

DAICLIC

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

TRIMESTRIEL

Publié par le club:

I.D.C. (International DAI Club)

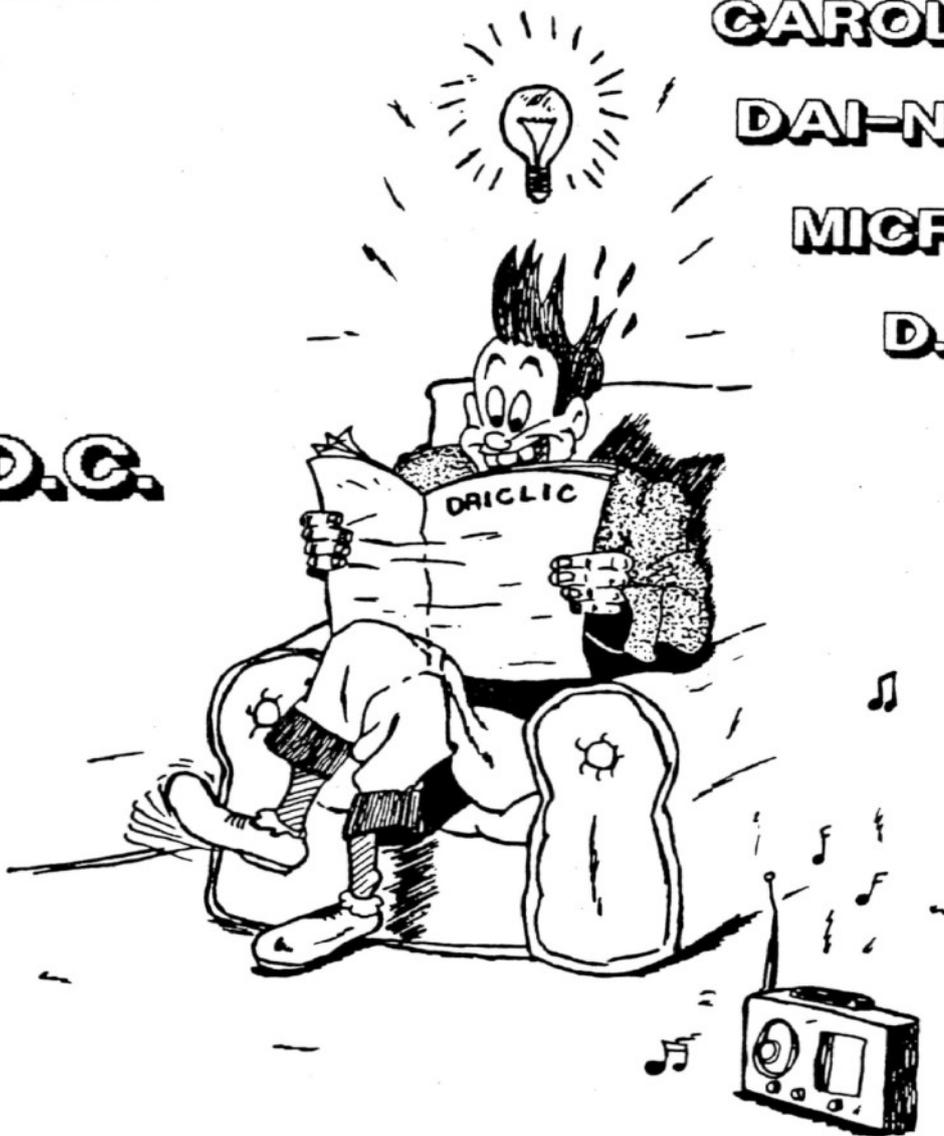
CAROLO DAI

DAI-NAMUR

MICRODAI

D.A.I.C.

I.D.C.



KSW₈₄

NUMERO 1 - FEVRIER 1985

SOMMAIRE

SOMMAIRE DU NUMERO 1 - FEVRIER 1985

<u>ARTICLES:</u>	<u>AUTEURS:</u>	<u>PAGE:</u>
COUVERTURE		0
SOMMAIRE		1
EDITORIAL	IDC	2
INFORMATIONS UTILES		4
IDC SOFTWARE (présentation)	IDC	5
IDC SOFT: DAITEXT		6
IDC SOFT: EDITEUR DE DESSINS		7
IDC SOFT: MAILING LIST		8
IDC SOFT: OTHELLO MICROTEx		8
ASSEMBLEUR: LE CHAIN MANQUANT	Yannick DUPAGNE	9
L'ASSEMBLEUR PAR LA PRATIQUE	Raymond VANLATHem	13
PROGRAMME BASIC EN FICHER MACHINE	Henri-Pierre LEGRY	17
ECRITURE ET LECTURE DE PROGRAMMES MACHINE	Henri-Pierre LEGRY	21
DAILASSEMENT (présentation)		23
DAILASSEMENT		24
AUTOCOLLANTS DAI		25
UTILISATION DE FGT	Louis LAMBRECHT	26
UTILISATION DU MATH PACKAGE ROM 1	Henri-Pierre LEGRY	29
COMPOSEUR AUTOMATIQUE DE NUMEROS DE TELEPHONE	Christian POELS	32
COMMANDES MDCR DANS UN PROGRAMME BASIC	Fabrice DULUINS	33
TESTS LOGICIELS (présentation)		34
TEST: DAITEXT	Marc VANDERMEERSCH	35
TEST: MAILING LIST	Marc VANDERMEERSCH	37
TEST: OTHELLO	Marc VANDERMEERSCH	39
TEST: EDITEUR DE DESSINS	Marc VANDERMEERSCH	40
BASIC: BALONS		41
CAROLO DAI NEWS	CAROLO DAI	42
DAIC NEWS	DAIC	44
DAI-NAMUR INFO	DAI-NAMUR	46
MICRODAI NEWS	MICRODAI	48
PETITES ANNONCES GRATUITES		50
GENERONS DES NOMBRES ALEATOIRES (1)	Raymond VANLATHem	51
BASIC ETENDU (1)	Fernand LEMOINE	54
HARD: INTERFACE CASSETTES AUDIO	Eric NEVE	56
HARD: DOUBLEUR DE VITESSE DE LECTURE	Eric NEVE	60

EDITORIAL

QUELQUES MOTS SUR I.D.C...

I.D.C. (International DAI Club) est une association non lucrative qui a été créée fin '84. Son but est de répondre à un besoin croissant chez les possesseurs de DAI d'avoir une vraie revue, des informations diverses, des logiciels, le tout en français. En plus des articles de la revue, notre club va tenter d'aider ses membres au travers d'un service informations. En effet, bon nombre d'entre vous (pas tous heureusement...) n'ont pas beaucoup de contacts soit avec des dealers, soit avec d'autres possesseurs de DAI de leur région. Si vous avez des problèmes précis concernant la programmation, les achats de matériel, etc..., nous essayerons de répondre à toutes vos questions. Si nous estimons que certains sujets peuvent intéresser d'autres personnes, nous les publierons dans la rubrique 'COURRIER'. Afin d'augmenter l'efficacité de ce service, envoyez-nous avec vos questions, une enveloppe adressée à votre nom pour le retour et un coupon-réponse. Pour les problèmes urgents, vous pouvez téléphoner à la rédaction (entre 17:30 et 18:30).

DAICLIC (édité par I.D.C.) est le magazine de liaison entre les clubs et les utilisateurs du DAI P.C. Il a pu voir le jour grâce à la précieuse collaboration des quatre clubs associés: D.A.I.C. (Bruxelles), CAROLO DAI (Charleroi), DAI-NAMUR (Namur) et MICRODAI (Liège). Pour les prochains numéros, ces clubs vont continuer à nous aider mais nous avons aussi besoin de vous tous! Quelque soit votre spécialité ou votre niveau en programmation, vos articles nous intéressent. Aucun sujet n'est trop simpliste ou trop complexe, il faut de tout pour faire une bonne revue! Nous comptons donc sur vous pour nous envoyer un maximum d'articles et de programmes. Encore une fois, pour nous faciliter la tâche, envoyez si possible vos programmes sur cassettes (DCR ou AUDIO). Pour les articles, vous pouvez nous envoyer de simples manuscrits mais nous préférons des textes imprimés (machine à écrire ou imprimante). Mieux: si vous utilisez un traitement de texte (FWP...) envoyez-nous le fichier sur cassette. Vos cassettes vous seront, bien entendu, renvoyées le plus rapidement possible. Merci pour votre aide!

Si certains d'entre vous sont vraiment intéressés par notre club, nous leur signalons qu'il peuvent participer activement à son extension. Comment? en devenant associé à I.D.C. Nous recherchons par exemple des personnes connaissant le néerlandais et l'allemand pour effectuer des traductions d'articles, de programmes ou de modes d'emploi de programmes. Nous sommes aussi intéressés par la création d'autres clubs régionaux (aussi en France) en plus des quatre actuels. Ce sont deux exemples concrets mais il y a aussi des tas de petites choses bien utiles au fonctionnement d'un club. Si vous êtes intéressé ou si vous avez quelque chose à nous proposer contactez-nous!

LOGICIEL: I.D.C. propose une formule originale aux concepteurs de logiciels: Les programmes commercialisés par I.D.C. rapportent à ceux-ci 90 % des bénéfices, ce qui n'est pas négligeable... Comment faire en pratique?: vous nous envoyez votre programme (sur cassette...) avec une documentation complète,

et nous l'étudions. S'il est accepté, il est automatiquement annoncé dans le prochain DAICLIC et tous les trois mois, vous recevez vos 90 %. C'est simple, non?... Dans tous les cas, votre cassette vous sera retournée le plus rapidement possible. Si le programme n'est pas commercialisé, il sera éventuellement publié dans la revue.

CLUBS ASSOCIES: Un club associé est tout à fait indépendant de I.D.C. du point de vue de la gestion interne et du programme des activités. La revue DAICLIC est un service supplémentaire que propose le club à ses membres. Si un membre est inscrit à I.D.C. et/ou à un ou plusieurs clubs régionaux associés, il ne recevra qu'une seule fois DAICLIC (logique...). Pour chaque abonnement supplémentaire, I.D.C. rembourse 500 FB (pour la Belgique) ou 600 FB (pour l'étranger) moins les frais administratifs éventuels. La cotisation de chaque club comprend l'abonnement à DAICLIC et l'accès aux services spécifiques de ce club.

I.D.C. vous remercie d'être parmi nous et nous attendons votre courrier pour nous faire part de vos premières impressions sur DAICLIC et vos suggestions pour l'améliorer!

I.D.C.

Parmi tous les articles qui nous ont été envoyés pour ce premier numéro, I.D.C. a décidé d'en choisir un et d'offrir un prix à son auteur!

Nous avons le plaisir d'annoncer que M. Henri-Pierre LEGRY (59 DOUAI, FRANCE) a remporté un abonnement gratuit d'un an à DAICLIC! Merci encore à M. LEGRY et à tous ceux qui nous ont aidé à élaborer cette première édition de DAICLIC!

I.D.C.

+++++
+++++ 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 sont 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
DULUINS ET POELS
ALLEE TOUR RENARD 4
1400 NIVELLES

ETRANGER: - mandat postal international uniquement

ADRESSES:

POUR VOUS INSCRIRE OU COMMANDER DES PROGRAMMES:

IDC
C/O Fabrice DULUINS
Allée Tour Renard 4
B-1400 NIVELLES (BELGIQUE)
☎ (0)67 / 21.82.10

POUR ENVOYER VOS ARTICLES ET PROGRAMMES A PUBLIER:

IDC - REDACTION
C/O Christian POELS
Rue des Bas-Sarts 10
B-4100 SERAING (BELGIQUE)
☎ (0)41 / 37.16.06

POUR COMMERCIALISER VOS PROGRAMMES:

IDC - SOFTWARE
C/O Marc VANDERMEERSCH
Avenue du Vert Bocage 17
B-1410 WATERLOO (BELGIQUE)
☎ (0)2 / 354.13.63

CONSEIL D'ADMINISTRATION DE I.D.C. (International DAI Club):

PRESIDENT: Christian POELS, rue des Bas-Sarts 10, B-4100 SERAING
SECRETAIRE: Marc VANDERMEERSCH, avenue du Vert Bocage 17, B-1410 WATERLOO
TRESORIER: Fabrice DULUINS, allée Tour Renard 4, B-1400 NIVELLES

CONTACTS CLUBS ASSOCIES:

CAROLO-DAI: Etienne SZIGETVARI, Route Provinciale 7, B-1361 CLABECQ (B)
DAIC: Jacques MOENS, Clos Fontaine des Ducs 6, B-1310 LA HULPE (B)
MICRODAI: Victor LESINEUX, Rue Latour 39, B-4020 LIEGE (B)
DAI NAMUR: Jean-Pierre MALLIEN, Rue des Nobles 183, B-5761 SOYE (B)

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

EDITEUR RESPONSABLE: Christian POELS, rue des Bas-Sarts 10, B-4100 SERAING

I.D.C. SOFTWARE:

Comme vous avez pu le constater depuis un peu plus d'un an, nous avons commencé à voir apparaître un peu partout, de plus en plus de bons (et même très bons) programmes... (et ce n'est pas fini !).

Attention, quand je dis cela, cela ne veut pas dire que tous les programmes parus avant l'année dernière sont nuls, loin de là !!!, mais vous serez tous d'accord pour dire que la moyenne de qualité des programmes a nettement augmenté depuis un an !!!

Fidèles au fait de vouloir étendre toutes les bonnes choses qui tournent autour du DAI, DAIClic a décidé de faire quelque chose pour les programmeurs ! Cette chose la voici :

Nous considérons que cette expansion des programmes ne doit en aucun cas diminuer mais surtout s'étendre !!!

Voici ce que nous vous proposons comme stimulant :
(par étapes !)

- 1 - vous envoyez votre programme à DAICLIC.
- 2 - nous le jugeons.
- 3 - nous l'acceptons.
- 4 - nous le mettons en vente (publicité dans DAICLIC).
- 5 - pour chaque cassette vendue, l'auteur du programme recevra 90 % du bénéfice net (prix de vente du programme moins les frais).

Qu'en dites-vous ??? Intéressant non ???

Alors, d'un seul élan : tous à vos claviers !
(mais n'oubliez quand même pas de finir de lire ce DAICLIC).

I.D.C.

ADRESSES UTILES:

DIRECTEUR DAI: Georges LEONARD
DIRECTEUR COMMERCIAL: Yvan ART
RESPONSABLE SOFT ET HARD: Martin HALLUT

PRODATA - GROUP
Avenue de Tervuren 412
B-1150 BRUXELLES - BELGIQUE
Tél.: 02/762.28.18
Télex: 62054

Réparations en France:
PRODATA FRANCE
Rue du Maréchal DEVAUX 3
F-91550 PARAY-VIELLE-POSTE

I.D.C. SOFTWARE

DAITEXT

DAITEXT est un programme de traitement de textes professionnel, d'une longueur d'environ 8 K octets, entièrement écrit en langage machine par Alain Mariatte.

Ce puissant système de gestion de textes regroupe toutes les principales qualités et fonctions préprogrammées d'un traitement de textes moderne :

- La 'frappe au kilomètre' : le système ne ralentit absolument pas, même après plusieurs pages de texte.
- Suppression d'une partie du texte.
- Justification à droite.
- Utilisation de polices de caractères de votre imprimante.
- Caractères accentués (même les accents circonflexes !).
- Lecture / Sauvegarde d'un texte (évidemment !).
- Impression d'un texte.
- Liste d'un texte.
- Edition d'un texte (corrections).
- Duplication de paragraphes.
- Grand espace mémoire réservée à l'édition de textes (+/- 36000 caractères en un seul fichier).
- Possibilité de récupération d'un fichier FWP (méthode à suivre livrée avec le programme.)

Il y a plusieurs versions de DAITEXT :

<u>Version :</u>	<u>Imprimante :</u>
- DAITEXT V1	SEIKO GP 80/100
- DAITEXT V2	+ ROM accents
- DAITEXT V3	+ DOS
- DAITEXT V4	+ ROM + DOS
- DAITEXT V5	SEIKO GP 250
- DAITEXT V6	+ ROM
- DAITEXT V7	+ DOS
- DAITEXT V8	+ ROM + DOS
- DAITEXT 20	EPSON (+ imprimantes standard accentuées)

DAITEXT 20 est différent des autres DAITEXT :

- Il est entièrement compatible KEN DOS.
- Il possède un puissant système TELEMATIQUE intégré. Ce système permet le transfert de textes, de programmes BASIC et de langage machine (sous forme de codes ASCII). Le système peut bien évidemment recevoir ou transmettre des informations.

De plus, chaque version de DAITEXT est livrée avec un programme de conversion DAITEXT SEIKO / DAITEXT EPSON.

Chaque système DAITEXT est entièrement compatible DCR et CASSETTE audio. (DAITEXT 20 est compatible KEN DOS).

Certains systèmes sont compatibles DOS INdata.

N'hésitez pas et n'attendez pas, commandez dès maintenant votre DAITEXT.

Prix :

DAITEXT V1 à V8 + traducteur : 1500 FB (audio)
DAITEXT 20 + traducteur : 1800 FB (audio)
(+ 150 FB pour la cassette digitale)

N'oubliez pas de préciser la version de DAITEXT correspondant à votre imprimante !!!

I.D.C. SOFTWARE

EDITEUR DE DESSINS

EDITEUR DE DESSINS est un programme qui vous aidera à faire tous vos dessins.

Ce programme tourne en mode 6 et possède des caractéristiques très alléchantes.

Ses principaux atouts sont :

- une table des symboles complètement redéfinissable.
- la possibilité d'enregistrer votre dessin sous forme de table d'ordre, c'est-à-dire que le programme maintient en RAM toutes les fonctions (ainsi que leurs paramètres) que vous aurez faites, donc, en prenant une toute petite place sur la cassette.
- un éditeur de dessins en mode 6 géré à l'aide d'un puissant système de 'sprites' (lutins).
- un éditeur qui vous permettra de supprimer ou de modifier la table d'ordre (donc, plus de 'gaffes').

