

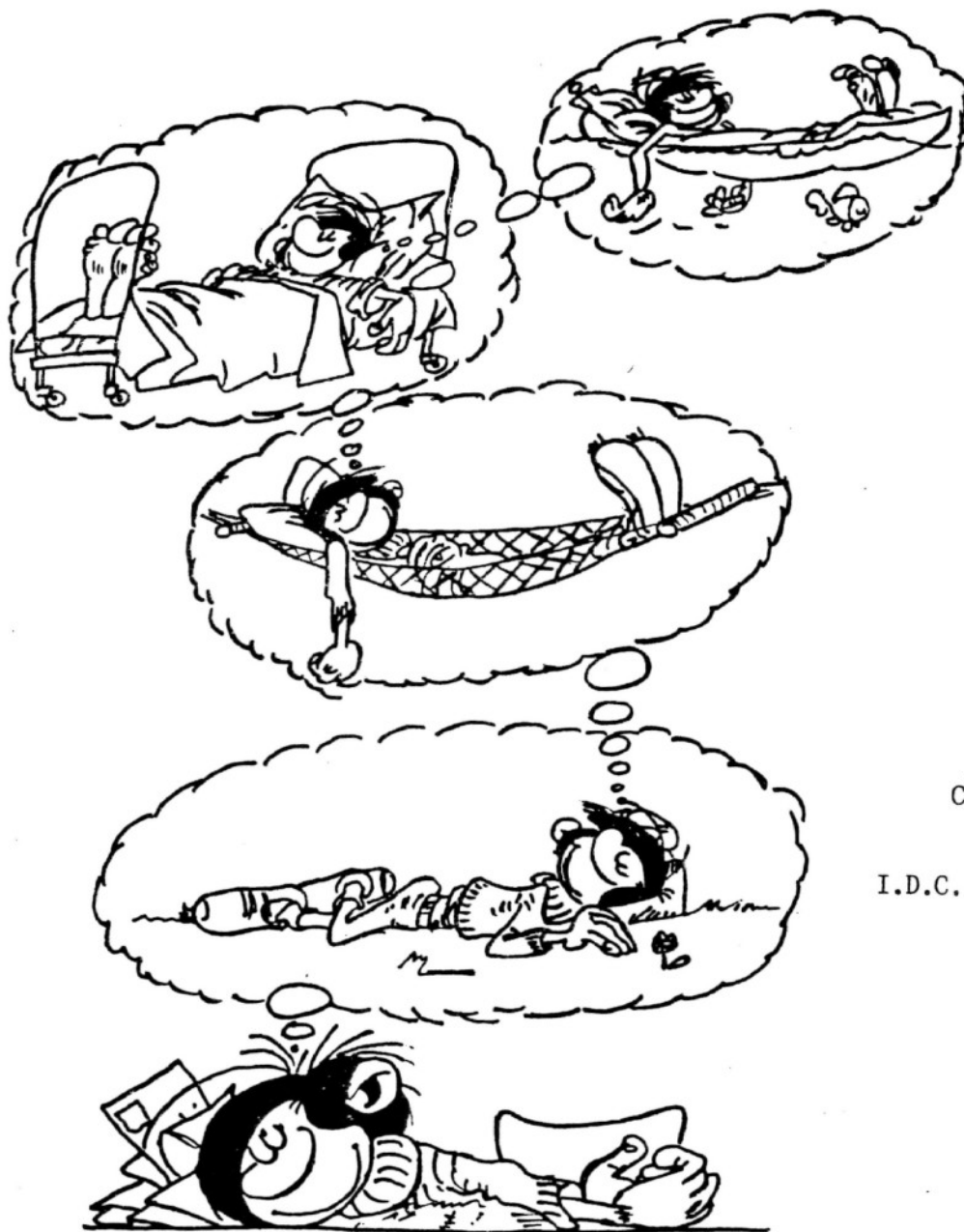
DAI CLIC

REVUE INTERNATIONALE DES CLUBS ET UTILISATEURS DU DAI P.C.

TRIMESTRIEL

Publié par le club:

I.D.C. (International DAI Club)



CAROLO DAI
D.A.I.C.
D.C.A.
I.D.C. BORDEAUX
MICRODAI

SOMMAIRE

SOMMAIRE DU NUMERO 3 - AOUT 1985

<u>ARTICLES:</u>	<u>AUTEURS:</u>	<u>PAGES:</u>
COUVERTURE		0
SOMMAIRE		1
EDITORIAL	IDC	2
INFORMATIONS UTILES		3
NOTE DE LA REDACTION	Fabrice Duluins	4
LOGICIEL : DAITEXT V21		4
COURRIER		5
CHAMPIONNAT INTERNATIONAL D'OTHELLO	L'Ordinateur Individuel	6
NOMBRES ALEATOIRES 2	Raymond Vanlathem	7
TENDANCES DU SICOB DE PRINTEMPS	Patrick Dutin	11
CRYPTO	Jan Boerrigter	13
THE 'HOLD' STORY	Pascal Janin	14
KEN-DOS : LECTURE DU DIRECTORY	Marc Vandermeersch	16
KEN-DOS : TRUC	Marc Vandermeersch	19
TEST LOGICIEL : DBASIC	Fernand Lemoine	20
HARDWARE : CARTE D'EXTENSION MEMOIRE	Eric Neve	21
TEST : BASIC V1.2	Fabien Fojud	25
DOSSIER ASSEMBLEUR : PART 2	Henri-Pierre Legry	30
PROGRAMME : PACMAN	Cimbalnik	33
L'ASSEMBLEUR PAR LA PRATIQUE 2	Raymond Vanlathem	39
HIGH SPEED LOADER	Henk Rison	44
DAI DOS 1541 : DESCRIPTION	Jan Boerrigter	47
COMPARAISON DAI DOS 1541 - VC1541	Jan Boerrigter	51
VC1541 : DESCRIPTION	Heinrich Tegethoff	53
ROUTINE : CIRCLE	Pascal Janin	57
DAI DOS 1541 : FICHIERS RELATIFS	Jan Boerrigter	58
FDD TOOLKIT - 1	Jan Boerrigter	63
PROGRAMME : SUPER VU-METRE	Pascal Janin	63
D.C.A. NEWS	DCA	64
I.D.C. BORDEAUX NEWS	IDC BX	66
D.A.I.C. NEWS	DAIC	69
DAILASSEMENT		71
TEST : DOS 3.0 (1)	Fernand Lemoine	72
TEST : DOS 3.0 (2)	Marc Vandermeersch	74
PROGRAMME : SNAKE	Pascal Janin	76
PETITES ANNONCES		80
DERNIERE MINUTE		80

COUVERTURE : d'après un dessin original de FRANQUIN.

Editorial

Enfin ! Ca y est ! Voici les vacances, que nous vous souhaitons (bien évidemment !) excellentes.

De notre coté, nous en profitons pour vous présenter un DAIClic NEWLOOK. En effet, une nouvelle reliure, un nouveau type d'impression. Espérons que cela sera plus agréable à la lecture !

A propos de vacances, nous espérons que vous les mettrez à profit pour nous envoyer un maximum de programmes et d'articles. Etant donné que les précédents numéros étaient assez fournis, nous n'avons plus grand chose de bien neuf à mettre dans le suivant!!! Ceci n'est pas vraiment un ultimatum, mais si nous ne recevons rien, on ne pourra rien vous donner...

Nous vous annonçons l'association d'un nouveau club: DCA (DAI Computer Association) d'origine Parisienne. Nous leur souhaitons donc la bienvenue parmi nous. Pour plus de détails concernant ce club, voyez les pages d'infos dans ce numéro.

Vous constaterez certainement l'abondance des articles concernant les divers systèmes de disquettes disponibles pour le DAI. Nous pensons que nombreux d'entre vous recherchent un système adapté à leurs besoins (prix, performances, ...) et nous espérons que ces articles les aiderons à se décider.

D'autre part, un projet de service automatique via modem est à l'étude. Ce système permettrait aux membres de IDC de recevoir programmes, informations via leur modem. Il permettrait également aux membres de poser diverses questions. Une permanence serait assurée certains jours durant quelques heures (indications dans DAICLIC.) Certaines informations urgentes qui ne peuvent attendre la parution du prochain DAICLIC seraient aussi fournies. On envisage aussi la création d'un service 'SOS DAI-TRESSE". Faites-nous part de vos suggestions sur ces différents points (qu'employez-vous déjà comme modem? type, prix,...).

Après cette bonne nouvelle, une moins bonne : Vu la forte augmentation du prix des cassettes DCR, nous nous voyons obligés de vous demander un supplément de 250 FB pour les programmes livrés sur cassette DCR (certifiée digitale) au lieu des 150 FB habituels.

Nous recherchons des volontaires qui seraient intéressés d'effectuer divers petits travaux pour la rédaction de DAICLIC (traductions, ...). Nous sommes plus particulièrement intéressés par de charmantes secrétaires (bénévoles ...), mais toute aide sera la bienvenue.

Bon, sur ce, nous vous quittons pour faire un petit plongeon dans l'eau quelque peu salée du Golfe de DAItona, en AnDAIlousie, où la température actuelle est de 43 degrés (à l'ombre.).

IDC on the beach.



+++++
+++++ INFORMATIONS UTILES +++++
+++++

ABONNEMENTS:

L'abonnement compris dans l'inscription est valable de janvier à décembre. Un abonnement contracté pendant l'année donne droit aux numéros déjà parus dans l'année. Si une personne est inscrite à plusieurs clubs associés, les abonnements supplémentaires lui seront remboursés à raison de 500 FB pour la Belgique et 600 FB pour l'étranger (moins les frais de remboursement).

Prix de l'inscription à I.D.C. : Belgique : 750 FB
Etranger : 850 FB

INSCRIPTIONS ET ACHATS DE PROGRAMMES:

BELGIQUE: - mandat postal

- virement sur le compte BBL: 371-0356842-45

RUB.IDC F.DULUINS ET C.POELS
ALLEE DE LA TOUR RENARD 4
1400 NIVELLES

ETRANGER: - mandat postal international uniquement.

ADRESSES: POUR LES INSCRIPTIONS ET LA BIBLIOTHEQUE DE LOGICIELS :

IDC
c/o Fabrice DULUINS
Allée Tour Renard 4
B-1400 NIVELLES (BELGIQUE)
Tél. (0)67 / 21.82.10

POUR LES ARTICLES ET PROGRAMMES A PUBLIER :

IDC - REDACTION
c/o Christian POELS
Rue des Bas-Sarts 10
B-4100 SERAING (BELGIQUE)
Tél. (0)41 / 37.16.06

POUR LES PROGRAMMES A EDITER :

IDC - SOFTWARE
c/o Marc VANDERMEERSCH
Avenue du Vert Bocage 17
B-1410 WATERLOO (BELGIQUE)
Tél. (0)2 / 354.13.63

CONTACTS CLUBS ASSOCIES:

CAROLO-DAI : Etienne SZIGETVARI, Route Provinciale 7, B-1361 CLABECQ.
DAIC : Jacques MOENS, Clos Fontaine des Ducs 6, B-1310 LA HULPE
DCA : Luc D'ARANTES, Rue Defresne Bast 6, F-95100 ARGENTEUIL
IDC BORDEAUX: Bruno DELANNAY, Rés. Les Acacias B3, Av. Saige, F-33600 PESSAC
MICRODAI : Victor LESPINEUX, Rue Latour 39, B-4020 LIEGE

COPYRIGHT:

Les articles publiés n'engagent que la responsabilité de leur auteur. Toute reproduction, même partielle, de ce magazine est interdite sans l'accord préalable de l'éditeur responsable.

EDITEUR RESPONSABLE: Fabrice DULUINS, Al. Tour Renard 4, B-1400 NIVELLES

Note de la rédaction.

Vous l'aurez de suite remarqué ce numéro de DAICLIC est tout à fait différent des précédents non par son contenu mais par sa forme. En effet, nous essayons de trouver un 'look' pour la revue qui soit à la fois agréable et attrayant pour le lecteur. N'oubliez pas de nous donner vos impressions et vos critiques, nous essayerons à partir de celles-ci, dans la mesure de nos possibilités, d'améliorer la qualité de VOTRE revue.

Nous remercions tout particulièrement Monsieur Roger LEMOINE qui nous a gracieusement prêté une machine à écrire BROTHER CE-61 (muni de l'interface IF-50) sur laquelle nous avons imprimé la revue. Nous espérons pouvoir faire paraître un test complet de l'utilisation ce type de machine dans le prochain numéro, si toutefois vous souhaitez avoir de plus amples renseignements sur les machines à écrire, imprimantes, ..., contactez les :

Ets ROGER LEMOINE
152, rue de Namur
B-1400 NIVELLES
Tél.(0)67/22.29.70

La machine à écrire qui nous a permis d'imprimer la revue, ne comportant pas le caractère ('double croix'), nous avons adopté comme convention d'utiliser en lieu et place de ce caractère soit 'Fr' soit 'H'.
Attention les nombres décimaux sont presque toujours précédés de ces signes.

Fabrice Duluins

Nouveauté : DAITEXT V.21.

Alain Mariatte et IDC ont le plaisir de vous annoncer un nouveau logiciel : DAITEXT V.21 + un EDITEUR DE MACROS.

DAITEXT V.21 est un programme de traitement de texte dérivé de la version V.20 (voir DAICLIC numéro 1) mais comportant plusieurs améliorations dont vous trouverez une description complète dans le prochain numéro .

L'Editeur de macros est un programme basic (joint à DAItex V.21) servant comme son nom l'indique à éditer des macros. Les macros sont des chaînes utilisées en télématique. On y stocke les mots de passe et les codes qui sont trop longs à taper au clavier.

DAITEXT V.21 est disponible au prix de 1.800 FB sur cassette audio et de 2.050 FB sur mini-cassette digitale .
Les utilisateurs possédant la version V.20 peuvent moyennant un supplément de 200 FB (ou de 450 FB pour les mini-cassettes digitales) obtenir la nouvelle version de DAItex.

Courrier.

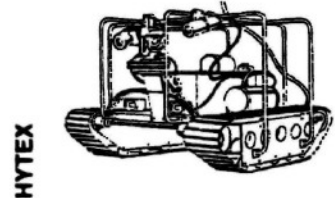
Monsieur Gilbert Pons (société Vidéo Reporter) nous a adressé récemment une lettre concernant une collaboration artistique avec vous, c'est-à-dire les membres des clubs associés à IDC.

Cette collaboration concerne la réalisation d'un dessin animé en 16 ou 35 mm sur micro-ordinateur .

Deux solutions :

- un thème par club avec la réalisation d'une séquence de 40 secondes
- un thème par créateur avec la réalisation d'une séquence de 2 à 3 s.

Thème : " HYTEC
ou
le Tunnel sous la Manche "



Pour plus de renseignements, adressez-vous à Vidéo Reporter, Mr Gilbert Pons, 3 rue Mozart, F-34500 Béziers (Tél.: (67).30.79.69).

Nous vous signalons que les 7 & 8 septembre 1985, la salle Claridge (24 ch. de Louvain, B-1040 Bruxelles) accueillera 'Chips & Puces', la première foire de la micro-informatique de seconde main. 'Chips & Puces' se présente comme un marché libre, une bourse de la micro occasion (logiciels, livres, périphériques, matériel...).

Il est possible de louer des tables à la journée au prix de 500 fb, le raccordement électrique est prévu. Si vous souhaitez réserver une table, écrivez à IDC en envoyant une enveloppe timbrée avec vos noms et adresse. Nous vous transmettrons alors le formulaire d'inscription.

Après la parution de ma lettre dans le numéro 2 de DAICLIC, je vous écris pour vous faire connaître la suite de mon 'affaire'.

Je voudrais tout d'abord vous remercier pour l'aide que vous avez bien voulu m'apporter et qui semble s'être avérée décisive. PRODATA-FRANCE a, en effet, fini par me rembourser en m'envoyant début juin un chèque de 1000 FF (8 mois ma commande).

Je leur avais écrit un mois auparavant et je venais de contacter une association de consommateurs qui n'a pas eu le temps d'agir. C'est pourquoi je présume que je dois cette heureuse, mais tardive, conclusion à votre intervention auprès de DAI-BELGIQUE.

Remerciez également de ma part les personnes qui spontanément et très gentiment se sont proposées de défendre mes intérêts auprès de PRODATA.

Je pense, comme vous, que, sans faire de la rubrique courrier une rubrique 'allo bobo', il serait bien que les DAIistes vous fassent part de leurs problèmes, même résolus, et qui peuvent constituer des expériences instructives pour les autres. En espérant, bien évidemment, que vous receviez le moins possible de lettres de ce genre.

Emmanuel BOUCHERON,
F-28 CHATEAUDUN (FRANCE).

DAICLIC: Nous sommes heureux de voir que votre problème a enfin été résolu! N'ayant pas reçu d'autres lettres telles que la votre, nous espérons que votre cas est une exception due à une négligence.

7^e Championnat International de programmes d'Othello-Reversi

7th Othello-Reversi Programs World Championship

Organisé par

L'ORDINATEUR L'INDIVIDUEL

Paris, le 20 juin 1985

Chers amis,

Vous savez, sans doute, que le prochain championnat de programmes d'OTHELLO-REVERSI se déroulera au SICOB, le 21 septembre prochain pour la catégorie ordinateurs de table, et les 21 et 22 septembre pour la catégorie ordinateurs de poche.

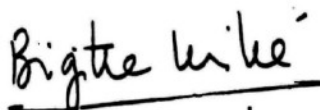
Nous serions très heureux de compter certains de vos adhérents parmi les participants et souhaitons recevoir leur inscription le plus rapidement possible afin de pouvoir organiser ce tournoi encore mieux que les années précédentes.

A cet effet, vous trouverez ci-joint le texte des règlements pour les deux catégories, et un bulletin d'inscription que vous pouvez évidemment photocopier. ☺

Dès réception des bulletins d'inscription, nous enverrons aux participants les informations pratiques concernant le tournoi.

Nous vous attendons nombreux,

Bien cordialement,



Brigitte MILLÉ

Promotion

☺ Pour recevoir le règlement et le bulletin d'inscription, envoyez 2 timbres à 12FB ainsi que vos coordonnées à F.Duluins

Pour les Français, envoyez 2 timbres à 2.10FF à B.Delannay (adresse dans la revue)

par Raymond VANLATHM (Club CAROLODAI / DAIClic)

Postscriptum No 1

1.- Un DAIiste m'a fait remarquer qu'à la mise sous tension de son DAI ou lors d'une première utilisation de la fonction RND(1), il n'obtenait pas la série de nombres aléatoires illustrant le premier article consacré à ce sujet. Si vous êtes dans le même cas, n'incriminez pas votre DAI ou n'en voulez pas à votre serviteur.

L'explication en est assez simple. Sous certaines conditions l'initialisation des cellules-mémoires ne se fait pas correctement, notamment après un STACK OVERFLOW. Avant d'utiliser la fonction RND, vérifiez le contenu des cellules-mémoires, ou se trouve le nombre source du générateur, par un DISPLAY en UTility: >D12E 130 (le contenu de #12D importe peu). Ces 3 positions de mémoire devraient contenir FF FF FF, lors d'une première utilisation de la fonction RND après une mise sous tension à froid, ou après une attente d'au moins 15 à 20 secondes (voire même plus) quand on éteint son DAI.

J'ai pu constater moi-même que dans certains cas l'un des octets FF au moins était remplacé par 7F ou F8 (voire FC, FE, ...): d'où la génération d'une série différente de celle annoncée dans l'article.

Pour obtenir les mêmes chiffres que ceux cités dans l'article, il suffit de modifier le contenu des cellules-mémoires concernées, par la commande SUBSTITUTE en UTility: après >S12E introduisez 3 fois FF.

2.- Un lecteur de DAICLIC m'a signalé que je n'ai pas encore parlé de la fameuse instruction RND(0), qui met en oeuvre le générateur hardware du DAI, lié à l'état du générateur de bruit blanc (NOISE). Ce dernier est censé produire des nombres parfaitement aléatoires. Mais les essais effectués sur divers DAIs ne me permettent pas de conclure valablement, vu les résultats difficilement contrôlables (en effet, ce type de générateur ne produit jamais 2 séries identiques, d'où la difficulté de soumettre ce générateur à des tests de fiabilité) et vu l'état défectueux de certains générateurs NOISE (dans la plupart des DAIs, il suffit d'apporter une petite modification à l'étage du générateur NOISE, pour obtenir une série plus ou moins valable de nombres aléatoires).

De toute façon, on a pu constater qu'une série générée par le hard du DAI est moins fiable (dispersion plus grande) que les séries produites par le soft de la ROM. Nous y reviendrons plus tard! Il y a lieu de noter que les programmeurs DAIistes utilisent très rarement l'instruction RND(0) dans leurs programmes. A tort ou à raison?

PRINCIPES du GENERATEUR "SOFT" du DAI

Dans le premier article nous avons abordé l'étude des générateurs de nombres aléatoires et nous avons vu comment se comportait d'une manière générale le DAI. Voyons maintenant d'un peu plus près ce qui se passe à l'intérieur de celui-ci.

Il est bon de se rappeler que tout nombre est mémorisé dans le DAI sous forme de 4 octets de 8 bits et qu'à un nombre hexadécimal peut correspondre non seulement un nombre entier, mais également une famille de nombres réels: tout dépend de l'interprétation demandée. Ainsi au nombre hexadécimal 02 80 00 00 correspond le nombre entier 41 943 040 et une famille de nombres réels voisins de 2.0 (allant de 1.99999988... à 2.00000079...). De fait, comme le DAI n'affiche que maximum 6 chiffres significatifs en virgule flottante, toute la famille de nombres réels sera affichée sous la forme de 2.0. Notons aussi que le nombre hexadécimal 01 FF FF FF, auquel correspond également une famille de nombres réels voisins de 2.0, donne un nombre entier décimal très différent: 33 422 847. Cela tient au système différent de représentation des nombres entiers et des nombres réels en binaire.

D'autre part, il est bon de savoir que la valeur de l'argument positif X de l'instruction RND, ne joue aucun rôle dans la détermination du nombre aléatoire. Tout se ramène à une routine RND(1), générant un nombre aléatoire, qui sera en dernière instance multiplié par l'argument X, avant d'être affiché à l'écran. Tout se passe comme si nous avions l'instruction RND(X) sous la forme: X * RND(1).

La formule de base utilisée dans le D A I est:

$$\begin{array}{ccccccc}
 * & \text{=====} & * & & & & \\
 * & X & = & AX & + & B & * \\
 * & n+1 & & n & & & * \\
 * & \text{=====} & * & & & &
 \end{array}$$

dans laquelle A = 59 (en hexadécimal: 3B) et B = 125 000 001 (en hexadécimal: 07 73 59 41), X(indice n) étant le nombre aléatoire précédent. Remarquons que A et B sont des nombres entiers; X est un nombre réel (en virgule flottante).

La valeur X(indice n+1) sera affinée avant d'être utilisée à son tour dans la formule en vue de générer le nombre suivant. La procédure d'affinement est la suivante: la valeur hexadécimale de X(indice n+1) est tronquée; seuls les 3 octets les moins significatifs sont conservés. Ainsi le nombre hexadécimal 70 CC OF 94 devient CC OF 94. Le DAI utilise un masque AND (ET) soit H00 FF FF FF, pour obtenir ce résultat: la présence d'un bit 1 dans un tel masque permet la recopie du bit correspondant du nombre de départ, tandis qu'un bit 0 met le bit correspondant du nombre résultant à zéro.

70 CC OF 94 =	0111 0000	1100 1100	0000 1111	1001 0100
Masque ET =	0000 0000	1111 1111	1111 1111	1111 1111

00 CC OF 94 =	0000 0000	1100 1100	0000 1111	1001 0100

Le 4ème octet (70) est alors remplacé par H01, ce qui nous donne le nombre 01 CC OF 94: c'est la valeur X(indice n+1) affinée. Remarquons que le premier chiffre hexadécimal du 3ème octet (celui qui suit 01) doit toujours être égal ou supérieur à 8; à défaut, le DAI ajoute automatiquement 8 à ce chiffre. Ainsi le nombre hexadécimal 71 7A FO 5D, qui sera tronqué comme 7A FO 5D, deviendra 01 FA FO 5D (le DAI ajoute 8

au 7 de 7A : soit $8 + 7 = F$, d'où FA). Le DAI utilise un masque OR (OU) soit H01 80 00 00, pour obtenir ce résultat: la présence d'un bit 1 dans un tel masque suffit pour mettre le bit correspondant du nombre résultant à un, tandis qu'un bit 0 permet la recopie du bit correspondant du nombre de départ.

```

OO 7A FO 5D = 0000 0000   0111 1010   1111 0000   0101 1101
Masque OU = 0000 0001   1000 0000   0000 0000   0000 0000
-----
01 FA FO 5D = 0000 0001   1111 1010   1111 0000   0101 1101

```

On dira que le nombre X(indice n+1) obtenu après affinement, interprété comme nombre réel, se trouvera dans l'intervalle 1.0 - 2.0; car les valeurs limites que peut prendre ce nombre sont 01 80 00 00 (nombre réel = 1.0) et 01 FF FF FF (nombre réel = 2.0).

L'opération complète (calcul de la formule, plus affinement) sera répétée 5 fois avant de fournir le nombre X(indice n+5), qui servira d'amorce au cycle suivant et qui deviendra le nombre aléatoire sortant, c'est-à-dire disponible pour l'utilisateur. Ce nombre subira cependant une dernière modification, à savoir: le nombre généré sera d'abord ramené dans l'intervalle 0.0 - 1.0 et puis multiplié par l'argument X.

EXEMPLE de REALISATION PRATIQUE

Données

Soit une routine utilisant RND(10) et soit le dernier nombre aléatoire sorti 0.359846. Nous devons nous attendre à voir sortir comme prochain nombre aléatoire 4.88753 (= $10 * 0.488753$) tel qu'indiqué dans l'article précédent (voir DAICLIC No 1 page 52).

REM: Nous avons utilisé au départ RND(-1) pour générer cette série de nombres aléatoires.

Tableaux

Les tableaux suivants donnent les valeurs obtenues aux différents stades de la procédure de génération du nombre aléatoire. Le tableau No 1 donne les valeurs en hexadécimal, tandis que les valeurs correspondantes sous forme d'entiers décimaux figurent au tableau No 2 et celles sous forme de nombres réels au tableau No 3. Les valeurs peu intéressantes, car n'ayant pas beaucoup de sens pour l'utilisateur sont mises entre parenthèses.

TABLEAU No 1	Colonne A	Colonne B	Colonne C	Colonne D
Ligne 0	H 04 A0 00 00	H 01 AE OF 6D		
Ligne 1	H 63 1D 8E 1F	H 6A 90 E7 60	H 90 E7 60	H 01 90 E7 60
Ligne 2	H 5C 65 53 20	H 63 D8 AC 61	H D8 AC 61	H 01 D8 AC 61
Ligne 3	H 6C EF BA 5B	H 74 63 13 9C	H 63 13 9C	H 01 E3 13 9C
Ligne 4	H 6F 55 84 F4	H 76 C8 DE 35	H C8 DE 35	H 01 C8 DE 35
Ligne 5	H 69 4B 36 37	H 70 BE 8F 78	H BE 8F 78	H 01 BE 8F 78
Ligne 6	H 01 BE 8F 78	H 7F FA 3D E0	H 03 9C 66 AC	

TABLEAU No 2	Colonne A	Colonne B	Colonne C	Colonne D
Ligne 0	(77 594 624)	28 184 429		
Ligne 1	1 662 881 311	1 787 881 312	(9 496 416)	26 273 632
Ligne 2	1 550 144 288	1 675 144 289	(14 199 905)	30 977 121
Ligne 3	1 827 650 139	1 952 650 140	(6 493 084)	31 658 908
Ligne 4	1 867 875 572	1 992 875 573	(13 164 085)	29 941 301
Ligne 5	1 766 536 759	1 891 536 760	(12 488 568)	29 265 784
Ligne 6	29 265 784	(2 147 106 272)	(60 581 548)	

TABLEAU No 3	Colonne A	Colonne B	Colonne C	Colonne D
Ligne 0	10.0	1.35985		
Ligne 1	(8.60168 E-10)	(1.34952 E-7)	(0.566031)	1.13206
Ligne 2	(1.15193 E-11)	(1.57651 E-9)	(0.84638)	1.69276
Ligne 3	(8.93056 E-7)	(1.88974 E-4)	(0.387018)	1.77404
Ligne 4	(2.54868 E-6)	(7.66251 E-4)	(0.784641)	1.56928
Ligne 5	(3.50232 E-8)	(1.13583 E-5)	(0.744377)	1.48875
Ligne 6	1.48875	0.488753	4.88753	

Commentaires

- En OA : valeur de l'argument X.
- En OB : amorce du cycle=valeur de X(indice n) dans l'intervalle 1.0-2.0. Cette valeur est celle obtenue au cours du cycle précédent et se trouve dans les cellules-mémoires H12D à H130. Si le contenu de H12D est différent de H01, le DAI le remplace d'office par H01; le contenu de H12E doit être compris entre H80 et HFF (question d'intervalle).
H01 AE OF 6D vaut 1.35985 (arrondi de 1.359846)
ou 28 184 429 (nombre entier).
- En 1A : produit de X(indice n) par A.
H01 AE OF 6D * H3B = H63 1D 8E 1F,
soit 28 184 429 * 59 = 1 662 881 311.
- En 1B : somme de AX(indice n) plus B.
H63 1D 8E 1F + H07 73 59 41 = H6A 90 E7 60,
soit 1 662 881 311 + 125 000 001 = 1 787 881 312.
- En 1C : valeur de X(indice n+1) tronquée.
H6A 90 E7 60 devient H90 E7 60.
- En 1D : valeur de X(indice n+1) dans l'intervalle 1.0 - 2.0.
H90 E7 60 devient H01 90 E7 60, soit en virgule flottante 1.13206.
- De 2A à 2D : 2ème boucle donnant X(indice n+2). La procédure est semblable à celle de la première boucle.
- De 3A à 3D : 3ème boucle donnant X(indice n+3). Notez que l'octet H63 (en 3C) devient HE3 (en 3D); en effet, comme 6 est plus petit que 8, nous avons 6 + 8 = HE (voir plus haut).
- De 4A à 4D : 4ème boucle donnant X(indice n+4).
- De 5A à 5D : 5ème boucle donnant X(indice n+5).
- En 6A : valeur identique à celle de 5D. La valeur trouvée en 5D, à savoir H01 BE 8F 78 (nombre réel 1.48875) servira d'amorce au cycle suivant.
Elle sera conservée dans les cellules-mémoires H12D à H130.
- En 6B : valeur ramenée dans l'intervalle 0.0 - 1.0.
Cela nous donne H7F FA 3D E0, soit le nombre réel 0.488753.
- En 6C : nombre aléatoire recherché. Comme l'argument X est égal à 10, le nombre aléatoire utilisable sera 10 * 0.488753 = 4.88753, ce que nous attendions!

