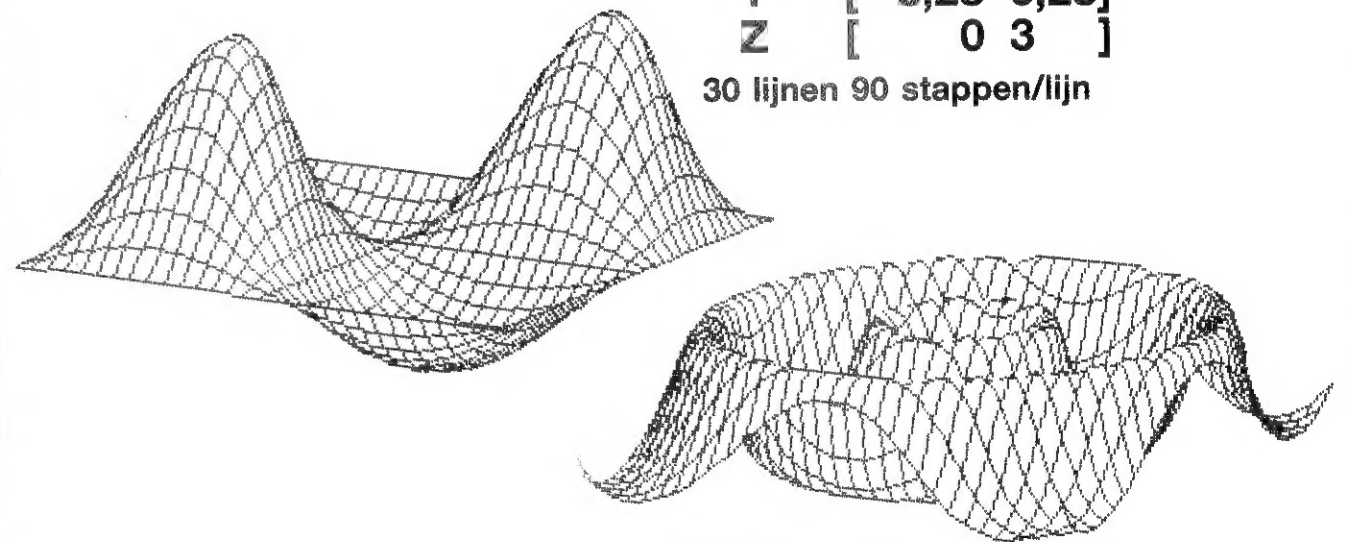


**DAI**  
**NAMIC**

34

**DAI**  
**ELIC**

7



$$Z = \text{EXP}(-\text{SIN}(X) \star \text{SIN}(Y))$$

$$X \quad [-6,28 \quad 6,28]$$

$$Y \quad [-6,28 \quad 6,28]$$

$$Z \quad [ \quad 0 \quad 3 \quad ]$$

30 lijnen 90 stappen/lijn

$$Z = \text{EXP}(-\text{COS}(X \star X + Y \star Y))$$

tweemaandelijks tijdschrift

mèi-juni 86

bimestriel

mai-juin 86

een uitgave van DAInamic VZW en IDC ASBL  
une publication de DAInamic VZM et IDC ASBL  
verantw. uitgever : w. hermans, mottaart 20, 3170 herselt

*International*

## COLOFON

DAInamic verschijnt tweemaandelijks.  
Abonnementsprijs is inbegrepen in de jaarlijkse  
contributie.  
Bij toetreding worden de verschenen nummers van de  
jaargang toegezonden.

### DAInamic redactie :

Dirk Bonné	wdw
Freddy De Raedt	Herman Bellekens
Wilfried Hermans	Frans Couwberghs
René Rens	Guido Goyvaerts
Bruno Van Rompaey	Daniël Goyvaerts
Jef Verwimp	Frank Drujff
Cedric Dufour	Willy Coremans

Vormgeving : Ludo Van Mechelen.

U wordt lid door storting van de contributie op het  
rekeningnr. **230-0045353-74** van de **Generale  
Bankmaatschappij, Leuven**, via bankinstelling of  
postgiro.

Het abonnement loopt van januari tot december.

DAInamic verschijnt de pare maanden.  
Bijdragen zijn steeds welkom.

### CORRESPONDENTIE ADRESSEN.

#### Redactie en software bibliotheek

Wilfried Hermans	
Mottaart 20	Kredietbank Herselt
3170 Herselt	nr. 401-1009701-46
Tel. 014/54 59 74	BTW : 420.840.834

#### Lidgelden / Subscriptions

Bruno Van Rompaey	Generale
Bovenbosstraat 4	Bankmaatschappij
B-3044 Haasrode	Leuven
België	nr. 230-0045353-74
tel. : 016/46.10.85	

#### Voor Nederland :

GIRO : 4083817  
t.n.v. J.F. van Dunne'  
Hoflaan 70  
3062 JJ ROTTERDAM  
Tel. : (010) 144802

#### Inzendingen : Games & Strategy

Frank Drujff  
's Gravendijkwal 5A  
NL 3021 EA Rotterdam  
Nederland  
tel. : 010/25.42.75

## DAICLIC INFOS :

DAICLIC paraît tous les deux mois.  
L'abonnement est compris dans la cotisation annuelle  
à IDC (du 1/1 au 31/12). A l'inscription, les numéros  
déjà parus dans l'année sont envoyés.

### Conseil d'administration de IDC :

Président : Christian POELS, rue des Bas-Sarts 10,  
B-4100 SERAING Tél. : 041/37.16.06  
Secrétaire : Marc VANDEMEERSCH, av. Vert Bocage 17  
B-1410 WATERLOO

Tél. : 02/354.13.63  
Trésorier : Fabrice DULUINS, allée Tour Renard 4,  
B-1400 NIVELLES Tél. : 067/21.82.10

Rédaction : Christian POELS  
Soumissions logiciels : Marc VANDERMEERSCH  
Inscriptions, vente logiciels : Fabrice DULUINS.  
(mode de paiement : voir ci-dessous)

### Cotisations :

Belgique : 1000 FB virement, chèque, cash, ...  
Compte BBL : 371-0356842-45.  
F. DULUINS et CH. POELS  
ALLEE DE LA  
TOUR RENARD, 4,  
1400 NIVELLES

Etranger : 1100 FB par mandat postal international  
uniquement.

### Services télématiques IDC :

MN2 Bruxelles-A : 02/242.70.08  
MN2 Liège-A : 041/79.66.66  
MN2 Paris-A : 1/39.71.82.91

### Branches Régionales :

IDC BORDEAUX : Bruno Delannay, Rés. Acacias B+ B3,  
Av. de Saige, F-33600 Pessac  
IDC BRUXELLES : Jacques Moens, Clos Fontaine Ducs 6,  
B-1310 La Hulpe  
IDC CHARLEROI : Etienne Szigetvari, R. Provinciale 7,  
B-1361 Clabecq  
IDC LIEGE : Philippe Rasquin, Rue Saivelette 89,  
B-4510 Saive  
IDC PARIS : Philippe Casier, Rue de Paris 31ter,  
F-92190 Meudon

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.

# INHOUD - SOMMAIRE

## DAINAMIC 34 DAICLIC 7

- 1 INHOUD-SOMMAIRE
- 2 DIGITALE FUNCTIES
- 8 STERBLOKKEN
- 10 PROGRAMMING IN ASSEMBLY LANGUAGE 4
- 15 TANDBERG TCCR-530
- 19 IMLEM SPL TRANSLATED
- 26 PUISSANCE 5
- 30 CIRKELSCHIJFSEGMENT
- 31 MARSGRAFIEKEN
- 33 EDITO
- 34 EUROPEAN MICRO CLUB
- 35 IDC BORDEAUX NEWS
- 36 IDC LIEGE NEWS
- 37 DCA NEWS
- 39 HELLO DAI'S SPEAKING
- 48 DAI ET MINITEL 2
- 51 DAI QUI RIT
- 52 LES MAUX DU DAI-LE CLAVIER
- 57 INTERRUPTIONS ET COMMUNICATIONS
- 63 BASIC - CARACTERES COULEUR
- 63 CASSETTES AUDIO RAPIDES
- 64 AMD 9511 - BUG
- 64 CARTE CENTRONICS - ERRATA

REDACTION  
J. VAN OOL  
F. VAN AMERONGEN  
C. W. READ  
I. BROEKMAN  
G. CATHCART  
M. EMMANUEL  
I. BROEKMAN  
C. VAN DIJK  
DAICLIC  
EMC ASBL  
IDC BORDEAUX  
IDC LIEGE  
IDC PARIS  
P. JANIN  
IDC BORDEAUX  
C. MORIN  
IDC BORDEAUX  
IDC BORDEAUX  
DAINAMIC GERMANY  
L. LEGRY  
C. PICARD  
L. LEGRY

# Digitale functies

WELKE FUNKTIE WILT U ZIEN ?