Il possède de nombreuses fonctions graphiques :

- PAINT rapide avec mélange de teintes
- traçage de RECTANGLE pleins ou vides.
- CERCLE.
- RAYON.
- DROITE.
- Choix de couleurs.
- ...

...sans oublier une routine d'écriture (caractères redéfinissables) très puissante. Votre texte est transformé en 'sprite' et vous pouvez le déplacer très facilement à l'écran, le reproduire autant de fois que vous le désirerez...

Ce programme nécessite un PADDLE 3-DIM et s'emploie à l'aide des petits signes placés à l'écran (icônes).

Le principe de dessin est similaire à celui du célèbre MAC PAINT qui tourne sur ordinateur Apple.

Prix : 1300 FB (Belgique et étranger)
(+ 150 FB sur cassette digitale)
(livré avec mode d'emploi complet !)

IDC SOFTWARE

MAILING LIST

MAILING LIST est un programme de gestion de fichiers d'adresses écrit par Christian Poels et commercialisé par IDC software (vous vous en doutiez !).

MAILING LIST est capable de gérer un fichier de 256 fiches, avec les caractéristiques suivantes :

- | | |
|----------------|--------------|
| - Nom. | - Ville. |
| - Prénom. | - Pays. |
| - Adresse. | - Téléphone. |
| - Code postal. | - Remarque. |

MAILING LIST possède des routines de Saisie, Consultation, Annulation, Modification, Appel téléphonique, Tri, Commandes DCR, Impression format étiquettes, Impression format Mailing.

Mailing possède une routine spéciale de sauvegarde sur cassette de ses fichiers. Grâce à cette routine, la sauvegarde se fait beaucoup plus rapidement que sur tout autre programme de gestion d'adresses.

Il reste cependant possible de sauver et de lire les fichiers d'après la méthode de sauvegarde SAVEA/LOADA, nettement moins rapide, mais cela peut permettre de charger des fichiers venant d'un autre programme de gestion d'adresses.

MAILING LIST, Prix : - audio : 1500 fb
- dcr : 1650 fb

OTHELLO MICROTEx

OTHELLO MICROTEx est le premier programme de jeu qui vous est présenté par IDC software.

Il s'agit d'un Othello haut de game qui allie beauté de présentation et finesse de jeu.

Ce programme, écrit par Christian Poels, a été classé quatrième au tournoi d'Othello organisé par la revue "L'Ordinateur Individuel" lors du dernier salon de micro-informatique 'MICROTEx', il y a juste un an, à Bruxelles.

Ce superbe programme, qui a demandé des heures de mise au point est à votre disposition pour 800 fb (+ 150 FB pour une version DCR) chez IDC software.

Plus de renseignements au sujet de ce programme dans le test paru à son sujet dans ce même DAIClic.

ASSEMBLEUR

Le CHAIN manquant...

S'il est une instruction qui fait cruellement défaut dans le Basic du DAI, ce pourrait bien être CHAIN "bidule" qui permettrait, depuis un programme en cours d'exécution, d'appeler un autre programme "bidule" en lui passant les valeurs de toutes les variables - numériques, chaînes, indicées ou non - communes à ces deux programmes. Il existe bien le LOAD "bidule" qui permet de chaîner le programme "bidule" à celui en cours d'exécution, mais il efface toutes les variables...

La routine que nous vous présentons comble cette lacune. Appelons-la CHAIN% - logée en l'adresse CHAIN% - et nous l'utiliserons en plaçant en une chaîne telle A\$ le nom du programme à chaîner:

```
A$="bidule":CALLM CHAIN%,A$
```

Lorsqu'il rencontre la commande LOAD "bidule", l'ordinateur exécute la routine située entre D270 et D2A7. Il commence par effacer le HEAP, effacer le programme Basic - en modifiant les pointeurs par un NEW - et la table des symboles. Ensuite, il charge le programme "bidule" dont la table des symboles avait déjà été "vidée" lors du SAVE "bidule" qui l'avait mis de côté.

Nous proposerons la démarche suivante. Lorsqu'il rencontrera notre nouvel ordre CHAIN, le DAI déplacera l'ancienne table des symboles en haut de RAM - juste avant la vidéo RAM, dont l'adresse se trouve en 2A5 - et la protégera en modifiant cette adresse de haut de RAM afin qu'elle pointe le premier octet de notre ancienne table des symboles. Ensuite, nous chargerons le nouveau programme Basic spécifié en utilisant la routine normale - excepté la vidange du HEAP -. Le programme chargé, il ne restera plus qu'à parcourir l'ancienne table des symboles. Pour chaque nom de variable rencontré, il faudra parcourir la nouvelle table des symboles. Si l'ancien nom est présent dans la nouvelle table des symboles, il faudra en transférer la valeur s'il s'agit d'une variable numérique ou le pointeur s'il s'agit d'une chaîne ou d'un tableau. Si l'ancien nom n'est pas repris dans la nouvelle table des symboles et qu'il s'agit d'une chaîne ou d'un tableau, il faudra récupérer la place occupée en HEAP par cette donnée en effaçant cette chaîne ou ce tableau.

Voici un résumé de la démarche:

début

déplacer la table des symboles en haut de RAM
modifier le pointeur de haut de RAM (2A5)
charger le nouveau programme Basic

parcourir l'ancienne table des symboles

pour chaque NOM rencontré
si ce NOM est dans la nouvelle table des symboles

alors

en transférer la valeur ou
le pointeur

sinon

s'il s'agit d'une variable indicée ou d'une
chaîne
alors l'effacer du HEAP

réadapter le pointeur en 2A5 et effacer ainsi l'ancienne
table des symboles
lancer l'exécution du programme sans effacer les variables

fin

Voici la liste en codes 8080. Placez-la en bas de RAM (300 ou ...) à l'adresse
CHAIN%

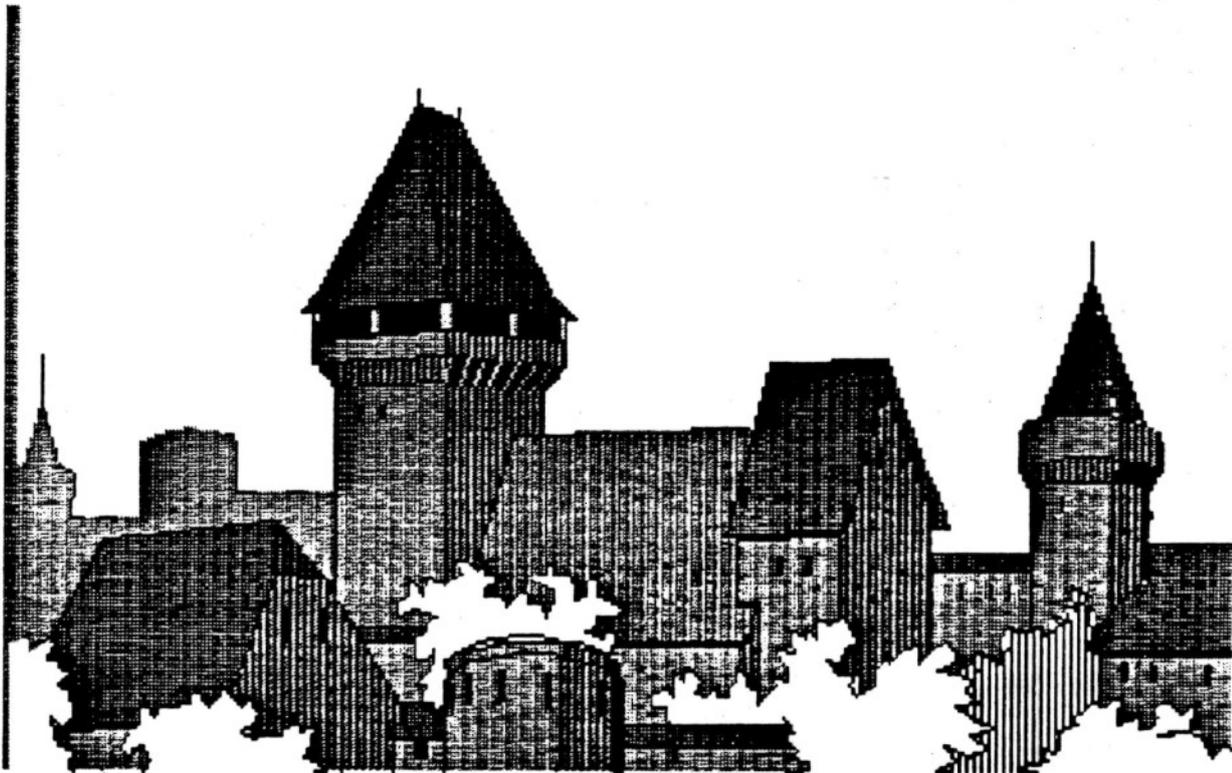
CHAIN	F5	PUSH PSW	sauvegarde des
	C5	PUSH B	registres du 8080
	D5	PUSH D	
	E5	PUSH H	
	2AA502	LHLD : 2A5	pointeur haut de RAM
	22....	SHLD : endf	le mettre de côté
	EB	XCHG	fin de RAM en DE
	2AA102	LHLD : 2A1	table des symboles
	E5	PUSH H	
	C1	POP B	en BC
	2AA302	LHLD : 2A3	fin de la table
	23	INX H	
	13	INX D	
suite	2B	DCX H	déplacement de la
	1B	DCX D	table des symboles
	7E	MOV A,M	en bas de RAM

	12	STAX D	
	7D	MOV A,L	
	91	SUB C	
	C2....	JNZ suite	
	7C	MOV A,H	
	98	SBB B	
	C2....	JNZ suite	
	EB	XCHG	HL = nouvelle fin de RAM
	22A502	SHLD : 2A5	modifier le pointeur
	E1	POP H	HL = VARPTR (nom du programme)
	5E	MOV E,M	
	23	INX H	
	56	MOV D,M	
	EB	XCHG	HL pointe vers le nom
	CD7ADE	CALL ODE7A	LOAD sans CLEAR
	2A3401	LHLD : 134	
	E5	PUSH H	
	211001	LXI H : 110	input from string
	223401	SHLD : 134	
	2AA502	LHLD : 2A5	ancienne table des symboles
	EB	XCHG	en DE
	2A....	LHLD : endf	ancienne fin de RAM
	22A502	SHLD : 2A5	
	EB	XCHG	en DE fin de RAM
suite	7E	MOV A,M	en HL ancienne table
	57	MOV D,A	T/L d'une variable
	E60F	ANI : OF	uniquement la longueur du NOM
	CA....	JZ fin	
	0601	MVI B : 1	sauter T/L
	223201	SHLD : 132	chercher ce NOM
	E5	PUSH H	en parcourant la
	CD57CA	CALL OCA57	nouvelle table des symboles
	DA....	JC trouvé	ce NOM est repris
	E1	POP H	ce NOM n'est pas repris
	CDB1CA	CALL OCAB1	quel type de variable ?
	7E	MOV A,M	T/L de cette variable
	E6FO	ANI : FO	uniquement le type
	FE40	CPI : 40	
	D2....	JNC array	si variable indiquée
	23	INX H	
	FE20	CPI : 20	
	CA....	JZ string	si chaîne de caractères
	23	INX H	sinon type numérique
	23	INX H	sauter les quatre
	23	INX H	octets qui en contiennent
	23	INX H	la valeur
	C3....	JMP suite	
string	CDA8CB	CALL OCBA8	effacer cette chaîne
	C3....	JMP suite	
array	CD5BCB	CALL OCB5B	effacer ce tableau
	C3....	JMP suite	
trouvé	EB	XCHG	DE pointe T/L info de
	E1	POP H	la nouvelle variable
	CDB1CA	CALL OCAB1	

	7E	MOV A,M	T/L info ancienne variable
	E60F	ANI : OF	
	4F	MOV C,A	la longueur en C
inc	13	INX D	déplacer l'information
	23	INX H	sur l'ancienne variable
	7E	MOV A,M	vers la nouvelle
	12	STAX D	
	0D	DCR C	
	C2....	JNZ inc	
	23	INX H	
	C3....	JMP suite	
fin	E1	POP H	remise du DAI dans
	223401	SHLD : 134	son état initial
	C3A1DF	JMP ODFA1	RUN sans CLEAR
endf	deux octets libres

N.B. Cette routine tourne sous BASIC V1.1.
T/L pour TYPE/LENGTH d'une variable ou de son nom.

DUPAGNE Yannick



ASSEMBLEUR

* L'ASSEMBLEUR par la PRATIQUE < 1 > *

par Raymond VANLATHM (Club CARLODAI)

Preambule

Il n'entre pas dans nos intentions de faire ici un expose theorique sur le Langage Assembleur, car il existe suffisamment d'ouvrages tres valables dans le commerce et pas mal de revues qui abordent le sujet avec plus ou moins de precision. Nous souhaitons proposer a nos lecteurs des applications pratiques, mais simples au depart, afin qu'ils puissent s'exercer sur leur DAI, mieux comprendre la portee des differentes instructions et leurs implications. Nous esperons contribuer ainsi a la reflexion, source d'enrichissement des cellules grises.

Si l'utilite de certaines routines, surtout au debut, n'est pas transparente, dites vous bien qu'il s'agit essentiellement de routines didactiques et qu'elles s'enchainent progressivement avant de constituer reellement des routines operationnelles. Cependant un esprit subtil et eclaire aura tot fait d'adapter les routines proposees et de les integrer eventuellement dans ces programmes.

Introduction

Un petit rappel n'est pas superflu cependant.

Un microprocesseur (dans le cas du DAI le 8080 A) ne peut executer que des instructions en code binaire (represente par l'etre humain sous forme de code hexadecimal) que l'on appelle "code machine". Comme il serait assez complique et assez long de programmer en Langage Machine (surtout pour des programmes conséquents), des langages de programmation ont ete developpes. Un de ces langages est le Langage Assembleur, dont il existe differents developpements (DNA, SPL, ...). Cependant dans chacun de ceux-ci chaque instruction de base est representee par un meme mnemonique (mot condense qui par association mentale rappelle la nature de l'instruction). Chaque instruction en Assembleur est convertie en un code machine par le programme d'assemblage.

Le jeu d'instructions du 8080 A comprend en Langage Machine 244 codes sur 256 combinaisons de codes hexadecimaux possibles. Chacun de ces codes constitue le code operatoire d'une instruction machine. Il y a autant d'instructions machine que de codes operatoires differents. Chaque instruction du 8080 A comporte un, deux ou trois octets. Le premier octet constitue toujours le code operatoire de l'instruction. Le ou les octet(s) suivant(s) eventuel(s) contient(en)t une donnee ou une adresse memoire (a 8 ou 16 bits). Sur les 244 instructions machine, il y en a 200 a un octet, 18 a deux octets et 26 a trois octets. D'autre part elles correspondent a 78 instructions en Assembleur, qui se composent d'un mnemonique different pour chaque instruction et eventuellement d'une partie operande a un ou deux elements.

Il est bon de se rappeler egalement que le 8080 A possede 6 registres generaux d'un octet B, C, D, E, H et L ; ceux-ci peuvent etre groupes par paires (B+C, D+E et H+L), ou le premier terme (B, D ou H) designe la paire et contient l'octet de poids fort, le second registre contenant l'octet de poids faible. Sont assimiles a des registres, l'accumulateur A (un octet), par ou passent les operations arithmetiques et logiques, ainsi que la cellule memoire M (deux octets), dont l'adresse se trouve dans la paire de registres H+L. Nous avons egalement un pointeur de pile SP (Stack Pointer) et un compteur ordinal de

deroulement du programme PC (Program Counter), chacun a deux octets.

Enfin il y a encore un registre special a un octet dont 5 bits seulement sont utilises: ce sont les indicateurs d'etat ou de condition (Flags), qui sont du type a bascule (flip-flop) et qui s'intitulent dans l'ordre decroissant des bits utiles: le signe S, le zero Z, le report auxiliaire AC (Auxiliary Carry), la parite P et le report C (Carry). Les 3 bits inutilises de l'octet sont le 6eme et le 4eme (toujours a l'etat 0), ainsi que le 2eme (toujours a l'etat 1). Notons que l'accumulateur A et le registre d'indicateurs peuvent former une paire appelee PSW (Program Status Word), dans laquelle l'accumulateur occupe l'octet au poids fort.

Application No 1

BUT: Vider des memoires a partir d'une adresse memoire de depart.

L I S T I N G D I D A C T I Q U E

300	97	SUB A	Vide l'accumulateur, en retranchant le contenu de A de celui de l'accumulateur.
301	00	NOP	N'a aucun effet; permet de reserver une cellule memoire pour intercaler une instruction sans devoir modifier l'implantation generale de la routine.
302	21 0B 03	LXI H, :30B	Charge l'adresse memoire de depart en H+L (#0B en L et #03 en H).
305	77	MOV M,A	Transfere le contenu de A en M (dont l'adresse se trouve en H+L).
306	23	INX H	Incremente le contenu de H+L d'une unite (donne l'adresse memoire suivante).
307	C3 05 03	JMP :305	Saut inconditionnel a l'adresse memoire #305 pour effectuer le transfert suivant du contenu de A en M.
30A	C9	RET	Permet de sortir de la routine; semblable a l'instruction RETURN du Basic.

C O M M E N T A I R E S

L'execution de la routine se fait en UTILITY (tapez UT en Basic), puis >G300 (precede ou non de >Z3). Les memoires se vident a partir de #30B (l'ecran se vide egalement), mais le DAI se plante: c'est normal, car rien ne met fin a l'execution de la routine. Faites RESET, puis *UT, et affichez le contenu des memoires de #30A a #B34F (page par page) au moyen de DISPLAY, pour verifier la mise a zero de celles-ci. Remarquons que les cellules memoires #3EA et #3EB contiennent respectivement #7F et #FF, suite au RESET.