De notre correspondant à Paris, Patrick DUTIN.

Ce n'est pas une chose facile que de définir brièvement le contenu d'un salon où se trouvent 3.500 produits représentés par environ 600 exposants. Si ce salon a le micro-ordinateur pour thème, il concerne peu le marché du 'home computer' et la plupart des principaux revendeurs d'ordinateurs domestiques, s'ils étaient présents, s'adressaient surtout au marché professionnel.

Une étude de marché, faite par une des plus importantes sociétés de Marketing pour les années 83 à 88, explique la volonté des constructeurs.

Evolution des marchés de l'informatique :

- * 7 % : Vente de temps machine.
- * 13 % : Prestations intellectuelles.
- * 22 % : Engineering.
- * 30 % : Progiciels.

En effet, on peut observer de gros investissements faits pour :

- * Progiciels (demande horizontale)
 - Comptabilités, payés, gestion de stocks, etc..
 - Tableurs type MULTIPLAN
 - Systèmes de décentralisation de l'information (CONVERGENT TECHNOLOGIE)
- * Progiciels (demande verticale)
 - Automobile.
 - B.T.P.
 - Laboratoires.
 - Notaires.
 - Medical.
 - Radiologues.
 - Huissiers.
 - Transports.
 - Négoces.
 - Avocats.
 - Experts comptable.
 - Restauration.

Les grands thèmes du salon :

- Utilisation du micro-ordinateur dans l'entreprise.
- Connaissance de l'environnement d'un micro.
- Le choix du micro-ordinateur.
- Le choix des programmes.
- Démystifier l'informatique.
- Appréciation des coûts hard et softs.
- Composition d'un cahier des charges.
- Le choix des imprimantes.
- Les carrières de l'informatique.
- Les réseaux.
- Le choix d'un tableur.
 - pour les finances.
 - pour le marketing.
 - pour la gestion du personnel.
 - pour la gestion d'une P.M.E.
- La gestion de fichiers.
- Les traitements de texte.
- Les logiciels intégrés.

Les nouveautes que j'ai pu remarquer :

- MASSCOMP, 1e MC 500
(systeme oriente pour l'acquisition de l'information dans un environnement industriel (1 million de points/seconde), amplification a gain programmable etc..., carte biprocesseur 68000 et 68010 MOTOROLA, resolution/12 bits, 16 a 64 voies)
- MATRA, 1e STM1
(micro monitorant l'acces au reseau MINITEL)
- MICRO-EXPANSION, 1e G.502
(drives de 1 Mega et carte accellerateur avec un 8088 pour APPLE II)
- SFENA et MICRO SYSTEME
(caisses enregistreuses intelligentes)
- MDS, 1e HERO
(Ordinateur professionnel de reseau avec D.B. relationnel et un logiciel graphique d'implementation a MULTIPLAN)
- GRID, 1e GRIDCASE
(portable a affichage par cristaux liquides 640 points par 200 avec 25 lignes de 80 col., process. INTEL 80C86)

Les plus grands du salon etaient certainement :

- APPLIED DATA RESEARCH FRANCE et
- CINCOM SYSTEMS

avec des progiciels integres pour I.B.M.

- Bases de donnees relationnelles
- Suivi de production.
- Outils d'aide au developpement.
- Applications VIDEOTEX

et il y avait, bien-entendu, le repertoire habituel d'exposants du SICOB :

- editions et revues informatiques.
- supports magnetiques divers..
- dispositifs anti-micro-coupures.
- vente de jeux pour APPLE, AMSTRAD, ORIC, SPECTRUM, etc..
- machines a dicter.
- machines a ecrire a memoire.
- fournitures diverses (listing, rubans, etc...)
- revendeurs de calculettes, etc...

J'espere que mon expose a ete suffisamment complet. Je pense qu'il faut noter une tendance a favoriser le developpement des progiciels, un interet sensible pour les applications sur reseaux tels le systeme de CONVERGENT TECHNOLOGIE ou les application MINITEL ou VIDEOTEX, TRANSPAC etc..., de nouveaux portables (GRID, HP) dotes d'affichages a cristaux liquides et qui ont bien besoin de s'ameliorer.

L'impression d'ensemble; d'une annee sur l'autre, reste la meme des prix incroyablement eleves pour des systemes qui sont loin de tous se valoir, des acheteurs naifs qui croient en l'intelligence des systemes; exemple de NORTH-STAR qui a distribue une grande quantite de ce merveilleux petit systeme mais dote d'une comptabilite a l'americaine, absolument inexploitable en France.

Ceci est une plainte. Une plainte de quelqu'un qui prend bien soin de sa bibliothèque de programmes. Quelqu'un qui est inquiet de perdre ses programmes à cause d'une simple défaillance de la cassette. Quelqu'un qui aime adapter les programmes suivant ses propres désirs.

De plus en plus, les programmes de la série 'DAInamic software' sont sauvés sur cassette (audio ou DCR) d'une telle façon qu'il est impossible de faire autre chose que de les exécuter. Des tas et des tas d'astuces sont utilisés pour empêcher l'utilisateur - moyen - de faire des copies du programme.

Je comprends très bien la nécessité commerciale de ceci et n'ai rien contre la lutte anti-piratage. Mais d'un autre côté, nous ne sommes qu'un club d'utilisateurs, et pas une organisation commerciale qui doit faire du bénéfice. Les programmes, en vente dans la bibliothèque de software, devraient non seulement être utilisés par leurs acheteurs mais devraient aussi les encourager à créer leurs propres programmes, leur apprendre des idées astucieuses pour obtenir des effets spéciaux, etc.

Au sujet du piratage. Une autre méthode pour éviter l'apparition de copies illégales pourrait être un prix plus bas des programmes. Les prix actuels sont assez élevés, et je vois encore une tendance à l'augmentation de ceux-ci. Je sais que le club, pour la revue, etc, a besoin de beaucoup d'argent. Mais faut-il que tout soit financé par la vente de logiciels ?

Au sujet des copies de programmes. Il est impossible d'éviter les copies audio des cassettes ordinaires. Et pour les cassettes DCR, il y a quelques programmes de copie disponibles chez certains utilisateurs du DAI. Le nombre de ces programmes augmentera en même temps que le nombre de programmes protégés. Chaque protection peut être supprimée et le sera, quelle qu'en soit la raison. Je l'ai fait moi-même parce que j'ai besoin de copies de sauvegarde et parce que j'en ai besoin pour adapter les programmes à un système de disquettes bon marché. Mais je n'aime pas ça, passer tant de temps pour découvrir comment les protections sont faites. Je pourrais utiliser ce temps à quelque chose de beaucoup plus intéressant!

Un mot à ceux qui écrivent des programmes (dont je suis). Sommes-nous aussi professionnels, sommes-nous aussi inquiets que d'autres apprennent les choses que nous avons nous-même découvert que nous devons protéger nos programmes?? Revenons au niveau de "l'amateur" et faisons profiter les gens de notre talent.

Un dernier mot aux utilisateurs de programmes. Avant de faire des copies illégales, rappelez-vous que les clubs ne peuvent exister que s'il y a assez d'argent. Alors, achetez vos programmes. Les gars qui écrivent des programmes passent beaucoup de temps pour vous permettre de profiter du résultat. Et ne rabaissez pas les réunions de clubs à des séances où on copie le plus possible de programmes.

Jan BOERRIGTER - Aout 1984.

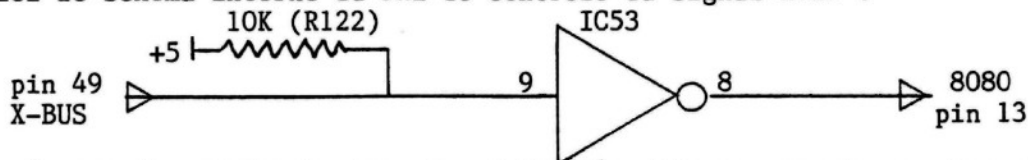
NDLR: L'auteur est membre du club néerlandophone DAInamic. Il est évident que son témoignage s'applique aussi à tous les autres clubs DAI.

C'est en lisant le numéro de TILT du mois de Juin que l'idée de ralentir le DAI m'est venue. En effet, dans cette revue d'intérêt général (sans 'pub' S.V.P !) était cité un concepteur anglais qui avait eu l'idée originale de créer une interface pour ralentir le CBM64. Fort de cette idée géniale, je me suis empressé de le copier.

Trois méthodes pouvaient alors être appliquée à notre cher DAI :

- ralentir la fréquence d'horloge en changeant la capacité en série avec le quartz de 18 Mhz; réalisable mais peu efficace au vu de la faible plage de variation des condensateurs actuels;
- stimuler l'interruption externe périodiquement via le BUS DCE pin 6; envisageable mais nécessitant une modification SOFT supplémentaire afin de dévier le " STACK OVERFLOW" encouru dans ce cas;
- la solution retenue : stimuler le signal HOLD du 8080 de façon périodique (et variable) afin de stopper le 8080 et ainsi le RA-LEN-TIR ! De plus, aucune modification coté SOFT. 'Le pied,quoi!'

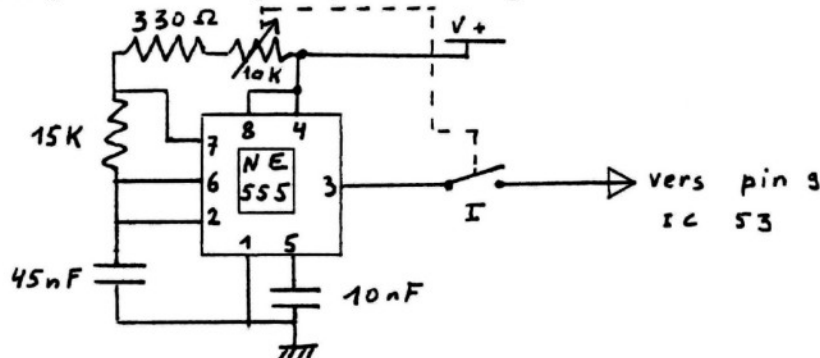
Voici le schéma interne du DAI de controle du signal HOLD :



On s'aperçoit aisément que le fait d'appliquer un niveau bas à l'entrée de la porte inverseuse de IC53 stoppe le 8080 tant que ce niveau reste présent. Par contre, seul le 8080 est influencé, donc :

- les interruptions via le IC109 fonctionnent toujours
- la fréquence d'horloge n'est pas modifiée
- aucune modification de l'affichage vidéo
- ...

Le petit montage électronique qui se chargera du travail est le suivant :



Le potentiomètre choisi est un modèle à interrupteur incorporé afin de pouvoir éventuellement restaurer la vitesse initiale (et légendaire!) du DAI.

Fonctionnement du montage :

- . si I est ouvert : l'état haut forcé par R122 à l'entrée de IC53 laisse fonctionner le 8080 sans ralentissement.
 - . si I est fermé : est appliqué à la pin 9 de IC53 le signal suivant issu du NE555.
-
- . si la résistance du potentiomètre diminue, Ta diminue et les temps d'arrêt du 8080 sont de plus en plus rapprochés: le DAI ralentit.
 - . si la résistance augmente, Ta augmente et les temps d'arrêt du 8080 sont de moins en moins rapprochés : le DAI augmente de vitesse.
 - . I devient nécessaire car même s'il y a arrêt une fois toutes les minutes, le DAI est quand même ralenti (hum...)

Ta : variable par 330Ω + pot. 10k
Tb : fixe par 15k

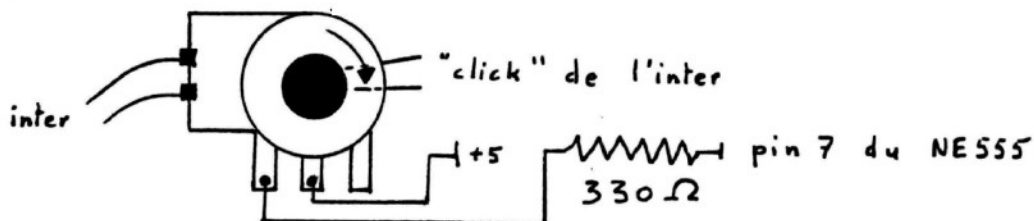
- la résistance de 330 Ohms (ou moins mais en aucun cas nulle) en série avec le potentiomètre sert à éviter l'arrêt complet du DAI en cas de résistance nulle du potentiomètre. C'est également cette résistance qui fixe la vitesse minimale pouvant être atteinte (vitesse et résistance varient dans le même sens).

Au vu de la simplicité extrême du montage, je ne vous fais pas l'injure de publier le tracé d'un éventuel circuit imprimé !

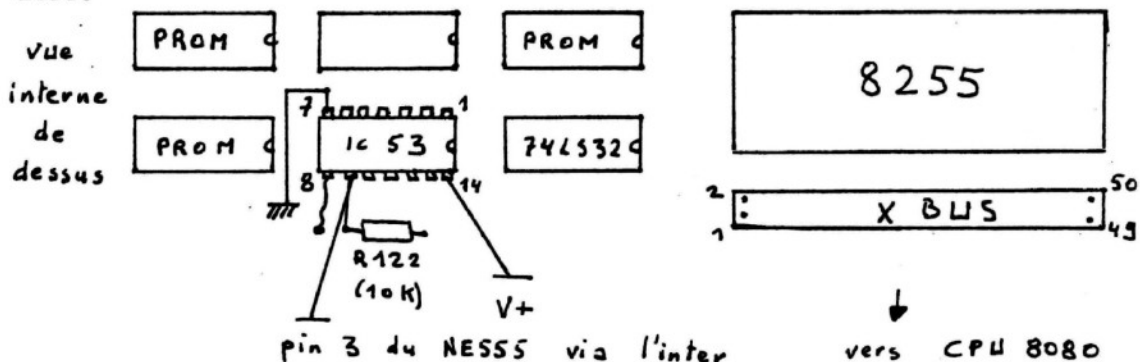
Par contre, ne pas se tromper dans la connexion du potentiomètre: il faut que I s'ouvre pour la résistance maximale du potentiomètre en bout de course.

Exemple : vue avant du potentiomètre (coté curseur)

.....



Les 3 fils de connexion du montage se soudent directement sur le IC53.



Quelques recommandations :

- Lors du chargement de K7,DISK (ou autre éventuellement), et avant un RESET ou un départ à froid:OUVRIR L'INTERRUPTEUR I (vitesse normale)
- Pour les possesseurs de KEN-DOS, vérifier que la carte interne branchée sur le X-BUS n'est connectée à la pin 49 (HOLD) de ce même X-BUS. S'il y a connexion (je n'ai pas pu le regarder car n'étant pas possesseur de KEN-DOS), mon montage risquerait de faire claquer le ou les circuits du KEN-DOS reliés à la pin 49, donc désolé pour vous ! Par contre les possesseurs de DCR sont à l'abri du danger ! Pour les possesseurs des drives du CBM64, procéder de la même manière que pour le KEN-DOS.

Pour essayer votre montage :

- départ à froid du DAI
 - en BASIC : 1 A = A+1 : PRINT A : GOTO 1
 - RUN ... et vous aurez la surprise de voir ralentir le programme en manoeuvrant le potentiomètre.
- Par contre, en mode direct, vous constaterez de visu que l'interruption 7 du curseur n'est pas altérée.

Après ces premiers pas, essayez un programme de jeu (rapide) par exemple PATROUILLEUR avec le ralenti: plus facile n'est-ce pas ?...

Pascal JANIN
Pour DAIClic 7/7/85

KEN-DOS : lecture du directory.

Cher DAIClic-istes, c'est à partir d'un petit programme de mon ami Alain Mariatte que je vais tâcher de vous expliquer comment lire et mettre dans une variable alphanumérique les noms de tous les fichiers se trouvant sur votre disquette.

Mais tout d'abord, à quoi peut bien servir une telle petite routine ?

Vous savez tous que lorsque vous 'deletez' plusieurs programmes d'une disquette (DIR+SHIFT+D) avant de la compacter, vous pouvez tous les faire revenir à leur état normal avec une seule commande 'REST'. Malheureusement, cela n'est pas le cas pour plusieurs autres commandes. Par exemple, la commande OPEN/CLOSE doit obligatoirement se faire manuellement et pour chaque programme. Pas question donc de faire simplement 'OPEN' pour 'ouvrir' tous les fichiers d'une disquette. C'est dans ce cas là qu'une telle routine est intéressante. En effet, il suffirait au programme d'aller mettre dans une variable le nom du fichier et puis, à l'aide du 'PRINT' et du 'CALLMHFOO3' (qui exécute ce 'PRINT'), faire ouvrir ou fermer tous les programmes d'une disquette. De même, pour des programmes que vous devez supprimer totalement de la disquette (soit 'DEL' + 'KILL'), vous êtes obligé de taper tous les DEL (SHIFT+D) depuis le directory et tous les 'KILL' entièrement à la main. Ne serait-il pas plus facile de mettre au point un programme qui passerait en revue les noms des fichiers de votre disquette et vous demandant chaque fois (à l'aide d'un simple GETC, par exemple) si oui ou non il faut supprimer totalement (DEL+KILL) ce fichier de la disquette.

Une telle routine peut également être intéressante pour mettre sur papier (et d'une manière bien tabulée) tous les programmes d'une disquette.

Mais, assez causé, entrons directement dans le vif du sujet, c'est-à-dire voyons 'comment ça marche ???'

Ce n'est pas compliqué. Il suffit de savoir comment le directory est structuré.

Sur le KENDOS (40 et 80 tracks), le directory est composé de 4 blocs de données qui peuvent contenir chacun jusqu'à 32 noms de fichiers différents. Un bloc comporte 1024 bytes de long (soit 1K, soit un secteur de la disquette.). Pour vous donner son directory, le KENDOS lit donc un secteur sur la disquette, le place à partir de HAF50 jusque HB34F et l'analyse. Mais comment fait-il pour l'analyser? Ce n'est de nouveau pas très compliqué ! Pour chaque nom de fichier, 32 bytes sont réservés dans ce directory.

Voici comment ces 32 bytes sont répartis :

byte(s) :	donne(nt) :
0	- numéro de fichier (ou indique si le fichier est existant)
1	- longueur du nom du fichier. (max 14 caractères)
2-15	- nom du fichier en ASCII (voir byte 2)

- 16 - statut du fichier :
 - 0 = OPENED
 - 1 = LOCKED
 - H10 = DELETED
- 17 - type de fichier :
 - 0 = BASIC (BAS)
 - 1 = BINAIRE (UTY)
 - 2 = FPT ARRAY (!ARY)
 - H12 = INT ARRAY (%ARY)
 - H22 = ALPHA ARRAY (\$ARY)
 - 3 = SOURCE DNA (SRC)
 - HF6 = SPL SOURCE (DBS)
 - HF4 = RND ou PASCAL (RND)
 - le reste : fichier RND.
- 18-19 - adresse de départ du pgm.
- 20-21 - longueur du pgm.
- 22-23 - adresse de lancement (ou, longueur du TEXTBUFFER s'il s'agit d'un fichier BASIC).

Les bytes suivant vous permettent de trouver le CODE et les DATES sous lesquels les fichiers sont enregistrés.

Pour ce qui est du décodage des DATES, je vous laisse vous amuser par vous même, c'est tordu !

Pour ce qui est du CODE (attention les pirates, mettez vos lunettes !), c'est bien simple, il suffit d'employer la petite astuce qui suit :

```

X=PEEK (BYTES 24)
Y=PEEK (BYTES 25)
CODE=256*(Y-HD)+X-HD

```

Si le fichier n'est pas protégé par un CODE, la somme de X et Y sera égale à zéro. Une astuce à ne pas mettre entre toutes les mains...mais maintenant il est peut-être un peu tard.

Le petit programme OPEN/CLOSE qui suit cet article n'est qu'une 'mise en musique' de tout ce que je viens de vous expliquer jusqu'à présent.

Pour l'employer, vous devez d'abord charger le bloc du directory que vous désirez 'étudier' et lancer le programme par un RUN 100. Le programme vous demandera alors de quel drive vient ce 'directory' ainsi que la commande à exécuter sur tous les fichiers de ce disque. C'est une application très simple de ce que je viens de vous expliquer, mais elle peut être approfondie dans d'autres directions et employée avec d'autres commandes différentes de OPEN et CLOSE.

D'autres possibilités sont envisageables, pourquoi ne pas charger par programme ce directory en mémoire. Cela serait bien plus agréable de faire cela que de devoir le charger à la main, en sortir et lancer la routine de recherche. (comme dans OPEN / CLOSE qui ne possède pas de chargement automatique).

De nouveau, ce n'est pas compliqué quand on sait où se trouvent les différents blocs de directory sur la disquette !

BLOC	TRACK	SECTEUR
1	0	3
2	0	5
3	0	2
4	0	4

Nous pourrions donc charger en mémoire (par exemple en H5000) le bloc 1 du directory de la disquette se trouvant dans le drive 2 à l'aide de la commande 'MANUAL'. Cela se ferait (dans l'exemple choisi) de la manière suivante : MANUAL 2 "R,HO,H3,H5000"

La recherche du nom de fichier se ferait donc de H5000 à H53FF plutôt que de HAF50 à HB34F. Simple non ?

Attention , lors de la conception de votre programme, n'oubliez pas de faire (comme à la ligne 120 du programme OPEN / CLOSE) une petite routine qui vérifie si il y a encore des fichiers dans la suite du directory. Si vous ne faites pas cela et que le directory de votre disquette n'est pas plein, vous risquez d'obtenir des résultats plus que bizarres. Attention également à ceci : si vous ne trouvez plus de titre à la fin d'un bloc de directory, cela ne veut pas dire qu'il n'y a plus d'autres programmes sur la disquette. Vous devez également 'checker' le bloc de directory suivant pour voir si il n'y a plus rien dedans. En effet, si vous supprimez un programme (DEL+KILL) du bloc directory 1, il n'y aura plus que 31 noms dans votre bloc 1. Cela ne veut pas dire pour autant qu'il n'y a rien dans votre bloc 2 ! Plus clairement : si vous supprimez un nom du bloc 1 (en supprimant un programme), le premier nom du bloc 2 restera à sa place et ne passera pas en position 32 du bloc 1. Il restera donc une place libre en place 32 du bloc 1...et votre programme pourrait croire qu'il n'y a plus rien après puisque le dernier nom qu'il trouve est le 31ème nom du bloc 1... Vous me suivez ?



Ultime finesse : pourquoi ne pas nous offrir en supplément le nom (donné avec la commande NAME) de la disquette ? Il suffit de charger en mémoire le secteur 1 de la piste 0. Plaçons ce secteur en H4000 par exemple. Pour charger ce secteur, cela donne simplement MANUAL drive "R,HO,H1,H4000".

En H4000, vous trouverez la longueur du nom de votre disquette. Rappelons que ce nom peut faire maximum 31 caractères. Les valeurs ASCII de ce nom se trouvent donc de H4001 à H4001+PEEK(H4000). Ces valeurs ASCII peuvent être mise dans une variable par une simple boucle FOR/NEXT, de la même manière que les noms de fichiers. (voir programme OPEN/CLOSE, ligne 140.)

Voilà, cela vous laisse déjà pas mal d'amusements en vue avec la lecture du directory de vos disquettes !

Marc Vandermeersch.
DAIC Bruxelles pour DAIClic.

N.B : Je tiens à remercier Alain Mariatte pour son programme OPEN / CLOSE ainsi que Messieurs les crackeurs DAIBolik et DAIBolo (ou Mesdames les crackeuses, que sais-je ?) pour leur programme KENZAP qui m'a permis de retirer certains renseignements supplémentaires sur la structure du directory, en particulier sur la manière de retrouver le CODE qui protège un fichier de la disquette.


```

1  REM #####
2  REM Fr                               Fr
3  REM Fr      OPEN / CLOSE fichier   Fr
4  REM Fr (c) Alain Mariatte & DAIClic Fr
5  REM Fr                               Fr
6  REM Fr  Programme fonctionnant sur Fr
7  REM Fr      tout systeme KEN-DOS.   Fr
8  REM Fr                               Fr
9  REM #####
10 MODE 0:PRINT CHR$(12):COLORT 8 0 0 0
20 PRINT "OPEN / CLOSE FILES"
30 PRINT "====="
40 PRINT :PRINT :PRINT
50 CALLM FrFOOO:REM BUF
60 PRINT "avant de commencer, faites le DIR choisi"
70 PRINT "PUIS RUN 100":STOP
100 INPUT "No de drive :";DRIVE:PRINT
105 INPUT "OPEN ou CLOSE (O/C)";MESSAGE$:PRINT
106 VERIF$=LEFT$(MESSAGE$,1)
107 IF VERIF$("<"O" AND VERIF$("<"o" AND VERIF$("<"C" AND VERIF$("<"c" THEN 105
108 IF VERIF$="C" OR VERIF$="c" THEN COMMANDE$="CLOSE":GOTO 110
109 COMMANDE$="OPEN"
110 FOR I=FrAF50 TO FrB34F STEP 32
120 IF PEEK(I+1)=0 THEN END
125 IF PEEK(I+16) IAND 16=16 THEN 200
130 NOM$="":LONGUEUR=PEEK(I+1)
140 FOR J=1 TO LONGUEUR:NOM$=NOM$+CHR$(PEEK(I+1+J)):NEXT
150 PRINT COMMANDE$;DRIVE;CHR$(Fr22);NOM$;CHR$(Fr22);:CALLM FrFOO3:PRINT
200 NEXT
1000 REM . #####
1010 REM . COMMANDE$ = COMMANDE A EXECUTER
1020 REM . NOM$      = NOM DU FICHER SUR LA DISQUETTE
1030 REM . LONGEUR   = NOMBRE DE CARACTERES QUE FAIT NOM$
1040 REM . DRIVE     = DRIVE AVEC LEQUEL ON TRAVAILLE
1050 REM . VERIF & MESSAGE = VARIABLES EMPLOYEES POUR
1060 REM .           VERIFIER ET METTRE UNE COMMANDE
1070 REM .           DANS LA VARIABLE COMMANDE$
1080 REM . #####

```

Truc KEN-DOS.

Vous aurez tous remarqué que lorsque vous essayez d'exécuter une routine GETC après une commande KEN-DOS, vous obtenez parfois de drôles de résultats.

Par exemple :

```

10 CALLMHFOOO:REM DELETE "TOTO"
20 G=GETC:IF G=0 THEN 20
30 GOTO 20

```

Si dans cette boucle GETC, vous pressez la touche 'RETURN', vous obtiendrez dans la plupart des cas un superbe 'SUBSCRIPT ERROR'. De même, les possesseurs du KENDOS V3.4 voient également leur DAI 'se planter' lors de l'emploi de la touche TAB dans cette même boucle GETC. Les possesseurs de KENDOS V3.7 auront également une surprise en pressant TAB : l'écran s'effacera mais la boucle GETC continuera à tourner.

