

Bulletin de liaison des utilisateurs du DAI Personal Computer **.dai club france.**

INFORMATIONS EN VRAC.....

-Cassettes , échanges:

Envoyez vos cassettes pour échange par enveloppe craft sans mettre la boîte de la cassette, c'est moins lourd et ça ne coûte que 2,60F. Les cassettes en retour avec les programmes demandés sont prêtes environ une quinzaine de jours après. Si vous habitez Paris vous pouvez aussi passer les prendre au local.

Quelques précisions pour les échanges: nous échangeons uniquement les programmes de conception personnelle (et originaux si possible) qui sont testés avant envoi (faites tester par vos copains c'est fou le nombre de bogues que l'on repère!). Pour les programmes où il y a des règles à respecter, ou une syntaxe spéciale ou des commandes non triviales, faites un mode d'emploi dans le programme lui-même. Nous passons actuellement beaucoup de temps en décryptage...

Mettre vos nom et adresse dans le programme.

Adjoindre avec votre cassette la liste des programmes désirés.

Nous pensons mettre bientôt en libre circulation les programmes notés avec une seule étoile, envoyez une cassette et soyez patients !

-Courrier

Adressez vos questions et réclamations uniquement par courrier, ça laisse des traces...

Envoyez nous les trucs que vous avez trouvés , de façon à ce que cela profite à tous, que ce soit une astuce d'utilisation ou une idée d'interface ou encore mieux des schémas de réalisations d'interfaces que vous auriez conçue.

-Achats groupés

Que les personnes intéressées par l'achat de matériel par le club et qui n'aurait pas encore envoyé leurs desiderata avec leurs coordonnées, le fasse rapidement si possible. Pour les imprimantes de type IMP 4 ou NEC 8023 cela devrait se faire très rapidement. N'oubliez pas votre numéro de téléphone pour vous contacter au moment du choix définitif.

EN VRAC (SUIITE)

Après moult essais de lecture et d'enregistrement sur le magnéto-cassette Panasonic, nous nous sommes fait à l'idée qu'il est inutilisable avec le DAI !

Le bureau de Dai Club France s'est adjoint l'aide de quelques bénévoles pour les envois, la reproduction des cassettes et le journal: Guillemé, Frémont, Maire, Leroux, Aussage. Toutes les personnes désirant participer à l'animation du club seront les bienvenues.

Très bonne nouvelle pour tous: la cotisation annuelle de 400F se transforme désormais en droit d'entrée au club de 250F plus une cotisation annuelle de 150F. Ce qui veut dire que les adhérents actuels n'auront que 150F à payer pour la cotisation 1983.

En réponse à de nombreuses demandes d'adhérents, les logiciels du catalogue seront disponibles sous forme de sélections à des tarifs club (encore à préciser). Etant entendu que le système d'échange continuera à fonctionner mais aussi avec un système de sélections. Ceci en raison du travail énorme nécessaire à la confection des cassettes à la demande.

Vous pourrez donc obtenir une sélection soit par échange, soit par achat au club. Des précisions seront données à la mise en place du nouveau système. Nous espérons ainsi satisfaire le plus grand nombre, tout en assurant au club une rentrée financière suffisante pour garantir la parution du journal et la location d'un local permanent.

Les adhérents qui n'ont pas reçu les revues numéro 1 ou/et 2 peuvent encore les demander, nous feront des tirages complémentaires de la n°1 si nécessaire.

Tous a vos burins, plumes, stylos ou claviers !

Adresse du collecteur
C. LEQUESNE
BOITE A IDEES DAI
LES BOIS
72450 MONTFORT LE ROTROU

Voila deja quelques questions :

QUESTION BA 1

A quoi sert l'instruction ON...GOSUB ?

REPONSE A LA QUESTION BA 1

ON...GOSUB remplace avantageusement une serie de IF...THEN
GOSUB pour certains choix multiples. (plus de 2)

Si un menu comporte 10 sous-programmes, apres l'affichage
on peut ecrire par exemple:

```
50 INPUT "VOTRE CHOIX"; CHOIX%
```

```
60 ON CHOIX% GOSUB 100,200,300,400,500,600,700,800,900,1000
```

```
70 GOTO menu
```

La ligne 60 remplacera 9 lignes IF CHOIX%=... THEN GOSUB ...
C. LEQUESNE

QUESTION BA 2

Comment simuler la fonction ONERR GOTO ?

QUESTION UT 1

Quel est le point d'entree & comment peut-on appeler la
routine TIMER ?

QUESTION UT 2

Comment simuler la fonction DELETE 100-400 (effacement de
toutes les lignes 100 a 200) ?

QUESTION UT 3

Ou se trouve le drapeau d'erreur ?

QUESTION MA 1

Comment travaille l'entree paddle ?

QUESTION MA 2

Quels sont les formats & frequences des cassettes ?

Il est possible de jouer sur ces valeurs pour faire écrire un programme par un autre:
 Par POKE, mettre en #A4, #A5, #A2, #A3 l'adresse où on veut écrire un programme. Faire POKE#131,2. 'Printez' des instructions, et finir par un CHR\$(0): Faire POKE#131,0:POKE#135,2
 Ce que l'on a écrit est alors analysé, compilé, et utilisable. Un exemple simple montrant ces possibilités est fourni dans le programme DATA#. Ce programme va lire par des PEEK le contenu de mémoires et les transforme en lignes de DATA. Ainsi lorsqu'on veut incorporer un sous-programme en langage machine à un programme en BASIC, il suffit de rentrer le sous programme en UT et le transformer avec DATA# en une suite de DATA qu'il suffira de POKER .
 Il est aussi possible d'avoir deux buffers en même temps : un en lecture et un en écriture, ce qui permet de créer un programme de renumérotation: on lit l'image d'un programme BASIC dans un buffer et on écrit simultanément le programme renuméroté dans un deuxième buffer.
 Encore une autre possibilité est d'accélérer l'exécution de graphiques ou de musiques: par exemple le tracé d'un cercle se fait ordinairement dans une boucle contenant un calcul de cosinus et sinus. Si ces valeurs peuvent être mises en DATA, les calculs trigo ne se feront que lors de la création du programme (de même pour les calculs de fréquence en musique).
 A vous de trouver d'autres utilisations!

```

LIST
1  REM AUTEUR LETAC JEAN-LUC
2  REM PROGRAMME PERMETTANT D'OBTENIR EN DATA LES
5  REM VALEURS D'UN PROGRAMME ASSEMBLEUR
10 INPUT "DEBUT DE LA MEMOIRE ";DEBZ:PRINT
20 INPUT "FIN DE LA MEMOIRE ";FINZ:PRINT
30 INPUT "PREMIERE LIGNE";IPZ:PRINT
40 POKE #131,2:POKE #A2,0:POKE #A4,0:POKE #A3,#A0:POKE #A5,#A0
41 REM CHANGER #A0 EN UNE AUTRE VALEUR POUR UN GROS PROGRAMME
50 FOR IX=DEBZ TO FINZ STEP 16
60 PRINT IPZ;"DATA";
70 FOR JZ=IX TO IX+15
80 IF JZ()IX THEN PRINT ", ";
90 KZ=PEEK(JZ)
100 PRINT "#";HEX$(KZ);
110 NEXT
120 IPZ=IPZ+1:PRINT
130 NEXT
140 PRINT CHR$(0)
150 POKE #131,0:POKE #135,2

