, c'est-à-dire le nombre qui ajouté à A champ f, donnerait zéro sur le champ f.

Remarque: Ces deux dernières instructions existent aussi en remplaçant A par B, C, D

f) Multiplications

A=A+A f %Multiplie A par 2 sur le champ f.

ASL f %Multiplie A par 16 sur le champ f.

ASRB f %Divise A par 2 sur le champ f.

ASR f %Divise A par 16 sur le champ f.

ASLC %Rotation circulaire vers la gauche des quartets de A sur l'ensemble du registre: le quartet numéro 0 devient le quartet de numéro 1, le quartet de numéro 1 celui de numéro 2, ... et le quartet de numéro 15 celui de numéro 0.

ASRC %Rotation circulaire vers la droite des quartets de A sur l'ensemble du registre: le quartet numéro 0 devient le quartet de numéro 15, le quartet de numéro 1 celui de numéro 0, ... et le quartet de numéro 14 celui de numéro 15.

Remarque: Ces six dernières instructions existent aussi en remplaçant A par B, C, D

Fin de la 1ère partie. Dans le prochain numéro, nous étudierons les instructions de branchement et nous vous proposerons quelques exemples commentés ainsi que quelques "ruses" à employer.



AVEZ-VOUS BESOIN DE MEMOIRE POUR VOTRE HP48 (SX-GX) ?

Société d'électronique, spécialisée en micro électronique depuis plus de 10 ans, nous fabriquons des cartes mémoire pour HP48 depuis 128KB jusqu'à 512KB. Celles-ci sont compatibles HP48SX et HP48GX, sur **port 1 ou 2 indifféremment**. En effet, pour les cartes 128KB, pas de problème particulier. Par contre, seul le port 2 sur HP48GX est prévu pour accepter jusqu'à 4MB (normes Hewlett Packard).

Avec nos cartes, le passage de banque est manuel, par interrupteur ; ce qui permet une utilisation aussi bien sur HP48SX ou HP48GX sur port 1 ou port 2. La taille des banques est de 128KB chacune. La compatibilité HP est de 100%.

De plus, les données sont conservées par micro-accumulateur (et non par pile), rechargeable depuis la HP, avec une sur-consommation négligeable et une durée de vie de 10 ans. Un micro interrupteur permet l'accès ou le blocage d'écriture.

MICROCARDS produit ses cartes en toute sécurité pour l'utilisateur (fiabilité égale à celle de chez HP !) et s'engage pour une garantie d'une année par échange standard.



3 bonnes raisons

d'acheter MICROCARDS

- 1) la compétence électronique
- 2) les 2 nouveautés mondiales:
 - *sauvegarde par accumulateur et
 - *switching par interrupteur manuel
- 3) des prix à couper le souffle !
directement du producteur
au consommateur

LES PRIX ?

référence CARDRAM 48HP128 (128KB)	495,00F TTC	} plus 45F de port colissimo recommandé
référence CARDRAM 48HP256 (256KB)	830,00F TTC	
référence CARDRAM 48HP512 (512KB)	1350,00F TTC	

MICROCARDS 42220 BURDIGNES cedex - FRANCE

tél (33) 77.39.68.13 fax (33) 77.39.19.60

vente par correspondance exclusivement (ouverture du lundi au vendredi inclus : de 9h à 13h et de 15h à 19h)



JEU DE TETRIS SUR HP 28 S

Ce jeu de moins de 1,5 Ko se joue seul. Il est inspiré du célèbre TETRIS russe. Pour jouer, on retourne le clavier de gauche sous la machine et on tourne celle-ci d'un quart de tour dans le sens inverse des aiguilles d'une montre. Pour diriger la pièce qui tombe, on utilise INS pour aller à gauche et SHIFT (la touche rouge) pour aller à droite. Pour faire tourner la pièce sur elle même, on utilise STO. Pour accélérer la chute de la pièce, on utilise '. L'activation ou la désactivation des effets sonores dépend du flag 51, comme en utilisation normale de la machine. Après avoir sélectionné le programme TRGST, c'est parti !!!

Le but du jeu est de placer les pièces qui tombent du creux d'une colonne, afin de constituer des lignes qui font toute la largeur de l'écran, sans trou. Toute ligne constituée disparaît, faisant redescendre le tas de pièces posées précédemment d'un cran. Bien sûr, le tas de pièces ne doit pas atteindre le haut de la colonne. La prochaine pièce à tomber est indiquée en haut à gauche de l'écran. Cette pièce étant une pièce à part entière, attention de ne pas "coincer" votre pièce en la serrant trop à gauche dès le départ. Le niveau et le nombre de lignes restant à compléter sont affichés en permanence. Qui arrivera, au mépris d'une vitesse de jeu croissante ainsi que d'obstacles apparaissant dans l'écran (à partir du niveau 3), à finir le niveau 10...?

Pour ne pas faire d'erreurs fatales (risque de Memory Lost) en entrant ce programme bien suivre les étapes successives. Si vous avez déjà rentré les programmes ASSR et CHKR lors de la saisie d'un autre jeu, passez à l'étape V.

I --- CREATION DU PROGRAMME ASSL:

Rentrez ce programme SANS ERREUR:

```
<< -> C << HEX "" 1 C SIZE FOR X "#" C X 1 + DUP SUB C  
X X SUB ++ STR -> B -> R CHR + 2 STEP 2 #8253h SYSEVAL  
#4F3Dh SYSEVAL >> >>
```

Stockez-le dans ASSL.

II --- CREATION DU PROGRAMME CHKL

Rentrez ce programme SANS ERREUR:

```
<< -> C << HEX 64 STWS #0h 1 C SIZE FOR X "#" C X DUP  
SUB + STR -> X * + NEXT >> >>
```

Stockez-le dans CHKL

III --- CREATION DU PROGRAMME ASSR

Rentrez la chaîne suivante **sans espace ni saut de ligne**:

```
76C20 0C8E1 1E3C0 69C20 F6000 8F180 50147 068F7 2F40A  
F2306 EE474 81E06 E68F9 38200 7D507 13606 13516 914A3  
103EA 31909 EA703 07EA1 59016 1170C D5CD8 DDEDD  
109F2 0
```

Dupliquez la chaîne et vérifiez-la avec CHKL. Si le résultat est différent de #E016h, elle a été mal rentrée, recommencez cette étape. Sinon exécutez ASSL et stockez le résultat (les 3 System Object) dans ASSR. Ensuite, purgez ASSL.

IV --- CREATION DU PROGRAMME CHKR

Rentrez la chaîne suivante **sans espace ni saut de ligne**:

```
76C20 07971 CD570 EEE80 01670 69C20 4B000 8F180 50143  
13117 4AF0A F3101 10214 3174D 2305E A81C1 0014B 3103B  
6A319 09A8 0307B 6AAE6 AE711 1E410 1AF13 1F3A75A775  
50A78A6E50F112A70102 17111 11188 A6BA8 F8B05 0147E  
71431 33179 11A15 57131 14216 4808C 09F20
```

Dupliquez la chaîne et vérifiez-la avec CHKL. Si le résultat est diffé-

rent de #1D0FBh, elle a été mal rentrée, recommencez cette étape. Sinon exécutez ASSR et stockez le résultat (le #0h DUP + SWAP System Object) dans CHKR. Ensuite, purgez CHKL.

V --- CREATION DU PROGRAMME TRGST

Rentrez la chaîne suivante **sans espace ni saut de ligne**. Après chaque saisie de chaîne, il faut dupliquer la dernière chaîne saisie et la vérifier avec CHKR. Si le résultat n'est pas celui indiqué, il faut retaper la chaîne. A la fin, vous devriez normalement vous retrouver avec 12 chaînes dans la pile.

```
76C20 4B211 9C211 D78E0 73D20 10C47 3D201 0E40D 9E04B  
211DD 0A0E4A2033 000A2A2A2A 2A2A2A2024 52594 74355  
44502 A2A2A 2A2A2 A2A2A 5832E 4A203 90004 45534 84543  
5E454 02020 20202 02022 38F2 0383F 29313 A0255 47494  
35024 53413 02020 20202 02020 20202 63239 393A0 C4953  
45454 02D49 4C494 45149 42554 02023 547D2 34952 5ED21  
16A5A 08031 1A37A 04B21 14B21 1535E 04467 0E57A 0E4A2  
01200 00202 0202E 49465 54145 50202 02027 3D201 0E4EA  
970EE E803A BB19C 211A3 7A073 D2010 E43F2 11670 90CF9  
E073D 2010C 471AE 0022A 069C2 0C400 08F18 050AF 2BFE1  
B048F F15C7 37100 00008 AFA31 B3"
```

CHKDR: #88B89h

"

```
16715 87A6E 55F8F 8B050 14216 4808C 3C0A0 23311 67090  
79CB0 CD570 2E3E0 76C20 73D20 10E4E D211A  
4B800 9F208 F3E07 6C209 C2117 3D201 0E4ED 21167  
090B6 5E03C 0A0D1 31167 09079 CB03C 0A017 31167  
09079 CB069 C20B3 0008F 18050 75F4D 51747 CE4DA 7D567  
8B68F 8B050 179E7 E7142 16480 8CF06 E009F 20974 E0535  
E0A26 7073D 2010C 4"
```

CHKR: #32C35h

"

```
EA970 69C20 61000 3444C FF842 8D04A B51C3 112E3  
E076C 2073D 2010E 42231 1A4B8 009F2 08F3E 076C2  
03F21 10DF8 073D2 010E4 ED211 44190 09F20 954E0  
73D20 10E49 74E00 DF800 FEF2E EE806 9C208 56008  
F1805 07EF3 10B17 474F3 10A1B 5300C 30F15 C011A  
10C1B 8FFFF AF014 2AF13 07A85 AF32F 32800 209C9  
B0E6A 7565F FA779 F4"
```

CHKR: #3389Dh

"

```
80B70 B7781 DA6E5 BE102 85084 1114D 2E610 93151 10879  
83112 8517F 73111 D2304 CA101 11484 07863 862C1  
D2CE1 0C31C 31B5D 00078 45600 31148 507F3 33208 07815  
93AA5 85084 11147 42311 1CC10 18401 14721 3111E 41018  
50"
```

CHKR: #106ABh



JEUX DE TRETIS SUR HP 28 S ... JEUX DE TRETIS SUR HP 28 S ... JEUX DE TRETIS

"
 85111 47DF2 87212 85084 11147 BE211 1CC10 18511
 1479D 23204 072B4 93AA5 85084 11147 EB211 1E410 18401
 147CA 2111C C1018 50851 11477 92872 12850 84111
 47582 111E4 10185 11147 37232 8007C 4493E 60668 08508
 41114 7452"
 CHKR: #10EBEh
 "

11C80 F10C8 84402 080F1 10C84 01147 33211 C80F1 0D540
 2380F 110C8 50851 11471 12872 03850 84111 47FF1 11C80
 F10C8 84402 080F1 10C85 11147 ED184 33201 074B3
 93A90 8536F 00340 0A00C E5DF8 F8B05 01431 31179
 87311 147CE"
 CHKR: #1428Ch
 "

49014 56A5E 11B14 58508 41114 73811 10CC1 00840 11471
 71110 E4100 85085 11147 C5187 2D385 08411 147A4 1110C
 C1001 14851 78313 1101B 04000 74238 73606 4ED61 4F310
 11B0A 00078 0331B 3AE7D 210C1 F848F FAF2B FE15B
 72791 2A117 F177"
 CHKR: #170CCh
 "

20A6F A6FA6 F5DD6 88020 31209 EF831 33131 10013 016F1
 67DB1 09D5A 6DA6D A6D15 A7159 71671 77A6D 5EE37 10000
 008AF A3120 15971 77A6E 55F31 011B0 20007 E5211
 9D711 81351 1CE61 0C6C6 F8F8B 05017 41431 33179
 11A14 51311 C4143 133"
 CHKR: #15AEAh
 "

17911 C1451 31142 16480 8C143 13317 91471 31017 07011
 110F0 01111 0F001 31007 20232 00270 13200 63013 20063
 02310 03602 31003 60033 00330 03300 33011 30071 03220
 04703 11007 40223 00170 07134 A0C49 016F6 6FFBE 4A0C4
 90163 66FF3 03A87 119CE 10987 05084 2303B E3A87 15E08
 2281E BB3A8 78329 51101 19D58 7073D 2309"
 CHKR: #24D9Dh
 "

8BD90 8526C 30315 18BA9 08526 C207A 60789 09765 08526
 81076 50871 A07C8 06700 74A0A 8BBB7 110CC 100A0 F568B
 E7110 D2304 CA100 A0F41 1111E 41011 60655 F111C
 CCC10 101BF 081CD 6C4CA 34048 FFCA1 31AF0 30EA8 A"
 CHKR: #15106h
 "

CD4C0 BF081 C64FF 01177 15F70 E7201 302A8 5BFC1
 7715F 70E72 15D7A 05DE 01302 A8517 715F7 0E7A1
 5D7A0 D5DE0 1DA32 200AB 7D68F 1DA10 0EF70 1D71F
 B110C 15B03 040EF 690C0 01323 20081 30801 1805C F3200
 4801C F96F2 E012E 3E076 C20"
 CHKR: #18679h
 "

CD570 07A20 A0000 FFFFF 35A80 09F20 8F3E0 76C20 73D20
 10C48 7670B BCB00 DF80C F9E07 3D201 0C471 AE002
 2A009 F2097 4E055 5E0CD 57007 A20A0 000FF FFF5D
 98073 D2010 C44B2 11EBB 80F68 80D4A E073D 2010E
 49C21 1EEE8 0CF9E 073D2 010E4 71AE0 022A0 555E0
 CD570 07A20 A0000 FFFFF 5D980 73D20 10E4D 13112
 3311E EE805 D980F 6880D 4AE02 E3E07 6C200 7A20A

0000F FFFF3 5A800 9F208 F3E07 6C20E 4A20F 00002 42514
 65F4D 7BB10 9F209 74E0A 2670E D211A 37A05 B9E00 9F20"
 CHKR: #69720h

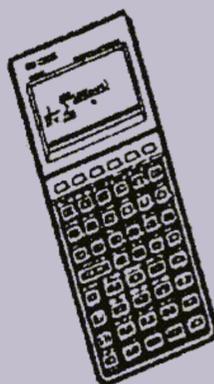
Accrocher les 12 chaînes les unes aux autres en appuyant 11 fois sur la touche +, puis dupliquez la chaîne totale, vérifiez-la avec CHKR, si le résultat est #159A4BAh, tout est o.k. et vous passez à la suite, sinon, recommencez l'étape V !!! Il ne vous reste plus maintenant qu'à exécuter ASSR et stocker le résultat (qui commence par 0 1 -> L N << 0 RDZ...) dans TRGST.

REMARQUES:

-- Le programme TRGST peut être utilisé comme tout programme ordinaire, mais si vous utilisez la commande VISIT dessus, il faudra terminer votre session par ON, au risque de le détruire.

-- Il est bon de ne pas purger ASSR et CHKR car vous n'aurez pas à refaire les étapes I à IV en rentrant un autre jeu.

SUR QUELLE MACHINE PROGRAMMEZ-VOUS ?



HP28 ?
 HP48 ?
 HP95 ?
 HP100 ?



Depuis plus de dix ans, notre association est reconnue dans le monde entier comme étant le club de référence chez les programmeurs et utilisateurs de calculateurs de poche Hewlett-Packard.

En plus des échanges d'informations et de programmes entre nos adhérents, nos activités comprennent l'édition d'une revue paraissant 10 fois par an, des réunions mensuelles, des contacts avec les clubs étrangers et bien d'autres choses encore...

Si vous êtes intéressé, n'hésitez pas à nous rejoindre !

PPC PARIS
B.P. 604 - 75028 PARIS CEDEX 01

Résolution numérique d'équations différentielles sur calculatrice HP42S

par Eric QUAGLIOZZI

La calculatrice HP42S, bien que possédant de nombreuses possibilités de calcul (notamment résolution d'équation non différentielles pour une variable quelconque et intégration numérique pour le calcul d'intégrales définies) est dépourvue de certains outils utiles aux scientifiques et mathématiciens tels que la dérivation numérique et la résolution numérique d'équations différentielles.

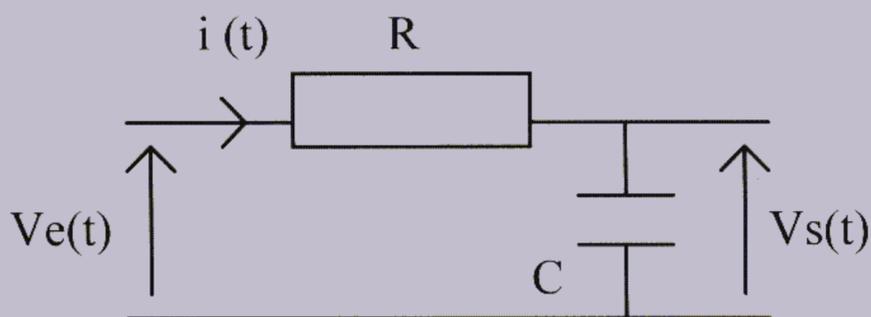
Pourquoi résoudre numériquement une équation différentielle?

1) Pour un certain nombre de types d'équations différentielles du premier ordre $\{Y' = f(t, Y)\}$ et du second ordre $\{Y'' = f(t, Y, Y')\}$ il existe des méthodes permettant de ramener le calcul des solutions à celui de primitives, parfois difficiles à calculer.

Mais au delà de l'ordre 2 $\{Y^p = f(t, Y, Y', \dots, Y^{p-1})\}$ la recherche d'une solution analytique devient très difficile, voire impossible.

2) En science, dans de nombreuses disciplines (mécanique, chimie, thermodynamique, électronique...) la loi d'évolution d'un système considéré en fonction d'une variable (par exemple le temps t) est souvent décrite par une équation différentielle. Donc pour connaître l'état de ce système pour une valeur particulière de la variable, il est nécessaire de résoudre l'équation différentielle (d'une façon ou d'une autre). Or il est très séduisant de pouvoir connaître l'état d'un système en s'affranchissant de la recherche analytique de son équation (en fonction des variables d'état).

Prenons comme exemple le circuit électrique suivant:



La loi d'évolution du système, en fonction du temps, est la suivante: $t v_s'(t) + v_s(t) = v_e(t)$ avec $t = RC$

C'est une équation différentielle du premier ordre que l'on peut encore écrire: $v_s'(t) = f(t, v_s(t))$

On voit immédiatement que la recherche de la solution $v_s(t)$ est "conditionnée" par l'expression du second membre $v_e(t)$. Il faudra calculer une primitive différente chaque fois que $v_e(t)$ aura une expression différente, ce calcul peut de plus devenir un peu "lourd".

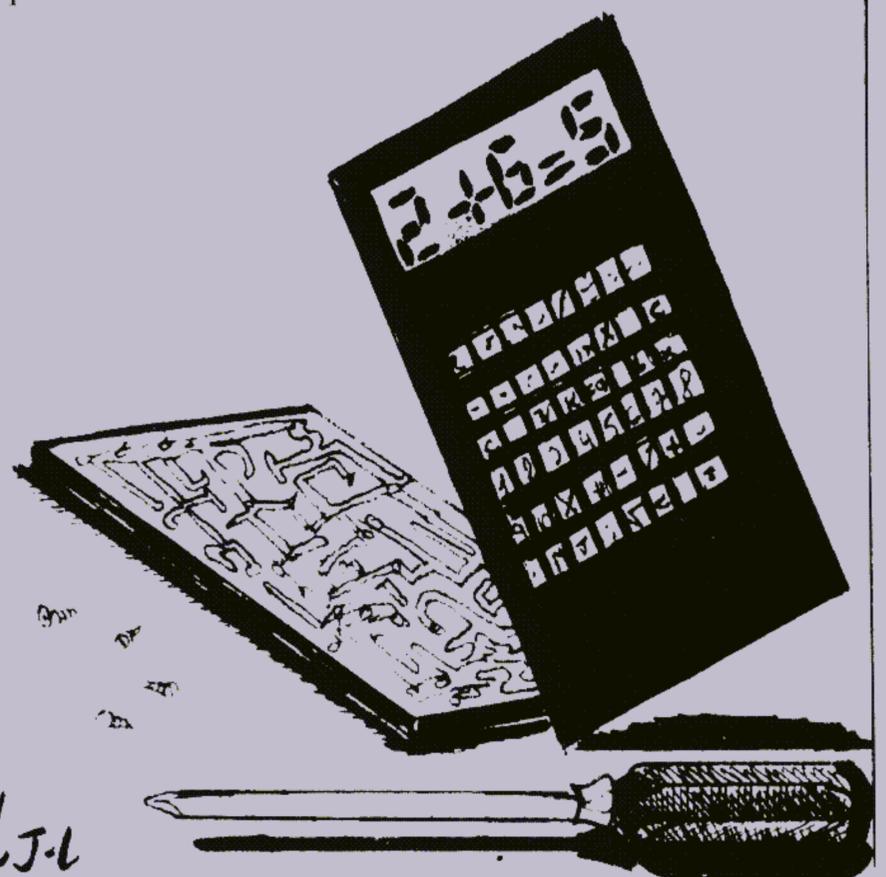
Ainsi, si l'on veut connaître la valeur de $v_s(t)$ à $t = 1s$ connaissant les conditions initiales $v_s(t = t_0)$ à t_0 pour $v_e(t) = U$ (une constante) puis $v_e(t) = A \sin(\omega t)$ puis ... il faudra choisir un bon crayon et se réserver une nuit tranquille.

Bien entendu on peut avoir recours à des "outils" tel que la Transformée de Laplace mais en sachant que cette application "n'opère" que sur une classe de fonctions telle que l'intégrale $\int_0^{+\infty} e^{-pt} \cdot f(t) \cdot dt$ converge.

De plus, la recherche de la solution nécessite quand même un nouveau calcul chaque fois que l'on change le second membre $v_e(t)$.

3) Enfin, un point non négligeable est que la résolution numérique d'une équation différentielle n'est pas forcément redondante mais peut être utilisée à des fins de vérification. Quantifier (à une erreur près liée à la méthode de résolution numérique) la solution pour une valeur de la variable (t par exemple) peut permettre de qualifier la solution analytique $Y(t)$ trouvée par intégration.

Voici donc exposé les "intérêts" de la résolution numérique d'équations différentielles.





RESOLUTION NUMERIQUE D' EQUATIONS DIFFERENTIELLES SUR HP42S

Nous allons à présent exposer quelques principes fondamentaux de résolution numérique d'équation différentielles d'ordre 1 à un pas (ce terme "un pas" sera explicité par la suite).

Discrétisation de la variable t :

Par une méthode numérique nous ne pouvons pas trouver une valeur de $Y(t)$ pour chaque valeur de t , cela impliquerait un nombre infini d'évaluations de $Y(t)$.

En effet, résoudre une équation différentielle d'ordre 1 (ceci est bien entendu valable pour tout ordre supérieur) c'est rechercher un fonction Y dérivable sur un intervalle $I \subset \mathbb{R}$ telle que $Y'(t) = f(t, Y(t))$. Donc puisque Y est dérivable alors elle est continue (attention la réciproque n'est pas forcément vraie!) sur I , puisque t peut prendre toutes les valeurs dans l'intervalle $I \subset \mathbb{R}$, c'est à dire une infinité, alors $Y(t)$ peut également avoir une infinité de valeurs.

On va donc discrétiser la variable t , c'est à dire subdiviser l'intervalle $I = [a, b]$ et construire ainsi des temps discrets (valeur particulières) de la sorte:

$$\begin{cases} t_0 = a \\ t_1 = a + h \\ t_2 = a + 2h \\ \dots \\ t_n = t_0 + nh \\ \dots \\ t_N = b \end{cases} \quad \text{où } h \text{ est appelé le PAS}$$

On va donc approcher la solution $Y(t_n)$ par pas successifs en partant d'une valeur initiale $Y(t_0)$ connue.

Par commodité on note

$$\begin{cases} Y(t_n) = Y_n \\ Y(t_n + h) = Y_{n+1} \end{cases}$$

Méthode d'EULER

Nous avons à résoudre

$$\begin{cases} Y'(t) = f(t, Y(t)) \\ Y(t_0) = Y_0 \text{ connue} \end{cases} \quad \text{(condition initiale)}$$

Le problème est d'évaluer la valeur de $Y(t)$ ne connaissant ni l'expression analytique de $Y(t)$, ni celle de $Y'(t)$ mais uniquement la relation les liants.

D'après le théorème des accroissement finis, puisque $Y(t)$ est continue sur $[a, b]$ et dérivable sur $]a, b[$ alors on peut écrire:

$$Y(t_n + h) = Y(t_n) + h Y'(t_n + \theta h) \text{ avec } 0 < \theta < 1$$

Soit encore:

$$Y_{n+1} = Y_n + h Y'(t_n + \theta h) \quad (1)$$

Se posent alors deux problèmes:

- 1/ on ne connaît pas $Y'(t)$, donc $Y'(t_n + \theta h)$
- 2/ $t_n < t_n + \theta h < t_{n+1}$, donc $t_n + \theta h$ ne fait pas partie de la suite des temps discrets $t_0, t_1, \dots, t_{n+1} + nh$ avec n entier.

La méthode d'EULER consiste donc à considérer que :

- 1/ puisque il faut résoudre $Y'(t) = f(t, Y(t))$ on peut en première approche remplacer $Y'(t)$ par $f(t, Y(t))$ l'équation (1).
- 2/ puisque $t_n < t_n + \theta h < t_{n+1}$, on peut choisir la valeur tn à la place de $t_n + \theta h$ sachant que l'on commettra une erreur plus ou moins grande.

Ceci donne donc la relation par récurrence suivante:

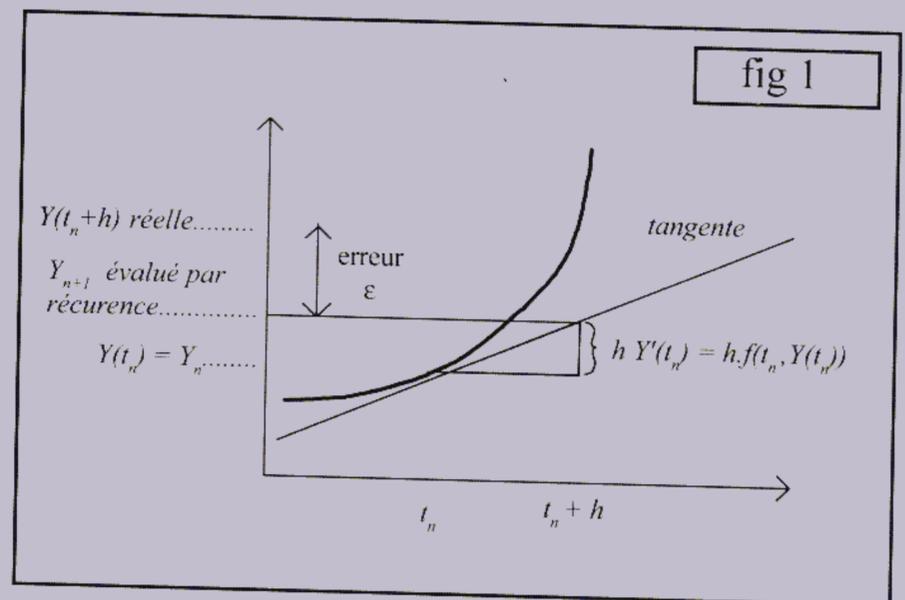
$$Y_{n+1} = Y_n + h f(t_n, Y_n)$$

La méthode d'EULER consiste donc à évaluer la valeur Y_{n+1} grâce à la valeur Y_n et la tangente à la courbe en t_n (figure 1). Ceci est une méthode à 1 pas car l'évaluation de Y_{n+1} ne nécessite que la connaissance de Y_n .

Une méthode multipas est de la forme:

$$Y_{n+1} = g(t_n, t_{n-1}, \dots, h, Y_n, Y_{n-1}, \dots)$$

La méthode d'EULER est en générale relativement peu performante, on lui préfère d'autre méthodes.



D'une façon générale la méthode consiste à trouver une fonction ϕ telle que

$$\begin{cases} Y_{n+1} = Y_n + h \phi(t_n, Y_n, h) \\ Y_0 = \alpha \text{ connue (condition initiale)} \end{cases}$$

"marche" le mieux possible. Il existe un certain nombre de méthodes telles que La Méthode du Point Milieu, la méthode de HEUN, les méthodes de RUNGE-KUTTA qui sont plus performantes.

Méthode de RUNGE-KUTTA

On recherche toujours à résoudre le problème :

$$Y'(t) = f(t, Y(t))$$

$$Y(t_0) = Y_0 \text{ connue}$$

Rechercher la solution $Y(t)$ de cette équation différentielle revient à intégrer $Y'(t)$, sachant que $Y'(t) = f(t, Y(t))$

$$\text{soit } Y(t) = Y_0 + \int_{t_0}^t f(t, Y(t)) dt$$

on vérifie bien que :

1/ $Y(t_0) = Y_0$ car l'intégrale vaut zéro. (puisque les bornes sont identiques et valent t_0)

2/ $Y'(t) = f(t, Y(t))$ puisque $\frac{dY}{dt} = f(t, Y(t))$ (car Y_0 est une valeur constante connue)

on peut donc écrire :

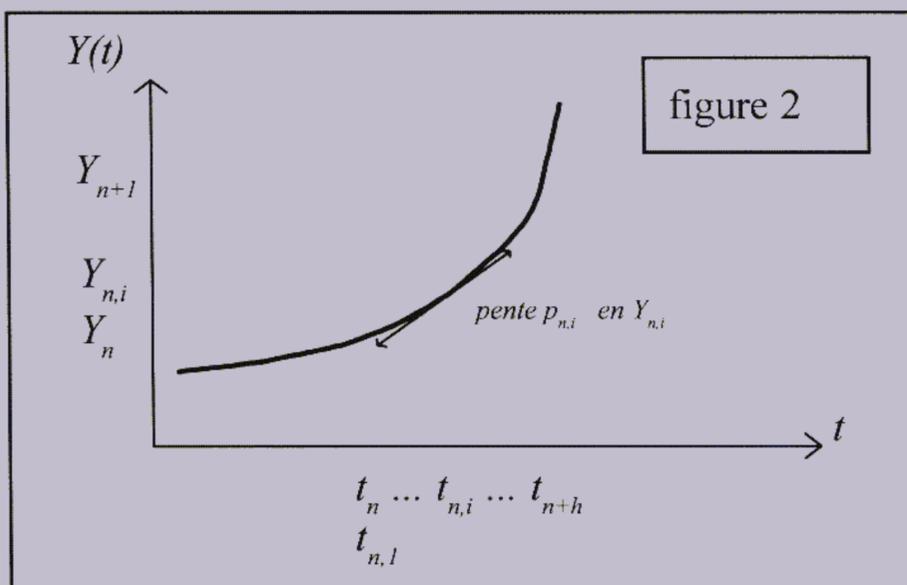
$$Y_{n+1} = Y_n + \int_{t_n}^{t_n+h} f(t, Y(t)) dt$$

par le changement de variable $t = t_n + u.h \Leftrightarrow dt = du$ cela donne :

$$Y_{n+1} = Y_n + h \int_0^1 f(t_n + uh, Y(t_n + uh)) du$$

L'idée est de subdiviser l'intervalle $[t_n, t_n+h]$ en un nombre de sous-intervalles q et de considérer que l'on connaît les valeurs de $Y(t)$ en ces points, soit $Y_{n,1}; Y_{n,2}; \dots; Y_{n,q}$ (figure 2). On peut donc appliquer une méthode numérique d'intégration (rectangle, trapèze, Simpson,...) et écrire :

$$Y_{n+1} = Y_n + h \sum_{j=1}^q b_j f(t_{n,j}, Y(t_{n,j}))$$



mais puisque on ne connaît pas les $Y_{n,i}$ la méthode consiste à les déterminer à nouveau par intégration :

$$Y_{n,i} = Y_n + h \int_0^{\alpha_i} f(t_n + uh, Y(t_n + uh)) du$$

que l'on va "approcher" aussi par une méthode d'intégration numérique, soit :

$$Y_{n,i} = Y_n + h \sum_{j < i} a_{ij} f(t_{n,j}, Y(t_{n,j})) \text{ pour } 1 \leq i \leq q$$

attention, on remarque que $j < i$ donc $i > 1$ et de plus une évaluation d'un $Y_{n,i}$ ne fait intervenir que les valeurs des $Y_{n,i-1}; Y_{n,i-2}; \dots; Y_{n,1}$ supposés déterminés auparavant avec comme valeur connue pour l'indice le plus petit :

$$\begin{aligned} Y_{n,1} &= Y_n \\ t_{n,1} &= t_n \\ p_{n,1} &= f(t_{n,1}, Y(t_{n,1})) = f(t_n, Y_n) \end{aligned}$$

où $p_{n,i}$ "représente" la valeur estimée de $Y'(t)$ pour $t = t_{n,i}$. La méthode consiste donc à associer à chacun des points intermédiaires $t_{n,i}$ dans l'intervalle $[t_n, t_n+h]$, la pente $p_{n,i}$ et de déterminer Y_{n+1} par une méthode d'intégration.