Vous pouvez egalement faire >Z3 (indispensable ici) suivi de >L300 300 30A, ce qui vous permettra de visualiser les etats successifs des differents registres a l'issue de l'execution de chaque instruction. Dans la cas present le temps d'execution est fort eleve, mais vous pouvez interrompre a tout moment le deroulement du programme en tapant BREAK, et verifier ensuite l'etat d'avancement de la mise a zero des cellules memoires par DISPLAY.

Notons aussi qu'il ne faut pas abuser de l'instruction NOP (00), car celle-ci, bien qu'inoperante, prend tout de meme 4 cycles d'horloge.

E X P E R I E N C E S

(1) En 300 vous pouvez remplacer SUB A (97) par XRA (AF) qui donne le meme resultat: OU exclusif entre le contenu de A et celui de l'accumulateur.

(2) Vous pouvez aussi remplacez en 300 & 301 SUB A et NOP (00) par MVI A, :0 (3E 00), ou la donnee 0 (un octet) est chargee dans l'accumulateur. Bien sur le

00 (code machine de l'instruction NOP) peut servir de donnée pour l'instruction MVI A (3E). Bien que cette dernière solution soit plus évidente que les deux autres, elle a l'inconvénient d'occuper deux octets de mémoire au lieu d'un et de demander 7 cycles d'horloge contre 4 pour les deux autres solutions.

N.B.: L'état des indicateurs diffère selon le type de mise à zéro de A. Ainsi avec SUB A nous obtenons F(Flag)= #56, avec XRA A nous avons F= #46 et MVI A,0 nous donne F= #02.

Application No 2

BUT: Charger des valeurs dans deux registres et visualiser leur contenu.

L I S T I N G D I D A C T I Q U E

300	21 00 13	LXI H, :1300	Charge l'adresse de visualisation en H+L.
303	01 01 0A	LXI B, :A01	Charge la donnée #A01 (deux octets) en B+C (#01 en C et #0A en B).
306	70	MOV M, B	Transfère le contenu de B en M (ici à l'adresse #1300).
307	23	INX H	Incmente le contenu de H+L d'une unité.
308	71	MOV M, C	Transfère le contenu de C en M (ici à l'adresse #1301).
309	00	NOP	Instruction nulle.
30A	C9	RET	Sortie de routine.

C O M M E N T A I R E S

En UT faites >G300, puis affichez par >D1300 1301, qui vous donne 0A 01. Ou bien faites >Z3 + >L300 300 30A, et ensuite >D1300 1301 pour visualiser leur contenu.

E X P E R I E N C E S

(1) En remplaçant en 30A l'instruction RET (C9) par HLT (76), qui arrête le déroulement du programme, le DAI se plantera. En principe, n'utilisez jamais HLT, sinon avec beaucoup de prudence.

(2) En 309, si vous remplacez NOP par PUSH B (C5), vous chargez le contenu de B+C sur la pile initiale du DAI, c'est-à-dire que vous sauvez cette donnée sur la pile. Avec >G300 vous n'avez pas de problèmes, et >D1300 1301 vous affichera 0A 01.

Mais si vous faites >Z3 + >L300 300 30A, le DAI se plantera après l'exécution du LOOK. En faisant BREAK (à temps, mais pas trop tôt) vous pouvez constater que la valeur de P (adresse de la prochaine instruction à exécuter), qui se trouve sur la ligne adresse de la dernière instruction exécutée (I= #30A), est

notre donnée #0A01. Or on aurait dû y voir figurer la valeur #EA04, qui donne l'adresse de retour de la ROM et qui se trouve maintenant en bas de la pile (#FBFE & #FBFF), dont l'adresse est fournie par S= #FBFE de la ligne adresse déjà citée.

Application No 3

BUT: Charger sur la pile le contenu de deux registres pour examen.

L I S T I N G D I D A C T I Q U E

300	31 02 13	LXI SP, :1302	Charge l'adresse de départ de la pile et modifie donc le pointeur de pile. Utilise une pile dans la RAM à la place de la pile initiale du
-----	----------	---------------	---

			DAI. Peut etre utile dans certaines applications specifiques.
303	00 01 0A	LXI B,:A01	Charge la donnee #A01 en B+C.
306	00	NOP) Instructions inoperantes servant de reserve) en vue d'insertions aisees.
307	00	NOP	
308	00	NOP	
309	C5	PUSH B	Sauve le contenu de B+C sur la pile.
30A	00	NOP	
30B	C9	RET	Permet la sortie de la routine.

COMMENTAIRES

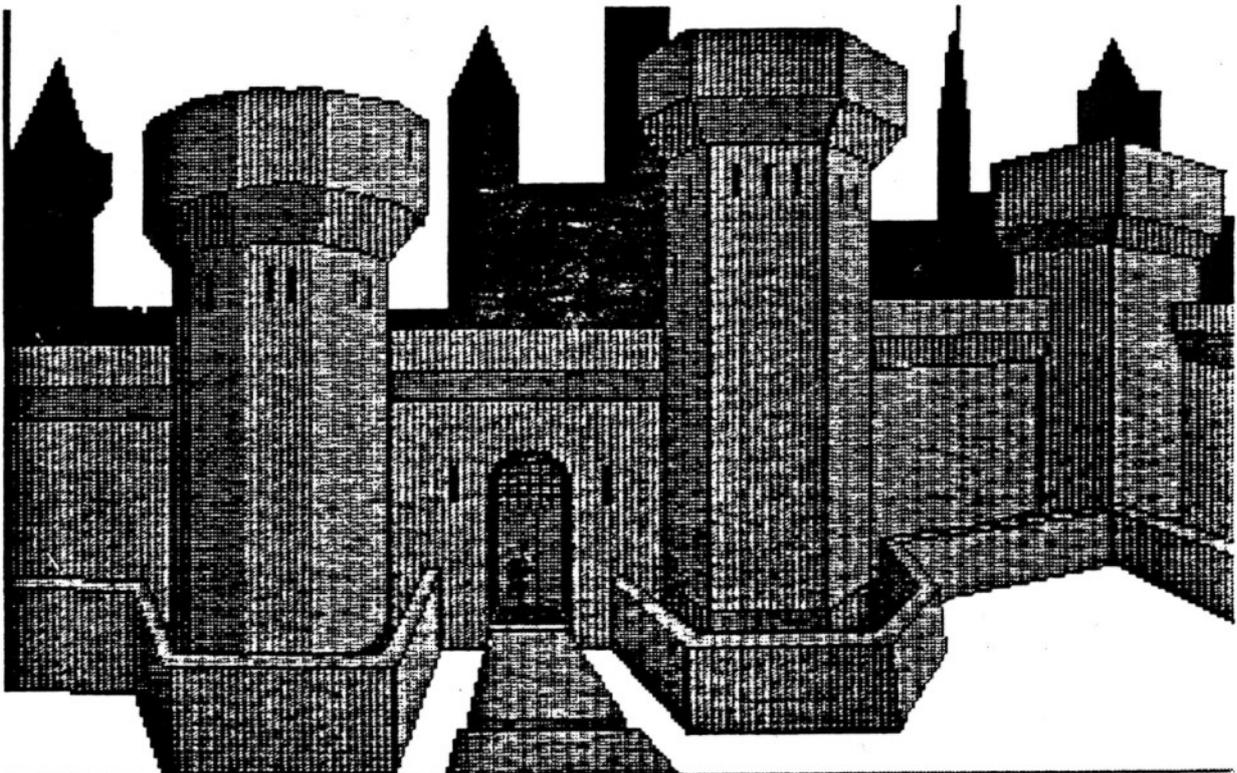
Sur notre lancee, prolongeons l'experience (2) de l'application precedente et voyons ce qui se passe avec une autre pile. En faisant >Z3 + >L300 300 30B nous avons P= #0A01, et le DAI se plante en l'absence de BREAK, comme avec la pile initiale.

EXPERIENCES

(1) En remplaçant les trois NOP de 306 a 308 par LXI H,:EA04 (21 04 EA) }charge l'adresse #EA04 en H+L(et le NOP de 30A par PUSH H (E5) }charge sur notre pile le contenu de H+L(, le DAI ne se plante plus, meme avec LOOK. En affichant par >D1300 1301, nous obtenons 01 0A (presentation inverse de l'application No 2). Nous remarquons ici l'importance capitale de l'adresse #EA04, qui doit figurer en bonne place sur la pile.

(2) Faisons >Z3 + >L303 303 30B }en sautant l'instruction LXI SP,:1302 en 300, nous reprenons la pile initiale(. Pour afficher la donnee il faut faire >DFBFC F8FD, ou vous trouverez 01 0A (l'adresse #FBFC est donnee par S a la derniere ligne du LOOK, soit celle concernant I= #30B).

* * * * *



BASIC & LM

SAUVEGARDE D'UN PROGRAMME BASIC SOUS SA FORME LANGAGE MACHINE

LE 'SAVE', COMMANDE BASIC

HEAP	I-----I	stockée en 29B/29C
	I	
	I VARIABLES (HEAP) I	
	I	
TXT BGN	I-----I	stockée en 29F-2A0
	I	
	I	
	I	
	I TEXTE BUFFER I	
	I	
	I	
	I	
TXT USE	I-----I	stockée en 2A1-2A2
STB BGN	I	
	I SYMBOL TABLE I	
	I	
STB USE	I-----I	stockée en 2A3-2A4

Les variables dont le programme aura besoin, sont stockées dans la partie réservée dont l'adresse est stockée en 29B/29C et la taille en 29D/29E.

Un 'SAVE' BASIC:

1. initialise le HEAP et la SYMBOL TABLE
2. sauvegarde le TEXTBUFFER
3. sauvegarde la SYMBOL TABLE

Un 'LOAD' BASIC:

1. initialise le HEAP et la SYMBOL TABLE
2. charge le TEXTBUFFER et stocke les adresses correspondantes en 29F/2A0 et 2A1/2A2
3. charge la SYMBOL TABLE et stocke l'adresse correspondante en 2A3/2A4

Par défaut, le HEAP pointeur (29B/29C) est initialisé à 2EC et la taille est de 100 (TEXTBUFFER: 2EC+100=3EC)

Donc, il faut sauvegarder, si le programme BASIC est seul, le TEXTBUFFER, la table des symboles et stocker leur adresses en 29F-2A4.

Il est donc plus commode de sauvegarder le programme de 29F à STB USE.

```
*LOAD          chargement du fichier
*UT
>D2A3 2A4
>AA BB
>W29F BBAA     nom du programme
```

Le programme BASIC est sauvegardé sous sa forme LM.

SAUVEGARDE D'UN PROGRAMME BASIC + UN PROGRAMME LM SOUS UNE FORME LM

Il devient intéressant de réunir un programme BASIC, dont on sait sauvegarder sous forme LM, et un programme LM pour charger l'ensemble sous forme LM. (exemple FGT (300-900) + programme BASIC)

La structure de la RAM devient:

2EC	I-----I	-	
	I	I	
	I PROGRAMME LM	I	> PROGRAMME LM
	I	I	
HEAP	I-----I	-	
	I VARIABLE HEAP	I	--> VARIABLES
TXT BGN	I-----I	-	
	I	I	
	I	I	
	I TEXT BUFFER	I	
	I	I	
TXT USE	I	I	> PROGRAMME BASIC
STB BGN	I-----I	I	
	I	I	
	I SYMBOL TABLE	I	
	I	I	
STB USE	I-----I	-	

Il faut, cette fois, conserver les variables 29B à 2A4 car elles indiquent les adresses où sont stockées les différentes parties. Le programme BASIC n'est plus stocké entre 2EC et STB USE mais entre 2EC+taille du programme LM+taille du HEAP et STB USE.

2EC étant la valeur initiale (après RESET) de la RAM.

** Il faut charger le MLP et le BASIC en ajustant les pointeurs 29F-2A4.

1ère méthode: utiliser un BOOTSTRAP LOADER (DBL).

Il suffit d'appuyer sur la touche BREAK lorsque le chargement du programme BASIC est commencé.

2ème méthode: rectifier les pointeurs.

On va prendre comme exemple FGT (300-900).

(remarquez que FGT est distribué de 29B à 900, ce qui rectifie les pointeurs 29F à 2A4 lors du chargement)

donc:

```

*UT
>R                                chargement du MLP

>S29B
>EC-01
>02-09                            début du HEAP -> après le programme LM (soit 901)
>00-
>01-
>EC-01
>03-0A                            début du TEXTBUFFER soit début du HEAP + 100
                                   =901+100=A01
>B
*LOAD                             chargement du BASIC

```

** Vous avez stocké, dans la RAM, le MLP+BASIC sous la forme décrite plus haut.

Il faut maintenant sauvegarder le programme de 29B à STB USE.

STB USE est contenu en 2A3/2A4

donc:

```

*UT
>D2A3 2A4
>XX YY
>B
*                                (préparez la cassette)

*UT

```

```
>W29B YYXX
>B
*
```

Vous disposez d'un programme BASIC+LM, sauvegardé sous forme LM.

```
Pour ressortir:  #UT
                  >R
                  >B
                  #RUN
```

AUTOSTART D'UN PROGRAMME BASIC ET LM

Lors d'un READ ou d'un LOAD, lorsque la commande est complètement terminée, un CALL 2D4 est exécuté pour fermer le fichier et stopper le moteur. 2D4 ne sert qu'à renvoyer vers D445 (cassettes audio) ou vers F31F (DCR). (JMP D445 ou JMP F31F)

AUTOSTART EN LANGAGE MACHINE

Il faudra stocker votre langage machine à partir de 2D4. Si les adresses de votre MLP vont, par exemple, de 5000 à 6000, vous devez sauvegarder la RAM de 2D4 à 8000, ce qui n'est pas très intéressant puisque de 2EC à 5000, la RAM ne vous sert pas!

Il faut donc un fichier dont l'adresse de début se situe aux alentours de 300. Pour obtenir l'autostart, le début de votre MLP doit fermer le fichier, c'est à dire, exécuter un RCLOSE (CD 45 D5 pour les cassettes audio et CD 1F F3 pour les DCR).

EXEMPLE: MLP de 300 à 1000
adresse de lancement: 300

On remarque que l'espace 2EC à 300 est libre. S'il ne l'était pas, il faudrait placer le petit programme suivant, de 1001 à 1006. Chargez le programme puis:

donc, on place en 2EC:

2EC	CALL	D445 (ou F31F)	CD45D4	RCLOSE
	JMP	300	C30003	lancement en 300

puis rectifier 2D5:

>S2D4	ou:
>45-EC	>1F-EC
>D4-02	>F3-02

Maintenant, sauvegardez de #2D5 à #1000

```
>W2D5 1000 nom-du-programme
>B
*
```

Remplacez la cassette, puis UT et R.

(Si votre programme LM se trouve, par exemple, de #5000 à #6000, vous le transférez en #300 - #1300. Votre petit programme, en #2EC, par exemple, devra faire un MODE avant de le lancer)

AUTOSTART D'UN PROGRAMME BASIC

La méthode reste la même mais il faut, au lieu de lancer le programme par un CD, le lancer par un RUN.

Je crée un exemple.

Tapez le programme BASIC suivant:

```
10 MODE 0
20 PRINT CHR$(12)
30 COLORT 0 3 0 0
40 PRINT "CA FONCTIONNE"
```

Chargez le sur cassette (ou DCR) et repositionnez la sur ce programme. Eteignez le DAI (pour remettre toutes les variables à leur état initial). On va placer notre petit programme LM de #2EC à #300.

```
>S2EC 3E 00 CD 06 D8 31 00 F9 01 FD 02 CD 1F F3 C3
    92 CB 87
```

I

I

V

pour cassettes audio: 45 D4

Ajoutez les pointeurs en #29B:

```
>S29B EC-00 02-03 00-00 01-01 EC-00 03-04
>B
#NEW
#LOAD
>UT
>D2A3 2A4
>43 04
>S2D5 1F-EC F3-02
(CAS:      45-EC D4-02)
>W29B 443 nom
```

Eteignez le DAI puis rechargez le programme.

Maintenant, si vous avez un MLP + programme BASIC, il faut réunir les deux puis les sauvegarder sous forme LM. Ajoutez le programme LM si vous disposez de la place entre #2EC et #300; sinon, placez ce programme après votre fichier LM (réunion LM+BASIC). Il faut alors corriger l'adresse que l'on place en #2D4, elle n'est plus de #2EC mais elle correspond à l'adresse où vous avez stocké votre petit programme LM.

D'autre part, dans le programme, il y a une adresse (soulignée) FD 02 soit #2FD. Elle correspond à #2EC+#11 (elle pointe sur le #87)

Ne pas oublier de rectifier.

Henri-Pierre LEGRY
F-59 DOUAI.

ASSEMBLEUR

ECRITURE & LECTURE

DES PROGRAMMES EN LANGAGE MACHINE

Pour lire ou écrire un fichier, il faut: ouvrir le fichier, le lire ou l'écrire et fermer le fichier.

Un CALL aux adresses suivantes correspond aux commandes:

#2C5	WOPEN	ouvre le fichier en écriture
#2C8	WBLK	écrit un bloc (données)
#2CB	WCLOSE	ferme le fichier en écriture
#2CE	ROPEN	ouvre le fichier en lecture
#2D1	RBLK	lit un bloc (données)
#2D4	RCLOSE	ferme le fichier en lecture

Il faut auparavant, stocker dans les registres, les valeurs attendues par l'un des CALL effectué à ces adresses.