AND (A)=CHR\$(0)



NAND (NA)=CHR\$(0)+CHR\$(3)



OR (O)=CHR\$(1)



NOR (NO)=CHR\$(1)+CHR\$(3)



NOT (NT)=CHR\$(11)+CHR\$(3)



EXOR (X)=CHR\$(2)



GEN. (G)=CHR\$(13)



SCHMT. (S)=CHR\$(9)



ENC (E)=CHR\$(10)



COUNT. (C)=CHR\$(6)+CHR\$(7)



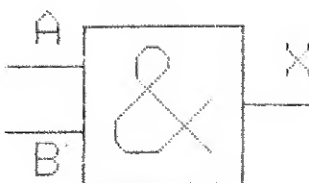
TAB-TOETS VOOR SCREENCOPY

PAGE 01 -- DIGITALE FUNCTIES

```

5 CLEAR 256:MODE 0
10 COLORT 8 0 0 0:CLEAR 2000:COLORG 0
  0 0 0:MODE 6:COLORG 9 15 10 14:
  MODE 6
15 POKE #131,1
20 DIM W$(15,0)
30 DF=1:FF=0:CC=20:SPATIE=6:X1=50:Y1=
  200
40 FOR A=0 TO 2:X1=50+A:Y1=200:CC=CC+1
45 T$="DIGITALE TECHNIK":GOSUB 20000:
  NEXT A:CC=20
50 FOR A=0 TO 2:X1=50+A:Y1=100:CC=CC+1
55 T$="AFD. ELECTRONICA.":GOSUB 20000:
  NEXT A
90 WAIT TIME 150 *** Use FGT as indicated in DAInamic 31 p.351 ***
95 MODE 6
  
```

AND

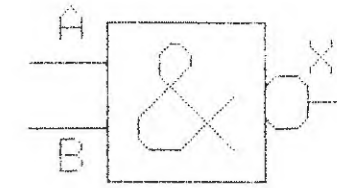


B	A	X
0	0	0
0	1	0
1	0	0
1	1	1

```

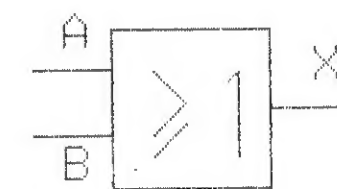
100 REM MENU
102 COLORG 9 15 10 14
105 DF=0:SPATIE=7
110 Y1=240:X1=50:CC=21
120 T$="WELKE FUNKTIE WILT U ZIEN ?":
  GOSUB 20000
130 CC=21:DRAW 260,235 270,254 22:DRAW
  260,235 270,216 22:DRAW 270,254
  275,244 22:DRAW 270,216 275,226 22
135 T$="CHR.MTS":X1=270:Y1=230:GOSUB
  20000:X1=280:Y1=220:T$="ALMELO":
  GOSUB 20000
140 CC=23:X1=20:Y1=210:T$="AND (A)=
  CHR$(0)":GOSUB 20000:T$=CHR$(0):X1=
  220:GOSUB 20000
145 X1=20:Y1=190:T$="NAND (NA)=CHR$(0)+
  CHR$(3)":GOSUB 20000:SPATIE=15:T$=
  CHR$(0)+CHR$(3):X1=220:GOSUB 20000:
  SPATIE=7
150 X1=20:Y1=170:T$="OR (O)=
  CHR$(1)":GOSUB 20000:T$=CHR$(1):X1=
  220:GOSUB 20000
155 X1=20:Y1=150:T$="NOR (NO)=CHR$(1)+
  CHR$(3)":GOSUB 20000:SPATIE=15:T$=
  CHR$(1)+CHR$(3):X1=220:GOSUB 20000:
  SPATIE=7
160 X1=20:Y1=130:T$="NOT (NT)=
  CHR$(11)+CHR$(3)":GOSUB 20000:
  SPATIE=15:X1=220:T$=CHR$(11)+
  CHR$(3):GOSUB 20000:SPATIE=7
165 X1=20:Y1=110:T$="EXOR (X)=
  CHR$(2)":GOSUB 20000:X1=220:T$=
  CHR$(2):GOSUB 20000
170 X1=20:Y1=90:T$="GEN. (G)=
  CHR$(13)":GOSUB 20000:X1=220:T$=
  CHR$(13):GOSUB 20000
175 X1=20:Y1=70:T$="SCHMT. (S)=CHR$(9)":
  GOSUB 20000:X1=220:T$=CHR$(9):
  GOSUB 20000
180 X1=20:Y1=50:T$="ENC (E)=
  CHR$(10)":GOSUB 20000:X1=220:T$=
  CHR$(10):GOSUB 20000
182 X1=20:Y1=30:T$="COUNT. (C)=CHR$(6)+
  CHR$(7)":GOSUB 20000:SPATIE=15:X1=
  220:T$=CHR$(6)+CHR$(7)+CHR$(8):
  GOSUB 20000:SPATIE=7
185 CC=21:X1=20:Y1=10:T$="TAB-TOETS
  VOOR SCREENCOPY":GOSUB 20000:
  SPATIE=8
190 Z=GETC:Z=GETC:Z=GETC
200 Z=GETC:IF Z=0 THEN 200
205 IF Z=9.0 THEN RESTORE:GOSUB 60000:
  GOSUB 30000:GOTO 100
210 IF Z=ASC("N") THEN 400:REM NAND OF
  NOR OF NOT
215 IF Z=ASC("A") THEN 1000:REM AND
220 IF Z=ASC("O") THEN 2000:REM OR
240 IF Z=ASC("X") THEN 4000:REM EXOR
250 IF Z=ASC("G") THEN 5000:REM
  KLOKGENERATOR
260 IF Z=ASC("S") THEN 6000:REM
  SCHMITTRIGGER
270 IF Z=ASC("E") THEN 7000:REM ENCODER
290 IF Z=ASC("C") THEN 9000:REM COUNTER
  
```

NAND



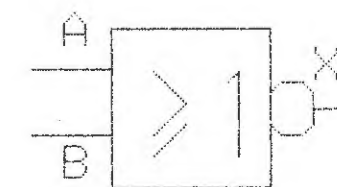
B	A	X
0	0	1
0	1	1
1	0	1
1	1	0

OR



B	A	X
0	0	0
0	1	1
1	0	1
1	1	1

NOR



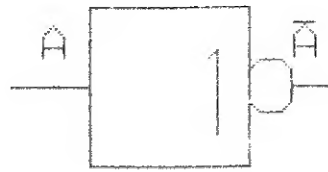
B	A	X
0	0	1
0	1	0
1	0	0
1	1	0

```

300 GOTO 190
400 Z=GETC:IF Z=0 THEN 400
410 IF Z=ASC("A") THEN 1000:REM NAND
420 IF Z=ASC("O") THEN 1100:REM NOR
425 IF Z=ASC("T") THEN 3000:REM NOT
430 GOTO 400

```

# NOT



A	Ā
0	1
1	0

```

1000 REM AND
1005 GOSUB 22000
1010 W$(0,0)="0 0 0":W$(1,0)="0 1 0":
W$(2,0)="1 0 0":W$(3,0)="1 1 1"
1020 NAAM$="AND":GOSUB 24000
1030 SYMB$=CHR$(0):GOSUB 25000
1040 GOSUB 26000:GOSUB 28000:GOSUB
27000:GOSUB 21000:GOSUB 23000
1070 GOTO 100:REM TERUG NAAR MENU

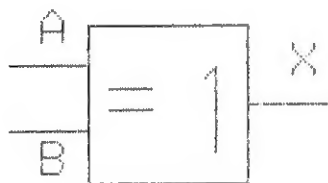
```

```

2000 REM OR
2005 GOSUB 22000
2010 W$(0,0)="0 0 0":W$(1,0)="0 1 1":
W$(2,0)="1 0 1":W$(3,0)="1 1 1"
2020 NAAM$="OR":GOSUB 24000
2030 SYMB$=CHR$(1):GOSUB 25000
2040 GOSUB 26000:GOSUB 28000:GOSUB
27000:GOSUB 21000:GOSUB 23000
2050 GOTO 100:REM TERUG NAAR MENU

```

# EXOR



B	A	X
0	0	0
0	1	1
1	0	1
1	1	0

```

3000 REM NOT/INVERTER
3005 GOSUB 22000
3010 W$(0,0)="0 1":W$(1,0)="1 0"
3020 NAAM$="NOT":GOSUB 24000
3030 SPATIE=15:SYMB$=CHR$(11)+CHR$(3):
GOSUB 25000:SPATIE=8
3035 GOSUB 29000:GOSUB 21000:GOSUB 23000
3050 GOTO 100:REM TERUG NAAR MENU

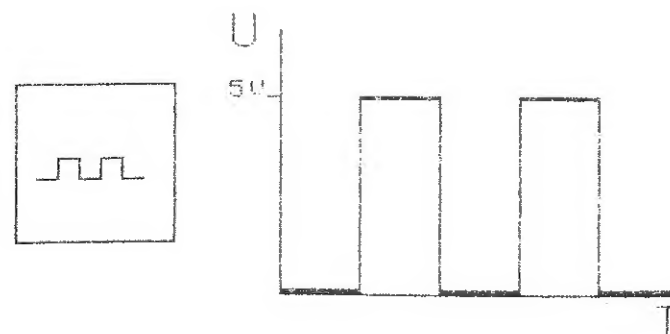
```

```

4000 REM EXOR
4005 GOSUB 22000
4010 W$(0,0)="0 0 0":W$(1,0)="0 1 1":
W$(2,0)="1 0 1":W$(3,0)="1 1 0"
4020 NAAM$="EXOR":GOSUB 24000
4030 SYMB$=CHR$(2):GOSUB 25000
4040 GOSUB 26000:GOSUB 28000:GOSUB
27000:GOSUB 21000:GOSUB 23000
4050 GOTO 100:REM TERUG NAAR MENU