Méthode Runge-Kutta "classique"

Elle est définie selon le principe exposé précédemment avec $q = 4$. L'algorithme est le suivant :

$$p_{n,1} = f(t_n, Y_n)$$

$$t_{n,2} = t_n + 1/2h$$

$$Y_{n,2} = Y_n + 1/2h p_{n,1}$$

$$p_{n,2} = f(t_{n,2}, Y_{n,2})$$

$$Y_{n,3} = Y_n + 1/2h p_{n,2}$$

$$p_{n,3} = f(t_{n,2}, Y_{n,3})$$

$$t_{n+1} = t_n + h$$

$$Y_{n,4} = Y_n + h p_{n,3}$$

$$p_{n,4} = f(t_{n+1}, Y_{n,4})$$

$$Y_{n+1} = Y_n + h/6 (p_{n,1} + 2p_{n,2} + 3p_{n,3} + p_{n,4})$$

— On débute l'algorithme avec $Y(t_0) = Y_0$ connue qui est la condition initiale.

— Le pas h ne doit pas être trop grand car cela entraînerait une erreur trop grande liée à la méthode (ne pas oublier qu'une tangente à une courbe peut être utilisée localement en un point pour se substituer à celle-ci mais localement seulement!) mais le pas h ne doit pas être non plus trop petit car se sont les erreurs d'arrondi qui se cumuleraient au point de "fausser" notablement le résultat.

— Un pas h de 10^{-2} à 10^{-3} est suffisant généralement.

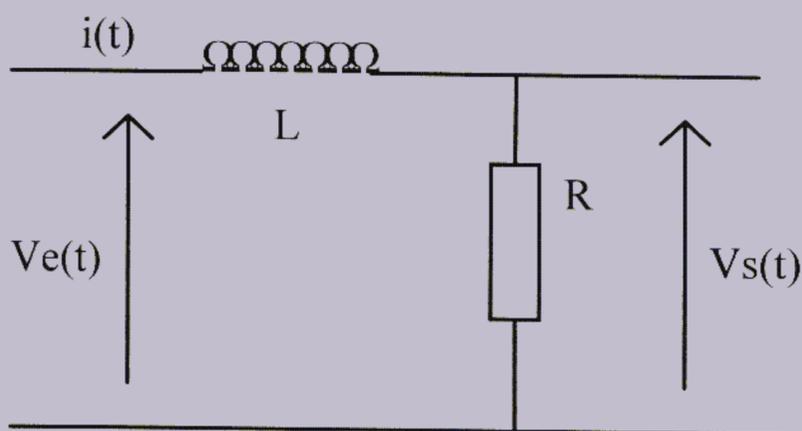


RESOLUTION NUMERIQUE D' EQUATIONS DIFFERENTIELLES SUR HP42S

Exemple d'utilisation du programme

Soit le circuit électrique suivant:

$$\tau \frac{dV_s}{dt} + V_s = V_e \quad \text{avec} \quad \tau = \frac{L}{R}$$



$L = 0,1 H$ et $R = 1 \Omega$

Supposons comme condition initiale $V_s(0) = 0 V$. On applique une tension sinusoïdale $V_e(t) = \sin(63t)$. Que vaut $V_s(t)$ à $t = 0,2 s$?

Etape 1

il faut entrer la fonction $V_e(t)$ dans la machine en considérant la variable t au premier niveau de la pile:

```
01 LBL "VE(T)
01 63
03 X
04 SIN
05 RTN
06 END
```

Vous devez lire à la ligne 00: 00 {15-Byte Prgm}

Etape 2

— Lancer le programme "RK43 de la méthode de Runge-Kutta 4

— Un menu vous propose :

Y_n T_n T_{max} H A B pour $AV'_s + BV_s = V_e$

— Entrez: $V_{s0} = Y_0$ dans Y_n , c'est à dire ici la val.0
 $t_0 = 0$ dans T_n , temps initial.
 $t = 0,2s$ dans T_{max} , temps final.
 $5 \cdot 10^{-3}$ dans H (représente le pas ce qui fait $0,2 / 5 \cdot 10^{-3} = 40$ itérations).
 $\tau = L/R = 0,1s$ dans A (coefficient devant V'_s)
 1 dans B (coefficient devant V_s)

— Appuyer sur R/S et patientezune minute.

— Vous lisez $Y(0,2s) = -0,1329619...V$

— La solution exacte étant :

$$V_s(t) = \frac{\tau\omega}{1 + (\tau\omega)^2} (e^{-t/\tau} + \frac{\sin\omega t - \cos\omega t}{\tau\omega}) + V_s e^{-t/\tau}$$

— La valeur exacte est donc $V_s(0,2s) = -0,13268614...v$

— Ce qui fait environ $-4 \cdot 10^{-7}$ d'erreur absolue.

— En changeant la condition initiale $V_s = 0,5V$ dans Y_n et en recommençant l'opération précédente on trouve numériquement

$Y(0,2s) = -0,0652942V$ au lieu de

$V_s(0,2s) = -0,0652938V$ ce qui fait une erreur de $-4 \cdot 10^{-7}$ environ.

Supposons maintenant que nous voulons connaître la valeur de $V_s(0,2s)$ mais avec comme tension d'entrée une rampe: $V_e(t) = 2t$.

Nous avons donc l'équation : $\tau \frac{dV_s}{dt} + V_s = 2t$

La solution de cette équation est :

$$V_s(t) = 2(t - \tau) + (V_{s0} + 2\tau) e^{-t/\tau}$$

Voici un tableau comparatif des résultats obtenus :

Condition Initiale	Pas h	Y(0,2) numérique	Vs(0,2) réelle	Erreur Absolue
0V	5.10-3	0,227067	0,227067	3.10-9
		0595V	0566V	
0V	2.10-3	0,227067	0,227067	1,2.10-10
		0567V	0566V	
1V	5.10-3	0,3624	0,3624	1,7.10-8
		02257V	02339V	
1V	2.10-3	0,3624	0,3624	4.10-10
		0234V	02339V	

Voici le programme de la rampe $V_e(t) = 2t$

```
00 {13 _ Byte Prgm}
01 LBL "VE(T)"
02 2
03 X
04 RTN
05 END
```

Voyez comme c'est simple à utiliser, on passe d'une sinusoïde à une rampe en quelques opérations.

Conclusion:

La méthode de Runge-Kutta quoique suffisante pour évaluer numériquement $Y(t)$ montre dans cet exemple qu'il y a mieux, notamment des méthodes multipas (P.E.C.E = Prédiction - Evaluation - Correction - Evaluation)

l'équation différentielle elle-même (coefficients et second membre) intervient dans l'erreur commise à chaque évaluation de Y_{n+1} sans compter les erreurs d'arrondi.

Il faut donc utiliser cette technique de résolution numérique



RESOLUTION NUMERIQUE D' EQUATIONS DIFFERENTIELLES SUR HP42S

d'équations différentielles en connaissance de ses limitations.

On notera qu'un pas de l'ordre de 10^{-3} semble convenir raisonnablement compte tenu des erreurs commises dans nos exemples et des valeurs de Tmini et Tmax.

Bibliographie:

. Analyse Numérique et Equations Différentielles

de Jean-Pierre DEMAILLY édition PUG (collection Grenoble Sciences)

. Initiation à l'Analyse Numérique de Robert THEODOR édition MASSON

Programme RK4

(Résolution numérique d'équations différentielles du 1er ordre)

- 00 {246 - byte Prgm}
- 01 LBL "RK4"
- 02 MVAR "YN"
- 03 MVAR "TN"
- 04 MVAR "TMAX"
- 05 MVAR "H"
- 06 MVAR "A"
- 07 MVAR "B"
- 08 VARMENU "RK4"
- 09 STOP
- 10 RCL "TMAX"
- 11 RCL - "TN"
- 12 RCL , "H"

- 13 IP
- 14 1 E-5
- 15 +
- 16 STO 04
- 17 RCL "B"
- 18 RCL , "A"
- 19 STO 00
- 20 LBL 01
- 21 RCL "TN"
- 22 RCL "YN"
- 23 XEQ 02
- 24 STO 01
- 26 RCL ' "H"
- 27 RCL + "TN"
- 28 X <> Y
- 29 RCL ' "H"
- 30 0.5
- 31 ' "
- 32 RCL + "YN"
- 33 XEQ 02
- 34 STO 02
- 35 RCL ' "H"
- 36 0.5
- 37 ' "
- 38 RCL + "YN"
- 39 0.5
- 40 RCL ' "H"
- 41 RCL + "TN"
- 42 X <> Y
- 43 XEQ 02
- 44 STO 03
- 45 RCL ' "H"
- 46 RCL + "YN"
- 47 RCL "TN"
- 48 RCL ' "H"
- 49 STO "TN"

- 50 X <> Y
- 51 XEQ 02
- 52 RCL + 01
- 54 ,
- 55 RCL 02
- 56 RCL + 03
- 57 3
- 58 ,
- 59 +
- 60 RCL ' "H"
- 61 RCL + "YN"
- 62 STO "YN"
- 63 DSE 04
- 64 GTO 01
- 65 "YN POUR TN ="
- 66 RCL "TMAX"
- 67 ENTER
- 68 AIP
- 69 | — " ."
- 70 FP
- 71 1E4
- 72 ' "
- 73 AIP
- 74 RCL "YN"
- 75 EXITALL
- 76 PROMPT
- 77 GOTO "RK4"
- 78 LBL 02
- 79 RCL ' 00
- 81 XEQ "VE(T)"
- 82 RCL , "A"
- 83 X <> Y
- 84 -
- 85 RTN
- 86 END

Société: Organisation Deb's		Rep Tr Mo Aj
Nom: Thierry Desbruères		
Pro: 47 45 20 44, fax: 46 24 76 81		
Ad. Pro: 139, avenue Charles de Gaulle 92200 Neuilly		
Notes: vous propose la prise en dépôt de votre Series 3 pour mise en vente sur le marché d'occasion à travers son réseau et à vos conditions contre l'achat d'un Series 3a.		
Trouver: deb's		462/462 1er

Anniversaires (en 1993-1994)				Age
Octobre	Novembre	Décembre	Janvier	Jo He Pi T à An
19 Poilot (1948, 53 ans) > Δ	18 Annie (1939, 54 ans) > Δ		1 Férié Nouvel An >	
23 Joëlle (1941, 52 ans) > Δ	18 Robert (1938, 63 ans) > Δ		1 Rentner cours Bourse du premier jour de l'année. > Δ	
			27 Anniversaire Louise (1943) (1943, 51 ans) > Δ	
				1er

Mes ruptures avec mon Série 3a sont, désormais, voulues.

Ven 5 Novemb	Détails d'alarme		Age Jo He Pi T à An
06	Alarme	Oui	
07	Délai avant	08:38	
08	Alarme à	12:38	
09	Nbre de jours avant	0	
10	Sonorité	+ Fanfare →	
11	Tester sonorité	Confirmer	
12		Menu	Entrée
e			1er

Faites sonner les clairons, il est arrivé; qui ?, le 3a

Anniversaires (en 1993-1994)				Age
Octobre	Novembre	Décembre	Janvier	Jo He Pi T à An
19 Poilot (1948, 53 ans) > Δ	18 Annie (1939, 54 ans) > Δ		1 Férié Nouvel An >	
23 Joëlle (1941, 52 ans) > Δ	18 Robert (1938, 63 ans) > Δ		1 Rentner cours Bourse du premier jour de l'année. > Δ	
			27 Anniversaire Louise (1943) (1943, 51 ans) > Δ	
				1er

Mes ruptures avec mon Série 3a sont, désormais, voulues

Tous les produits PSION sont en stock chez DEB'S (logiciels, câbles, mémoires et divers) ; seuls les Séries 3a sont parfois sujets à délais.



CASIO PB-2000

Décomposition d'un nombre en ses facteurs

Par le Dr. Herbert A. Kellner

Ce petit programme permet la factorisation d'un nombre LONGINT. Celui-ci est écrit en DL-PASCAL du CASIO PB-2000 avec le module Pascal, mais avec les modifications indiquées pour PC, le programme peut tourner également en Turbo Pascal.

```

PROGRAM factorize; (*29/10/93*)
(* CASIO PB-2000C+module DL-
PASCAL*)
(* (c) 1993 Dr. Herbert A. Kellner *)

USES Crt;
(*Version for PC :
CONST
  a : ARRAY[0..7]
    OF BYTE=(4,2,4,2,4,6,2,6); *)

TYPE
  pointer=^element;
  element=RECORD
    contenu:BYTE;
    next : pointer;
  END;
  ar8 =ARRAY[0..7] OF BYTE;
VAR
  a:ar8=(4,2,4,2,4,6,2,6);
  racine,ptr,save: pointer;
  n,divi : LONGINT;
  i : BYTE;

FUNCTION read_int_chk:LONGINT;
VAR
  n1: LONGINT;
  str1,strn :STRING;
  err : BYTE; (*for PC: INTEGER!*)
BEGIN
  WRITE ('INT.POS Number=? : ');
  READLN(strn);
  VAL(strn,n1,err);
  n1:=ABS(n1);
  STR(n1,str1);
  IF (strn<>str1) OR (err<>0) THEN
    BEGIN
      WRITELN(#7,'STOP.INPUT
ERROR?');
      HALT;
    END;
  IF n1=0 THEN HALT;
  read_int_chk:=n1;
END;

```

```

PROCEDURE initialize_pointer;
VAR
  ind:BYTE;
BEGIN
  NEW(ptr);
  racine:=ptr;
  save:=ptr;
  FOR ind:=0 TO 7 DO
    BEGIN
      racine^.next:=ptr;
      racine:=ptr;
      racine^.contenu:=a[ind];
      NEW(ptr);
    END;
  racine^.next:=save;
  ptr:=save;
END;P

PROCEDURE divprnt;
VAR
  expo:BYTE;
BEGIN
  expo:=0;
  WHILE n MOD divi=0 DO
    BEGIN
      n:=n DIV divi;
      expo:=SUCC(expo);

```

```

    END;
  IF expo<>0 THEN
    IF expo<>1 THEN
      WRITE('*',divi,'exp(',expo,')')
    ELSE WRITE('*',divi);
  END;
  BEGIN (*main_program*)
  REPEAT
    n:=read_int_chk;
    initialize_pointer;
    WRITE('1');
    FOR i:=0 TO 3 DO
      BEGIN
        divi := i SHL 1 + 1;
        IF i=0 THEN divi:=SUCC(divi);
        divprnt;
      END;
    REPEAT
      divi:=divi+ptr^.contenu;
      ptr:=ptr^.next;
      divprnt;
    UNTIL n DIV divi<divi;
    IF n<>1 THEN WRITE('*',n);
    WRITELN;
  UNTIL FALSE;
END.

```





PROGRAMMES POUR CASIO PB1000

par Daniel PRATLONG

GRAPHES SUR CASIO PB1000

Cet ensemble de programmes vous permettra de tracer des courbes en deux dimensions : en coordonnées cartésiennes, paramétriques, ou polaires. Les trois premiers programmes (Graph, Param, Polar) sont indépendants. Par contre, les 4 autres programmes sont nécessaires aux trois premiers. Vous devez respecter l'orthographe exacte des noms de chaque programme (en respectant les minuscules/majuscules dans l'écriture). Pour mieux comprendre le problème regardez la ligne 50 du programme Graph. Avant de lancer un programme vous devez l'éditer pour entrer votre fonction sur la deuxième ligne. Ensuite, le fonctionnement est assez proche d'une CASIO Graphique. Le scale (scl) est l'écart entre deux graduations sur le repère. Si vous ne donnez pas de valeur, la valeur par défaut sera 1 (une unité par graduation). Pour les fonctions trigonométriques, il est intéressant de donner $scl = \pi/2$ ou $\pi/4$. Si vous tapez A au lieu de donner une valeur pour Y min, le repère créé sera orthonormé. Dans ce genre de repères, un cercle est un cercle (sinon c'est une ellipse !). Le paramètre speed est l'écart entre deux points réellement calculés de la courbe. Pour $speed=1$ tous les points sont calculés, mais le traçage, est plus lent. Si vous ne donnez pas de valeur pour speed, il vaudra 4 par défaut. Dans l'écran suivant, il vous est demandé, si vous voulez superposer une image sur votre courbe. Si tel est le cas, remplacez le N par le nom du fichier-image. Si vous voulez sauver votre courbe après le traçage, procédez de la même façon.

Graph :

```
10 GOTO50
20 Y=SINX
30 RETURN
40 K=0:RESUME120
50 GOSUB»Range»:GOSUB»Superpo»: GOSUB»Repere
70 ONERRORGOTO40
80 FORX=XI TOXM STEPc
90 GOSUB20:A=X*U+a:B=-Y*V+b
100 IFK=0THENC=A:D=B:K=1
110 DRAW(C,D)-(A,B):C=A:D=B
120 NEXT
130 GOSUB»Fin»:GOTO50
```

Param :

```
10 GOTO60
20 X=COS(T+T)
30 Y=SINT
40 RETURN
50 K=0:RESUME170
60 GOSUB»Range
70 CLS:PRINT»T min =>»Ti
80 PRINT»T max =>»Tm
90 LOCATE7,0:INPUT»»,Ti
100 LOCATE7,1:INPUT»»,Tm:IFTi>=Tm THEN90
110 GOSUB»Superpo»:GOSUB»Repere
120 ONERRORGOTO50
130 FORT=Ti TOTm STEPc
140 GOSUB20:A=X*U+a:B=-Y*V+b
150 IFK=0THENC=A:D=B:K=1
160 DRAW(C,D)-(A,B):C=A:D=B
170 NEXT
180 GOSUB»Fin»:GOTO60
```

Polar :

```
10 GOTO50
20 R=H
30 RETURN
40 K=0:RESUME160
50 GOSUB»Range
60 CLS:PRINT»H min =>»Hi
70 PRINT»H max =>»Hm
80 LOCATE7,0:INPUT»»,Hi
90 LOCATE7,1:INPUT»»,Hm:IFHi>=Hm THEN80
100 GOSUB»Superpo»: GOSUB»Repere
110 ONERRORGOTO40
120 FORH=Hi TOHm STEPc
130 GOSUB20: X=R*COSH: Y=R*SINH: A= X*U+a:B=-Y*V+b
140 IFK=0THENC=A:D=B:K=1
150 DRAW(C,D)-(A,B):C=A:D=B
160 NEXT
170 GOSUB»Fin»:GOTO50
```

Range :

```
10 CLS:PRINT»Range
20 ONERRORGOTO130
30 PRINT»X min :>»XI,» max :>»XM,» scl :>»SX;
40 LOCATE7,1:INPUT»»,XI: LOCATE7,2: INPUT»»,XM: IFXM<=XI THEN40
50 LOCATE7,3:INPUT»»,SX: IFSX<=0THENSX=1
60 CLS:PRINT»Y min :>»YI,» max :>»YM,» scl :>»SY,»speed :>»P
70 LOCATE7,0:INPUT»»,YI $: IFYI$=>»A»THENw=1: YI=-1E20
ELSEYI=VAL(YI$):w=0
80 LOCATE7,1:INPUT»»,YM: IFYM<=YI THEN70
90 LOCATE7,2:INPUT»»,SY: IFSY<=0THENSY=1
100 LOCATE7,3:INPUT»»,P: IFP<=0THENP=4
110 IFLE=0THENLE=150
120 RETURN
130 CLS:INPUT»Error...»,AS:RESUME10
```

Repere :

```
10 GOTO60
20 RESUME110
30 RESUME130
40 RESUME160
50 RESUME180
60 UH=SX*LE/(XM-XI):UV=SY*31/(YM-YI): a=-XI/SX*UH:b=YM/SY*UV:
IFw=1THENUV=UH:YI=YM-SY*31/UV:b=YM/SY*UV
70 IFa=>0ANDa<LE THENDRAW(a,0)-(a,63)
80 IFb=>0 AND b<63 THEN DRAW(0,b)-(LE,b)
90 O=b-1: IFO<0 OR O>63 THEN140
100 ONERRORGOTO20:FORg=a TOLE STEPc:UH:DRAW(g,O)
110 NEXT
120 ONERRORGOTO30:FORg=a TO0STEP-UH:DRAW(g,O)
130 NEXT
140 O=a+1:IFO<0ORO>LE THEN190
150 ONERRORGOTO40:FORg=b TO0STEP-UV:DRAW(O,g)
160 NEXT
170 ONERRORGOTO50:FORg=b TO63STEPUV:DRAW(O,g)
180 NEXT
190 ANGLE1:K=0:U=LE/(XM-XI):V=31/(YM-YI):e=P/U:RETURN
```

Superpo :

```
10 CLS:IFSS$=>»» THENSS$=>»N»
20 IFP$=>»» THENP$=>»N»
30 PRINT»Sauver : «SS,»Superposition : «P$
40 LOCATE9,0:INPUT»»,SS:LOCATE16,1: INPUT»»,P$
50 IFP$=>»N» THENCLSELSECL: BLOADP$,25089
60 RETURN
```

Fin :

```
10 IFSS<>»N» THENBSAVES$,25089,1536
20 LOCATE30,0:INPUTQ$
30 LOCATE23,6:PRINT»X'X... 2":LOCATE23,5:PRINT»Range: 1 40
A=VAL(INKEY$):ONA GOTO60,50:GOTO40
50 CLS:PRINT»X'X=>»:LE: LOCATE4,0:
INPUT»»,LE:IFLE<=0ORLE>191THEN50ELSE30
60 RETURN
```



PROGRAMMES POUR CASIO PB1000... PROGRAMMES POUR CASIO PB1000...

CALCUL MATRICIEL SUR PB1000

Ce programme permet de faire pratiquement tous les calculs possibles et imaginables sur les matrices et les vecteurs (addition, multiplication, puissance, déterminant, inversion, valeurs propres, systèmes linéaires...). Il est configuré, pour manipuler des matrices de dimension maximale 6x6, vous pouvez modifier ce paramètre en début de programme. Pour cela, vous devez modifier la ligne 5 et la ligne 10 (si vous ne modifiez pas la ligne 5, toutes vos données seront effacées à chaque lancement du programme). Dans ces deux lignes, m = dimension maximum. Attention ! Pour des grandes matrices il vous faut de la mémoire ! Et il faut que votre machine soit bien configurée ! Consultez le manuel si vous avez des doutes sur le fonctionnement de la fonction CLEAR. Ma PB1000 est configurée de la façon suivante : CLEAR 800,200,8000

L'utilisation du programme est assez intuitive. Certains menus utilisent l'écran sensitif, d'autres le clavier. Certaines fonctions sont assez gourmandes en temps de calcul... soyez patients ! Un détail : lors de la saisie d'une matrice (fonction [IN A] par exemple) il vous est demandé, un N° de colonne. Si vous tapez T, vous saisissez toutes les colonnes. Si vous tapez Z, toutes les colonnes seront initialisées à zéro. Enfin, si vous tapez 0, ou rien du tout, vous revenez au menu. Autre chose : lors du calcul des valeurs propres, il vous est demandé, deux valeurs d'initialisation de P et Q. Les valeurs que vous donnez n'auront aucune influence sur le résultat final. en général, lancez le calcul avec P=0 et Q=0. Si cela ne marche pas, essayez autre chose ! (P=1 Q=2, ...) Ce programme a été testé et utilisé, très souvent en situation réelle (examen). Je pense que vous pouvez lui faire confiance. Néanmoins, il est assez long à taper. Relisez-vous plusieurs fois, et évitez d'apporter des modifications. Daniel PRATLONG.

```
5 IFm=6THEN100
10 CLEAR:m=6:DIMA(m,m), B(m,m), C(m,m),D(m,2*m),VE(3,m)
30 DIMd(4,2),M$(3),DV(3), VV$(3): DEFCHR$(240)=»1 0101040F000":
DEFCHR$(241)=»FC304884103E»: DEFCHR$(242)=»FC3048842E3A»:
DEFCHR$(243)=»F84080807800
40 M$(1)=»A»:M$(2)=»B»: M$(3)=»C »: KW$=»W=»+CHR$(241)+»U+»+
CHR$(242)+»V»:KA$=CHR$(241)+»A+»+CHR$(242)+»B»: DIC$=»Dimensions
incompatibles !
50 VV$(1)=»U»:VV$(2)=»V»: VV$(3)=»W»: p=1E-7:PADEF$= »Vecteur/matrice
pas defini(e) !»
100 z=0:CLS: PRINTREV»MATRICES ET VECTEURS»NORM.»1_Manip. vect. |
4_Calc. vect. 2_Manip. matricl 5_Calc. matric 3_Calc. V=A.U | 6_System A.x=U»:
190 E=VAL(INKEY$):ONE GOTO1000,2000,4000 ,3000, 3500, 8000:GOTO190
499 'Det(A)
500 DR=1:RG=N
510 FORK=1TON
520 I=K
530 IFI>N THENDR=0:RG=RG-1:GOTO610
540 IFABS(D(I,K))<p THENI=I+1:DR=-DR:GOTO530
550 IFI>K THENFORJ=1TON+P: U=D(I,J):D(I,J)= D(K,J):D(K,J)=U: NEXT
560 FORI=K+1TON+P: D(K,I)=D(K,I)/D(K,K):NEXT
570 DR=DR*D(K,K):D(K,K)=1
580 FORI=1TON
590 IFI<>K THENFORJ=K+1TON+P:D(I,J)=D(I,J)-D(I,K)*D(K,J): NEXT:D(I,K)=0
600 NEXT
610 NEXTK:RETURN
900 IFL=1THENU=A(I,J)
910 IFL=2THENU=B(I,J)
920 IFL=3THENU=C(I,J)
930 IFL=4THENU=D(I,J)
940 RETURN
950 IFM=1THENA(I,J)=U
960 IFM=2THENB(I,J)=U
970 IFM=3THENC(I,J)=U
980 IFM=4THEND(I,J)=U
990 RETURN
1000 CLS: PRINTREV»VECTEURS»TAB(24): NORM.»[Retour][ IN U ][ IN V ][ IN
W ][OUT U ][OUT V ][OUT W ][ V=U ][ U=V ][ W=U ][ U=W ][ W=V ][ V=W
]»CHR$(11)
1150 E=ASC(INKEY$)-243
1155 IFE=0THEN100
1160 ONINT((E-1)/3+1) GOSUB1300,1500,1600,1600: IFE>0THEN1000 ELSE
1160
1300 NV=E:CLS:PRINT»Entree de «VV$(NV):INPUT»Dimension : «,E 1330
IFE<10RE>m THEN1420
1340 DV(NV)=E: FORI=1TODV(NV)
1370 PRINTVV$(NV)»(«I»: INPUT)»»,NN
1400 VE(NV,I)=NN:NEXT
1420 RETURN
1500 CLS:NV=E-3
1520 IFDV(NV)=0THENPRINTPADEF$:GOTO1590
1530 FORI=1TODV(NV):PRINTVV$(NV)» («I»)»VE(NV,I);
1580 GOSUB9040:NEXT
1590 GOSUB9040:RETURN
1600 IFE=7THENVD=1:VA=2
1610 IFE=8THENVD=2:VA=1
1620 IFE=9THENVD=1:VA=3
1630 IFE=10THENVD=3:VA=1
1640 IFE=11THENVD=2:VA=3
1650 IFE=12THENVD=3:VA=2
1660 DV(VA)=DV(VD) : FORI=1TODV(VD):VE(VA,I)=VE(VD,I);
NEXT:E=3+VA:GOSUB1500:RETURN
2000 CLS:PRINTREV» MATRICES» TAB(24):NORM.»[Retour][ IN A ][ IN B ][OUT
A ][OUT B ][OUT C ][ C:=A ][ A:=C ][ C:=B ][ B:=C ][ B:=A ][ A:=B
][[Transp]»CHR$(11)
2150 E=ASC(INKEY$)-243:IFE=0THEN100
2170 IFE=12GOTO2890
2180 IFE>5GOTO2700
2190 IFE>2GOTO2500
2191 IFE>0GOTO2200
2195 GOTO2150
2200 CLS:PRINT»Entree de «;M$(E)
2210 INPUT»Nb lignes =»,NL: IFNL>m ORNL<0THEN2210
2230 INPUT»Nb colonnes =»,NC:IFNC>m ORNC<0THEN2230
2250 IFNC<>0GOTO2280
2260 IFNL<>0GOTO100
2270 d(E,1)=0:d(E,2)=0:GOTO2000
2280 IFNL=0 GOTO100
2290 d(E,1)=NL:d(E,2)=NC:M=E
2300 INPUT»N° COLONNE :»,Z$:IFZ$=»T»THENFORJ=1TONC:
GOSUB2400:NEXT:GOTO2300
2320 IFZ$=»Z»THENFORI=1 TONL: FORJ=1TONC:U=0:
GOSUB950:NEXT:NEXT:GOTO2300
2330 J=VAL(Z$):IFJ<=0GOTO2000
2340 IFJ<=NC THENGOSUB2400
2350 GOTO2300
2400 FORI=1TONL
2410 PRINTM$(E)»(«I»,»J»:INPUT)»»,NN
2440 U=NN:GOSUB950:NEXT:RETURN
2500 CLS:L=E-2:IFd(L,1)=0THENPRINTPADEF$: GOSUB9040:GOTO2000 2530
PRINT»Nb lignes:»d(L,1),»Nb colonnes:»d(L,2)
2550 INPUT»N° COLONNE:»,Z$
2560 IFZ$=»T»THENGOSUB2670:GOTO2550
2570 J=VAL(Z$):IFJ<=0GOTO2000
2580 IFJ<=d(L,2)THENGOSUB2600
2590 GOTO2550
2600 FORI=1TOD(L,1):GOSUB900: PRINTM$(L)»(«I»,»J»)»U;
2650 GOSUB9040:NEXT:RETURN
2670 FORJ=1TOD(L,2):GOSUB2600 :NEXT:RETURN
2700 IFE=6THENL=1:M=3
2710 IFE=7THENL=3:M=1
2720 IFE=8THENL=2:M=3
2730 IFE=9THENL=3:M=2
2740 IFE=10THENL=1:M=2
2750 IFE=11THENL=2:M=1
2760 CLS:PRINT»Transfert de «M$(L)» dans «M$(M)
```