EXEMPLE: ECRITURE sur CASSETTE ou DCR d'un programme en langage machine, #1000 à #2000, ayant pour nom 'FICHIER'.

```
désactive les interruptions  DI
commute la banque 3         MVI A    0C0H
                             CALL    0D806H
```

type de programme 1 => #31

#31 = HEX\$(ASC("1"))

HL pointe sur le NOM

```
MVI A    31H
LXI H    400H
```

ouverture du fichier

```
CALL    2C5H
```

HL pointe sur adresse de départ

```
LXI H    1000H
```

D longueur du programme

```
LXI D    1000H
```

écriture du fichier

```
CALL    2C8H
```

fermeture du fichier

```
CALL    2CBH
```

en #400:

```
#46    'F'
#49    'I'
#43    'C'
#48    'H'
#49    'I'
#45    'E'
#52    'R'
```

Il est à remarquer, pour les curieux, que l'on peut mettre n'importe quel caractère dans A (avant le CALL 2C5) pour le type de programme. Essayer puis, faites CHECK.

LECTURE

On désire maintenant lire le fichier que l'on vient de stocker.

programme de #1000 à #2000

longueur #2000-#1000=#1000

nom FICHIER

type 1 = CHR*(#31)

désactive les interruptions	DI	
commute la banque 3	MVI A	0C0H
	CALL	0DB06H
type de programme attendu #31	MVI A	31H
FF pour afficher le titre		
00 pas d'affichage	MVI B	0FFH
ouverture du fichier	CALL	02CEH
HL pointe sur l'adresse où on va stocker le fichier	LXI H	1000H
D pointe sur l'adresse indiquant la fin de la RAM disponible, l'ordinateur calcule l'espace disponible et vérifie qu'il est supérieur à la longueur du fichier	LXI D	0F900H
stockage (lecture)	CALL	2D1H
fermeture du fichier	CALL	2D4H

L'ouverture du fichier provoque la mise en route du moteur (K7 ou DCR) et la fermeture du fichier stoppe le moteur.

Lors l'un LOAD ou d'un READ, l'ordinateur termine toujours ces commandes par un CALL 2D4 (RCLOSE)

AUTOSTART

Pour exécuter un programme en autostart, il suffit de placer le pointeur sur l'adresse de départ du programme

ordre naturel (READ) en K7:

```

-----> 2CE          I--> 2D1          I--> 2D4          I--> retour au moniteur
      I          I          I          I          I
      I          I          I          I          I
      I--> D325 --I    I--> D340 --I    I--> D445 --I
              (ret)          (ret)          (ret)
  
```

ordre à respecter (programme à lancer en 1000 par exemple):

```

-----> 2CE          I--> 2D1          I--> 2D4          I-->
      I          I          I          I          I
      I          I          I          I          I
      I-...-I      I-...-I          I          I--> D445 --I
                          V          I
                          1000--I
  
```

Donc, pacer en 2D5-2D6 l'adresse de départ du programme (1000) et commencer le programme par CALL D445.

>S2D4 45-00 D4-10

1000 CD D4 45

...

Puis sauvegarder de 2D5 à 2000. Il vaut mieux bien sur avoir le programme de 300 à 1300.

Dernier détail: pour neutraliser un programme autostart, il faut que, en chargeant celui-ci, on ne modifie pas l'adresse contenue dans 2D5-2D6 soit D445. Pour cela, charger le fichier entre 1000 et 3000 (pour l'exemple précédent) avec un >R1000, puis faire un MOVE: >M1000 3000 0. Vous trouverez l'adresse de départ du programme en 2D5-2D6. L'exemple est ici mal choisi pour un autostart, la mémoire de 0 à 1000 étant inutilement sauvegardée (sauf pour 2D5/2D6). Il est préférable de travailler sur un programme commençant en #300.

Vous disposer maintenant de toutes les données utiles à la copie ou l'étude de tous les programmes. Il est possible que pour certains programmes, que je n'ai pas encore rencontré, on utilise une interruption pour démarrer le programme. La méthode reste la même; il faut toutefois travailler avec les vecteurs d'interruption au lieu de 2D5/2D6.

Henri-Pierre LEGRY
F-59 DOUAI.

DAILASSEMENT

DAIlassement est une petite, toute petite rubrique, de délasserment insérée dans DAICLIC. Certains diront, (et ils ont raison de le dire !!!), que rien que lire DAICLIC est déjà un délasserment ! D'accord, mais nous instaurons tout de même une rubrique DAIlassement dans un autre délasserment !!!

Enfin, bref, limitons les dégats, nous espérons que cette petite rubrique vous fera rire ou sourire !

N'hésitez pas à nous envoyer vos petits dessins, vos petites blagues sans oublier toutes les excellentes caricatures que votre ame de super-dessinateur peu créer !

A bientôt donc, mais n'oubliez pas, encore une fois, que c'est grace à vous que cette rubrique vivra !!!

A DAItot !

Marc Vandermeersch.

DAILASSEMENT

Et voici enfin la rubrique réservée exclusivement aux "INTELLEC'S", on s'est surpassés pour cette première fois !

Tout d'abord, voici une phrase d'auteur bien connue qui devrait vous plaire :

"Il n'y a que DAI qui m'aïlle !"

(S.V.P, riez ou souriez au moins, vous pourriez me vexer !!!)

Et ensuite, après cette surchauffe de vos méninges :

DAIclic à l'immense plaisir de vous offrir le plus petit programme de jeu existant sur DAI !!!

En effet: il ne prend que deux lignes de programme !

```
10 REM *****
20 REM *** PLUS PETIT PROGRAMME DE JEU ***
30 REM ***                               ***
40 REM *** (c) P. JANIN & DAIclic '85 ***
50 REM ***                               ***
60 REM *****
70 REM
80 REM LE PLUS PETIT PGM DE JEU SUR LE DAI !!!
90 REM IL TIENT 2 LIGNES, 'REM' EXCLUS !!!!!!!
100 REM AVANT DE LE RENTRER, TAPEZ 'IMP INT' !!!
110 POKE #2B9,0:X%=X%+PEEK(#2B1) MOD 2-PEEK(#2B2) MOD 2:A%=#BAA9-(ABS(X%) MOD
30)*2:POKE A%,1:CURSOR RND(30),0:PRINT 888:S%=S%+1
120 IF PEEK(A%)<>#38 GOTO 110:PRINT CHR$(12);"SCORE ";S%;" Perdu ! ESPACE...
";X%=ABS(X%) MOD 30:S%=0:CALLM #D6DA:GOTO 110
```

Après l'avoir tapé, faites RUN (logique !!!).

Vous vous déplacez à l'aides des touches 1 et 0 (gauche et droite). Le but du jeu est d'éviter les petits 888 qui arrivent sur vous !!! Quand vous vous etes 'planté', pressez SPACE pour recommencer !!!

Vous etes d'accord avec moi que ce petit programme est formidable n'est-ce-pas ???

Ce chef d'oeuvre de programmation nous a été envoyé par Pascal Janin, merci Pascal !!!

Maintenant, le défi est lancé , si vous faites plus court, envoyez-nous vos réalisations. (n'oubliez pas de nous les envoyer sur cassette, cela nous évitera les longs moments de frappe de programmes.)(hum !!!)

Votre serviteur, l'inimitable DAIBIL (devinez qui!...).

AUTOCOLLANTS

Vous qui aimez votre DAI, montrez-le!

Comment?: En demandant les superbes autocollants proposés par DAICLIC.

Placez-en partout où vous pourrez ou offrez-les à vos amis DAIistes!... (les autocollants résistent à l'eau)



Pour les recevoir, rien de plus simple; envoyez-nous un petit papier avec votre nom (+ votre adresse si vous n'êtes pas abonné) et la mention 'I LOVE MY DAI'. Mettez-le dans une enveloppe adressée à DAICLIC en n'oubliant pas de joindre des coupons-réponse internationaux (1 par commande + 1 par autocollant). Vous pouvez-vous procurer ces coupons dans tous les bureaux de poste de Belgique et d'ailleurs.

=====

Listing
=====



```

10  GOTO 300
100 REM          Routine
110 C% = FC%*#40 + CC%
120 F% = FF%*#80 + PF%*#40 + ZF%*#20 + VF%*#10 + DF%
130 POKE #2F0,C%
140 POKE #2F1,F%
150 POKE #2F2,X% MOD 256 : POKE #2F3, X%/256
160 POKE #2F4,Y%
170 POKE #2F5,SP%
180 POKE #2F6,IL%
190 CALLM#300,AS
200 RETURN
300 REM          Entrée des variables

310 REM  C%=      Donne les couleurs qui seront utilisées dans
          CC%=    est la couleur du caractère, de 0 à 15.
          FC%=    Fill Colour
                   indique lequel des 4 registres de couleur
                   est à utiliser en fond de couleur pour
                   les caractères :de 0 à 3,soit :
                   #00 registre 0
                   #40 registre 1   de la dernière instruction
                   #80 registre 2   COLORG 0 1 2 3
                   #C0 registre 3

320 REM  F%=      Byte de flags

          FF%=    0 ou 1 =  Fill Flag :
                   si=0
                   la chaine de caractères va se superposer
                   à un contenu quelconque de l'écran sans
                   rien effacer.(Fill Colour,Fill Flag UNSET)
                   si=1   la chaine de caractères s'écrit
                   sur le fond de couleur commandé par
                   l'instruction FC% (Fill Colour, Fill Flag SET)

          PF%=    0 ou 1 =  Position Flag :
                   si=0
                   la chaine de caractères va se reconstruire
                   au départ de la dernière valeur at
                   si=1
                   la chaine de caractères sera construite
                   à la suite immédiate en X% et Y% de la
                   chaine précédente.

          ZF%=    0 ou 1 =  upZide-down Flag :
                   si=0
                   l'écriture se fait dans le sens normal,
                   lisible de gauche à droite horizontalement,
                   ou verticalement de bas en haut

```

```

                si=1
                l'écriture tourne de 180 degrés

VF%= 0 ou 1=   Vertical Flag :
                si=0
                l'écriture se fait dans le sens normal de
                l'écran
                si=1
                l'écriture se fait dans le sens vertical

DF%= 0 à 7=    Dimension Flag :
                la taille des lettres est agrandie de 2^DF%

330  REM  X%=    0 à XMAX =Coordonnée X normale

340  REM  Y%=    0 à YMAX =Coordonnée Y normale
                Attention : Y% doit être plus petit que
                YMAX - Hauteur de la matrice * DF%

                Attention : les valeurs de X% et Y% sont testées
                lors de leur entrée, mais non leurs valeurs
                relatives lors de l'affichage des chaînes :

350  REM  SP%=   eSPacement entre les matrices de caractères

360  REM  IL%=   InterLigne entre les matrices de caractères

370  REM  A$=    Chaîne de caractères à afficher
                Selon la tableau du FGT utilisé, il faut
                s'assurer de la présence des caractères :
                l'appel d'un caractère inexistant appelle
                une erreur (l'écran passe en Mode XA et
                imprime un C suivi de ----).
                Les dessins ou pictogrammes sont le plus
                souvent inscrits sous les CHR$(A%),
                A%= de 1 à 21.
                Attention le Space Bar est souvent omis !!

380  REM  Entrée de vos variables par :

                COLORG A% A% A% A% : MODE A%
                FC%=0:CC%=0
                FF%=0:PF%=0:ZF%=0:VF%=0:DF%=0
                X%=0:Y%=YMAX-A%:SP%=0:IL%=0
                A$="AAAAA"
                GOSUB 100:REM entrée dans la routine d'affichage

390  REM  Toutes vos variables conservent leur valeur
                Exemple : A$=A$+"BBB":GOSUB 100

400  REM  Dans 90% des cas, vous n'utiliserez que quelques
                variables :
                X%=0:Y%=0:CC%=0:A$="AAA":GOSUB 100

60000 GOTO 60000 :REM n'oubliez pas de terminer le programme
                par une boucle.

                COMMENT CHARGER UN F G T
                =====

                Positionnez en premier le langage machine :

```

(celui-ci commence généralement en #29B et se termine selon sa table vers #BFF)

Tapez : (UT) (return) (R) (return) et attendez que le curseur réapparaisse, ensuite, tapez (Z3) (return) (B)

Positionnez ensuite votre Basic

Tapez: (NEW) (LOAD)

Vous avez alors le choix de travailler :
soit sous éditeur : (EDIT) (return)
soit sous Basic : (LIST) (return)

REM: F G T, comme beaucoup d'autres programmes de DAInamic en code-machine possèdent des adresses entre #297 et #2EA. Les copies au départ de cassettes vers le MDCR nécessite l'annulation de certaines commandes :

(CAS) (return)
 Adresse le lecteur de cassettes
(UT) (return)
(S296) (space bar) 02- (0) (curseur gauche)
 Déconnecte les commandes spéciales au MDCR
(R) (return)
 Lit le langage machine
(Z3) (return)
(GF2F2)(return)
 Initialise les commandes spéciales et le tableau du MDCR
(W29B BFF NOM DU FICHER MLP) (return) (space bar)
 Sauve le langage machine sur MDCR
REM: controlez l'adresse de fin de programme

=====
Louis Lambrecht
Data Applications International Club

nov 1984
Brussels

ASSEMBLEUR

UTILISATION DU MATH PACKAGE ROM 1

=====

Il est délicat, les initiés le comprendront, de réaliser des opérations complexes telles qu'une division ou une racine carrée, en assembleur. Il devient alors intéressant d'utiliser les routines qui existent déjà dans la ROM. Vous gagnez de la place mémoire et vous évitez peut-être un "mal de tête".

a) Mathematic accumulator MACC

4 octets (32 bits) sont réservés par l'ordinateur pour effectuer ses opérations sur 32 bits (adresses #D5 à #D8). Si un deuxième nombre est requis, pour une addition par exemple, il est stocké sur 4 octets, HL doit pointer sur le 1er octet.

b) Notation FPT-INT

Si vous êtes capables de travailler, en assembleur, avec des réels, vous êtes forts. Pour l'instant, nous nous contenterons de travailler avec des nombres entiers pour base. Attention, certaines opérations (SQR, LOG,...) nécessitent l'emploi de réels. Une routine permet de convertir un nombre entier en réel et vice-versa. Pourquoi se torturer l'esprit?

Un nombre de 32 bits est stocké sur 4 octets dans la mémoire dans son ordre naturel. Par exemple, le nombre #18A724C1 sera stocké dans la RAM: 18 A7 24 C1.

Attention: l'opération DW ou SHLD, stocke 16 bits en inversant les 2 octets.

SHLD (H=18, L=A7) -> stocké A7 18

Remarque: SHLD + LHLD neutralise cette inversion.

Par exemple, si vous désirez stocker les 4 octets contenus dans les cases mémoires #5000 à #5003, dans MACC:

#5000 11 22 33 44

LHLD 5000H MACC EQU 0D5H

SHLD MACC

LHLD (5000+2)H

SHLD MACC

Les 32 bits sont aussi stockés dans MACC de la même façon.

LES ROUTINES

=====

Les routines mathématiques s'utilisent de la façon suivante:

- * Mettre dans MACC le 1er nombre.
- * Pointer HL sur le 1er octet du 2ème nombre s'il est nécessaire.
- * Lancer la routine.
- * Le résultat se trouve dans MACC.

Pour une opération entière, les deux nombres sont entiers, le résultat aussi, même chose pour les réels.

a) Routine de chargement de MACC

Les 4 octets, dont le premier est pointé par HL, sont chargés dans MACC.

Exemple: charger les octets contenus en #5000-#5003 dans MACC:

```
LXI H    5000H
RST      4      > Routine de chargement
DB       0CH    >
```

Routine inverse: charger MACC dans #5000 à #5003:

```
LXI H    5000H
RST      4
DB       0FH
```

équivalent à :

```
LHLD     5000H
SHLD     MACC
LHLD     5002H
SHLD     MACC+2H
```

b) routines de calculs ENTIERS (INT)

```
addition  MACC=MACC+(HL)    RST    4
                                         DB    4EH
```

```
soustraction MACC=MACC-(HL) RST    4
                                         DB    51H
```

```
1er nombre MACC          entier
2ème nombre pointé par HL  entier
résultat MACC            entier    (V. récapitulatif)
```

c) Routines de calculs REELS (FPT)

Les deux nombres sont des réels, le résultat est réel. Si l'on dispose d'entiers, une routine permet de les changer en réels:

```
mettre de nombre dans MACC (RST 4, DB 0CH)
```

```
puis      RST    4
          DB     4BH    (V. récapitulatif)
```

Attention, le résultat ne doit pas dépasser 32 bits pour les nombres entiers.