```

# GENERATOR



```

5000 REM KLOKGENERATOR
5010 GOSUB 22000
5020 DFO=DF:DF=1
5030 SPATIE=6:NAAM$="GENERATOR":GOSUB
24000:SPATIE=8
5040 DF=DFO:SYMB$=CHR$(13):GOSUB 25000
5042 DF=1
5045 DRAW 150,80 150,180 22:DRAW 150,80
300,80 22
5050 X1=130:Y1=170:T$="U":GOSUB 20000
5055 X1=290:Y1=60:T$="T":GOSUB 20000
5060 DRAW 145,155 150,155 22:DF=0:X1=
130:Y1=153:T$="5V":GOSUB 20000
5065 FOR A=151 TO 180:DOT A,81 21:DOT A,
82 21:WAIT TIME 2:NEXT A:DRAW 180,
81 180,155 21

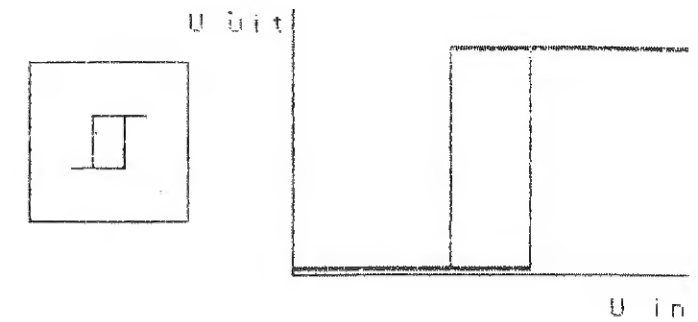
```

```

5070 FOR A=180 TO 210:DOT A,154 21:DOT
A,155 21:WAIT TIME 2:NEXT A:DRAW
210,81 210,155 21
5075 FOR A=210 TO 240:DOT A,81 21:DOT A,
82 21:WAIT TIME 2:NEXT A:DRAW 240,
81 240,155 21
5080 FOR A=240 TO 270:DOT A,154 21:DOT
A,155 21:WAIT TIME 2:NEXT A:DRAW
270,81 270,155 21
5085 FOR A=270 TO 300:DOT A,81 21:DOT A,
82 21:WAIT TIME 2:NEXT A
5089 DF=DFO
5090 GOSUB 21000:GOSUB 23000:DF=DFO
5099 GOTO 100:REM TERUG NAAR MENU

```

# SCHMITT-TRIGGER



```

6000 REM SCHMITTRIGGER
6005 GOSUB 22000
6010 NAAM$="SCHMITT-TRIGGER"
6020 X1=50:Y1=200:DF=1:CC=22:T$=NAAM$:
GOSUB 20000
6030 DF=3:SYMB$=CHR$(9):GOSUB 25000
6040 DRAW 150,80 150,180 23:DRAW 150,80
300,80 23
6045 DF=0
6050 X1=110:Y1=170:T$="U uit":GOSUB
20000
6060 X1=270:Y1=63:T$="U in":GOSUB 20000
6070 FOR A=151 TO 240:DOT A,82 22:DOT A,
83 22:WAIT TIME 2:NEXT A
6080 DRAW 240,82 240,165 21
6090 FOR A=240 TO 300:DOT A,165 22:DOT
A,166 22:WAIT TIME 2:NEXT A:WAIT
TIME 20
6100 FOR A=300 TO 210 STEP -1:DOT A,165
23:DOT A,166 23:WAIT TIME 2:NEXT A
6200 DRAW 210,82 210,166 23
6300 FOR A=210 TO 151 STEP -1:DOT A,82
23:DOT A,83 23:WAIT TIME 2:NEXT A
6990 GOSUB 21000:GOSUB 23000
6995 GOTO 100:REM TERUG NAAR MENU

```

# DECODER

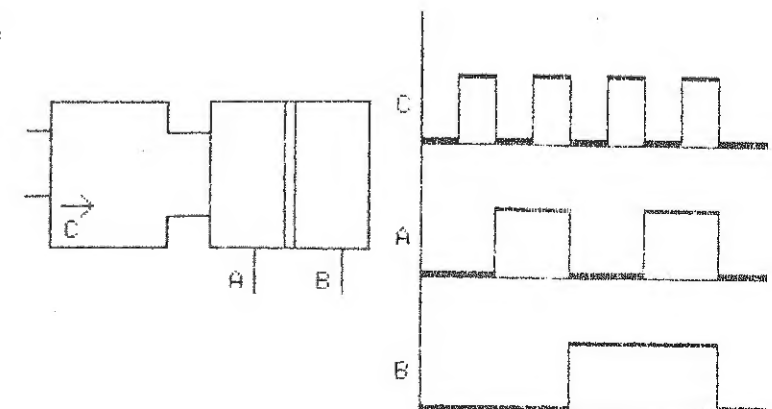
D	C	B	A	GRAY-CODE
0	0	0	0	0 0 0 0 0
0	0	0	1	1 0 0 0 1
0	0	1	0	2 0 0 1 1
0	0	1	1	3 0 0 1 0
0	1	0	0	4 0 1 1 0
0	1	0	1	5 0 1 1 1
0	1	1	0	6 0 1 0 1
0	1	1	1	7 0 1 0 0
1	0	0	0	8 1 1 0 0
1	0	0	1	9 1 1 0 1
1	0	1	0	10 1 1 1 1
1	0	1	1	11 1 1 1 0
1	1	0	0	12 1 0 1 0
1	1	0	1	13 1 0 1 1
1	1	1	0	14 1 0 0 1
1	1	1	1	15 1 0 0 0

```

7000 REM DECODER
7005 GOSUB 22000
7010 NAAM$="DECODER":GOSUB 24000
7020 SYMB$=CHR$(10):GOSUB 25000
7030 GOSUB 21000
7040 MODE 6
7050 X1=50:Y1=220:T$="DECODER":GOSUB
20000
7060 DRAW 0,211 XMAX,211 23:SPATIE=7:DF=
0
7070 X1=1:Y1=213:T$="D C B A
GRAY-CODE":GOSUB 20000
7090 W$(0,0)="0 0 0 0 BIN/GRAY
0 0 0 0 0":W$(1,0)="0 0 0 1
1 0 0 0 1"
7100 W$(2,0)="0 0 1 0
2 0 0 1 1":W$(3,0)="0 0 1 1
3 0 0 1 0"
7110 W$(4,0)="0 1 0 0 INVULRAAM
4 0 1 1 0":W$(5,0)="0 1 0 1
5 0 1 1 1"

```

# TELLER



```

7120 W$(6.0)="0 1 1 0      A
      6 0 1 0 1":W$(7.0)="0 1 1 1
      7 0 1 0 0"
7130 W$(8.0)="1 0 0 0      0 1 3 2
      8 1 1 0 0":W$(9.0)="1 0 0 1 C 4
      5 7 6      9 1 1 0 1"
7140 W$(10.0)="1 0 1 0      12 13 15 14
      10 1 1 1 1":W$(11.0)="1 0 1 1
      8 9 11 10 D 11 1 1 1 0"
7150 W$(12.0)="1 1 0 0
      12 1 0 1 0":W$(13.0)="1 1 0 1
      B      13 1 0 1 1"
7160 W$(14.0)="1 1 1 0
      14 1 0 0 1":W$(15.0)="1 1 1 1
      15 1 0 0 0"

7165 Y1=208
7170 FOR X=0 TO 15:Y1=Y1-13:T$=W$(X):
      GOSUB 20000:NEXT X
7180 DRAW 55,0 55,211 23:DRAW 180,0 180,
      211 23
7190 DRAW 75,48 156,48 21:DRAW 75,48 75,
      101 21
7200 DRAW 75,61 156,61 21:DRAW 95,48 95,
      101 21
7210 DRAW 75,75 156,75 21:DRAW 115,48
      115,101 21
7220 DRAW 75,88 156,88 21:DRAW 135,48
      135,101 21
7230 DRAW 75,101 156,101 21:DRAW 156,48
      156,101 21
7240 DRAW 95,110 135,110 21:DRAW 115,40
      155,40 21
7245 DRAW 95,109 135,109 21:DRAW 115,39
      155,39 21
7250 DRAW 71,61 71,88 21:DRAW 160,48
      160,75 21
7255 FILL 76,49 155,100 19:NAAM$=
      "DECODER"
7260 GOSUB 21020:GOSUB 23000:SPATIE=8:
      GOTO 100

9000 REM TELLER
9005 GOSUB 22000
9010 NAAM$="TELLER":GOSUB 24000
9015 SPATIE=15:SYMB$=CHR$(6)+CHR$(7):
      GOSUB 25000:SPATIE=8
9020 DRAW 20,120 50,120 23:DRAW 20,145
      50,145 23
9030 DRAW 127,83 127,100 23:DRAW 160,83
      160,100 23
9035 DRAW 190,40 190,190 23:DRAW 190,40
      320,40 23
9040 DRAW 190,90 320,90 23:DRAW 190,140
      320,140 23
9045 DFO=DF:DF=0
9050 X1=55:Y1=102:T$="C":GOSUB 20000
9055 X1=117:Y1=83:T$="A":GOSUB 20000
9060 X1=150:T$="B":GOSUB 20000
9070 X1=180:Y1=150:T$="C":GOSUB 20000
9080 Y1=100:T$="A":GOSUB 20000
9090 Y1=50:T$="B":GOSUB 20000