PROGRAMMES POUR CASIO PB1000 ... PROGRAMMES POUR CASIO PB1000 ...

```
2770 GOSUB2810:GOTO2000
2810 d(M,1)=d(L,1):d(M,2)=d(L,2): FORI=1TOd(M,1):FORJ=1TOd(M,2):
GOSUB900:GOSUB 950:NEXT: NEXT:RETURN
2890 CLS:PRINTREV»TRANSPOSER»NORM
2900 INPUT»Matrice: «,Z$
2910 M=ASC(Z$)-64:IFM<1ORM>3THEN2900
2950 d(4,1)=d(M,2):d(4,2)=d(M,1): L=M:IFd(4,1)=0THEN2990
2970 FORJ=1TOd(4,1): FORI=1TOd(4,2): GOSUB900:D(J,I)=U: NEXT:NEXT
2980 L=4:GOSUB2810:GOTO2000
3000 CLS:PRINTREV»CALCUL VECTORIEL»NORM,»1_»KW$» | 3_Vectoriel
U^V2_Scalaire U.V | 4_Retour»;
3080 E=VAL(INKEY$):ONE GOSUB3100,3200,3300,100
3086 IFE=0THEN3080
3090 GOTO3000
3100 CLS:PRINTREVKW$;NORM
3110 IFDV(1)<>DV(2)THENPRINT DIC$:GOTO3190
3120 GOSUB9020
3140 DV(3)=DV(1)
3150 FORI=1TODV(1)
3160 VE(3,I)=K1*VE(1,I)+K2*VE(2,I)
3170 NEXT
3180 E=6:GOSUB1500
3190 RETURN
3200 CLS:PRINTREV»PRODUIT SCALAIRE»NORM
3210 IFDV(1)<>DV(2)THENPRINTDIC$:GOTO3290 3220 X=0
3230 FORI=1TODV(1):X=X+VE(1,I)*VE(2,I):NEXT 3260 PRINT»U.V = «X;
3290 GOSUB9040:RETURN
3300 CLS:PRINTREV»PRODUIT VECTORIEL»NORM
3310 IFDV(1)<>3ORDV(2)<>3 THENPRINTDIC$:GOTO3370
3320 DV(3)=3
3330 VE(3,1)=VE(1,2)*VE(2,3)-VE(2,2)*VE(1,3)
3340 VE(3,2)=VE(1,3)*VE(2,1)-VE(2,3)*VE(1,1)
3350 VE(3,3)=VE(1,1)*VE(2,2)-VE(2,1)*VE(1,2)
3360 E=6:GOSUB1500
3370 RETURN
3500 CLS:PRINTREV»CALCUL MATRICIEL»NORM;TAB(24)»[Retour]
3510 PRINT»[«KA$»][ A.B ][ A»CHR$(243)» ]»,«[Det(A) ][ A»CHR$(240)» ]
[v.prop]»;
3570 E=ASC(INKEY$)-242
3580 ONE GOTO100,4500,5000, 5500 ,100,4500, 5000,5500,100, 6000,6500,7000
3590 GOTO3570
4000 CLS:PRINTREV»V=A.U »NORM
4020 GOSUB9010:IFz THENz=0: GOTO4140
4030 IFDV(1)=0THEN PRINTPADEF$: GOTO4140
4040 IFd(1,2)<>DV(1)THEN PRINTDIC$: GOTO4140
4050 DV(2)=d(1,1)
4060 FORI=1TODV(2)
4070 VE(2,I)=0
4080 FORJ=1TODV(1)
4090 VE(2,I)=VE(2,I)+ A(I,J)*VE(1,J)
4100 NEXT
4110 NEXT
4120 E=5
4130 GOSUB1500:GOTO100
4140 GOSUB9040:GOTO100
4500 CLS:PRINTREV»C=» KA$;NORM
4520 GOSUB9020
4540 IFK1=0GOTO4590
4550 IF(d(1,1)<>d(2,1)OR d(1,2)<>d(2,2))ANDK2<>0THENPRINTDIC$:GOTO4680
4560 GOSUB9010:IFz THENz=0:GOTO4680
4570 d(3,1)=d(1,1):d(3,2)=d(1,2)
4580 IFK2=0 GOTO4620
4600 IFd(2,1)=0THENPRINT PADEF$: GOTO4680
4610 d(3,1)=d(2,1):d(3,2)=d(2,2)
4620 FORI=1TOd(3,1)
4630 FORJ=1TOd(3,2)
4640 C(I,J)=K1*A(I,J)+K2*B(I,J)
4650 NEXT:NEXT
4670 L=3:GOSUB2670
4680 GOSUB9040:GOTO3500
5000 CLS
5010 PRINTREV»C=A.B»NORM
5020 GOSUB9010:IFz THENz=0:GOTO5150
5030 IFd(2,1)=0THENPRINT PADEF$:GOTO5150
5040 IFd(1,2)<>d(2,1)THENPRINT DIC$:GOTO5150
5050 d(3,1)=d(1,1):d(3,2)=d(2,2)
5060 FORI=1TOd(1,1)
5070 FORJ=1TOd(2,2)
5080 C(I,J)=0
5090 FORK=1TOd(1,2)
5100 C(I,J)=C(I,J)+A(I,K)*B(K,J)
5110 NEXT
5120 NEXT
5130 NEXT
5140 L=3:GOSUB2670
5150 GOSUB9040:GOTO3500
5500 CLS:PRINTREV»C=A» CHR$(243);NORM
5520 GOSUB9010:IFz THENz=0: GOTO5800
5530 GOSUB9000:IFz THENz=0: GOTO5800
5540 INPUT»n=»,NN: NN=INT(ABSNN)
5550 N=d(1,1):d(3,1)=N:d(3,2)=N
5560 FORI=1TON:FORJ=1TON: C(I,J)=0:D(I,J)=A(I,J): NEXT:C(I,I)=1:NEXT
5570 IFNN=0GOTO 5790
5580 PE=0:U=NN/2
5590 IFU=INT(U)THENPE=1: NN=U
5600 IFPE=0THENNN=NN-1
5610 FORI =1TON
5620 FORJ =1TON
5630 K=J+N:D(I,K)=0
5640 FORL=1TON
5650 U=C(I,L)
5660 IFPE=1THENU=D(I,L)
5670 D(I,K)=D(I,K)+U*D(L,J)
5680 NEXT
5690 NEXT
5700 NEXT
5710 FORI=1TON
5720 FORJ=1TON
5730 U=D(I,J+N)
5740 IFPE=0THENC(I,J)=U
5750 IFPE=1THEND(I,J)=U
5760 NEXT
5770 NEXT
5780 IFNN>0THEN5580
5790 L=3:GOSUB2670
5800 GOSUB9040:GOTO3500
6000 CLS:PRINTREV »DETERMINANT DE A»NORM
6020 GOSUB9010:IFz THENz=0:GOTO6110
6030 GOSUB9000:IFz THENz=0:GOTO6110
6040 L=1:M=4:GOSUB2810
6050 P=1:N=d(1,1):GOSUB500
6060 PRINT»Det(A)=»DR
6100 IFRG<d(1,1)THEN PRINT»Rang(A)=»:RG
6110 GOSUB9040:GOTO3500
6500 CLS
6510 PRINTREV»C=A» CHR$(240);NORM
6520 GOSUB9010:IFz THENz=0:GOTO6670
6530 GOSUB9000:IFzTHENz=0:GOTO6670
6540 INPUT»Multiplication par Det(A) «,Z$
6550 N=d(1,1):d(3,1)=N:d(3,2)=N
6560 FORI=1TON
6570 FORJ=1TON:D(I,J)=A(I,J): NEXT
6580 FORJ=N+1 TON+N:D(I,J)=0:NEXT
6590 D(I,N+1)=1
6600 NEXT
6610 P=N:GOSUB500
6620 IFDR=0THENPRINT»La matrice n'est pas inversible.»:GOTO6670
6630 PRINT»Det(A)=»:DR
6640 IFZ$<>»O» THENDR=1
6650 FORI=1TON:FORJ=1TON: C(I,J)= DR*D(I,J+N):NEXT:NEXT
6660 L=3:GOSUB2670
6670 GOSUB9040:GOTO3500
7000 CLS:PRINTREV»VALEURS PROPRES DE A»NORM
7020 GOSUB9000:IFz THENz=0:GOTO7630
7030 IFd(1,1)<2THEN100
7040 N=d(1,1):A(0,0)=1:d(4,1)=N: d(4,2)=N:CT=1
7050 INPUT»P=»,P:INPUT»Q=»,Q
7060 FORI=1TON:FORJ=1TON:C(I,J)=0: NEXT:NEXT
7070 FORK=1TON
7080 FORI=1TON:C(I,I)=C(I,I) +A(0,K-1):NEXT
7090 FORI=1TON
7100 FORJ=1TON
7110 D(I,J)=0
7120 FORL=1TON:D(I,J)=D(I,J)+ C(I,L)*A(L,J): NEXT
7130 NEXT
7140 NEXT
7150 L=4:M=3:GOSUB2810: A(0,K)=0
7160 FORI=1TON:A(0,K)=A(0,K)+C(I,I): NEXT
7170 A(0,K)=-A(0,K)/K
7180 NEXT
```



PROGRAMMES POUR CASIO PB1000 ... PROGRAMMES POUR CASIO PB1000 ...

```

7190 IFN=2THENP=A(0,1):Q=A(0,2): GOTO7400
7200 J=0
7210 B(0,0)=A(0,0):B(0,1)=A(0,1)-P*B(0,0)
7220 FORK=2TON
7230 B(0,K)=A(0,K)-P*B(0,K-1)-Q*B(0,K-2)
7240 NEXT
7250 C(0,0)=B(0,0):C(0,1)=B(0,1)-P*C(0,0)
7260 IFN=3THEN7300
7270 FORK=2TON-2
7280 C(0,K)=B(0,K)-P*C(0,K-1)-Q*C(0,K-2)
7290 NEXT
7300 C(0,N-1)=-P*C(0,N-2)-Q*C(0,N-3)
7310 DD=C(0,N-2)*C(0,N-2)-C(0,N-1)*C(0,N-3)
7320 U=B(0,N-1)*C(0,N-2)-B(0,N)*C(0,N-3)
7330 V=B(0,N)*C(0,N-2)-B(0,N-1)*C(0,N-1)
7340 IFDD=0THENPRINT»Valeurs de P et Q inadaptees !»:GOTO7630 7350 X=U/
DD:Y=V/DD
7360 P=P+X:Q=Q+Y
7370 IF(ABSX+ABSY)<p THEN7400
7380IFJ<100THENJ=J+1:GOTO7210
7390 PRINT»Le calcul necessite trop d'iterations !»:GOTO7630
7400 T=P*P-4*Q
7410 IFT<0THEN7490
7420 U=(-P+SQRT)/2
7430 IFABSU<p THENU=0
7440 PRINT»L»CT»»»U
7450 U=(-P-SQRT)/2
7460 IFABSU<p THENU=0
7470PRINT»L»CT+1"»»U::GOSUB9040 :GOTO7540
7490 U=ABS(SQR-T)/2
7500 IFABSU<p THENU=0
7510 IFABS(P/2)<p THENP=0
7520 PRINT»L»CT»»»;-P/2"»»ROUND(U,-8)»i»
7530 PRINT»L»CT+1"»»;-P/2"»»ROUND(U,-8)»i»::GOSUB9040

```

```

7540 N=N-2:CT=CT+2
7550 FORI=0TON:A(0,I)=B(0,I):NEXT
7560 IFN>2THEN7200
7570 IFN=2THENP=B(0,1): Q=B(0,2):GOTO7400
7580 IFN<1THEN7620
7590 U=-B(0,1)/B(0,0)
7600 IFABSU<p THENU=0
7610 PRINT»L»CT»»»U
7620 d(3,1)=0:d(3,2)=0
7630 GOSUB9040:GOTO3500
8000 CLS
8010 PRINTREV»SYSTEME : A.x=U»NORM
8020 GOSUB9010:IFz THENz=0:GOTO8180
8030 IFDV(1)=0THENPRINTPADEF$:GOTO8180
8040 GOSUB9000:IFz THENz=0:GOTO8180
8050 IFd(1,2)<>DV(1)THENPRINTDIC$:GOTO8180
8060 N=DV(1):DV(3)=N
8070 FORI=1TON:D(I,N+1)=VE(1,I): NEXT
8080 L=1:M=4:GOSUB2810
8090 P=1:M=4:GOSUB500
8100 IFDR=0THENPRINT»Le systeme est degenerate !»:GOTO8180
8110 PRINT»Solution:»
8120 FORI=1TON
8140 PRINT»X»I»»»D(I,N+1);
8160 VE(3,I)=X
8170 GOSUB9040:NEXT
8180 GOSUB9040:GOTO100
9000 IFd(1,1)<>d(1,2)THENPRINTDIC$:z=1
9005 RETURN
9010 IFd(1,1)=0THEN PRINTPADEF$:z=1
9015 RETURN
9020 PRINTCHR$(241)::INPUT» = «,K1
9030 PRINTCHR$(242)::INPUT» = «,K2:RETURN
9040 INPUT»»,ZS:RETURN

```

CALCULS SUR LES DATES POUR CASIO PB1000

Lorsqu'on le lance, ce programme affiche sur un premier écran l'heure et la date sous une forme intelligible. Si vous tapez sur une touche vous vous retrouvez devant un menu qui vous propose les choix suivants :

- [1] Calendrier perpétuel
- [2] Calcul du nombre de jours entre deux dates
- [3] Calcul de la date n jours après une date donnée
- [4] Retour à la pendule

L'utilisation de chacun des sous-programmes est intuitive. Daniel PRATLONG.

```

10 CLS
20 LOCATE24,0:PRINTTIMES$
30 T=1:GOSUB130:T=0:PRINTA$
40 LOCATE9,2: PRINTMID$(DATE$,4,2)» «::GOSUB60:
PRINTM$» «RIGHT$(DATE$,4)
50 IFINKEY$=»»THENLOCATE24,0: PRINTTIMES$: GOTO50ELSE90
60 RESTORE80:I=VAL(LEFT$(DATE$,2)): FORJ=1TOI:READM$:NEXT:RETURN
70 DATADimanche,Lundi,Mardi,Mercredi, Jeudi,Vendredi,Samedi
80 DATAJanvier,Fevrier,Mars,Avril,Mai, Juin,Juillet,Aout,Septembre,Octobre,
Novembre,Decembre
90 CLS:PRINT»Calendrier:.....[1]»,»Entre deux dates:[2]»,
«Date + n jours:..[3]»,»Retour:.....[4]»;
100 I=VAL(INKEY$):ONI GOTO120,270,320,10:GOTO100 110 *** Calendrier **
120 G$=»»:GOSUB230
130 IFT THENJ=VAL(MID$(DATE$,4,2)):
M=VAL(LEFT$(DATE$,2)):A=VAL(RIGHT$(DATE$,4))
140 MO=M:AN=A
150 IFM>2THEN170
160 A=A-1:M=M+12
170 E=INT(A/100):K=A-E*100: B=INT (2.60001*(M-2)-.2)+J+K:B=B+INT(K/4)+
INT(E/4)-2*E:B=B-INT(B/7)+1:RESTORE70
180 FORI=1TOB:READA$:NEXT
190 IFT THENRETURN
200 PRINT»Le»J;CHR$(8); MO;CHR$(8);AN»est un «A$

```

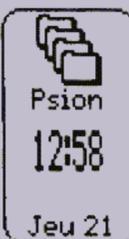
```

210 INPUT»»,AS:GOTO90
220 *** Saisie date **
230 CLS:PRINT»Donnez une»G$» date:»:INPUT»JJ: «,J:
INPUT»MM: «,M:INPUT»AAAA: «,A
240 IFJ<1ORJ>31ORM<1ORM>12 ORA<1901THENPRINT»Erreur»:GOTO230
250 RETURN
260 *** Entre 2 dates **
270 GOSUB290:G$=»»:GOSUB230: GOSUB280:R=N:G$=» autre»:
GOSUB230:GOSUB280:R=R-N:PRINTABSR»jours entre ces 2 dates.»:GOTO210
280 N=365.25*(A-1901)+A(M)+J:N=INT(N):RETURN
290 ERASEA:DIMA(12):A(1)=0:A(2)=31: A(3)=59.25:A(4)=90.25:
A(5)=120.25:A(6)=151.25:A(7)=181.25: A(8)=212.25:A(9)=243.25:
A(10)=273.25:A(11)=181.25:A(8)=212.25: A(9)=243.25:A(10)=273.25:
A(11)=304.25:A(12)=334.25
300 RETURN
310 *** Date +n **
320 CLS:GOSUB290
330 G$=»»:GOSUB230:GOSUB280: INPUT»+ n jours: «,n:N=N+n: a=INT(N/
365.25+1901)
340 FORm=1TO11:j=INT(N-(a-1901)*365.25-A(m))+1
350 IFj<32THEN360ELSENEXT
360 PRINTn»jours aprËs le»J;M;A» c'est le»j;m;a:GOTO210 370 PRINTn»jours
après le»J;M;A» c'est le»j;m;a:GOTO210

```



Société Organisation Deb's
 Contact Thierry Desbruères
 Adresse 139, av Charles de Gaulle
 92200 Neuilly sur Seine
 ☎Téléph (1) 47 45 20 44
 ☎Fax (1) 46 24 76 81



Ven 5 Novembre 1993

14	15
16 16:00 Rosalie sieste coquine (☎13:00) Δ	17
18	19 19:30 Pharmacie pour reconstituant Δ
20	21 21:00 Prendre mon venin de cobra royal avant Stéphanie Δ
22	

1er

Journée d'enfer avec votre Serie 3a

ROUTINE SONORE POUR PB1000

En assembleur automodifié
Par Pierre-Philippe COUPARD

Cet article s'adresse à tous les possesseurs d'un PB1000 de CASIO, que la machine soit de base ou étendue. Le PB1000 de CASIO offre beaucoup de possibilités à son utilisateur des possibilités d'entrées/sorties intéressantes, un écran tactile, 2 langages puissants ... Cependant, une chose lui manque vraiment pour combler son utilisateur : une gamme de sons variés. Nous vous proposons ici une routine en assembleur qui comblera ce défaut.

De quoi s'agit-il ?

En fait, le PB1000 peut générer deux sons, que l'on peut entendre en tapant les instructions BEEP 0 (son grave) et BEEP 1 (son aigu). Mais deux sons, c'est insuffisant pour égayer un jeu, ou pour faire de petites mélodies. Donc, il nous faut générer ces sons nous-même. Pour celà, il nous faut programmer une routine en assembleur qui se chargera d'exciter le buzzer (c.à.d le composant électronique qui produit le son à l'intérieur du PB1000) à la bonne cadence. De plus, pour des raisons de simplicité, nous devons passer à cette routine des paramètres sous forme de variables mais pas avec des POKES car ils prennent du temps en BASIC et rendent le programme illisible. Enfin, ces paramètres seront en unités internationales, c'est-à-dire en Hertz (Hz) pour la fréquence du son et en secondes (s) pour la durée du son. Comme ça, tout le monde se comprend. Ces paramètres se nomment «Sfreqcy» et «Slength» ce qui, a défaut d'être original, a le mérite d'être clair. Comment ça marche ?

Le buzzer piézo-électrique ne connaît que 2 états : soit sa lame de quartz est «en arrière», soit elle est «en avant». En fait, cette lame de quartz se comporte un peu comme la membrane du haut-parleur interne de tous les PC de bureau, qui est piloté de la même façon.

L'état du buzzer est déterminé par les bits 6 et 7 du registre interne PD (Port de Données) du processeur HDC61700 : si ces deux bits sont à 1, la lame est en avant, s'ils sont à 0, elle est en arrière. Il suffit donc d'inverser ces deux bits très rapidement pour générer un son.

Comme dans un PC, nous devons donc utiliser la méthode classique pour créer un son, à savoir une boucle temporisée selon la fréquence désirée.

L'algorithme correspondant est le suivant:

```
/* Début de boucle */
/* Inversion de l'état du buzzer */
/* temporisation réglable */
/* Fin de boucle : arrêt quand on atteint la durée d'émission voulue */
```

Si ce principe marche très bien sur un PC, il n'en est pas de même pour notre PB1000. En effet, le HDC61700 est cadencé «seulement» à 910 kHz, et, pour obtenir un son d'une fréquence moyenne ou élevée, la durée de la temporisation devient comparable à la durée des autres instructions qui composent la boucle. Celà signifie que, dans ces fréquences, il suffit d'ajouter ou de retrancher la valeur minimale de précision à la temporisation pour que le son saute carrément une octave, ou au moins plusieurs tons! En pratique, la routine ainsi programmée n'offre donc pas un choix de fréquences suffisant pour considérer comme acceptable l'écart entre la fréquence demandée et la fréquence réellement émise par le buzzer. Alors, que faire ?

Détaillons le problème : la temporisation est évidemment une boucle vide qui effectue un certain nombre d'itérations dans le seul but de perdre du temps. On ne va pas s'amuser à temporiser en remplissant la mémoire avec des instructions NOPs (No



Operation), d'autant que la zone assembleur maximale allouée sur PB1000 est de 4 Ko. Or, une telle boucle vide comporte quand même des instructions, et ces instructions durent chacune un certain nombre de cycles machine. L'expérience (et le tableau des mnémoniques assembleur) montre que la durée minimale de la boucle, donc le «pas» de réglage de la temporisation, est de l'ordre d'une dizaine de cycles par itération. Vu la cadence de l'horloge, celà engendre une précision faible.

L'idéal serait de pouvoir disposer d'une temporisation réglable avec le «pas» le plus petit possible, soit un seul cycle machine, pour compenser le manque de vitesse de l'horloge. Cependant, c'est évidemment impossible avec une boucle simple. Il faut donc ruser ...

La ruse !

Il existe une astuce pour résoudre notre problème. Prenons un exemple: supposons qu'on dispose d'une boucle de temporisation qui dure 10 cycles par itération et qu'on veuille une temporisation pendant 24 cycles. On ne peut pas demander à la boucle d'effectuer 2.4 itérations. On lui demandera donc d'en faire 2, soit 20 cycles en tout. A ce stade, il nous reste encore 4 cycles à perdre pour atteindre notre objectif. Il suffit donc de rajouter à la suite de la boucle une instruction de saut qui dure 4 cycles et à qui on demandera de sauter à l'instruction qui la suit.

Cette instruction sera donc parfaitement inutile, sauf pour perdre 4 cycles. Cette instruction s'appelle JP m (m est l'adresse absolue de saut).

Supposons maintenant qu'on veuille temporiser pendant 26



cycles : Il suffit de rajouter à la suite du JP l'instruction NOP, déjà présentée plus haut, qui dure 2 cycles : $20+4+2=26$ cycles. L'adresse de saut de JP sera alors l'adresse du NOP, puisqu'il le suit immédiatement.

Si on voulait 28,30 ou 32 cycles, on rajouterait de la même manière le nombre de NOPs correspondant à la suite de JP. Si on voulait 34 cycles, on enlèverait tous les NOPs et on demanderait alors à la boucle de faire 3 itérations, soit 30 cycles : $30+4=34$ cycles ... et ainsi de suite.

Par conséquent, si on écrivait 5 versions différentes de la routine de son, chacune comportant une boucle à vide suivie de son JP et du nombre de NOP adéquats, et que l'on branchait celle qui nous intéresse en fonction du nombre de cycles restant à perdre après la boucle à vide, on aurait déjà une routine avec une temporisation réglable à deux cycles près.

Pour avoir une temporisation réglable à un cycle près, il suffirait de remplacer l'instruction de saut absolu JP m (qui dure 4 cycles) par l'instruction JR n (qui dure 3 cycles) en prenant garde à remplacer l'adresse de saut absolue m par un offset de saut relatif n. On réécrirait alors 5 autres versions différentes de la routine de son avec également chacune une boucle à vide, un JR et le nombre de NOPs adéquats.

Au total, on aurait donc 10 versions de la routine de son, 5 pour les temporisations paires et 5 autres pour les temporisations impaires. Grâce à ce système, on aurait une précision de un cycle, et c'est ce qu'on recherche.

Cependant, si ce système à l'avantage de marcher, il consomme une quantité de mémoire importante, et tout les possesseurs de PB1000 n'ont pas la chance d'avoir une extension de 32 Ko. C'est là qu'intervient la technique du code auto-modifié qui va nous permettre de rendre à notre programme une taille raisonnable.

Le code auto-modifié

La simple évocation de ce procédé va rappeler des souvenirs à tous les programmeurs qui ont travaillé sur des micros d'un autre âge. Le code auto-modifié est exactement ce qu'il nous faut ici car le PB1000 se rapproche plus du Sinclair ZX81 (paix à son âme) que du PC 486.

Tout d'abord, précisons que cette technique peut s'appliquer à n'importe quel langage mais il faut ABSOLUMENT que la machine sur laquelle tourne le programme ne possède pas de mémoire cache (ou antémémoire), ce qui bien sûr est le cas du PB1000.

Le principe général consiste tout simplement à créer un programme qui modifie ses propres instructions. Dans notre exemple, au lieu de créer 10 routines de son, on va en faire une seule, comportant une boucle de temporisation, une instruction JP et 4 NOPs. La routine sera précédée d'un programme qui se chargera de déterminer le nombre d'itérations de la boucle et le nombre de cycles restant à perdre. Si ce dernier nombre est impair, on ira écrire le code de JR n à la place de celui de JP m, puis on déterminera l'argument de l'instruction de saut pour qu'elle saute au 4^e, 3^e, 2^e ou 1^{er} NOP et on ira l'écrire juste après le code de l'instruction. Enfin, on démarrera la routine de son.

Les programmes :

Dans la routine en assembleur, pour remplacer une division entière par deux rotations binaires plus rapides, nous avons choisi une boucle de temporisation de 16 cycles, d'où le nombre élevé de NOPs. Cela ne change en rien le principe. De plus, nous

avons inclus deux opérations en chaîne de caractères qui permettent au programme de préparation de récupérer les valeurs des variables «Sfreqcy» et «Slength», et de simplifier le calcul du nombre d'itérations de la boucle à vide et de la boucle principale. Le calcul de ces expressions et le transfert des résultats dans les registres internes fait appel à deux routines système.

Les valeurs suivantes sont à respecter ABSOLUMENT sous peine de risquer le plantage :

$74 < \text{Sfreqcy} < 6594$ (en Hz)

$8.10^{-5} < \text{Slength} < 4.9$ (en s)

Ce code a été optimisé pour être le plus rapide possible. Cependant, il faut tenir compte du fait que la préparation du son demande une fraction de seconde qui augmente la durée totale du processus.

La procédure d'assemblage est standard : on tape le source en fichier texte et on l'assemble grâce à l'option idoine. La routine entière, une fois assemblée, ne prend que 245 octets qui passent inaperçus, même sans extension mémoire.

Un exemple est donné en BASIC qui montre comment on exploite la routine. Il s'agit d'un petit «synthétiseur» qui permet de jouer de la musique dans la gamme de Do majeure diatonique, et il est accordé !

P.P.C

Programme à assembler sous le nom «SOUND.EXE» :

```

ORG &H7000           ;Adresse de debut de la routine : &H7000
                    (HEXA)
START PREPA         ;La routine est lancee au label PREPA
EXPRW: EQU &H9989   ;Routine systeme : calcul d'une expr.
                    numerique
BIN02: EQU &H98CD   ;Routine systeme : conversion reel ->
                    entier
BUFFR: DS 2         ;Buffer (EXPRW et BIN0 modifient les
                    registres internes. On stocke donc des
                    variables en RAM avant)
LOOP: LD $2,$1      ;Debut de la boucle de creation du son
PST PD,$3           ;Basculement du buzzer flip-flop (bits
                    6 et 7 du registre PD)
WAIT: SB $2,1       ;Debut de la boucle de temporisation
                    grossiere (Cette instruction pourra etre
                    remplacee par une instruction de saut
                    si cette temporisation est inutile) :
NOP                 ;Cette boucle permet de faire une
                    premiere
NOP                 ;temporisation réglable par paquets de
                    16
NOP                 ;cycles selon la valeur de $2. Elle est
                    sautee
NOP                 ;par remplacement d'une instruction si
                    $2=0
JP NZ,WAIT
JP 0                ;(JP est une instruction à 4 cycles. Elle
                    restera JP ou deviendra JR (instruction
                    à 3 cycles) selon que le nombre de
                    cycle à attendre et pair ou impair.
                    «0» sera remplace plus loin dans la rou-
                    tine par l'adresse adéquate pour sauter
                    des NOPs (sil'instruction reste JP) ou
                    par un déplacement relatif (si l'instruc-
                    tion devient JR).;

```



NOP ;La temporisation précédente est complétée par
 NOP ;une seconde temporisation réglable à 1 cycle
 NOP ;pres (1 NOP = 2 cycles, et selon l'instruction
 NOP ;de saut JP ou JR), de 0 à 14 cycles.
 NOP ;Au final, les deux temporisations permettent
 NOP ;d'obtenir une grande temporisation réglable au
 NOP ;cycle près (précision !)
 XR \$3,192 ;Inversion des bits 6 et 7 du PD
 SBW \$15,\$10
 JR NZ,LOOP ;Fin de la boucle de création du son
 PST PD,\$4 ;On remet PD comme on l'a trouvé (c'est plus propre)
 RTN ;On rend la main au programme appelant

 OP1: DB «2*Slength*Sfreqcy»,0 ;Cette opération sera utilisée pour ; calculer le nombre de boucles LOOP ...
 OP2: DB «455000/3/Sfreqcy»,0 ;... et celle-ci pour calculer le nombre de cycles de temporisation.
 PREPA: PRE IX,&H7038 ;Début: on charge l'adresse de OP2 dans IX
 CAL EXPRW ;Calcul de OP2 : retourne un réel
 CAL BIN02 ;Conversion en entier
 LDW \$10,20
 SBW \$15,\$10 ;On retranche le nbr. de cycles dus aux instructions autour de la boucle WAIT
 DIUW \$15 ;multiplication par 16 sur un mot
 DID \$15 ;On divise par 16 sur un octet
 LDW \$10,&H7000
 STW \$15,(\$10) ;On sauve le mot dans BUFR
 PRE IX,&H7026 ;On charge l'adresse de OP1 dans IX
 CAL EXPRW ;Calcul de OP1 : retourne un réel,
 CAL BIN02 ;conversion en entier
 LDW \$10,&H7000
 LDW \$0,(\$10) ;On recharge le mot dans BUFR
 SBC \$0,3
 JR NC,CONT1 ;Si nbr. de cycles à sauter est >3, on va à CONT1
 SB \$1,1 ;sinon on ôte 1 au nbr. de boucles à 16 cycles
 AD \$0,16 ;et on ajoute 16 au nbr. de cycles dans la boucle
 WAIT
 CONT1: GST PD,\$4 ;On stocke la valeur originale de PD dans \$4
 LD \$3,\$4
 OR \$3,192
 SB \$3,64 ;et on calcule dans \$3 le PD utilise en flip-flop
 LDW \$6,&H7007
 SBC \$1,0 ;Si le nbr. de boucles à 16 cycles est nul,

JR Z,CONT2 ;on va à CONT2 directement,
 LDW \$8,10
 ADW \$6,\$8 ;sinon, on déplace l'adresse du JP à changer dans la boucle de 10 octets
 CONT2: LD \$8,\$0 ;On détermine si le nombre de cycles dans la
 OR \$8,1 ;boucle WAIT est pair ou impair
 SBC \$8,\$0
 JR Z,PIMP ;S'il est impair, on va à PIMP, sinon à PPAIR
 PPAIR: SB \$0,4
 LDW \$8,\$6
 CAL CALC ;On calcule l'offset de l'adresse du JP
 LD \$12,&H37 ;On charge le code de l'instruction JP,
 ST \$12,(\$6) ;on écrit l'instruction dans la boucle,
 ADW \$6,\$10 ;on ajoute l'offset à l'adresse du JP,
 STW \$8,(\$6) ;et on écrit l'adresse de saut juste apres JP
 JP LOOP ;On saute à la boucle de création du son
 PIMP: SB \$0,3
 LDW \$8,0
 CAL CALC ;On calcule l'offset de l'adresse du JR
 LD \$12,&HB7 ;On charge le code de l'instruction JR
 ST \$12,(\$6) ;on écrit l'instruction dans la boucle
 ADW \$6,\$10 ;on ajoute l'offset à l'adresse du JR
 SB \$8,1
 ST \$8,(\$6) ;et on écrit le relatif de saut juste apres JR
 JP LOOP ;On saute à la boucle de création du son
 CALC: BID \$0 ;CALC s'occupe de calculer l'offset pour
 LDW \$10,10 ;l'adresse de l'instruction qui viendra sauter
 SBC \$1,0 ;les NOPs (ou toute la boucle à 16 cycles, si
 JR NZ,CONT3 ;le nombre total de cycles est < 16) dans LOOP
 ADW \$8,\$10
 CONT3: SB \$10,\$0
 ADW \$8,\$10
 LDW \$10,1
 RTN

Programme de «synthétiseur» en BASIC utilisant «SOUND.EXE»

```

10 CLEAR:CLS:DIM N(7):FOR N=0 TO 7:READ N(N):NEXT N
PRINT REV:»SYNTHETISEUR EN DO MAJEUR«;NORM:
PRINT»Utilisez les touches '1' à '9' pour entendre les notes de
'Do' a 'Do' ...«;:Slength=.3
20 N=ASC(INKEY$)-49:IF N<0 OR N>7 THEN 20
30 Sfreqcy=N(N):CALL»SOUND.EXE«:GOTO 20
100 DATA 262,294,330,350,393,440,495,524
  
```

TRUCS ET ASTUCES...