OPERATION	1er nbre	2è nbre	résultat	OP	BD	nature
FPT addition	MACC	(HL)	MACC	MACC+(HL)	00	FPT
FPT soustraction	MACC	(HL)	MACC	MACC-(HL)	03	FPT
FPT multiplication	MACC	(HL)	MACC	M.*(HL)	06	FPT
FPT division	MACC	(HL)	MACC	M./(HL)	09	FPT
FPT ABS	MACC	/	MACC	ABS(M.)	1B	FPT
FPT FRAC	MACC	/	MACC	FRAC(M.)	21	FPT
FPT INT	MACC	/	MACC	INT(M.)	1E	FPT
FPT à (puissance)	MACC	(HL)	MACC	M.à(HL)	24	FPT
FPT LOG	MACC	/	MACC	LOG(M.)	27	FPT
FPT EXP	MACC	/	MACC	EXP(M.)	2A	FPT
FPT ALOG	MACC	/	MACC	ALOG(M.)	30	FPT
FPT LOGT	MACC	/	MACC	LOGT(M.)	2D	FPT
FPT SIN	MACC	/	MACC	SIN(M.)	36	FPT
FPT COS	MACC	/	MACC	COS(M.)	39	FPT
FPT TAN	MACC	/	MACC	TAN(M.)	3C	FPT
FPT ASIN	MACC	/	MACC	ASIN(M.)	3F	FPT
FPT ACOS	MACC	/	MACC	ACOS(M.)	42	FPT

	OPERATION	Résultat	DB	nature
Résultat MACC	racine carrée	SQR(M.)	33	FPT
1er nombre MACC	ATN	ATAN(M.)	45	FPT
2ème nombre (HL) si néc.				
<u>exemple:</u>				
<u>a) addition</u>	addition	M.+(HL)	4E	INT
	soustraction	M.-(HL)	51	INT
	multiplication	M.*(HL)	54	INT
	division (quotient)	M./(HL)	57	INT
	division (reste)	M.MOD(HL)	5A	INT
	IAND	M.IAND(HL)	63	INT
	IOR	M.IOR(HL)	66	INT
	IXOR	M.IXOR(HL)	69	INT
	INOT	M.INOT(HL)	6C	INT
	SHL	M.SHL(HL)	6F	INT
	SHR	M.SHR.(HL)	72	INT
	change signe	-MACC	1B	FPT
	change signe	-MACC	60	INT
	INT->FPT	M.->FPT	4B	/
	FPT->INT	M.->INT	4B	/
	MACC->(HL)		0F	/
	(HL)->MACC		0C	/
	MACC->A,B,C,D		15	/
	A,B,C,D->MACC		12	/

b) racine carrée (entier)

5000 -> 1er nbre (9)
5020 -> résultat

```

LXI H 5000H
RST 4      (9->MACC)
DB 0CH
RST 4      (9.0->MACC)
DB 4BH
RST 4      (SQR(9)=3->MACC)
DB 33H
RST 4      (3->MACC)
DB 4BH

```

```
LXI H 5020H
RST 4 (3->5020)
DB 0FH
```

Je joins à titre d'exemple, un programme d'ellipse réalisé en assembleur. Ce programme, sous une forme basic serait beaucoup plus court et tout aussi rapide. Analysez-le et vous trouverez beaucoup d'exemples d'utilisation de la ROM 1.

- Henri-Pierre LEGRY -

```
10 REM COMPOSEUR AUTOMATIQUE DE NUMEROS DE TELEPHONE
20 REM (C) Ch.POELS - novembre 1984
30 REM d'apres un programme original de E.NEVE
40 REM
50 REM MONTAGE A REALISER:
60 REM
70 REM . remote (CAS1 ou 2)
80 REM . I I
90 REM . I I
100 REM . I I
110 REM . I I
120 REM . I I
130 REM -----I I-----
140 REM ligne RTT telephone
150 REM -----
160 REM
170 REM On utilise simplement le remote d'une sortie
180 REM cassette comme interrupteur sur un fil de la
190 REM ligne.
200 REM
210 COLORT 0 10 0 0:MODE 0:POKE #74,1:POKE #75,95
220 TEMPO=0.7:REM ou TEMPO = 1 suivant la ligne
230 POKE #40,0:REM on décroche...
240 PRINT CHR$(12);TAB(13);"COMPOSEUR DE NUMEROS DE TELEPHONE"
250 PRINT TAB(13);"-----"
260 PRINT TAB(16);"( Ch.POELS - novembre 1984)"
270 PRINT ;PRINT ;INPUT "Numero de telephone";T$:PRINT
280 PRINT ;PRINT "[ SPACE ] pour composer le numero...";CALLM #D6DA:PRINT
290 FOR I%=1 TO LEN(T$)
300 1 C$=MID$(T$,I%-1,1)
310 1 PRINT C$;
320 1 C%=ASC(C$)-48:IF C%<0 OR C%>9 THEN 380
330 1 IF C%=0 THEN C%=10
340 1 FOR J%=1 TO C%
350 2 POKE #40,#30:WAIT TIME 3.33*TEMPO:POKE #40,#0:WAIT TIME 1.55*TEMPO
360 1 NEXT J%
370 1 WAIT TIME 20
380 NEXT I%
390 GOTO 270
```

MDCR & BASIC

COMMENT UTILISER FACILEMENT LES COMMANDES MEMOCOM DANS UN PROGRAMME BASIC ?

Les possesseurs d'un MEMOCOM ont pu se rendre compte de la pauvreté du manuel d'utilisation concernant la façon d'utiliser les différents ordres à l'intérieur d'un programme BASIC.

Ces quelques lignes vous permettrons peut-être d'y remédier.

1. Programmation à travers un 'REM'

Intéressons-nous aux ordres REWIND et SKIP (la structure est semblable pour les autres commandes).

La syntaxe à respecter est la suivante :

```
100 CALLM #F000 : REM REW Z
150 CALLM #F000 : REM SKIP Z
```

Cet ordre (CALLM#F000) ne peut être suivi que d'ordres relatifs au MDCRD et pas d'ordres BASIC 'normaux', jusque là pas de gros problème.

L'inconvénient de taille est que Z doit avoir une valeur comprise entre 0 et 9, de plus, une tentative du style :

```
5 F% = 5
10 CALLM #F000 : REM SKIP F%
```

est vaine et provoque un SYNTAX ERROR IN LINE 10

En d'autres mots cela signifie que pour 'sauter' par exemple 15 fichiers, il faut procéder en deux étapes :

```
100 CALLM #F000 : REM REW 9
150 CALLM #F000 : REM REW 6
```

ou encore

```
100 FOR B = 1 TO 15
150 CALLM #F000 : REM REW 1
200 NEXT B
```

Dans les deux cas, l'écriture est lourde, pour surmonter ces difficultés, il suffit d'utiliser d'autres points d'entrée que #F000.

2. Programmation directe

Le manuel d'utilisation reprend les adresses d'implantation des routines des différentes commandes :

#F000 : REMS	= exécute REM statement	
#F003 : DNERR	= initialise ON ERROR	
#F006 : NONERR	= met hors service ON ERROR	
#F009 : OPEN	= ouvre le DCR sélectionné	
#F00C : DCR	= sélectionne le DCR	
#F00F : CAS	= sélectionne le CAS	
#F012 : CRWND	= commande REWIND 'F' files	F représente un nombre compris entre 0 et 255, stocké en #1B0
#F015 : CSKIP	= commande SKIP 'F' files	
#F018 : CVER	= commande VER	
#F01B : CLAST	= commande LAST	
#F01E : CLOOK	= commande LOOK	

On peut utiliser les routines REW et SKIP implantées respectivement en #F012 et #F015, en prenant soin d'indiquer le nombre de fichiers à 'sauter' en #1B0.

50 POKE #1B0,25 : CALLM #F012 équivalent à REW 15.

Cette procédure a l'avantage de pouvoir paramétrer un ordre SKIP ou REW de la manière suivante :

MDCR & BASIC

```
50 INPUT " NOMBRE DE FICHIERS A SAUTER "; F
60 POKE #180,F
70 CALLM #F015
```

équivalent à SKIP F.

Les autres commandes peuvent s'utiliser de la même manière :

exemples :

écrire	50	CALLM #F01B
est équivalent à	50	CALLM #F000 : REM LAST
de même	100	CALLM #F01E
est équivalent à	100	CALLM #F000 : REM LOOK

Il est aisé de voir quelle est la solution à la fois la plus souple et la plus avantageuse.

Ces procédés sont tout aussi valables en mode direct, ce qui permet pour les instructions REWIND F et SKIP F de tourner l'interdiction d'avoir la valeur de F comprise entre 0 et 9.

Fabrice DULUINS Pour D.A.I.C. 20 décembre.

TESTS LOGICIELS

Encore et toujours des tests !!!

Mais oui! Il faut savoir à quoi s'attendre lorsque vous achetez un programme, c'est la moindre des choses...

Si vous désirez voir des tests (ou écrire) des tests de logiciels particuliers, n'hésitez pas à nous écrire...

Si vous jugez que les tests sont mal écrits, si les rubriques des cotes (Rapidité, Aspect, Facilité d'emploi, etc...) ne répondent pas à ce que vous attendiez, dites-le nous !!!

Pour les tests de logiciels venant d' I.D.C software (voir dans cette revue !) et les autres, nous tâcherons d'être les plus impartiaux, il évident que nous n'allons pas aller écraser une parution de nos amis de DAInamic simplement pour mettre DAIClic en valeur.

De toutes façons, si nous faisons cela, vous vous en rendriez bien vite compte !!!

Marc Vandermeersch.

LOGICIEL:

DAITEXT

Je ne m'attarderai pas sur les caractéristiques de DAITEXT, qui vous sont suffisamment présentées dans ce numéro.

Je me pencherai plus sur l'emploi et l'aspect visuel de ce très bon programme de traitement de textes, écrit par Alain Mariatte.

Après chargement et lancement du programme, DAITEXT se présente de la manière suivante :

Les lignes 3 à 23 de couleurs noir/gris.
Les lignes 0 à 3 de couleur noir/orange.
Une ligne verticale orange coupant la page en deux parties.

Vous écrivez votre texte en mode 'commande' et pas en mode éditeur, ce qui évite le ralentissement du système à cause du scrolling (déplacement horizontal ou vertical) de l'écran. Vous tapez votre texte dans la partie grise de l'écran (partie supérieure.).

Dans la partie orange de l'écran (partie inférieure), sont affichées toutes les commandes à votre disposition.

La ligne verticale représente la fin de votre ligne sur imprimante, cette ligne 'imprimante' peut faire deux lignes 'écran' puisque nous ne travaillons pas en mode éditeur.

Cette ligne verticale représentant la fin de ligne papier est évidemment déplaçable, suivant le nombre de caractères par ligne que vous désirez obtenir sur votre imprimante .

Voyons maintenant d'un peu plus près le style, le contexte général d'emploi de DAITEXT :

Une fois le programme lancé, vous vous trouvez directement près à taper votre texte. Pour ce faire, ce n'est pas très compliqué, il suffit de savoir ce que vous voulez écrire (ce n'est pas toujours évident pour certains), et de taper votre texte comme sur un bonne vieille machine à écrire mécanique.

Vous êtes à ce moment là dans le mode TEXTE, c'est-à-dire que la machine n'attend plus que vous (ou plutôt vos doigts) pour travailler (brave petit DAI va !!).

Si vous désirez entrer une COMMANDE (par exemple, charger un fichier), rien de plus simple, il suffit de passer dans le mode que nous appellerons mode COMMANDE (logique !), en pressant FH (flèche haute, vous savez, une des touches grises (ou vertes) ?).

Un menu compacté apparaît alors dans la case réservée aux commandes dans le bas de l'écran. (voir plus haut, case orange).

Il suffit alors de choisir votre commande en pressant la touche correspondante à l'initiale de la commande voulue.

Il y en a plus d'une, (L)ister, (I)mprimer, (E)diter, (F)in, (T)abulation,...

Comme vous pouvez le constater, les commandes ne sont pas bien dures à enclencher !!!

Ces commandes sont pour la plupart simples, très simples d'emploi,

par ailleurs, un mode d'emploi très complet est livré avec le programme.

Toutes les commandes DAITEXT fonctionnent sans aucun problème.

J'ai particulièrement apprécié les commandes d'édition et de correction de texte (duplication de ligne, paragraphe...), qui sont simples d'emploi et très pratiques.

Pour corriger une partie de votre texte, le programme se met alors en mode EDITEUR, ce qui permet un repérage rapide de la faute.

J'ai également apprécié la possibilité d'imprimer des caractères accentués sur papier. (ils ne sont pas représentés en accentués sur l'écran). En effet, il était possible d'obtenir assez facilement les caractères accentués è,é,à,ç sur les autres traitements de textes, mais pour obtenir des trémas ou des accents circonflexes, c'était compliqué. (tréma, back space, caractère...).

En bref : DAITEXT est un programme de traitement de textes valable, qui tourne en toute sécurité (pas de plantages jusqu'à présent !). Il est rapide et simple d'emploi. Aux premiers abords, on est un rien dérouté par la frappe au "kilomètre" et par le fait que l'on passe directement à la ligne en fin de ligne écran (donc pas de scrolling horizontal), mais cela passe rapidement. Il est écrit en français, avec possibilité d'emploi de caractères accentués. Le mode d'emploi est clair, précis tout en n'étant pas un roman fleuve ! Ce mode d'emploi est très bien conçu, on y trouve rapidement le renseignement voulu sur la marche à suivre pour employer toutes les fonctions du programme. Sa facilité d'emploi vous plaira certainement.

Possesseurs de Ken Dos, la version DAITEXT 20 est totalement compatible (possibilité de chargement de fichier depuis le directory).

Possesseurs de floppys INdata, les versions DOS sont totalement compatibles.

Cotation (/10):

- Facilité d'emploi	: 10
- Présentation	: 7
- Mode d'emploi	: 9
- Vitesse de frappe	: 10
- Vitesse de traitement	: 8
- Compatibilité imprimantes	: 10
- Cote personnelle	: 9
=====	
- Cote moyenne	: 9 (!)

Auteur : Alain Mariatte
Diffusion : IDC SOFTWARE

Marc Vandermeersch
pour DAIC Bruxelles.

TEST LOGICIEL:

MAILING LIST est un programme de gestion d'adresses écrit par Christian Poels.

Pour commencer, j'ai tout spécialement apprécié la routine de lecture et de sauvegarde des fichiers qui est super rapide !

Les fichiers d'adresses traditionnels utilisent des tableaux (SAVEA & LOADA) pour les sauvegardes. Cela a pour inconvénient qu'il est toujours nécessaire de réserver une certaine zone mémoire libre égale à la taille du fichier pour la mise en place des variables. Ceci limite par deux les possibilités de stockage des fiches.

De plus, le système SAVEA/LOADA prend pas mal de temps au chargement, à cause de la réorganisation des tableaux.

Le système employé dans ce programme est différent, et nettement plus avantageux : la solution est de sauver l'entièreté de la zone des variables en un seul bloc sur cassette, et ceci grâce à une routine machine appropriée.

Le chargement des fichiers est donc de ce fait rapide et sur.

A titre d'exemple, un fichier de 256 fiches prend environ 30 secondes pour être lu sur DCR avec cette routine. Si dans ce cas, nous avions employé les routines LOADA et SAVEA du DAI BASIC, le chargement du même fichier aurait pris un peu plus de 4 minutes sur DCR. (à cause de la réorganisation des tableaux qui prend pas mal de temps.)

Il est évident que si le fichier ne comporte qu'une seule fiche remplie (ce qui est plutôt rare !), la sauvegarde sur cassette durera également 30 secondes.

Un autre avantage de ce système est que l'on peut enregistrer un nouveau fichier sur un ancien sans risquer de "mordre" sur le programme suivant de la cassette. (ce qui évitera à certains quelques bonnes blagues.)

Pour ce qui est du programme lui-même, il est très facile d'utilisation, grâce à ses nombreux menus.

Le menu de départ vous propose une saisie, une consultation, un tri, une impression format 'étiquettes' ou 'listing', une annulation des données ainsi qu'un appel vers un sous-menu 'commandes DCR' qui vous permettra de gérer votre lecteur de cassettes digitales.

Tout se fait très simplement...

Dans la plupart des fonctions du programme, il faut sélectionner certaines fiches que l'on désire voir ou supprimer (consultation, impression, annulation...).

Il est très facile de retrouver les fiches voulues :

Un menu reprenant toutes les caractéristiques d'une fiche (nom, prénom, adresse, code postal, ville, pays, téléphone et remarque).

Il suffit de presser la touche correspondant au numéro de la caractéristique à l'aide de laquelle vous aller retrouver une ou plusieurs fiches.

Le programme vous demandera alors une clé de sélection, c'est à dire ce que vous savez de la (ou des) fiche(s) que vous recherchez pour la caractéristique donnée précédemment.

MAILING LIST

Cette clé de sélection ne doit pas obligatoirement représenter la caractéristique entière correspondant à la (ou aux) fiche(s) recherchée(s).

Pour obtenir par exemple sur étiquette la liste de toutes les personnes dont le nom commence par la lettre D, il suffit de passer dans le mode 'étiquettes'. A ce moment là, le programme vous donne toutes les caractéristiques sur lesquelles vous pouvez 'jouer' pour trouver les fiches voulue. Vous pressez '1' (numéro correspondant à la caractéristique 'Nom') et l'ordinateur vous demande ensuite la clé de sélection... A ce moment, vous pressez 'D', le critère de recherche. Si votre imprimante est allumée (eh oui !), toutes les personnes dont le nom commence par 'D' dans votre fichier veront leur noms et adresses figurer sur une superbe étiquette ! (les étiquettes ne sont pas livrées !).

Le 'Tri' des fiches est très rapide ! Félicitations car en Basic, il me semble difficile de faire mieux !

Une autre fonction amusante et utile est la fonction 'téléphone'. Là, le Dai vous demande la personne à appeller (méthode de recherche de la fiche expliquée plus haut) et forme pour vous le numéro à l'aide du petit relais de la sortie cassettes. (vous devez bien évidemment raccorder votre Dai à la ligne !!!)

En bref : MAILING LIST est une splendide réalisation, beaucoup plus performante que plusieurs autres.
Aucun plantage durant tous mes essais.
La fonction téléphone est très pratique.
La vitesse du tri est appréciable.
La présentation soignée rend ce programme très simple d'emploi.

En remarque, j'ajouterais que j'ai employé des anciennes versions de ce programme depuis plus d'un an, et elles ne m'ont jamais joué de tours!

Cotation : (/10)

- présentation	: 9
- rapidité recherche	: 7.5
- rapidité tri	: 9
- mode d'emploi	: 8
- sécurité	: 10
- capacité	: 8
- cote personnelle	: 9
<hr/>	
- cote moyenne	: 8.5

Marc Vandermeersch.
(Pour D.A.i.C.)

TEST

OTHELLO

Après une présentation soignée jusque dans les détails (un superbe 'déroulement' d'écran), l'ordinateur vous demande la couleur des pions que vous désirez (blancs/noirs).

Après cela, deux options s'ouvrent à vous :

- A - Recommencer une partie complète.
- B - Continuer une partie en cours ou analyser des positions.

La solution A vous entrainera dans une lutte effreignée contre la machine... L'ordinateur joue très très bien (trop bien même, je n'ai pas encore réussi à le battre). (grrrrrrr !!!!!!!!!!!!!)