```

```

9100 FOR A=191 TO 204:DOT A,141 21:DOT
      A,91 21:DOT A,41 21
9110 DOT A,142 21:DOT A,92 21:DOT A,42
      21:WAIT TIME 2:NEXT A
9120 DRAW 204,141 204,165 21
9130 FOR A=204 TO 218:DOT A,165 21:DOT
      A,91 21:DOT A,41 21
9140 DOT A,166 21:DOT A,92 21:DOT A,42
      21:WAIT TIME 2:NEXT A
9150 DRAW 218,141 218,166 21:DRAW 218,
      91 218,116 21
9160 FOR A=218 TO 232:DOT A,141 21:DOT
      A,115 21:DOT A,41 21
9170 DOT A,142 21:DOT A,116 21:DOT A,42
      21:WAIT TIME 2:NEXT A
9180 DRAW 232,141 232,166 21
9190 FOR A=232 TO 246:DOT A,165 21:DOT
      A,115 21:DOT A,41 21
9200 DOT A,166 21:DOT A,116 21:DOT A,42
      21:WAIT TIME 2:NEXT A
9210 DRAW 246,141 246,166 21:DRAW 246,
      91 246,116 21:DRAW 246,41 246,66 21
9220 FOR A=246 TO 260:DOT A,141 21:DOT
      A,91 21:DOT A,65 21
9230 DOT A,142 21:DOT A,92 21:DOT A,66
      21:WAIT TIME 2:NEXT A
9240 DRAW 260,141 260,166 21
9250 FOR A=260 TO 274:DOT A,165 21:DOT
      A,91 21:DOT A,65 21
9260 DOT A,166 21:DOT A,92 21:DOT A,66
      21:WAIT TIME 2:NEXT A
9270 DRAW 274,141 274,166 21:DRAW 274,
      91 274,116 21
9280 FOR A=274 TO 288:DOT A,141 21:DOT
      A,115 21:DOT A,65 21
9290 DOT A,142 21:DOT A,116 21:DOT A,66
      21:WAIT TIME 2:NEXT A
9300 DRAW 288,141 288,166 21
9310 FOR A=288 TO 302:DOT A,165 21:DOT
      A,115 21:DOT A,65 21
9320 DOT A,166 21:DOT A,116 21:DOT A,66
      21:WAIT TIME 2:NEXT A
9330 DRAW 302,141 302,166 21:DRAW 302,
      91 302,116 21:DRAW 302,41 302,66 21
9340 FOR A=302 TO 320:DOT A,141 21:DOT
      A,91 21:DOT A,41 21
9350 DOT A,142 21:DOT A,92 21:DOT A,42
      21:WAIT TIME 2:NEXT A
9995 GOSUB 21000:GOSUB 23000
9999 GOTO 100

10000 REM NAND
10005 GOSUB 22000
10010 W$(0.0)="0 0 1":W$(1.0)="0 1 1":
      W$(2.0)="1 0 1":W$(3.0)="1 1 0"
10020 NAAM$="NAND":GOSUB 24000
10030 SPATIE=15:SYMB$=CHR$(0)+CHR$(3):
      GOSUB 25000:SPATIE=8
10040 GOSUB 26000:GOSUB 28000:GOSUB
      27000:GOSUB 21000:GOSUB 23000
10050 GOTO 100:REM TERUG NAAR MENU

```

```

11000 REM NOR
11005 GOSUB 22000
11010 W$(0.0)="0 0 1":W$(1.0)="0 1 0":
      W$(2.0)="1 0 0":W$(3.0)="1 1 0"
11020 NAAM$="NOR":GOSUB 24000
11030 SPATIE=15:SYMB$=CHR$(1)+CHR$(3):
      GOSUB 25000:SPATIE=8
11040 GOSUB 26000:GOSUB 28000:GOSUB
      27000:GOSUB 21000:GOSUB 23000
11050 GOTO 100:REM TERUG NAAR MENU

20000 REM SUBROUTINE FGT
20001 C=FC*#40+CC:F=FF*#80+PF*#40+ZF*#20+
      VF*#10+DF
20002 POKE #2F0,C:POKE #2F1,F
20003 POKE #2F2,X1 MOD 256:POKE #2F3,
      X1/256:POKE #2F4,Y1
20004 POKE #2F5,SPATIE:POKE #2F6,ID
20005 CALLM #300,T$:RETURN

21000 REM SPATIEBALK SUBROUTINE
21005 DFO=DF:CCO=CC
21008 DRAW 250,3 250,15 22:DRAW 250,15
      330,15 22
21010 DRAW 250,3 330,3 22:DRAW 330,3 330,
      15 22
21015 DF=0:X1=251:Y1=5:T$="SPATIEBALK":
      GOSUB 20000
21020 Z=BETC:IF Z=9 THEN RESTORE:GOSUB
      60000:GOSUB 30000:GOTO 21050
21030 IF Z<>32 THEN 21020
21050 DF=DFO:CC=CCO
21060 RETURN

22000 REM BEELD SCHOONMAKEN + VARIABELEN
      DEFINIEREN VOOR NIEUWE BLZ.
22005 MODE 6
22010 RETURN

23000 REM BEELD SCHOONMAKEN + VARIABELEN
      DEFINIEREN VOOR MENU
23005 MODE 6
23006 DF=0:RETURN

24000 REM NAAM VAN DE BLZ.
24005 REM GOSUB 22000:REM BEELD
      SCHOONMAKEN
24010 DF=3:X1=50:Y1=200:CC=22:T$=NAAM$:
      GOSUB 20000:RETURN

25000 REM SYMBOOL OP HET SCHERM
25005 DF=3:X1=50:Y1=100:CC=23:T$=SYMB$:
      GOSUB 20000:RETURN

26000 REM 3 AANSLUITDRADEN
26005 DRAW 20,145 50,145 23:DRAW 20,120
      50,120 23
26010 IF NAAM$="NAND" OR NAAM$="NOR"
      THEN DRAW 127,130 140,130 23:RETURN
26020 DRAW 110,130 140,130 23:RETURN

```

```

27000 REM VARIABELEN OP HET SCHERM
27005 X1=30:Y1=150:DFO=DF:DF=1:T$="A":
      GOSUB 20000
27010 Y1=100:T$="B":GOSUB 20000
27015 X1=125:Y1=137:T$="X":GOSUB 20000:
      DF=DFO:RETURN

28000 REM WAARHEIDSTABEL OP HET SCHERM
28002 DRAW 190,150 285,150 23:DRAW 220,
      170 220,70 23:DRAW 250,170 250,70
      23
28004 DFO=DF:DF=1
28005 X1=200:Y1=150:T$="B":GOSUB 20000
28010 X1=232:T$="A":GOSUB 20000
28015 X1=264:T$="X":GOSUB 20000
28020 X1=200:Y1=130:T$=W$(0.0):GOSUB
      20000
28025 Y1=110:T$=W$(1.0):GOSUB 20000
28030 Y1=90:T$=W$(2.0):GOSUB 20000
28035 Y1=70:T$=W$(3.0):GOSUB 20000
28037 DF=DFO:RETURN

29000 REM WAARHEIDSTABEL +
      AANSLUITDRADEN (INVERTER)
29002 DRAW 190,150 255,150 23:DRAW 220,
      170 220,100 23
29005 DFO=DF:DF=1
29010 X1=200:Y1=150:T$="A":GOSUB 20000
29020 X1=232:T$="A":GOSUB 20000:DRAW 234,
      167 242,167 23
29030 X1=200:Y1=130:T$=W$(0.0):GOSUB
      20000
29040 Y1=110:T$=W$(1.0):GOSUB 20000
29060 DRAW 20,130 50,130 23:DRAW 127,130
      140,130 23
29070 X1=30:Y1=137:T$="A":GOSUB 20000
29080 X1=125:T$="A":GOSUB 20000:DRAW 127,
      154 135,154 23:REM A NIET
29090 DF=DFO:RETURN

30000 REM SCREENCOPY
30005 IF NAAM$<>"DECODER" THEN FILL 250,
      3 330,15 20
30006 IF NAAM$="DECODER" THEN FILL 76,49
      155,100 18
30008 POKE #131,0:COLORG 8 0 0 0
30010 CALLM #5400
30015 POKE #131,1
30020 RETURN
60000 FOR A=#5400 TO #54B2
60001 1 READ B
60002 1 POKE A,B
60003 NEXT A:RETURN
60004 DATA #C5,#D5,#E5,#F5,#F3,#3A,#40,
      #00,#F5,#E6,#3F,#F6,#80,#32,#40,
      #00,#32,#06
60005 DATA #FD,#FB,#21,#00,#00,#4C,#CD,
      #84,#EB,#3E,#F9,#80,#E6,#FB
60006 DATA #C6,#07,#4F,#13,#D5,#C5,#3E,
      #1B,#CD,#97,#54,#3E,#4E,#CD,#97,
      #54,#3E,#00

```



# Assembly Language part 4

PROGRAMMING THE DAI IN MACHINE AND ASSEMBLY LANGUAGES  
by C W Read

## Part 4 - CHANGING MODES, COLOURS & BITS

The DAI's ROM routines for setting or changing screen MODEs and colours can be used in machine language programmes. The routines, like those for the editing facility, are called by RST 5 (op code EF) instructions: RST 5 Data 18 for changing mode, RST 5 Data 6 for text colours, and RST 5 Data 1B for graphics colours. A precondition for the mode change instruction is that the requisite mode code (one from the list below) should be in the accumulator.