Vous êtes des milliers à apprécier cette rubrique ; son but étant de tirer toujours plus de sa machine. Continuez à nous écrire et à nous envoyer vos trucs et astuces pour le plus grand bien de tous.

Faire défiler les caractères sur l'écran de votre Sharp PC 1403. (par Loïc FIEUX)

L'écran de votre ordinateur Sharp PC 1403 est capable d'afficher 24 caractères. Comment faire si nous souhaitons afficher un texte comportant plus de 24 caractères ? Le petit programme qui suit apporte une réponse en proposant de faire défiler le texte. Ce programme est écrit en Basic, il va de soi qu'il est destiné à n'être qu'un sous-programme d'une application plus lourde. Pour cette raison, les numéros des lignes sont remplacés par #, #+1, #+2 et #+3.

Listing :

```
#:»S» CLEAR :DIM T$(0)*50: INPUT «Votre texte
<<T$(0): IF LEN T$(0)<25 PRINT T$(0): END
#+1:WAIT 30: PRINT LEFT$( T$(0),24): CALL 1208
#+2:FOR J=2 TO LEN T$(0)-24: PRINT MID$(
T$(0),J,24): CALL 1208: NEXT J: WAIT
#+3:PRINT MID$( T$(0),J,24): END
```

Exemple d'utilisation : placez une chaîne dans la variable T\$(0) et brancher l'exécution sur la ligne #+1.

N.B. : Les habitués des clubs d'utilisateurs de Sharp PC auront peut-être reconnu l'une des multiples versions de cet indispensable utilitaire de scrolling.



La face cachée des variables... (par Loïc FIEUX)

Les variables (mémoires numériques) des calculatrices Casio graphiques sont au nombre de 26 (4 dans le cas de la fx 3900P, 28 mémoires pour les fx 7700/7800 et 8700/8800). Vous pouvez bien entendu copier dans ces variables des nombres de votre choix. Il vous faut néanmoins savoir que la machine se charge de placer certains résultats dans certaines variables à l'occasion de calculs. Ainsi, les variables I et J conservent le dernier résultat de conversion de coordonnées. Selon le cas, I contient x ou r et J contient y ou teta. X et Y contiennent les coordonnées du pointeur validées par une pression sur [EXE] (en utilisant Plot ou Trace). P, Q, R, U, V et W contiennent des résultats de calculs statistiques. En mode SD (ou SD1), W contient n, V contient la somme des x et U contient la somme des carrés de x. En mode LR (LR1), U, V et W ont les mêmes fonctions qu'en mode SD, par ailleurs, R contient les sommes des produits xy, Q contient la somme des y et P contient la somme des carrés de y.

N'oublions quelques mémoires spéciales. Ans mémorise le dernier résultat numérique obtenu. Ibr contient la dernière expression validée (fx 8000G et 8500G). Des nombres sont en mémoire en permanence, il s'agit de 10 [10^x], de e [SHIFT] [ln] (e=2,71828...), de π (π=3,14159...) etc.

Déchiffrez le numéro de série de votre matériel Hewlett-Packard (par Loïc FIEUX)

Sur tous les appareils Hewlett-Packard, vous trouverez un numéro de série de la forme AASSFNNNNN. Ajoutez 1960 à AA, vous trouverez l'année de construction. SS est un nombre entre 01 et 52, il correspond à la semaine de fabrication dans l'année de fabrication. F correspond au centre de production (A : Etats-unis, S : Singapour). NNNNN est le nnnnnième machine construite pendant la semaine considérée. Ainsi, mon HP-48GX porte le numéro 3333S16671, elle est 16671ème machine produite à Singapour pendant la semaine du 16 au 22 août 1993.



Quelques astuces graphiques pour votre Casio fx (par Loïc FIEUX)

Les astuces qui suivent sont destinées aux utilisateurs des calculatrices graphiques Casio fx. Certaines de ces astuces sont relatives aux manipulations du repère de l'écran dans lequel sont tracées les représentations graphiques de fonctions et autres motifs programmés par l'utilisateur. En mode direct, la touche [RANGE] permet la modification des paramètres de l'écran. Dans un programme, l'instruction Range permet de spécifier les paramètres de l'écran. Les paramètres Xmin, Xmax, Xscl, Ymin, Ymax et Yscl dont il est question ci-après, sont les paramètres de la fonction Range. N'oubliez pas que la syntaxe de celle-ci, dans un programme est Range Xmin, Xmax, Xscl, Ymin, Ymax, Yscl.

- Supprimer les graduations sur les axes

Vous pouvez masquer les graduations des axes du repère orthogonal sur lequel sont tracées mes courbes. Pour cela, il vous suffit, à l'aide de la commande Range, de spécifier Xscl=0 (pour masquer les graduations de l'axe des abscisses) et/ou Yscl=0 (pour masquer les graduations de l'axe des ordonnées).

- Ne pas faire apparaître les axes du repère

Les axes du repères peuvent ne pas être visibles. Pour masquer l'axe des abscisses, il faut que Xmin>0 et Xmax>0, ou bien, Xmin<0 et Xmax<0. Pour masquer l'axe des ordonnées, il faut que Ymin>0 et Ymax>0, ou bien, Ymin<0 et Ymax<0.

- Exploiter la position du pointeur

Le pointeur est l'index clignotant (une croix ou un point selon le type de votre calculatrice) que vous déplacez sur l'écran graphique à l'aide des flèches de déplacement. Vous faites apparaître le pointeur en utilisant Trace en mode direct. Dans un programme, vous avez intérêt à utiliser Plot Disp (où Disp est le triangle de Display obtenu, selon le cas, par [SHIFT] [RANGE] [F3] ou [ALPHA] [GRAPH] ou [SHIFT] [:]). Ainsi, vous déplacez le pointeur et ses coordonnées (sa position) à l'instant où vous avez appuyé sur [EXE] sont enregistrées dans les mémoires X et Y. Les contenus des variables X et Y sont exploitables, ils correspondent à une position sur l'écran graphique. Cela ne vous donne pas des idées de programmes graphiques ?

- Eviter le scintillement de l'écran lors des tracés

Comme vous le savez, les Casio fx graphiques utilisent un écran disposant de deux modes, le mode graphique et le mode texte. Sur les fx de la série 7000/8000, l'écran graphique laisse apparaître le contenu de l'écran texte lors d'un tracé faisant intervenir de nombreuses instructions PLOT (le plus souvent incluses dans des boucles (label Lbl et commande saut Goto). Le «clignotement» ne peut être supprimé, l'astuce consiste à nettoyer l'écran texte en affichant huit lignes vides avec un petit sous-programme dans le style de celui-ci : 8->I:Lbl n:»_»:Dsz I:Goto n Où n est à remplacer par un nombre de 0 à 9. Remarquez que l'on exploite ici un «effet secondaire» de la fonction de décrémentation Dsz,

celle-ci empêchant l'exécution de la fonction qui la suit quand la variable qu'elle décrémente devient nulle.

- Un repère orthonormé pour ma fx graphique

Il existe trois types d'écrans différents montés sur les fx graphiques. Le plus courant est l'écran 6144 pixels (96x64) dont 5985 (95x63) sont utilisables, cet écran est monté sur la fx 7000/7200, 7500, 7700/7800, 8000, 8500 et 8700/8800. Les fx 6000 et 6500 (diffusées entre 1986 et 1988) disposaient d'un écran de 3072 pixels (96x32) dont 2945 (95x31) utilisables. Le bébé fx, la fx 6300/6800, a un tout petit écran de 960 pixels (40x24) dont 897 (39x23) utilisables. Ces informations techniques vont nous permettre de définir des repères orthonormés.

Dans le cas d'un écran 95x63, il faut que :

$$(X_{max}-X_{min})=(95/63)(Y_{max}-Y_{min})$$

$$\text{ou } (Y_{max}-Y_{min})=(63/95)(X_{max}-X_{min})$$

Dans le cas d'un écran 95x31, il faut que :

$$(X_{max}-X_{min})=(95/31)(Y_{max}-Y_{min})$$

$$\text{ou } (Y_{max}-Y_{min})=(31/95)(X_{max}-X_{min})$$

Dans le cas d'un écran 39x23, il faut que :

$$(X_{max}-X_{min})=(39/23)(Y_{max}-Y_{min})$$

$$\text{ou } (Y_{max}-Y_{min})=(23/39)(X_{max}-X_{min})$$

Notez que :

$$95/63 = \text{env. } 1,5.$$

$$95/31 = \text{env. } 3.$$

$$39/23 = \text{env. } 1,7.$$

- Centrer l'origine du repère

Pour centrer l'origine du repère, il faut, d'une part, que les valeurs absolues de Ymin et Ymax soient égales, d'autre part, que les valeurs absolues de Xmin et de Xmax soient égales.

- Erreur «Ma Error» avec Graph Y=

L'instruction Graph Y= déclenche une «Ma Error» si les paramètres de l'instruction Range sont incorrects, en particulier, si : Xmin = Xmax et/ou si Ymin=Ymax.

Votre pocket et votre calculatrice ont trouvé leur maître !

- Plus de 10.000 écrans d'informations sur votre pocket ou votre calculatrice !
- Echangez vos trucs et astuces à travers 30 forums thématiques !
- Des milliers de programmes !
- Trucs et astuces, Scoops, Infos,
- Messagerie, Petites annonces,
- Micro-informatique, etc.



3615

Le minitel de votre ordinateur de poche
et de votre calculatrice graphique !

CALCULATOR

Démo graphique pour Casio FX 180P (par Damien MARMISSOLLE)

«CARL LEWIS» by DAM.

```

10 CLS : VAC : DIMA$ (H)
20 RESTORE 1000 : READ A$ : LOCATE I,0 :
  PRINT « « ; CHR$ (252) ;; DEFCHR$ (252) = A$
22 READ A$ : LOCATE I, 1 : PRINT « « ;
  CHR$(253);;
  DEFCHR$ (253) = A$
24 READ A$ : LOCATE I+2,0 : PRINT CHR$ (254);;
  DEFCHR$ (254) = A$
26 READ A$ : LOCATE I+2,1 : PRINT CHR$ (255);;
  DEFCHR$ (255) = A$
35 RESTORE 1010
50 FOR K = 1 TO 4 : READ A$ : DEFCHR$(251+K)
  = A$ : NEXT
60 I=I+1 : IF I<30 THEN 20 ELSE 10
1000 DATA 00040FEDE , 040A10E0C0 , 0404000000 ,
  1E02000000
1010 DATA 0000000608 , 0000003020 , FEDE081000 ,
  FEE2000000
  
```

Du TTL en RS232C sur FX-850P et FX-880P...

Les mouvements d'informations à l'intérieur de nos 850 et 880 favorites, se font en mode parallèle. C'est-à-dire que le microprocesseur équipant ces dernières échange des informations avec sa mémoire grâce à 8 lignes parallèles appelé bus. Ce moyen de transport de l'information pourra aussi être utilisé pour transmettre les données nécessaires à une imprimante, reliée par vos soins à votre ordinateur de poche par l'intermédiaire de la FA 6 (du moins pour l'instant...). Votre 800 (FX-850P ou FX-880P) étant prévue pour communiquer par l'intermédiaire d'un éventuel modem (le Minitel, par exemple), mais les échanges en parallèle devenant impossibles sur de longues distances, les ingénieurs de CASIO ont eu la bonne idée d'équiper les 800 de la norme RS232C (enfin presque...). D'ailleurs, tout le monde vous le dira : Pour échanger des informations, le réseau Télécom reste le canal le plus pratique. Mais il ne peut laisser passer qu'un état à la fois parmi les deux possibles (1 ou 0). Par contre, en informatique, l'octet destiné à transmettre peut revêtir 256 valeurs différentes. Il devra donc subir une transformation et ses éléments binaires vont être placés chronologiquement, à la queue leu leu sur une même ligne. Cette modification s'appelle la sérialisation d'où le nom de transferts «en série». Par définition, les unités binaires sont des portes et donc, chacune d'entre elles sera à l'état 1 ou à l'état 0 et activée sur la ligne de sortie de l'interface par la présence ou l'absence d'impulsions électriques.

Deux modes de transmission sont possibles : le mode asynchrone et le mode synchrone. Nous nous limiterons au mode asynchrone, adopté par notre cher 800. Sous ce mode, l'envoi des données s'effectue sous forme d'un flot de bits. Un caractère est représenté par 7 ou 8 bits selon le choix de la procédure. Il est précédé d'un bit de START (départ, pour ceux qui ne connaissent pas l'anglais...) et terminé par un bit de parité et/ou un ou deux

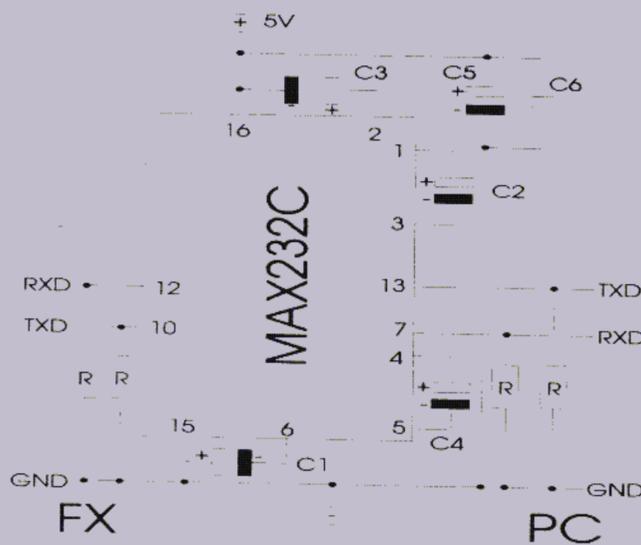
bits de STOP. La détection d'erreur s'effectuera donc par le contrôle de parité (un octet est dit pair lorsque le nombre de bits à l'état 1 est pair, le bit de parité sera donc ajouté à un octet dans le seul but de le rendre pair : un octet = 10101110 = 5 bits à l'état 1 donc le bit de parité sera 1 car 5+1 = 6 (nombre pair), si l'octet = 10011010 = 4 bits à l'état 1 donc le bit de parité sera 0 car 4+0=4 (nombre pair). Compris.), Je reprends ma phrase, donc, la détection d'erreur s'effectuera par le contrôle de parité, moyen cependant peu fiable ; Le meilleur, mais qui prend du temps, est le renvoi systématique du caractère envoyé pour contrôle.

La liaison entre une 800 équipée d'une FA6 et un autre ordinateur se révèle relativement simple, si l'on utilise un raccordement simplifié. La solution la plus simple est de n'utiliser que les liaisons TXD (Transmit Data, transmission des données), RXD (Receive Data, Réception des données) et la masse, en prenant soin de croiser les fils, c'est à dire : TXD et RXD d'un appareil reliés respectivement à RXD et TXD de l'autre. Les broches concernées sont vues page 73 du manuel de l'utilisateur de la 800 (la masse est traduite par GND).

Pour ceux qui ne possèdent pas de FA6, nous vous proposons un schéma théorique qui permettra aux personnes ayant quelques bases en électronique de réaliser une interface RS232C pour leur 800. Pour les autres, il faudra attendre le schéma pratique du prochain numéro... Les broches de la 800 sont les suivantes : RXD (5 en haut), TXD (7 en bas) et GND (15 en bas). Cette interface est basé sur l'utilisation d'un MAX232, qui aura pour conséquence de transformer les signaux TTL (0V et 5V) de la CASIO en RS232C (-12V/+12V) de votre AMIGA/ST/PC. Cette interface peut donc être utilisée sur tous les pockets fonctionnant en TTL.

La commande pour sauvegarder un programme par l'intermédiaire de la RS 232C est : SAVE»COM0: (+ configuration), SAVE#»COM0: (+ configuration) pour le MEMO, LOAD et LOAD# pour charger (il faut bien sûr un programme de transfert sur l'autre ordinateur, mais tous les softs de communication font l'affaire, Terminal sous Windows par exemple). Pour la partie configuration du port RS232C, le manuel de l'utilisateur est très bien fait, alors reportez-vous à la page 87 et 88. Bonne communication et A+.

Câble FX800P - ST/PC/AMIGA
(par DOM pour CASIO News)



R. 100 K.
C1, C2, C3, C4, C5. Condensateurs électrolytiques de 10 ou 22 µF (16V et +)
C6. Condensateur de 100 nF (16V et +)

Cet excellent article nous vient du club CASIO News dont l'adresse est la suivante :

CASIO News : Anthony TARDIVEL
La Gravelle - 35410 NOUVOITOU



MISE EN FRACTION SUR TI GALAXI 67 ET HP11C

par Abdelkédir MARZOUKI.

Ce programme est valable pour les calculatrices qui n'ont pas de fonctions directes de la mise en fraction.

Par exemple:

- on affiche 12,56
- appuyer sur enter ou =, donne 12,56
- appuyer sur A b/c donne 12 - 14/25
ou 12 14/25.
- appuyer sur 2Nd ou INV A b/c donne
314/25.

EXEMPLE SUR TI GALAXY 67.

Input C : Lbl 0 : B+1-> B : CxB-> A : fp(A)->D : if D=0 : goto 1 : goto 0 : Lbl 1 : disp A disp B :-> = STO

fp = partie décimale

if D=0 = dans le menu Test

Il faut mettre B comme compteur de nombre de zéro jusqu'à N nombres, et le programme test, si le nombre A qui est le numérateur qu'on cherche, doit être un nombre entier. Le programme s'arrête et affiche:

A numérateur

B dénominateur.

Input C Facultatif. Supprimer input C. Ecrire le prg. à partir de Lbl 0 jusqu'à disp

B. Appuyer sur ENTER ,affiche

B = ?

Mettre 0 B = 0

Enter

Review YN ?

Appuyer NO

C = ?

Mettre C = 12,56

Enter

Review YN ?

Appuyer NO

Affichage A = 314

Appuyer sur Enter

B = 25

Pour un autre ex. appuyer sur EVAL

EXEMPLE SUR HP11C.

C STO 2

B STO 0

A STO 1

1 Lbl 0 42 21 0

2 Rcl 0 45 0

3 1 1 B = mémoire 0 Compteur de 0 à N.

4 + 40 Au départ mettre la mémoire 0 à 0.

5 STO 0 44 0

6 Rcl 2 45 2 -> le nombre C

7 Rcl 0 45 0 -> le nombre B

8 X 20

9 STO 1 44 1 -> le nombre A = CxD

10 Rcl 1 45 1

11 Frac 42 44 la partie décimale de A, si elle est égale

12 x=0 43 40 si = à zéro, donne: A = Nombre entier

13 goto 1 22 1

14 goto 0 22 0

15 Lbl 1 42 21 1

16 Rcl 1 45 1

17 R/S 31

18 Rcl 0 45 0

19 R/S 31

Rcl 1 Numérateur

Rcl 0 Dénominateur

Exemple de mise en fraction :

12,56 -> C

B = 0+1 = 1

$$A = CxB = 12,56 \times 1 = 12,56$$

$$B = 1+1 = 2$$

$$A = 12,56 \times 2 = 25,12$$

$$B = 24+1 = 25$$

$$A = 12,56 \times 25 = 314$$

On rencontre le premier nombre entier de A qui est le numérateur. Le nombre B qui s'arrête à 25, donne le dénominateur.

Donc la fraction est: $\frac{314}{25}$

GRANDES FACTORIELLES SUR TI 66

par Daniel PRATLONG

Ne jetez pas vos vieilles TI66 ! Elles peuvent encore servir, en TP par exemple...

Imaginez qu'une fiole se renverse... hihihihhi ! La TI66 ne craint rien, mais il lui manque une fonction : la fonction factorielle.

C'est pourtant une fonction assez utile en maths (en probabilités).

Elle est définie par la formule suivante : $n! = 1 \times 2 \times 3 \times \dots \times n$

$$\text{Ex : } 5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

En général, les calculatrices s'arrêtent à 69! (= 1.7112... E 98).

Si vous demandez le calcul de 70! vous aurez droit à un joli message d'erreur. Le programme suivant permet de calculer de très grandes factorielles.

$$\text{Ex : } 500! = 1.2201... E 1134$$

Il utilise la formule de Stirling dont vous trouverez la démonstration dans tous les bons livres de maths.

Mode d'emploi:

Après avoir tapé votre nombre enfoncez sauvagement la touche RST puis R/S. Le programme affiche n! en une seule fois si $n < 20$ ou bien en deux fois. Dans ce cas, la mantisse s'affiche d'abord, tapez R/S et l'exposant s'affiche. J'ai mis un «LBL E» en début de programme pour que vous puissiez le lancer (par la touche E) même s'il ne commence pas en 00 (si vous avez une che. J'ai mis un «LBL E» en début de programme pour que vous puissiez le lancer (par la touche E) même s'il ne commence pas en 00 (si vous avez une fonction avant).

000 LBL	020 00	040 LNx	060 06	080 (100 /
001 E	021 B'	041 LOG	061 1	081 1	101 RCL
002 STO	022 LBL	042)	062 0	082 /	102 00
003 00	023 A'	043 +	063 y^x	083 2	103 =
004 2	024 1	044 (064 (084 8	104 STO
005 0	025 =	045 2	065 RCL	085 8	105 07
006 x><t	026 R/S	046 x	066 02	086 -	106 RCL
007 RCL	027 RST	047 PI	067 -	087 1	107 04
008 00	028 LBL	048 x	068 RCL	088 /	108 +
009 x>=t	029 C'	049 RCL	069 06	089 3	109 RCL
010 C'	030 RCL	050 00	070)	090 7	110 04
011 CP	031 00	051)	071 =	091 3	111 x
012 x=t	032 x	052 LOG	072 STO	092 /	112 RCL
013 A'	033 (053 /	073 04	093 RCL	113 07
014 LBL	034 RCL	054 2	074 (094 00	114 =
015 B'	035 00	055 =	075 1	095)	115 R/S
016 RCL	036 LOG	056 STO	076 /	096 /	116 RCL
017 00	037 -	057 02	077 1	097 RCL	117 06
018 x	038 1	058 INT	078 2	098 00	118 R/S
019 DSZ	039 INV	059 STO	079 +	099)	119 RST

Votre pocket et votre calculatrice ont trouvé leur maître !

- Plus de 10.000 écrans d'informations sur votre pocket ou votre calculatrice !
- Echangez vos trucs et astuces à travers 30 forums thématiques !
- Des milliers de programmes !
- Trucs et astuces, Scoops, Infos,
- Messagerie, Petite annonces,
- Micro-informatique, etc.



3615 CALCULATOR

Le minitel de votre ordinateur de poche
et de votre calculatrice graphique !

Programmes sur TI-85

par Noël BONNET

Calcul de médiane :

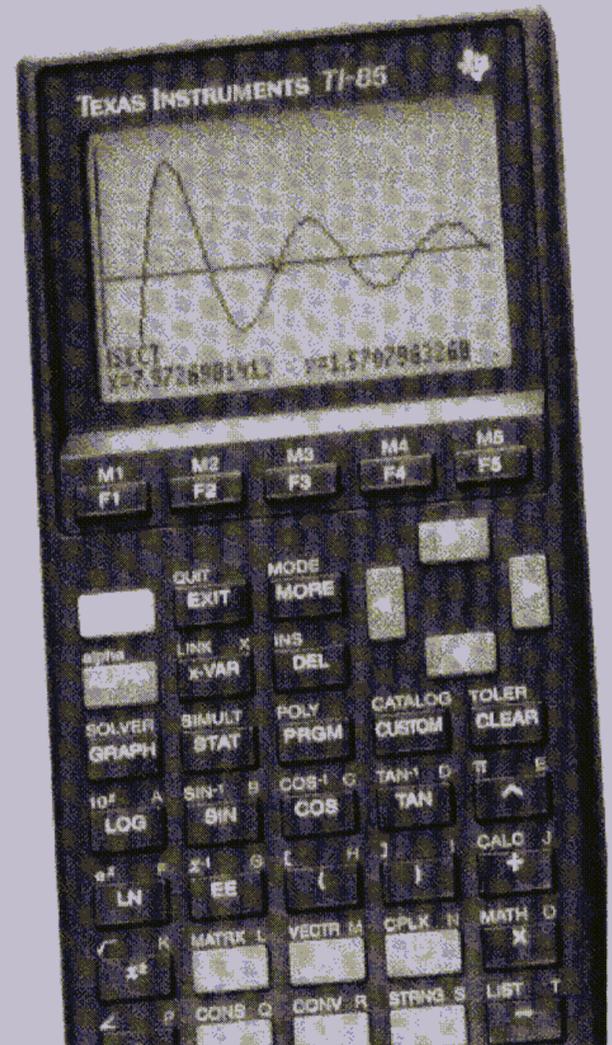
Si l'effectif est impair, il faut faire +1, ce qui permet d'obtenir le rang médian. Ce rang doit être replacé dans les Effectifs Cumulés Croissants (ECC). Il faut donner la valeur de l'ECC d'avant et celui d'après. Il faut donner la classe correspondante à l'ECC d'après (début et fin de classe, pas centre de classe). Les résultats /correspondance que vous obtiendrez devront être écrits tel quel sur la copie. Le "delta x" est l'évolution de "x". Le "x", c'est le "delta x" plus le début de classe. Tous les calculs apparaissent à l'écran, les résultats sont donnés deux fois, une en décimal et une en fraction.

```

PROGRAM : MEDIANE           : Output (2, 16, "-")
                             : Output (2, 17, B
                             : Output (3, 5, J
                             : Output (3, 10, "=>"
                             : Output (3, 15, I
                             : Pause
                             : Output (5, 1, "ecart : rang - deb => ΔX "
                             : Output (6, 1, R
                             : Output (6, 6, "-")
                             : Output (6, 7, D
                             : Output (6, 15, "=> ΔX"
                             : Output (7, 7, M
                             : Output (7, 15, "=>Δ X"
                             : Pause
                             : CILCD
                             : M * I / J → Z
                             : Disp "   ecart classe"
                             : Disp "   * (rang - deb ECC)"
                             : Disp "ΔX= -----"
                             : Disp "   ecart ECC"
                             : Disp "ΔX=", Z, Z@Sfrac
                             : Pause
                             : Z + B → U
                             : Disp "X = ΔX+ deb class", U, U@Sfrac

: CILCD
: Disp "Effectif (impair +1)"
: Input E
: E / 2 → R
: Disp "rang"
: Disp "eff / 2", R
: Pause
: Disp "dans ECC le rang est"
: Input " compris entre", D
: Input "           et", F
: F - D → J
: Input "classe correspondante ", B
: Input " ", C
: C - B → I
: R - D → M
: CILCD
: Output (1, 1, "ecart ECC => ecart classe"
: Output (2, 1, F
: Output (2, 5, "-")
: Output (2, 6, D
: Output (2, 10, "=>"
: Output (2, 12, C

```



Représentation d'un diagramme circulaire :

Il faut rentrer l'effectif total, puis chaque effectif. Il n'y a pas d'unité spécifique. On peut donc se servir de % de degrés ou de tout autre chose. A fin de pouvoir dessiner le diagramme en même temps que le programme, à chaque entrée d'effectif, le diagramme se trace et vous aurez à l'écran la correspondance en degrés pour l'effectif rentré, les degrés cumulés de tous les effectifs (quand on arrive à 360°, c'est la fin) et le cumul des effectifs (quand on arrive à l'effectif total, c'est la fin). Si le cumul des effectifs est supérieur à l'effectif total, il y a une erreur et il faut donc tout recommencer.

```

PROGRAM : CIRCLE           : Circl(0, 0, 8
                             : If D < 360
                             : Then
                             : Lbl A
                             : Input "Eff.classe", E
                             : Goto A
                             : F + E → F
                             : End
                             : F * 360 / T → D
                             : If D > 360
                             : Then
                             : E * 360 / T → C
                             : Disp "degres classe", C
                             : Disp "degres totaux", D
                             : Disp "eff cumule", F
                             : If D = =360
                             : Disp "Ok"
                             : Pause
                             : Line(0, 0, X, Y
                             : Disp G

: Cldrw
: Degree
: Func
: -17 → xMin
: 17 → xMax
: 1 → xScl
: -11 → yMin
: 9 → yMax
: 1 → yscl
: Input "eff. total", T
: 0 → F

```



PROGRAMMEZ VOTRE TI 85 ... PROGRAMMEZ VOTRE TI 85 ... PROGRAMMEZ

Equation d'une droite à partir de deux points :

Ce programme donne l'équation d'une droite de la forme "y = ax + b", à partir de deux points. Les résultats sont donnés deux fois, une en décimal, l'autre en fraction et le détails des calculs apparaît à l'écran.

l'écran.

```

PROGRAM : Droi2P      : Disp "2° Point"           : Disp "A = (y1 - y2) / (x1 - x2)", A,   : Disp "B = y- Ax", B, BsFrac
: Disp "1° Point"     : Input "X", V           ASFrac                               : Pause
: Input "X", X        : Input "Y", R           : Pause                               : Disp "FIN"
: Input "Y", Y        : (Y - R) / (X - V) → A : Y - A * X → B

```

Droite de MAYER :

Ce programme permet de trouver les deux points par lesquelles la droite de MAYER passe. On rentre l'effectif (attention, si il est impair, il faut faire +1 et prendre deux fois la valeur qui est la moitié de l'effectif). Le "F" correspond à la séparation de l'effectif en deux. Il faut prendre les "F" premiers "X" et "Y" pour le premier point et les "F" derniers "X" et "Y" pour le de deuxième point. On peut écrire des additions pour rentrer les ensembles de points "X" et "Y", un cumul est donné juste après. Le détail des calculs apparaît et les résultats apparaissent en décimal et en fraction. Pour trouver l'équation il faut lancer le programme "DROI2P" et prendre les valeurs trouvées dans ce programme.

```

PROGRAM : MAYER      : Pause                               : Input "ensemble y", S
: Input "effectif paire", E : W / F → U                           : Disp S
: E / 2 → F          : Z / F → V                           : Pause
: Disp "F = 1 / 2 ef", F : Disp "1° point M", "X=ens x / f",    : R / F → P
: Pause             : U, U@SFrac, "Y=ens y / f", V,       : S / F → Q
: Disp "1° Point M"  : V@SFrac                             : Disp "2° point P", "X = ens x / F",
: Input "ensemble x", W : Pause                               : P, P@SFrac, "Y = ens y / F", Q,
: Disp W            : Disp "2° point P"                   : Q@SFrac
: Pause             : Input "ensemble x", R               : Pause
: Input "ensemble y", Z : Disp R                               : Disp "FIN"
: Disp Z            : Pause

```

Résolution d'équation du 2° degré :

Il faut donner les valeurs de A, B et C pour le calcul du discriminant. Les solutions sont donnés en décimal et fraction. Si le discriminant n'est pas un carré parfait, il faut reprendre les calculs qui apparaissent à l'écran pour cette raison.

```

PROGRAM: DEG2      : Goto PASOL                       : Goto FIN                               : Disp "x1 = ( - b-
: Disp "AX² + BX + C = 0" : End                               : Lbl SOL                               √D) / 2a", R,
: Input "A", A      : If D = = 0                       : Disp "Δ = 0" ,"1 sol double"         RSFrac
: Input "B", B      : Then                               : - B / (2 * A) → T                     : Pause
: Input "C", C      : Goto SOL                          : Disp "X1 = X2 = -b/2a", T, TsFrac    : (- B + √D) → L
: Disp "Discriminant" : End                               : Pause                                  : L / E → S
: Disp "Δ = b² - 4 ac" : If D > 0                          : Goto FIN                               : Disp "x2 = ( - b
: B² - 4 * A * C → D : Then Sol2                         : Lbl SOL2                              + √D) / 2a", S,
: Disp D            : End                               : Disp "Δ > 0 ", "2 sol"              SSFrac
: Pause             : Lbl PASOL                        : 2 * A → E                             : Pause
: If D < 0          : Disp "Δ < 0", "Pas de sol"       : (- B - √D) → K                         : Lbl FIN
: Then              : Pause                             : K / E → R                             : Disp "Fin"
:                                                            : Disp "{sol}"

```

Résolution d'un système d'équations à deux inconnues :

Il faut entrer les valeurs des variables de la première équation puis de la deuxième. La méthode du discriminant est utilisée. Les détails des calculs sont donnés à l'écran et les résultats sont donnés en décimal et sous forme de fraction.

```

PROGRAM : EQ2IN    : Input "2° x ", B                   : C * H - I * G → E                     : E / D → Y
: Disp "EQ.2 inc"  : Input "2° y ", H                   : Disp E, E@SFrac                       : Disp Y, Y@SFrac
: Disp "(a x + b y = r" : Input "2° r ", I                   : Pause                                  : Pause
: Disp "("         : Disp "Δ = x1 * y2 - x2 * y1"       : Disp "Δy = x1 * r2 - x2 * r1"        : Disp "y = Δy / Δ "
: Disp "(a'x + b'y = r' " : A * H - G * B → D                 : A * I - B * C → F                     : F / D → Z
: Input "1° x ", A : Disp D, D@SFrac                   : Disp F, F@SFrac                       : Disp Z, Z@SFrac
: Input "1° y ", G : Pause                               : Pause                                  : Pause
: Input "1° r ", C : Disp "Δx = r1 * y2 - r2 * y1"     : Disp "x = Δx / Δ"                     : Disp "S = (x, y)"
:                                                            : Disp "Fini"

```

INTERFACE A 32 ENTREES-SORTIES SUR PORT PARALLELE (2ème partie)

(par G.C. et Pierre MARTIN)

Suite aux nombreuses remarques faites par nos lecteurs nous allons revoir en détail la description de l'interface sur port parallèle présentée dans notre précédent numéro.