Si vous choisissez la solution B, vous pourrez placer différents pions de différentes couleurs dans la grille de jeu, et par après, vous pourrez jouer avec la grille de jeu telle que vous l'aurez conçue. (c'est jusqu'à présent la seule manière avec laquelle j'ai pu battre mon redoutable adversaire)...

Dans tous les cas, vous voyez à l'écran le temps (en temps réel) que vous avez mis pour votre dernier coup (de même pour l'ordinateur), votre moyenne de temps par coup (et aussi celle de l'ordinateur) et pour finir, l'addition de la durée de tous vos coups (et de ceux de l'ordinateur).

Ces temps ne sont qu'indicatifs, vous pouvez prendre tout le temps que vous désirez pour jouer votre coup.

L'ordinateur met, en moyenne, entre 1/2 et 1 minute pour répondre, ce n'est peut-être pas très rapide mais c'est efficace !!!

Durant le temps de réponse de l'ordinateur, vous voyez où ce dernier en est dans sa réflexion. En effet, il teste une par une toutes ses possibilités de jeu. Une croix est affichée dans la case correspondant à celle que l'ordinateur étudie en particulier. Un carré se place autour de la case qu'il retient, en attendant de peut-être trouver mieux parmi tous les coups qu'il peut encore tester.

En bref : Un superbe programme d'Othello, manifestement le meilleur du point de vue présentation et qualité de jeu, 'tournant' sur le DAI. Un programme qui se doit d'être dans la collection de programmes des 'intellectuels' et des autres.

Cotation (/10)

- Présentation	: 10
- Rapidité	: 7
- Maniabilité	: 8
- Qualité de jeu	: 9
- Facilité d'emploi	: 10
- Aptitude à battre les DAI-istes	: 10 (jusqu'à preuve du contraire)
- Cote personnelle	: 9
=====	
- Cote moyenne	: 8

Auteur : Christian POELS
Distribution : IDC SOFTWARE

Marc Vandermeersch,
Pour DAIC Bruxelles.

EDITEUR DE DESSINS

Ce programme est, à mon avis, l'un des meilleurs programmes d'édition graphique existant sur le DAI.

Il est à la fois puissant, rapide et fait appel à plusieurs petites routines machines n'existant pas sur le DAI telles que les 'sprites' et le 'paint' qui sont très rapides.

Une fois le programme lancé, vous vous retrouvez à l'écran, représenté par une petite flèche (sprite) que vous pouvez faire bouger très rapidement à l'écran.

Dans le bas de l'écran, plusieurs cases remplies de petits dessins (icônes) n'attendent que vous pour dessiner.

Le mode d'emploi est très simple, vous placez la flèche dans la case correspondant à la fonction voulue et vous pressez le bouton du paddle. La fonction se met en mémoire et vous pouvez l'employer.

Je ne m'attarderai pas sur toutes les fonctions existantes qui sont celles que tout bon éditeur de dessin se doit de posséder (ce qui n'est pas toujours le cas !).

Les possibilités de mélange graphisme/texte sont énormes grâce à un puissant système d'écriture/sprite entièrement (et très facilement) redéfinissable.

Un bon point pour ce programme est que je ne suis pas encore parvenu à le planter. (et pourtant j'ai essayé !!!)

Un autre gros avantage du système est que vous pouvez sauver votre dessin sous forme d'array que vous pourrez relire par après à l'aide du programme. Ce système permet donc de sauver une suite de bytes sur cassette correspondant à toutes les instructions à faire pour faire apparaître le dessin à l'écran. C'est donc nettement, mais alors là, nettement plus rapide qu'une copie d'écran puisque qu'un dessin valable ne prend en moyenne que 1K, 1,5K...

Cela plaira donc aux possesseurs de cassettes audio (aini qu'aux possesseurs de dcr !!!), mais ceux qui veulent s'entêter à sauver leur dessin sous forme de copie d'écran pourront toujours le faire !

Une autre routine très intéressante est celle qui vous permet d'éditer un dessin que vous avez fait. En effet, je vous le rappelle, le programme retient en mémoire toutes les fonctions ainsi que leur paramètres (avec possibilité de sauvetage, voir plus haut). Cela s'appelle une table d'ordres. Et bien, cette table d'ordres est totalement et facilement modifiable par la suite... Plus clairement : si vous faites une 'gaffe' (qui aurait gaché plusieurs heures de travail sur un autre éditeur de dessins), vous pourriez la supprimer sur cet éditeur-ci !

En Bref : Editeur de dessin est un programme rapide et sûr qui devrait plaire à tous les artistes amateurs de DAI.
Il est très facile d'emploi.
Le mode d'emploi est plus que convenable.
Il est très agréable de pouvoir modifier son dessin par la suite à l'aide de l'éditeur de la table d'ordre.

Cotes (/10) :

- vitesse : 8

- aspect : 8
- surete : 10 (jusqu'à preuve du contraire)
- qualite : 9
- maniabilite : 10
- cote personnelle : 9
=====

- cote moyenne : 9 (!)

Auteur : Walter Costa (France)
Diffusion : I.D.C SOFTWARE
Prix : 1400 FB (voir dans ce DAIClic)
(+ 150 fb sur Dcr)

Marc Vandermeersch.
(Pour D.A.i.C.)

BASIC

PAGE 01 -- BALONS

```
100 MODE 0:MODE 5
110 COLORG 0 1 2 3
125 FOR J=1.0 TO 50.0
126 1 R%=RND(15.0)+5.0
127 1 R=2.0*R%+1.0
128 1 C=RND(13.0)+2.0
130 1 X=RND(336.0-2.0*R)+R
140 1 Y=RND(256.0-2.0*R)+R
141 1 DRAW 335,255 X,Y 1
142 1 N=1046.37*EXP((INT(RND(12.0))/12.0)*LOG(2.0))
148 1 SOUND 2 0 RND(15.0) 3 FREQ(N)
150 1 FOR I=0.0 TO R
151 2 IF 0.0<>FRAC(I/2.0) GOTO 170
152 2 A=SQR(R^2.0-(R-I)^2.0)
155 2 DRAW X-A,Y+R-I X+A,Y+R-I C
157 2 DRAW X-A,Y-R+I X+A,Y-R+I C
170 1 NEXT
180 NEXT
```


Le club CAROLO DAI a été fondé fin février 1983 afin de rapprocher différents utilisateurs du DAI isolés avec leur machine.

Le club est ouvert à tout le monde, même aux personnes ne possédant pas d'ordinateur DAI, sans aucune limite d'âge.

Nous disposons:

d'un local, prêté aimablement par l'ASBL en micro-informatique COPERNIC.

d'un matériel complet: DAI 48k, moniteur RVB
double lecteur de disquettes INDATA 640 k
lecteur de cassettes digitales MEMOCOM DCR
(différents membres du club disposant d'un autre système de disquettes apportent leur matériel de temps à autre)

d'une bibliothèque d'informatique générale et d'ouvrages spécifiques au DAI (manuels, hardware...)

d'un ensemble de plusieurs centaines de programmes sur cassettes audio, accessibles aux membres.

Nos activités:

- échange de programmes personnels
 - présentation des nouveaux logiciels disponibles dans le commerce
 - participation à la revue DAIClic !!!
 - dépannages et réparations
 - prêt de matériel
 - en collaboration avec COPERNIC, organisation de stages de micro-informatique pour le grand public
 - des journées de démonstration (ex. le Meeting International du DAI PC qui s'est tenu le samedi 20 octobre 1984 à Nivelles (BELGIQUE))
 - des exposés sur des points précis:
 - la video-ram, l'apprentissage de l'assembleur, le CP/M
 - les techniques de programmation (structuration, tri..),
 - création de nombreux périphériques: table traçante, modem, crayon optique
 - réalisation d'un DAI PORTABLE qui fonctionne parfaitement
-
- mise au point de nouveaux logiciels: EXTENDED BASIC , GENERATEUR AUTOMATIQUE DE PROGRAMME

Rien ne vous empêche de venir
nous dire bonjour
un vendredi soir

Cordiale invitation à tous

Les membres du club CAROLO DAI

DAIC NEWS

DAIC

Cher ami DAI-iste.

Si tu ne connais pas le DAIC (Digital Applications International Club) , tu dois absolument y venir à l'une de ses prochaines réunions.

Le DAIC rassemble en effet de nombreux passionnés d'informatique et du DAI en particulier. En venant aux réunions, tu pourras commenter avec eux tes expériences et développer tes connaissances.

Les activités du club sont en effet aussi diverses que nombreuses, dans le domaine du Hardware (interfaces, réalisations diverses), comme dans celui du Software (loisirs, enseignement, gestion ...).

Des réunions ont lieu
tous les premiers mercredis du mois
(à partir de 18 heures) et
tous les troisièmes dimanches du mois
(à partir de 14 heures),

à l' ECOLE MOYENNE ET SUPERIEURE D' INFORMATIQUE,
434, CHAUSSEE D'ALSEMBERG - 1180 BRUXELLES (UCCLE).

=====
Pour te donner une idée des sujets abordés,
voici les thèmes des réunions du 1er semestre :

3 / 1 : Programmes musicaux.
20 / 1 : Programmes d'éducation.
6 / 2 : Assembleur.
17 / 2 : Synthèse de la parole.
6 / 3 : Applications de gestion.
17 / 3 : Réalisations hardware : drives
3 / 4 : Utilisations graphiques.
21 / 4 : Robots, radio-amateurisme.
8 / 5 : Pascal.
19 / 5 : Jeux d'aventure.
5 / 6 : Traitement de texte.
16 / 6 : Communications.

Il va de soi que tous les autres sujets peuvent également être abordés à ces réunions ; nous disposons d'ailleurs de plusieurs salles pour ceux qui désireraient présenter des sujets particuliers.

Il est notamment possible d'y voir ou d'y échanger des programmes (la programmathèque est actuellement riche de près de 1500 programmes de toutes sortes).

Il est également possible de se procurer des cassettes éditées par le club. Elles sont au nombre de 4 actuellement :

JEUX 1 et 2 : OHELLO 2008 , RUBICOLOR , REFLEXION , NEW-YORK COURSE, LA PUCE, DONKEY-KONG, ANNEAUX HONGROIS ...

RESISTANCE DES MATERIAUX : POUTRES, PORTIQUES, TREILLIS, SECTIONS EN BETON (rectangulaires et en T), EN ACIER, DALLES ...

MODELES DE GESTION : SIMPLEXE , ARBRES DE DECISION, FILE D'ATTENTE, LISSAGES, REGRESSIONS, FINANCEMENT, INVESTISSEMENT ...

Il est encore possible d'acheter du matériel et des accessoires à des prix intéressants : manettes de jeu, DCR, imprimantes, cassettes, disquettes ...

Tu peux enfin obtenir tous renseignements complémentaires auprès de :

Jacques MOENS - Président.
Clos Fontaine des Duucs,6 à 1310 LA HULPE.
Tél.: 02/657.95.60.

Eric NEVE - Vice-Président.
Hoogstraat, 25 à 1512 DWORP.
Tél.: 02/380.35.81.

Fabrice DULUINS - Trésorier.
Allée Tour Renard,4 à 1400 NIVELLES.
Tél.: 067/21.82.10.

DAI-NAMUR INFO

DAI-NAMUR INFO

Respectant une très ancienne tradition, le club DAI-NAMUR a été le dernier en date à voir le jour .

Comme toutes les autres organisations du genre, il est essentiellement destiné à servir de point de rencontre aux amateurs, avertis ou non, intéressés par la programmation du DAI .

Actuellement une demi-douzaine de membres se réunissent régulièrement chaque vendredi soir à partir de 19H30 . Notre local est aimablement prêté par l'Institut Saint Aubin à Namur .

Nos premières activités ont été tout d'abord de nous familiariser avec les affres du langage machine .

Les réalisations à notre actif sont une série de petits utilitaires tel celui présenté dans ce numéro (le chain manquant...) . Nous travaillons actuellement à la mise au point d'une nouvelle EPROM de caractères permettant l'affichage à l'écran des différents accents et des symboles spéciaux tels que les signes mathématiques, l'alphabet grec, les indices et exposants les plus utilisés .

Nous disposons des systèmes les plus divers : cassettes audio, digitales, floppies Indata et Ken-Dos sont les dénominateurs communs de nos sauvegardes .

Face à l'intérêt qui se manifeste au sujet de la création de IDC, nous escomptons augmenter assez rapidement le nombre de nos membres actifs . Déjà nous pouvons vous annoncer la venue de deux amis néerlandophones .

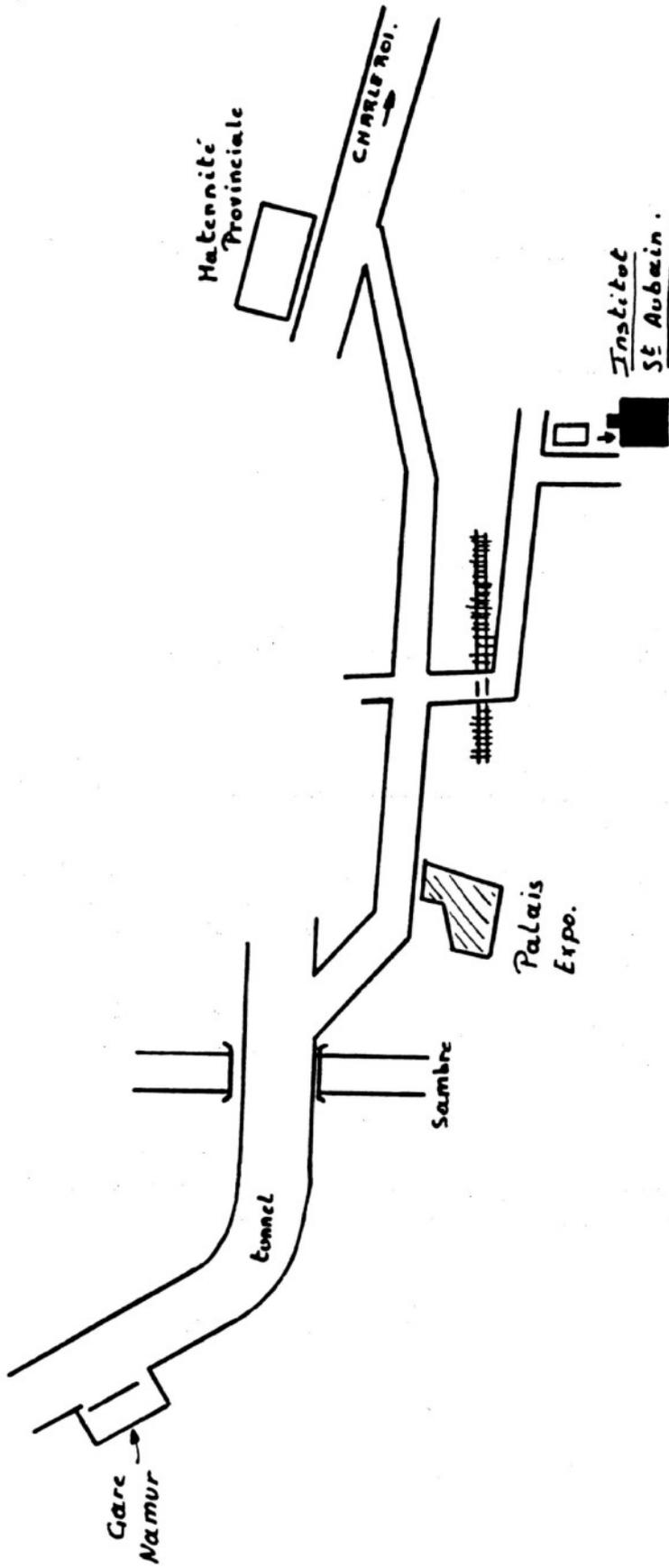
La cotisation annuelle se monte à 1000 fr, dans cette somme sont compris l'abonnement à IDC et à une autre revue destinée à l'usage interne du club.

Pour se rendre à nos réunions, nous vous prions de vous référer au plan ci-joint. Pour de plus amples renseignements vous pouvez vous adresser aux personnes suivantes :

Mr Lange Guy
41, rue du Tram
5700 Sambreville (Velaine)

ou

Mr Mallien Jean-Pierre
183, rue des Nobles
5761 Floreffe (Soye)



MICRODAI NEWS

Chers membres et possesseurs de DAI,

Voici qu'une nouvelle année commence et le Club MICRODAI de Liège n'a pas encore eu l'opportunité de vous remettre ses bons voeux. Qu'à cela ne tienne, c'est avec plaisir et en son nom que je vous souhaite une bonne et heureuse année 1985, ainsi que la réalisation de tous vos voeux.

Désireux de satisfaire au maximum ses membres et dans un esprit de solidarité avec les autres clubs DAI de Belgique, le Club MICRODAI a décidé d'abonner ses membres à la revue DAICLIC. Par conséquent, tous les membres du club MICRODAI recevront à l'avenir ce trimestriel qui comportera un certain nombre de pages uniquement consacrées à notre club.

En complément à cette revue, le Club MICRODAI éditera, sous l'aspect de feuilles volantes, des articles propres à la revue OCTET.

Le Club MICRODAI se tient à la disposition des autres clubs et des DAListes isolés pour consultation ou échange de programmes ou d'informations.

suite =====>>

Depuis peu, les activités du Club MICRODAI se déroulent dans un nouveau local situé au "DOMAINE MILITAIRE de la CHARTREUSE, rue Achile LEBEAU, 4030 LIEGE", les heures d'ouverture du local sont respectivement de 19h à 22h ou plus suivant les cas.

L'adresse pour la correspondance reste la meme, nous vous la rappelons, "CLUB MICRODAI, 39 rue LATOUR, 4020 LIEGE".