Pour remédier à cette erreur, il suffit de faire un POKEH118,HFF après chaque commande KENDOS.

Ce petit truc est (paraît-il) mentionné à la page 28 du nouveau manuel du KEN-DOS...mais ce sont seulement les tous derniers acheteurs de KEN-DOS qui possèdent ce manuel !

Marc Vandermeersch, DAIC Bruxelles pour DAIClic.

Le BASIC du DAI bien que remarquable sous certains aspects semi-compilation, graphisme, son) ne disposait cependant pas de facilités que l'on rencontre actuellement sur pratiquement toutes les machines comme:

IF...THEN... ELSE, PRINT USING, ON ERROR GOTO, DEF FN...

Ces lacunes sont maintenant comblées avec un logiciel 'extraordinaire': DBASIC, qui ajoute environ 90 possibilités nouvelles au jeu d'instructions existant sans en supprimer !
Jugez-en:

-utilisation d'étiquettes en remplacement des numéros de lignes
ex ON I GOTO "DEBUT,"TRAITEMENT,"FIN

-lecture de DATA à partir de n'importe quelle ligne

-structuration par blocs et non par ligne pour
les tests conditionnels IF ... THEN
ELSE
END IF
les boucles WHILE... DO REPEAT....
WEND UNTIL...

-puissante gestion d'erreurs avec ON ERROR GOTO, RESUME, ERROR, ERR

-possibilité de définir des procédures ou des fonctions utilisables en mode direct ou par programme. DBASIC permet le transfert de variables, de constantes de tableaux ou de fonctions entre le programme appelant et la procédure (ou fonction). Celles-ci forment des blocs structurés admettant des variables locales (définie uniquement dans la procédure et nulle part ailleurs) ainsi que la récursivité (la propriété de s'appeler elle-même)

DBASIC vous apporte en plus un total de 56 messages d'erreur ainsi que la possibilité d'utiliser des fichiers d'extension (type \$) qui vous permettent d'augmenter ou de diminuer à votre guise la taille et donc la puissance de ce logiciel. Deux extensions sont disponibles à ce jour :

- \$DCR pour gérer un lecteur de cassettes digitales MEMOCOM DCR
- \$DIO pour lire et écrire par programme une partie de la mémoire grâce aux instructions DLOAD et DSAVE.

Le DAI dispose maintenant, avec le BBC, de l'un des plus puissants BASIC existant sur micro-ordinateurs !!!

Deux points noirs seulement:

la programmation en DBASIC ressemble à celle du PASCAL et impose d'abandonner certaines habitudes pour en acquérir d'autres, plus efficaces;

la taille de ce logiciel (environ 10K en mémoire vive) pourrait poser des problèmes lors de son utilisation avec les modes graphiques.

Auteur du programme : Willy COREMANS du club DAInamic.

Vous pouvez vous procurer le programme par l'intermédiaire d'IDC pour un prix de 2100 FB sur cassette audio et de 2350 FB sur mini-cassette digitale.

Hardware : carte d'extension mémoire .

Comme promis dans le numéro précédent, voici le schéma technique de l'extension mémoire (source R.F.A.) légèrement réadapté.

De façon simplifiée, le principe de travail de la mémoire d'extension est basé sur le 'BANKING' de blocs de mémoire dans la zone allant de : #0000-#BFFF et ce par autorisation ou blocage de chargement ou de lecture de la MEM. d'extention.

- La Carte Mère comporte une unité de sélection : le 8255 bien connu, et des circuits de décodage d'adresses.

- La carte RAM / EPROM peut recevoir 6 UNITES : des EPROMS 2716 (2K) ; 2732 (4K) ; 2764 (8K) ET OU des RAMS Statiques du type 6116 2K / 6264 8K ! Soit un total de 48 K Bytes !!!

- A partir de la Carte Mère, on peut empiler ainsi jusqu'à 8 cartes mémoires de 48 K bytes maximum chacune.

Ce qui nous porte à un TOTAL : 384 K BYTES.

- 1) Le décodage sur la Carte Mère sélectionne celle-ci à l'adresse #F900.
- 2) Le mot d'ordre se donne à #F903
Ex: POKE #F903, #80 -> Les PORTS B et Clow se positionnent en SORTIE donc la MEM. est LUE.
- 3) Un POKE en #F901 ira DESELECTIONNER un certain nombre de BLOCS de RAM ou EPROM de 8 KB.
- 4) Les ZONES de TRAVAIL sont :

BLOC	ZONE #	No de Bloc
1	0000 - 1FFF	1
2	2000 - 3FFF	2
3	4000 - 5FFF	4
4	6000 - 7FFF	8
5	8000 - 9FFF	16
6	A000 - BFFF	32

Il faut faire POKE #F901, 255 - NoB - NoB

Ex : vous désirez ELIMINER, BLOQUER les Blocs 2,4 et 5

POKE #F901, 255-2-8-16 -> SOIT POKE #F901, 229.

Les autres RESTENT en LIGNE.

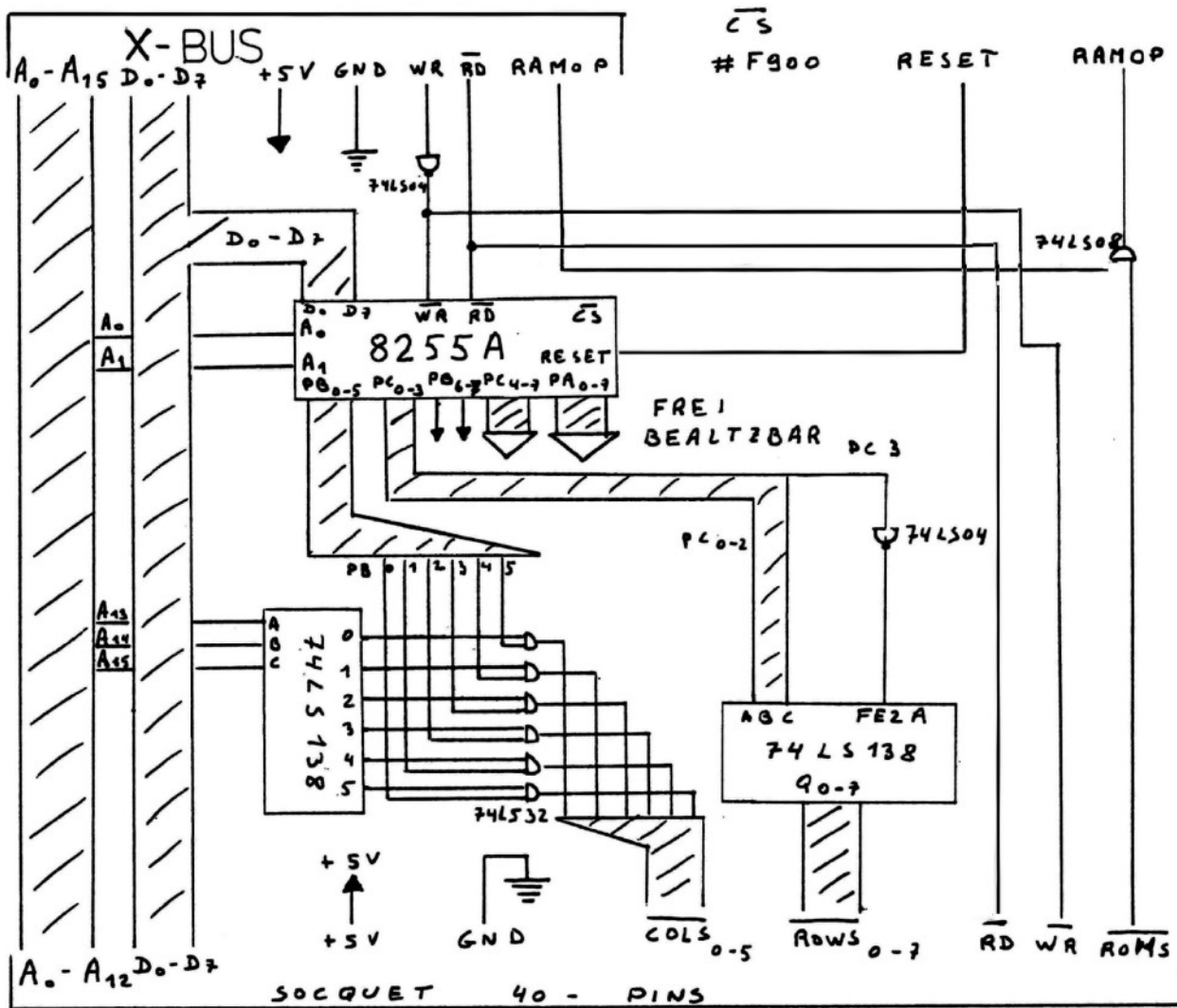
- 5) Choisissez la Carte RAM / EPROM à LIRE ou ECRIRE :
POKE #F902, 8 + No de carte 0 à 7 .
48 K à la fois !!!
Le BIT 3 à "8" autorise l'extension.
- 6) NB :Il est déconseillé, sauf précautions particulières, d'utiliser la ZONE 1 de 0000 - 1FFF car dans celle-ci se trouvent les pointeurs BASIC, les ADR. CURSEUR, les routines Restart, etc.
En LM on peut par exemple inhiber les interruptions (DI).
A part cela, toutes les zones de #2000 à #BFFF sont utilisables pour des programmes Basic.

Pour relier l'Extention Mémoire au DAI, on utilise un câble plat de 50 conducteurs. Quelques petites liaisons doivent encore être réalisées :

- 1) Une ligne RESET : raccord à la pin 1 du 8224
- 2) Une ligne CS (neg) pour l'adressage entre #F900 à #F9FF, relier à la pin 10 de IC 45 (74LS155)
- 3) Une ligne RAMOP (pas la même que celle du X-BUS).
Modification voir figure 6 : pin 11 du IC 44 (74S288).
- 4) Une petite coupure doit être faite quelques mm avant la pin 11 de IC44. A cette pin se relie RAMOP qui va vers la Carte Mère.

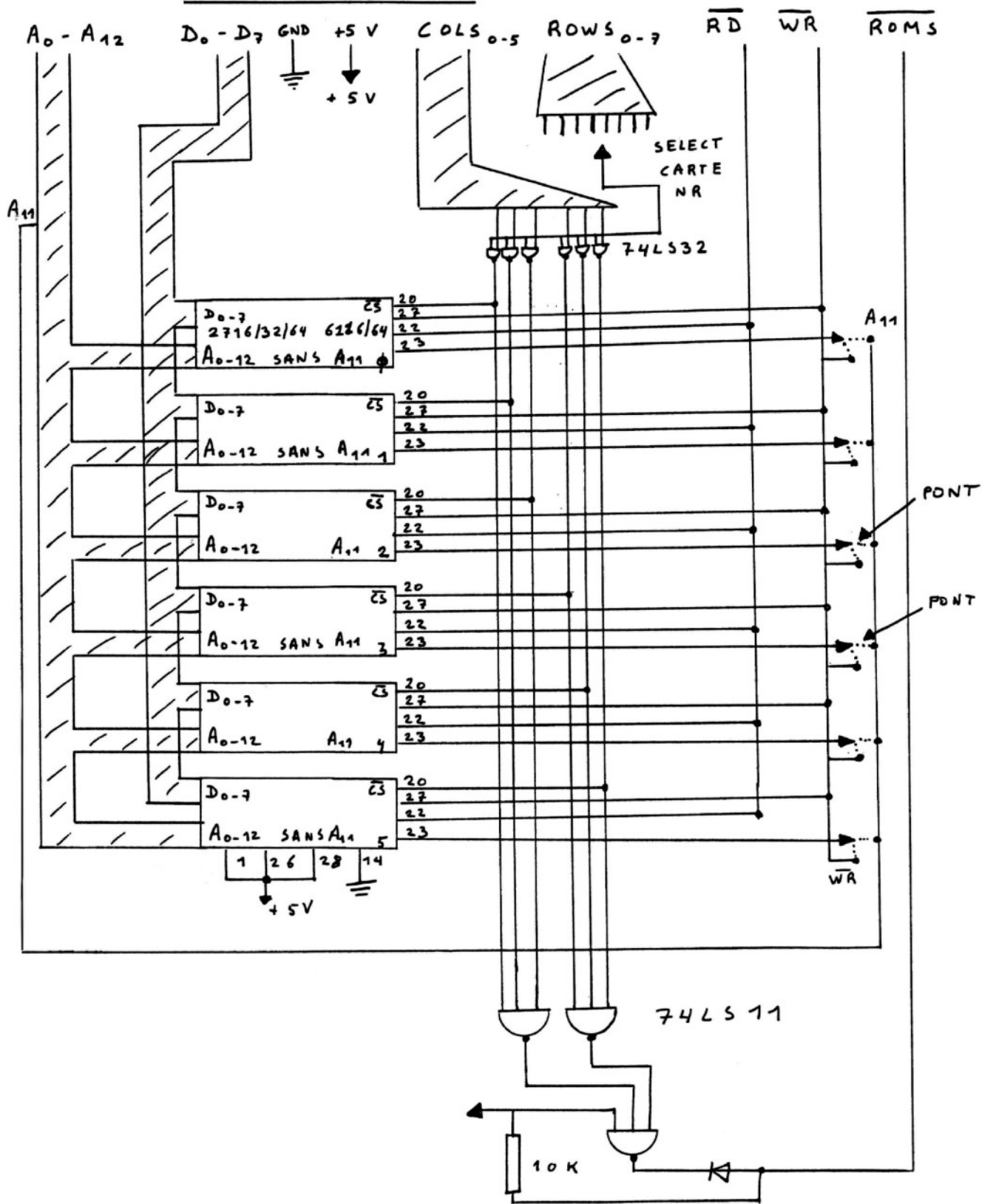
M.E.U. Main Board

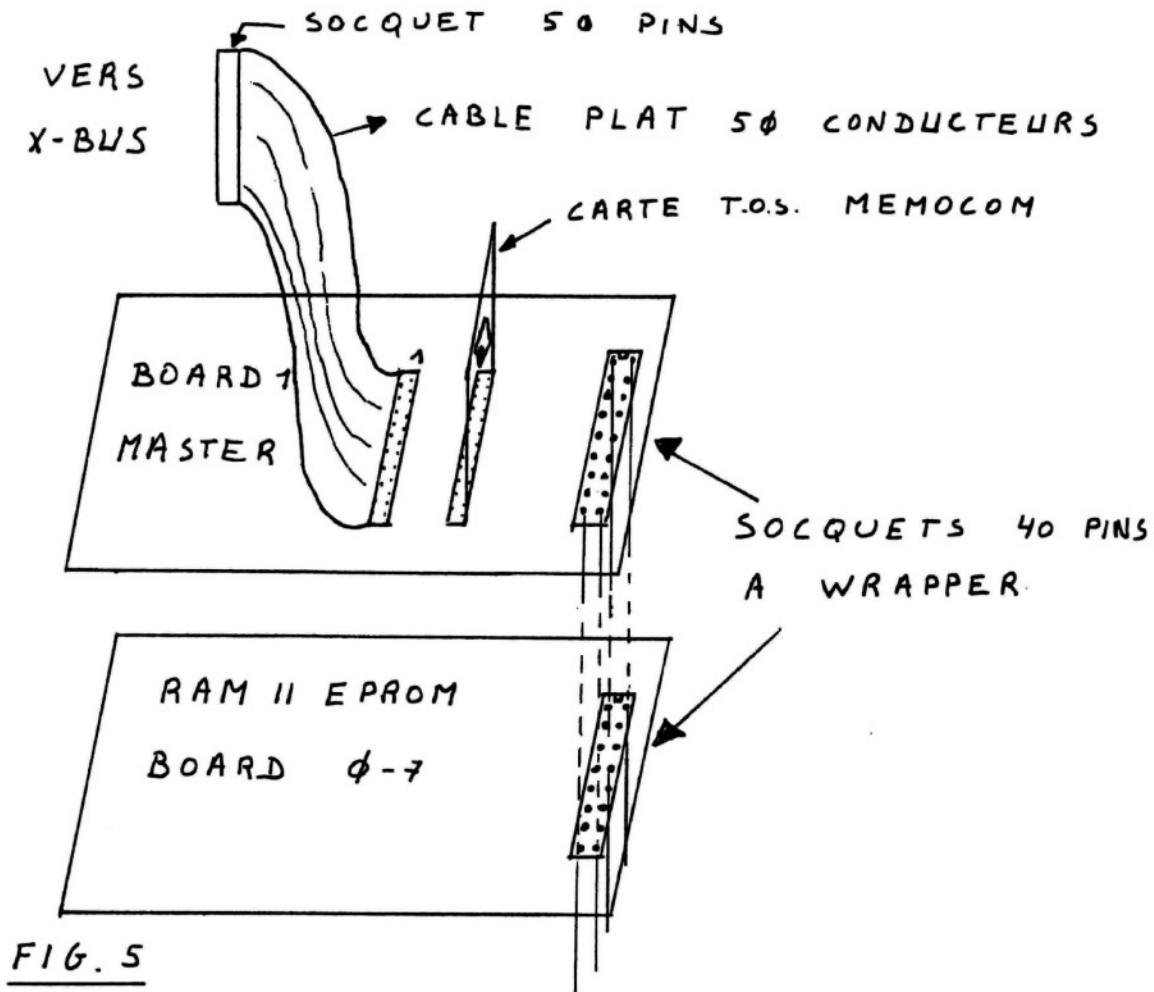
(Mem.Ext.Unit)



1-8	:	D ₀ - D ₇	27	:	<u>ROWS</u> ₁	35	:	<u>ROWS</u> ₅
9	:	<u>RD</u>	28	:	<u>COLS</u> ₄	36	:	<u>COLS</u> ₂
10-12	:	A ₁₀ - A ₁₂	29	:	<u>ROWS</u> ₂	37	:	<u>ROWS</u> ₆
13-22	:	A ₉ - A ₀	30	:	<u>COLS</u> ₀	38	:	<u>ROWS</u> ₈
23	:	GND	31	:	<u>ROWS</u> ₃	39	:	<u>ROWS</u> ₇
24	:	+5V	32	:	<u>COLS</u> ₃	40	:	<u>WR</u>
25	:	<u>ROWS</u> ₀	33	:	<u>ROWS</u> ₄			
26	:	<u>COLS</u> ₅	34	:	<u>COLS</u> ₁			

EPROM / RAM CARD





CÔTÉ SOUDURES PLATINE DA1

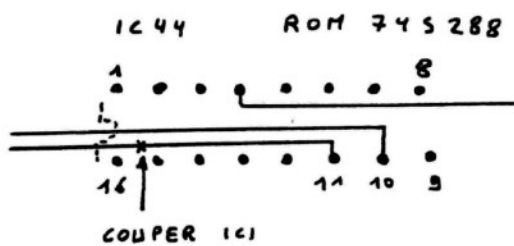


FIG. 6



Tout ceci étant fait, on dispose d'une belle banque de RAMS et ROMS dans laquelle le directeur peut stocker ou se servir et cela à l'abri des impôts, ce qui est rare ! La cotation du Byte est élevée.

ERIC NEVE 'S.EL.S.'
 25 Hoogstraat 1512 DWORP
 Tél. (0)2 / 380.35.81

Club DAIC pour DAIClic

TEST

BASIC V1.2

Monsieur TEGETOFF, membre du club DAINAMIC allemand, est le réalisateur de BASIC V1.2 . Ce n'est nullement un nouveau basic mais une amélioration du V1.0 ou V1.1. Ceci en essayant de rester, au maximum compatible tant en BASIC qu'en MLP avec les machines n'étant pas équipées de BASIC V1.2 . Le soft se trouve dans trois EPROMs (2764) munies d'adaptateurs qu'il suffit de loger dans les supports à la place des anciennes ROMs. Pour les heureux possesseurs d'un programmeur d'EPROMs le soft est disponible sur K7.

A la commande on peut choisir sur:

- un autre texte que "DAI PERSONNAL COMPUTER"
- ainsi que sa couleur
- un autre COLORT que 8 0 0 0
- un autre COLORG que 0 5 10 15
- un autre baud rate
- la redéfinition du clavier

Pour ma part j'ai commandé les EPROMs en standard c'est à dire sans rien y changer . Voici donc ce que l'on obtient :

LE BASIC

1) A l'allumage, en cas de reset ,d'appui sur BREAK,après EDIT ou en cas d'erreur le vecteur F131 sera toujours remis à 1,ce qui réjouira bon nombre de possesseurs d'imprimantes.

2) La routine de scrutation du clavier a été modifiée :la touche CTRL a une double fonction . Pour le "CAPS-CHANGE" il faut agir sur SHIFT + CTRL. CTRL + autre touche permet maintenant d'afficher, sans passer par l'inévitable PRINT CHR\$(x) ,les codes suivants:

CTRL + TOUCHE = CHR\$!	CTRL + TOUCHE = CHR\$	
0	176 !	1	124
2	140 !	3	127
4	64 !	5	28
6	182 !	7	96
8	123 !	9	125
/	92 !	:	186
[27 !	\$	29
^	30 !	-	31
A...Z	1..26 !	a...z	1..26

SHIFT+0 = CHR\$(95)

([et] =Touche à gauche de DEL)

Un petit exemple:

au lieu de frapper ? CHR\$(27) ; "R"; CHR\$(3)

Il suffira d'écrire ? "xRy" où x=CTRL+ [et y=CTRL+C.

Il faut noter que sous EDIT ou BASIC tous les caractères sont acceptés.

3) LES COMMANDES

NEW, RUN, RUN x, UT et CHECK sont autorisées dans un programme.
Instructions raccourcies:

```
pn = POKE Fr131,n
l..= LIST...
e..= EDIT...
wn = WAIT TIME n
"....." = LOAD ".....":RUN
```

Toutes ces instructions sont bien sûr programmables excepté "....." et lors d'un listing elles apparaîtront en "clair" TALK a été supprimé, car très peu de programmes utilisent TALK, mais peut-être listé et édité.

PAUSE a été créé :

```
FOR N=1 TO 100: ? N, SQR(N) :PAUSE:NEXT N
```

-Par une action sur une touche le programme s'arrêtera et attendra une pression sur SPACE pour redémarrer. Pour les machines non équipées de V1.2, le programme fonctionnera aussi mais sans possibilité d'arrêt à part BREAK ou la fin de la boucle.

MODE 7, MODE 7A, MODE 8 et MODE 8A

-Ces modes sont maintenant acceptés sans passer par un MLP. Pour la compatibilité entre machines il faudra taper:

```
IF PEEK(FrC7DD) <> 50 alors passer par mlp pour obtenir MODE 7 à 8
```

-La résolution est de 512 x 240.

-Pour les connaisseurs: la zone mémoire "temporary save area" a été supprimée. Cela donne 3500 octets utilisateur en plus (en MODE 5, 6), un passage plus rapide "full graphic/graphic-texte" et un petit inconvénient: lors de ces passages l'écran "s'irise" une fraction de seconde, mais on s'y fait vite.

4) RUN initialise les FOR et les GOSUB ce qui n'était pas le cas.

5) LOADA, SAVEA, HEAP...

-Un read close est effectué et le K7 sera stoppé en cas d'erreur.

-Les variables "vides" ne sont plus placés dans la HEAP ce qui donne un gain de place et de temps.

-L'initialisation des variables ne touche que la "symbol table".

6) LOAD

-Le LOAD accepte une variable alpha. pour le nom du fichier.

EX: A\$="TEXTE":LOAD A\$

7) CHECK

-CHECK accepte tout les 'file type'

8) IMP

-IMP STR =IMP STR A-Z, IMP STR a-z, IMP INT c, IMP FPT c et IMP STR c sont enfin possibles.

9) RND(x)

-RND est plus rapide :7,8ms au lieu de 25,3ms.

10) IF...THEN

-THEN peut-être omis mais réapparaîtra lors d'un listing.

11) INPUT, READ

-INPUT et READ acceptent des expressions math. comme 3*PI, EXP(3,4)+..., XMAX-5...etc....

12) A la place de "C" indiquant un dépassement de ligne il y a un " ", ce qui donne un écran plus propre.

L'EDITEUR

-L'éditeur a subit une véritable cure de jeunesse.
-Qu'importe la taille du texte, le curseur se déplacera toujours avec la même rapidité du début à la fin de celui-ci.
Le texte s'affiche proprement en TAB(6) et, pour tromper l'attente, le passage EDIT/BASIC ou BASIC/EDIT, s'accompagne d'un nombre de "." équivalent au nombre de lignes.

Les nouvelles commandes:

CTRL+F1. gauche = saut au début de ligne
CTRL+F1. droite = saut en fin de ligne
CTRL+FL. haut = saut au début du texte
CTRL+F1. bas = saut en fin de texte
BREAK BREAK = retour à EDIT (pour éviter qu'un rebond de contact ne détruise tout le travail)

BREAK+DEL = BREAK+BREAK de V1.0 ou V1.1.

BREAK+M = MERGE

a) EDIT: LOAD". . . ." -BREAK M provoquera l'assemblage de 2 programmes avec priorité des anciens numéros de ligne en cas de numéros identiques.

b) Pour dupliquer des lignes :EDIT, on change les numéros des lignes à dupliquer puis BREAK M (remplace le BREAK BREAK POKE F135,2).

UTILITY

Comme les commandes s'effectuent de la même manière qu'avant j'utilise donc les abréviations du manuel page 106

Les commandes:

(B) :retour au BASIC

(H) :MODE 0

Touches CTRL+L:PRINT CHR\$(12)

(G)o :Gadr+SPACE ou Gadr+RETURN
-Z3 n'est plus nécessaire et le retour du programme appelé laissera l'écran intact.

(D)isplay :Dladr hadr+SPACE ou RETURN
-donne un listing hexa pouvant être manipulé comme un LIST en BASIC.

(I)nit memory :Iladr hadr byte+SPACE ou RETURN
-remplace Fill.

(M)ove :même fonction qu'avant.

(S)ubstitute :Sadr+SPACE ou RETURN
 -rempli la même fonction qu'avant mais si SPACE est utilisé un saut de ligne s'effectuera tout les huit octets avec indication de la nouvelle adresse. Pour sortir de Substitute BREAK ou Flèche gauche qui provoquera un Display de la zone mémoire touchée.

(V)ector :V+RETURN
 -donne un display de I,0 + les huit interruptions
 :VI ou VO ou V(0 à 7) + SPACE ou RETURN
 -permet d'agir comme un Substitute sur les vecteurs M, T et G ont été supprimés. I et O sont les vecteurs d'entrée et sortie .

e(X)amine reg. :X+RETURN
 -donne un Display des registres
 :XA ou XB,C,D,E,H,L,S ou P
 -permet d'agir comme un Substitute sur les registres

(Z)ero :Z+RETURN =Z3
 :ZV =initialisation des vecteurs
 :ZX =mise à 0 des registres et initialisation de SP et PC.

(W)rite :Wladr hadr \$......RETURN
 -\$ s'inscrit automatiquement après space et indique l'autorisation d'écrire du texte.

Remarques sur le texte:
 -Le texte est entre "... " signifiera que les codes ASCII auront la valeur Fr20 à Fr7f.
 -le texte est entre '...' signifiera que les codes ASCII auront la valeur FrB0 à FrFF.
 -Pas de guillemet signifiera que le texte sera constitué de codes (de Fr00 à FrFF).
 -Ces options sont mixables par ex.:\$C "texte" OD 'fichier" OO .
 -La taille du texte est limitée à 127 caracteres ou octets.
 -Ces remarques restent valables pour R, L, P, et F

(R)ead :R \$...ou Radr \$ ou R+RETURN
 - même fonction qu'avant mais le nom du fichier s'inscrit sur l'écran .

(L)oad :comme pour R
 -A la différence de R ,L300 où 300 n'est plus un offset mais l'adresse de départ du fichier.

(P)oke :Padr \$.....
 -Poke du texte des codes (127 maxi) voir Write.

(A)scii dump :Aladr hadr n
 -Donne un listing hexa+ASCII
 n=0 ou space =list des codes ASCII de Fr20 à Fr7F
 n=1 =list des codes ASCII de FrB0 à FrFF
 n=2 =list des codes ASCII de Fr00 à FrFF

(F)ind :Fladr hadr m n \$....
 -C'est une instruction superbe pour les bidouilleurs Elle permet de trouver , dans la dump , du texte, des

codes etc...

Ex:F300 1000 \$CD ? EF

Tous les Call EF xx se trouvant de F300 à F1000 seront listés avec indication de la place mémoire. (? est utilisé comme "jocker")

m=0 ou space à FFF indique le nombre d'octets à afficher avant et après le texte recherché.

n=0 ou space à 2 =n dans la commande A

n)2 donne un listing hexa uniquement

(L)ook a été supprimé car l'auteur considère cette instruction inutilisable.