POURQUOI AVOIR CHOISI D'ECRIRE UN PROGRAMME EN MODULA-2 PLUTOT Q'EN TURBO "XX" !

Le MODULA-2 a été développé par le père de PASCAL, Le Pr NICKLAUS WIRTH, 10 ans apres la création de ce premier. De ce fait c'est un langage encore plus structuré incorporant le concept de module. Sa syntaxe est plus systématique que le PASCAL et l'apprentissage en est facilité. Le MODULA de Fitted Software Tools bénéficie d'une bonne réputation, il est disponible en SHAREWARE, est très proche de la description du Pr WIRTH et possède un assembleur intégré pour les routines de bas niveaux. Son écran d'accueil ressemble à s'y méprendre à celui du Turbo Pascal version 3.0 pour ceux qui ont connu. Par contre son éditeur est multifenêtres (jusqu'à 4), et ses commandes sont configurables pour les habitués du CtrlKD (dont nous faisons partie). De plus ce même éditeur est capable de supporter un écran autre que 80x25 (132x44 par exemple : il est alors nécessaire de choisir le bon mode écran, avec l'utilitaire livré avec votre carte vidéo, avant de lancer l'éditeur).

Il est donc très approprié à la réalisation de programmes pédagogiques et à l'étude de la programmation structurée par l'étudiant désargenté.

Voilà donc de bonnes raisons de vous faire connaître ce langage à travers cette réalisation avant de devenir des inconditionnels forcennés du C++.

Où se procurer le COMPILATEUR MODULA-2 de Fitted Software Tools ?

Tout d'abord le compilateur MODULA 2 FST est disponible en téléchargement sur les serveurs suivants : (nous avons vérifié) par TELETEL 3 (36 15)

RUSTY (V 3.1), FREEWARE (V 2.0a), TEASER (V2.0a et V 3.1), et SM1.

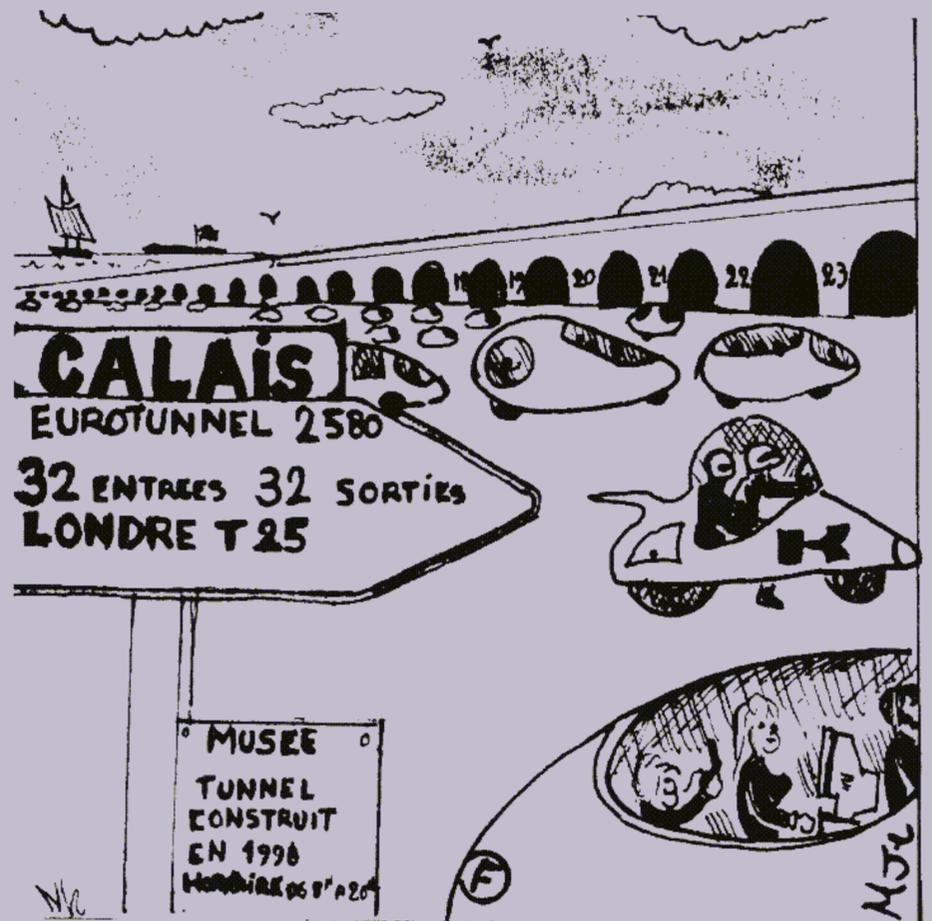
Cette liste n'est pas limitative et d'autres serveurs peuvent le proposer, la publicité faite pour ceux énumérés ci-dessus est complètement désintéressée.

En règle générale le téléchargement s'effectue avec un logiciel spécial pour chaque serveur (gratuit le plus souvent), ce dernier peut être commandé à partir de la rubrique adéquate une fois connecté au serveur.

Le câble nécessaire à l'interface minitel/PC peut être standard quelque soit le serveur. De nombreux détaillants le propose y compris les serveurs eux-mêmes.

Le compilateur MODULA est accessible dans la rubrique " programmation " à partir du menu téléchargement pour PC.

PS1 : A noter toutefois que la version 3.1 de ce compilateur



supporte la programmation orientée objet et que TEASER propose un tutorial pour apprendre pas à pas ce langage (CORONADO SOFTWARE).

PS2 : Il est aussi disponible auprès de son auteur ; FITTED SOFTWARE TOOLS pour 50\$ et dans ce cas vous devenez utilisateur légal, avez accès au source des bibliothèques, et vous êtes tenu informé des mises à jour, ce qui à été notre cas en janvier 93 pour la version 3.1.

FITTED SOFTWARE TOOLS
P.O. Box 867403 Plano, TX 75086
BBS 214/517-4629

1) LE PROGRAMME INTERFACE

En fait il s'agit d'un module de bas niveau réalisant l'interface entre le Hardware (interface registres à décalages / port imprimante) et les niveaux logiciels supérieurs.

L'utilisation des services de l'interruption 17H du BIOS permet de s'affranchir de l'implantation du port imprimante qui n'est pas forcément au standard PC (cas du PORTFOLIO où l'interface parallèle est adressée à partir de 8078H). MODULA-2 impose de créer deux fichiers, un pour le module de définition (IN_OUT.DEF) et le deuxième pour le module d'implémentation (IN_OUT.MOD). Sauf bien sûr pour le module principal MAIN.MOD puisqu'il n'exporte rien vers les autres modules.



INTERFACE A 32 ENTREES-SORTIES SUR PORT PARALLELE ... INTERFACE A 32

2) LISTING DU FICHIER DE DEFINITION "IN_OUT.DEF"

Le module IN_OUT utilise un BITSET (ensemble de bits) plutôt qu'un tableau de 32 booléens ceci dans un but pédagogique.

Les opérations d'inclusion, d'exclusion et d'appartenance sur un élément X de ce type d'ensemble vont nous permettre de manipuler très simplement ce type de données.

3) LE SCHEMA ELECTRONIQUE DE L'INTERFACE 32 Entrées / 32 Sorties :

Le schéma comporte 2 blocs bien distincts :

LE PREMIER est le registre à décalage pour les 32 entrées, ce registre comporte 4 circuits intégrés CMOS du type CD4021 rebouclés circulairement.

Après chargement de ce registre par les états logiques présents sur ses entrées, les bits sont mis en rotation de la droite vers la gauche à chaque coup d'horloge. Le fil PE de la sortie imprimante, raccordé sur la cellule de rang le plus élevé, voit donc défiler un nouveau bit à chaque pas d'horloge.

LE DEUXIEME est le registre à décalage pour les sorties, ce dernier comporte 4 circuits CMOS du type CD4094 montés en cascade. Ce dernier type de circuit comporte en fait un registre de travail pour les décalages et un deuxième registre pour tamponner les sorties afin qu'elles restent dans un état stable pendant les opérations de décalage.

Les bits déjà présents dans le registre de travail sont décalés de la droite vers la gauche alors que le niveau logique présent sur la sortie D2 du port imprimante entre par la cellule de rang le plus faible et ceci à chaque pas d'horloge. L'ordre de transfert vers le registre tampon est émis dès que les 32 bits sont tous entrés dans le registre de travail.

- Le signal STROBE de la sortie imprimante sert d'horloge de décalage série pour ces registres.

- La sortie D2 (bit de données N° 1) permet le chargement des registres d'entrées (CD4021) et le transfert des données du registre à décalage vers le registre tamponnant les sorties (CD4094).

- Les données sont émises en série sur la sortie D1 (bit de données N° 0).

- Les données sont reçues sur le fil PE ou PaperEnd (bit N° 5 du registre de status).

Des réseaux de 8 résistances en boîtier SIL sont prévus sur les entrées afin de les polariser, le point commun de ces réseaux (patte N°1) sera soit relié au 0 Volt soit au +Valim selon le montage de ces composants sur le circuit imprimé. En effet dix trous sont prévus par boîtier SIL à neuf broches, les trous les plus extrêmes sont respectivement le 0 Volt et le +Valim.

Une source d'alimentation extérieure est nécessaire pour alimenter le circuit interface. Elle doit délivrer une tension stabilisée de +5 volts afin que les niveaux logiques de cette interface soient compatibles avec ceux du port imprimante.

Pendant si l'utilisateur final désire alimenter son circuit interface avec une tension de 9, 12 voir 15 Volts pour des problèmes de niveaux logiques avec ses capteurs et ses circuits de commande, il devra toujours utiliser une alimentation stabilisée et prendre la précaution d'insérer entre le port imprimante et la logique CMOS de l'interface 2 circuits convertisseurs de niveaux comme le MC14504 de Motorola. Le premier convertira les signaux TTL du port imprimante : D1, D2 et STROBE en niveaux CMOS 12 Volts (ou autre), le deuxième effectuera exactement l'inverse pour le fil PE.

4) LA FABRICATION DU CIRCUIT ELECTRONIQUE

Ce circuit est dit circuit à double faces.

Nous vous proposons 3 schémas de cette interface, directement exploitables.

Le premier correspond à une face du circuit ; celle où l'on placera les composants.

Le deuxième correspond à une autre face ; celle du dessous, où l'on soudera les composants.

Le troisième correspond à l'implantation des composants ; c'est la face sérigraphie, qui ne sert que d'élément de référence et de positionnement.

Le pas (c'est à dire l'écartement entre les trous) est de 2,54mm soit 1/10 ème de pouce (norme courante).

Voici la manière de procéder.

Faites faire un film positif des 2 schémas (face composants et face soudure), par exemple chez un imprimeur ; ceux-ci souvent possèdent un appareil de reprographie. Ou alors, utilisez les films présensibilisés qui permettent la fabrication de positifs à partir d'une feuille de papier, avec une insolation adaptée (disponible chez de nombreux magasins d'électronique). Enfin, il existe dans de nombreuses villes des magasins d'électronique proposant leur service pour la fabrication de films (voire de circuits).

Si toutefois, vous n'avez pas le temps ou les moyens d'effectuer ces films ou les circuits, nous pouvons vous fournir les circuits epoxy 16/10 ème nus tout prêts à être montés en double face trous métallisés et vernis épargne. Dans ce cas, faites-nous parvenir un chèque de 130F + 30F de port (compter environ 2 semaines pour recevoir le circuit chez vous).

Pour ceux qui sont tentés par l'aventure de la fabrication du circuit, voici comment procéder.

Muni de vos films parfaitement contrastés noir et incolore, vous avez là encore plusieurs possibilités.

Vous pouvez les faire fabriquer par un spécialiste, via un magasin d'électronique. (très onéreux !) Ou bien, certains magasins possèdent des machines à fabriquer les circuits (intéressant pour les gens pressés !). Enfin, vous pouvez les fabriquer vous même (très bonne initiation !). Pour ce faire, il vous faut acquérir une plaque d'epoxy présensibilisée double face, épaisseur 16/10 ème de mm d'environ 20cm par 20. Pourquoi aussi grande nous diriez-vous ? Parceque la première fois, il n'est pas rare de rater ses premiers essais de



INTERFACE 32 ENTREES-SORTIES SUR PORT PARALLELE... INTERFACE 32 ENTR

fabrication ; nous verrons pourquoi par la suite.

Achetez également du perchlore de fer (1 litre) ainsi que quelques bacs assez grands de type Tupperware, avec couvercle. Enfin il faut vous procurer une ampoule (250watts environ) spéciale à insolation (tous ces produits sont disponibles dans les magasins d'électronique et ne coûtent pas très cher).

Le but de la manoeuvre est de graver la représentation des 2 films sur les 2 faces cuivrées de la plaque d'epoxy.

a) l'insolation des faces :

Ceci consiste au moyen de l'ampoule et en plaquant le film (se reporter à la notice fournie avec la plaque ou demander conseil au commerçant), à insoler pour fixer la résine photosensible dont les faces de la plaque sont revêtues. Après insolation, seules les pistes du circuit seront visibles, le reste de la résine est parti. Ici, le temps d'insolation, ainsi que la distance de la lampe par rapport à la plaque sont déterminants (faites des essais avec des petits morceaux de plaque).

b) la gravure au perchlo :

Tout d'abord, attention au perchlo ! ce produit est dangereux, il est très abrasif. Au besoin, protégez-vous les yeux avec des lunettes, une éclaboussure dans l'oeil est vite arrivée. Si cela était le cas, rincez immédiatement à l'eau. Les mains craignent également, portez des gants type chirurgien par exemple. De plus, lorsque votre travail sera terminé, refermez soigneusement la boîte hermétique du perchlo, car celui-ci produit des émanations également dangereuses à fortes doses.

Les 2 faces correctement insolées (il faut que les pistes soient bien visibles et contrastées), rincez abondamment à l'eau courante en frottant très doucement sur la plaque pour éliminer les résidus de résine non désirés. Ensuite, plongez la plaque dans le perchlo, si possible en ayant ménagé une petite surface pour maintenir la plaque avec une pince à linge en plastique (pas de fer !), et remuez doucement. Au bout de quelques dizaines de secondes, vous verrez apparaître les premières traces de gravure, le cuivre est rongé par le perchlo, sauf là où la résine est placée (c'est-à-dire sur vos pistes). Vous pouvez accélérer la gravure en plaçant votre bac dans un autre bac plein d'eau chaude, selon le principe du bain-marie. Attention de retirer au bon moment votre plaque, sinon, gare aux pistes rongées ! Ne la retirez pas trop tôt non plus, car les pistes auront de nombreux court-circuits. Faites des essais avec des petits morceaux et chronométrez. Lorsque votre circuit est terminé, rincez-le à nouveau.

c) Le perçage des trous et la préparation du circuit :

Tout d'abord, essayez de découper le plus correctement votre plaque en forme de rectangle, avec une scie ou autre outil, sans abimer le cuivre.

Percez les trous qui permettront de placer les composants avec une mini perceuse et des mèches spéciales epoxy (chez votre commerçant) de 0,8mm. Vous pouvez éventuellement revêtir la face composant d'un verni épargne en bombe. Vous

auriez pu aussi avant tout chose, étamer à froid votre plaque (bidon d'étain chez votre commerçant). L'avantage de l'étamage est qu'il permet de bien meilleures soudures et évite au cuivre de s'oxyder avec le temps. L'inconvénient est qu'il coûte cher. Après perçage, vous remarquerez que le circuit étant un double face, certains trous servent de passage d'une face à l'autre. Vous pouvez faire communiquer les faces avec les pattes des circuits intégrés à implanter, et compléter les trous vacants avec des queues de résistances par exemple que vous aurez soin de positionner correctement. Il existe aussi dans le commerce, des petits rivets qui permettent ce passage d'une face à l'autre.

d) Le montage :

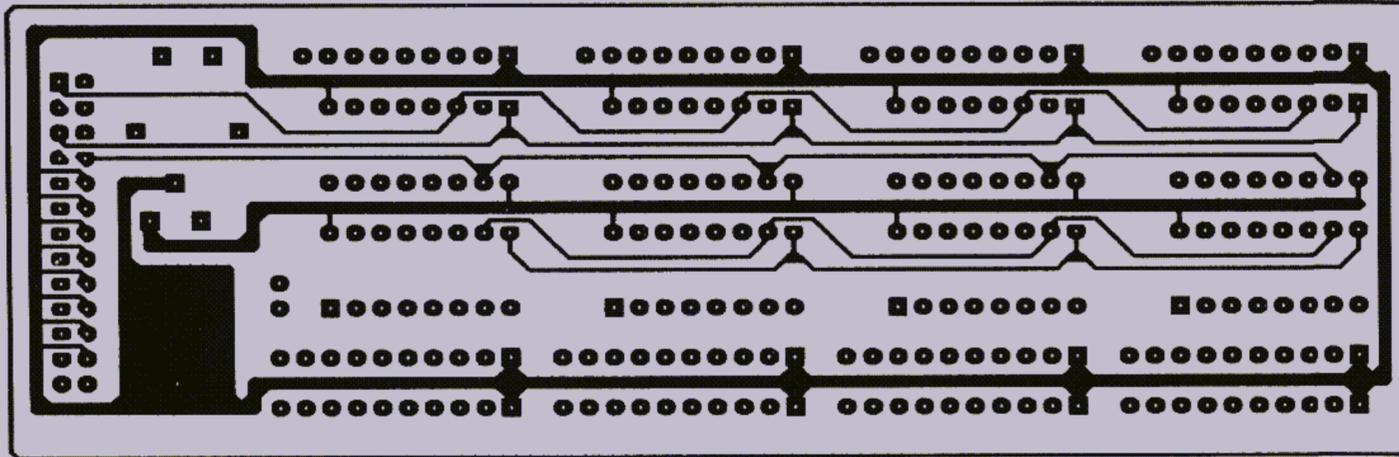
Il vous restera à souder les composants. Ceux-ci se trouvent facilement dans le commerce. Faites attention au sens. Placez devant vous le point gravé sur le composant. La première patte à droite est la patte 1. La dernière du même côté est la masse. La dernière de l'autre côté (patte 14 ou 16 le plus souvent) est le +5volts. Vous devrez aussi acquérir du câble en nappe 25 points, ainsi qu'un connecteur à sertir qui viendra s'enficher dans le port parallèle de votre ordinateur, et un autre à souder pour se connecter sur la carte. 4 réseaux de résistances de 100K ohms 9 points polarisées au + 5 volts viendront compléter ce montage. Enfin, vous remarquerez un régulateur de tension 5 volts (7805). Il permet d'abaisser la tension de 9 à 5 volts dont ont besoin les composants pour fonctionner. Respectez le sens du régulateur et vérifiez que vous obtenez bien 5volts et quelques broutilles et non pas 6 volts par exemple (ce qui voudrait dire que vous avez inversé les fils d'arrivée et de sortie du régulateur) ; la masse est la patte du milieu, l'entrée 9 volts est à droite, la sortie 5 volts, à gauche. N'oubliez pas de fixer un mini radiateur sur le régulateur pour éviter qu'il chauffe trop.

La lettre C près du régulateur indique un pont de soudure permettant d'utiliser la pin 26 située exactement à l'opposé de la pin 1 (inutilisée car il ne faut que 25 pins pour le connecteur parallèle) du connecteur sur la carte, comme source d'alimentation + 5 volts supplémentaire (pour utilisation d'un capteur par exemple).

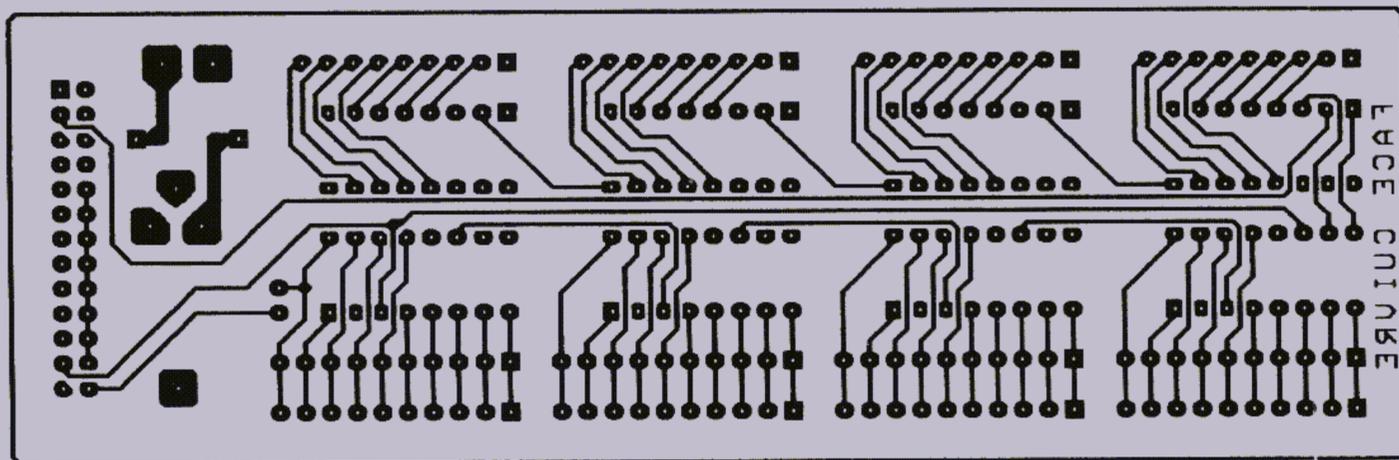
Pour les soudures, utilisez un fer de puissance correcte (environ 40 watts) et non pas sous dimensionnée, et gare aux soudures sèches qui font croire que la patte est soudée alors que ce n'est pas le cas ! Vérifiez toutes vos soudures une à une et si vous possédez un multimètre muni d'un testeur de continuité (indispensable pour travailler en électronique et détecter par exemple les court-circuits : on trouve des appareils de bonne qualité à bas prix maintenant, préférez l'affichage LCD qui facilite la lecture), testez les pistes une à une après chaque composants ; n'ayez pas peur de travailler lentement mais très rationnellement, ceci paie à long terme.

Voilà, bon courage !!! Dans le prochain numéro de POCHÉ COMPUTERS, nous étudierons la réalisation d'un premier capteur, adaptable à cette interface.

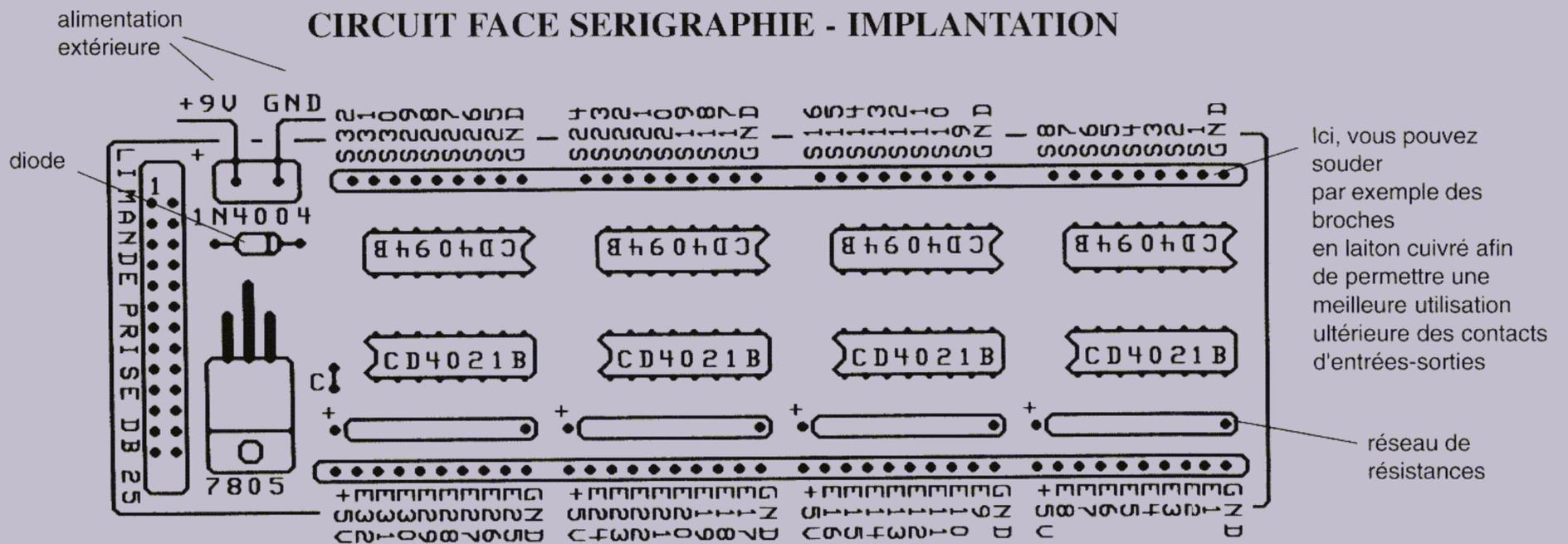
CIRCUIT FACE COMPOSANTS



CIRCUIT FACE SOUDURES



CIRCUIT FACE SERIGRAPHIE - IMPLANTATION



Copyright 1993 POCHE COMPUTERS tous droits réservés.

Il est autorisé d'utiliser ces dessins, de les copier à titre personnel, excepté dans un but d'exploitation commerciale



TRANSFERTS DE DONNEES ENTRE LE PORTFOLIO ET LE REDACTEUR 4.

par François-Pierre LANGLOIS

testeur ayant contribué à la réalisation du REDACTEUR 4, ainsi que du futur correcteur grammatical (sortie fin 93).