Par la meme occasion, nous vous rappelons que l'inscription au club MICRODAI s'élève à 1000 F, revues OCTET (complémentaire) et DAICLIC comprises. Les virements, versements et chèques (uniquement en Belgique) sont à adresser à "MICRODAI, 39 rue LATOUR 4020 LIEGE"; numéro de compte "CGER 034-0080601-56". Pour l'étranger, la cotisation est de 1100 F (à verser par mandat international uniquement).

Le président

Président:
Victor LESPINEUX
39, rue LATOUR
4020 LIEGE
BELGIQUE

Local:
DOMAINE MILITAIRE
de la CHARTREUSE
rue Achile LEBEAU
4330 LIEGE
BELGIQUE

MICRODAI NEWS

PETITES ANNONCES

PETITES ANNONCES

Vends DAI + MDCR + 12 cassettes + jeu d'échec + documentation sur DCE BUS: 50000 FB.

M. COTON; T.02/4274801.

Vends 7 cassettes DAINAMIC (GAMES, DAI DEMO, DARGON,...) + revues DAINAMIC '83 + mode d'emploi français de MULTISOFT (original): 2500 FB.

M. FOULARD; T.02/3545877.

Vends DAI + DCR + paddles + documentation (revues DAINAMIC + doc. générale) + 12aine de cassettes audio + 30aine de cassettes digitales (+/- 400 programmes): +/- 40000 FB.

Giacomo INFANTINO; 138, rue de la Chatqueue; B-4100 SERAING; T.041/369199.

Vends 8 cassettes DCR avec programmes divers (catégorie au choix): 1200 FB. Christian POELS (I.D.C.).

Vends DAI + péritel + 2 paddles 3D + MDCR + 6 K7 DCR + logiciel (valeur + de 2500 FF) + doc. Le tout: 9000 FF (valeur réelle: + de 14000 FF).

Marcel TOULON; 22, rue des Peupliers; F-78790 SEPTEUIL.

Vends MDCR (sans TDS) avec cable et 6 K7 DCR: 10000 FB.

Léon LONEUX; rue Bois de Mont 260; B-4220 SERAING; T.041/336292.

Vends Ken-DOS 2 x 200 K: 40000 FB. Victor LESPINEUX; rue Latour 39; B-4020 LIEGE; T.041/431030.

REMARQUES: Ces petites annonces gratuites pour les abonnés sont exclusivement réservées à des propositions entre particuliers sans objectif commercial et relatives à l'informatique individuelle. Pour les ventes de matériels, indiquez le mois et l'année de l'achat. Les annonces de ventes ou échanges de programmes sont refusées systématiquement. DAICLIC se réserve le droit de refuser une annonce sans avoir à fournir de justification.

Envoyez vos annonces à la rédaction:

I.D.C. - DAICLIC
C/O Christian POELS
Rue des Bas-Sarts 10
B-4100 SERAING

RANDOM

=====

GENERONS des NOMBRES ALEATOIRES < 1 >

=====

par Raymond VANLATHM (Club CARLODAI)

Introduction

Chacun d'entre nous a déjà fait appel à un générateur de nombres aléatoires par l'intermédiaire de l'instruction RND. Ces nombres sont couramment utilisés dans la résolution de certains problèmes, en vue de réaliser des simulations (en entreprise notamment), dans le déroulement de la plupart des jeux, en didactique dans le choix des exercices, et j'en passe. Dans chaque cas nous désirons que les événements qui surviennent, soient inattendus pour l'utilisateur, comme s'ils ne dépendaient que des seules lois du hasard. Or par définition, les nombres aléatoires sont des nombres pris au hasard, qui ne sont donc pas prévisibles d'avance.

Et que fait votre DAI ?

Comme la plupart des ordinateurs il est à même de générer des nombres dits aléatoires, car les séries de nombres générées sont en réalité pseudo-aléatoires, même si elles ont l'apparence aléatoire. En effet, ces nombres sont obtenus à partir d'une formule mathématique, qui est par essence même déterminée, et donc loin d'être aléatoire. Pour faire démarrer une série, il faut nécessairement introduire un nombre appelé amorce ou nombre source (SEED en anglais), soit par l'utilisateur, soit par l'ordinateur lui-même, ce qui donne l'apparence de séries parfaitement aléatoires.

Normalement c'est votre DAI qui fournira automatiquement l'amorce de la série lors de la mise sous tension, et qui générera ensuite le nombre aléatoire initial. A chaque exécution de l'instruction RND, le DAI génère le nombre aléatoire suivant de la série.

Essayons la routine suivante, après une mise sous tension :

```
100 FOR I%=1 TO 5 : PRINT RND(1), : NEXT
```

et exécutons la deux fois de suite. Nous obtenons à l'écran :

```
RUN
0.345451  0.917845  0.703101  0.633126  0.13235
RUN
0.183837  0.527697  0.651674  0.451856  0.531174
```

c'est-à-dire deux suites différentes de nombres aléatoires, constituant les 10 premiers nombres de la série.

Remarquons qu'un RESET ne permet pas d'obtenir à nouveau ces deux suites, et que l'exécution de la routine ci-dessus, affichera à l'écran les nombres aléatoires suivants. Il faut nécessairement éteindre votre DAI et attendre au moins 15 à 20 secondes avant une nouvelle mise sous tension.

Le DAI fera amorcer la série et générera le nombre initial 0.345451, lors de la première exécution de RND(1). Si tel n'est pas le cas, c'est que votre DAI n'a pas fourni l'amorce prévue lors d'une mise sous tension à froid.

Nous venons de faire l'essai de la routine avec l'argument X de RND égal à 1. En donnant une autre valeur positive à X, nous obtiendrons les memes séries, les nombres étant cependant multipliés par la valeur de l'argument. Ainsi avec RND(10), les premiers nombres affichés seront :

3.45451 9.17845 7.03101 6.33126 ...

Ecrire $A = \text{RND}(10)$ ou $A = 10 * \text{RND}(1)$, cela revient au meme.

Et si X prend une valeur négative ?

Que se passe-t-il alors ? La routine générera toujours le nombre initial. Ainsi avec $X = -1$, nous obtiendrons a chaque RUN, cinq fois le meme nombre -0.571069, même sans éteindre le DAI: cette suite s'appelle "série répétitive" par opposition à "série continue" qui désigne la série des nombres générés par une valeur positive de l'argument X.

Ce qui est valable pour $X = -1$ le sera également pour toute valeur négative que prendrait l'argument: le nombre généré est négatif, bien que différent (selon la valeur de X), mais il sera toujours répété.

Dans le cas de $X = -10$ nous obtenons toujours -3.21069. A noter que $A = \text{RND}(-10)$ n'est pas équivalent à $A = 10 * \text{RND}(-1)$. En effet, $10 * (-0.571069) = -5.71069$, ce qui est différent. Par contre RND(-5) nous donne comme nombre généré -1.60534 (multiplié par 2 = -3.21068): nous voyons qu'ici $A = \text{RND}(-10)$ est équivalent à $A = 2 * \text{RND}(-5)$, à l'arrondi de la 5eme décimale près. Nous en verrons plus tard la raison.

A première vue obtenir des séries répétitives avec des arguments négatifs ne semble pas être d'une quelconque utilité en programmation. De fait seul le nombre initial ainsi obtenu pourra servir lors de la mise au point d'un programme, afin que les données fournies soient toujours les memes et dans le meme ordre.

Complétons la routine initiale par la ligne suivante :

```
10    Y = RND(-1)
```

et exécutons la routine complète deux fois de suite. L'écran affichera :

```
RUN
0.389234    0.359846    0.488753    0.388517    0.171452
RUN
0.389234    0.359846    0.488753    0.388517    0.171452
```

Ici pas nécessaire d'éteindre son DAI, d'attendre 15 à 20 secondes avant une nouvelle mise sous tension, et de devoir recharger son programme. Modifions un peu la procédure d'exécution de notre routine. Après un premier RUN, faisons RUN 100. Nous obtenons à l'écran:

```
RUN
0.389234    0.359846    0.488753    0.388517    0.171452
RUN 100
0.969859    0.698164    0.996478    0.145397    0.145897
```

Le 2eme RUN générera les 5 nombres aléatoires suivants de la série continue. Remarquons cependant que la série générée après l'exécution d'un RND avec argument négatif (dans le cas présent avec $X = -1$) est différente de celle générée normalement par un argument positif.

Que se passe-t-il, si à la ligne 10 nous remplaçons -1 par un autre argument négatif ? Les séries continues obtenues sont différentes, lorsque X est impair.

Par exemple avec X= -2, -4, -8, -16, ... nous obtenons la même série que celle générée avec X= -1. Et avec X= -3, -6, -12, -24, ... nous obtenons une autre série. De même avec X= -5, -10, -20, -40, ... encore une série différente... Il existe cependant une relation entre les nombres aléatoires de même rang de 2 séries différentes. C'est ainsi que tous les nombres de la série générée par X= -3 diffèrent de 0.5 de ceux de la série obtenue avec X= -1, mais diffèrent de 0.25 ou de 0.75 des nombres de la série générée par X= -5.

Si X= -1 nous avons:				
0.389234	0.359846	0.488753	0.388517	0.171452
Si X= -3 nous avons:				
0.889234	0.859846	0.988753	0.888517	0.671452
Si X= -5 nous avons:				
0.639234	0.109846	0.738753	0.138517	0.421452

Cela peut être intéressant dans certaines applications. En effet, dans un jeu de PILE [X=1] ou FACE [X=0] par exemple, en début de partie avec X= -1 nous obtenons 5 fois FACE, avec X= -3, PILE sort 5 fois et avec X= -5, nous avons successivement PILE - FACE - PILE - FACE - FACE.

Remarquons encore que pour obtenir une série continue de nombres aléatoires négatifs, il suffit de multiplier l'instruction RND avec argument positif par un nombre négatif; par exemple: A= -1 * RND(1), ce qui n'est pas du tout équivalent à A= RND(-1)!

Des astuces, ... encore des astuces

Lors de la mise sous tension, pour éviter un retour automatique au nombre initial, vous pouvez utiliser une routine d'amorce qui permet de démarrer la série avec le Nième nombre de la série, appelé nombre de départ. Cette même routine permet aussi de sauter N nombres de la série entre deux exécutions du programme.

```
20 INPUT "Entrez un nombre compris entre 1 et 100 ";AZ
```

```
30 FOR J%= 1 TO AZ : Z= RND(1) : NEXT
```

Cette routine, utilisée avec la ligne 10, donnera pour AZ= 2 le nombre 0.488753 comme nombre de départ; et sans la ligne 10 le nombre 0.703101. Le choix de l'amorce AZ peut se faire de manière aléatoire, par exemple : ouvrir au hasard un annuaire téléphonique, pointer son doigt aveuglément sur un numéro de téléphone et en prendre les deux derniers chiffres; ou bien regarder sa montre et noter les secondes comme valeur de AZ.

Cette procédure a l'inconvénient d'être assez lente, surtout si le nombre introduit (AZ) est élevé.

Un moyen plus performant (car très rapide et très varié), où nous pouvons utiliser 6 chiffres du numéro de téléphone (au lieu de 2), ou l'heure complète (heure, minutes, secondes) affichée par votre montre.

A vous de choisir!

Voici donc la routine alternative:

```
20 INPUT "Entrez 2 nombres de maximum 3 chiffres (entre 0 et 999)  
séparés par une virgule comme 345,78 ";L1%,L2%
```

```
30 L%= L1% * 1000 + L2% : A%= L1% * 0.127 + 128
```

```
40 B%= L2% * 0.251 : C%= L% MOD 997 / 4
```

```
50 POKE #12E,A% : POKE #12F,B% : POKE #130,C%
```

A la limite, en n'utilisant que la ligne 50, et en y remplaçant AZ, B% et C% par la valeur décimale 255, à chaque RUN vous initialisez la série aléatoire, effet semblable à une mise sous tension à froid: c'est un autre moyen pour mettre des programmes au point.

Nous verrons plus tard sur quoi nous agissons en utilisant cette routine. Signalons d'emblée que AZ doit être compris entre 128 et 255, tandis que B% et C% doivent être plus petits que 256.

BASIC ETENDU

BASIC ETENDU (1)

Il s'agit d'une librairie modulaire de nouvelles commandes écrite en DBASIC.

LIBGEN.D contient douze nouvelles commandes/instructions générales utilisables en mode direct ou dans un programme. Elles sont regroupées en deux séries :

1) ECRAN

CLS effacement de l'écran sans provoquer de saut de page à l'imprimante

SPEED n modification de la vitesse d'affichage
n<15 ralentissement perceptible
n>60 vitesse normale nécessaire pour l'imprimante (EPSON)

NORMAL après un COLORT A B C D
NORMAL utilise les couleurs A pour le fond et B pour le texte

INVERSE après un COLORT A B C D
INVERSE utilise les couleurs C pour le fond et D pour le texte

WINDOW n,m définit une fenêtre d'écran depuis la ligne n jusqu'à la ligne m pour le scrolling et l'effacement
(n=1:bas de l'écran, m=24:sommet de l'écran)

2) AUTRE

PRTON connecte l'imprimante

PRTOFF déconnecte l'imprimante

REPT avec un GETC permet l'auto-répétition des touches

SWAP A\$,B\$ échange de deux variables alphanumériques

INSTR(I,X\$,Y\$) I: constante entière, X\$ et Y\$ variables alphanumériques
recherche, à partir de la Ième position, du caractère X\$ dans la chaîne Y\$. S'il ne s'y trouve pas, alors INSTR = 0

exemple A\$="A";B\$="BEAU"
PRINT INSTR(2,A\$,B\$) réponse : 3
PRINT INSTR(4,A\$,B\$) réponse : 0

DFTP "X","Y" définition comme variables réelles de toutes les variables dont le premier caractère commence par X jusqu'à Y.
attention Respecter l'ordre de l'alphabet pour X et Y

DFSTR idem pour des variables alphanumériques

Un grand merci à P.JANIN pour sa routine de vidéo inverse.

Le prochain module concernera le calcul matriciel LIBMAT.D.

Fernand Lemoine
Rue du Collège 34
B-6071 CHATELET

Club CARLODAI 20 novembre 1984

BASIC ETENDU

```
2 REM *****
3 REM ** BASIC ETENDU PARTIE 1 **
4 REM ** LIBGEN.D LIBRAIRIE GENERALE **
5 REM ** FERNAND LEMOINE RUE DU COLLEGE 34 B-6071 CHATELET **
6 REM ** CLUB CAROLODAI NOVEMBRE 1984 **
7 REM *****
8 REM
9 ON ERROR GOTO "TRAP
10 "ECRAN REM COMMANDES CLS,SPEED,INVERSE,NORMAL,WINDOW
20 PROCEDURE CLS :PRINT CHR$(27);CHR$(68);CHR$(12);:END PROC
30 PROCEDURE SPEED D:POKE #FF05,D:END PROC
40 B=#235
45 WHILE A<>#C9 DO READ A
50 POKE B,A:B=B+1
52 WEND
54 DOKE #70,#23D:DOKE #6C,#235
56 DATA #F5,#CD,#45,2,#F1,#C3,#FD,#C6
58 DATA #F5,#CD,#45,2,#F1,#C3,#A9,#D9
60 DATA #3E,0,#32,#76,0,#C9
65 PROCEDURE INVERSE :POKE #246,#FF:END PROC
70 PROCEDURE NORMAL :POKE #246,0:END PROC
85 PROCEDURE WINDOW B,H
86 IF B<0 OR H>23 OR B=H THEN ERROR 60:END IF
88 LOCAL CALCB,CALCH,CALCI
90 CALCH=#BFEB-((23-H)*#86):CALCB=#B35F+(B*#86)
92 CALCI=CALCB-16
94 DOKE #8A,CALCH:DOKE #8C,CALCB
96 DOKE #8E,CALCI
98 END PROC
100 "DIVERS REM COMMANDES PRTON,PRTOFF,REPT,SWAP,INSTR,DFFTP,DFSTR
110 PROCEDURE PRTON :POKE #131,0:END PROC
120 PROCEDURE PRTOFF :POKE #131,1:END PROC
130 PROCEDURE REPT :FOR A=#2B1 TO #2B8:POKE A,0:NEXT:END PROC
140 PROCEDURE SWAP VAR X$,Y$:LOCAL HELP$
145 HELP$=X$:X$=Y$:Y$=HELP$:END PROC
150 FUNCTION INSTR(H VAR CAR$,CHAIN$)
155 LG=LEN(CHAIN$):SW=0:A=H-1
160 IF H<=LG THEN REPEAT D=0
165 IF MID$(CHAIN$,A,1)=CAR$ THEN D=A+1:SW=1
168 ELSE A=A+1:END IF
170 UNTIL SW=1 OR A=LG
172 END IF
175 FN = D:END FN
180 PROCEDURE DFFTP X$,Y$:XY=ASC(X$):YX=ASC(Y$)
182 FOR H=XY TO YX:POKE (H+#234),0:NEXT:END PROC
184 PROCEDURE DFSTR X$,Y$:XY=ASC(X$):YX=ASC(Y$)
186 FOR H=XY TO YX:POKE (H+#234),#20:NEXT:END PROC
190 "TRAP IF ERR=60 THEN PRINT "VALEUR INCORRECTE",:RESUME "ECRAN:END IF
```

ATTENTION ce programme ne peut fonctionner que si le programme D-BASIC est en memoire, il est inutile d'essayer si vous ne le possédez pas.

HARD

HARDWARE : Interface très utile pour les utilisateurs de cassette audio.

par Eric NEVE "S.E.L.S." Club D.A.I.C. 21.12.1984
25 Hoogstraat B-1512 DWORP
Tél.02/380.35.81

Le petit système décrit ci-dessous est destiné à tous ceux (nombreux) qui ont éprouvé des difficultés lors du chargement de PROGRAMMES se trouvant sur cassette audio de provenance autre que celles venant de leurs correspondants habituels, dont ils connaissent les caractéristiques de niveaux d'enregistrement mais aussi de VITESSE.