-Le ? a été remplacé par des messages d'erreur

Ex:COMMANDE INVALID, SYNTAX ERROR ,TYPE MISMATCH et LOADING ERROR n .

CONCLUSION

Il y a encore beaucoup de choses à dire mais la moitié de la revue ne suffirait pas. Enfin pour terminer BASIC V1.2 est compatible avec 'KEN-DOS', tous les programmes BASIC fonctionnent exceptés ceux utilisant TALK (il y en a très peu), en MLP les seuls qui ne fonctionnaient pas (dans mon cas) sont ceux utilisant CALL FCD67 (TALK), mais un rajout d'une trentaine d'octets a tout remis en ordre.

BASIC V1.2 est disponible chez:

HEINRICH TEGETOFF
IN DER PROVITZE 87
4630 BOCHUM 1
(ALLEMAGNE)
Tél.0234/534350

PRIX: Les trois EPROMs avec adaptateurs: 114 DM
Le soft seul (sur K7) : 30 DM



JANVIER 1985

 D.A.I.C.L.I.C

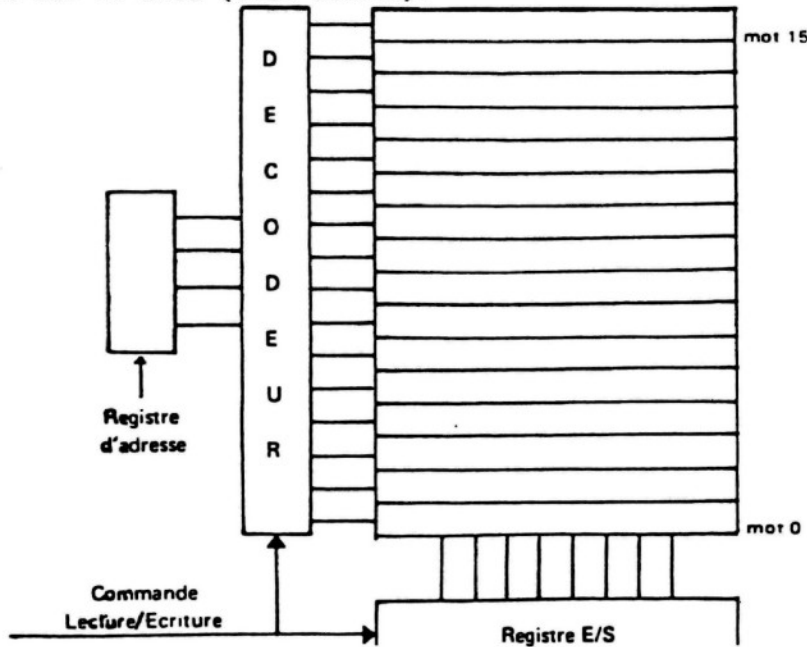
HP LEGRY
 par
 59500 DOUAI, FRANCE

Nous allons aborder sommairement l'architecture interne d'un microprocesseur. Ne vous attendez pas à trouver un plan détaillé. Cet article a pour but de donner des bases suffisantes en vue d'une initiation au langage assembleur.

I - LA MEMOIRE

Dans l'article MICROPROCESSEUR PART.1, nous avons vu le code binaire et hexadécimal. Ils sont fondamentaux.

Une mémoire est une juxtaposition de cases-mémoires. La mémoire du 8080A peut stocker des mots de 8 bits ou octets. Pour sélectionner une case mémoire et donc lire son contenu ou y écrire un octet, il faut en connaître la position, c'est à dire son adresse. Cette adresse est codée sur 16 bits (deux octets).



Le bus de donnée, par lequel transite la valeur lue ou à écrire dans les cases mémoires est de 8 bits (un octet). Celui des adresses est de 16 bits. Une commande ,lecture écriture (R/W), permet d'indiquer si la donnée est à lire ou à écrire, car le bus "entrées sorties" est unique.

A - LA MEMOIRE VIVE (RAM)

Une mémoire vive (random access memory), permet de lire des données et de les stocker. Vous pouvez lire et écrire dans ces cases.

B - MEMOIRE MORTE

a) ROM : read only memory

Avec ce type de mémoire, vous ne pouvez que lire les données contenues dans les cases mémoires. Le stockage est définitif. La mémoire, une fois remplie, ne peut être effacée.

b) EPROM

- - - - -

On ne peut que lire cette mémoire mais le stockage peut être effacé par l'utilisation d'ultraviolets et on peut ainsi la réutiliser.

Ce sont les principales mémoires utilisées. Quant aux autres, reportez vous aux ouvrages spécialisés.

2 - ARCHITECTURE DU MICROPROCESSEUR

A - LE MATERIEL

Un microprocesseur comprend essentiellement :

a) Unité arithmétique et logique

.....

Cette unité est un ensemble de circuits logiques capable de réaliser des fonctions logiques et arithmétiques. L'une des opérands sera toujours contenu dans un registre associé étroitement à l'UAL : l'accumulateur.

b) Les registres généraux (H,L,B,C,D,E)

.....

Ce sont des cases mémoires de 8 bits, parfois associés 2 par 2, que l'UAL utilise pour stocker temporairement des valeurs (opérands ou résultat).

Contrairement aux cases mémoires de la RAM, ils sont accessibles directement et n'ont pas d'adresses. Chacun d'entre-eux possède un nom (ou plutôt une lettre), reconnu par le microprocesseur.

c) L'accumulateur (A)

.....

L'accumulateur est un registre privilégié de l'UAL, qui y cherche un opérande et stocke généralement le résultat de ses opérations.

d) le registre d'état (PSW)

.....

Ce registre, 8 bits, indique l'état de l'octet contenu dans l'accumulateur à la suite d'une opération arithmétique ou logique.

CARRY CY : Bit mis à 1 lorsque le nombre dépasse 255 décimales (FFF)

Appelé aussi retenue, il indique l'overflow (dépassement).

PARITE P : bit mis à 1 si le nombre est pair

ZERO Z : bit mis à 1 si le nombre est nul

SIGNE S : bit mis à 1 si le nombre est négatif

Le "nombre" correspond au résultat d'une opération réalisée par l'UAL

e) Le compteur ordinal (PC)

.....

Le compteur ordinal contient l'adresse, dans la ROM ou la RAM, de la prochaine instruction à exécuter. Il est incrémenté automatiquement lors de l'exécution d'une instruction.

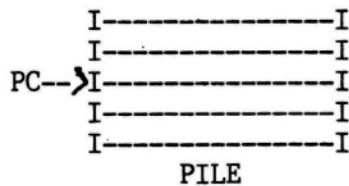
Il contient, si vous désirez faire un parallèle avec le basic, le numéro de la ligne à laquelle il doit aller (ce n'est qu'une image !!).

f) La pile

.....

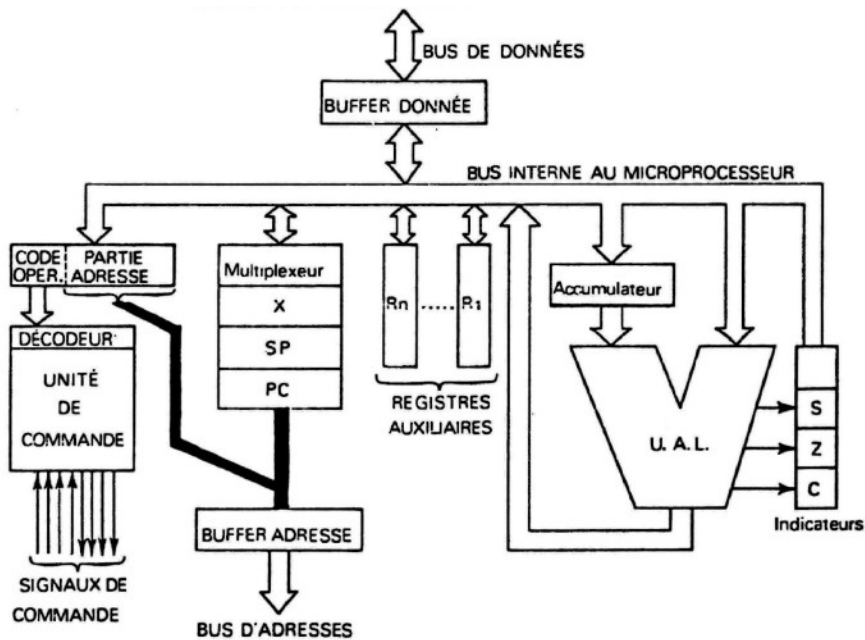
La pile est une petite partie réservée de la mémoire. Elle est utilisée pour stocker des adresses ou le contenu de registres que l'on veut sauvegarder. Si, par exemple, vous appelez un sous-programme, il faut conserver l'adresse à laquelle le programme doit continuer. Si, de plus, dans ce sous programme vous utilisez les registres dans lesquels il y a des valeurs que vous voulez retrouver quand le sous programme a été exécuté, il vous faut sauvegarder tout cela. La pile est conçue pour stocker facilement ces valeurs.

L'adresse de la première position mémoire libre dans la pile est stockée en permanence dans un registre appelé pointeur de pile (Stack Pointeur SP). Il est incrémenté par l'écriture d'une valeur dans la pile et décrementé par toute lecture de la pile. Ce type de pile est appelé LIFO (Last In, First Out). C'est à dire que la première valeur à sortir de la pile sera la dernière à y être entrée.



PILE

PLAN RECAPITULATIF



3 - MICROPROCESSEUR 8080A

3 Paires de registres	HL,BC,DE	!	1 pointeur de pile	SP
1 accumulateur	A	!	1 registre d'état	PSW
Données sur 8 bits		!	Adresse sur 16 bits	
Mémoire RAM	F0 à FBFFF	!	Pile F800 à F900	
Réservation interne	F0 à F2EC	!		

Ceci termine le second chapitre consacré au microprocesseur. J'espère qu'il vous a donné une idée de ce qu'il représente. Retenez leurs fonctions. Dans le chapitre suivant, nous attaquerons l'assembleur...

HP.LEGRY

BASIC

```

1      REM
2      REM
3      REM .....  P A C M A N (??!) .....
4      REM .....  pour DAI .....
5      REM
6      REM .....  B. CIMBALNIK .....
7      REM .....  Le 2 JUIN 85 .....
8      REM
9      REM
10     GOSUB 7000
30     ENVELOPE 0 2,8;4,12;8,16;8
40     ENVELOPE 1 4,8;8,12;16,13;0
50     SC=0:RC=0:VI=3:S1!=440.0:C!=2.0:K!=0.0
90     GOSUB 6000
100    IF PEEK(F556D)<>FCB THEN SOUND 2 1 15 0 FREQ(900.0):PRINT :
      PRINT "ERREUR DE DATA EN LM !":PRINT :SOUND OFF :STOP
110    RESTORE
120    READ TA!:IF TA!<>T! THEN 120
150    MODE 6A:COLORG 9 10 13 15:COLORT 9 9 9 9:PRINT CHR$(12)
200    FOR Y!=203.0 TO 0.0 STEP -18.0:FOR X!=0.0 TO 256.0 STEP 18.0
210    2  READ DE!
220    2  IF DE!=1.0 THEN FILL X!,Y! X!+18,Y!+18 15
230    2  IF DE!=0 THEN FILL X!+7,Y!+7 X!+11,Y!+11 13
240    NEXT:NEXT

499    REM (PL = No LUTIN),(PX = ABSCISSE),(PY = ORDONNEE),(AF =
      ADRESSE ROUTINE).
500    PL!=F5572:PX!=F5570:PY!=F5571:AF!=F5500
510    POKE PL!,1
520    X!=170.0:Y!=158.0
530    X1!=10.0:Y1!=158.0:X3!=X1!:Y3!=Y1!
700    CURSOR 5,3:PRINT "I SCORE I":CURSOR 25,3:PRINT "I RECORD
      I":CURSOR 45,3:PRINT "I VIES I"
720    CURSOR 8,2:PRINT SC:CURSOR 29,2:PRINT RC:CURSOR 49,2:PRINT VI
760    COLORT 9 14 9 9
800    GOSUB 2020:POKE PX!,X!:POKE PY!,Y!:CALLM AF!
820    CURSOR 15,1:PRINT "< SHIFT > pour commencer."
830    Z!=GETC:IF PEEK(F2B0)<>64 THEN 830
840    SOUND 2 1 15 0 FREQ(1000.0):CALLM AF!:CURSOR 15,1:PRINT "
      "

850    SOUND OFF :GOTO 1010

997    REM
998    REM OI boucle saisie clavier IO
999    REM
1000   IF S2!=1.0 THEN SOUND OFF :GOTO 4000
1003   IF X!218.0 THEN X!=236.0
1005   IF X!3239.0 THEN X!=19.0
1008   CALLM AF!
1010   POKE PY!,Y!:POKE PX!,X!
1020   CALLM AF!
1030   GOSUB 2000:Z!=GETC:IF PEEK(F2A9)=64 THEN GOSUB 5000:GOTO 1000
1040   IF PEEK(F2AA)=64 THEN GOSUB 5100:GOTO 1000
1050   IF PEEK(F2AB)=64 THEN GOSUB 5200:GOTO 1000
1070   IF PEEK(F2AC)=64 THEN GOSUB 5300:GOTO 1000
1080   GOTO 1030

1998   REM
1999   REM OI animation sprite 0 IO
2000   REM

```



```

2010  SOUND 0 0 10 2  FREQ(100.0):POKE PL!,0:POKE PY!,Y3!:POKE PX!,
      X3!:CALLM AF!
2020  POKE PL!,0:POKE PY!,Y1!:POKE PX!,X1!
2030  CALLM AF!
2040  X3!=X1!:Y3!=Y1!:SOUND 0 OFF
2045  IF Y1!<Y! THEN GOSUB 5500:GOTO 2080
2050  IF Y1!>Y! THEN GOSUB 5600:GOTO 2080
2060  IF X1!<X! THEN GOSUB 5700:GOTO 2080
2070  IF X1!>X! THEN GOSUB 5800:GOTO 2080
2080  IF (X1!+8.0<X!+16.0 AND X1!+8.0>X!) AND (Y1!+8.0<Y!+16.0 AND
      Y1!+8.0>Y!) GOTO 9000
2083  IF X1!<=18.0 THEN X1!=235.0
2085  IF X1!>=240.0 THEN X1!=19.0
2090  POKE PX!,X!:POKE PY!,Y!:POKE PL!,1:SOUND 0 OFF :RETURN

2997  REM
2998  REM OI affichage score IO
2999  REM
3000  SOUND 2 1 15 3 FREQ(700.0):SC=SC+5:CURSOR 8,2:PRINT SC:IF
      RC<SC THEN RC=SC:CURSOR 29,2:PRINT RC
3010  IF SC>=S1! THEN S1!=S1!+440.0:S2!=1.0
3900  SOUND OFF :RETURN

3997  REM
3998  REM OI partie gagnee IO
3999  REM
4000  SOUND OFF :S!=34.0
4005  SOUND 2 1 15 3 FREQ(S!):S!=S!+1.0:IF S!^2=200.0 THEN 4010
4007  GOTO 4005
4010  CURSOR 15,1:PRINT "PARTIE GAGNEE !!":SOUND OFF
4020  C!=C!+1.0:IF C!>=7.0 THEN C!=7.0
4050  S2!=0.0:K!=0.0
4090  WAIT TIME 150:CALLM AF!:POKE PY!,Y!:POKE PX!,X!:POKE PL!,1:
      CALLM AF!
4092  CALLM AF!:FILL X1!-7,Y1!-50 X1!+24,Y1!-18 9
4095  CURSOR 15,1:PRINT " "
4100  RESTORE
4110  READ TA!:IF TA!<>T! THEN 4110
4120  GOTO 200

4997  REM
4998  REM OI controle sprite 1 IO
4999  REM
5000  IF SCRNX(X!+1,Y!-25)=15 OR SCRNX(X!+15,Y!-25)=15 THEN RETURN
5030  IF SCRNX(X!+8,Y!-23)=13 THEN FILL X!+1,Y!-20 X!+14,Y!-26 9:
      GOSUB 3000
5050  Y!=Y!+6.0
5080  RETURN
5100  IF SCRNX(X!+1,Y!-49)=15 OR SCRNX(X!+15,Y!-49)=15 THEN RETURN
5130  IF SCRNX(X!+8,Y!-47)=13 THEN FILL X!+1,Y!-50 X!+14,Y!-44 9:
      GOSUB 3000
5150  Y!=Y!-6.0
5180  RETURN
5200  IF SCRNX(X!-6,Y!-42)=15 OR SCRNX(X!-6,Y!-28)=15 THEN RETURN
5230  IF SCRNX(X!-4,Y!-35)=13 THEN FILL X!-1,Y!-30 X!-7,Y!-40 9:
      GOSUB 3000
5250  X!=X!-6.0
5280  RETURN
5300  IF SCRNX(X!+20,Y!-42)=15 OR SCRNX(X!+20,Y!-28)=15 THEN RETURN
5330  IF SCRNX(X!+20,Y!-35)=13 THEN FILL X!+23,Y!-30 X!+17,Y!-40 9:

```



```

      GOSUB 3000
5350  X1=X1+6.0
5380  RETURN

5497  REM
5498  REM OI   controle sprite 0   IO
5499  REM
5500  IF SCRN(X1!+1,Y1!-28)=15 OR SCRN(X1!+15,Y1!-28)=15 THEN GOTO
5550
5520  Y3=Y1!:Y1=Y1!+C!
5540  RETURN
5550  IF SCRN(X1!-2,Y1!-42)⟨⟩15 OR SCRN(X1!-2,Y1!-28)⟨⟩15 THEN X3!=
X1!:X1=X1!-C!:RETURN
5555  IF SCRN(X1!+19,Y1!-42)⟨⟩15 OR SCRN(X1!+19,Y1!-28)⟨⟩15 THEN
X3=X1!:X1=X1!+C!:RETURN
5560  IF SCRN(X1!+1,Y1!-45)⟨⟩15 OR SCRN(X1!+15,Y1!-45)⟨⟩15 THEN Y3!=
Y1!:Y1=Y1!-C!:RETURN
5580  RETURN

5599  REM
5600  IF SCRN(X1!+1,Y1!-45)=15 OR SCRN(X1!+15,Y1!-45)=15 THEN GOTO
5650
5620  Y3=Y1!:Y1=Y1!-C!
5640  RETURN
5650  IF SCRN(X1!-2,Y1!-42)⟨⟩15 OR SCRN(X1!-2,Y1!-28)⟨⟩15 THEN X3!=
X1!:X1=X1!-C!:RETURN
5655  IF SCRN(X1!+19,Y1!-42)⟨⟩15 OR SCRN(X1!+19,Y1!-28)⟨⟩15 THEN
X3=X1!:X1=X1!+C!:RETURN
5660  IF SCRN(X1!+1,Y1!-28)⟨⟩15 OR SCRN(X1!+15,Y1!-28)⟨⟩15 THEN Y3!=
Y1!:Y1=Y1!+C!:RETURN
5680  RETURN

5699  REM
5700  IF SCRN(X1!+19,Y1!-42)=15 OR SCRN(X1!+19,Y1!-28)=15 THEN GOTO
5750
5720  X3=X1!:X1=X1!+C!
5740  RETURN
5750  IF SCRN(X1!+1,Y1!-28)⟨⟩15 OR SCRN(X1!+15,Y1!-28)⟨⟩15 THEN Y3!=
Y1!:Y1=Y1!+C!:RETURN
5755  IF SCRN(X1!+1,Y1!-45)⟨⟩15 OR SCRN(X1!+15,Y1!-45)⟨⟩15 THEN Y3!=
Y1!:Y1=Y1!-C!:RETURN
5760  IF SCRN(X1!-2,Y1!-42)⟨⟩15 OR SCRN(X1!-2,Y1!-28)⟨⟩15 THEN X3!=
X1!:X1=X1!+C!:RETURN
5780  RETURN

5799  REM
5800  IF SCRN(X1!-2,Y1!-42)=15 OR SCRN(X1!-2,Y1!-28)=15 THEN GOTO
5850
5820  X3=X1!:X1=X1!-C!
5840  RETURN
5850  IF SCRN(X1!+1,Y1!-25)⟨⟩15 OR SCRN(X1!+15,Y1!-25)⟨⟩15 THEN Y3!=
Y1!:Y1=Y1!+C!:RETURN
5855  IF SCRN(X1!+1,Y1!-47)⟨⟩15 OR SCRN(X1!+15,Y1!-47)⟨⟩15 THEN Y3!=
Y1!:Y1=Y1!-C!:RETURN
5860  IF SCRN(X1!+19,Y1!-42)⟨⟩15 OR SCRN(X1!+19,Y1!-28)⟨⟩15 THEN
X3=X1!:X1=X1!-C!
5880  RETURN

5997  REM
5998  REM OI   choix du tableau   IO

```

```

5999 REM
6000 MODE 0:PRINT CHR$(12):COLORT 9 14 9 9:POKE F75,95:RESTORE
6010 CURSOR 7,20:PRINT "CHOISISSEZ VOTRE TABLEAU (2 a 7) : "
6020 CURSOR 45,20:INPUT T!
6030 IF T!<2 OR T!>7 THEN 6020
6050 POKE F75,32:CURSOR 23,13:PRINT "UN INSTANT !":RETURN

6997 REM
6998 REM OI presentation IO
6999 REM
7000 MODE 0:PRINT CHR$(12):COLORT 9 9 9 9:POKE F75,32
7010 CURSOR 4,13:PRINT "PACMAN":POKE FBAB3,F75A
7020 CURSOR 40,5:PRINT "(C). CIMBALNIK B.":COLORT 9 15 9 9
7030 WAIT TIME 250:PRINT CHR$(12)
7050 RETURN

8997 REM
8998 REM OI vie = vie-1 IO
8999 REM
9000 COLORG 15 10 13 15:WAIT TIME 5:COLORG 9 10 13 15:WAIT TIME 5
9010 COLORG 15 10 13 15:WAIT TIME 5:COLORG 9 10 13 15:WAIT TIME 5:
COLORG 9 9 13 15
9015 SOUND OFF :S!=200.0
9020 FOR FI!=1.0 TO 100.0:S!=S!-1.0:SOUND 2 1 15 3 FREQ(S!):NEXT
9040 SOUND OFF :CALLM AF!:POKE PY!,Y!:POKE PX!,X!:POKE PL!,1:CALLM
AF!
9043 POKE PX!,X1!:POKE PY!,Y1!:POKE PL!,0:CALLM AF!:CALLM AF!
9045 VI=VI-1:CURSOR 49,2:PRINT VI:IF VI<=0.0 THEN 9500
9050 CURSOR 15,1:PRINT "LE FANTOME VOUS A PRIS UNE VIE !!."
9055 CURSOR 7,2:PRINT " "
9060 COLORG 9 10 13 15:RESTORE
9070 WAIT TIME 150:CURSOR 15,1:PRINT "
"

9100 GOTO 500

9497 REM
9498 REM OI fin des vies IO
9499 REM
9500 SOUND OFF :S!=200.0:K!=0.0
9503 SOUND 2 1 15 3 FREQ(S!):S!=S!-1.0:IF S!<=35.0 THEN 9507
9505 GOTO 9503
9507 SOUND OFF :WAIT TIME 25:MODE 0:CURSOR 23,13:PRINT "FIN DU
JEU"
9510 CURSOR 5,2:PRINT " ESPACE }... pour recommencer"
9515 CURSOR 5,1:PRINT " F }..... pour fin"
9518 Z!=GETC:IF Z!=70.0 THEN PRINT CHR$(12):END
9520 IF Z!=32 THEN SOUND 2 1 15 0 FREQ(1000.0):RESTORE:VI=3:SC=0:
C!=1.0:S2!=0.0:S1!=440.0:GOTO 9530
9525 GOTO 9518
9530 SOUND OFF :GOSUB 6000:GOTO 100
9600 GOTO 9600

9997 REM
9998 REM OI data's dessin labyrinthe IO
10000 DATA 2
10001 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10002 DATA 1,0,0,0,0,1,0,0,0,0,0,0,1,1,1
10004 DATA 1,0,1,1,0,0,0,1,1,0,1,0,0,0,1
10006 DATA 1,0,0,0,0,1,0,0,0,0,0,0,1,0,1
10008 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1

```

10012 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10014 DATA 1,1,0,1,1,0,1,1,0,1,1,0,1,0,1
10016 DATA 1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
10018 DATA 1,0,1,0,1,1,0,1,0,1,1,0,1,0,1
10020 DATA 1,0,1,0,1,1,0,0,0,1,1,0,1,0,1
10022 DATA 1,0,0,0,0,0,0,1,0,0,0,0,0,0,1
10024 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10030 DATA 3
10031 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10032 DATA 1,0,0,0,0,0,0,0,0,0,1,0,0,0,1
10034 DATA 1,0,1,1,0,1,1,1,1,0,1,0,1,0,1
10036 DATA 1,0,0,0,0,0,0,0,0,0,1,0,1,0,1
10038 DATA 1,0,1,1,0,1,0,1,1,0,1,0,1,0,1
10040 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10042 DATA 1,0,1,0,1,0,1,1,0,1,0,1,1,0,1
10044 DATA 1,0,0,0,1,0,0,0,0,1,0,1,1,0,1
10046 DATA 1,0,1,0,0,0,1,1,0,1,0,0,0,0,1
10048 DATA 1,0,1,1,0,1,1,1,0,0,0,1,0,1,1
10050 DATA 1,0,0,0,0,0,0,0,0,1,0,0,0,1,1
10052 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10055 DATA 4
10056 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10058 DATA 1,0,0,0,0,0,1,0,0,0,0,0,0,0,1
10060 DATA 1,0,1,0,1,0,0,0,1,1,0,1,1,0,1
10062 DATA 1,0,0,0,0,0,1,0,0,0,0,0,0,0,1
10064 DATA 1,1,0,1,1,0,1,1,0,1,0,1,1,0,1
10066 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10068 DATA 1,0,1,1,0,1,0,0,1,1,1,1,0,1
10070 DATA 1,0,0,0,0,1,0,0,0,1,1,1,0,0,1
10072 DATA 1,0,1,1,0,1,0,1,0,0,0,0,0,1,1
10074 DATA 1,0,1,1,0,0,0,1,0,1,0,1,0,1,1
10076 DATA 1,0,0,0,0,1,0,0,0,1,0,0,0,1,1
10080 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10085 DATA 5
10086 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10088 DATA 1,0,0,0,1,0,1,0,1,0,1,0,0,0,1
10090 DATA 1,0,1,0,0,0,1,0,1,0,0,0,1,0,1
10092 DATA 1,0,1,0,1,0,0,0,0,0,1,0,1,0,1
10094 DATA 1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
10096 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10098 DATA 1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
10100 DATA 1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
10102 DATA 1,0,1,0,1,0,0,0,0,0,1,0,1,0,1
10104 DATA 1,0,1,0,0,0,1,0,1,0,0,0,1,0,1
10106 DATA 1,0,0,0,1,0,1,0,1,0,1,0,0,0,1
10108 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10110 DATA 6
10112 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10114 DATA 1,0,0,0,0,0,0,0,1,0,0,0,0,0,1
10116 DATA 1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
10118 DATA 1,0,0,0,1,0,0,0,0,0,1,0,0,0,1
10120 DATA 1,0,1,0,1,0,1,1,0,1,1,0,1,0,1
10122 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10124 DATA 1,1,0,1,1,1,0,1,1,1,1,0,1,1,1
10126 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,1
10128 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1
10130 DATA 1,0,0,0,0,0,0,0,0,0,0,1,0,0,1
10132 DATA 1,0,1,0,1,1,1,0,1,0,1,0,0,1,1
10134 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10135 DATA 7



10136 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
10138 DATA 1,0,0,0,0,0,0,0,0,0,1,0,0,0,1
10140 DATA 1,0,1,0,1,1,0,1,1,0,1,0,1,0,1
10142 DATA 1,0,0,0,0,0,0,0,0,0,0,0,1,0,1
10144 DATA 1,0,1,1,1,1,1,1,1,1,1,0,1,0,1
10146 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,0,2
10148 DATA 1,1,0,1,1,0,1,1,1,1,0,1,1,0,1
10150 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,1
10152 DATA 1,0,1,0,1,0,1,1,0,1,1,0,1,0,1
10154 DATA 1,0,1,0,1,0,1,0,0,1,0,0,0,0,1
10156 DATA 1,0,0,0,0,0,0,0,1,0,0,1,1,0,1
10158 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

Après avoir encodé la partie BASIC du programme, il vous reste à introduire les deux parties en LM avant de pouvoir vous détendre.