MODE	Code	MODE	Code	MODE	Code
0	FF	3	04	5	08
1	00	3A	05	5A	09
1A	01	4	06	6	0A
2	02	4A	07	6A	0B
2A	03				

Example: To change screen to MODE 6A

```
Source      Object code      Comment
MVI A 0BH   3E 0B           ;Load #0B into accumulator
RST 5       EF
DB 18H      1B             ;Change to new mode and return for
...                               ; next instruction
```

The precondition for both colour instructions is that the HL register pair must be pointing at a list containing the four colours; the RST routine will step HL to get each colour in turn. The colours (in hex form) are stored as labelled data in the programme, thus:

```
COLOR1 DB 08H,00H,0BH,01H ;Colours 8,0,11,1
COLOR2 DB 07H,01H,0CH,0FH ;Colours 7,1,12,15
```

Then the following instructions can go at the beginning of your programme, for setting text colours, &/or at a later position for changing colours:-

```
Source      Object code      Comment
LXI H COLOR1 21 xx xx     ;Load HL with address of colour data
RST 5       EF
DB 06H      06             ;Load colour registers, change to new
...                               ; colours & return for next instrn.
```

The same method is used to set graphics colours but RST data is then 1B.

The possible use of an alternative cursor symbol for the editor of the file writer programme was mentioned in Part 3. With text colours set as above the alternative symbol can now be a coloured block. Memory addresses 0074, 0075 are the Cursor Type byte and the Cursor Information byte respectively. The normal horizontal line symbol has ASCII code #5F in the information byte. A different ASCII character code there will provide an alternative symbol. For example, #45 will produce an E or #FF a block. To colour the block the cursor type byte has to be changed from 01 to 00. The third colour register provides the colour. Here is how to do it (subroutine labelled EdCur) and how to restore it to normal again (labelled NmlCur). Insert both subroutines at the end of your source code file, immediately before the END statement.

```
;CHANGE CURSOR SYMBOL FOR EDITOR:
EdCur MVI A 0FFH ;Load A with #FF, ASCII code for block symbol,
STA 75H ; put it in cursor information byte.
XRA A ;Zero the accumulator
STA 74H ;Store 0 from accumulator in cursor type byte
RET ;Return to main programme

;RESTORE NORMAL CURSOR SYMBOL:
NmlCur MVI A 5FH ;ASCII for horizontal line symbol,
STA 75H ; put it in cursor information byte.
MVI A 1H ;Put 1 in cursor type to alternate cursor
STA 74H ; symbol with character, as normal
RET
```

Insert a call to the first subroutine at the start of the editor initialisation routine, and a call to the second at the start of the BREAK routine thus:

```
InitEd CALL EdCur ;
LHLD BufSt ... etc ;
BREAK CALL NmlCur ;
LHLD 0A4H ... etc ;
```

We will now look at the Rotate group of instructions that move bits in the accumulator right or left; rotate, because a bit moved off one end of the byte rotates around an imaginary loop to enter the other end. Thus no bits are actually lost. The instructions, with their op codes, are:

RAR Rotate Accumulator Right (1F), RRC Rotate Right without Carry (0F)  
RAL Rotate Accumulator Left (17), RLC Rotate Left without Carry (07)  
Each instruction moves all the bits in register A one step to the right or left. The Carry works differently according to the type of rotate called: RAR and RAL actually rotate 9 bits, ie: the byte plus the Carry flag bit, so that a bit moved out of the byte at either end enters the carry bit, displacing the previous carry to the the other end of the byte, thus:

	Carry	Byte in accumulator
Present state	1	0001 0110
After one RAL	0	0010 1101
After a 2nd RAL	0	0101 1010

RRC and RLC only rotate 8 bits, hence "without Carry". Now however the Carry flag is used to record the status of the bit moved out of one end of the byte and into the other; if that bit is 1 the Carry is set; if the bit is 0 Carry is not set:

	Carry	Byte
Present state	0	0101 0110
After one RLC	0	1010 1100
After a 2nd RLC	1	0101 1001

All odd numbers have 1 in the least significant bit; therefore RAR and RRC can be used to test if the value in the accumulator is odd or even. One RAR will put the LSB in the Carry bit. Alternatively, one RRC will rotate the LSB into the MSB and the Carry bit will adopt the status of the bit rotated. Thus either method will set the carry flag if the accumulator contained an odd value. If you glance back to the list of Mode codes at the beginning of this article you will notice that the code for MODE 0 and every split mode is an odd number. When the DAI's screen driving package needs to know if the screen, or part of it, is in a text mode, it loads the stored mode code into the accumulator and uses RRC (or sometimes RAR) to find if the code is an odd number. There is also an occasion when the carry is deliberately set (made 1) and rotated into the accumulator to change the mode code from even to odd. It is used when a graphics screen has to be changed to split-mode for

displaying an error message. This extract from the ROM routine shows how neatly it is done:-

```

...
LDA 009DH      ;Load accumulator with mode code stored at #009D
RRC           ;Rotate puts status of LSB in carry. Is code odd?
JC  ABORT     ;Yes, current mode is a text mode
STC          ;No, its a graphics mode. Set the carry bit to 1
RAL         ;Rotate back to move the carry into the LSB, thus
           ; changing the code in accumulator to odd, for
           ; split mode. eg: existing MODE 6 becomes MODE 6A.
CALL CHANGE   ;Change to split mode and store new code at #009D
... etc

```

That STC instruction is new. It means Set The Carry flag; the op code is 37. It simply makes the carry bit 1. There is no 'unsetting' instruction to make the carry bit 0 although CMC, Complement the Carry flag (op code 3F) will do that if the carry has been previously set. Alternatively all the instructions in the AND, OR and Exclusive OR groups put a zero in the carry flag.

This next routine uses a Rotate instruction. The file writing routines, NEW and EDIT, produced in Part 3, allow keyboard generated ASCII coded characters and controls like TAB and Carriage Return to be typed into the file. Sometimes however non-ASCII values are needed as control codes for printers or to put the £ sign on the screen. BASIC uses the expression CHR\$. eg: the £ sign can be produced by either PRINT CHR\$(27) or PRINT CHR\$(#1B). How can that be programmed in assembler? If 1B is typed the result from the keyboard is two ASCII coded bytes, #31 representing 1 and #42 representing the B. Therefore the ASCII coding has to be removed and the two bytes compressed into one byte to obtain the required #1B. Here is what happens, shown with each byte split into two half-bytes (or nibbles) of 4 bits:

	1st byte	2nd byte
Figure or letter required	1	B
Equivalent ASCII code from keyboard,	#31	#42
and expressed as Binary Coded Decimal	0011 0001	0100 0010
Subtract #30 to remove ASCII coding	- 0011 0000	- 0011 0000
Results (decimals 1 and 12)	= 0000 0001	= 0001 0010
If >9 it will be #A to #F, subtract 7	...	- 0000 0111
Results in binary and	= 0000 0001	= 0000 1011
equivalent in hexadecimal	01	0B

We now have the required values but they are still in two bytes.

```

Take the 1st byte      0000 0001 (#01)
Move left 4 steps to get 0001 0000 (#10)
Now add the 2nd byte   + 0000 1011 (#0B)
Result, one byte      = 0001 1011 (#1B)

```

Although the above example used #1B any other valid hex byte may be needed. The routine should therefore validate the characters from the keyboard to ensure that each is in the hex range "0" to "F", rejecting all others. Compare instructions can be used to validate a range of values. They set the carry flag if the compare operand is greater than the value in the accumulator. Thus CPI '0' would check the bottom of the range and CPI 'B' (top of range+1) would ensure that anything greater than F was rejected.

The DAI's editor sees an 00 byte as end-of-text marker so 00 must also be treated as invalid. It is easier to reject it after conversion, when it is one byte, than when it comes from the keyboard as two #30s.