```

INSTRUCTIONS DE TRAITEMENT DE CHAINES DE CARACTERES

.....
LEN(A\$) RIGHT\$(A\$,N)
LEFT\$(A\$,N) MID\$(A\$,D,L)
.....

Ces instructions permettent de sortir des sous-chainés de caracteres et s'avèrent très utiles dans la manipulation des données, lors de traitement de fichiers.

LEN(A\$) Donne la valeur numérique du nombre de caracteres de la chaîne A\$.

exemple: A\$="BONJOUR":LEN(A\$)=7

RIGHT\$(A\$,N) Permet de sortir les N caracteres de la chaîne A\$ en partant de la droite.

exemple: RIGHT\$(A\$,3)="OUR"

LEFT\$(A\$,N) Permet de sortir les N caracteres de la chaîne A\$ en partant de la gauche.

exemple: LEFT\$(A\$,3)="BON"

MID\$(A\$,D,L) Permet de sortir les L caracteres de la chaîne A\$ situés après le D-ième caractère.

exemple: MID\$(A\$,1,3)="NJO". Attention le premier caractère est numéroté 0 (zero).

Ces instructions permettent de manipuler les variables alphanumériques en particulier dans le fichier. Supposons que nous ayons un fichier clients comportant des zones dont:

- N\$(I)=Nom (I : numero de la fiche)
- P\$(I)=Prenom
- A\$(I)=Adresse
- C\$(I)=Code postal

Supposons que nous ayons à réaliser un programme de tri sur le fichier, par exemple sortir toutes les fiches clients habitant un département donné.

Il suffit d'explorer le début du code postal sur toutes les fiches et de les sortir si elle correspond au département sélectionné.

Ce qui donne le sous-programme suivant:

```
500 REM Sous-programme de tri par code postal
510 INPUT "Département choisi";C1$
520 FOR I=0 TO NOMBREDEFICHE-1
530 IF LEFT$(C$(I),2)=C1$ THEN 50
540 GOTO 60
550 PRINT N$(I),P$(I),A$(I),C$(I)
560 NEXT I
570 RETURN
```

L'utilitaire suivant vous permettra sans doute de decouvrir l'implantation de routines ou de mots-cles de l'interpreteur BASIC, nous l'esperons.

```

1  REM
5  REM LE CRITERE PERMET DE TROUVER UNE CHAINE PARTICULIERE
6  REM DANS LE DUMP.POUR ARRETER LE DEFFILEMENT TAPER LA BARRE
7  REM D'ESPACE.POUR CONTINUER APRES ARRET TAPER LE CURSEUR
8  REM BAS (FLECHE VERS LE BAS)
10 MODE 0:COLORT 9 13 0 0:PRINT CHR$(12)
15 PRINT TAB(10):INPUT "ADRESSES DE DEBUT,FIN";DEBZ,FINZ:DDZ=DEBZ-8:FLAGZ=0
16 PRINT TAB(10):INPUT "CRITERE DE RECHERCHE";CRI$
25 L0=LEN(CRI$)
30 DIM A$(7.0),B$(7.0):PRINT :PRINT :PRINT :PRINT
35 DDZ=DDZ+8:FOR IZ=0 TO 7
40 AZ=PEEK(DDZ+IZ)
41 A$(IZ)=HEX$(AZ)
42 IF LEN(A$(IZ))<2.0 THEN A$(IZ)="0"+A$(IZ)
43 IF AZ<=32.0 THEN AZ=32
46 B$(IZ)=CHR$(AZ)
47 C#=C#+B$(IZ):IF LEN(C#)>L0 THEN C#=RIGHT$(C#,L0)
48 IF C#=CRI$ THEN FLAGZ=1
50 NEXT IZ
60 PRINT HEX$(DDZ);" -- ";PRINT TAB(10);
80 FOR IZ=0 TO 7:PRINT A$(IZ);" ";NEXT IZ
90 PRINT TAB(40);:FOR IZ=0 TO 7:PRINT B$(IZ);:NEXT IZ
100 IF FLAGZ=1.0 THEN PRINT " <<<<";FLAGZ=0
105 PRINT
110 XZ=GETC
120 IF XZ=32 THEN FZ=1
130 IF XZ=17 THEN FZ=0
140 IF FZ=1 GOTO 110
190 IF DDZ<=FINZ GOTO 35
200 PRINT "*****"
210 GOTO 15

```

L'adresse #FF05 controle la vitesse de sortie des caracteres sur l'ecran et vers l'interface RS 232. Reglez donc vous-meme par un unique "POKE" la vitesse :

```

10  REM VARIATION DE VITESSE D'AFFICHAGE
20  VITESSE=#7E
30  PRINT "VITESSE = ";VITESSE;
40  PRINT TAB(20);"012345678901234567890123456789012345"
50  WAIT TIME 20:VITESSE=VITESSE-2.0
60  IF VITESSE<=0.0 GOTO 80
70  POKE #FF05,VITESSE:GOTO 30
80  POKE #FF05,#FF
100 END

```


Article de Jan Boerrigter dans le bulletin DAINamic 01/82

Routines de restart du DAI

1/ Le microprocesseur du DAI(8080) dispose de 8 instructions mono-octet équivalentes à des appels de sousroutines (CALL), ce sont les RST 0 à 7. Dans la plupart des micro-ordinateurs ces instructions sont utilisées pour des interruptions. Cet article décrit la façon dont le DAI utilise ces interruptions.