4000 44 44 EE EE FF FE FF FE FO 1E CF E6 FF FE FF FE
4010 FF FE 73 9C 73 9C 3F F8 3F F8 1F FO OF EO OO OO
4020 00 00 OF EO 1F FO 3F F8 7F FD FF FE 1F FE 01 FE
4030 01 FE 1F FE FF FE 7E FD 3E F8 1F FO OF EO OO OO

5500 F5 C5 D5 E5 21 5A 00 3A 71 55 CD 8F DE 3A 70 55
5510 4F E6 07 5F A9 1F 1F 2F 4F 06 FF 03 09 01 44 66
5520 09 22 6E 55 21 E0 3F 3A 72 55 3C 01 20 00 09 3D
5530 C2 2E 55 16 10 D5 46 23 4E 23 E5 AF 57 B3 CA 4E
5540 55 78 1F 47 79 1F 4F 7A 1F 57 1D C2 41 55 2A 6E
5550 55 78 AE 77 2B 2B 79 AE 77 2B 2B 7A AE 77 01 5E
5560 00 09 22 6E 55 E1 D1 15 C2 35 55 C3 56 CB 64 8C
5570 C8 5D 00



par Raymond V A N L A T H E M (Club CAROLODAI / DAIClic)

Postscriptum No 1

1.- Etat des Indicateurs

Dans l'application No 1 (voir DAICLIC No 1 page 15) nous avons constaté que l'état des indicateurs diffère selon le type de mise à zéro de l'accumulateur A. Il est très souvent utile de représenter la valeur de F (Flag) sous sa forme binaire, comme suit:

Instruction	!	Flag -->	S	Z	O	AC	O	P	1	C(y)
SUB A	!	H 56 -->	0	1	0	1	0	1	1	0
XRA A	!	H 46 -->	0	1	0	0	0	1	1	0
MVI A,:0	!	H 02 -->	0	0	0	0	0	0	1	0

La dernière instruction ne fait que transférer la donnée (0) dans l'accumulateur, sans effectuer d'opérations: les indicateurs ne subissent aucune modification. Par contre avec SUB A nous effectuons une opération arithmétique et avec XRA A une opération de comparaison: d'où l'adaptation des indicateurs au résultat de l'opération effectuée. Voyons cela d'un peu plus près: nous allons supposer que l'accumulateur A contient au départ la valeur HC6. Le raisonnement reste valable pour tout autre valeur contenue en A.

SUB A. Souvenez vous qu'un ordinateur n'effectue pas la
 ===== soustraction de la manière traditionnelle, mais remplace
 cette opération par l'addition du complément à 2 du nombre
 à soustraire.

Nous avons donc $HC6 - HC6 = 0$, ou ce qui revient au même $HC6 + (-HC6) = 0$.

HC6 en binaire	-->	1100 0110	-->	1100 0110
- HC6 complément à 1	-->	0011 1001		
		+		

- HC6 complément à 2	-->	0011 1010	-->	+ 0011 1010

		RESULTAT :	-->	(1) 0000 0000
				Ce qui vaut évidemment zéro (0). = (--)

Examinons maintenant les indicateurs:

- S = 0, car le résultat est positif, le 8ème bit (souligné 2 fois) étant égal à zéro.
- Z = 1, car le résultat est égal à zéro (les 8 bits sont à zéro).
- AC = 1, car lors du calcul, il y a eu report (intermédiaire) du 4ème au 5ème bit (<-->), c'est-à-dire du quartet de droite au quartet de gauche.
Cela peut se produire lors de la partie addition (comme dans l'exemple ci-dessus), ou lors du passage au complément à 2.
- P = 1, car le résultat contient un nombre pair de 1 (ou de 0, ce qui revient au même).
- C(y) = 0, car il n'y a pas de report: le 9ème bit (1) indique un dépassement en soustraction (retenue), qui complémenté donne l'indicateur Cy, soit donc mis à la valeur zéro (0).

XRA A. Il s'agit d'une opération logique du type XOR (OU
 ===== Exclusif), qui compare deux octets, bit à bit. Si les deux
 bits comparés sont identiques, le bit résultant est égal à
 zéro, mais si leur état est différent, au contraire le bit
 résultant sera égal à un.
 Soit donc l'OU Exclusif entre HC6 et HC6.

HC6	--)	1 1 0 0	0 1 1 0
HC6	--)	1 1 0 0	0 1 1 0

RESULTAT :		0 0 0 0	0 0 0 0
		=	

Comme tous les bits sont identiques deux à deux, le résultat
 comprendra nécessairement 8 bits à l'état zéro.

Il en résulte l'état suivant des indicateurs:

S = 0, car le résultat est positif (8ème bit = 0).
 Z = 1, car le résultat est égal à zéro.
 P = 1, car le résultat ne contient aucun 1 (nul est considéré comme
 un nombre pair).
 AC = 0, car il n'y a pas eu de report intermédiaire.
 Cy = 0, car il n'y a eu aucun report.

Remarquons que d'une manière générale, après une instruction de
 comparaison, les indicateurs Cy et AC sont mis à zéro automatiquement.

MVI A, :0. Soulignons que cette opération n'affecte aucun indicateur.
 ===== Dans l'application No 1, en faisant)Z3 nous avons
 initialisé les registres, d'où l'obtention de F = 02. Il
 n'en va pas de même lorsqu'on utilise cette instruction à
 l'intérieur d'un programme.

Faites l'expérience suivante: entrez le programme en Langage
 Machine ci-dessous (à partir de l'adresse mémoire H300), puis après
)Z3 faites)L300 300 305, et vous obtiendrez comme valeurs
 successives de A et de F:

LM	ASSEMBLEUR	I=	A=	F=
3E 00	MVI A, :0	300	00	02
97	SUB A	302	00	56
3E 00	MVI A, :0	303	00	56
C9	RET	305	00	56

Le premier MVI A met le Flag (état des indicateurs) à 02,
 tandis que le second conserve le Flag résultant de l'instruction SUB
 A, soit H56. Refaites l'expérience en remplaçant en 302 SUB A (97) par
 XRA A (AF) et vous obtiendrez les 4 valeurs suivantes de F: 02 - 46 -
 46 - 46. Là aussi le second MVI A conserve le Flag résultant de
 l'instruction qui le précède.

2.- Cycles d'Horloge

Dans la même application No 1 nous vous avons signalé que
 l'instruction MVI A (3E) demandait presque le double de cycles
 d'horloge par rapport aux deux autres solutions. Il est peut-être
 intéressant de s'y attarder un peu.

Vous savez sans doute que le microprocesseur 8080 A est contrôlé par un oscillateur (appelé "horloge") fonctionnant à une fréquence de 2 MHz (mégahertz), c'est-à-dire à 2.000.000 de cycles par seconde. Un seul de ces cycles dure donc 1/2.000.000 ème de seconde, soit 500 ns (nanosecondes) ou 0,5 us (microsecondes). L'instruction la plus rapide (comme SUB A) a besoin pour son exécution de 4 cycles d'horloge, soit $4 * 500 \text{ ns} = 2 \text{ us}$, tandis que la plus lente (XTHL) requiert 18 cycles d'horloge, soit $18 * 500 \text{ ns} = 9 \text{ us}$.

Il est ainsi possible de calculer le temps d'exécution exact d'un programme, pour autant que la durée de chaque cycle soit constante. En effet un écart de 80 KHz (kilohertz) entraîne une différence de 20 ns par période. Il est donc nécessaire que l'oscillateur (multivibrateur) soit piloté par cristal pour assurer une fréquence stable.

Il peut paraître mesquin de s'attarder sur des différences de quelques cycles ou de quelques nanosecondes, mais lorsqu'une exécution de haute précision est souhaitée, il ne peut pas y avoir de différence. D'autre part quelques cycles en moins à l'intérieur d'une boucle: cela compte, surtout lorsque celle-ci est parcourue de nombreuses fois. Dans le cas de l'application No 1 le choix de l'instruction est peu contraignante, car SUB A, XRA A ou MVI A se trouvent en dehors de la boucle.

Signalons que NOP, bien qu'étant inopérante, utilise 4 cycles d'horloge: il ne faut donc pas en abuser!

Après cette assez longue mise au point, passons à la suite des applications!

Application No 4

BUT: Mise à zéro d'un nombre déterminé de cellules-mémoires à partir d'une adresse mémoire de départ.

LISTING DIDACTIQUE

```

300 21 30 04 LXI H,:430 Charge l'adresse mémoire de départ en H+L.
303 06 80 MVI B,128 Charge le nombre de cellules-mémoires à vider
305 36 00 MVI M,0 Charge la valeur 0 en M (dont l'adresse se
trouve en H+L).
307 23 INX H Incrémente le contenu de H+L d'une unité
(donne l'adresse mémoire suivante).
308 05 DCR B Décrémente le contenu de B d'une unité: sert
de compteur.
309 C2 05 03 JNZ Saut conditionnel à l'adresse mémoire H305,
quand l'indicateur Z est égal à zéro.
30C C9 RET

```

COMMENTAIRES

Pour éviter que le DAI ne se plante (voir Application No 1), nous avons fixé une limite, à savoir le nombre de cellules-mémoires à vider. Dans le cas présent nous allons mettre à zéro les cellules-mémoires de H430 à H4AF inclus. Vérifions! En UTILITY, par l'instruction FILL, soit)F400 4FF AC , remplissons la zone de mémoire H400 à H4FF de la valeur hexadécimale AC. Faisons)G300 , puis)D400 4FF : nous obtenons le résultat désiré.

En faisant $\text{)Z3 +)L300 300 30C}$, vos pouvez constater que le registre B décroît de 80 à 0 et que parallèlement le registre L croît de 30 à AF (INX H agit d'abord sur le registre L).

L'instruction JNZ assure le saut conditionnel aussi longtemps que le contenu de B n'a pas atteint la valeur 0. En effet, lorsque B vaudra 0 à l'issue de DCR B, l'indicateur Z basculera et vaudra 1: la condition n'est plus remplie, il n'y aura plus de saut et le programme passera à l'instruction se trouvant en 30C.

Notons également le chargement immédiat en M de la valeur 0 sans passer par une mise à zéro de l'accumulateur comme dans l'application No 1.

EXPERIENCES

(1) Apportez les modifications suivantes au programme:

- ajoutez en 2FF l'instruction SUB A (97);
- remplacez en 305 l'instruction MVI M,0 (36 00) par dans l'ordre NOP (00) et MOV M,A (77);
- modifiez en 309 le code hexadécimal C2 05 03 en C2 06 03 (NOP ne doit pas être comprise dans la boucle).

Le résultat sera identique. Pourquoi alors ajouter une instruction supplémentaire? Parce que la boucle sera parcourue plus rapidement. En effet, MOV M,A ne demande que 7 cycles d'horloge, alors que MVI M,0 requiert 10 cycles d'horloge. Une différence de 3 cycles d'horloge à l'intérieur d'une boucle peut être conséquente. Ainsi pour vider 30.000 cellules-mémoires, vous gagnez 90.000 cycles d'horloge, soit 45 ms (millisecondes).

(2) Modifiez l'adresse de départ et/ou le nombre de cellules-mémoires à vider, notamment de manière à modifier l'octet de poids fort de l'adresse mémoire, par exemple vider 256 cellules-mémoires à partir de H480 (jusqu'à H57F inclus). Par la commande LOOK observez l'incréméntation de H+L (instruction INX H).

Lorsque L aura atteint la valeur de HFF, H augmentera d'une unité (de 4 passera à 5) et L retournera à zéro pour croître ensuite à nouveau.

Application No 5

BUT: Remplir 32 pages de 256 cellules-mémoires, chaque page comprenant 256 nombres hexadécimaux différents (de 00 à FF).

LISTING DIDACTIQUE

300	21 00 50	LXI H, :5000	Charge l'adresse mémoire de départ en H+L.
303	11 00 70	LXI D, :7000	Charge l'adresse mémoire limite supérieure (à ne pas franchir).
306	06 00	MVI B, :0	Initialise le registre B (mis à zéro).
308	70	MOV M,B	Transfère le contenu de B en l'adresse mémoire (contenue en H+L).
309	04	INR B	Incréménte B d'une unité.
30A	23	INX H	Incréménte H+L d'une unité (donne adresse mémoire suivante).
30B	7D	MOV A,L	Transfère le contenu de L dans l'accumulateur.
30C	BB	CMP E	Compare le contenu de E avec celui de l'accumulateur (donc avec celui de L).
30D	C2 08 03	JNZ :308	Saut conditionnel si Z = 0 (résultat non nul).
310	7C	MOV A,H	Transfère le contenu de H dans l'accumulateur.
311	BA	CMP D	Compare le contenu du registre D avec celui de A (donc de H).
312	C2 08 03	JNZ :308	Saut conditionnel si Z = 0 (résultat non nul).
315	C9	RET	

COMMENTAIRES

L'instruction de comparaison CMP soustrait le contenu du résultat con cerné de celui de l'accumulateur, sans modifier ce dernier: le résultat de l'opération n'est donc pas chargée en A, contrairement à l'instruction SUB. Les indicateurs de condition sont affectés comme dans une soustraction. Si le résultat est nul, c'est-à-dire quand le contenu du registre est le même que celui de l'accumulateur, l'indicateur Z bascule à un et le saut conditionnel n'est pas exécuté. Donc quand le contenu de L vaudra H00, le saut en 30D ne s'effectuera pas et lorsque le contenu de H vaudra H70, le programme se termine (pas de saut en 312).

Pour vérifier l'état des cellules-mémoires après exécution du programme, la solution la plus pratique est la visualisation page par page. Après)G300, faites)D5000 50FF jusqu'à)D6F00 6FFF. La visualisation par LOOK est très longue. Vous pouvez modifier les valeurs en 300 et/ou en 303 pour réduire le temps de défilement. Il vous est toujours loisible d'interrompre l'exécution de la commande LOOK au moyen de la touche BREAK. Et il est possible de continuer ensuite l'exécution du programme en tapant simplement)L (suivi de RETURN bien entendu).

EXPERIENCES

(1) Au lieu d'utiliser)G300 ou)L300 300 315 en UTility, vous pouvez exécuter le programme en BASIC par l'instruction CALLM H2FC. Il faut alors ajouter 8 lignes d'instructions-machine:

2FC	F5	PUSH PSW)	Sauve sur la pile l'état des indicateurs
2FD	C5	PUSH B)	le contenu de l'accumulateur
2FE	D5	PUSH D)	et des paires de registre
2FF	E5	PUSH H)	B+C, D+E et H+L.

.....
315 E1 POP H) Restitue ce qui a été sauvé sur la pile
316 D1 POP D) afin de se retrouver dans la situation
317 C1 POP B) précédant l'exécution du programme
318 F1 POP PSW) en langage-machine.

Les lignes 300 à 312 restent inchangées; l'instruction RET (C9) de 315 est placée en 319. Même si tous les PUSH et POP ne sont pas toujours indispensables, il est pratique et prudent de les utiliser automatiquement avec CALLM, pour éviter toute déconvenue.

(2) Vous pouvez remplacer les lignes 315 à 319 par une seule ligne 315:

315 C3 4D C1 JMP :C14D Saut inconditionnel à l'adresse indiquée.

Ici vous faites appel à une sous-routine de la ROM dont l'adresse est HC14D (POP ALL and RETURN). C'est plus court en programmation, cela occupe 2 octets de mémoire en moins, mais la durée d'exécution augmente de 10 cycles d'horloge (60 au lieu de 50, chacune des instructions POP, RET et JMP requérant 10 cycles d'horloge).

(3) En remplaçant en 309 l'instruction INR B (04) par NOP (00), vous maintenez le contenu de B à sa valeur zéro. C'est encore une autre manière de mise à zéro des cellules-mémoires, en donnant ici une adresse limite au lieu d'une longueur.

(4) Il y a intérêt à sortir l'instruction NOP de la boucle afin de réduire le temps d'exécution. Modifions par exemple le programme de la façon suivante:

306	00	NOP
307	00	NOP
308	97	SUB A
309	77	MOV M,A

Et en 30D, ainsi qu'en 312, remplaçons C2 08 03 par C2 09 03 (JNZ :309): ainsi ni les 2 instructions NOP, ni SUB A se trouvent dans la boucle.

"HIGH SPEED LOADER"

Le High Speed Loader (Chargeur haute vitesse) a été développé il y a quelques années par Hein KOP et moi-même, pour rendre possible le chargement très rapide de programmes fréquemment utilisés, au moyen de la lecture d'une banque d'EPROMs connectée au bus DCE. Le système était construit autour d'une carte pour EPROMs disponible chez 'ELEKTUUR'.

Il pouvait contenir quatre EPROMs du type 2716 (4x2K) ou 2732 (4x4K). Ce système était placé grâce à un connecteur à 32 pins sur une carte de base qui contenait le reste de l'électronique.

Le résultat était que nous pouvions lire un programme de 10K en une seconde. Pour les programmes comme SPL, c'était donc une solution idéale! Quelques jeux étaient aussi stockés sur EPROM et les enfants adoraient ça. Une carte EPROM avec le jeu était enfichée, et une commande suffisait pour charger le jeu. Comme ATARI...

Cependant, à cause du prix élevé des EPROMs à l'époque, peu furent faites mais elles fonctionnaient d'une façon très satisfaisante. Un autre désavantage était que la carte d'EPROMs n'était prévue que pour des EPROMs 2716 ou 2732.

Avec le développement du système DAI-DOS 1541, nous avons décidé d'inclure le software pour le HSL dans le software du système 1541, parce que le SPL a été utilisé durant ce projet et était disponible sur une EPROM du HSL.

De temps en temps, des gens nous questionnent à nouveau au sujet de ce projet. Nous avons donc fait une vérification et nous nous sommes débarrassé de quelques problèmes tels que l'initialisation à l'allumage qui n'était plus nécessaire car c'est maintenant sur notre interface DAI-DOS.

Le résultat final est le schéma à la fin de l'article avec cette description. Nous n'avons pas l'intention de développer un circuit imprimé de base à moins que la demande ne soit telle que nous n'ayons pas le choix. Le HSL peut être connecté au bus DCE en même temps que des DCR ou des drives VC1541.

Il possède sa propre adresse de sélection de périphérique et n'interférera pas avec d'autres équipements.

DESCRIPTION TECHNIQUE

La sélection de périphérique est effectuée par PB4 et décodée par un 74LS138 d'une telle façon que seul le HSL soit sélectionné lorsque PB4 est haut.

Une seule sortie du 74LS138 est utilisée, les autres peuvent être utilisées pour adresser d'autres périphériques connectés au bus DCE. Le compteur d'adresses pour les EPROMs consiste en deux 74LS393 synchronisés avec PC0 et un signal reset PC1.

Les deux signaux sont inversés deux fois par un 74LS14. Ceci sert reformer l'horloge et le signal reset qui parcourent le câble plat. Dans certains cas, il est nécessaire de placer une résistance de 390 ohms en série dans les deux lignes.

Les lignes de données sont bufferisées grâce à un 74LS245 qui n'est actif que pendant le moment où le HSL est sélectionné. De cette façon, les signaux indésirables sont évités.

La sélection des EPROMs se fait par une combinaison de PROM 74LS157 et 82S23. Vu cette combinaison, avec PB2 et PB3, nous avons la possibilité de morceler un programme sur plusieurs EPROM et de le relire sans interruption en un bloc.

Exemple :

L'EPROM 1 (2732) contient un programme appelle "Spooler".
Les EPROM 2, 3 et 4 contiennent "SPL".
Avec la commande HSLO, le programme "Spooler" sauve dans la première EPROM est chargé.

Avec la commande HSL1, "SPL" est chargé, ce programme est stocké dans trois EPROMs qui seront sélectionnées l'une apres l'autre.
Lorsque vous donnez la commande HSL2, une erreur "NOT AVAILABLE" apparait sur l'écran.

Au début de chaque nouveau programme sur EPROM ou banque, cinq bytes sont utilisés pour fournir au software les informations pour un démarrage correct.

1er byte : FF Opcode non utilisé.
Indique un NOUVEAU PROGRAMME.
2eme et 3eme bytes : Adresse de depart du programme en mémoire.
4eme et 5eme bytes : Longueur totale du programme.

Le contenu de la PROM 82S23 utilisee est le suivant :

Adresses : 0: 1: 2: 3: 4: 5: 6: 7: 8: 9: A: B: C: D: E: F

Donnees : E: D: B: 7: D: B: 7: F: B: 7: F: F: 7: F: F: F

Le circuit imprimé (numéro EPS 82558-2) était disponible a :

ELEKTUUR PRINTSERVICE
POSTBUS 75
NL-6190 AB BEEK(L)

Le connecteur utilisé est disponible dans presque tous les bons magasins d'électronique.

Il vous est toujours possible de réaliser votre propre arrangement, dans ce cas vous n'aurez pas besoin de cette carte et de ce connecteur.

Des variations, des changements et des améliorations sont possible, nous en réalisons nous-même chaque jour. Si vous avez une bonne idée, faites la nous savoir.

Sur papier, nous avons une version qui peut gérer des EPROMs de type 2716, 2732, 2764, 27128. Cependant, nous n'avons pas de petite carte à EPROMs qui pourrait contenir des soquets a CI de 28 pins avec le même arrangement que celui du connecteur utilise sur la carte d'ELEKTUUR.

Pour utiliser plus de quatre EPROMs sur une carte à EPROMs, une modification du DOS doit être faite car ce n'est pas prévu.

Henk RISON
Luxemburgstraat 17
NL-6164 BS GELEEN

SYSTEME D'EXPLOITATION POUR:

- DISK DRIVE VC1541
- MEMOCOM DCR
- IMPRIMANTE PARALLELE
- HIGH SPEED DATA LOADER

GENERALITES:

Cet article est une introduction au DAI-DOS 1541 déjà décrit dans le test du DAICLIC numéro 2, page 71.

Le système d'exploitation gère jusqu'à 4 lecteurs de disquettes VC1541 avec 4 enregistreurs digitaux MDCR et 2 enregistreurs audio.

Le système d'exploitation se trouve dans une EPROM 2764 utilisant les adresses F000-F7FF et F900-F9FF. Les adresses FA00-FAFF sont utilisées pour une RAM 6116 supplémentaire.

Le software est entièrement compatible avec les versions V1.0 et V1.1 du BASIC. Aucune adresse entre 0000 et BFFF n'est utilisée sauf celles déjà employées par le software du MDCR. Pour l'adressage, le concept DCE-RWC (Real World Cards) est utilisé.

Le software du MDCR est aussi compatible tant qu'il est utilisé à partir du BASIC, ou sous UTILITY via les points d'entrée donnés dans le manuel du MDCR.

Sont aussi implémentés, le software pour la gestion d'une imprimante parallèle et d'un chargeur rapide de données (banques d'EPROM) HSL. Ainsi que quelques commandes additionnelles.

ORGANISATION DE LA MEMOIRE:

La zone mémoire F000-F7FF est utilisée par quelques banques ROM switchées:

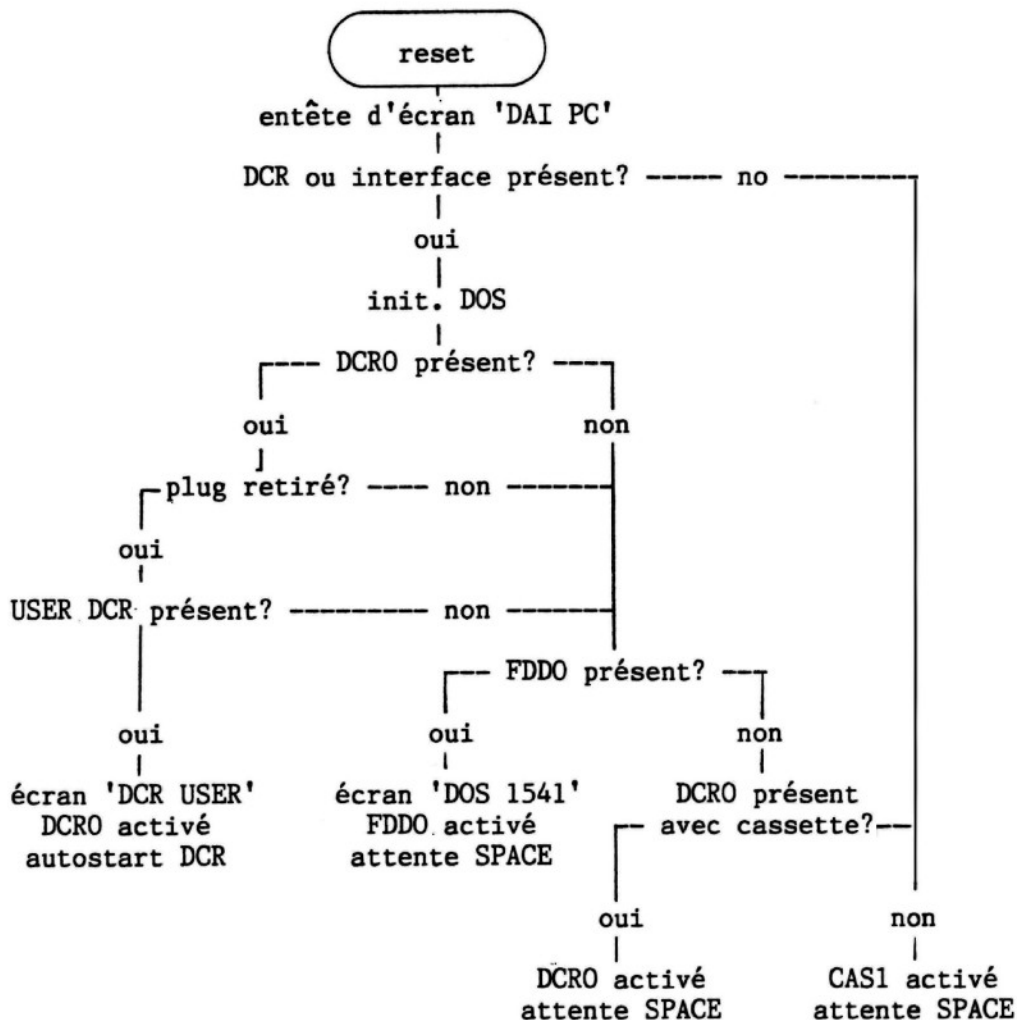
- Banque 0: Routines générales pour le lecteur de disquettes (FDD), l'enregistreur de cassettes digitales (DCR), l'imprimante parallèle et les commandes supplémentaires.
- Banque 1: Le système d'exploitation des disquettes.
- Banque 2: Les systèmes d'exploitation pour l'enregistreur digital et le chargeur rapide de données.
- Banque 3: Libre pour utilisation personnelle, sauf F100-F1FF.

Les adresses F900-F9FF sont utilisées pour quelques routines d'intérêt général. Cette zone se trouve dans la banque 3 de l'EPROM.

Une imprimante parallèle peut être connectée au connecteur 26 broches de l'interface. Le chargeur rapide de données doit être connecté directement à un connecteur sur le cable plat.

ALLUMAGE:

Lorsque le DAI est allumé, ou quand le bouton RESET est pressé, voici ce qu'il se passe:



COMMANDES:

Toutes les commandes du DAI pour écrire et lire comme LOAD, SAVE, LOADA, SAVEA et les UTILITY W(rite) et R(ead) sont aussi valable pour le DCR et le FDD. Seule la commande CHECK ne marche pas avec le FDD. Tous les fichiers doivent être sauves sur disque avec un titre. 16 caractères maximum sont autorisé dans les titres de fichier (FDD). Les fichiers existants peuvent être réécrits par "%:titrefichier".

Les commandes DCR sont équivalentes aux commandes décrites dans le manuel du MDCR, excepté que seules les commandes abrégées sont admises. Si un DCR est sélectionné, toutes les commandes FDD sont désactivées (erreur de DEVICE MISMATCH).

Pour le FDD, les commandes suivantes sont disponibles:

- FDD X : Sélectionne un drive 0-3. Les commandes DCR sont désactivées et provoqueront une erreur 'DEVICE MISMATCH'.
- FORMAT : FORMAT "NOMDISQUETTE,ID".
Formate une disquette. Chaque disquette doit avoir un nom de maximum 16 caractères et un code d'identité unique de 2 caractères.
- SCRATCH : SCRATCH "NOM1,NOM2,...".
Efface les fichiers NOM1, NOM2,...
- RENAME : RENAME "ANCIEN=NOUVEAU".
Change le nom d'un fichier.
- COPY : COPY "NOUVEAU=ANCIEN".
Copie un fichier.
- INIT : INITialise le drive à l'état d'allumage.
- VALID : Réorganise la disquette pour avoir le maximum d'espace disponible.
- DIR : Affiche la directory de la disquette. Au moyen des flèches, un programme BASIC ou MACHINE peut être chargé directement depuis la directory. Les programmes BASIC démarreront automatiquement après chargement. Les programmes machines vous positionnent en UTILITY après chargement.
- FVER : FVER "NOM".
Vérifie l'enregistrement du fichier indiqué au moyen d'un test de checksum.

Du fait que tout les fichiers sont ouverts et fermés automatiquement, il n'y a pas de commandes OPEN et CLOSE.