Here then is the routine, labelled HEXCH1, which takes two characters from the keyboard, validates them, converts them to one hex byte and puts that in the text buffer and on screen. Register C is used to count the acceptable entries; register D to hold the first converted digit while the second is being converted. If you wish to use the facility in the file writing programme it can be inserted at the end of the source programme, before the END line.

```

HEXCH1 MVI C 2H      ; 2 characters expected from keyboard
HEXCH2 CALL GETC    ;Get character from keyboard
JC  BREAK          ;If BREAK key pressed
JZ  HEXCH2         ;If no key pressed
CPI 30H           ;ASCII 0?      ) Verify that
JC  HEXEND        ;Abort          ) characters typed
CPI 47H           ;ASCII 6?      ) are in the range
JNC HEXEND        ;Abort          ) '0' to 'F'
CPI 3AH           ;Is it ASCII 0 to 9?
JC  HEXCH3        ;Yes, skip next instruction
SUI 7H            ;No, it is an alpha character. Deduct 7
HEXCH3 SUI 30H     ;Deduct 30 to convert from ASCII to Hex
DCR C            ;Record 1 hex digit received
JZ  HEXCH4        ;If register C=0 that was 2nd hex digit
RLC             ;Rotate 4 times to move 1st hex
RLC             ; digit from the low nibble into
RLC             ; the high nibble of the byte.
RLC
MOV D,A         ;Save the result in register D.
JMP HEXCH2     ;Now get 2nd character
HEXCH4 ADD D    ;1st digit is in D (high nibble), 2nd digit in
              ; A (low nibble). Adding them leaves both in
              ; 1 byte in A. If both digits are 0, adding
              ; them will set the zero flag, indicating
              ; that the result is 00, so abort.
JMP INSERT    ;Put result (2 Hex digits) in edit buffer.
HEXEND JMP KEYBD3 ;Go back to keyboard for next command

```

The keyboard section of the main programme must recognise when the hex digits facility is needed. Use of a key that is rarely wanted for text is one possibility: the key for the ~ symbol (ASCII 7E) may do. Should it later be necessary to print a ~ the HEXCH1 routine can be used to enter the hex digits 7E which will put ~ on the screen and in the buffer. Substitute another key if you wish. Insert the following two lines in KEYBD3 section, between JZ KEYBD3 and CALL INSERT:

```

CPI 7EH      ;Is it ~ key?
JZ  HEXCH1   ;To insert 2 hex characters

```

Rotate commands are again used when two hex digits in a byte have to be separated and ASCII-coded for printing. That process also uses a Logical AND instruction for clearing half a byte.

There are six Logical instructions:

```

ANI (op code E6) which ANDs the Immediate data with the accumulator.
ANA r which ANDs the data in register r with the accumulator.
ORI (op code F6) and ORA r, corresponding logical OR instructions.
XRI (op code EE) and XRA r, logical EXCLUSIVE OR instructions.

```

ANI and ANA are used to zero particular bits in a byte. ORI and ORA are used to set particular bits to 1. XRI and XRA are used to complement particular bits. The data (either immediate or in register r) is called a mask.

'r' is any register A to L or M (the Memory at address in HL).



## Summary:

Instruction	Bit in mask	Corresponding bit in accumulator
ANA and ANI	0	0
ANA and ANI	1	No change
ORA and ORI	0	No change
ORA and ORI	1	1
XRA and XRI	0	No change
XRA and XRI	1	Complemented

Sign, parity and zero flags are set in accordance with the result, but the carry flag is always made zero as logical commands cannot borrow or carry bits.

## Examples:

Byte in accumulator as Binary Coded Decimal	#3B		#97
ANI #0F (mask)	0011 1011	ANI #F0 (mask)	1001 0111
1st result	0000 1011		1001 0000
ORI #30 (mask)	0011 0000	ORI #16 (mask)	0001 0110
2nd result	0011 1011		1001 0110
XRI #F0 (mask)	1111 0000	XRA A (mask)	1001 0110
3rd result	1100 1011		0000 0000

Suppose for example that an address stored at location EndTxt has to be displayed. It is a four character address stored as two bytes, low byte first. It is to be printed the right way around so we will start with the second byte, convert it to two bytes, add the ASCII character coding and print it. Then repeat the process with the first byte.

```

PrtAdr MVI C 2H ;Counter for 2 bytes
        LXI H EndTxt ;Address of store into HL
        INX H ;Step HL to next address and
GetByt MOV A,M ; move the contents to A
        ANI 0F0H ;Delete bits in low nibble
        RRC ;Move the four bits of the
        RRC ; high nibble to the cleared
        RRC ; low nibble.
        CALL T&P ;Test, make ASCII and print
        MOV A,M ;Get same byte in A again
        ANI 0FH ;Delete the high nibble
        CALL T&P ;Test and print next digit
        DCX H ;Step HL back to pick up the low byte
        DCR C ;Decrement byte counter
        JNZ GetByt ;Repeat for next byte
        JMP KEYBD1 or wherever appropriate.
    
```

```

;Subroutine to handle alpha digits, make ASCII and print.
;Value to be tested and coded is already in the accumulator.
T&P CPI 0AH ;Is result 0 to 9 ?
     JC ASCI ;Yes
     ADI 7H ;No. So it is a letter. Add 7
ASCI ORI 30H ;Add ASCII coding. (see Note)
     CALL OUTC ;Print it
     RET
    
```

Note: The ORI 30H instruction could alternatively be ADI 30H.

To be continued.

Laat je computer aan het woord blijven.

Bij het thema communicatie hoeft men niet noodzakelijkerwijs te denken aan verbindingen via telefoonlijnen. We zullen een communicatiemogelijkheid beschrijven tussen een computer met een RS-232poort en een wat sporadisch randapparaat, de volledig elektronisch bestuurbare audiocassetterecorder van Tandberg, de TCCR-530.

De TCCR-530 is uitgerust met een 8048 processor, die via een RS-232-C 25-pins D-connector kan communiceren.

Hij accepteert ASCII-strings die afgesloten dienen te worden met <CR> oftewel CHR\$(13): bijvoorbeeld "PL"<CR> staat voor het commando "PLAY".

De TCCR-530 ziet er uit als een diplomatenkoffertje (breedte 44,5 cm, hoogte 14,2 cm, diepte 28 cm), heeft een enigszins oplopend bedieningspaneel, waarop onder andere een gecombineerde seconden- en minutenteller zichtbaar is.

De normale functies van een audiocassetterecorder zijn via wat klein uitgevallen toetsen bereikbaar en er is een RECAP-toets voor het terughalen van het laatste geluidsfragment.

Naast de RS-232 geeft het achterpaneel nog verbindingsmogelijkheden te zien voor een LINE-verbinding met radio of tuner/versterker naast een REMOTE CONTROL waarop bijvoorbeeld een infrarood- of een voetbesturing (niet meegeleverd), maar ook een met relais schakelende cassettecontrolepoort van een computer kan worden aangesloten.

De LINEverbinding is uiteraard ook geschikt om computerprogramma's van tape te laden, waarbij de speaker kan worden dichtgedraaid. Dok voor Basicode uiteraard.

Er is een speaker ingebouwd, maar die is via de speakerplug uitschakelbaar, zodat via externe speakers of een koptelefoon gewerkt kan worden.

Tot nog toe klinkt dat allemaal vrij normaal voor een audiocassetterecorder, maar het zou geen Tandberg zijn, als er niet veel meer te beleven zou zijn.

Allereerst zijn er wat fluitsignalen beschikbaar, die de gebuiker attent maken op zaken zoals bandeinde en cassette-inlegfouten, naast signalen die je waarschuwen als je geluid wilt opnemen zonder cassette of op een beschermde cassette. Comfortabele service van de processor, nietwaar?

Ik gaf al aan dat er een digitale teller op het frontpaneel te zien was. Deze telt seconden en minuten en schakelt na 59 minuten en 59 seconden weer over op 00.00, dat is pas "real time".

In beginsel aardig om de werkelijke bandlengte van je cassettes eens mee te controleren. (Blijkt aardig te kloppen overigens.)

Het model dat hier beschreven is, is een mono-uitvoering, maar er schijnt ook een stereoversie te bestaan.

In dat geval kan er een audiokanaal gebruikt worden voor bijvoorbeeld stuurpulsen voor een diaprojector.

De computer verbonden met de TCCR-530.

Zowat elke computer van single-chip tot mainframe moet in staat zijn via de RS-poort een halfduplextransmissie te onderhouden met de TCCR, zelfs een 20 mA current loop mogelijkheid is voorzien.

De volgende pinnen worden gebruikt:

- |                   |        |
|-------------------|--------|
| 1. transmit data  | pin 2; |
| 2. receive data   | pin 3; |
| 3. signal ground. | pin 7. |

Om de correcte logische niveaus te bereiken kan het nodig zijn voor uw computer meer pinnen te moeten solderen dan dat hier staat aangegeven, maar de handleiding is daar duidelijk genoeg over.

Van de fabriek uit is de machine ingesteld op 1200 baud bij een karakterlengte van 7 bits en EVEN pariteit.

Klopt dat niet dat moet ofwel aan de computerkant een wijziging plaatsvinden ofwel aan de kant van de TCCR.

Voor dat laatste is er de PARAMETERmode, die via een combinatie van frontpaneeltoetsen bereikt kan worden.

Je kunt dan de BAUDrates veranderen (150, 300, 600, 1200, 2400, 4800 en 9600 Bd). Het recorderadres is instelbaar: het is mogelijk om 96 recorders te adresseren.

Tevens is de antwoordtijd van de TCCR instelbaar, wat voor sommige communicatieprotocollen wel zinnig kan zijn.

Het tellervenster geeft dan de betreffende coderingen op in bijv. hexadecimalen.

Er worden in de handleiding voordoende voorbeelden gegeven om een en ander voor uw computermerk in te stellen.

Snapt de TCCR uw instellingen niet, dan genereert hij een vraagteken tot verdere stimulering van uw noeste arbeid.

BBC-gebruikers zijn in het voordeel, want voor hun merk wordt in de handleiding een en ander in soft- en hardware nader uitgewerkt.

De volgende tweelettercommando's zijn in gebruik en kunnen vanaf het toetsenbord ingevoerd worden of in de software opgenomen worden. Denk aan <CR> of CHR\*(13) als afsluiting van een stukje tekstfile.