2/RST 1, RST 4, RST 5.

Ces 3 restart sont utilisés pour sélectionner un "bank" (bank switching). Le DAI utilise dans la zone de mémoire E000-EFFF 4 banks de 4K de ROM superposés. Les bits 6 et 7 de l'octet situé en FD06 permettent de choisir un bank parmi ces 4. Normalement c'est le bank 0 qui est en service, mais on peut activer les autres par programme. Pour cela les RST 1, RST 4, RST 5 sont utilisés. Ces instructions sont suivies par un octet de donnée. Quand le microprocesseur exécute une instruction RST le program counter (registre PC) est chargé avec l'adresse de la routine correspondante dans la zone 0000-003F (interrupt vector). Ces routines chargent le PC avec une adresse comprise dans 0062-0071 où se trouvent les instructions qui préparent la sélection de la ROM appropriée:

- RST 1: ROM bank 3 (encode, utility)
- RST 4 ROM bank 1 (math. package)
- RST 5 ROM bank 2 (screen package)

La routine générale de sélection des ROM étant à l'adr. C6CF. L'octet de donnée qui suit les instructions RST est l'adresse relative de la routine dans la ROM.

Exemple RST 5, data : 18 correspond à une routine qui est dans la ROM bank 2 à l'adresse E018 (E000+18), en l'occurrence la routine de changement de mode.

Quand on effectue une sélection de bank, l'ancienne sélection est conservée en mémoire, au retour de la routine celle-ci est restaurée.

3/Les autres instructions RST

Toutes les autres instructions RST sont utilisées pour les interruptions. Les interruptions sont générées par le circuit 5501 (Timer and Interrupt controller).

4/RST 7, Clock interrupt

Un signal de synchro tiré du signal vidéo (T=20 ms), est utilisé comme signal d'horloge. A chaque fois que ce signal est présent, le PC est mis à 0038. (adresse du RST 7). La routine charge le PC à D9A9 où se poursuivent les opérations...

On n'autorise que les interruptions de pile et on vérifie le contenu du timer en 01BE/F. A chaque fois qu'un RST 7 est effectué ce compteur est décrémenté. Il ne se passe rien tant qu'il est différent de zéro.

Quand ce compteur est à zéro alors on décrémente 01C0 à chaque fois qu'un RST 7 arrive. Encore une fois rien de particulier tant que nous ne sommes pas arrivés à zéro. Mais quand 01C0 (clock timer) est à zéro un RST 5, :12 est envoyé au microprocesseur. Cette routine (située donc dans le bank 2 en E012) fait clignoter le curseur en fonction du contenu des pointeurs situés en 0074-0077.

Après avoir changé le contenu de la position mémoire occupée par le curseur, l'ancien masque d'interruption est restauré et le programme retourne à sa séquence normale.

5/ RST 6 interruptions du clavier

A chaque fois qu'une interruption venant du timer 4 est présente, le registre PC est mis à D578 via la routine du vecteur d'interruption en 0030.

La routine recharge le timer 4 et autorise uniquement les interruptions de pile et les interruptions d'horloge.

Le compteur clavier (01C1) est décrémenté à chaque interruption RST 6. Quand le résultat est différent de zéro la routine est avortée. Sinon le compteur est rechargé et une scrutation du clavier est effectuée (routine GETC).

En sortie le masque d'interruption d'origine est restauré.

6/ RST 3, SOUND interrupt (int. son)

A une interruption du timer 3 la routine du vecteur d'interruption en 0018 charge D755 dans le registre PC.

Le RST 3 autorise les interruptions horloge et son uniquement. Le timer 3 est rechargé et le bank 1 est sélectionné.

Puis le programme continue en EE6E (bank 1) qui est le programme SOUND

A la sortie l'ancien masque d'interruption et l'ancien bank est encore une fois restauré.

7/ RST 2 Interruption de pile

Quand il y a un débordement de pile (stack overflow), une interruption RST 2 en résulte. Via l'adresse du vecteur d'interruption en 0010 le registre PC est mis à D9E2.

Le RST 2 remet à zéro le pointeur de pile en F900. Les entrées en cours et l'encodage des lignes stockées sont suspendus.

les timers son et clavier sont rechargés. Un message d'erreur "STACK OVERFLOW" est alors imprimé.

8/ RST 0 Utility

Le RST 0 est utilisé uniquement par l'Utility. Après le RST 0 le PC est mis à 0000.

l'adresse nécessaire à cette routine n'existe qu'après les commandes Z2 ou Z3 de l'Utility. Alors les octets 0062/63 sont mis à EB5D, adresse de départ de la routine RST 0 dans le bank 3.

Le RST 0 est provoqué par le timer 0, et est utilisé dans la commande LOOK de l'Utility.

Au moment d'un RST 0 tous les registres du microprocesseur sont sauvés dans la zone 0053-005E. Ensuite le programme continue à l'adresse qui est donnée par la commande LOOK et indique la prochaine instruction à effectuer.

Le programme vérifie cette instruction. Si c'est un CALL ou RST alors la nouvelle adresse est également sauvée. Puis une vérification est faite pour voir si l'adresse de la prochaine instruction est dans la fenêtre définie par LOOK. Quand c'est le cas, le contenu de tous les registres, y compris le pointeur de pile, les registres PC et PSW sont visualisés sur l'écran.

En sortie, le timer 0 est rechargé, le masque d'interruption mis et parmi d'autres instructions les registres sont de nouveau restaurés.

Parce que le programme est exécuté avec les interruptions RST 0, il est beaucoup plus lent qu'en exécution normale.