Quelques commandes supplémentaires sont disponibles:

- BOOT : BOOT "OBJECT,BASIC".
Charge un programme machine, suivi par un programme BASIC .
Cette commande remplace le bootstrap loader DBL.
- UBL : UBL "NOM".
Comme BOOT mais pour des fichiers constitués d'une partie machine et d'une partie BASIC, sauvés avec le même nom de fichier + une extension :
 - NOM/O partie machine
 - NOM/B partie BASIC
- CAS X : Sélectionne un enregistreur audio. Si aucun numéro n'est donné, l'enregistreur 1 est sélectionné.
Les commandes DCR et FDD provoqueront une erreur DEVICE MISMATCH.
- HSL X : Lit les données de l'EPROM 0-3 dans le HSL.

- USR : Appelle une routine machine déclarée USER via les adresses de la table E/S en F02DA.
- LNON : Active la fonction d'auto-numérotation des lignes. En mode direct, si la première entrée est un espace, un numéro de ligne est automatiquement généré.
- LNOFF : Désactive la fonction ci-dessus.
- <TAB> : Efface l'écran.
- /C : Le curseur d'origine (souligné clignotant) est rappelé.
- /D : Sortie sur écran seulement.
- /E : Entrée depuis le buffer d'édition.
- /F : IMP FPT.
- /H : 'HELP'. Affiche les directions d'E/S importantes.
- /I : IMP INT.
- /M : MODE 0.
- /P : Sortie sur imprimante parallèle.
- /S : Sortie sur écran et RS232.
- /T : Rappelle les couleurs texte d'origine (8 0 8 0).
- /O : Valeurs diverses d'origine: mode 0, efface écran, couleurs d'origine, curseur d'origine et IMP INT.
- : READ LINENUMBER: Il est possible de lire les DATA d'une ligne de programme donnée via:
 - 10 A%={numéroligne}: CALLM F033A,A%
 - 20 READ ...

MESSAGES D'ERREUR:

Pour le DCR et le FDD, les messages d'erreur de chargement standards sont utilisés. Le FDD renvoie lui-même quelques messages d'erreur. Ils peuvent être trouvés dans le manuel du VC1541. En plus, quelques messages d'erreur du DOS seront affichés.

UTILISATION DES COMMANDES DOS EN BASIC:

Toutes les commandes DOS ne sont valables qu'en mode direct. Pour permettre l'utilisation de ces commandes depuis le BASIC, elles doivent être placées dans un REM suivant un CALLM F0FOO:

```
10 CALLM F0FOO:REM FDDO:DIR
20 A$="NOM":CALLM F0FOO:REM SCRATCH A$
```

QUEL EST LE PRIX:

On peut acheter le lecteur de disquettes dans tout bon magasin d'ordinateurs. Faites attention aux prix affichés, ils peuvent varier fortement d'un endroit à l'autre (entre 12000 FB et 15000 FB!).

La carte d'interface et la carte EPROM coûtent 325 Florins. Le câble pour l'interface (pas nécessaire si vous avez déjà un câble pour le DCR) coûte 50 Florins en plus, mais peut aussi facilement être fait par vous-même.

COMMENT L'OBTENIR:

Envoyez 2 EUROchèques dûment complétés (maximum 300 Florins par chèque), ou un mandat postal international pour le montant de:

325 F1 (interface + carte EPROM)
375 F1 (idem + cable plat)

à:

Peter WIEGERS
Corneliusstraat 5
6432 CZ HOENSBROEK (PAYS-BAS)

Délais: entre 6 et 8 semaines.

Pour plus d'information:

Jan BOERRIGTER
Fabritiusstraat 15
6174 RG SWEIKHUIZEN (PAYS-BAS)
Tél. 04493-2093 (19-21 H)

Vous pouvez également bénéficier d'une réduction si vous participez à une commande groupée (10 pièces) par l'intermédiaire d'IDC. L'interface et la carte EPROM coute environ 6200 FB (selon le cours du Florin) ou 315 F1. Si vous souhaitez participer a une commande groupée, contactez C.Poels ou F.Duluins (à l'heure ou nous écrivons ces lignes, 6 personnes ont pris une option d'achat, ne tardez pas à vous inscrire!)

Le 'DAI DOS 1541' est une co-production de: Jan BOERRIGTER, Hein KOP, Henk RISON et Peter WIEGERS.

Jan BOERRIGTER .

COMPARAISON DAI DOS 1541 - VC1541 INTERFACE

Vous ne le savez peut etre pas encore, mais il existe deux sortes d'interfaces permettant d'utiliser le drive Commodore 1541: le système Hollandais et le système Allemand. Pour vous aider à y voir plus clair, voici quelques informations de M.Jan BOERRIGTER qui est l'un des concepteurs du système Hollandais (celui qui nous avons testé dans le numéro 2). L'auteur nous prévient lui-même de l'aspect non-objectif de cet article puisqu'il connaît mieux son système que le système Allemand. Comme il faut rester juste, vous trouverez aussi dans cette revue l'avis de M.Heinrich TEGETHOFF (un des concepteurs Allemands) qui donne évidemment une version tout a fait différente sur la question. A vous maintenant de vous faire votre propre opinion.

DAI DOS 1541 ou VC1541 là est la question ?

"Comparé avec l'interface VC1541 'Allemand', les remarques suivantes peuvent être faites:

1. Le DAI DOS 1541 est 100% compatible DCE-bus, l'interface Allemand l'est à 0%. Ceci veut dire que notre interface peut être utilisé avec tous les autres équipements DCE-bus sans aucun problème.

2. L'interface Allemand ne peut gérer de DCR ni d'imprimante parallèle. Le DAI DOS supporte jusqu'à 4 drives VC1541, 4 DCR, une imprimante parallèle et un 'High Speed data Loader' ou HSL (pour lecture rapide d'EPRoMs).

3. Le software Allemand est basé sur le protocole du Commodore: les fichiers doivent être ouverts et fermés par l'utilisateur. Notre DOS ouvre et ferme tous les fichiers automatiquement. Par conséquent, le chargement ou la sauvegarde de programme correspondent tout à fait au protocole du DAI. La sélection de la mémoire de masse se fait par les commandes FDD 0-3, DCR 0-3 ou CAS 1-2, et le reste est en accord avec le manuel du DAI.

4. De nombreuses commandes supplémentaires sont implémentées dans notre DOS (V. article dans ce numéro), comme les commandes abrégées pour certaines fonctions du DAI, un bootstrap loader pour des programmes LM et BASIC, une option autonumérotation des lignes, etc....x

5. Le DAI-DOS 1541 lit automatiquement les erreurs du VC1541 à la fin de chaque opération FDD. L'information de statut du VC1541 est aussi utilisée pour un traitement d'erreur automatique.

6. La directory n'utilise pas d'espace mémoire utilisateur. Les programmes BASIC et LM peuvent être chargés depuis la directory par manipulation des touches curseur. Les programmes BASIC démarrent automatiquement.

7. Le DAI-DOS 1541 ne peut gérer qu'un fichier à la fois. Mais comme de toutes façons, un seul fichier à la fois ne peut être lu, et que aucune commande OPEN et CLOSE n'est nécessaire, ce n'est pas un désavantage pour l'utilisateur.

8. Le DAI-DOS 1541 ne peut utiliser les options du VC1541 pour les fichiers relatifs et les commandes d'accès direct. Mais une disquette d'utilitaires peut être commandée à 'Micro Serive' (l'éditeur du DAI Firmware Manual). Elle contient un moniteur spécial et des démos, permettant aussi l'utilisation de ces options.

9. Toutes les commandes pour DCR, FDD, etc, peuvent aussi être utilisées depuis le BASIC, au moyen de la même construction que celle utilisée pour le MEMOCOM DCR.

10. Un manuel de 15 pages est livré avec le DAI-DOS 1541, décrivant en détail toutes les possibilités."

Jan BOERRIGTER .

RENSEIGNEMENTS SUR LE SYSTEME DE DRIVES VC1541
ET SUR L'INTERFACE ALLEMANDE.

par les auteurs de celui-ci, en réponse à l'article de M. Boerrigter

Etant l'un des coresponsables du système allemand de brancher les lecteurs de disquettes Commodore sur le DAI, je veux corriger quelques impressions que donne l'article de M. Boerrigter dans ce numéro, et en outre donner des renseignements sur ces lecteurs eux-mêmes pour les intéressés d'un système VC-1541:

- les points 3 et 7 de l'article sont tout simplement faux. Les différentes sortes de fichiers, programmes BASIC et LM, les rangées (arrays), les fichiers SPL, et tous les autres genres de fichiers se laissent traiter tout à fait automatiquement (selon le protocole conventionnel du DAI), tout comme si on avait branché un lecteur de cassettes digitales ou audio. Les erreurs qui peuvent survenir sont traités tout aussi automatiquement. C'est donc ridicule de parler d'un "protocole Commodore" ennuyant. Celui-ci est offert en sus, car: Le DAI DOS 1541 regarde le floppy comme lecteur non intelligent, comme les lecteurs de cassettes digitales ou audio, mais l'interface VC-1541 les regarde comme moyen de sauvetage intelligent, ce qu'elle est en effet, ayant une propre unité centrale (6502) et 16 kilooctets de mémoire morte, 2 KO de mémoire vive. Par les commandes supplémentaires, l'interface allemande fait usage de 100% des possibilités de ce lecteur, le système néerlandais de beaucoup moins. De cela résultent beaucoup des points faibles du DAI DOS 1541.

- A part les fichiers de programmes usuels (rangées de données (arrays) y compris, etc.), le lecteur connaît les fichiers séquentielles, pour traiter des ensembles de données plus grands que la mémoire du DAI (plus de 160 000 octets). Il connaît les "fichiers relatifs" qui peuvent servir à une gestion de banques de données sur disquette. Il peut exécuter des programmes dans sa propre mémoire vive indépendamment du DAI (Fichiers USER etc.) et connaît l'accès direct à secteurs et octets individuels. Sans des commandes qui effectuent ces possibilités, on ne sait en plus: - obtenir des renseignements sur le status de la disquette, par exemple, sur sa capacité libre et sur beaucoup d'autres choses qui sont décrites dans d'épaisses livres sur cette station. - faire exécuter de propres programmes sur la station. En particulier, notre programme de chargement rapide (chargement de programmes à 15000 baud, par comparaison: K7 600 baud, K7 digitales 6000, DAI DOS 3400 et interface VC-1541 3700 baud) ne fonctionne pas sans de telles possibilités. Ce programme, avec 8 autres programmes intéressants, est donné gratuitement aux acheteurs du VC-1541. - toucher directement aux secteurs individuels de la disquette, possibilité qu'admet toute station de disquettes, y compris la VC-1541. Les commandes OPEN et CLOSE, dites "superflues", ouvrent par exemple la possibilité d'un tel accès direct, qui est indispensable pour implémenter la langue FORTH et pour divers programmes utilitaires. En outre ils permettent de gérer sur disquette plusieurs grands fichiers à la fois, par exemple pour un traitement de textes qui veut mélanger plusieurs textes préalablement préparés.

DAI DOS 1541 ne sait offrir aucun de ces possibilités, car il ne connaît que la méthode traditionnelle mais vieillie et très ennuyant d'incorporer ses commandes dans des programmes, de la manière

1000 CALLM #F000:REM SCRATCH "..fichiers.."

Le système allemand permet, sans CALLM ou trucs pareils, d'employer de véritables nouvelles commandes BASIC, par exemple
1000 COM "S:..fichiers.."

Les deux lignes font la même chose. La commande COM est en outre la clé de toutes les actions que DAI DOS 1541 nomme individuellement, comme formater, effacer, renommer, copier, initialiser. Mais par l'intermédiaire de cette commande le DOS propre de la station VC-1541 peut accepter plus de 20 commandes individuelles comme par exemple l'exécution d'un programme par l'unité centrale de la station elle-même. Pour donner un exemple de l'emploi des autres 17 nouvelles commandes BASIC:

```
100 OPEN 5 B 15 "I":PUT# 5;"M-R",#FA,2,3:GET# 5;L,H,H:
FREEBYTES=(H*256+L)*254:CLOSE 5
```

demande la capacité libre de la disquette en octets.

Pour le chargement de programmes LM à partir de BASIC il existe la commande RMP, par exemple

```
2400 INPUT "Nom de fichier ";FN$:PRINT :RMP FN$:RETURN
```

- Avec la version simple de l'interface VC-1541, que n'importe qui saurait bricoler soi-même en suivant le schéma (qui sera publié, je crois), et le software pour le prix de la K7 et les frais d'envoi, il n'est pas possible de brancher une imprimante parallèle ou un MDCR en plus. C'est surtout gênant pour l'imprimante, pas autant pour le lecteur Memocom, car on ne s'en servira guère plus après 2 ou 3 semaines. Mais nous faisons mention ici de la deuxième génération d'interfaces allemandes, qui ne sont plus connectés au DCE-bus et qui sont par conséquent compatibles à 100%. En outre, avec ses 26 KO de programmes en LM et ses 121 nouvelles commandes BASIC à un prix de 250 DM environ (environ 5000 FB), il livre beaucoup dans le domaine du software. Le chargement rapide Quickload est employé automatiquement.

Un commentaire sur quelques points de la "comparaison" de M. Boerrigter:

2) et 3) La technique propre des stations Commodore permet toujours de brancher 4-5 lecteurs à la fois. Le système allemand admet cette possibilité aussi, et permet par le moyen du software d'associer des adresses logiques aux lecteurs et de les sélectionner.

6) Le DIRectory allemand ne requiert également pas de place dans la mémoire du DAI. Les noms des fichiers peuvent être abrégés, ce qui donne un confort comparable au chargement au départ du directory.

7) Ne gérer qu'un fichier à la fois? La VC-1541 en sait gérer trois à la fois, ou on peut par exemple rediriger la sortie de données vers un fichier au lieu de les envoyer à l'écran/imprimante, possibilités qui n'existe pas sans OPEN et CLOSE.

8) Il n'est pas nécessaire d'acheter un Toolkit d'utilitaires, avec les commandes BASIC supplémentaires de l'interface VC-1541 on les programme facilement soi-même. Les plus importantes (Copier, analyser, éditer des disques, Quickload) sont gratuites avec le système VC-1541.

Le système allemand a été testé depuis un an, ce qui veut dire que le système marche bien et des erreurs ne se montrent plus.

Heinrich Tegethoff

Des renseignements pour obtenir l'interface VC-1541:

- a) L'interface simple:
- Software et schémas pour la construction soi-même:
M. Poels les a reçus.
 - L'Eprom + Software sur disquette (sans interface ni carte X-bus): 25 DM / 550 FB chez
Heinrich Tegethoff
In der Provitz 87
D-4630 Bochum
Paiement par eurocheque ou mandat postal international
 - Interface, eprom, software: 95 DM / 1950 FB
Comme carte X-bus on peut se servir d'une carte MDCR, puisque le ComDOS V2.3 pour l'interface VC-1541 emploie un eprom de 2 KO tout comme le TOS MDCR.
 - Carte X-bus: 30 DM / 650 FB
 - L'ensemble: 125 DM / 2600 FB
- b) Interface sur carte d'expansion pour l'X-bus
- Carte X-bus allemande pour jusque 128KO d'eprom/RAM, 2 KO de RAM auxiliaire, avec interface Commodore.
Important: il ne s'agit pas de n'importe quelle carte X-bus pour Eproms supplémentaires, mais d'un système avec le software nécessaire pour gérer le tout. Les programmes sont exécutés avec la double vitesse que dans le RAM et n'y occupent pas de place.

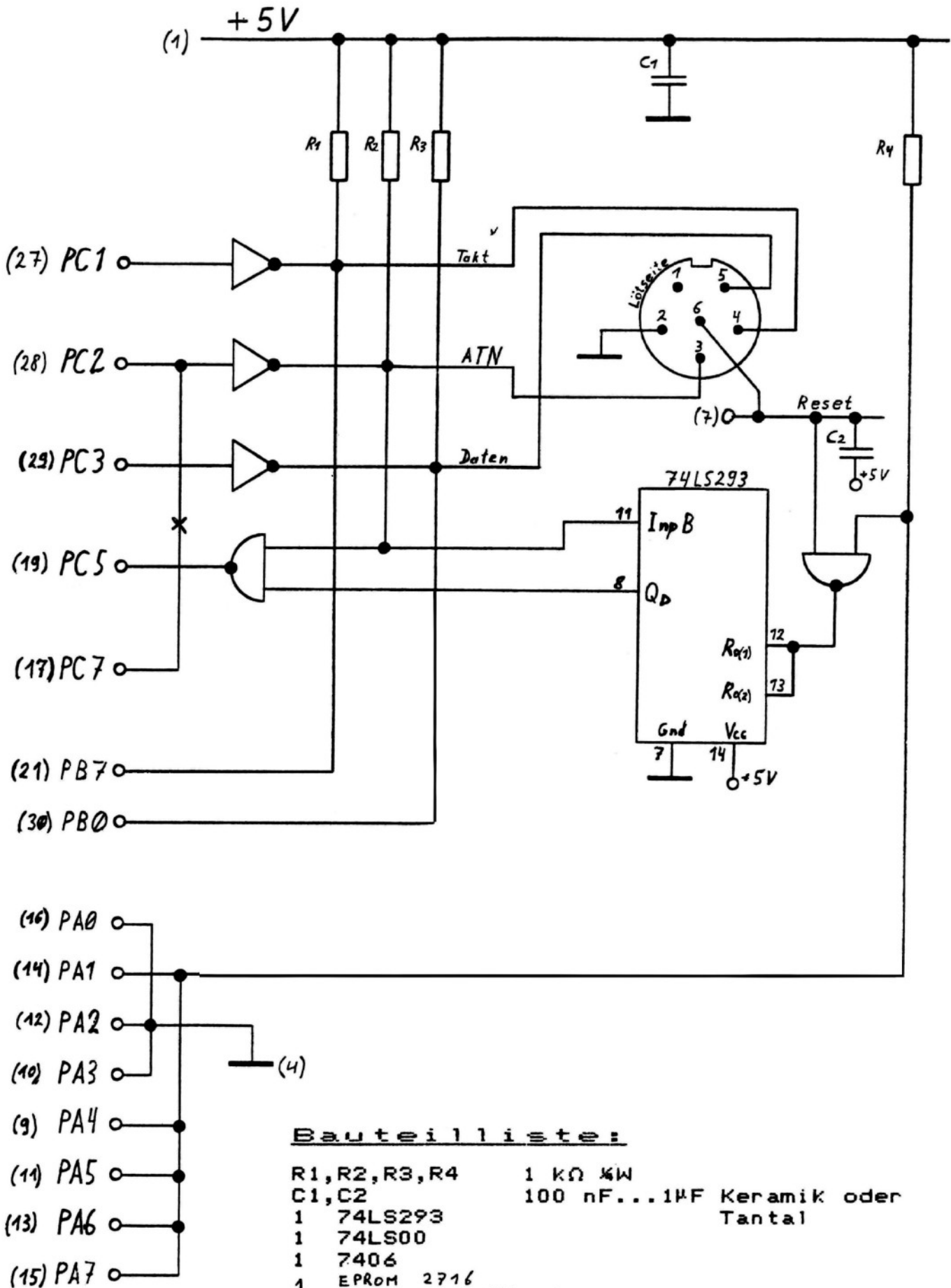
Versions:

- a) Interface, 2 KO de RAM, 16 KO de ROM, système de gestion, 65 nouvelles commandes BASIC, par exemple Labels pour les lignes, IF-THEN-ELSE, REPEAT-UNTIL, horloge par microsecondes, CIRCLE, ELLIPSE, PAINT (très rapides), DOS avec Quickload, commandes abrégées, programmes utilitaires etc.
Prix: 175 DM / 3600 FB
- b) Avec SGT, un programme très rapide pour textes en graphiques. 27 commandes, cartes de jeux, 13 styles de caractères, utilitaires, 24 pages de documentation.
Vitesse: 650 caractères par seconde (FGT: 36)
Prix supplémentaire: 70 DM / 1500 FB
Version démo peut être demandée.
- c) D'autres programmes, comme copie d'écran, impression formatée, aides de conversion, utilitaires, existent sur eproms. Le prix n'est pas encore fixé (6 juillet).
- d) BASIC V1.2 avec la carte:
- BASIC V1.2 seul : 114 DM / 2300 FB
 - avec carte 128 KO : 275 DM / 5600 FB
 - avec SGT : 340 DM / 6900 FB

SGT inclus, mon système contient 121 commandes BASIC supplémentaires, c'est-à-dire 183 en tout! En sus, il y a 19 abréviations de commandes importantes, dont 6 en BASIC V1.2.

Heinrich Tegethoff

Interface DAI - VC 1541



Bauteilliste:

R1, R2, R3, R4	1 kΩ ¼W
C1, C2	100 nF...1µF Keramik oder Tantal
1	74LS293
1	74LS00
1	7406
1	EPROM 2716
1	DCE-BUS-CONNECTOR

La petite routine ci-dessous (de Pascal JANIN), permet de tracer des cercles d'une façon ULTRA-RAPIDE! De plus, il n'y a aucune approximation car le tracé se fait POINT par POINT dans n'importe quel mode graphique. Si le cercle "sort" de l'écran, seules les parties visibles seront affichées! (ex: rayon de 350 en mode 5 ou 6).

Pour l'utiliser dans vos programmes:

Fr228/Fr229: abscisse centre
 Fr230/Fr231: ordonnée centre
 Fr232 : couleur
 Fr224/Fr225: rayon cercle (entre 1 et 65535 inclus!)
 Fr226/Fr227: (rayon cercle)/SQR(2)
 CALLM Fr3AEO

Vous pouvez aussi l'utiliser avec la démo en BASIC.

```

3AEO F5 C5 D5 E5 21 00 00 22 D2 3A 2A 24 02 22 D0 3A
3AFO 7C 65 6F 22 D6 3A 21 00 00 22 D4 3A 21 D4 3A E7
3B00 0C E7 54 E7 0F 21 20 02 E7 0F 2A 22 02 7C 65 6F
3B10 22 22 02 2A 26 02 2B 22 26 02 C3 10 3C 00 00 CD
3B20 5E 3B 2A D2 3A 23 22 D2 3A C3 10 3C 2A D0 3A 2B
3B30 22 D0 3A EB 2A 26 02 CD 14 DE C2 F8 3B E1 D1 C1
3B40 F1 C9 00 00 00 00 00 00 00 00 00 00 00 00 00
3B50 00 00 00 00 00 00 00 00 00 00 00 00 00 2A D0
3B60 3A CD E2 3B 22 DC 3A 2A D2 3A CD E2 3B 22 DE 3A
3B70 2A D0 3A EB 2A D2 3A CD C1 3B 2A D0 3A EB 2A DE
3B80 3A CD C1 3B 2A D2 3A EB 2A DC 3A CD C1 3B 2A D2
3B90 3A EB 2A D0 3A CD C1 3B 2A DC 3A EB 2A D2 3A CD
3BA0 C1 3B 2A DC 3A EB 2A DE 3A CD C1 3B 2A DE 3A EB
3BB0 2A D0 3A CD C1 3B 2A DE 3A EB 2A DC 3A CD C1 3B
3BC0 C9 D5 EB 2A 30 02 19 44 4D D1 2A 28 02 19 EB 2A
3BD0 94 00 CD 14 DE D8 EB 78 B7 C0 3A 32 02 EF 1E C9
3BE0 00 00 7D 2F 6F 7C 2F 67 23 C9 D3 D7 8D 8A A0 D0
3BF0 E1 D1 C1 F1 FB C9 00 00 2A D0 3A 7C 65 6F 22 D6
3C00 3A 21 00 00 22 D4 3A 21 D4 3A E7 0C E7 54 E7 0F
3C10 2A D2 3A 7C 65 6F 22 DA 3A 21 00 00 22 D8 3A 21
3C20 D8 3A E7 0C E7 54 21 D4 3A E7 4E E7 15 5A 51 3A
3C30 21 02 B8 DA 2C 3B C2 1F 3B 2A 22 02 CD 14 DE DA
3C40 2C 3B C3 1F 3B 00 00 00 00 00 00 00 00 00 00
  
```

```

10 REM DEMO CIRCLE
20 CLEAR 300:DIM XC(5),YC(5):FOR I=1 TO 5:READ XC(I),YC(I):NEXT
30 MODE 6:COLORG 0 1 2 14
40 FOR J=1 TO 5:C=21+(J MOD 3)
41 FOR K=40 TO 70 STEP 2
42 POKE Fr228,XC(J) IAND FrFF:POKE Fr229,XC(J) SHR 8
43 POKE Fr230,YC(J) IAND FrFF:POKE Fr231,YC(J) SHR 8
44 POKE Fr232,C:A=K/SQR(2):A=A
45 POKE Fr224,K IAND FrFF:POKE Fr225,K SHR 8
46 POKE Fr226,A IAND FrFF:POKE Fr227,A SHR 8
47 CALLM Fr3AEO:NEXT:NEXT
49 GOTO 49
50 DATA 55,170,110,85,165,170,220,85,275,170
  
```

DAI-DOS 1541: FICHIERS RELATIFS

Cet article décrit le principe des fichiers relatifs et la façon dont ils sont gérés par le DAI-DOS 1541 avec le drive Commodore VC1541.

PRINCIPES/AVANTAGES DES FICHIERS RELATIFS:

Accès rapide à chaque enregistrement - De grands fichiers peuvent être gérés.

Les fichiers relatifs sont souvent appelés fichiers 'random'. Ils sont orientés disque. Ceci veut dire que le contenu du fichier n'est pas chargé dans l'espace mémoire de l'ordinateur, mais reste sur la disquette. Seul l'enregistrement recherché est lu depuis la disquette vers la mémoire de l'ordinateur.

L'avantage de ce système de manipulation de fichiers est que de très grands fichiers peuvent être gérés. Les fichiers séquentiels - pour le DAI, cela veut dire: les tableaux - peuvent atteindre au maximum 32K, à cause de la taille de la HEAP dans le DAI. Les tableaux de chaînes ne peuvent excéder 20K du fait de l'espace RAM double requis pour la lecture et l'écriture. Les fichiers relatifs sur la disquette peuvent atteindre 164K, limite de la disquette elle-même.

A fichier relatif consiste en un certain nombre d'enregistrements. Un enregistrement est un bloc d'informations. Prenons l'exemple d'un fichier d'adresses. Pour chaque personne dans ce fichier, nous avons besoin de stocker le nom, l'adresse et la ville. Ces trois points ('CHAMPS') réunis forment un 'ENREGISTREMENT'. L'ensemble des enregistrements de toutes les personnes de ce fichier d'adresses est appelé 'FICHER'.

Un autre avantage est la façon dont les mises à jour sont faites. Un fichier relatif à un accès aléatoire, ceci veut dire que chaque enregistrement du fichier peut être adressé sans devoir charger tout le fichier dans la mémoire de l'ordinateur. Pour mettre à jour un fichier séquentiel (tableau), le fichier entier doit être lu, les mises à jour faites, et le fichier doit de nouveau être sauvé. Pour les fichiers relatifs, seul l'enregistrement à mettre à jour doit être chargé dans l'ordinateur et peut après être sauvé. Ceci économise beaucoup de temps.

Pour gérer des fichiers relatifs sur le DAI-DOS 1541, on a besoin du programme moniteur 'FDDMON/0' (voir plus bas). Ce petit programme machine autorise l'utilisation des options supplémentaires du drive VC1541.

STRUCTURE DU FICHER ET DES ENREGISTREMENTS:

Les fichiers relatifs doivent avoir des enregistrements de longueur fixe. Ceci est dicté par la façon dont ils sont gérés par le système d'exploitation du VC1541. En plus, seules les chaînes de caractères sont autorisées comme enregistrements. La longueur maximale d'un enregistrement est de 254 caractères.

Le nombre maximal d'enregistrements est de 65535 (FFFFF), ce qui veut dire en pratique que le nombre d'enregistrements utilisables est limité par la capacité de la disquette.

Dans notre exemple de fichier d'adresses, prenons:

nom	20 caractères
adresse	20 caractères
ville	15 caractères

La longueur totale d'un enregistrement est maintenant $20+20+15=55$ caractères.

La longueur d'un enregistrement doit être déterminée à l'avance. Une fois fixée, elle ne peut plus du tout être changée! Elle est utilisée par le drive pour l'organisation du fichier sur la disquette.

Pour permettre la manipulation de chaque simple champs, nous devons prendre garde au fait que chaque champs a exactement ces nombres de caractères. Si par ex., un nom n'a que 14 caractères, 6 espaces supplémentaires doivent être ajoutés au nom. Plus tard, nous verons comment nous réunissons les champs pour former un enregistrement, et comment un simple champs peut être isolé d'un enregistrement.

De plus, le fichier doit avoir un nom de maximum 16 caractères.