De TCCR echoot uw invoer terug naar de computer, om te kennen te geven dat het complete commando is ontvangen.

ST =STOP	PL =PLAY	RE =RECORD
WF =WIND FAST	RF =REWIND FAST	RC =RECAP
CU =CUE	RV =REVIEW	EJ =EJECT
CC =CLEAR TAPE	MH =MOTOR HALT	MS =MOTOR START
COUNTER	DK =DISABLE KEYBRD	EK =ENABLE KEYBOARD
DS =DISABLESOUND	ES =ENABLE SOUND	RT =READ TAPECOUNTER
RM =READ MODE	PS =PROGRAM SENSOR	GT =GOTO bandpositie
TD =naar positie	SY =synchroniseer	HS =HANDSHAKE
	teller	

Deze collectie commando's geeft wel aan, dat je alle kanten opkunt met de TCCR, je kunt zelfs software en tekst of muziek op dezelfde band hebben staan en naar wens inladen of ten gehore brengen.

Wat voor toepassingsgebieden kent de TCCR?

=====

Voor circa anderhalf- tot tweeduizendgulden haalt u een luxe

audiocassetterecorder in huis die om te beginnen als zodanig te gebruiken is.

Daarnaast komt dit apparaat goed van pas in onderwijsituaties waar op natuurlijke wijze uitgesproken menselijke taal of waar muziek(fragmenten) gebruikt moet worden.

Denk aan door de computer af te nemen diktees, aan uitleg bij simulatiesoftware, bepaalde vormen van 'interactieve audio'.

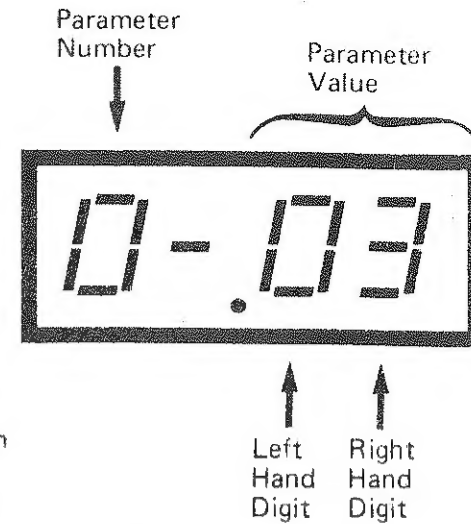
De zakenman die al zijn relaties zijn voordelige aanbiedingen wil laten weten, laat via zijn computer en zijn auto-dial-modem de clienten rondbellen, en laat de ingesproken mededeling via de TCCR horen.

Misschien is dit laatste ook een idee voor uw vereniging, om bij wanbetalers het geheugen op te frissen.

Het leren bedienen van de computer met de TCCR kan ook een toepassingsmogelijkheid zijn.

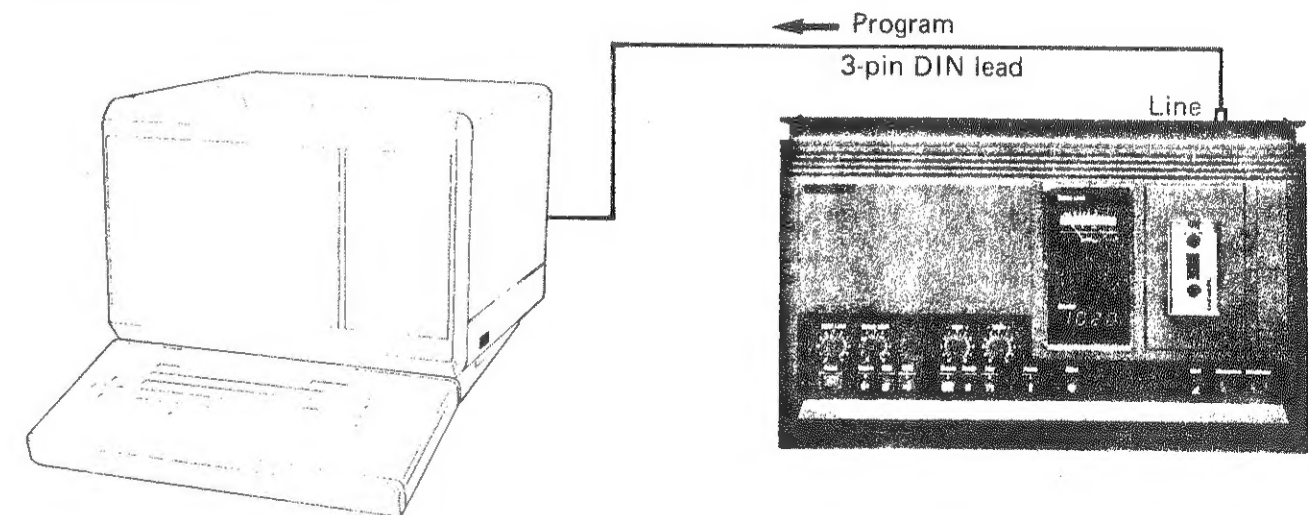
Er zijn misschien een paar nadeeltjes te noemen, namelijk de overigens uitstekende maar Engelstalige handleiding, het niet meeleveren van een microfoon, en het feit, dat je uitsluitend een cassette kunt verwijderen als het apparaat staat ingeschakeld.