Jan Boerrigter Octobre 81

Traduit (à peu près) par C. Aussage

ROUTINES MATHÉMATIQUES DU DAI

A

LES DIVERSES ROUTINES MATHÉMATIQUES SONT APPELÉES DE LA FAÇON USUELLE:

RST :4

DATA :XX

LES VALEURS DE L'OCTET DE DONNÉE ET LES FONCTIONS APPELÉES SONT LISTÉES CI-DESSOUS. LES VALEURS SONT VALIDES AUSSI BIEN AVEC QUE SANS LE MATH-CHIP (HARDWARE MATH MODULE). LA ROUTINE DE RESTART SÉLECTIONNE AUTOMATIQUÉMENT LES POINTS D'ENTRÉE CORRECTS. LES ADRESSES RÉELLES DES ROUTINES SONT ÉGALEMENT INDICUÉES DANS LE TABLEAU.

DATA	CALL	CALL+	FONCTION
====	=====	=====	=====
:00	EDAA	E474	ADDITION FLOTTANTE (X) + (Y)
:03	EDB4	E479	SOUSTRACTION FLOTTANTE (X) - (Y)
:06	EOFE	E47E	MULTIPLICATION FLOTTANTE (X) * (Y)
:09	E108	E483	DIVISION FLOTTANTE (X) / (Y)
:0C	E112	E588	ENTRÉE ARGUMENT PAR ADRESSE DANS (HL)
:0F	E11C	E599	SORTIE RESULTAT, ADRESSE DANS (HL)
:12	E126	E55F	ENTRÉE ARGUMENT PAR REGISTRES A,B,C,D
:15	E133	E56F	SORTIE DU RESULTAT PAR REG. A,B,C,D
:18	E140	E488	ABS(X)
:1B	E14A	E493	CHANGEMENT DE SIGNE (X)*(-1)
:1E	E443	E498	PARTIE ENTIÈRE FLOTTANTE INT(X)
:21	E154	E4A0	PARTIE FRACTIONNAIRE FLOTTANTE (X)
:24	E855	EDA1	(X) ^ (Y)
:27	E745	E4B1	LN(X)
:2A	E667	E4B6	EXP(X) e^(X)
:2D	E870	E4BB	LOGT(X)
:30	E880	E4C0	ALOG(X)
:33	E5F8	E4CC	SQR(X)
:36	E7D2	E4D1	SIN(X)
:39	37D9	E4D6	COS(X)
:3C	EB94	E4DB	TAN(X)
:3F	E96C	E4E0	ASIN(X)
:42	E9C1	E4E5	ACOS(X)
:45	EBAC	E4EA	ATAN(X)
:48	E414	E4EF	CONVERSION DE FLOTTANT EN ENTIER
:4B	E3DE	E498	CONVERSION D'ENTIER EN FLOTTANT
:4E	E16D	E4F4	ADDITION ENTIÈRE (X) + (Y)
:51	E18D	E4F9	SOUSTRACTION ENTIÈRE (X) - (Y)
:54	E1AC	E4FE	MULTIPLICATION ENTIÈRE (X) * (Y)
:57	E22B	E503	DIVISION ENTIÈRE (X) / (Y)
:5A	E238	E508	RESTE DE LA DIVISION ENTIÈRE (X) / (Y)
:5D	E30B	E517	ABS(X) (ENTIER)
:60	E315	E522	CHANGEMENT DE SIGNE ENTIER (X) * (-1)
:63	E32E	ED19	(X) IAND (Y)
:66	E345	ED26	(X) IOR (Y)
:69	E35C	ED33	(X) IXOR (Y)
:6C	E373	ED43	COMPLEMENT (X)
:6F	E3A5	ED6C	DECALAGE DE (X) A GAUCHE DE (Y) PLACES
:72	E39B	ED55	DECALAGE DE (X) A DROITE DE (Y) PLACES
=====	=====	=====	=====

UTILISATION DES ROUTINES MATHÉMATIQUES

Avant d'appeler une des fonctions ci-dessus le premier opérande doit être chargé dans l'accumulateur arithmétique. Pour cela deux routines sont disponibles:

1/ Le nombre (flottant ou entier) constitué de 4 octets est chargé dans les registres A-B-C-D (A= octet de poids fort).

```
RST :4
DATA :12      Entrée par registres
```

2/ L'adresse du nombre est chargée dans le registre (HL) (HL).

```
LXI H,NUMBER
RST :4
DATA :0C      Entrée par adresse
```

Le second opérande (si nécessaire) est toujours passé à la routine en chargeant son adresse dans (HL) d'abord.

```
LXI H,NUMBR2
RST :4
DATA :4E      Addition entière
```

Après la liste d'opérations ci-dessus, NUMBR2 est additionné à NUMBER. Le résultat reste dans l'accumulateur arithmétique, nous pouvons enchaîner d'autres opérations avec ce résultat comme nouvel opérande ou bien sortir le résultat.

Pour obtenir le résultat deux routines sont utilisées:

```
1/ RST :4
    DATA :15      Résultat dans les registres
                        A-B-C-D
```

```
2/ RST :4
    DATA :0F      Charge le résultat dans la
                        mémoire adressée par (HL).
```

Voici un exemple d'utilisation:

```
START LXI H,INMBR1      NOMBRE ENTIER
      RST :4
      DATA :0C          ENTREE DU NOMBRE 1
      RST :4
      DATA :4B          CONVERSION ENTIER-FLOTTANT
      RST :4
      DATA :33          RACINE CARREE
      LXI H,RESLT1
      RST :4
      DATA :0F          RESULTAT DANS RESLT1
      LXI H,FNMBR
      RST :4
      DATA :00          ADDITIONNER FNMBR AU RESULTAT PRECEDENT
      RST :4
      DATA :15          RESULTAT DANS A-B-C-D
      ....
      ETC...
```

* Attention à ne pas confondre l'accumulateur arithmétique avec l'accumulateur (registre A): voir le manuel pour la zone de travail arithmétique.