COMMUNICATION DAI - VC1541:

Les fichiers relatifs ne peuvent être gérés qu'en combinaison avec le programme moniteur FDDMON/O. Ce programme doit être chargé en même temps que votre programme BASIC. Si le nom du programme BASIC est 'MAILING LIST/B', alors chargez les deux morceaux avec:

BOOT "FDDMON/O,MAILING LIST/B"

Au moyen des routines machines du programme FDDMON/O, l'ensemble de la communication est gérée. Le plus simple est de déclarer (INT) des variables dans votre programme BASIC avec les adresses des différentes routines, par ex.:

OPEN =H400	PRNT3=H406	ERROR=HF2E8
CLOSE=H421	INP3 =H418	etc...

Voir le manuel du 'FDD TOOLKIT-1' (voir ci-dessous) pour toutes les adresses.

Contrairement aux procédures normales de 'save' et 'load' sur le DAI-DOS 1541, pour les fichiers relatifs, l'ouverture et la fermeture du fichier doit être effectuée par l'utilisateur. Pour toute communication avec les fichiers relatifs, le canal 3 du VC1541 est utilisé, en combinaison avec le canal 15 (canal de commande/erreur) pour les instructions du système. Le canal d'erreur n'est pas lu automatiquement.

Seul un fichier relatif peut être utilisé à la fois. Si vous voulez gérer des données de deux différents fichiers relatifs, fermez toujours le premier avant d'ouvrir le deuxième, autrement un message d'erreur 'NO CHANNEL' surviendra.

OUVERTURE/CREATION/FERMETURE DE FICHER RELATIF

Le format pour ouvrir un fichier relatif est le suivant:

```
10 NOMFICHER$="MAILING LIST"
20 LONGENREG=56      (1 de plus que la longueur réelle car le
                    RC est envoyé après la chaîne)
30 OPEN$=NOMFICHER$+",L,"+CHR$(LONGENREG)
40 CALLM OPEN3,OPEN$
```

Lorsque le fichier relatif est ouvert pour la première fois, il est conseillé de réserver directement un certain nombre d'enregistrements. Ce n'est pas nécessaire, mais cela augmente la vitesse de gestion du fichier. Réserver un certain nombre d'enregistrement veut dire que, sur la disquette, des enregistrements 'vides' sont écrits. Si par ex., vous pensez que vous aurez besoin de 200 enregistrements, il suffit d'écrire sur le 200ème enregistrement. Tous les enregistrement 1-199 seront alors aussi initialisés. Plus tard, l'écriture des enregistrement 1-199 sera plus rapide. Ce principe peut toujours être répété. Si par ex., les 200 enregistrement sont utilisés, vous pouvez écrire dans le 400ème enregistrement, et alors les enregistrement 201-399 sont aussi initialisés. Pour libérer ce dernier enregistrement, il suffit d'y écrire HFF - CHR\$(255).

D'abord, le VC1541 doit être positionné sur cet enregistrement. Cela se fait comme suit:

```
50 NUMENREG=200      (par exemple)
60 HB=NUMENREG SHR 8
70 LB=NUMENREG IAND 255
80 POS$="P"+CHR$(3)+CHR$(LB)+CHR$(HB)+CHR$(0)
90 CALLM PRNT15,POS$
```

La chaîne POS\$ dit au VC1541 qu'il y a une commande 'P'osition sur le canal 3, donne le numéro d'enregistrement, et dit qu'il doit pointer sur le premier byte (byte 0) de cet enregistrement. Ceci est fait par le canal de commande 15.

Pour écrire HFF dans cet enregistrement, les instructions suivantes suffisent:

```
100 OP$=CHR$(255)
110 CALLM PRNT3,OP$
```

L'instruction print à la ligne 110 provoquera un message d'erreur 'RECORD NOT PRESENT', parce que l'enregistrement 200 n'existe pas encore au moment où HFF y est écrit. Le message d'erreur peut être ignoré.

Si un grand nombre d'enregistrements est réservé, cela peut prendre quelques temps. Pour 500 enregistrements de 40 caractères, cela prend environ 2.5 minutes.

La fermeture d'un fichier se fait via:

```
120 CALLM ERROR:CALLM CLOSE3
```

Toujours lire le canal d'erreur avant de fermer le fichier relatif. Un message d'erreur éventuel sera affiché sur l'écran.

ECRITURE DE DONNEES DANS UN FICHER RELATIF:

Une fois le fichier réservé comme décrit ci-dessus, les données peuvent être entrées. Dans notre exemple, nous avons 3 variables: NOM\$, ADR\$ et VILLE\$, avec chacune une longueur définie. Ces longueurs sont sauveées dans les variables LN, LA et LV:

```
nom:      NOM$      LN=20
adresse:  ADR$      LA=20
ville:    VILLE$    LV=15
```

Un court programme décrit pas à pas comment les données sont envoyées vers le fichier (l'utilisation de FDDMON/0 est nécessaire!).

```
10 INPUT NOM$:PRINT
20 IF LEN(NOM$)>LN THEN 10
30 NOM$=NOM$+SPC(LN-LEN(NAM$))
40 INPUT ADR$:PRINT
50 IF LEN(ADR$)>LA THEN 40
60 ADR$=ADR$+SPC(LA-LEN(ADR$))
70 INPUT VILLE$:PRINT
80 IF LEN(VILLE$)>LV THEN 70
90 VILLE$=VILLE$+SPC(LV-LEN(VILLE$))
```

Cette routine entre les données pour chaque champs de l'enregistrement. Les champs trop longs ne sont pas acceptés. Les champs trop courts sont ajustés à la longueur correcte par ajout d'espaces à droite.

Le nombre d'enregistrements dans lesquels les données doivent être entrées, doit être entré maintenant:

```
100 INPUT NUMENREG:PRINT
110 HB=NUMENREG SHR 8:LB=NUMENREG IAND 255
```

Maintenant le fichier relatif peut être ouvert:

```
120 NOMFICHER$="MAILING LIST":LONGENREG=56
130 OPEN$=NOMFICHER$+",L,"+CHR$(LONGENREG)
140 CALLM OPEN3,OPEN$
```

Après ouverture du fichier, le pointeur doit être initialisé à l'enregistrement voulu:

```
150 POS$="P"+CHR$(3)+CHR$(LB)+CHR$(HB)+CHR$(0)
160 CALLM PRNT15,POS$
```

Maintenant les données peuvent être envoyées vers les enregistrements concernés:

```
170 OP$=NOM$+ADR$+VILLE$
180 CALLM PRNT3,OP$
```

Cette procédure d'entrée peut être répétée pour d'autres enregistrements, par ex. via une boucle. Quand tout est fini, le fichier relatif doit être fermé:

```
190 CALLM ERROR:CALLM CLOSE3
```

LECTURE DE DONNEES D'UN FICHER RELATIF:

Les données peuvent être lues de deux façons depuis un fichier relatif: caractères par caractères (GET) ou comme chaîne (INPUT). La méthode GET n'est pratique que si seulement un caractère ou une partie d'enregistrement est requis. Si tout l'enregistrement doit être lu, la méthode INPUT doit être utilisée.

Dans l'exemple, les deux méthodes seront utilisées. La méthode GET pour voir si un enregistrement est vide, la méthode INPUT pour lire l'enregistrement.

La gestion est un petit peu 'artificielle' à cause de la façon dont sont liés le programme machine FDDMON/O et le BASIC. Deux variables doivent être déclarées au début du programme:

```
1 IN$=" "  
2 IP$=SPC(255)
```

La variable IN\$ est utilisée pour le stockage temporaire du caractère lu par la commande GET. Dans les 255 espaces de la variable IP\$, on stockera les caractères lus par la commande INPUT.

```
10 CALLM OPEN3,OPEN$           ) V 'écriture de données',  
20 CALLM PRNT15,POS$          ) lignes 100-160  
30 CALLM GET3,IN$  
40 IF ASC(IN$)=255 THEN 100
```

Si le premier caractère de l'enregistrement est HFF, alors l'enregistrement est vide.

```
50 CALLM PRNT15,POS$  
60 CALLM INP3,IP$
```

Cette seconde commande de positionnement est requise pour pointer à nouveau sur le début de l'enregistrement. Maintenant, tout l'enregistrement est sauvé dans IP\$, et doit être divisé en trois parties: le nom, l'adresse et la ville.

```
70 NOM$=MID$(IP$,0,LN)  
80 ADR$=MID$(IP$,LN,LA)  
90 VILLE$=MID$(IP$,LN+LA,LT)
```

Les différentes chaînes peuvent être mises à jour et réécrites comme décrit ci-dessus.

Ne pas oublier de fermer le fichier une fois que toutes les lectures et écritures sont terminées:

```
100 CALLM ERROR:CALLM CLOSE3
```

REFERENCES:

Pour une explication détaillée, avec de nombreux exemples, référez-vous au livre 'DAS GROSSE FLOPPY-BUCH' par ENGLISCH/SZCZEPANOWSKI, publié par DATA BECKER GmbH, DUSSELDORF (RFA). Ce livre est aussi traduit dans d'autres langues.

Jan BOERRIGTER, Juin 85
Traduction: Ch.POELS

Pour les utilisateurs du DAI-DOS 1541, une disquette contenant quelques utilitaires pratiques est disponible. Ils permettent au DAI-DOS 1541 d'utiliser toutes les ressources possibles du drive VC1541.

La disquette 'FDD-toolkit-1' contient les programmes suivants:

- Impression de la directory sur imprimante parallèle.
- Lecture de programmes CBM-64.
- Utilisation de commandes d'accès direct vers le VC1541.
- !!! - Manipulation de fichiers relatifs (enregistrements).
- Protection contre l'effacement de fichiers (LOCK/UNLOCK).
- Changement du numéro de drive (adresse physique).
- Affichage du contenu des secteurs.
- Programmes de démo.

La disquette peut être commandée par mandat international (disponible dans tous les bureaux de poste) pour la somme de 60 Florins, à l'adresse ci-dessous:

MICRO SERVICE
Fabritiusstraat 15
NL-6174 RG SWEIKHUIZEN
PAYS-BAS

Tél. 04493-2093 (19-21h)

Indiquez sur votre mandat: 'FDD TOOLKIT-1'.

Jan BOERRIGTER.
Traduction: Ch.POELS.

Programme

Voici un petit programme en LM de Pascal JANIN qui permet de se servir du DAI comme SUPER VU-METRE! (Prévu à l'origine pour un ampli.) Pour le démarrer, faire G330. Le programme visualise, sous la forme de 6 barres lumineuses (en 6 couleurs, 3 par 3), l'état des 6 entrées paddles.

```
-----
>D300 40F
0300 00 00 21 EF BF 0E 0B 36 D7 2B 36 40 2B 36 20 2B
0310 36 10 0D C2 0C 03 0E 0C 2B 36 20 2B 36 E0 0D C2
0320 18 03 2B 05 C2 05 03 C9 21 DC 03 EF 06 C9 00 00
0330 06 08 3E F0 32 11 03 3E 30 32 1D 03 CD 02 03 06
0340 08 3E C0 32 11 03 3E A0 32 1D 03 CD 05 03 06 14
0350 3E 10 32 11 03 3E E0 32 1D 03 CD 05 03 CD A5 03
0360 3E 38 CD B0 03 21 E7 B9 CD E0 03 3E 39 CD B0 03
0370 21 E3 B9 CD E0 03 3E 3A CD B0 03 21 DF B9 CD E0
0380 03 3E 3B CD B0 03 21 CF B9 CD E0 03 3E 3C CD B0
0390 03 21 CB B9 CD E0 03 3E 3D CD B0 03 21 C7 B9 CD
03A0 E0 03 C3 60 03 21 00 C0 22 72 00 C3 28 03 00 00
03B0 32 06 FD 3E 30 01 06 FC 02 21 FF FF 22 00 FC 3A
03C0 01 FD EB 3E 00 02 2A 00 FC CD 14 DE DA C2 03 CD
03D0 26 DE 11 CE FF 19 C3 03 04 00 00 00 00 02 00 0E
03E0 0F 0F 0F E6 1F 47 4F 04 11 30 00 05 CA F5 03 36
03F0 1E 19 C3 EB 03 3E 21 91 D8 CB 47 05 CB 36 20 19
0400 C3 FB 03 DA 09 04 21 00 00 7D 2F 25 C0 3E 00 C9
```



Nous souhaitons tout d'abord la bienvenue aux membres du club DCA (Dai Computer Association), nous espérons que vous trouverez tout ce dont vous aurez besoin au sein du club.

Pour arriver à ces fins, nous avons décidé de nous associer à IDC qui diffuse une revue interclub, DAIClic. Parallèlement, nous fournissons à nos membres des compléments d'informations propres à DCA, des programmes, des réalisations de cartes et bien d'autres choses. Nous comptons également produire des fascicules thématiques sur le MODEO, le son, le graphisme, les interfaces...

Le club vous propose entre autres :

- un service de documentation
- un service de réalisation d'interfaces
- des réunions
- ...

L'échange de programmes personnels entre nos membres est possible et souhaitable.

Afin d'avoir une gestion équitable pour chacun de vous, nous vous demandons de nous fournir des enveloppes suffisamment affranchies pour le retour ou l'envoi de tout document. Je pense que vous comprenez !

Pour plus d'informations :

Mr Hertzog Pascal.
2 av Jean Moulin
F-94350 Villiers/Marne.

Mr D'Arantes Luc.
& rue Defresne Bast
F-95100 Argenteuil.

P.S.

Le traitement de texte décrit à la page suivante est disponible gratuitement pour tous les membres d'IDC. Pour le recevoir, il suffit d'envoyer une enveloppe suffisamment affranchie (1) avec vos nom et adresse, ainsi qu'une cassette à DCA.

(1) Pour les DAiistes ne résidant pas en France, joindre des coupons réponses internationaux vendus dans toutes les postes.

DCA

TEXT-ICULE

Quand on dispose d'un ordinateur et d'une imprimante, il est normal de songer à l'utiliser pour faire son courrier, le mode d'emploi de certains programmes ou tenir de petits fichiers du genre DAIOthèque, videothèque et autres tranches de thèques...

Or le DAI possède un éditeur génial sans équivalent même sur les ordinateurs professionnels. Il est donc extrêmement tentant d'utiliser la souplesse et le scrolling dans toutes les directions fournis par la routines de la ROM.

Aussi via le club DCA je mets gratuitement à disposition de ceux qui désirent un petit traitement de texte baptisé TEXT-ICULE.

Ce programme présente les fonctions suivantes:
- Sur les 32K disponibles pour les données 31K sont dédiés au texte utilisateur ce qui représente environ 10 à 15 pages imprimées.

- Ces 31000 caractères peuvent être répartis par l'utilisateur en 2 pages mémoire gérées séparément par le programme. Soit plusieurs pages imprimées par page mémoire.

- Il est possible d'insérer une page mémoire en un point quelconque de l'autre page mémoire.

- Il est possible de transférer ou de copier tout ou en partie d'une page mémoire sur l'autre.

- Des deux points ci dessus, il découle qu'il est possible de déplacer ou de dupliquer du texte dans une page mémoire en utilisant l'autre comme manœuvre.

- Chaque page mémoire peut être lue ou écrite sur Memocom individuellement. Elle est écrite selon la dimension donnée par l'utilisateur, ce qui permet de sauver le texte sur microcassette à l'endroit où il a été lu, (sauf si on augmente la dimension)


- Il est bien sûr possible de lire dans une page plus grande lorsqu'il n'y a plus assez de place disponible dans la page mémoire.

Il est possible d'imprimer certaines parties de texte en caractères élargis grâce à des caractères de commande insérés dans le texte.

Sauts de page commandes par un caractère spécial en début de ligne.

The BILLOTLOGIC Company

Billot Michel



Ne soyez pas
un DAI-ISTE
tombé
du nid !

n e w s n e w s n e w s n e w s
e n
w I. D. C. Bordeaux. e
s w
n e w s n e w s n e w s n e w s

A L'INTENTION DE TOUS LES CLUBS ASSOCIES

De très bonnes nouvelles !!

Du coté de la BIBLIOTHEQUE :

Le CLUB de BORDEAUX édite 3 livres :

Les deux premiers sont spécialement à l'intention des personnes faisant leur premières armes sur le DAI.

Intitulé : ' APPRENDRE A PROGRAMMER SUR LE DAI '

- TOME 1 : explication et applications de 100 instructions et commandes sur le basic du DAI
- TOME 2 : exploitez la mémoire vidéo , avec la ROM de caractères nombreux exemples et applications

Chaque Tome, d'environ 70 a 80 pages est vendu 50 FFrancais (port inclus).

Le troisième livre intéressera toute personne désirant exploiter au mieux son DCR , en effet ce n'est ni plus ni moins que le 'Désassemblage Commente' de toutes les routines de la ROM du MEMOCOM . Ce travail de longue haleine, qui a demandé de longues journées de travail à l'auteur, est rédigé en Anglais dans un soucis d'universalité.

Intitulé 'TOS COMMENTE'

Il est distribué à 150 FFrancais (port inclus).

- Nous vous rappelons aussi que vous pouvez acheter grâce a I.D.C. Bordeaux.

Un ouvrage vendu par un club Flam and , rédigé en Anglais : la 'BIBLE' de l'assembleur pour DAI c'est à dire le désassemblage commente de toutes les routines de la ROM basic , intitulé 'FIRMWARE MANUAL'.

Du coté matériel HARD :

- De nouveaux contacts avec plusieurs commerçants ont été pris en raison de nombreuses demandes faites par nos Adhérents désirant acquérir une imprimante.

** De ceci il ressort que le meilleur rapport Qualité / Prix est actuellement en faveur de deux modèles d'imprimantes MANNESMANN TALLY (publicité gratuite) la 80 S et la 85 pour lesquelles des réductions très intéressantes ont été obtenues pour nos membres.

(Joindre à votre demande 5 FFrancais en timbres pour frais de photocopies)

De nouveaux contacts sont en cours pour d'autres marques telles les nouvelles SEIKOSHA scrutez bien vos prochains DAICLICS.

- Je vous rappelle que vous pouvez toujours profiter des prix obtenus par I. D. C. Bordeaux pour les achats de KEN-DOS ou de K7 DCR (= Memocom)
(Renseignements: joindre 4 timbres à 2.20 FFrancais pour frais de photocopie)

Encore de bonnes nouvelles!! :

Du coté de la LOGITHEQUE :

- TROIS nouveaux programmes (entre autres) valent la peine d'être plus particulièrement annoncés car ils sont d'une qualité professionnelle :

** Un Traitement de texte (un des plus performants que je connaisse)
' D A I W O R D S T A R '
entièrement en langage machine , une version Allemande et une ' Angliscisée !

** Deux jeux de 'stratégie' (WARGAMES) réalisés par un adhérent de I.D.C.Bx. Mr DUBREUIL dont la qualité et l'originalité m'avaient fait lui proposer une commercialisation de ces logiciels. Mr DUBREUIL a préféré répondre que son plus cher désir était que ses jeux soient distribués le plus largement possible . Que voila une décision qui apporte du DYNAMISME et un vrai esprit de GROUPE à un club !!

*** SI A CETTE OCCASION certains désirent adhérer à I.D.C. Bordeaux ils recevront , en signe de bienvenue , ces trois programmes ainsi que les modes d'emplois indispensables à leur fonctionnement GRATUITEMENT ; sur K7 Audio ou K7 DCR ou disk.KEN-DOS (support GRATUIT) (préciser le support désiré)

Du coté des nouvelles ACTIVITES :

- La liste des membres désirant avoir des contacts directs les uns avec les autres est enfin arrêté !!. Que ceux qui m'ont envoyé l'autorisation d'inscrire leurs coordonnées pensent à la réclamer. Si d'autres personnes désirent la recevoir il n'est pas trop tard pour demander son inscription , quel que soit votre Club d'affiliation à I.D.C.

- Elle est fournie sous forme de fichier texte de notre FABULEUX traitement de texte : DAIWORDSTAR , envoyez vos K7 / disquettes !!

- Une étude menée depuis peu nous permet déjà d'annoncer que dans les DAICLICS à venir vous trouverez toutes les indications nécessaires pour transformer votre DAI en SUPER MINITEL (normes Françaises) , pour un prix ridicule (quelques Francs). TOUS les problèmes importants ont été résolus (connection,protocoles,parité...) OUVREZ L'OEIL

- Le responsable de la 'Section Hard' , Mr PAUPERT (cf adresse dans le précédent numéro) désire étendre ses activités de coordinateur et propose à toute personne voulant faire partager ses 'projets avancés' à d'autres travaillant sur les mêmes vues , de prendre contact avec lui.

** Cependant afin de ne pas le submerger, ne seront pris en considération que les projets qui ont dépassé le simple stade d'intention mais qui sont en cours de réalisation.

** Le regroupement de mêmes projets peut faire profiter chaque réalisateur de l'expérience des autres.

*** QUE CHACUN APPORTE SA PIERRE ***

A L'INTENTION DE NOS ADHERENTS

et de TOUTE personne nous écrivant.

- Le volume du courrier arrivant au CLUB grandissant chaque jour il devient indispensable de prendre quelques mesures :

Des détails auxquels nous n'avions pas pensé :

* Afin de nous éviter de longues heures de recherche dans les nombreux programmes de la LOGITHEQUE ou articles de la BIBLIOTHEQUE vous devrez recopier , en plus du titre du programme ou de l'article demandé , le COMMENTAIRE et la CATEGORIE indique dans la LISTE du CLUB

* Lorsque vous remboursez certains frais en timbres prière de n'envoyer que des timbres a 2.10 FF. ou 5 FF. car les autres ne nous sont d'aucune utilité et si les timbres de quelques centimes peuvent toujours être utilisés il n'en est pas de même pour ceux à 10 FF. par exemple !

* De plus pour des raisons imposées par le '!' MINISTERE de L'INTERIEUR !!' je demande à tous d'éviter de (me) téléphoner au CLUB le WEEK END !!

* En raison du calcul très juste de notre fond de roulement, la majeure partie de l'abonnement servant à éditer la REVUE que vous lisez actuellement , nos diverses activités fonctionnent grâce au bénévolat des membres du bureau , aussi pour ne pas nous rendre la tâche plus ingrate il vous est demandé de ne pas oublier de joindre à TOUT courrier une enveloppe à votre adresse , suffisamment timbrée , pour la réponse.

J'espère ainsi que nous pourrons continuer nos activités présentés et poursuivre notre développement dans différents domaines.

Le Président et Trésorier
I. D. C. Bordeaux
s/c DELANNAY Bruno
Res. les ACACIAS Bt. B3
Av. de SAIGE
33600 PESSAC

(56) 45.87.70



JEU-CONCOURS

Le jeu-concours du n°2 de DAICLIC a connu un énorme succès. Merci à tous ceux qui ont bien voulu y participer, surtout à ceux qui savaient ne pas arriver les premiers. Cette fois nous proposerons d'ailleurs un concours où la vitesse de réponse ne dépendra plus de la poste mais de votre sagacité !

En ce qui concerne les réponses elles-mêmes, on peut les classer en 3 catégories :

1. la majorité d'entre vous a résolu le problème en écrivant deux boucles du genre :

```
10 FOR A=10 TO 99:FOR B=0 TO 99
20 C=100*A+B: IF C=A*A+B*B THEN PRINT C
30 NEXT:NEXT
```

Le temps mis par ces programmes pour trouver la bonne réponse tourne autour des 2 minutes.

2. certains ont résolu le problème avec une seule boucle :

```
10 FOR A=10 TO 99
20 C=A*A:S=100*A
30 B=INT(SQR(S-C))+1:C=C+B*B:S=S+B
40 IF C=S THEN PRINT S
50 NEXT
```

Le temps de réponse tombe alors à quelques secondes seulement.

3. ceux qui ont utilisé les fonctions du genre LEFT\$,RIGHT\$,MID\$,STR\$,VAL...

Le temps de réponse peut dans ce cas monter au-delà des 15 minutes quand on utilise deux boucles ...

A signaler enfin quelques réponses originales utilisant une démonstration indiquant que la réponse est basée sur la deuxième racine d'une équation du second degré :

```
10 PRINT STR$((100+SQR(10000-4*1056))/2)+"33"
```

La réponse est alors instantanée !

Si vous désirez donc la connaître rapidement, vous savez ce qu'il vous reste à faire.

Quant au gagnant, il s'agit de Jean-François BLONDET de NEUFCHATEAU en BELGIQUE.

Encore un grand bravo et un grand merci à tous pour vos encouragements.

Voici donc un nouveau jeu-concours où il s'agit cette fois d'écrire en BASIC un programme trouvant le plus rapidement possible la réponse au problème suivant :

Le nombre 217848712 est un palindrome, c'est-à-dire qu'il reste identique à lui-même qu'on le lise de gauche à droite ou de droite à gauche. Il suffit de placer avant le 4, le signe x, et après, le signe =, pour obtenir une multiplication exacte.

En effet : $2178 \times 4 = 8712$

Il existe un autre palindrome, ne commençant pas par 1, qui a cette forme-ci :

. x =

Quel est ce nombre de 9 chiffres ?

Vos réponses doivent parvenir à Jacques MOENS
Clos Fontaine des Ducs, 6 à 1310 LA HULPE (Belgique).
Le programme trouvant le plus rapidement la réponse exacte
vaudra à son auteur de recevoir une cassette bourrée de
programmes.

B O N N E C H A N C E !

D-ROTATION

```
10 MODE 6:COLORG 6 0 8 9:MODE 6:R%=30
20 CY%=100:CX%=100:Y%=100:L%=R%*R%:A%=2:DX%=A%:DY%=A%
30 CY%=CY%+DY%:CX%=CX%+DX%:Y%=2+Y%
40 IF GETC =32 THEN COLORG RND (15.0) RND (15.0) 0 0
50 IF CX%+X%>330 THEN DX%=-A%:CX%=CX%+DX%
60 IF CX%-X%<4 THEN DX%=A%:CX%=CX%+DX%
70 IF CY%+Y%>250 THEN DY%=-A%:CY%=CY%+DY%
80 IF CY%-Y%<4 THEN DY%=A%:CY%=CY%+DY%
90 IF Y%>R% THEN Y%=-R%
100 X%=SQR(L%-Y%*Y%)
110 DRAW CX%-X%,CY%+Y% CX%+X%,CY%-Y% 17
120 GOTO 30
```

Le CLUB DAIC· souhaite une excellente fin de vacances
à tous les lecteurs de DAICLIC.

Il rappelle à ses membres que les réunions bimensuel-
les (1ers mercredis et 3èmes dimanches du mois) reprendront dès
le 4 septembre 1985 à l'adresse habituelle :
434, chaussée d'Alsemberg à 1180 Bruxelles.

DAIlassement numéro 3.

Hello les intellectuels, content de vous revoir pour un DAIlassement plus que poussé (eh oui les vacances n'arrangent pas les choses et le soleil non plus) !

Pour vous en sortir dans ce DAIlassement, il vous suffira de deviner ce qui apparaîtrait à l'écran après un RUN du programme. Attention, il est interdit de taper ce programme, cela serait trop facile !

```
10 REM ##### EXCLUSIVE COPYRIGHT BY DAIBIL 1985
20 REM (JUSQUE LA, CA VA !)
30 A$="LUT":B$="SA"
40 GOSUB 1000
50 PRINT A$;B$;ATA$="CHOPE"
60 PRINT "VOUS";CHR$(#20);"ETES VRAIMENT ";CHR$(68);"OUE"
70 PRINT "POUR CE":GOTO 1200
80 PRINT "D'EXERCICE.":PHRASE$="EH BIEN CA ALORS ! "
90 A$="TANT MIEUX":B$=" MAIS MAINTENANT":C$=" POUR"
100 D$=" LE PLUS":E$=" VOUS":F$="DUR VIENT !"
110 GOSUB 10000
120 REM PROGRAMME PRINCIPAL / THE big blague !
130 PRINT "C'EST L'HISTOIRE";:PRINT " D'UN DAI"
140 READ DAICLIC$:PRINT DAICLIC$
150 READ TOTO$:PRINT TOTO$
160 PRINT "LE GARCON BIEN EVIDEMMENT TRES ETONNE SE DIT : "
170 PRINT CHR$(#22);"JE VAIS LE PIGEONNER UN ";
180 PRINT "B";"O";"N";" COUP !";CHR$(#22)
190 GOSUB 12000
200 PRINT "JE VAIS LUI FAIRE PAYER 100 $ POUR SA "
210 PRINT "PECHO";CHR$(#22):REM dernier mot ecrit en verland)"
220 PRINT "LE DAI, ASSIS AU COMPTOIR VIDE SA ";ATA$;" ET"
230 PRINT "DEMANDE":PRINT SPC(1);"LE PRIX DE LA ";ATA$;SPC(1);
240 PRINT "POUR PAYER. LE GARCON REPONDS : 100 $"
250 READ BLUF$:PRINT BLUF$
260 PRINT "LE GARCON DIT TOUT HAUT ";PHRASE$
270 PRINT "PREMIERE FOIS QUE JE VOIS UN DAI DANS MON CAFETAR !"
280 PRINT "SUR CE";:READ REP$:PRINT " : ";REP$
290 PRINT CHR$(#22);"ET AU PRIX OU SONT LES CHOPEES. C'EST AUSSI"
300 PRINT "L";CHR$(#41);" DERNIERE !";CHR$(#22)
310 PRINT :PRINT "VACHEMENT ";DAIBIL$;" NON ?":GOTO 15000
1000 C$=A$:A$=B$:B$=C$
1010 RETURN
1200 PRINT "GENRE"
1201 DATA QUI RENTRE DANS UN CAFE ET QUI
1210 GOTO 80
1400 DATA COMMANDE UNE BIERE (UNE CHOPE; UNE PINTE !)
1401 REM COMME ON DIT ICI
10000 PRINT A$;C$;E$;" ";B$;D$;" ";F$
10010 RETURN:PHRASE$="EH BIEN CA ALORS; C'EST LA"
12000 PRINT CHR$(#22);:DAIBIL$="INTELLECTUEL"
12010 RETURN
14000 DATA LE DAI PAIE SANS RIEN DIRE ET S'EN VA
14001 DATA LE DAI REPOND
15000 END:REM END (il en faut de temps en temps!)
```



TEST

TEST du nouveau DOS 3.0

Encore un nouveau système de floppy disk pour le DAI : la version 3.0 de la société PRODATA !
Rappelons que cette firme a commercialisé les versions DOS 2.0 et DOS 2.1 qui laissaient fort à désirer. La situation a bien changé, fort heureusement !