I) POUR COMMENCER:

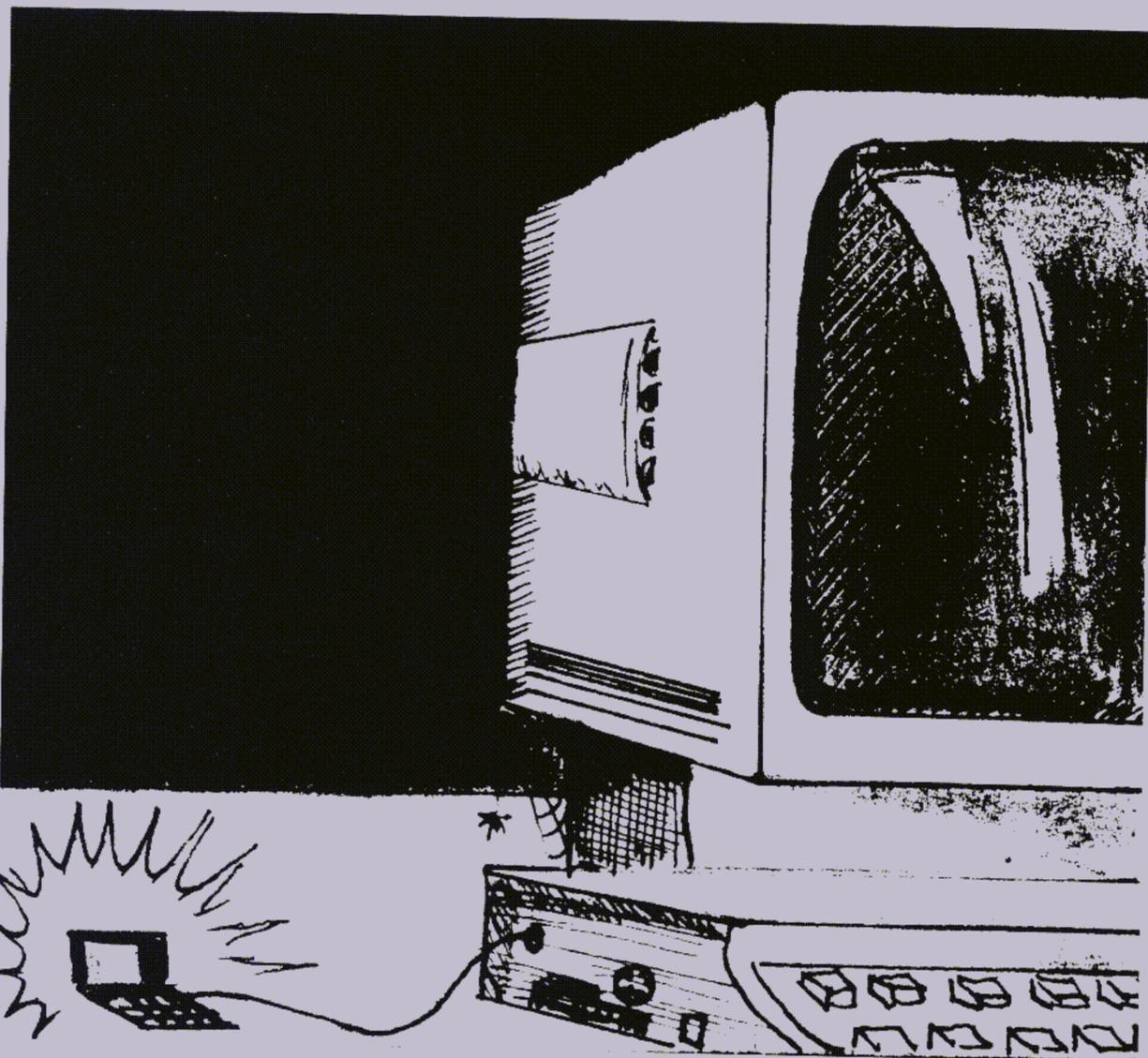
Quand on utilise un outil de bureautique aussi performant que le rédacteur 4, il est bien tentant de faire circuler ses fichiers entre son ordinateur de poche et sa machine de bureau.

C'est ce que je vous propose de faire ici, notamment pour le problème spécifique de la récupération de vos fichiers vers la base de données du Rédacteur 4, plus connue sous le nom d'**AZTHEQUE**. Bien que le transfert de fichier d'un portfolio vers un ordinateur Atari ST (dans les deux sens), soit une manipulation très simple, pourvu que l'on soit équipé du logiciel Foliolink et de son câble de liaison spécifique, il est intéressant de constater qu'il existe un comportement d'utilisation très différent selon que l'on fait ce transfert une seule fois pour la récupération de données d'un ordinateur vers l'autre ou bien qu'il s'agisse de transferts réguliers.

Le deuxième cas de figure étant bien plus conforme à la logique et à la philosophie des ordinateurs de poche.

Dans le premier cas que je décrirai comme étant la voie rapide, il s'agit ni plus ni moins d'interpréter convenablement le mode d'emploi des logiciels (fonction importation et exportation notamment) et des machines. Dans l'autre il faudra tenir compte d'astuces tirées de la pratique des utilisateurs aguerris à ce genre de manipulations.

Le choix du logiciel Foliolink (qui ne peut fonctionner sans son câble bidirectionnel, ce qui est une protection efficace), et de l'interface parallèle du portfolio est dicté par la simplicité et la rapidité de la méthode qui est totalement dans la philosophie du ST et du GEM. Le fait que beaucoup de Portfolio aient été livrés avec une interface parallèle d'origine, valide cette configuration.



ASTUCE POUR LES UTILISATEURS CONNAISSANT DEJA FOLIOLINK :

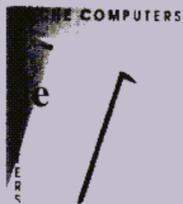
Il est souvent pénible à chaque utilisation du logiciel d'avoir à éditer la ligne du sélecteur qui systématiquement par défaut sur C:*.*, alors que les utilisateurs sérieux du Portfolio auront toutes leurs données sur une CARDAM, autrement dit sur l'unité A.

Pour ne pas avoir à faire cette manipulation à chaque fois, utilisez un simple éditeur de secteur comme Diskdoctor. Puis éditez le fichier FLINKST.PRG. Lancez une recherche sur la chaîne C:\ et remplacez par A:\ et enfin sauvegardez. Le programme fonctionnera dorénavant ainsi modifié (essayez quand même d'abord sur une copie de sauvegarde).

II) LA VOIE RAPIDE : DU FOLIO VERS L'ATARI.

Transférez le fichier sur le disque dur de l'Atari, ce qui se fera sans problème grâce à la simplicité biblique du logiciel FOLIOLINK.

Dans Azthèque créez un fichier en l'ouvrant sous un nouveau nom (ex: NOUVEAU.AZT.) et utilisez la fonction «Importer», c'est tout... Attention quand vous importez un fichier dans Azthèque il vous est demandé de choisir le format d'importation, vous sélectionnez évidemment PORTFOLIO. Mais à ce moment le sélecteur d'objet choisira automatiquement *.DAT comme extension (c'est malheureusement un des trop nombreuses bogues d'Azthèques), il vous suffira simplement de modifier la ligne de sélection par *.ADR.



TRANSFERTS de DONNEES entre le PORTFOLIO et le Rédacteur 4 - TRANSFERTS

DE L'ATARI VERS LE FOLIO :

Il est indispensable de comprendre le minimum en ce qui concerne la structure d'un fichier de données.

Un fichier est composé de fiches, elles-mêmes structurées en champs ou rubriques (c'est la même chose), à l'intérieur desquels vous allez inscrire vos données.

Exemple: FICHE 1

NOM : BEAUPOUMON

PRENOM : Mimy

TEL : 61.69.69.69

FICHE 2

NOM:....

Tous les programmes de gestion de fichiers auront le souci de mettre des séparateurs, des repères en quelques sortes, différents entre les champs et les fiches proprement dites.

Or, pour le Portfolio, le choix de ces séparateurs ont été fait de manière très sommaire. Il s'agit, pour les connaisseurs, d'un CR ou CHR\$ (13) pour les champs, et de deux CR ou deux fois CHR\$ (13) pour les fiches.

Cette disposition rudimentaire vous impose de remplir impérativement tous les champs du fichier AZTHEQUE avec au moins une donnée dans chacun : cette donnée peut être un simple espace si vous n'avez rien à y inscrire. C'est la condition sine qua non pour récupérer votre fichier sans le déstructurer sur le Portfolio. Voir la partie «La méthode de travail cohérente» pour ne jamais oublier ce point indispensable. Si votre fichier AZTHEQUE respecte cet impératif, le reste coule de source : Cliquez la fonction «Exporter». Vérifiez bien que vous sélectionnez la première à la dernière fiche, qu'il y ait un champ et un seul par ligne et que celui-ci soit à gauche du masque d'exportation.

Cliquez sur le format ASCII (et non pas REDACTEUR) et inscrire 1 sur la ru-

brique nombre de lignes entre deux fiches. Pas de trait de séparation. Nommez votre fichier en mettant l'extension .ADR

Transférez le fichier via FOLIOLINK, il est prêt pour la consultation sur votre PORTFOLIO.

III) LA METHODE DE TRAVAIL COHERENTE:

Mon expérience de l'utilisation d'un ordinateur de poche à servir de terminal de saisie et de consultation de données, gérées par ailleurs dans un micro de bureau m'amène à vous faire les suggestions suivantes, ainsi que quelques précisions.

Tout d'abord ne travaillez jamais sur un même fichier se trouvant sur vos deux machines. On a vite fait lors des transferts, d'écraser des données temporaires (une nouvelle adresse, par exemple) inscrites sur un des deux micros. Ma méthode pour la gestion de mes adresses, consiste à avoir un gros fichier complet géré sur le PC de bureau, mais souvent consulté lors de mes déplacements sur mon Portfolio, et un autre plus petit, géré sur le Portfolio, contenant d'une part les adresses des contacts les plus fréquents, et sur lequel je note d'autre part les nouvelles adresses qui seront manuellement inscrites sur le fichier du PC de bureau. Ce petit fichier est pourtant régulièrement transféré vers AZTHEQUE pour consultation.

Pour des raisons analogues, nommez différemment le fichier géré sur une machine (celui qui sert à la saisie) et celui qui sert à la consultation sur l'autre.

Pratiquement, ce qui donne pour le carnet d'adresses, par exemple :
Sur le Portfolio, 2 fichiers :
FOLIO_P.ADR et CARAD_P.ADR
FOLIO_A.AZT et CARAD_A.AZT

DU FOLIO VERS L'ATARI.

Dans la voie rapide, j'ai passé sous si-

lence le fait qu'en s'y prenant comme indiqué, vous récupérez bien vos données, mais AZTHEQUE se conserve le privilège (abusif) d'imposer ses propres noms de rubriques. Il serait agréable de pouvoir conserver la présentation d'un fichier AZTHEQUE, nom et présentation des champs, telle qu'elle aura été définie une fois pour toute.

Pour ce faire :

Ouvrir un fichier sous le nom de CLONE.AZT, importez le fichier FOLIO_P.ADR, sauvegardez.

Après avoir soigneusement préparé la structure d'un fichier sur

Azthèque, sauvez ce fichier. Ex : FOLIO_A.AZT.

Puis utilisez la fonction «charger modèle», (seule la structure du fichier est chargée, mais celui-ci est vide) et chargez le modèle

FOLIO_A.AZT qui aura été défini une bonne fois.

Enfin avec la fonction «ajouter» vous incorporerez les fiches du fichier CLONE.AZT.

Par la suite, vous opérerez toujours de même (gardez la check-list, soit cet article, sous la main).

DE L'ATARI VERS LE FOLIO :

Je vous ai soigneusement caché dans la voie rapide, qu'AZTHEQUE a la détestable habitude d'envoyer un saut de ligne comme première fiche, ce qui aura pour application pratique que la première fiche sera ignoré par le carnet d'adresse du PORTFOLIO.

L'astuce consiste à introduire dans votre fichier AZTHEQUE une fiche bidon qui sera toujours la première quelque soit le mode de tri. Il suffit de remplir tous les champs de cette fiche avec des O.

Enfin, afin de ne pas laissez un vide dans un champ lors d'une saisie d'une fiche AZTHEQUE (ref : la voie rapide), donnez le statut «obligatoire» pour tous les champs. Cela se fait par le choix de l'item «modifier les rubriques» du menu «option».

PORTFOLIO : TRUCS ET ASTUCES

par JL LECOMTE

Suite à de nombreuses demandes de lecteurs non spécialistes en informatique, mais fervents utilisateurs du Portfolio, nous démarrons ici une rubrique pour personnes débutantes ou désirant tout simplement approfondir la connaissance de ce merveilleux petit ordinateur. Vous retrouverez cette rubrique dans les prochains numéros de POCHE COMPUTERS. Nous allons décrire ici quelques trucs et astuces afin d'exploiter au mieux votre appareil et rendre plus facile son exploitation.

1) rappel général :

Le Portfolio est organisé autour d'un microprocesseur de type 8086. La ROM interne (dans laquelle sont contenus le bios, le DOS 2.11 et les différents logiciels intégrés) est de 256K, la mémoire vive est de 128K, extensible à 512K. Il est possible de configurer cette mémoire vive en créant un mini disque dur virtuel (nommé C:) avec l'ordre FDISK. Cet ordre fait partie de tout un jeu d'instructions du DOS dont il est important de connaître les principales ; nous verrons un peu plus loin l'explication de ces ordres.

Mais revenons à notre mémoire vive. L'ordre FDISK doit être suivi d'un chiffre compris entre 8 et 80 dans le cas d'un Portfolio de base, et beaucoup plus (jusqu'à 464) pour un Portfolio de 512K. De nombreux utilisateurs ont quelquefois des problèmes, pour avoir négligé la configuration de la mémoire. En effet, sur les 128K annoncés, le directory (catalogue) et les différents fichiers système ne laissent que 126K environ disponibles. De plus, la création d'un disque virtuel étant obligatoire, l'organisation de la mémoire pourra par exemple se décomposer comme suit :

Soit un FDISK 40 (réservation de 40K octets pour un disque virtuel), nous pourrons lire à l'écran du Portfolio après avoir tapé l'ordre du DOS, CHKDSK : (ordre qui permet d'examiner le contenu de la mémoire).

39168 octets en capacité totale du disque

256 en 1 répertoire

38912 octets disponibles sur disque

126976 mémoire totale

71848 octets libres

Qu'est-ce que cela signifie ? Tout d'abord, précisons que sous C: (c'est à dire sous DOS dans la mémoire du Portfolio), après avoir tapé FDISK 40, il vous sera indiqué que tous les fichiers existants seront perdus ; ceci est normal puisque cet ordre efface complètement la mémoire interne (mais pas la ROM et ses logiciels ! rassurez-vous). Vous répondrez donc "oui". Donc, la première ligne indique la capacité du mini disque dur que vous avez créé. La deuxième ligne indique la place occupée par le directory. La troisième indique naturellement le solde de la mémoire disponible. La quatrième ligne nous renseigne sur la mémoire totale (vous remarquerez qu'elle n'est pas de 128K). Enfin la dernière ligne indique la mémoire restante dite "de travail". Vous pouvez refaire un autre Fdisk et changer la valeur du nombre qui suit cet ordre. Plus vous augmentez ce nombre et plus votre disque dur augmente, mais votre mémoire de travail diminue d'autant. Cependant à quoi sert d'avoir deux types de mémoires me direz-vous ? Eh bien le Portfolio se comporte à ce sujet comme un PC de table ; à savoir qu'il a besoin pour fonctionner d'une certaine quantité de mémoire qui lui sert à **charger** programmes et fichiers ; c'est la mémoire de travail, et qu'il **stocke** les programmes et fichiers dans son mini disque dur. Certains utilisateurs, pour ne pas avoir suffisamment dimensionné la mémoire de travail, se sont retrouvés avec un message d'erreur : "mémoire pleine". D'autre part, si vous la dimensionnez trop grande, vous n'avez plus assez de place pour sauvegarder vos fichiers et programmes. Il faut donc éva-



luer vos besoins (ce qui n'est pas évident au départ), et modifier en conséquence. Par exemple, un gros fichier du carnet d'adresses (environ 40K) peut éventuellement bloquer lors d'un chargement.

Donc, nous pouvons dire que :

MEMOIRE TOTALE = MEMOIRE DE TRAVAIL plus MEMOIRE DE STOCKAGE.

La mémoire de travail est volatile ; c'est à dire qu'elle s'efface après chaque extinction du Portfolio. Par contre, la mémoire de stockage, elle, conserve les données même l'appareil éteint.

2) Les principaux ordres du DOS :

Nous avons vu l'ordre FDISK et l'ordre CHKDSK, mais qu'est-ce que le DOS. C'est un logiciel implanté dans la ROM interne (donc inaltérable) qui permet de créer une interface entre l'homme et l'ordinateur, et d'organiser ses commandes et son travail. Nous avons aussi le BIOS, qui est aussi un logiciel implanté en ROM, mais qui permet de faire la première interface entre l'électronique et les logiciels qui commande l'appareil (par exemple le DOS) ; c'est ce qui permet de **Booter**, c'est à dire de démarrer. Ce Bios, a fait couler beaucoup d'encre et rendu des utilisateurs malheureux. Pourquoi ? parceque les premières versions de ce bios (1.025 et 1.052) comportaient des Bugs ; c'est à dire des erreurs internes et faisaient planter la machine avec souvent des conséquences désastreuses (perte complète de données). Depuis, la version 1.072 est apparue, beaucoup plus fiable et surtout la version 1.130. Vous pouvez vous assurer que vous n'avez pas de version ancienne en tapant au clavier : VER. En haut à droite de l'écran, vous verrez apparaître le numéro.

Donc, voyons quelques ordres importants du DOS.

Le premier, et le plus utile est l'ordre DIR qui signifie aller voir le contenu de la mémoire, du catalogue des fichiers et programmes (nous n'indiquons pas ici qu'il faut valider à chaque ordre du Dos, car cela est implicite). Au sujet de l'organisation de la zone mémoire de stockage, l'erreur que font beaucoup de personnes débutantes, est de placer toutes sortes de fichiers dans une même **racine**. Au bout de quelque temps, il devient très difficile de vous



y retrouver. Des fichiers de nature complètement différente sont mélangés. Nous pouvons comparer l'organisation de la mémoire de stockage comme un arbre généalogique, en partant des grands parents, par exemple, pour arriver aux petits enfants et ainsi de suite. La racine étant les grands-parents, cela correspond à la visualisation que vous faites lorsque vous tapez DIR immédiatement après avoir allumé votre Portfolio. Cependant, il est possible et recommandé, afin de clarification, de compartimenter cette mémoire en **directory ou répertoires** (correspondant par exemple pour reprendre notre image, aux parents). Lorsque vous tapez DIR en Racine, vous voyez apparaître : SYSTEM <REP>. Cela signifie qu'un répertoire (REP) ou directory a été créé (ici automatiquement par le Portfolio), et que vous pouvez le visualiser.

a) créer un répertoire :

Imaginons que vous travailliez avec le carnet d'adresses et avec le tableur. Afin de ne pas mélanger les deux catégories de fichiers que vous allez générer ; xxx.adr pour les adresses et xxx.wks pour le tableur, vous allez créer 2 sous répertoires. Sous C:, tapez l'ordre MD espace ADRESSE. MD veut dire en anglais Make Directory (fabriquer un répertoire). Et puis adresse pour le nom de votre répertoire (prenez le nom que vous voulez). Tapez DIR maintenant, vous voyez sur l'écran ADRESSE <REP>. Faites la même chose avec le répertoire TABLEUR ; vous avez maintenant 3 répertoires : SYSTEM - ADRESSE - TABLEUR. Pour reprendre notre symbolisme de tout à l'heure, nous pouvons considérer que la racine (les grands-parents) ont 3 enfants (les 3 répertoires).

b) opérations concernant les répertoires :

Bien, maintenant, créons un fichier d'adresses, par exemple. Enfoncez la touche rouge ATARI en bas à gauche et, tout en la maintenant enfoncée, pressez la lettre A, puis relâchez immédiatement les deux touches.

Tapez maintenant une adresse quelconque. Puis pressez la touche bleue Fn (en bas à gauche) et simultanément, le chiffre 1. Vous voyez apparaître un menu. Validez sur Fichiers, puis positionnez-vous sur **saue**. Enlevez le nom existant avec la touche flèche, puis tapez ADRESSE\essai et validez. Explications du symbole du trait oblique tourné vers la gauche et séparant ADRESSE de **essai**. Ce signe signifie un chemin d'accès à partir du sous répertoire Adresse. Vous l'obtenez en pressant la touche CTRL+ALT+la

touche au dessus légèrement sur la gauche. Ce trait est à différencier du trait oblique tourné vers la droite. Le mot **essai** indique le nom du fichier (choisissez le nom que vous voulez). Tapez maintenant plusieurs fois sur Echap pour revenir sous DOS en C: Faites DIR. Vous voyez toujours vos 3 sous-répertoires. Mais où est donc passé notre fichier **essai** ? Eh bien, dans le répertoire ADRESSE. Comment le visualiser ? Tapez CD un espace et ADRESSE. Tapez DIR. Que voyez-vous ? :votre fichier **essai** qui a pris automatiquement pour extension **adr**. L'ordre CD permet d'aller voir dans un répertoire. Maintenant, comment revenir à la racine (chez les grands-parents !). Tapez tout simplement CD.. (CD avec 2 points), et faites DIR ; vous y êtes. Vous voyez, c'est très simple. CD.. vous permet de revenir à un niveau **immédiatement** supérieur. En effet, ADRESSE et TABLEUR sont au même niveau, mais rien de nous empêche de créer un **sous-directory** (sous-répertoire). Par exemple, faites CD ADRESSE, (vérifier par DIR que vous êtes bien dans le répertoire ADRESSE), puis faites maintenant MD CLIENTS, puis MD AMIS. Vous venez de créer dans le répertoire ADRESSE, 2 sous répertoires nommés Clients et Amis (par exemple pour distinguer les adresses des clients de celles des amis). Faites CD CLIENTS puis DIR. que voyez-vous ?Aucun fichier. C'est normal puisque vous n'en n'avez pas créé. Si vous lancez le carnet d'adresses maintenant, vos fichiers seront créés directement dans le répertoire ADRESSE (faites l'essai). Par contre, si au moment de sauvegarder, vous spécifiez un second niveau avec le trait oblique \ (antislash) puis le nom clients puis à nouveau \ et un autre nom de fichier, vous sauvegarderez directement dans ce sous répertoire. Ceci semble assez compliqué, mais il n'en n'est rien. Faites beaucoup d'essais et cela ira de mieux en mieux. Il faut bien comprendre que vous travaillez sur plusieurs niveaux. La racine (Grands parents), le directory ADRESSE (les parents) et le sous directory CLIENTS (les enfants). Tapez plusieurs fois Echap pour revenir sous DOS. Question : êtes-vous dans la racine principale ? Tapez CD (tout seul). Que voyez-vous ? Vous voyez C:\adresse\clients (en effet, CD seul vous indique le niveau où vous êtes). Donc vous êtes chez les enfants et non pas chez les grands-parents ! Revenez à la racine par CD.. une première fois et vous êtes chez les parents, une deuxième fois et vous êtes chez les grands-parents. Maintenant, rien ne vous empêche de créer des **sous sous répertoires** ! Attention aux mélanges ...

La suite dans le prochain numéro. A bientôt et bon courage.

TECHNIQUE DE COMMUNICATION PARALLELE PORTFOLIO <-> PC V.106 par M.MANSEL

Avertissement : L'auteur ne demandant aucune rétribution pour ce logiciel, il n'en garantit pas le bon fonctionnement même si une utilisation prolongée ne lui en donne que des satisfactions. L'utilisateur est donc responsable de l'usage du logiciel et de ses conséquences.

1) POURQUOI CP ? (communication parallèle)

L'utilitaire FT.COM fourni par ATARI ne permet pas d'utiliser les commandes DOS pour gérer le disque A: du Portfolio. L'objectif de CP était de simuler un disque local et supplémentaire sur le PC hôte, disque qui serait celui du Portfolio.

2) COMMENT FAIRE ?

2.1) Le matériel :

La liaison centronics est unidirectionnelle... pour les données, même si elle était prévue au départ pour fonctionner dans les 2 sens (sinon pourquoi 36 broches sur ce connecteur initial ?). Mais l'imprimante remonte des signaux d'états que le PC peut relire (absence papier, imprimante éteinte ou occupée).

Croisons quelques fils et la communication devient possible:

data0	2 - 15	Error	Busy	11 - 6	data4
data1	3 - 13	Select	Paper	12 - 4	data2
data2	4 - 12	Paper	Select	13 - 3	data1
data3	5 - 10	Ack	Error	15 - 2	data0
data4	6 - 11	Busy	gnd	25 - 25	gnd
Ack	10 - 5	data3	PC#1		PC#2

2.2) Le logiciel :

Une ligne est commune : la 25, c'est la masse électrique des signaux 2 zones sont symétriques et correspondent au croisement des signaux ; ce qui sort du PC#1 entre dans le PC#2.

Ces zones comprennent 5 fils donc 5 signaux maximum (un peu



Le PORTFOLIO : TRUCS et ASTUCES... Le PORTFOLIO : TRUCS et

plus qu'un quartet, ce qui nous ramène au bon vieux temps de l'octal). Comment écrire ? Simple, en faisant un OUT sur le port imprimante voulu, mais un OUT sur simplement 5 bits. Comment lire ? Simple, en testant l'état de l'<imprimante> ou plutôt du port d'imprimante voulu par INT17h, on voit que les 5 bits de poids faibles envoyés sont reçus comme 5 bits de poids fort dans le mot d'état, avec une complémentation pour les 2 plus forts XOR 11000000b. Mais ce ne sont pas des octets et il n'y a aucun signal de synchronisation. C'est au logiciel de la faire !

J'ai arbitrairement choisi le bit 0 (ou d0). Le principe est simple car le changement d'état de ce bit indique qu'un quartet a été transmis et le récepteur acquitte par une opération identique.

PRINCIPE 1 : Les octets sont transmis en 2 fois par quartets

PRINCIPE 2 : Une synchronisation indique quel octet est transmis cela fait 5 bits

-synchro à 0 pour quartet de poids faible

-synchro à 1 pour quartet de poids fort

CONCLUSION 1 : Au début d'un dialogue, il faut attendre que la synchro passe à 1

CONCLUSION 2 : Si en cours de dialogue, on attend un nouvel octet et que la synchro est à 0 il y a problème.

Idem si la synchro est à 1 entre 2 quartets d'un même octet

SCENARIO :

INITIALISATION

attente synchro à 1

si délai dépassé == ERREUR

échange d'octets pré-déterminés

ECHANGE

si synchro = 0 erreur

réception quartet P faible

si délai dépassé == ACTION selon contexte

si synchro = 1 erreur

réception quartet P fort

si délai dépassé == ERREUR

reconstitution octet

REPETITION ECHANGE

Pour alléger le logiciel et rendre la communication par cette voie accessible à l'utilisateur, j'ai ajouté l'interruption 62h inutilisée par DOS.

3) INTERRUPTION 62H

Il y a 4 fonctions dans cette IT

Initialisation de la communication

ENTREE AH=0

SORTIE AH=0

MODIFIES /

Ecriture d'un caractère

ENTREE AH=1 AL=caractère

SORTIE AH = 0:Ok

1:Récepteur pas prêt

2:premier quartet non acquitté

3:second quartet non acquitté

MODIFIES AX

Lecture d'un caractère sans attente

ENTREE AH=2

SORTIE AH = 255:Ok AL=car

0:Rien à lire

1:Second quartet pas reçu

MODIFIES AX

Acquisition de la synchro de l'émetteur

ENTREE AH=3

SORTIE AL = 0:synchro à 0

1:synchro à 1

MODIFIES AX

Le fichier ECP_TEST.ASMF illustre l'emploi de cette IT.

4) CONTENU DE LA DISQUETTE ET UTILISATION

(pour l'obtenir envoyez 20F en timbres plus une disquette au magazine)

La disquette doit contenir

cp_lire.moi	ce fichier
cp_inst.com	exécutable installation de l'IT62h
cp_driv.sys	driver de périphérique pour le PC
cp_port.com	exécutable à lancer pour le Portfolio
cp_test.com	pour les tests avant de lancer le pilote
cp_test.asm	source du précédant pour exemple

Commencer par confectionner le câble décrit précédemment en utilisant un connecteur mâle et un connecteur femelle, ce qui permet de l'insérer dans la liaison PC/Portfolio existante.

Copier par FT ECP_????.COMF sur le Portfolio. Lancer ECP_INSTF sur le PC et le Portfolio. Si l'IT est déjà occupée par un de vos pilotes ou utilitaire, c'est signalé. A vous de l'ôter...

Lancer ECP_TESTF sur les 2 machines : Ce que vous tapez sur un clavier doit apparaître sur les 2 écrans. Sinon, revoyez vos soudures ! Sortie par ESCape. Vous pouvez également lancer ECP_TESTF sur Portfolio et par exemple DIR | CP_TEST sur le PC. Les 2 écrans doivent afficher le contenu du répertoire courant du PC. Sortie par ^C sur le PC et ESCape sur le Portfolio. Si et seulement si tous ces tests sont positifs, modifier CONFIG.SYS du PC pour ajouter DEVICE=ECP_DRIV.SYSF en copiant ce dernier sous la racine du disque du PC. Lancer ECP_PORTF sur le Portfolio et faire un «BOOT» du PC. Le driver doit annoncer qu'il a installé le disque du Portfolio sur D: (ou plus si D: est déjà pris). Supposons qu'il annonce D : faire

DIR D:

TYPE D:... un fichier

COPY D:... C:

et pour finir

copy C:... D:

Sur le Portfolio l'appui sur n'importe quelle touche provoque le retour sous DOS

Tout le logiciel gère des time-out et le Portfolio affiche la nature des opérations en cours «l»ecture ou «e»criture ainsi que le status qui doit rester à 00.

SI TOUT S'EST DERoule NORMALEMENT, LE PC VOIT LE DISQUE DU PORTFOLIO COMME UN DES SIENS TOUTES LES COMMANDES DOS, TOUS LES UTILITAIRES FONCTIONNENT SUR LE DISQUE DU PORTFOLIO

5) REPOSES A QUELQUES QUESTIONS :

*ça ne marche pas sur mon PC ?

Il se peut que des problèmes de synchronisation apparaissent avec des PC rapides. Tout a été développé avec un clône AT 286 à 12MHz

*Comment démarrer le PC si le Portfolio n'est pas connecté ? Pas, de problème, un time-out autorisera le PC à continuer son BOOT. Attention, l'IT62h ne sera pas installée.

*Est-ce limité à un dialogue entre PC et Portfolio ?

A priori non, cela doit fonctionner entre 2 PC mais je n'ai fait aucun test. Par contre, ne pas tenter d'installer ECP_DRIVF sur le Portfolio, il est «un peu court». Plus tard peut-être...

*Y aura-t'il d'autres versions de ECPF ?

Oui, si les améliorations demandées sont faisables et justifiées.



TURBO-FORTH pour PORTFOLIO

par REM CORP.

Un ordinateur dans sa poche et ne pas pouvoir programmer, quelle déception. Il y a bien le Basic, le C ou le Pascal, mais ceux-ci sont souvent très gourmands en mémoire.

Avec FORTH, véritable langage de programmation, apparu dans le monde de l'astronomie et vulgarisé par les calculatrices HP, vous disposez d'un outil efficace et compact. Rustique mais simple, le langage FORTH nécessite rigueur, mais apporte de grandes satisfactions à celui qui s'y investit. Loin d'être exotique, FORTH est utilisé dans des applications majeures:

- logiciel VP-PLANNER, un clône de LOTUS 123
- logiciel IBM-Cadd, concurrent direct d'Autocad
- dans le CDI de Philipps
- dans le télescope spatial Hubble
- pour le contrôle d'automates sur chaînes de production chez RENAULT, PEUGEOT, CITROEN...

TURBO-Forth en est une version française. La version spécifiquement adaptée au PORTFOLIO d'ATARI offre même des possibilités étonnantes.

UN INTERPRETEUR

TURBO-Forth pour PORTFOLIO D'ATARI offre la souplesse d'un interpréteur. Toute commande tapée au clavier est directement exécutée. Il reconnaît près d'une centaine de commandes. Sa syntaxe, de type RPN (pour Reverse Polish Notation) est identique à celle du langage PostScript. Si ce dernier est un langage de description de page, alors FORTH est un langage de description d'applications. Exemple de quelques commandes simples:

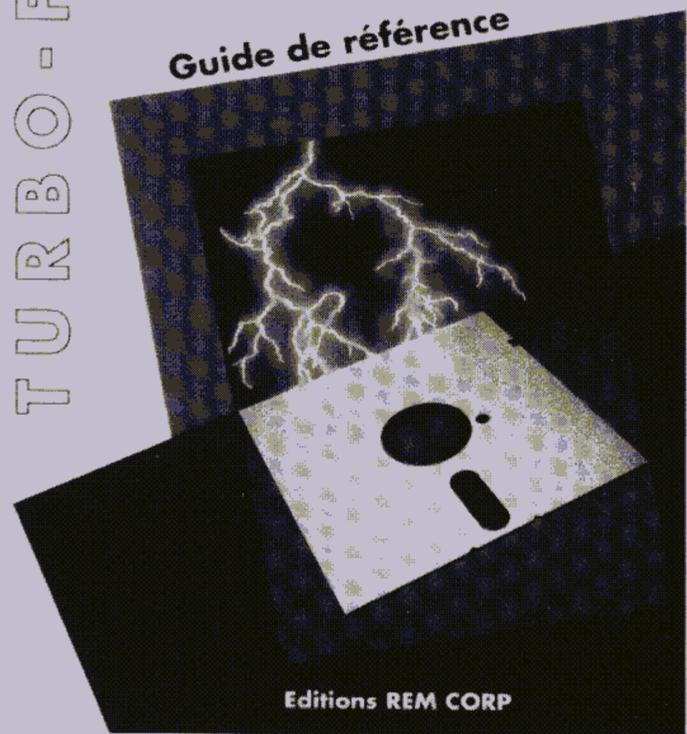
- DARK efface l'écran
- DIR affiche le contenu du répertoire
- LIST <fichier> liste le contenu d'un fichier ASCII
- adr len DUMP exécute un DUMP mémoire etc...