Ces 3 courts programmes montrent par un exemple d'animation graphique haute resolution comment se fait le déplacement des données en mémoire : Une sous-routine en langage machine en optimise l'effet.

```

10  MODE 6:COLORG 1 3 5 14
20  FILL 60,90 200,130 14
22  FILL 120,131 140,255 5:FOR YZ=131 TO 255 STEP 5
24  DRAW 120,YZ 140,YZ 3:NEXT YZ
25  FILL 260,60 290,255 3:FOR YZ=60 TO 255 STEP 5
27  DRAW 260,YZ 290,YZ 14:NEXT YZ
30  KX=0
35  FOR IX=#BF EF TO #9CC7 STEP -90
40  POKE IX,#20+KX
50  NEXT IX
90  WAIT TIME 5:KX=1-KX
100 GOTO 35

10  MODE 6:COLORG 1 3 5 14
20  FILL 60,90 200,130 14:FILL 110,30 150,90 14
22  FILL 120,131 140,240 5:FOR YZ=131 TO 240 STEP 5
24  DRAW 120,YZ 140,YZ 3:NEXT YZ
25  FILL 260,30 290,240 3:FOR YZ=30 TO 240 STEP 5
27  DRAW 260,YZ 290,YZ 14:NEXT YZ
30  FOR IX=#2 TO #F
40  POKE #9CC7,#20+IX
45  POKE #799F,#2F-IX
50  WAIT TIME 2:NEXT IX
60  FOR IX=#F TO #1 STEP -1
70  POKE #9CC7,#20+IX
75  POKE #799F,#2F-IX
80  WAIT TIME 2:NEXT IX
100 GOTO 30

40  REM *****
50  REM PROGRAMME PERMUTATION EN MODE 6A
60  REM ATTENTION CE PROG. UTILISE LA ZONE
70  REM #300 A #346 POUR UNE ROUTINE ET LA ZONE
80  REM #400 A #409 COMME TAMPON DE DONNEES.
90  REM *****
100 MODE 6A:COLORG 1 3 5 14
110 FOR IX=0 TO 200 STEP 4
120 DRAW 0,IX 330,IX 14:NEXT IX
140 FILL 10,100 130,120 3
150 FOR RX=0 TO 70
160 READ ZX:POKE #300+RX,ZX
170 NEXT RX
200 FOR IX=1 TO 200:CALLM #300:NEXT IX
210 REM LES ADRESSES #305 #306 >> HAUT DE LA ZONE
220 REM #308,#309 >>TAMPON* #318 >>NB DE LIGNES A PERMUTER
230 REM #30B,#31A,#336 >> NB D'OCTETS PERMUTES PAR LIGNE
240 REM NB DE PTS=NB D'OCTETS*4
250 DATA #F5,#E5,#D5,#C5,#21,#F3,#A7,#11,#00,#04,#0E,#0A
260 DATA #7E,#12,#2B,#13,#0D,#C2,#0C,#03,#21,#99,#A7,#06
270 DATA #63,#0E,#0A,#7E,#11,#5A,#00,#19,#77,#11,#A5,#FF
280 DATA #19,#0D,#C2,#1B,#03,#11,#B0,#FF,#19,#05,#C2,#19
290 DATA #03,#11,#5A,#00,#19,#0E,#0A,#11,#00,#04,#1A,#77
300 DATA #13,#2B,#0D,#C2,#3A,#03,#C1,#D1,#E1,#F1,#C9

```

L'ASSEMBLEUR TROISIEME PARTIE

Suivant la critique generale des membres du CLUB, cet article sera cette fois -ci plus concis et plus simple que d'habitude, afin de garder des proportions raisonnables vis à vis de la partie "BASIC" de la revue, et de s'adresser meme à ceux qui debutent l'assembleur.

L'article repond aujourd'hui à 3 questions totalement independantes :

- 1 - comment faire bouger un point avec les paddles à toute vitesse ?
- 2 - comment reduire d'un facteur 10 le temps d'execution d'un tri ?
- 3 - comment se servir des instructions PUSH et POP avec les registres ?

```

001          *          *****
002          *          *   ESSAI PADDLES   *
003          *          ***** par C. FREMONT
004          *
005          *  DEPLACEMENT D'UN POINT SUR ECRAN
006          *  AVEC LES PADDLES .
007          *
008          *
009          *          ORG   :5000          ADRESSE D'IMPLANTATION
010 5000 E5          PUSH  H
011 5001 C5          PUSH  B
012 5002 D5          PUSH  D
013 5003 F5          PUSH  PSW
014 5004 3E00          ROUT  MVI   A,0          PADDLE 0 => A
015 5006 CDC6EB          CALL  :EBC6          ADRESSE ROUTINE PADDLE
016 5009 3AD800          LDA   :D8          ADRESSE LECTURE VALEUR
017 500C 323A50          STA   VAL1          VALEUR PADDLE 0 => VAL1
018 500F 3E02          MVI   A,2          PADDLE 2 => A
019 5011 CDC6EB          CALL  :EBC6
020 5014 3AD800          LDA   :D8
021 5017 323B50          STA   VAL2          VALEUR PADDLE 2 => VAL2
022 501A 3A3A50          LDA   VAL1
023 501D 6F           MOV   L,A          VAL1          => L
024 501E 2600          MVI   H,00          00          => H
025 5020 3A3B50          LDA   VAL2
026 5023 4F           MOV   C,A          VAL2          => C
027 5024 3E0F          MVI   A,15          COULEUR 15 => A
028 5026 C32B50          JMP   AFF          SAUT A AFF (AFFiche)
029 5029 3E00          COUL  MVI   A,0          COULEUR 0 => A
030 502B EF           AFF   RST   5          AFFICHAGE ECRAN
031 502C 1E           DATA :1E
032 502D FE0F          CPI   15          TEST COULEUR :SI A=15
033 502F CA2950          JZ   COUL          SAUT A COUL
034 5032 C30450          JMP   ROUT          SI NON RETOUR A ROUT
035 5035 F1           POP   PSW
036 5036 D1           POP   D
037 5037 C1           POP   B
038 5038 E1           POP   H
039 5039 C9           RET
040 503A 00          VAL1 ***          RESERVE PADDLE 0
041 503B 00          VAL2 ***          RESERVE PADDLE 2
042 503C          END

```

Pour faire tourner ce programme, tapez **MODE5:CALLM#5000** apres assemblage.
OK, mais comment fonctionne-t-il ?