Fiche technique du système :

- deux unités de 160K (320K) simple face, simple densité, 40 pistes de 16 secteurs de 256 octets pour environ 60000 FB
- ou : - deux unités de 320K (640K) simple face, double densité, 80 pistes de 16 secteurs de 256 octets environ 70000 FB

Ce test a été fait sur un système 640K, mais les commandes du DOS, objet de notre curiosité, sont identiques sur les deux versions.

Occupation de la mémoire :

Le DOS complet prend 10K de RAM, il laisse donc 34745 bytes à l'utilisateur. Le TINYDOS ne prend plus que 2,5K de RAM, il laisse donc 42383 bytes à l'utilisateur (le DOS complet est placé de F300 à F2B90 au chargement).

Vitesse :

Temps de chargement du DOS : 11s dont 3s d'attente pour la détection du DCE-BUS
BACKUP d'une disquette de 320K : 2 min 20 s
Chargement d'une page en MODE 5: 9s
Création-ouverture-écriture de 300 enregistrements de 9 caractères puis fermeture du fichier : 35s

Caractéristiques particulières :

- utilisation possible de MDCR ou de lecteurs de cassettes.
- allocation dynamique : réutilisation de l'espace disponible sur la disquette.
- gestion de fichiers à accès séquentiel et à accès direct (16 fichiers simultanément).
- possibilité de déplacer le DOS en mémoire ou de réduire sa taille (en TINYDOS, mais alors, nous n'avons plus accès aux commandes de lecture/écriture).
- gestion des erreurs.
- structure des fichiers compatible avec le système CP/M.
- autostart des programmes LM ou Basic.
- en option, utilisation du CP/M 2.2 sans modification de hardware.

Les commandes sont faciles et assez claires dans l'ensemble.

La procédure à suivre pour l'encodage de vos programmes est agréable et inédite :

première ligne	DOS
à chaque ligne suivante,	une commande
dernière ligne	NODOS

Chaque ligne de ce bloc de commandes est directement et automatiquement transformée pour être compréhensible par le DOS.
Par exemple, NODOS devient POKEF131,PEEK(F1E):REM NODOS.
Dommage cependant que cette facilité ne fonctionne pas dans l'éditeur mais uniquement l'encodage de votre programme.

Liste des commandes :

- *MOVEDOS : déplacement du DOS en mémoire (très utile pour les programmes LM).
 - *TINYDOS : version réduite du DOS gérant uniquement les lectures/écritures (très utile pour les longs programmes tournant en mode graphique).
 - *ASSIGN : pour allouer les périphériques en entrée/sortie : cassette, MDCR, disque, écran, imprimante, clavier, fichier...
 - *AUTO : numérotation automatique des lignes de votre programme par incrémentation de 10.
 - *RENUMBER : par incrémentation de 10 également.
 - *ON ERROR GOTO : gestion des erreurs dans les commandes du DOS.
 - *DATE, IDISK (formatage), DELETE, COPY, BACKUP, RENAME, PROTECT (protège contre l'écriture), RECOVER (déprotège), VERIFY, RESETD (pour réactiver le DOS).
 - *DIR : pour obtenir un directory avec affichage contrôlable par l'utilisateur (différence avec la version 2.0).
Vous pouvez stopper le défilement des noms de fichier à l'aide d'une touche du clavier et le relancer en pressant la barre d'espace (comme avec la commande LIST du BASIC).
 - *COMPACT : optimisation de la gestion de la place libre sur une disquette après de nombreuses manipulations.
 - *CREATE, OPEN, OPENI (ouverture en lecture uniquement), CLOSE, CLOSEA (fermeture de tous les fichiers).
 - *Commandes concernant les fichiers à accès direct :
 - FIELD pour déterminer les zones d'un enregistrement
 - GET lecture
 - PUT écriture
- La longueur de l'enregistrement varie au gré de l'utilisateur et il n'y a ni conversion ni manipulation intermédiaires à effectuer avant de sauver une donnée sur le disque. Cette gestion de fichiers est nettement plus simple qu'en BASIC MICROSOFT.
- *LOAD, SAVE, LRUN chargement + exécution d'un programme.
 - *DLOAD, DSAVE : programme en LM.
 - *SAVET, LOADT : programme sous forme de fichier texte ;
utile pour des fusions, pour un traitement de textes ou pour le transfert vers d'autres machines.
 - *RDSK, WDSK : lecture ou écriture directe d'un secteur de la disquette.
 - *SIZE : détermination de la longueur d'un fichier.
 - *SEEK : recherche de l'existence d'un fichier.
 - *COMMAND, STORE : pour diriger le flux des commandes vers une disquette ou vers le DOS (emploi d'une commande DOS lorsque l'on se trouve sur un autre périphérique (MDCR,...)).

En résumé, ce nouveau DOS 3.0 relance l'intérêt de l'emploi des drives d'INdata qui, sous DOS 2.x laissaient fort à désirer ! Ce DOS est un travail sérieux fourni par INdata (espérons que ce travail sera largement récompensé).

Le prix d'une adaptation d'un système 2.0 en 3.0 est d'environ 9000 FB pour de plus amples renseignements, contacter DAI-Groupe PRODATA, 412 Bte 11 Avenue de Tervuren, 1150 Bruxelles.

Il paraît toutefois que la société PRODATA va commercialiser des lecteurs de faible encombrement et permettant de travailler en double face, soit 320 ou 640K par disquette. Cela reste encore à prouver, inutile de vous rappeler que ce DOS devait déjà sortir l'année dernière. Affaire à suivre, donc...

Fernand LEMOINE, club CAROLODAI pour DAICLIC.

Test du DOS V3.0

Vous trouvez donc un deuxième article traitant du nouveau DOS de la société DAI - Groupe PRODATA.

Le GROS avantage du système est son jeu d'instructions pour l'emploi de fichier à accès direct sur la disquette. De ce point de vue, le DOS V3.0 surpasse, ECRASE la plupart de ses concurrents. A coté de cela, les commandes en elles-mêmes sont assez complètes, mais le jeu d'instructions n'est peut-être pas assez 'poussé' !

Après 15 jours d'essais fructueux (ou infructueux !), je vous livre mes petites remarques quand à ce système (je tiens à signaler que je suis possesseur d'un Ken-Dos) :

- Il est très (voire trop) facile de 'planter' le système. En effet, certaines fautes de syntaxe dans les commandes peuvent faire sortir les commandes DOS de la mémoire. Dans un tel cas, vous êtes obligé de faire un RESET pour recharger le DOS, mais vous aurez bien évidemment perdu ce que vous aviez en mémoire ! Ce qui est loin d'être agréable. Personne n'est à l'abri d'une 'fote de frape'.

- Il y a également certains problèmes avec les messages d'erreurs du DOS. Il y a énormément de messages différents (ce qui est un bien en soi), mais leur action n'est pas toujours évidente.

Quand vous faites une erreur, soit le message apparaît lisiblement en anglais, soit un nombre hexadécimal correspondant à une erreur. Vous devrez donc vous reporter au manuel pour voir quelle est l'erreur. Le problème vient du fait que ces messages d'erreurs ne viennent pas toujours au bon moment.

Par exemple, vous essayez de charger un programme qui ne se trouve pas sur la disquette, cela donne ceci :

```
*LOAD "TOTO" (TOTO n'existe pas !)  
  (un peu de patience, le drive tourne... et puis :)  
      FILE NOT FOUND      (logique)  
      FILE NOT OPEN       (moins logique !)  
un deuxième FILE NOT OPEN (de moins en moins logique !)  
et enfin : +++ DRIVE ERROR 01 +++ (ce qui, d'après le manuel,  
correspondant à 'SYNTAX ERROR').  
Vous constatez donc qu'il y a quelque chose d'anormal dans ces  
réponses du DOS ! Attention, je ne dis pas que tous les messages  
d'erreurs ne marchent. Je pourrais cependant encore vous citer  
quelques autres irrégularités...
```

- Le mode d'emploi compte une cinquantaine de pages, est clair, précis et en anglais...mais il n'explique que l'emploi des commandes en BASIC (avec des exemples), rien n'est mentionné pour celui qui désire accéder aux routines DOS par l'Assembleur, aucune 'carte' de la mémoire et aucun 'point d'entrée' dans le DOS ne sont donnés !!!

- Le système d'encodage de vos commandes disques dans un programme (voir article précédent) est très pratique et original. Sans plus !

- Anecdote 'amusante': quand vous mettez une disquette dans le lecteur vous entendez un petit clic qui est dû à la mise en place du mécanisme qui éjectera la disquette lorsque la 'porte' du lecteur s'ouvrira à nouveau. Et bien ici, vous n'entendez pas qu'un 'clic' même un DAIClic, vous entendez également le moteur des drives qui se met à tourner et puis qui s'arrête au bout de 10 secondes.

De même, quand le système est allumé et que vous travaillez normalement sans 'jouer' avec des commandes DOS, le moteur des drives se met en

marche de temps en temps puis se coupe après 10 secondes, tout seul ?

- On reproche évidemment au DOS 3.0 la place qu'il prend en mémoire (en effet 10 KB, ce n'est pas rien). Cette place est bien évidemment compensée par de très bonnes commandes... Et puis, me direz-vous, il existe le TINYDOS qui diminue la place prise par le DOS de 10K à 2.5K, en ne gardant que les fonctions lecture/écriture, mais gros problème : les fonctions pour faire des fichiers à accès direct sont perdues, principal attrait du système ! Or, avec 34 k de RAM (donc quand le DOS complet est en mémoire), il suffit d'un programme assez développé pour arriver à cela. Ne serait-il pas envisageable pour PRODATA de créer un SEMI-TINYDOS qui prendrait moins de 10 K de RAM mais qui laisserait les fonctions d'accès direct à l'utilisateur ?

- Un programme est livré avec le DOS, permettant de lire et de transcrire en DOS V3.0 des fichiers sauvés sous DOS V2.x . Malgré toute ma bonne volonté ainsi que celle d'un ami possesseur du DOS V2.1, nous ne sommes pas parvenus à lire une disquette de l'ancien DOS sur le nouveau. Peut-être un problème avec le programme qui nous a été donné à tester ?

- Un CP/M tournerait sur ce nouveau DOS. Je n'ai pas eu l'occasion de le voir (raison pour laquelle j'emploie la forme conditionnelle), mais cela ne me semble pas trop intéressant sur une machine telle que le DAI car nous n'aurons pas énormément de mémoire libre une fois le CP/M en mémoire ce qui risque d'être très vite ennuyeux... Autre chose, le DAI ne tourne que sous 60 colonnes et non sous 80 (standard CP/M), ce qui peut être assez gênant. Cela n'est pas vrai pour tous les programmes CP/M, certains ne nécessitent pas les 80 colonnes.

- Deux commandes me déçoivent également quand je compare le système au KEN-DOS et au 'petit' système Commodore :

* La commande COMPACT du DOS v3.0 formate la disquette 'cible' avant de recopier les programmes de la disquette mère . Cela aurait été mieux si, comme dans les deux autres systèmes cités plus haut, le DOS rajoutait les programmes à compacter à la suite d'autres programmes se trouvant déjà sur la disquette 'cible' sans les effacer.

* La commande DIRECTORY n'est pas aussi efficace que sur les deux autres systèmes (toujours cités plus haut !) : le DOS se contente de vous donner simplement les titres des fichiers enregistrés sur la disquette. Pas moyen donc de pointer le programme voulu dans le directory pour le charger directement, etc...

Ne croyez cependant qu'avec toutes les critiques que je formule à l'égard de ce système, je le méprise pour autant. Il n'en est rien. Je vous fais juste part de quelques petites choses qui m'ont frappé ou déçu. Puisse INdata (pardon,DAI-Groupe PRODATA) y remédier.

En bref, ce système est très performant pour tout ce qui est gestion de fichier. Ce système peut servir à un niveau semi-professionnel mais la faible mémoire libre (34K) que laisse ce DOS risque de décourager plus d'un !

Marc Vandermeersch, DAIC Bruxelles pour DAIClic.

BASIC

```
1 REM SNAKE II - Pascal JANIN
2 ENVELOPE 0 16
10 CLEAR 130:DIM N(16.0),O(8.0)
11 GOSUB 20000
12 XCURS=PEEK(#75):POKE #75,32
15 COLORT 0 13 1 14
600 PRINT CHR$(12)
610 POKE #B46B,#6A
615 POKE #B46A,#DA
620 GOTO 700
650 FOR I=#BF69 TO #B5FD STEP -134.0:POKE I,#77:NEXT I
700 SECOURS%=3:SCORE%=0
710 FUEL=9.0
715 POKE #BFEE,#D2
800 FOR I=0.0 TO 59.0
801 CURSOR I,23:PRINT CHR$(29):SOUND OFF
802 CURSOR I,3:PRINT CHR$(29)
803 NEXT I
804 FOR I=4.0 TO 22.0
805 CURSOR 0,I:PRINT CHR$(29):SOUND OFF
806 CURSOR 59,I:PRINT CHR$(29)
807 NEXT I
808 POKE #BF68,#DD
809 POKE #B46A+2*134,#D2:POKE #B46A+134,#DA
810 CURSOR 3,22:FOR G=1.0 TO 30.0
820 X%=RND(55.0)+5.0:Y%=RND(16.0)+3.0
821 POKE #BFEE-(#B6*Y)-(2*X%),29
822 NEXT
832 FOR I%=1 TO 16:N(I%)=#BEDF-2.0*I%+#A:NEXT
840 FOR I=1.0 TO 15.0:POKE N(I),127:NEXT I:POKE N(16.0),17
845 POKE N(16.0)-3.0,#FF
850 GOTO 6060
990 ENVELOPE 0 16:Y=17.0:X=15.0:P%=0:M%=0
996 ENVELOPE 1 16,10;0,10;
1000 B=GETC
1001 POKE #B431,INT(FUEL+48.0):POKE #B431-3,0
1002 FUEL=FUEL-7E-3
1003 IF FUEL<=(-7E-2) THEN 8000
1004 IF FUEL<1.0 THEN POKE #B431-3,255:SOUND 1 1 15 1 FREQ(3520.0)
1005 IF B<16.0 OR B>19.0 THEN WAIT TIME 2:GOTO 2000
1010 POKE N(1.0),32:POKE N(16.0),127
1011 POKE N(16.0)-3.0,0
1012 CALLM #16DA
1015 FUEL=FUEL-2.85E-2
1025 NOISE 0 15
1030 IF B=16 THEN N(16.0)=N(16.0)+#86
1040 IF B=17.0 THEN N(16.0)=N(16.0)-#86
1050 IF B=18.0 THEN N(16.0)=N(16.0)+2.0
1060 IF B=19.0 THEN N(16.0)=N(16.0)-2.0
1070 IF PEEK(N(16.0))<>32.0 THEN 5000:REM analyse si obstacle
1080 POKE N(16.0),17:POKE N(16.0)-3.0,#FF:NOISE OFF
2000 P%=P%+1:IF P%<M% THEN 1000
2005 POKE #BFEE-(#B6*Y)-(2*X%),32
2006 POKE #BFEE-(#B6*Y)-(2*X)-3,0
2010 SOUND 0 0 15 1 FREQ(RND(880.0)+440.0)
2015 Y=INT(RND(16.0)+3.0):X=INT(RND(52.0)+5.0):IF PEEK(#BFEE-(#B6*Y)
-(2*X))<>32 THEN 2015
2020 M%=RND(70.0)+35.0:P%=0
2030 POKE #BFEE-(#B6*Y)-(2*X)-3,#FF
2031 POKE #BFEE-(#B6*Y)-(2*X),RND(6.0)+33.0
2040 SOUND OFF
```

```

2050 GOTO 1000
4999 REM determination du type d'obstacle rencontre
5000 Z=PEEK(N(16.0))
5020 IF Z=127 THEN 6000:REM queue
5030 IF Z=29.0 THEN 7000:REM bord de l'ecran
5040 IF Z>32.0 AND Z<=38.0 THEN 5200:REM obstacle mange
5050 SOUND 0 0 15 1 FREQ(2092.75)
5060 CURSOR 0,1:PRINT "<<< QUOI ??? >>>"
5070 WAIT TIME 100
5080 CURSOR 25,1:PRINT "FIN DU JEU ."
5090 GOTO 65530
5199 REM obstacle mange
5200 SC%=INT((104.0-M%)/8.625)+1.0
5250 NOISE OFF
5300 FOR C=1.0 TO 5.0:SOUND 0 0 15 2 FREQ(783.73)
5400 POKE N(16.0),SC%+48
5450 WAIT TIME 11:SOUND 0 0 15 2 FREQ(1046.37)
5500 POKE N(16.0),17:WAIT TIME 6.5
5550 NEXT C:SOUND OFF :SCORE%=SCORE%+SC%
5555 IF FUEL>=8.5 THEN FUEL=9.999:GOTO 5600
5560 FUEL=FUEL+0.6+(0.1*SC%)
5600 CURSOR 0,1:PRINT "SCORE =";SCORE%;" pts"
5700 IF SCORE%>=100.0 THEN 10000:REM victoire !!!
5800 GOTO 2010
5999 REM rencontre avec sa queue
6000 NOISE OFF :FOR C=1.0 TO 5.0:SOUND 0 0 15 0 FREQ(50.0)
6010 POKE N(16.0),29:POKE N(16.0)-3.0,#FF
6020 WAIT TIME 15:SOUND 0 0 15 0 FREQ(31.0)
6030 POKE N(16.0),127:POKE N(16.0)-3.0,0:WAIT TIME 10
6040 NEXT C:SOUND OFF
6050 WAIT TIME 50:SECOURS%=SECOURS%-1:IF SECOURS%<0 OR FUEL=0 THEN 8000
6054 POKE N(16.0),29
6055 FOR I=1.0 TO 15.0:POKE N(I),32:NEXT I
6056 POKE #BFEF-#86*Y-2*X,32:POKE #BFEF-(#86*Y)-(2*X)-3,0
6057 G=GETC:G=GETC:G=GETC:G=GETC:G=GETC:G=GETC:G=GETC:GOTO 832
6060 CURSOR 0,1:PRINT "SCORE =";SCORE%;" pts"
6065 CURSOR 18,1:PRINT "FUEL ="
6070 CURSOR 31,1:PRINT "VIE No";SECOURS%
6080 GOTO 990
7000 NOISE OFF :FOR C=1.0 TO 5.0:SOUND 0 0 15 0 FREQ(31.0)
7010 POKE N(16.0),17:POKE N(16.0)-3.0,#FF
7020 WAIT TIME 10:SOUND 0 0 15 0 FREQ(50.0)
7030 POKE N(16.0),29:POKE N(16.0)-3.0,0:WAIT TIME 13
7040 NEXT C:SOUND OFF
7050 GOTO 6050
8000 REM defaite .....
8010 POKE #75,XCURS
8015 FOR R=0.0 TO 10.0
8016 IF R MOD 4.0=0.0 THEN RESTORE
8017 READ AS%
8020 FOR I=1.0 TO 16.0
8030 SOUND 0 0 15 0 FREQ(180.0-I*9.0)
8040 POKE N(I),AS%
8060 NEXT
8070 NEXT
10000 G=GETC:G=GETC:G=GETC:G=GETC:G=GETC:G=GETC:G=GETC:G=GETC
10005 IF GETC<>0.0 THEN 65529
10010 COLORT 1 14 0 13
10020 SOUND 0 0 15 0 FREQ(880.0):WAIT TIME 15
10030 COLORT 0 13 1 14
10040 SOUND 0 0 15 0 FREQ(523.0):WAIT TIME 15
10050 IF GETC<>0 THEN 65529
10060 GOTO 10005

```



```

20000 MODE 0
20001 COLORT 0 5 0 10
20010 PRINT CHR$(12)
20011 0(1.0)=15.0:0(2.0)=12.0:0(3.0)=1.0:0(4.0)=9.0:0(5.0)=9.0:0(6.0)
    =1.0:0(7.0)=12.0:0(8.0)=15.0
20020 FOR I%=0 TO 7:POKE #BF68-I%*134,#D0+0(I%+1.0):NEXT
20030 CURSOR 0,22
20040 PRINT "   SSS   N   N   A   K   K   EEEE"
20045 PRINT "  S   S  NN  N   A A   K K   E"
20050 PRINT "   S   N  NN  A   A   K K   E"
20055 PRINT "    S   N  NN  A   A   KK    EEE"
20060 PRINT "     S   N  NN  AAAAA  K K   E"
20065 PRINT "S   S   N  NN  A   A   K K   E"
20070 PRINT " SSS   N   N   A   A   K   K   EEEE"
20080 POKE #BF68-I%*134,#D5
20081 PRINT
20082 POKE #75,32
20085 FOR I%=0 TO 20
20086 CURSOR I%,13:PRINT " ";
20087 NOISE 0 15
20090 FOR J%=1 TO 15:POKE #76,0:PRINT CHR$(127);:NEXT:POKE #76,#FF:PRINT
    CHR$(17)
20095 NOISE OFF :WAIT TIME 1
20096 NEXT
20100 POKE #BF68-11*134,#DE
20110 PRINT
20115 PRINT "DEPLACEZ LE 'SNAKE' (en francais : SERPENT)"
20116 PRINT "A L'AIDE DES TOUCHES DU curseur, AFIN DE"
20117 PRINT "GOBER LES VITAMINES bleues et jaunes"
20118 PRINT "QUI APPARAISSENT ET DISPARAISSENT UN PEU PARTOUT."
20119 PRINT "UNE VITAMINE GOBEE VOUS DONNE: 1) DU fuel"
20120 PRINT "                                     2) DU bonus"
20130 PRINT "PLUS ELLE VOUS DONNE DE FUEL ET DE BONUS,"
20131 PRINT "MOINS ELLE RESTE AFFICHEE LONGTEMPS !"
20132 PRINT "DEPECHEZ-VOUS ALORS !!!!!"
20135 PRINT "a 100 pts , VOUS AVEZ GAGNE !"
20136 PRINT "ATTENTION AUX MURS ,AUX OBSTACLES ..."
20137 PRINT "ET NE VOUS MORDEZ PAS LA QUEUE !!!!! ";
20138 WAIT TIME 800:RETURN
29999 STOP
65525 DATA 47,35,92,42
65529 REM fin du jeu
65530 NOISE OFF :SOUND OFF :CURSOR 0,1:POKE #75,XCURS

```



HARDWARE

CARACTERES 16 COULEURS EN MODE 0

Sur d'anciennes révisions du DAI, il est impossible d'afficher du texte (MODE0) en utilisant les 16 couleurs. La modification proposée le permet:

Article original: DAINAMIC GERMANY (RFA)

Traduction: Andres KURT - BUTGENBACH (B)

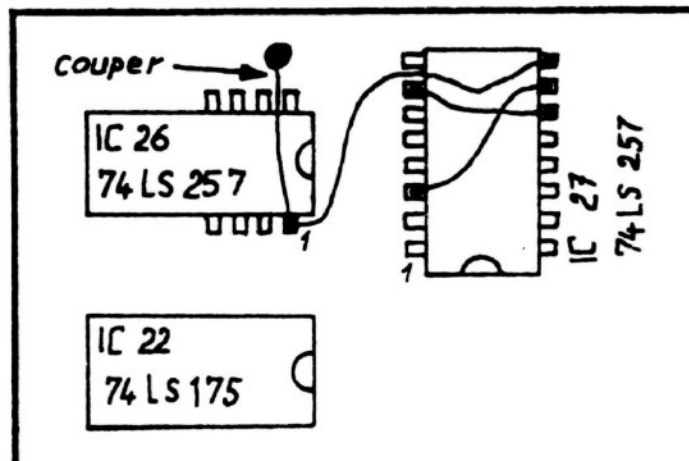
Matériel utilisé:

- fer à souder 40W
- pompe à désouder
- un couteau bien aiguisé
- du fil isolé

Cette modification n'est possible que sur les machines de révision 3, 4 et 5. Veillez à ce que les soudures soient bien faites et que les pistes du circuit imprimé soient bien coupées.

Travail à exécuter:

- couper la piste du circuit imprimé à l'endroit indiqué sur le schéma près du circuit intégré IC 26 (74LS257) pin 1.
- relier la pin 3 à la 10 de l'IC 27.
- relier la pin 7 à la 11 de l'IC 11.
- relier la pin 9 de l'IC 27 à la pin 1 de l'IC 26.



Côté soudures.

PETITES ANNONCES GRATUITES

- Offre à saisir: à vendre DAibasic V1.1: 20000FB + MEMOCOM neuf et 6 cassettes MDCR remplies de programmes: 16000 FB. Le tout pour 25000 FB. Frédéric DUFOUR - Ambrch 6 - 4090 BPS 37 - F.B.A.

- A vendre décodeur (TELEREADER 670) RTTY/CW/ASCII tous modes et vitesses avec interfaces vers imprimante, moniteur ou TV FF3000/FB19500. Possibilité d'échange contre station CB de base JUMBO HAM. Echange livre américain "8080/8085 ASSEMBLY LANGUAGE PROGRAMMING" indispensable pour l'assembleur contre FIRMWARE MANUAL + SERVICE MANUAL. Philippe FRILLEY - 250, Chaussée de Jules CESAR - F-95600 EAUBONNE.

- Cherche personnes intéressées par des contacts divers au niveau du DOS 3.0 et du CP/M (possède double drive PRODATA 2 x 160K). Guy LATRAN - 105, Avenue Gilson - B-6808 IZEL.

- Vends: cassettes DCR CERTIFIEES: 1600 FB les 6 (port compris); cassettes DCR non certifiées: 700 FB les 6 (port compris) (cause achat disquettes); EPROM nouveaux caractères DAI: 700 FB. Manuel DAI en français: 350 FB. Christian POELS (IDC).

- Vends MEMOCOM état neuf + 2 boites de cassettes: 9999 FB sans les cassettes. Wilfried GOHIMONT - 11, Rue du 12^e ligne - B-4800 VERVIERS

REMARQUE: Ces petites annonces gratuites pour les abonnés sont exclusivement réservées à des propositions entre particuliers sans objectif commercial et relatives à l'informatique. DAICLIC se réserve le droit de refuser une annonce sans avoir à fournir de justification.

Dernière minute

La société DAI nous communique les informations suivantes :

- La fonction 'END OF FILE' n'existe pas dans le DOS 3.0 mais on peut la simuler par l'utilisation de la commande 'ON ERROR GOTO'.
- Une nouvelle version du DOS, la V3.1, est actuellement fournie avec 4 types de drives :

1 x 320 K : 36.500 FB	2 x 320 K : 49.500 FB
1 X 640 K : 40.000 FB	2 X 640 K : 57.000 FB

Tous ces prix sont hors T.V.A., toutes les unités sont livrées avec des drives double face et avec le DOS V3.1.

I D C I D C I D C I D
D C
C BONNES VACANCES I
I D
D C I D C I D C I D C



DAITEXT V.21

- Traitement de texte
- MACRO-COMMANDES
- Modem Mode PLUS : 300 / 1200 BAUDS
- Compatible DAITEXT V.20
- Compatible KEM-DOS

HALF DUPLEX

CR / LF selectable

Protocole V.23 videotex

FULL DUPLEX

Mode APPEL / REPONSE

Standard V.22

Audio : 1800 FB

OCR : 2050 FB

KEM DOS 80 trk : 2050 FB