En effet, sur 10 lecteurs, il n'y en a pas 5 qui tournent à la bonne vitesse de 4,5 cm/sec; à 5% près !

Si vous tentez de lire un programme enregistré avec un lecteur qui tournait PLUS LENTEMENT que le votre, les données vont entrer trop vite dans votre DAI et surtout le TITRE, alors comme un canasson devant un obstacle : REFUS. Par contre, l'inverse n'est pas vrai: un programme sauté à une PLUS GRANDE vitesse sera lu correctement à vitesse + LENTE.

Je parle de l'importance du TITRE car une fois que celui-ci a été lu correctement, le DAI est capable de lire à DOUBLE VITESSE, eh oui !

Niveaux d'enregistrements:

- 1) Les cassettes dites 'normales' doivent être de BONNE qualité.
Les cassettes au CrO₂ (Ferro-chrome) DOIVENT être lues avec le SWITCH sur CrO₂ s'il existe, faute de quoi, le résultat sera décevant !
- 2) Pour éviter des " LOADING ERROR 2 (niveau) ou 3 (coupure); avec de mauvais records, on se 'mitonne' un petit amplificateur, avec contrôle de niveau d'Entrée et de Sortie. Fig.1
On entre avec un signal du lecteur de 2 volts maximum c'est-à-dire sans déformations et l'on envoie au DAI de 1 à 8 volts (réglable).
La bonne moyenne est de 4 à 6 Volts 'PROPRES' (comme le Monsieur...) !
- 3) Si vous vous trouvez en présence d'enregistrements trop lents, il vous faudra réduire la vitesse moteur de votre cassetophone .
En général, ce n'est pas un GROS problème. Fig 2
- 4) Sur la figure 2 est représentée l'alimentation Basse Tension telle qu'on la trouve dans la plupart des lecteurs de cassettes.
Pour savoir si l'on peut alimenter l'ampli (9V) au moyen de celle du lecteur, il suffit de savoir combien de piles de 1,5 volts sont nécessaires.
4 piles (6 volts): trop peu pour l'ampli.; 6 piles (9 volts): c'est OK.
- 5) Si, comme c'est souvent le cas, le " jus" Moteur passe dans la commande à distance, il vous suffit d'intercaler un potentiomètre de 100 Ohms BOBINE (non un carbone // puissance) dans cette ligne.
Le Switch 2 en AVANT du réglage vous permettra de commander l'avance ou le rebobinage de la bande sans devoir commander un CHECK ou un LOAD au DAI, ce qui croyez moi est TRES pratique quand on cherche un programme.
De plus, l'indicateur de signal de l'ampli vous indiquera si vous êtes sur un PGM ou entre deux, ou s'il n'y a rien d'écrit sur la bande.
Le Switch 3 permet d'enclencher ou non la vitesse normale sans toucher au réglage de la vitesse basse.

Fig. 1) Ampli - Cassette recorder T© Dai.

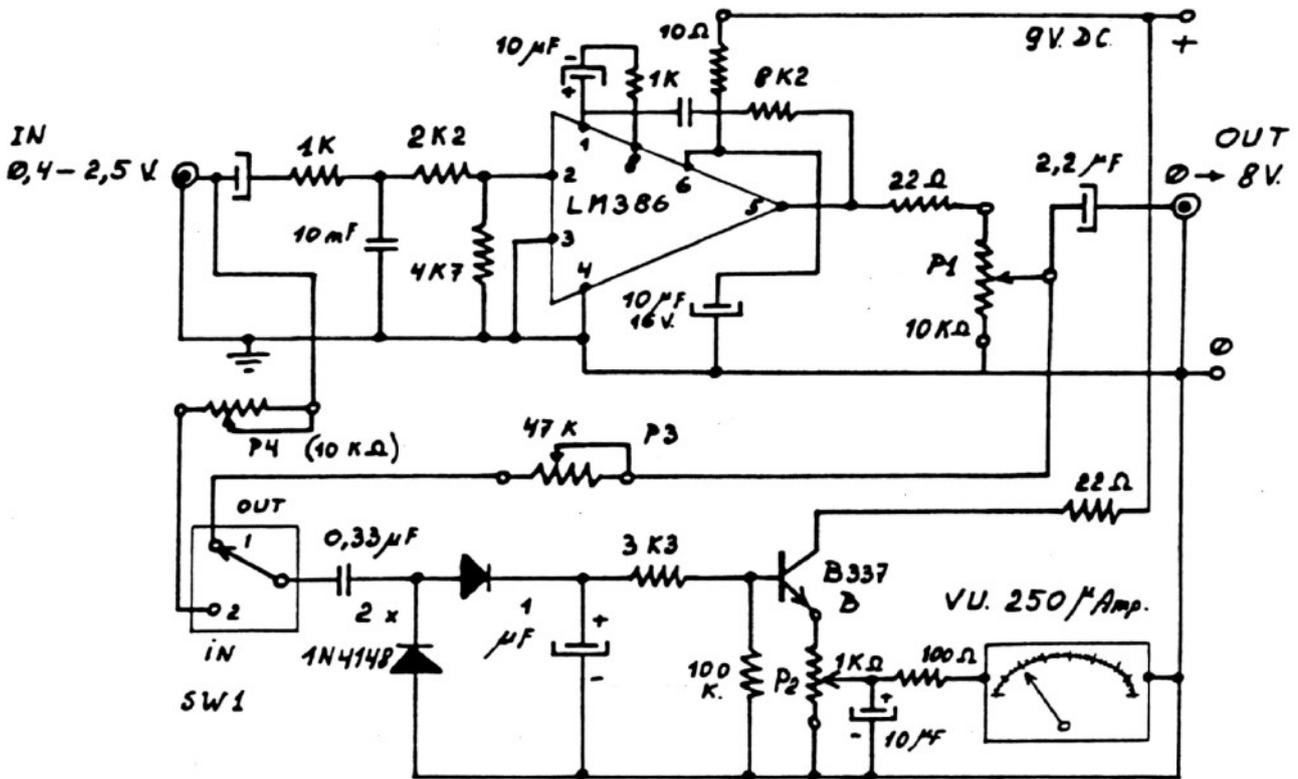


Fig. 2) Alimentation 9 Volts (D.C.)

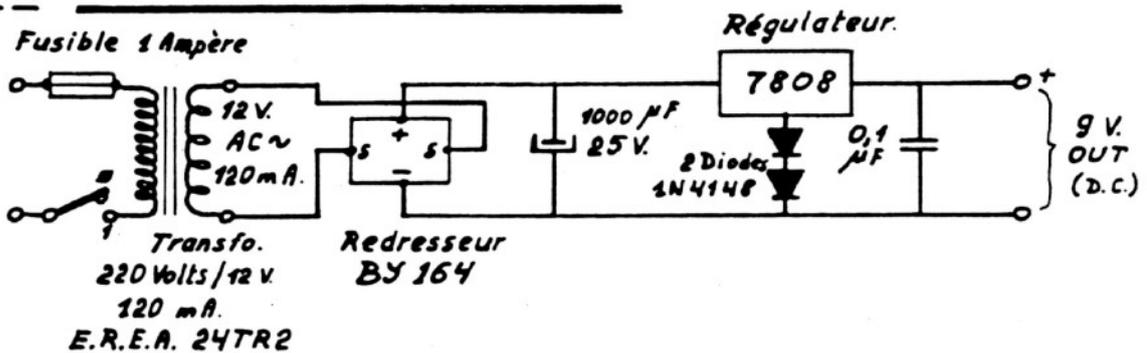
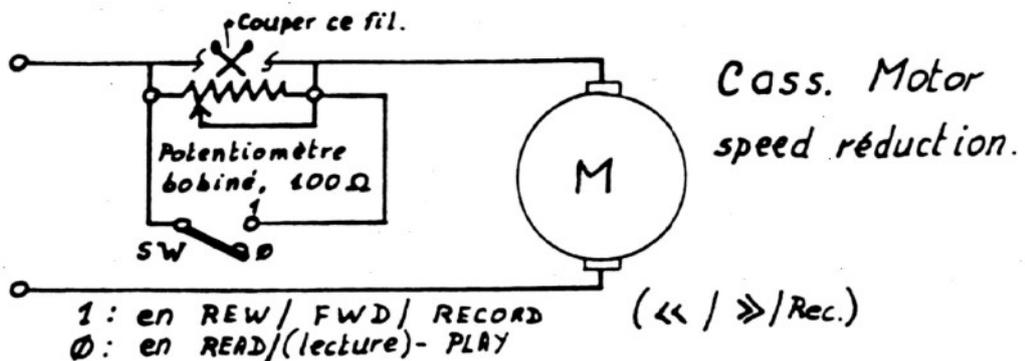


Fig. 3)



6)

REGLAGES ET ESSAIS

**** le DAI NON CONNECTE A LA SORTIE AMPLI ****

Fig.1 : 1) Le lecteur en position de LECTURE d'un LONG programme, régler le Volume de sortie pour trouver avec un simple VOLTMETRE AC une tension de sortie de 2,5 volts environ.

Sur le boîtier ampli, mettre le switch (SW1) en position 2 (input).

Ajuster le potentiomètre P4 à environ 1/3 de l'échelle totale, P2 étant à fin-course .

Switcher SW1 en position 1 (out), le bouton de réglage P1 presque au maximum, mesurer environ 8 volts, régler P3 presque à fond d'échelle; si l'aiguille n'y arrive pas AUGMENTER P2 et diminuer un peu P3; avec ce nouveau réglage de P2 (plus sensible), réajuster un peu le réglage de P4. Ceci fait, le lecteur débitant un programme à 0,5 à peu près.

Au moyen du réglage principal P1, faire varier le niveau de sortie de : 0 à 8 volts ce qui est plus que suffisant.

Ces niveaux variables en plus d'une VARIATION DE LA VITESSE MOTEUR du lecteur de cassettes vous permettront de " RECUPERER " des programmes imprégnables autrement.

N.B. La partie de la Figure 1 encadrée xxx MESURE xxx, moyennant une bonne CALIBRATION peut servir de VOLTMETRE AC ou DC jusqu'à 10 Volts, et ce, avec 4,5 volts ou 9 volts.

Un bon conseil : (le conseiller a PAYE !):

Le LM 324 (4 amplis), circuit d'entrée cassettes et amplis SON dans le DAI 'claque' facilement avec un niveau d'entrée un peu trop ELEVE, et si c'est le cas REMPLACEZ le par un MOS-FET: TL 074 ou TL 084 (TEXAS) plus sensible et PLUS COSTAUD que le LM 324 (Pinning identique).

PROFITEZ-en pour le mettre sur SOCQUET. Vite enlevé vite remplacé !

Et maintenant à vos fers à souder.

par Eric NEVE "S.E.L.S." Club D.A.I.C. 21.12.1984
25 Hoogstraat B-1512 DWORP
Tél.02/380.35.81

Après vous avoir proposé un ampli. à brancher derrière un lecteur de cassettes, je vous décris cette fois le DOUBLEUR de VITESSE de LECTURE.

- 1) Le circuit se compose:
- a) d'un détecteur de signal.
 - b) d'un temporisateur.
 - c) d'une petite logique de commande.
 - d) d'un petit relais de commutation de vitesse.

2) Fonctionnement:

Un circuit détecteur + ampli, capte un signal sur la bande; dès que cela se produit, il commande une temporisation qui ATTEND 2 Secondes environ, c'est à peu près la durée de passage d'un long TITRE précédant les datas, avant de commander le mini-relais de variation de vitesse.

Ce relais reste enclenché aussi longtemps que le SIGNAL est présent.

Dès que le programme a été lu, le relais retombe et le lecteur se retrouve à sa vitesse normale jusqu'au prochain programme rencontré.

!!Ceci ne marche sur votre DAI CHERI que pour autant que :

1o) votre enregistrement soit BON et PROPRE c'est-à-dire sans déformations excessives, en 2 mots BIEN ENREGISTRE.

2o)Ceci est moins marrant pour ceux qui ne l'ont pas: il FAUT que votre lecteur dispose d'une REGULATION ELECTRONIQUE ou du moins une REGULATION ELECTRIQUE de la VITESSE du MOTEUR (par exemple Philips).

Il est évident, n'est ce pas mon cher Watson, que s'il n'y a pas moyen d'augmenter la vitesse de rotation du moteur, il serait difficile de lire en double vitesse!

POURQUOI ? Parce que pas mal de moteurs de cassettes sont pourvus d'un régulateur de vitesse MECANIQUE c'est-à-dire à masselottes.

MAIS si l'on enlève les-dites masselottes, le moteur tournera plus vite et dans ce cas on peut le CONTROLER .C&FD.

* Ya qu'à * démonter le moteur, et ajouter une alimentation réglable, c'est pas la mer à boire pour ceux qui le veulent vraiment !

Le SCHEMA électrique du lecteur lèvera tous les doutes à ce sujet.

3) EXPLICATIONS du SCHEMA :

Fig.1 Le montage électronique n'appelle que peu de commentaires; mais la partie encadrée "pontage" constitue une option: le PONT entre 1-3 fait que le relais S'ENCLENCHE au moment où la temporisation de lecture s'est écoulée. C'est le montage 'normal'.

Le pontage 2-3 a pour effet que le relais est commandé dès la mise sous tension et est désexcité après la temporisation.

C'est selon les BESOINS.

Le relais simple comporte 3 contacts (outre les 2 contacts de la BOBINE : 1 Commun (CO); 1 contact Normalement Fermé (NF); 1 contact Normalement Ouvert (NO).

En temps Normal, le relais ne travaille pas, CO et NF sont en contact.

En commande, le relais travaille, CO et NO sont en contact.

C'est la position de ces fameux contacts qui décidera de la vitesse du moteur Cassette, Normal ou Rapide.

Solutions 1 ou 2 de variations de vitesse:

La première est plus simple à réaliser par un non-professionnel car elle ne demande que peu de tests.

Dans la seconde solution, on s'attaque à la régulation moteur elle-même et cela permet pas mal de réglages différents. D'où l'avantage de celle-ci.

Doubleur de vitesse de lecture.

Figure 1.

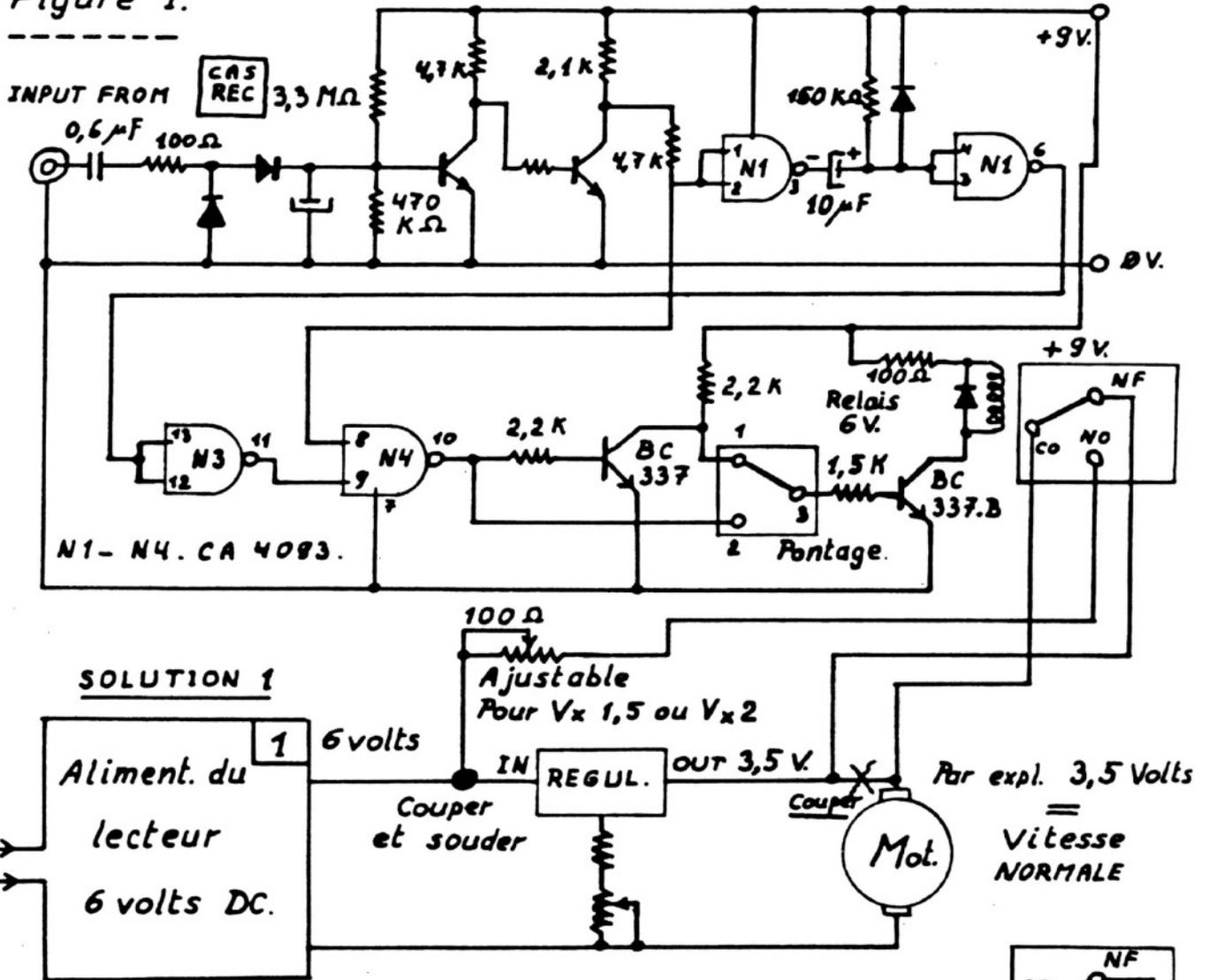


Figure 2 SOLUTION 2