La routine :ERC6 donne la valeur correspondant aux numeros de paddle 0 et 2 (voir revue no.2), valeurs qui sont stockees dans des emplacements reserves, puis dans les registres H L et C. Il ne reste plus qu'à appeler la ROM avec RST 5 et DATA :1E pour dessiner le point et l'effacer juste apres. Le programme est tres simple : la partie ROUT calcule la position, AFF l'allume et l'eteint.

Voici maintenant un exemple d'application d'une sous-routine qu'on a interet à traduire en langage machine, parce que sa fonction est limitee et d'une longueur monotone en BASIC : C'est le tri par selection de 128 octets prealablement tires au hasard et places en memoire.

```

R29 REM Tirage de #80 octets au hasard
30 FOR I=#7000 TO #70FF:POKE I,RND(256):NEXT
31 NOISE 0 15:WAIT TIME 10:NOISE OFF
99 REM Tri Par selection des octets
100 FOR I=#7000 TO #70FE:P0=I:PE=PEEK(P0)
101 FOR J=I+1 TO #70FF:IF PEEK(J)<PE THEN P0=J:PE=PEEK(P0)
102 NEXT:POKE P0,PEEK(I):POKE I,PE:NEXT
109 REM Affichage des octets tries
110 NOISE 0 15:WAIT TIME 10:NOISE OFF
111 FOR I=#7000 TO #70FF:PRINT PEEK(I),:NEXT
  
```

Remarquez qu'entre les 2 NOISE il s'ecoule plus de vingt secondes à nos chronos. Et maintenant, si l'on reduisait le BASIC à ceci...

```

30 FOR I=#7000 TO #70FF:POKE I,RND(256):NEXT
100 NOISE 0 15:WAIT TIME 10:NOISE OFF
101 CALLM #400
102 NOISE 0 15:WAIT TIME 10:NOISE OFF
110 FOR I=#7000 TO #70FF:PRINT PEEK(I),:NEXT
  
```

...voilà ce qu'il conviendrait d'assembler à partir de l'adresse #400 :
Ajoutez-y les PUSH et POP habituels, bien sur, et le tri par selection se ramene à moins de 2 secondes.

0001	SELECT	LXI	H, :7000	[Adresse de depart
0002	SELEC1	PUSH	H	[HL↑ = I
0003		MOV	D,H	[DE = P0
0004		MOV	E,L	[
0005		MOV	B,M	[C = PEEK(I)
0006		MOV	C,B	[B = PE
0007		INX	H	[HL = J = I+1
0008	SELEC2	MOV	A,M	[A = PEEK(J)
0009		CMP	B	[IF A>B...
0010		JNC	SELEC3	[...THEN SELEC3
0011		MOV	D,H	[P0 = J
0012		MOV	E,L	[
0013		MOV	B,M	[PE = PEEK(J)
0014	SELEC3	INX	H	[J = J + 1
0015		MOV	A,L	[IF HL < #7100...
0016		CPI	0	[...
0017		JNZ	SELEC2	[...
0018		MOV	A,H	[...
0019		CPI	:71	[...
0020		JNZ	SELEC2	[...THEN NEXT J
0021		MOV	A,C	[A = PEEK(I)
0022		STAX	D	[POKE P0,A
0023		POP	H	[HL = I (J efface)
0024		MOV	M,B	[POKE I,PE
0025		INX	H	[I = I + 1
0026		MOV	A,L	[IF I < #70FF...
0027		CPI	:FF	[...
0028		JNZ	SELEC1	[...
0029		MOV	A,H	[...
0030		CPI	:70	[...
0031		JNZ	SELEC1	[...THEN NEXT I

LA PARTIE ASSEMBLEUR

Vous avez noté que le BASIC a été traduit mot à mot ou presque, ce qui me dispense de longs commentaires.

On augmente le nombre d'éléments en changeant l'adresse de fin (lignes 16 et 19) ou celle de début (ligne 1).

N'en restez pas là ! Suivant cette technique, réalisez vous-même le tri ultra-rapide de SHELL-METZNER... Et nous le ferons volontiers paraître !!

LES INSTRUCTIONS PUSH ET POP

On est généralement méfiant envers ces 2 mnémoniques qui peuvent aisément causer un magnifique "stack overflow", pourvu qu'on en oublie un ou qu'on en mette un de trop. On sait pourtant qu'ils sont nécessaires à la bonne marche d'une routine machine à l'intérieur d'un programme BASIC.

LES REGISTRES CONCERNES

Le micro-processeur du 8080 possède une paire de registres SP appelée pointeur de pile (stack pointeur), où la pile est un entassement de paires d'octets en mémoire vive (les 16 bits du registre pointeur peuvent adresser toute la

PRINCIPE DE FONCTIONNEMENT

PUSH sauvegarde une paire de registres dans la pile :

- 1 - les 8 bits de poids forts sont rangés à l'adresse mémoire immédiatement inférieure au pointeur de pile.
- 2 - les 8 bits de poids faibles sont rangés à l'adresse avant la précédente.
- 3 - le pointeur de pile est automatiquement décrémente de 2 ($SP=SP-2$).

Exemple : B=:6A C=:30 SP=:13A4

PUSH B

B=:6A C=:30 SP=:13A2 et :13a3 contient :6a
et :13a4 contient :30

POP transfère 2 zones mémoires de la pile vers la paire de registre concernée. L'opération est exactement l'inverse de PUSH.

J. LEROUX

Voici d'elegantes courbes qui interesseront certainement les amateurs de mathematiques et de physique relativiste. Les deux programmes suivants dessinent sur votre ecran des representations tridimensionnelles d'espaces-temps tels qu'EINSTEIN les a imaginees il y a 66 ans.

Les graphiques se dessinent assez rapidement (2 mn) ; vous pouvez aussi modifier, dans le second programme, la position de l'observateur fictif (ligne 60), et meme la profondeur et la largeur du "trou" en substituant votre equation personnelle en ligne 600.