UN COMPILATEUR

TURBO-Forth, c'est aussi un compilateur simple. Si simple qu'il compile en une seule passe, sans linker et sans générer de fichiers .OBJ. Il convertit directement un fichier ASCII en programme exécutable. Exemple:

- : BOUCLE
- 10000 DO LOOP ;
- \équivalent de

TURBO-FORTH



\ FOR N=0 TO 10000:NEXT

\ du langage BASIC

Si petit que soit le fichier source, il sera directement compilé. Sans directive particulière, le dictionnaire FORTH sera complété par tous les noms de procédures définis par le programmeur. Ainsi, pour exécuter le mot BOUCLE défini ci-avant, il suffit ensuite de taper BOUCLE depuis l'interpréteur. Une telle procédure s'exécutera en trois dixièmes de seconde.

Outre la rapidité, un programme FORTH est extrêmement compact. Si compact, qu'il est capable de tenir dans un espace mémoire plus restreint que son

équivalent écrit en assembleur. Cette particularité est possible grâce à son moteur interne tout à fait particulier.

UN MOTEUR MULTI-TACHE

TRUBO-Forth pour PORTFOLIO est rapide et compact. Mais il est aussi multi-tâches sous DOS. Sans programme résident, sans driver spécifique, sans patch logiciel, le moteur interne de TURBO-



TURBO-FORTH pour PORTFOLIO... TURBO-FORTH pour PORTFOLIO

Forth est capable d'exécuter simultanément deux ou plusieurs programmes Forth différents. Le programmeur peut parfaitement avoir le contrôle de l'interpréteur pendant qu'une tâche de fond surveille un processus spécifique (voie série, voie parallèle, etc...).

Le moteur multi-tâche de TURBO-Forth est de type Round-Robin, c'est à dire similaire à celui utilisé dans un réseau à jeton. Toute tâche qui a le jeton s'exécute jusqu'à ce qu'elle passe le jeton à la tâche suivante. Dans certains processus, ce système est plus efficace que le multi-tâche en temps partagé de Windows ou d'UNIX.

DES OUTILS DE DEVELOPPEMENT

Il va sans dire que TURBO-Forth est également très performant en mode mono-tâche. Il dispose d'une palette d'outils de développement et de mise au point particulièrement sophistiqués :

- un débogueur traceur. Cet outil permet une mise au point en mode pas à pas ou continu de toute définition Forth, avec possibilité d'accès à l'interpréteur, reprise du traçage, abandon du traçage.

- des fonctions DUMP et LDUMP pour permettre d'inspecter le contenu de la mémoire intra et extra-segment.

- une fonction SEE pour décompiler et désassembler les programmes écrits en Forth ou en assembleur. Cette fonction est complétée par une sous-fonction IDIS capable de désassembler toute routine d'interruption.

- un assembleur en notation RPN disposant de structures de contrôle, possibilité de création de macro-instructions. Avec l'assembleur, toute fonction est immédiatement utilisable par l'interpréteur. Exemple:

```
CODE 3DUP ( n — n n n )
```

```
ax pop
```

```
ax push ax push ax push
```

```
NEXT END-CODE
```

La nouvelle fonction 3DUP est utilisable directement depuis l'interpréteur.

UN LANGAGE DIDACTIQUE

TURBO-Forth est livré avec deux manuels :

MANUEL D'APPRENTISSAGE et GUIDE DE REFERENCE.

La simplicité du langage FORTH permet de maîtriser et comprendre les mécanismes internes du PC. Il est particulièrement adapté au développement d'applications dites 'temps-réel', c'est à dire nécessitant performance et compacité.

Mais le langage FORTH est également utilisable dans une grande variété d'applications:

- graphisme,
- intelligence artificielle,
- télécommunications,
- automation, robotique
- mathématiques et physique (cf Julian V. NOBLE: SCIENTIFIC FORTH, ed Mechum Banks)

TURBO-Forth est livré avec ses sources. TURBO-Forth est écrit en TURBO-Forth et s'auto-compile à l'aide d'un méta-compileur. Si le programmeur dispose d'un PC classique, il peut créer sans difficultés une version personnalisée de TURBO-Forth.

TURBO-FORTH ET L'ENVIRONNEMENT DU PORTFOLIO

TURBO-Forth compile des fichiers source au format ASCII. Ces fichiers sources peuvent être édités avec l'éditeur intégré du Portfolio. Ils peuvent également provenir de fichiers utilisés sur PC.

TURBO-Forth PORTFOLIO n'utilise aucune interruption spécifique du PORTFOLIO. Ses applications sont donc portables vers tout environnement de type PC (version DOS 2.0 et suivants, équipés du processeur 8086 à 80486 et PENTIUM, NEC V20/V30) ou tout type de pocket compatible PC.

TURBO-Forth PORTFOLIO est livré avec divers fichiers source d'exemples dont des fonctions graphiques. Ces fonctions permettent de tracer des points et segments de droite dans la limite des capacités graphiques du PORTFOLIO. Des fonctions complémentaires permettent de mélanger texte et graphisme sous Forth.

TURBO-Forth:

- Taille mémoire requise: 32 Ko. Editeur MP7

**TURBO-FORTH est disponible chez MICROCARDS
au prix de 795F ttc sur disquette avec manuels.**



TRANSFERT SHARP PC 3100 <> PC

par Didier CESSSES

Lettre d'un utilisateur à propos du SHARP PC3100.

Tout d'abord pour ce qui est de l'ergonomie générale de l'appareil : rien que du bon ! le clavier, l'écran, le look et l'encombrement sont impeccables. Pour la prise en main, le programme install permet une configuration correcte. (langue, date, clavier et autres options) Cependant les fichiers CONFIG.SYS et AUTOEXEC.BAT sont dans le disque virtuel D: ce qui permet de rajouter ce qu'on veut dedans comme device driver, path) ... (voir plus loin sur DD.BIN), le disque c: étant protégé en écriture (ROM oblige). Ensuite quand on met en marche... on est très vite confronté au problème des transmissions. J'avais avant le PC3100 un Portfolio d'ATARI qui communiquait via la prise parallèle avec mon PC sans aucun problème.

Le sharp contient en ROM dans la directory \UTILS du disque c: deux utilitaires de communications de la famille LAPLINK. Laplink lui même et un autre DD (?) (trois fichiers DD.EXE DD.BIB et DDINSTAL.EXE) qui permet de prendre le contrôle des disques d'un autre PC...

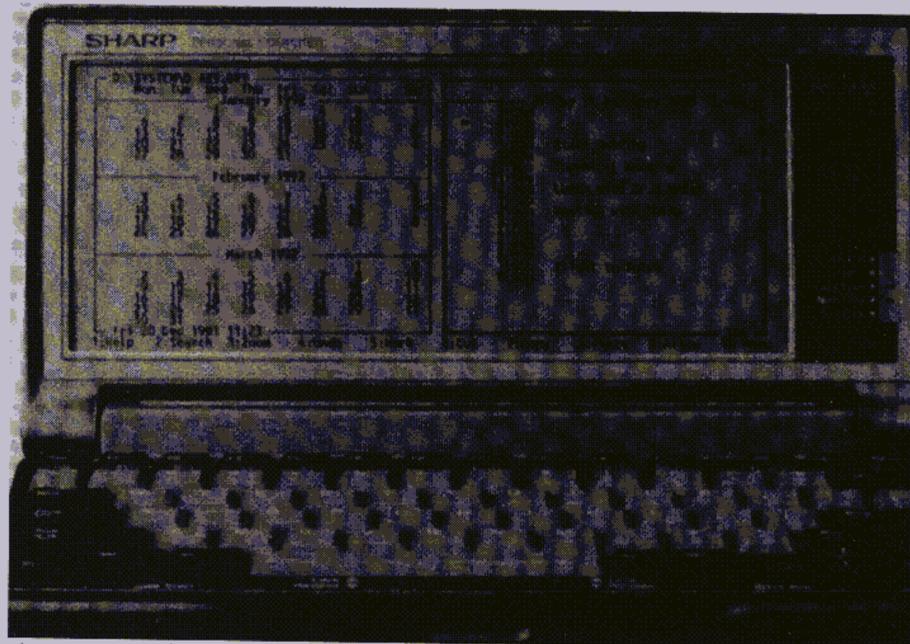
Le premier problème à résoudre est celui de la connexion. Le câble Laplink n'est pas standard, il y a des ponts et des fils croisés. Donc deux solutions soit acheter, en plus des connections Sharp obligatoires, un câble LL, ou alors acheter des prises séries 9 et 25 femelles du câble, un plan et de quoi faire des soudures. Le Laplink fournit en ROM du Sharp contient une procédure d'auto-installation sur l'ordinateur serveur. (grâce à la touche F10... je n'ai pas réussi à le faire fonctionner, j'ai eu le message «l'ordinateur ne répond pas» ????) pour ma part j'ai ressorti une vieille version de LL (V2.6a) que j'ai installée sur mon 486, la configuration ne prend que quelques minutes. Attention les versions plus récentes ne fonctionnent pas avec le même câble.

Et ô merveilles des merveilles j'ai pu recharger des fichiers à partir de l'ordinateur de bureau sans aucun ennui.

Deux remarques cependant:

-1- les communications coûtent très cher en énergie, j'ai mis mes accus à genoux à plusieurs reprises en transférant différentes config de works dos (version 2 qui fonctionne très bien au demeurant) et de Finesse, petit logiciel graphique de PAO sous GEM qui tourne (avec souris) sans aucun problème. En fait, il faut bien regarder de positionner l'option faible consommation activée dans le menu Pop Up, et alors, la consommation est beaucoup moindre.

-2- Les fichiers que j'avais sur l'ATARI Port Folio sont complètement compatibles avec les Intégrés du Sharp; les fichiers textes c'est l'évidence mais aussi les feuilles de calcul et surtout les fichiers ADRESSE (Bases de donnée) qui répondent à la même logique d'enregistrement séparés uniquement par une ligne vide (CR+LF pour les initiés). j'ai donc passé mes fichiers adresses directement des sauvegardes du PF sur le Sharp via le PC.



Autre partie importante:

Le fichier DD.exe. qui permet de se servir des disques d'un autre ordinateur (le serveur).

Là aussi il faut le câble, et il faut Laplink qui marche pour transférer le trois fichiers DD.EXE DD.BIN et DDINSTAL.EXE sur le PC serveur.

ATTENTION il y a pour certains drivers de souris des incompatibilités importantes avec DD.BIN, qui peuvent aller au plantage complet. Donc si souris sur le sharp => analysez bien le driver. J'ai bricolé un bon moment pour réussir à faire marcher tout ça. C'est un peu sioux mais à terme on vient assez aisément à bout des différentes embûches qu'on rencontre.

Première étape: Il faut recopier les trois DD*.* sur le disque E: et tromper l'écriture avec une commande ASSIGN C=D pour permettre les modifs du config.sys. Ensuite il faut lancer le programme DDINSTAL on obtient l'écran suivant:

```
LapLink
Device Driver (DD.BIN) installation program
Version 3.00a Copyright (C) 1987-89 Traveling Software, Inc.
Device Window hotkey: ALT-D
Printer: Lead Server
Serial port: COM1 COM2
DD.BIN path: \
Baud: 115200
Server Sector Size: 512bytes
COMMANDS: Help Save Restore Uninstall Quit
```

pour le sharp c'est COM1, le path (le lieu où on trouve) du fichier DD.BIB c'est D:\.

Il faut ensuite faire la même opération pour le serveur en veillant bien à ce que les tailles de secteurs (Serveur sector Size) soient identiques.

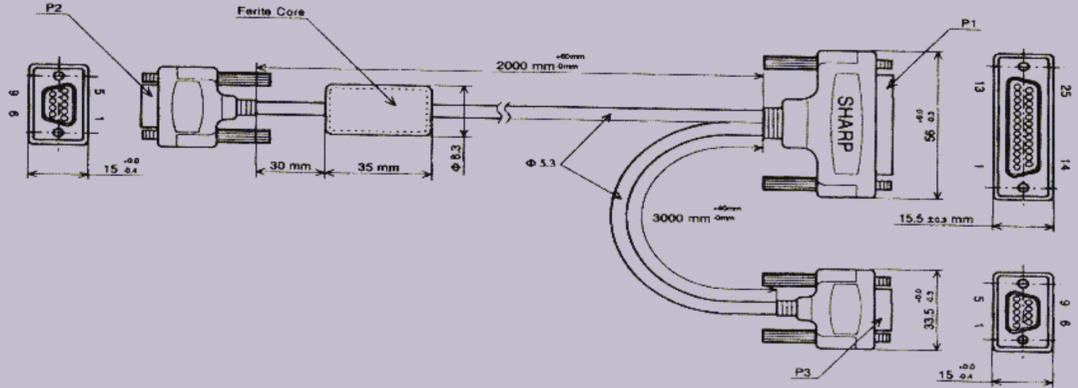
Ensuite: Il faut lancer DD sur le serveur et taper ALT+D sur le Sharp on voit alors la config s'afficher qui vous dit que le disque A: du serveur correspond au disque F: sur le Sharp que le disque B: du serveur c'est le G: du sharp et ainsi de suite. Après <ESC> on peut accéder aux disques du serveur en tapant F: par exemple pour lire ou écrire sur le disque A: du PC serveur.

En résumé, le bilan est de toute façon nettement positif. Quel fabuleux ordinateur que le Sharp PC 3100 !



Transfert PC <-> SHARP PC 3100... Transfert PC <-> SHARP PC 3100

A la demande de nombreux lecteurs, nous publions ici le câble Lap link, ainsi que la nomenclature des différents connecteurs du SHARP PC3100. En espérant que cela donnera des idées d'électronique à quelques personnes éclairées.



- Materials:**
- Cables: 7 Conductors, 26AWG, UL2464, Double Shield, Grey Color, Circle Diameter $\phi 5.3\text{mm}$ - BFL
 - "D" Connector DB-25S-10 1pcs
 - DB-09s-10 2pcs
 - Phosphor Bronze with 10u Gold Plate on contact area.
 - Thumber Screw: Grey Color - 6pcs.
 - Aluminium Label with "SHARP" logo - 1pcs.
 - Copper Foil (For shielding) - 6pcs.

PIN CONNECTION TABLE

P2	P1	P3
DB-09S-10	DB-25S-10	DB-09S-10
PIN 2	PIN 2	PIN 3
PIN 3	PIN 3	PIN 2
PIN 8	PIN 4	PIN 7
PIN 7	PIN 5	PIN 8
PIN 4	PIN 6	PIN 6
PIN 5	PIN 7	PIN 5
PIN 6	PIN 20	PIN 4
SHIELD	SHIELD	SHIELD

Vous remarquerez le bus d'extension à 80 pins et les 2 connecteurs pour cartes PCMCIA à 68 pins.

CN1

PIN	SIGNAL	PIN	SIGNAL
1	GND	41	GND
2	PACIN	42	PACIN
3	PACIN	43	PACIN
4	N.	44	_EXPPE
5	N.	45	_EXPBV2
6	N.	46	_EXPBV1
7		47	KBC
8	EXPDET	48	KBD
9	GND	49	PACOUT
10	EA0	50	PACOUT
11	EA1	51	PACOUT
12	EA2	52	
13	EA3	53	ALE
14	EA4	54	TC
15	EA5	55	DACK2
16	EA6	56	IRQ3
17	EA7	57	IRQ4
18	EA8	58	IRQ5
19	EA9	59	IRQ6
20	EA10	60	IRQ7
21	EA11	61	ECLK
22	EA12	62	DACK0
23	EA13	63	DREQ1
24	EA14	64	DACK1
25	EA15	65	DREQ3
26	EA16	66	DACK3
27	EA17	67	_JR
28	EA18	68	_JW
29	EA19	69	_MR
30	AEN	70	_MW
31	ERDY	71	TP1
32	ED0	72	TP2
33	ED1	73	TP3
34	ED2	74	TP4
35	ED3	75	DREQ2
36	ED4	76	GND
37	ED5	77	IRQ2
38	ED6	78	GND
39	ED7	79	RESET
40	_IOC	80	GND

CN2 KEY

PIN	SIGNAL	PIN	SIGNAL
1	KS0	6	KS5
2	KS1	7	KS6
3	KS2	8	KS7
4	KS3	9	KS8
5	KS4	10	KS9

CN3 KEY

PIN	SIGNAL	PIN	SIGNAL
1	KI0	6	KI5
2	KI1	7	KI6
3	KI2	8	KI7
4	KI3	9	KI8
5	KI4	10	KI9

CN4 SPEAKER

PIN	SIGNAL	PIN	SIGNAL
1	SOUND	2	GND

CN5 STATIC LCD

PIN	SIGNAL	PIN	SIGNAL
1	SPCK	4	CAP LOCK
2	CARD A	5	NUM LOCK
3	CARD B	6	SCL LOCK

CN6 AC-ADAPTER

PIN	SIGNAL	PIN	SIGNAL
1	ACD	3	ACG2
2	ASG1		

CN7 SERIAL PORT

PIN	SIGNAL	PIN	SIGNAL
1	I8	7	I3
2	I7	8	O2
3	O6	9	I1
4	O5	10	VCC
5	GND	11	GND
6	I4	12	GND

CN8 LCD

PIN	SIGNAL	PIN	SIGNAL
1	LF	7	_VEE
2	CP1	8	LCD0
3	GND	9	LCD1
4	CP2	10	LCD2
5	GND	11	LCD3
6	VDD	12	DF

CN9 PARALLEL PORT

PIN	SIGNAL	PIN	SIGNAL
1	STB	11	BUSY
2	PD0	12	PE
3	PD1	13	SLCT
4	PD2	14	AFD
5	PD3	15	ERROR
6	PD4	16	INIT
7	PD5	17	SLIN
8	PD6	18	GND
9	PD7	19	GND
10	ACK	20	GND

CN10

PIN	SIGNAL	PIN	SIGNAL
1	GND	35	GND
2	JD3	36	JBCD1
3	JD4	37	N.C
4	JD5	38	N.C
5	JD6	39	N.C
6	JD7	40	N.C
7	_JCEO1	41	N.C
8	JA10	42	N.C
9	_JOE	43	N.C
10	JA11	44	N.C
11	JA9	45	N.C
12	JA8	46	JB17
13	JA13	47	JB18
14	JA14	48	JB19
15	_JWE	49	JB20
16	N.C	50	JB21
17	JBVCC	51	JBVCC
18	JBVPP	52	JBVPP
19	JA16	53	JB22
20	JA15	54	JB23
21	JA12	55	JB24
22	JA7	56	JB25
23	JA6	57	N.C
24	JA5	58	N.C
25	JA4	59	N.C
26	JA3	60	N.C
27	JA2	61	_JRGO
28	JA1	62	JBV
29	JA0	63	JBV
30	JD0	64	N.C
31	JD1	65	N.C
32	JD2	66	N.C
33	JBWP	67	JBCD2
34	GND	68	GND

CN11

PIN	SIGNAL	PIN	SIGNAL
1	GND	35	GND
2	JD3	36	JACD1
3	JD4	37	N.C
4	JD5	38	N.C
5	JD6	39	N.C
6	JD7	40	N.C
7	_JCEO1	41	N.C
8	JA10	42	N.C
9	_JOE	43	N.C
10	JA11	44	N.C
11	JA9	45	N.C
12	JA8	46	JA17
13	JA13	47	JA18
14	JA14	48	JA19
15	_JWE	49	JA20
16	N.C	50	JA21
17	JAVCC	51	JAVCC
18	JAVPP	52	JAVPP
19	JA16	53	JA22
20	JA15	54	JA23
21	JA12	55	JA24
22	JA7	56	JA25
23	JA6	57	N.C
24	JA5	58	N.C
25	JA4	59	N.C
26	JA3	60	N.C
27	JA2	61	_JRGO
28	JA1	62	JAV
29	JA0	63	JAV
30	JD0	64	N.C
31	JD1	65	N.C
32	JD2	66	N.C
33	JAWP	67	JACD2
34	GND	68	GND



SHARP PCE500

quelques informations sur l'interface de communication

par Pierre MARTIN

Voici la liste des différents signaux ainsi que leurs fonctions respectives. La communication entre le Sharp et les autres ordinateurs devraient être facilité pour beaucoup d'utilisateurs. Nous rappelons à nos lecteurs que, en théorie seuls 3 fils suffisent pour communiquer : Un fil d'émission, un fil de réception et un fil de masse. Dans la pratique, les autres signaux sont utilisés notamment afin de vérification de transferts.

Pin number	Signal name	Symbol	Signal direction	Functions
2	Send Data	SD (TXD)	Output	Data signal to be sent
3	Receive Data	RD (RXD)	Input	Receiving data signal
4	Request to Send	RS (RTS)	Output	This signal become high level by data transmission and low level by transmission end.
5	Clear to Send	CS (CTS)	Input	When data is sent, transmission is executed if this signal is in high level, and is stopped if the signal is in low level.
7	Signal Ground	SG		Adjust the reference electric potential between input/output devices.
8	Carrier Detect	CD	Input	Transmission is executed when this signal is in high level and is stopped when the signal is in low level.
11	Receive Ready	RR	Output	High level when reception is possible and low level when reception is impossible.
14	Equipment Ready	ER (DTR)	Output	When serial input/output device circuit is open (if executed OPEN command), signal becomes high level.
1	Frame ground	FG		Grounding for maintenance
10 13		VC		Supply voltage

*Notes 1) High level means voltage level of VC. Low level means voltage level of SG.
2) As the inside is composed of C-MOS parts, if voltage exceeding permissible range (voltage level between SG-VC) is given to input/output pin, the inside may be broken.*

Dans un prochain numéro, nous aborderons l'étude et la fonction de chaque registre composant l'architecture interne du Sharp ; tels que ceux qui intéressent la mémoire Ram, le clavier, l'écran ou l'uart.

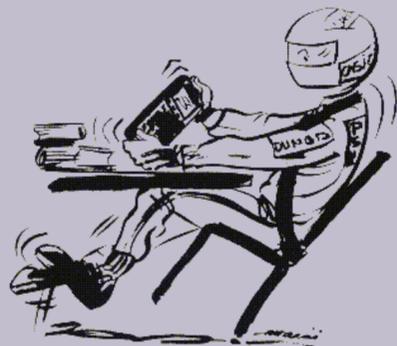
LIVRES... LIVRES... LIVRES... LIVRES...

La collection «Calculatrices Efficaces» de Dunod

Depuis le début de l'année 1991, les éditions Dunod développent la collection «Calculatrices Efficaces». Une vingtaine de titres traitent de l'utilisation de toutes les calculatrices graphiques disponibles. Cette collection s'adresse à tous les utilisateurs de calculatrices graphiques, le lycéen, l'étudiant et le développeur trouveront leur bonheur dans des livres abordant les sujets qui les passionnent : utilisation scolaire de la calculatrice, initiation à la programmation, programmation avancée, recueil de programmes, trucs et astuces, etc. Les auteurs savent de quoi ils parlent puisqu'ils sont à la fois utilisateurs quotidiens de calculatrices et spécialistes d'un domaine d'applications.

Très dynamique, cette collection accueille des petits nouveaux pour la rentrée 93 :

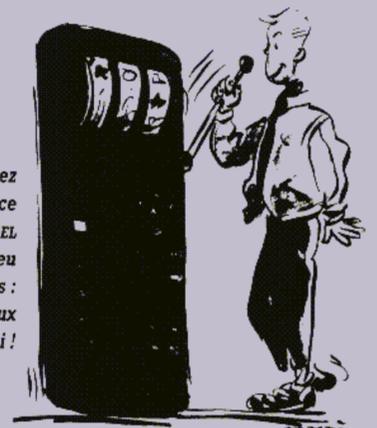
- «Casio fx : faites vos jeux !», en 224 pages, vous trouverez les listings des meilleurs jeux shareware disponibles sur Casio fx,



Transformez votre calculatrice en console de jeu !
+ bonus : 15 super-jeux livrés prêts à l'emploi !



- «Sharp EL 9200/9300 : faites vos jeux !», en 160 pages, tous les meilleurs jeux adaptés aux sérieuses graphiques de Sharp,
- «Votre fx 6800G», en 192 pages, tout ce qu'il faut savoir pour exploiter au mieux votre fx 6800G dans le cadre d'une utilisation lycéenne.



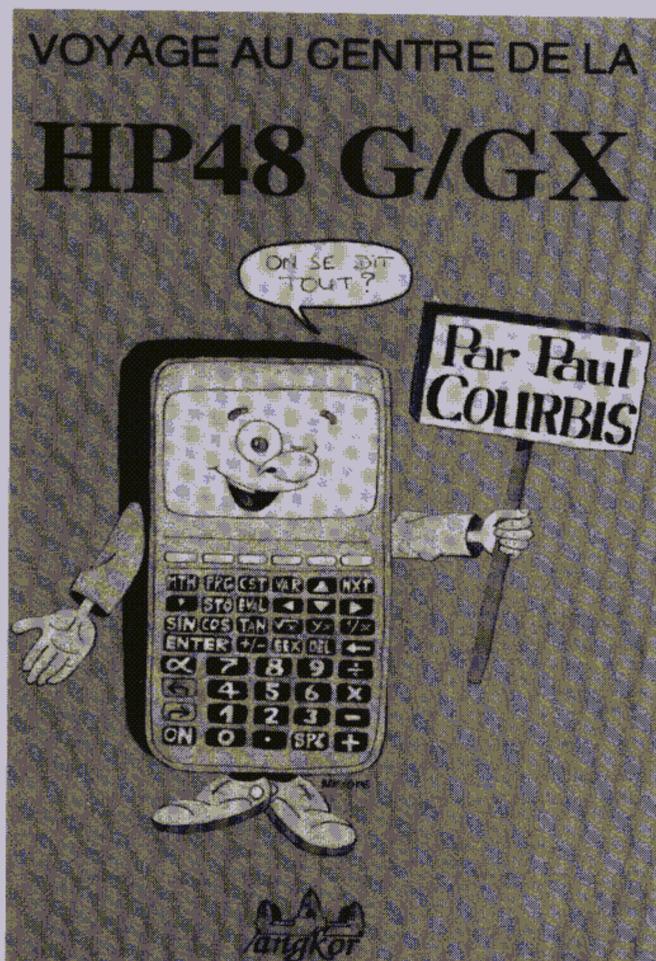
Transformez votre calculatrice Sharp EL en console de jeu + bonus : les meilleurs jeux prêts à l'emploi !



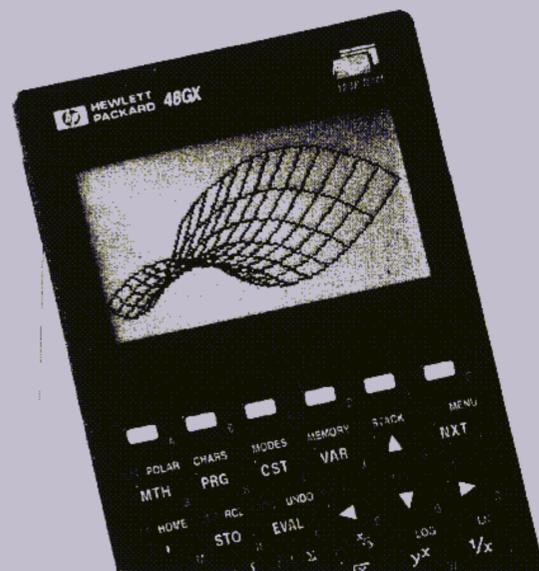
NOUVEAUTES POUR LA HP 48

1) Après le succès du "Voyage au centre de la HP48SX, Paul COURBIS réédite et sort "Voyage au centre de la HP48GX". Le must de ce qu'il faut savoir sur cette fabuleuse calculatrice !

2) Le livre "Top Niveau" pour les programmeurs acharnés ! (directement depuis les US) "ADVANCED USER'S REFERENCE MANUAL" à ne pas mettre entre toutes les mains !