J.LEROUX

```

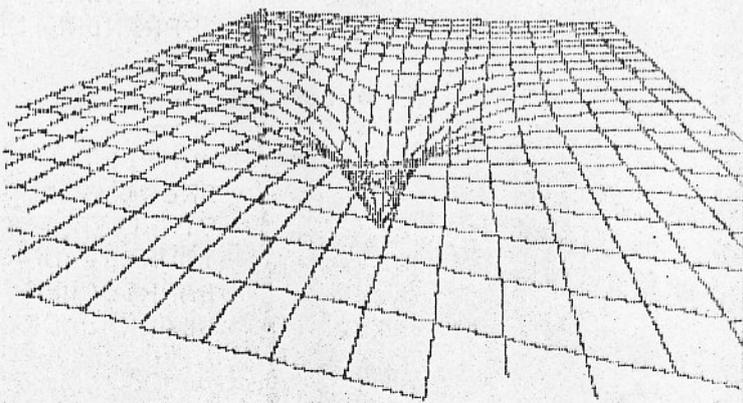
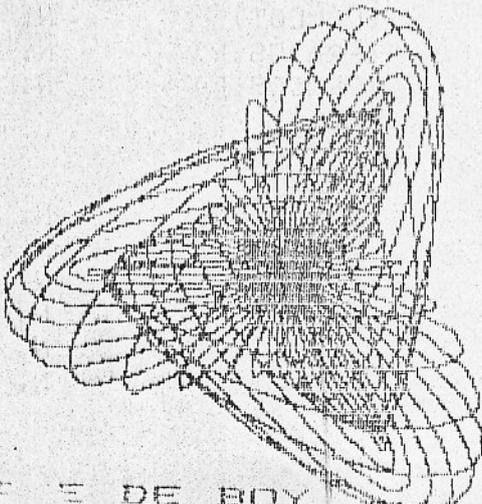
1 CLEAR 100:COLORG 0 13 0 0:MODE 6
50 REM SURFACE DE BOY
100 FOR MU=0.0 TO PI STEP 0.1
101 D=10.0+1.41*SIN(6.0*MU-PI/3.0)
102 E=1.98*SIN(3.0*MU-PI/6.0):A=D+E:B=D-E
103 AL=SIN(3.0*MU)*PI/8.0:SAL=SIN(AL):CAL=COS(AL)
104 C1=A*A-B*B:C2=SQR(A*A+B*B):CM=COS(MU):SM=SIN(MU)
105 X1=C1/C2+A:Z1=C2+A:SA=Z1*SAL:Z=Z1*CAL
106 X0=X1*CM-SA*SM:Y0=X1*SM+SA*CM
110 FOR TE=0.2 TO 6.4 STEP 0.2:CT=A*COS(TE)
111 ST=B*SIN(TE):X1=C1/C2+CT-ST:Z1=C2+CT+ST:SA=Z1*SAL
112 X=X1*CM-SA*SM:Y=X1*SM+SA*CM:Z=Z1*CAL
113 DRAW 168+5*X0,110+5*Y0 168+5*X,110+5*Y 13
114 X0=X:Y0=Y:NEXT:NEXT MU:WAIT TIME 1000

```

```

1 REM ESPACE-TEMPS RELATIVISTE
50 PRINT "20,5,20 = NORMAL"
51 PRINT "40,5,20 = VISION ELOIGNEE"
52 PRINT "20,5,10 = VU DE LA SURFACE"
60 INPUT "CHOIX DU POINT DE VUE";L,M,N
110 COLORG 0 13 0 0:MODE 6A
150 S=L*L+M*M:R=SQR(S):S=S+N*N:Q=SQR(S):S=L*L+M*M:T=S+N*N
200 FOR U=-20.0 TO 20.0 STEP 2.0:FOR V=-20.0 TO 20.0
215 IF V=(-20.0) THEN RR=1.0
217 GOSUB 600:GOSUB 520:NEXT V:NEXT U
300 FOR V=-20.0 TO 20.0 STEP 2.0:FOR U=-20.0 TO 20.0
320 IF U=(-20.0) THEN RR=1.0
330 GOSUB 600:GOSUB 520:NEXT U:NEXT V:END
520 H=T-U*L-V*M-W*N:C=T*(V*L-U*M)*4.0/(R*H)+128.0
540 D=96.0+3.0*Q*(W*S-N*(U*L+V*M))/(R*H)
545 IF D<0.0 OR D>255.0 OR C<0.0 OR C>335.0 THEN RETURN
550 IF RR=1.0 THEN DOT C,D 13:RR=0.0:GOTO 570
560 DRAW A,B C,D 13
570 A=C:B=D:RETURN
600 W=300.0/(10.0+U*U+V*V)-10.0:W=-W:RETURN

```



SURFACE DE BOY

DEFORMATION RELATIVISTE

DAI CLUB FRANCE

CATALOGUE

PROGRAMMES DISPONIBLES POUR VOS ECHANGES .

N'oubliez pas de signer vos programmes , de les titrer , et d'expliquer en quelques lignes les regles ou le mode d'emploi .
Estimez leur valeur (1 a 5 etoiles) dans leur categorie , et indiquez nous le nom du ou des programmes que vous desirez en echange .

Demonstration :

**	VOITURE MUSIQUE	(.....)	N. 7
*	DEMO MACHINE	(.....)	N. 17
*	PAYSAGE	(.....)	N. 20
*	VERRE GALACTIQUE	(.....)	N. 31
**	128 COULEURS	(.....)	N. 42
*	BUNNY	(.....)	N. 45
**	ETINCELLE-KALEIDOSCOPE	(.....)	N. 62
**	BLUENOSE	(R.MARCEL)	N. 67

Graphisme :

***	BIDULES OPTIQUES	(.....)	N. 13
**	CHAPEAU	(.....)	N. 30
*	BLACK GIRL	(.....)	N. 33
**	TEXTE GRAPHIQUE	(.....)	N. 46
*	LISSAJOU	(.....)	N. 47
**	DEMO GRAPHIQUE	(MINGASSON)	N. 57
****	NAVETTE SPATIALE	(MOENS)	N. 63
**	FRANCE-SINUS-PERSONNE	(.....)	N. 72

Utilitaire :

**	PENDULE	(.....)	N. 9
*	DUMP	(Aussage)	N. 19
*	TALK	(.....)	N. 32
**	DAI HORLOGE	(.....)	N. 44
**	EXEMPLES UTILES	(DAI Clabe)	N. 51
**	SUM-SUM A	(.....)	N. 52
***	GRAPHIQUE ELECTRONIQUE	(.....)	N. 53
***	ZOOM	(.....)	N. 64
*	TAILLE CARACTERES	(.....)	N. 73
**	DESSINEZ-VOUS ?	(VOLLOT)	N. 77
***	PROF MOMO	(PRADOS.P)	N. 87
***	APPRENDRE LE MORSE	(PRADOS.P)	N. 89

Jeux avec clavier :

**	MORPION	(.....)	N. 8
**	BREAK OUT	(.....)	N. 10
**	OTHELLO	(.....)	N. 11
**	TOUR de HANOI	(.....)	N. 12
*	BARRICADES	(.....)	N. 14
****	BLACK JACK	(Aussage)	N. 18

suite...