HP 48G Series
Advanced User's
Reference Manual



SUITE DU NUMERO 1 SUR LE GRAPHISME DU HP 95LX

prototypes des fonctions incluses dans grafx95.c

```
void init_graph(void);
void exit_graph(void);
void get_graph_info(GRAPHINFO95 *graphinfo);
void set_logorigin(int x, int y);
void set_clip_region(int left, int top, int right, int bottom);
void move_pen(int x, int y);
void set_color(char color);
void set_rule(char rule);
void set_line_type(int type);
void set_fill_mask(char *mask);
void plot(int x, int y);
int get_pixel(int x, int y);
void draw_line(int x_dest, int y_dest);
void draw_rectangle(int x_dest, int y_dest, char fill_type);
void get_image(IMAGE *image, int x, int y, int nb_cols, int nb_lines);
void put_image(IMAGE *image, int x, int y, int invert, int rule);
void write_text(int x, int y, char *text, int rotate_flag);
#endif
```

```
void get_graph_info(GRAPHINFO95 *graphinfo)
{
    int seg, off;

    seg = FP_SEG((void far *) graphinfo);
    off = FP_OFF((void far *) graphinfo);

    asm mov ah,02h
    asm mov es,seg
    asm mov di,off
    asm int 5fh
}
```

SET_LOGORIGIN : positionnement de l'origine logique
Entrées : x, y = coordonnées de la nouvelle origine logique
Valeur retournée : Aucune

```
void set_logorigin(int x, int y)
{
    asm mov ah,03h
    asm mov cx,x
    asm mov dx,y
    asm int 5fh
}
```

SET_CLIP_REGION : Positionnement de la fenêtre de travail
Entrées : left, top = coordonnées du coin supérieur gauche
right, bottom = coordonnées du coin inférieur droit
Valeur retournée : Aucune

```
void set_clip_region(int left, int top, int right, int bottom)
{
    asm mov ah,04h
    asm mov cx,left
    asm mov dx,top
    asm mov si,right
    asm mov di,bottom
    asm mov int 5fh
}
```

MOVE_PEN : Positionne le curseur graphique
Entrées : x, y = coordonnées du point
Valeur retournée : Aucune

```
void move_pen(int x, int y)
{
    asm mov ah,08h
    asm mov cx,x
    asm mov dx,y
    asm int 5fh
}
```

SET_COLOR : sélection de la couleur courante
Entrées : color = nouvelle couleur (1:Noir, 0:Blanc)
Valeur retournée : Aucune

```
void set_color(char color)
{
    asm mov ah,09h
    asm mov al,color
    asm int 5fh
}
```

SET_RULE : Sélection de la loi de remplacement
Entrées : rule = loi de remplacement (FORCE, AND, OR, XOR)
Valeur retournée : Aucune

GRAFX95.C Version 1.01

Fonctions graphiques pour le HP95

Création : 25/02/93

Auteur : Jacques Belin

(PPC Paris #123)

Dernière modification : Correction bug sur inversion params
get_imag()

Note : Développé et testé uniquement sur Turbo C++ 1.01
TASM nécessaire (pour l'assemblage en ligne de grafx95.c)

```
#include "grafx95.h"
```

```
#pragma inline autorisation assemblage en ligne
```

INIT_GRAPH : Positionnement du HP95 en mode graphique

Entrées : Aucune

Valeur retournée : Aucune

```
void init_graph(void)
{
    asm mov ax,00h
    asm mov al,0A0h
    asm int 5fh
}
```

EXIT_GRAPH : Retour en mode texte

Entrées : Aucune

Valeur retournée : Aucune

```
void exit_graph(void)
{
    asm mov ah,00h
    asm mov al,7
    asm int 10h
}
```

GET_GRAPH_INFO : Lecture de la table d'entrée vidéo

Entrées : graphinfo = pointeur sur la table d'entrée

Sortie : graphinfo initialisé

Valeur retournée : Aucune



void set_rule(char rule)

```
{
asm mov ah,0ah
asm mov al,rule
asm int 5fh
}
```

SET_LINE_TYPE : sélection de l'aspect des lignes
Entrées : type = aspect de la ligne (sur 16 bits)
Valeur retournée : Aucune

void set_line_type(int type)

```
{
asm mov ah,0bh
asm mov cx,type
asm int 5fh
}
```

SET_FILL_MASK : Sélection du masque de remplissage
Entrées : mask = pointeur sur le masque de remplissage (8 octets)
Valeur retournée : Aucune

void set_fill_mask(char *mask)

```
{
int seg, off;
seg = FP_SEG(mask);
off = FP_OFF(mask);
asm mov ah,01h
asm mov es,seg
asm mov di,off
asm int 5fh
}
```

PLOT : Affiche un pixel
Entrées : x, y = coordonnées du point
Utilise : couleur courante
Valeur retournée : Aucune

void plot(int x, int y)

```
{
asm mov ah,07h
asm mov cx,x
asm mov dx,y
asm int 5fh
}
```

GET_PIXEL : Lecture de la couleur d'un pixel
Entrées : x, y = coordonnées du point
Valeur retournée : couleur = 1-> Noir, 0-> Blanc

int get_pixel(int x, int y)

```
{
asm mov ah,0ch
asm mov cx,x
asm mov dx,y
asm int 5fh
return(_AX);
}
```

DRAW_LINE : tracé d'une ligne à partir du point courant
Entrées : x_dest y_dest = coordonnées du point terminal
Utilise : couleur, type de ligne et loi de remplacement courantes
Valeur retournée : Aucune

void draw_line(int x_dest, int y_dest)

```
{
asm mov ah,06h
asm mov cx,x_dest
asm mov dx,y_dest
asm int 5fh
}
```

DRAW_RECTANGLE : tracé d'un rectangle à partir de la position courante du curseur, tenant compte de la couleur et de la loi de remplacement courante.

Entrées : x_dest, y_dest = coordonnées du coin de fin
fill_type = Type de tracé

OUTLINE = tracé d'un cadre, utilisant le type de ligne courant.
SOLID = Tracé d'un rectangle plein, utilisant la couleur courante
PATTERN = Tracé d'un hachurage, tenant compte du masque courant

Utilise : couleur, type de ligne, masque de remplissage et loi de remplacement courantes
Valeur retournée : Aucune

void draw_rectangle(int x_dest, int y_dest, char fill_type)

```
{
asm mov ah,05h
asm mov cx,x_dest
asm mov dx,y_dest
asm mov al,fill_type
asm int 5fh
}
```

GET_IMAGE : Lecture d'une zone graphique dans un buffer

Entrées : image = pointeur sur un buffer image
x, y = coordonnées du coin supérieur gauche de l'image
nb_cols = nombre de colonnes de l'image
nb_lines = nombre de lignes de l'image

Sortie : buffer image initialisé
Valeur retournée : Aucune

NOTE IMPORTANTE : Turbo C utilisant le registre BP pour la pile des paramètres, il est nécessaire de le sauvegarder avant de l'utiliser pour 'x1'. L'interruption 5F semblant restituer l'intégralité des registres après exécution, j'ai utilisé BX pour la sauvegarde. Prions pour que HP ne change rien dans le futur !

void get_image(IMAGE *image, int x, int y, int nb_cols, int nb_lines)

```
{
int seg, off;
int x1, y1;

x1 = x+nb_cols-1;
y1 = y+nb_lines-1;
seg = FP_SEG((void far *) image);
off = FP_OFF((void far *) image);
```

```
asm mov ah,0dh
asm mov cx,x
asm mov dx,y
asm mov es,seg
asm mov di,off
asm mov si,x1
asm mov bx,bp
```

sauvegarde de BP dans BX

```
asm mov bp,y1
asm int 5fh
asm mov bp,bx
```

restitution de BP

```
}
```



PETITES ANNONCES GRATUITES...

(maxi 150 caractères ou espaces)

Les petites annonces sont publiées sous l'entière responsabilité de leur auteur. Nous nous réservons le droit de ne pas publier certaines petites annonces n'ayant aucune relation directe ou indirecte avec les micros de poche ou les calculatrices.

Vends Portfolio + interface parallèle + Docs état neuf : 1150F Pierre tel HB 61.41.68.68 Dom 62.12.48.17

VDS Portfolio 512K + Pack-Alim + Interface parallèle + câble liaison + carte ram 128K + PBASE
Prix à débattre / tél 35 80 22 86

VDS Portfolio + BEE CARD 128K + 32K 1500F tél 69 46 23 31 avant 19 H
Région Paris (91)

VDS Portfolio 512KO + Interface parallèle pour imprimante et PC : 1500F
Alain au 64 59 83 57 Après 20 H

VDS HP41 CV + Moduletime + Modulexfunctions + 2 Module SX Memory + Lec Cartes + Imprimante1000F tél 95 51 51 60 HB ou 95 21 82 19 après 18 H

Achète pour HP48GX câble liaison minitel et BBTPH. Achète tous jeux en assembleur sur disquette. tél 48 60 82 74 / Pham Vien 21, Rue Charcot 93420 VILLEPINTE

VDS Quaderno Olivetti DD20 CGA Neuf 3000F + Interface 8 sorties / 4 entrées + Log 500F portable XT écran CGA 500F Mr Douguet 8 Bd F. Blancho Nantes 44200

VDS Atari STF 512 + 512 RAM + LECT Double face + 2 Joy + 200 Disk + Ext 512 à monter + Souris / Faire offre + Oric Atmos + écran + Hebdogiciels (30)
tél (1) 39 58 03 35 . Bruno

VDS Casio FX 8000 / 200F
Sharp EL 9300 / 800F, Casio FX850P / 800F
Yannick tél 48 85 62 01

VDS Portfolio 256K Packalim + 3 cartes RAM + 2 cartes ROM + 1 Interface Parallèle + Housse / 2500F (à débattre)
Mr Roger / tél (1) 39 54 53 41

VDS HP 48S Garantie / 6 Mois / 600F
tél (1) 30 43 79 23 Après 18H Mr Morgant P. 4 Rue Marguerite Beruoets
78280 GUYANCOURT.

VDS ATARI STF 512K + 512K Ram + Lecteur Disk DF + Souris neuve + 120 Disks + 2 Joysticks + Accessoires livres ORIC ATMOS + écran / Faire offre pour chacun ou l'ensemble, Bas prix. Ou échange possible contre voiture



fourgonnette type 4L F4, ou ordinateur de poche type PSION. Demander Bruno au (1) 39 58 03 35

Etudie toutes propositions.

VDS Organiser II Psion modèle LZ (4 lignes LCD) avec doc / état neuf / 500F. tél 43 93 57 30 (laisser message si répondeur)

VDS Super jeux pr Casio FX / 8800 / 8500 / 7800 Demander liste + 2 Progs gratuits à: T. Lebrun, 24 Rue Turenne 38000 Grenoble
tél 76 87 57 81

VDS Casio FX 880 + Listings (jeux, utils) + Extension 32 KO + Manuel et boîte d'origine / Le tout en très bon état / 950F
tél 99 62 15 31

VDS Canon X-07 + Extension 8K + Programmes + Schéma Interfaces + Docs Assembleur / 450F Contacter Tanguy tél 99 62 15 31 / Echange Progs sur 880P

VDS HP 28S neuve + Doc + Livre / 400F à débattre
Cherche contact sur HP 28 et 48
tél 99 81 56 04 le Week-End

Achète toutes revues, livres, matériel ou logiciels concernant le Z88 / Recherche particulièrement sa revue technique et un club existant en France ou en Angleterre.

Achète imprimante matricielle 9 aiguilles avec interface série et parallèle à bas prix.. Thierry

PEYCRU ESSA 147,cours de la Marne 33998 BORDEAUX-ARMEES. tel:56.91.42.97 (heures des repas).

Collectionneur ach. Calc. Programmable toutes marques ou Pocket / Cherche contact Canon X07 et ses périfs.
P. Alex 3PL Hildesheim Apt 29 / 16000 ANGOULEME. tél:45 65 39 11

VDS Atari Portfolio 512KO + ACC SAUV, Carte Ram 256, 3 Rom 256 KO, Interface parallèle, Housse cuir, Accus, Câbles / Garantie 5 Mois / Prix / 3900F / tél:38 64 63 82.
DELABROUILLE Laurent.

VDS TI85, Calc Graph 1500 fonct 32KO, Câble, Nbr progs, TBE, Garantie 1 An / 650F à débattre / tél:47.75.32.95après18H.

VDS Foliolink, Liaison Atari Portfolio / Prix à débattre / VDS carte Vidéo Trident SVGA TBE / 300F / tél:(1) 48.50.79.65

VDS Psion Series 3 256KO Ss Garantie 1500F + Traducteur / 300F + 3 Banque 250F + Jeux 200F. Emmanuel TARTU 75, rue J. Labourbe 37700 St PIERRE DES CORPS. tel:47.32.07.41

Achète Pockets, (sub)notebooks, ordi vieux et «rares»...
tél:(1)47.80.87.14 / Olivier.

VDS HP 95 LX 512K :2000F. tel: (1)60.20.06.85



POCHE COMPUTERS N°1

Vous pouvez l'obtenir directement chez nous
pour 35F + 4F de port (35F+ 10F de port hors de France)

**FAITES - NOUS CONNAITRE
DEMANDEZ - NOUS DES
AFFICHES COULEURS A
PLACER DANS LES LYCEES
LES FACs ET AILLEURS !!**



Vous aimez ce magazine ? Vous voulez le retrouver à chaque numéro, de plus en plus intéressant ?
Oui !!!, alors agissez ! abonnez-vous, c'est le meilleur moyen de nous aider efficacement

BULLETIN D'ABONNEMENT à POCHE COMPUTERS

(à recopier, ou à photocopier ou à découper et à expédier à
POCHE COMPUTERS - 42220 - BURDIGNES - FRANCE)

/ M. Prénom

Adresse Code Postal Ville

Profession (facultatif) Tél Société.....

Je possède ou un ordinateur ou une calculatrice de marque

..... Je suis intéressé surtout par tels ou tels produits

Oui, je désire m'abonner pour 6 numéros à POCHE COMPUTERS à partir du numéro et je règle par chèque la somme de **180F** (215F hors de France ; règlement par mandat postal uniquement). J'ai bien noté que je recevrai chaque numéro chez moi quelques jours avant sa parution en kiosque, en franco de port.

Je ne désire qu'un ou plusieurs numéros : N° et je règle la somme de **35F** par numéro demandé + **4F** de port. (35F + 10F de port hors de France ; règlement par mandat postal uniquement)

Vous pouvez obtenir également le numéro 1 (paru fin septembre 1993) au prix de 35F + 4F de port.

Indiquez ici, vos impressions concernant POCHE COMPUTERS. Soyez assurés que l'on vous lira avec attention.



MICROCARDS

Le spécialiste du pocket et de la calculatrice

(vente - fabrication - réparations)

TARIF TTC au 15 décembre 1993

Ce tarif remplace et annule les précédents

DESIGNATION	Prix U	Nb	Total	DESIGNATION	Prix U	Nb	Total
SHARP PC 3100 2MB	3990F			PSION series 3a 256K	Tél		
Lecteur de disquettes 1,44MB	2495F			PSION series 3a 512K	Tél		
Alimentation secteur 1 ampère	165F			SSD RAM 128K	790F		
Alimentation allume-cigares	185F			SSD RAM 512K	1690F		
Adaptateur série	255F			SSD RAM 1MB	2990F		
Adaptateur parallèle	255F			SSD FLASH 128K	520F		
Câble laplink (transfert PC et MAC)	285F			SSD FLASH 512K	1290F		
Accus 1200mA	46F			SSD FLASH 1MB	2290F		
par 3	129F			SSD FLASH 2MB	2990F		
par 6 pour lecteur de disquettes	240F			Liaison série pour PC	790F		
Etui de protection cuir	395F			Liaison série pour MAC	790F		
Etui de protection toile matelassée	295F			Liaison parallèle	440F		
Packalimentation (alim secteur + chargeur interne + 3 accus 1200mA)	380F						
Carte mémoire : voir en bas de page				ATARI PORTFOLIO			
				PORTFOLIO 128K	1690F		
HEWLETT PACKARD				PORTFOLIO 256K	1990F		
HP 48GX	2150F			PORTFOLIO 512K	2510F		
Cartes mémoire ram pour 48SX-GX				Cardram 128K	495F		
48HP128 (128K)	495F			Cardram 256K	690F		
48HP256 (2x128K)	830F			Cardram 512K	960F		
48HP512 (4x128K)	1350F			Cardram 1MB	1490F		
HP 95LX 512K	2920F			EMI 256K	520F		
HP 95LX 1MB	3990F			EMI 384K	790F		
PACK 512K pour HP95LX 512K				EMI 512K	980F		
(augmente la mémoire de votre HP95 de 512K à 1MB ; nécessite l'envoi du HP95)	1150F			Etui de protection	295F		
HP 100LX	6320F			Manuel technique et développement (150 pages)	250F		
convertisseur série/parallèle	660F			Packalimentation 1200mA	420F		
				Alimentation de sauvegarde	245F		
Cartes mémoire PCMCIA				Interface parallèle	460F		
Ram 128K	490F			Interface série	520F		
Ram 256K	690F			Câble interface PC	80F		
Ram 512K	950F			Câble interface imprimante	80F		
Ram 1MB	1590F			Rom 1.130	370F		
RAM 2MB	2690F			PBASIC	470F		
Carte traducteur Français-Anglais	1420F			MATHS PF SOLVER	380F		
Carte FAX-MODEM PCMCIA 1.0	1760F			Gestion familiale	490F		
				Le contrôleur	590F		
SHARP PCE500				Jeux1	380F		
carte mémoire 128K (sauvegardée)	495F			Jeux2	380F		
carte mémoire 256K (sauvegardée)	690F			Utils1	380F		
				Utils2	380F		
PSION series 3/3a				Utils3	380F		
Lecteur de disquettes pour 3 ou 3a	3590F			Mac Folio	380F		
Pack 128K (fait passer la mémoire à 256K de votre PSION 128K ; nécessite l'envoi du PSION)	360F			Transfert ST <> Portfolio	380F		
				Turbo-Forth	795F		
sous total A ----->				sous total B ----->			
Port colissimo + assurance :				Mme/M:NOM.....			
Pour une valeur de 1500,00F -----> 45F				Adresse			
Pour une valeur de 4000,00F -----> 60F						
Au delà, 45F/ou 60F par tranche de 1500F/4000F				TEL..... FAX			
Supplément pour chronopost : + 110F				Je possède un ordinateur de marque type			
TOTAL GENERAL (sous total A + sous total B + port)				à régler par chèque ou mandat carte * Date le / / 1993			

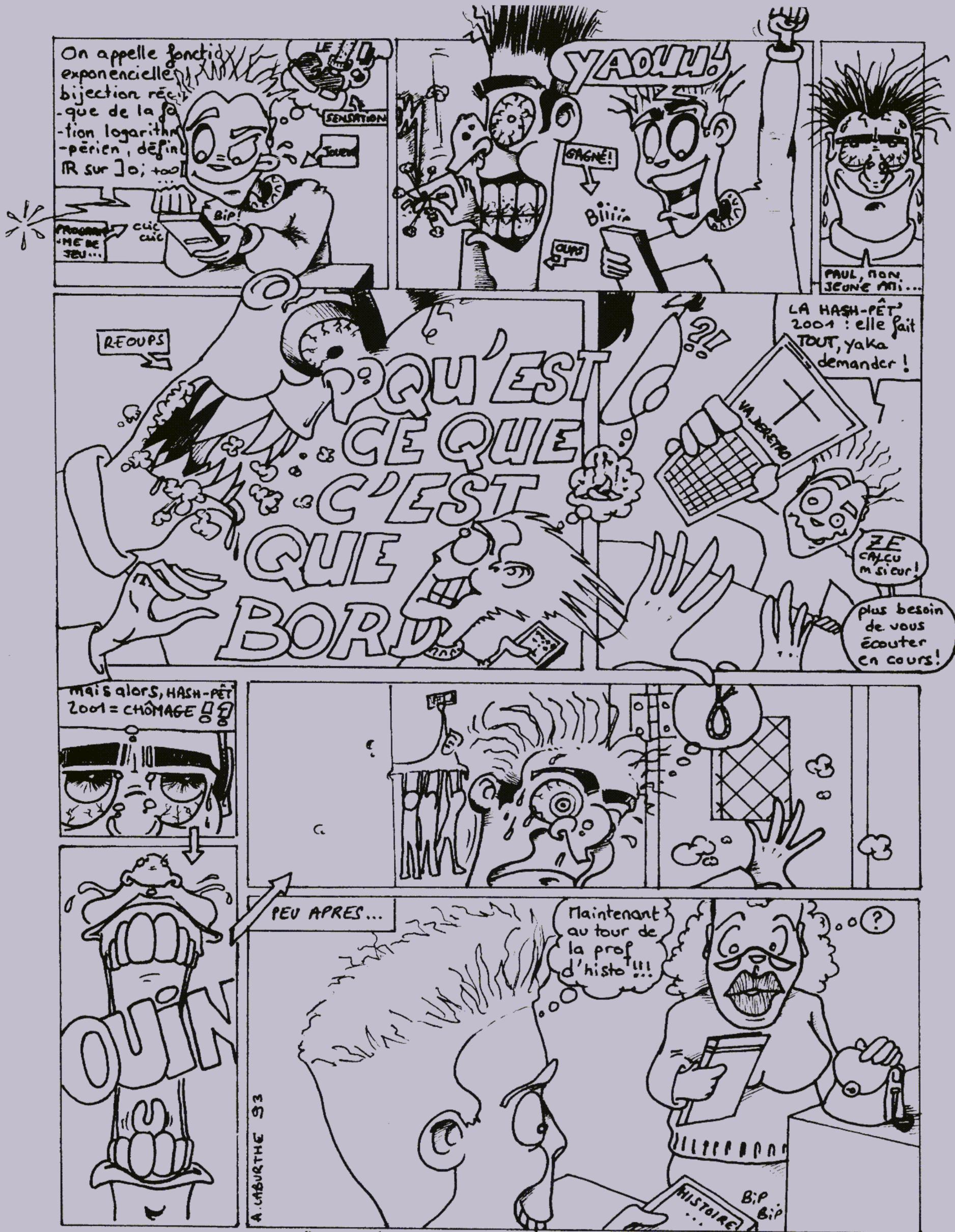
MICROCARDS - 42220 BURDIGNES cedex - Tel 77.39.68.13 Fax 77.39.19.60

Garantie sur tous nos articles - Vente par correspondance exclusivement - ouverture du lundi au vendredi inclus - de 9h à 13h et de 15h à 19h - document non contractuel



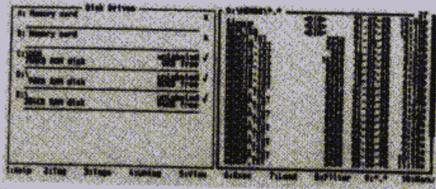
HASH-PET' 2001

par A.LABURTHE

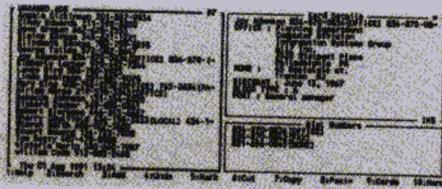


"HASH-PÉT' 2001": DELIRE COSMIQUE DU DESSINATEUR

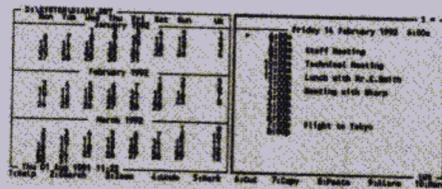
Gestionnaire d'Informations Personnelles



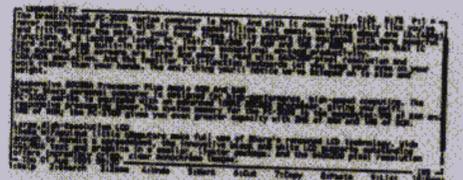
■ Gestion de fichiers pour lancer les applications et gérer les fichiers plus simplement qu'avec les commandes DOS



■ Carnet d'adresses personnel et professionnel



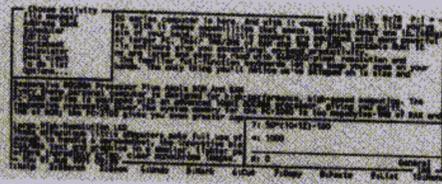
■ Emploi du temps journalier avec alarme



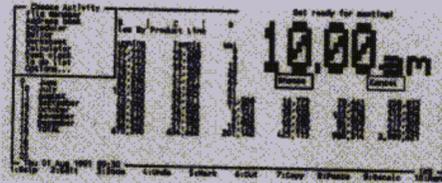
■ Traitement de textes pour prise de notes et organisation de rapports



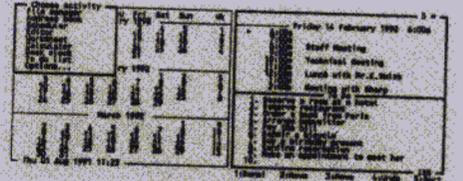
■ Feuille de calcul pour traitement des données



■ Calculatrice 12 chiffres avec mémoire

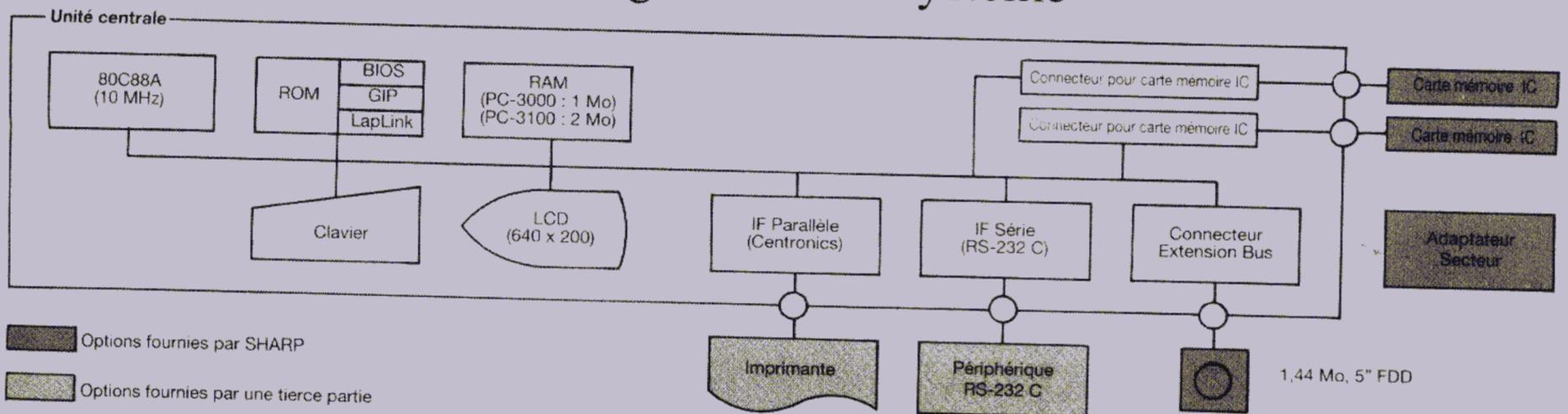


■ Horloge de bureau



■ Liste des priorités pour organisation des tâches journalières

Configuration du Système



GRAND ECRAN LCD ET CLAVIER COMPLET 77 TOUCHES

Ouvrez le PC-3000/3100 et vous serez immédiatement séduit par son clavier complet et son grand écran. Suffisamment important pour travailler sur une feuille de calcul, son affichage LCD noir et blanc met en valeur une résolution de 640 x 200 et une émulation CGA/MDA. Le PC-3000/3100 dispose également d'un clavier ergonomique et complet de 77 touches.

PROCESSEUR 80C88A ET GRANDE MEMOIRE

La mémoire du PC-3000/3100 se répartit en une RAM de 1 Mo/2Mo et une ROM de 1 Mo, ce qui est supérieur à bien des ordinateurs portables et de bureau. De plus son processeur 80C88A cadencé à 10 MHz vous donne une grande vitesse d'accès mémoire. Le gestionnaire d'informations personnelles est également fourni en standard.

CET ORDINATEUR PARLE COURAMMENT 7 LANGUES

Le PC-3000/3100 est livré en standard avec des possibilités multilingues pour le gestionnaire d'informations personnelles, incluant l'anglais, l'allemand, le français, l'italien, l'espagnol, le hollandais et le suédois.

POSSIBILITES D'EXTENSION

Deux connecteurs de cartes IC acceptent des cartes mémoire au standard PCMCIA 1.0, pour le stockage rapide et sûr des données, verrouillables pour plus de sécurité et de facilité d'utilisation. N'importe quel travail fait en déplacement peut être instantanément transféré dans votre ordinateur principal grâce à LapLink™ et un port RS232C. Le PC-3000/3100 est de plus totalement compatible IBM® PC.

LE TOUT DANS MOINS D'UNE LIVRE

C'est vrai : le PC-3000/3100 pèse moins de 500 grammes et ne mesure que 222 (L) x 112 (P) x 25,4 (H) mm, s'installant douillettement dans la paume de votre main !

SPECIFICATIONS :

Matériel

CPU	80C88A Vitesse d'horloge : 10 MHz
Mémoire ROM	1 Mo
RAM	1 Mo/2Mo (PC-3000/3100)
Ecran	FSTN B/W LCD Émulation : CGA/MDA Résolution graphique : 640 x 200
Clavier	AZERTY 77 touches
Ports E/S	Série RS232C x 1 Parallèle Centronics x 1 PCMCIA 1.0 x 2 Bus d'extension pour FD 3,5"
Dimensions (LxPxH)	222 x 112 x 25,4 mm
Poids	480 g (sans les piles)
Alimentation	Piles R6 x 3 (*), pile lithium x 1 (*) Adaptateur secteur optionnel

Logiciel

Système d'exploitation	MS/DOS® 3.3
Logiciels intégrés	LapLink™, GIP (Gestionnaire d'Informations Personnelles)
Options	Unités de disquette 3,5", 1,44 Mo Adaptateur secteur Câble de conversion parallèle Câble de conversion série Câble LapLink™

(*) Un jeu de piles R6 et une pile lithium sont livrés en standard.

• La conception, les caractéristiques et les options sont sujettes à des modifications sans préavis.

IBM® est une marque déposée de International Business Machines Corporation, MS/DOS® est une marque déposée de Microsoft Corporation, LapLink™ est une marque déposée de Traveling Software Inc.

SHARP

Distribué par :

SHARP*

SÉRIE PC-3100 ORDINATEURS PERSONNELS

LA PUISSANCE au creux de la main pour

2 866 F. H.T.*

Dimension : 222 X 112 X 25,4
Poids : 480 g

Véritable écran PC

80 colonnes X 25 lignes
(graphique CGA 640 x 200)
Autonomie moyenne
(3 piles R6) 50 Heures !

**2 MB de RAM !
(extensible à 18 MB !)**

2MB de ROM
Compatible PC et DOS 3.3 en ROM

OPTIONS :

Lecteur de disquettes portatif sur bus

(alimentation piles et secteur)

3 1/2 pouces 1,44 MB

Disque dur portatif 2 1/2 pouces 44MB

Mini souris

Mini souris stylo

Housse de transport

câble série (transferts)

Alimentation allume-cigares
ou secteur

Cartes mémoire

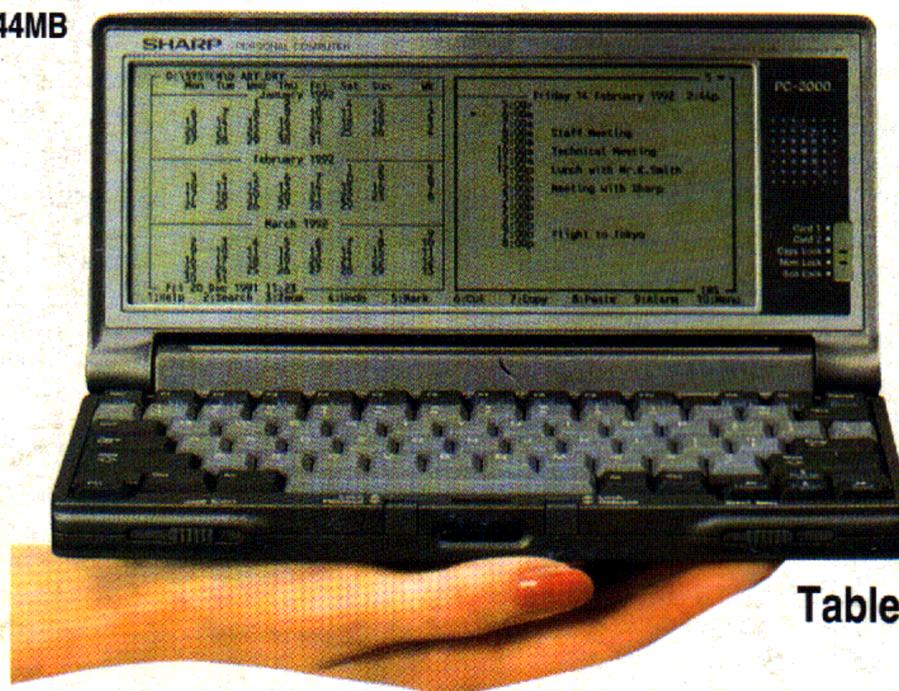
PCMCIA 128 KB à 2 MB

Carte de crédit FAX MODEM

Carte de crédit traducteur

ANGLAIS-FRANÇAIS

FRANÇAIS-ANGLAIS



CLAVIER MÉCANIQUE EXTRA

(type Machine à écrire)
2 slots pour carte PCMCIA

Sortie série et
sortie parallèle intégrées

Sa Taille ?

Vous pouvez le mettre
dans un étui
de cassette vidéo

Fourni avec :

Laplink

SHARP <-> PC

SHARP <-> MAC

Tableur compatible Lotus 1, 2, 3

Calculatrice - Horloge

Organisation de tâches journalières

Gestion de fichiers

Carnet d'adresses

Agenda

Traitement de textes

Base de données

MICROCARDS*

42220 BURDIGNES cédex (FRANCE)

Tél.: 77.39.68.13 - Fax: 77.39.19.60

Tél.: 77.39.16.33

***PC 3100**

1 MB RAM : 2 866 F.H.T. / 3 400 F.TTC

2 MB RAM : 3 330 F.H.T. / 3 950 F.TTC

+ 60F. de port colissimo

Voir bon de commande en page intérieure