***	JEU DES CHEVRES	(MAIRE)	N. 23
**	CONTROLE REFLEXE	(DAI Clabe)	N. 26
*	SIMON	(DAI Clabe)	N. 28
**	CHECKER'S DAME	(.....)	N. 29
*	ROULETTE RUSSE	(.....)	N. 34
***	SOLITAIRE	(.....)	N. 36
**	LE COMPTE EST BON	(.....)	N. 37
**	DONJON-DRAGON	(.....)	N. 38
****	OTHELLO II	(VINCENT)	N. 39
**	MATTIX	(VINCENT)	N. 41
***	MASTERMIND	(MARY)	N. 43
***	BATAILLE SOUS LA MER	(.....)	N. 49
**	POKER	(LEROUX)	N. 54
***	LABYRINTHE	(LEROUX)	N. 55
*	PSYCHO	(MINGASSON)	N. 60
***	JEU DES 'L'	(WEISS)	N. 61
****	DAI CHESS	(H.BAKKER)	N. 65
***	ANDROMEDE	(.....)	N. 66
***	CARRE MAGIQUE	(C.POELS)	N. 68
**	QUATRE EN LIGNE	(.....)	N. 70
**	CHASSEZ LES TROLLS	(C.FARMENTIER)	N. 74
***	OCTRIX	(VINCENT)	N. 79
***	RUBIK'S CUBE	(PRADOS.F)	N. 82
**	MACHINE A SOUS	(PRADOS.F)	N. 83
*	BONJOUR	(PRADOS.F)	N. 84
**	STARS	(PRADOS.F)	N. 85
**	PUISSANCE 4	(PRADOS.F)	N. 86
***	TESTAMENT	(PRADOS.F)	N. 88
**	COURSE DE VOITURE	(MOENS)	N. 90
***	MISSIONNAIRES	(MOENS)	N. 91
*	GRUE + CHRONO	(MOENS)	N. 92
**	SURVIE GALACTIQUE	(MOENS)	N. 93
**	INDIANAPOLIS	(MOENS)	N. 94
**	BATAILLE NAVALE	(MOENS)	N. 95
**	ENTREPRISE	(MOENS)	N. 96
**	DEFIMATH	(MOENS)	N. 97
***	TIC TAC TOE	(MOENS)	N. 98
***	FENDU	(MOENS)	N. 99
**	LABYRINTHE GRENADE	(MOENS)	N. 100

Jeux avec paddles :

***	FRUSTRATION	(.....)	N. 15
**	ISOLA	(.....)	N. 16
**	CHASSE AU PIGEON	(.....)	N. 21
**	COW BOY	(Microtel EVRY)	N. 22
**	ALUNISSAGE	(.....)	N. 24
***	TIR AU PISTOLET	(Clabe)	N. 25
**	ATOMIC	(.....)	N. 27
***	OBSTACLE	(R.MARCEL)	N. 71
**	METEORITES	(F.LAGARDE)	N. 75
**	LUNAR MODULE	(VINCENT)	N. 76
**	CASSE BRIQUES	(VINCENT)	N. 78
***	OBSTACLE-CHASE	(VINCENT)	N. 80

Musique :

****	MARCHE TURQUE	(Pascal & Bubu)	N. 1
**	PRELUDE I de J.S.BACH	(.....)	N. 2

suite...

**	PRELUDE II de J.S.BACH	(.....)	N. 3
****	PIECE de BARRIOS	(R.SIP)	N. 4
**	MENUET J.S.BACH	(.....)	N. 5
**	MENUET de BEETHOVEN	(.....)	N. 6
*	TUBULAR BELLS	(.....)	N. 35
**	SOLFEGGIETTO	(VINCENT)	N. 40
*	SIRENE	(.....)	N. 48
***	RAG 12 TH	(TONYTEE)	N. 50
***	ETUDE N.5 de SOR	(MINGASSON)	N. 56
***	OXYGENE	(MINGASSON)	N. 58
***	EMBARRASSMENT	(MINGASSON)	N. 59
***	ORGUE ELECTRONIQUE	(.....)	N. 69
**	PRELUDE VI de J.S.BACH	(VINCENT)	N. 81

— NOTATION DES PROGRAMMES —

La notation actuelle des programmes presente de nombreuses anomalies .
L'attribution des etoiles est faite en fonction de la valeur globale
du programme .

Afin de palier a cette injustice la notation sera revue en tenant
compte :

1	- de la PRESENTATION	/	20
2	- de l'EXPLICATION (Emploi, Regles)	/	20
3	- de l'utilisation GRAPHIQUE	/	20
4	- de l'utilisation SONORE	/	20
5	- de l'ALGORITHME utilise	/	20

Cette notation sera faite par trois personnes pour chaque categorie
(demonstration , utilitaire) .

Une moyenne sur 20 sera ainsi definie et donnera :

de	1 a 4	==>	*
	5 a 8	==>	**
	9 a 12	==>	***
	13 a 16	==>	****
	17 a 20	==>	*****

La nouvelle notation des differents programmes du catalogue est
actuellement en cours , et paraitra dans notre prochain numero .

Les echanges marchent bien , nous venons de depasser le N.100.

Trois nouvelles categories devraient pouvoir rapidement entrer
au catalogue :

- a) EDUCATIF
- b) DOMESTIQUE
- c) PROFESSIONNEL