Inno Broekman  
tel. 02520-29279

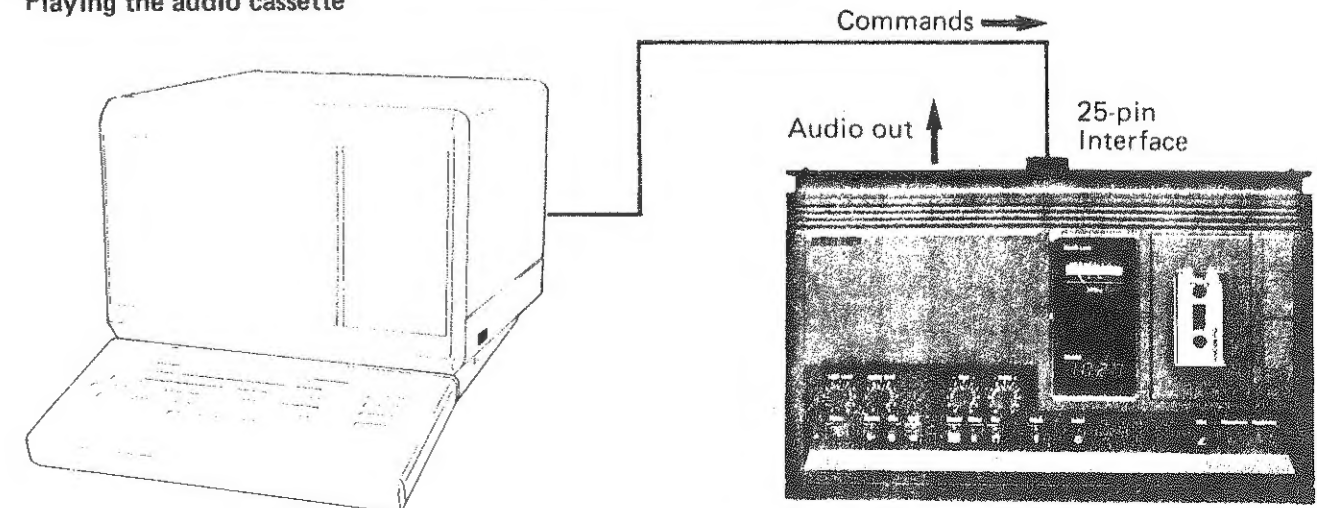


Parameter Display

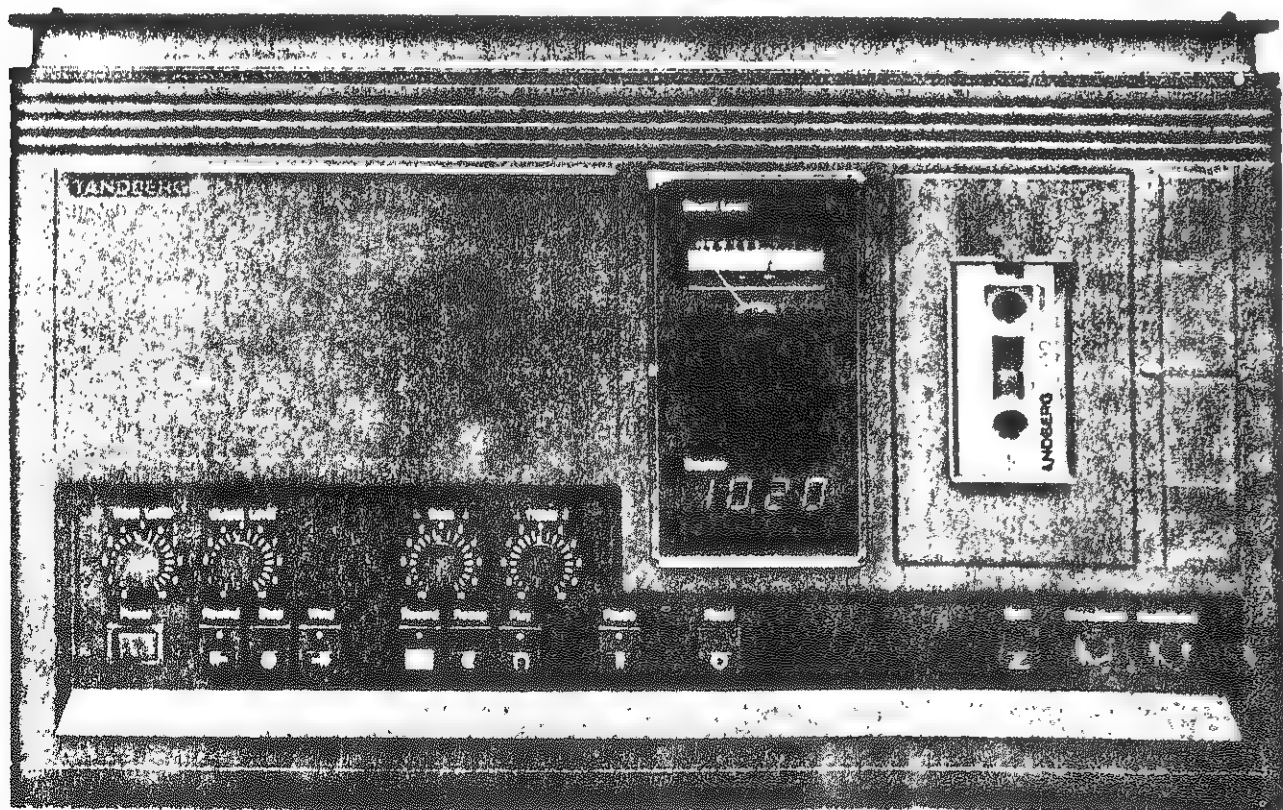
#### Loading the computer program



#### Playing the audio cassette



Power requirements	230 V ± 10%, 50 Hz 115 V ± 10%, 60 Hz
Power consumption	35 watts maximum
Operating temperature	10 - 45°C/50 - 115°F ambient (suitable for use in tropics)
Track configuration	Half track (IEC 574-10)
Tape	Compact cassette
Tape speed	1 7/8 ips (4.75 cm/s)
Speed tolerance	+ 1%
Speed variation	0.25% Peak weighted record/playback (DIN 45507) 0.08% WRMS playback (JIS)
Frequency range	40 - 14 000 Hz ± 3 dB
Signal/tape noise	58 dB A curve RMS
Motors	4
Tape transport	Dual capstan closed loop
End stop	Microprocessor controlled
Tape counter	4 digit electronic counter ("real time")
Microprocessor	8048
Erasing	> 75 dB
Computer interface	RS 232C, 25-pins male delta-connector (MIN D) Minimum configuration (no handshake) 20 mA Current loop, same connector as RS 232 Message format Communication
Inputs	Line, 5-pins DIN Sensitivity < 65 mV Impedance > 100 kohms Microphone, 1/4" jack Sensitivity 0.2 mV/200 ohms
Outputs	Line, 5-pins DIN Headphone, 1/4" stereo jack External Speaker 0.5 V/10 kohms 1.5 V/8 ohms 8 watts RMS in 4 ohms 13 watts music power
Dimensions	Width Height Depth
Weight	13 1/2" (445 mm) 5 1/2" (142 mm) 11" (280 mm) 13 1/2 lbs (6 kg)



# Implem translated

SPL V1.1 PAGE 1

IMPLEM TRANSLATED BY GEORGE CATHCART 1985

```

1          TITL      'IMPLEM TRANSLATED BY GEORGE CATHCART 1985'
2          ; /***** VERSION FOR AUDIO-CASSETTE ONLY *****/
3          ; /* USE AS BASE TO MAKE NEW MODI VERSIONS */
4          ; /* FROM SPL V1.1 MXX. */
5          ; /* DEFAULTS ARE SET AS IN SPL V1.1 M00 */
6          TRUE     EQU      0H
7          FALSE    EQU      0FFFFH
8          CHANGD   EQU      0H
9          SAME     EQU      0FFFFH
10         PARALL   EQU      0H
11         SERIAL   EQU      0FFFFH
12         NOPRIN   EQU      8000H
13         SAVED    EQU      0H
14         KILLED   EQU      0FFFFH
15         PRSENT   EQU      0H
16         NPRSNT   EQU      0FFFFH
17         USED     EQU      0H
18         NTUSED   EQU      0FFFFH
19         ;
20         DCRFTV   EQU      120H          ;fout vector - DCR
21         EFSW     EQU      135H          ;input switch - DAI
22         DCRTBL   EQU      297H          ;pointer for DCR-command's
23         RSP      EQU      299H          ;stack pointer save address
24         LOOKUP   EQU      0CA34H       ;search routine in ROM
25         TPRINT   EQU      0DD94H       ;serial print routine - DAI
26         BASNEW   EQU      0DEB8H       ;cold start BASIC
27         HEAPRS   EQU      0DECAH        ;warm start BASIC
28         GTFILE   EQU      0F0CDH       ;Get number DCR-files
29         BAUDRG   EQU      0FFF5H       ;baud rate register TMS 5501
30         ;
31         DCRFT    EQU      1BAH          ;fout variable - DCR
32         LBLMAX   EQU      3180H         ;maximum size symbol table
33         ;
34         SPL      EQU      8400H         ;begin SPL (always xy00H)
35         TOPADR   EQU      8400H         ;top symbol table (always xy00H)
36         SYMTOPI EQU      TOPADR+8H     ;MS byte top symbol table
37         SBTADR   EQU      TOPADR-2H     ;save address start symbol table
38         SRCADR   EQU      TOPADR-0AH    ;save address start source file
39         ;
40         ADROUT   EQU      SPL+2386H     ;SPL - address print
41         COLDST   EQU      SPL+136H      ;SPL - cold start
42         DCRCTR   EQU      SPL+1AA2H     ;DCR - fout control
43         DIFTBL   EQU      SPL+266AH     ;SPL - choose new tabel
44         DUMMYR   EQU      SPL+252DH     ;SPL - dummy routine
45         SPCOUT   EQU      SPL+23D3H     ;SPL - space print
46         STROUT   EQU      SPL+23AAH     ;SPL - string print
47         WAITSB   EQU      SPL+17FDH     ;SPL - wait for[ ]or[break]
48         ;
49         BDRATE   SET      0C0H          ;baud rate set 9600 Bd
50         SRCTOP   SET      SAME         ;top symbol table
51         ;
52         HIDING   SET      SAME         ;supress code when using T
53         ;
54         DFWORD   SET      SAME         ;default 16-bit notation
55         ;
56         DFBYT    SET      SAME         ;default 8-bit notation
57         ;
58         FILECH   SET      SAME         ;read/write file type (&)

```

```

59 ; adjust : character (<<33H)
60 RGDEFL SET SAME ;default lines per page
61 ; adjust : number
62 ; (0 := continuous list
63 TOFILE SET TRUE ;reset SPL for back-up
64 BASGEB SET SAME ;begin BASIC program's
65 ; adjust : address
66 BASPRG SET KILLED ;keep/destroy BASIC
67 SOUND SET SAME ;sound signal
68 ; adjust: tone-bytes
69 COLORS SET SAME ;text colour
70 ; adjust : colour-bytes
71 ; PUT "D"
72 WIDTH SET 80D ;width (columns) printer
73 ; PUT "H"
74 TABELS SET SAME ;tab positions
75 ; adjust : table-bytes
76 ; (0H forces an end)
77 ; /* DISABLE DCR COMMANDS */
78 DCR SET NPRSNT ;DCR present/not present
79 PRINTR SET SERIAL ;printer type
80 MAXADR SET SAME ;maximum start address
81 ; adjust : address
82 MINADR SET SAME ;minimum start address
83 ; adjust : address
84 DEFADR SET SAME ;default start address
85 ; adjust : address
86 SETPRT SET SAME ;SET & EQU when using T
87 ; adjust : characters
88 ADRPRT SET USED ;show memory address using 0 & i
89 ;
90 ;
91 IF SRCTOP=CHANGD
92 ORG SPL+163H
93 DW TOPADR
94 ORG SPL+169H
95 DW TOPADR-8H
96 ORG SPL+1AEH
97 DW TOPADR
98 ORG SPL+8ECH
99 DB SYMTOP
100 ORG SPL+0F19H
101 DW TOPADR
102 ORG SPL+0F28H
103 DB SYMTOP
104 ORG SPL+1134H
105 DB SYMTOP
106 ORG SPL+1794H
107 DB SYMTOP
108 ORG SPL+1833H
109 DW SRCADR
110 ORG SPL+1846H
111 DW SBTADR
112 ORG SPL+1853H
113 DW SRCADR
114 ORG SPL+1863H
115 DW TOPADR-1H
116 ORG SPL+18F4H

```

```

117 DW TOPADR
118 ORG SPL+1904H
119 DW SRCADR
120 ORG SPL+190EH
121 DW SBTADR
122 ORG SPL+1995H
123 DB SYMTOP
124 ORG SPL+19E7H
125 DW TOPADR
126 ORG SPL+1A35H
127 DW SBTADR
128 ORG SPL+1A39H
129 DW SBTADR
130 ORG SPL+1A3FH
131 DW SRCADR
132 ORG SPL+1A43H
133 DW SRCADR
134 ORG SPL+1A52H
135 DW SBTADR
136 ORG SPL+1A5BH
137 DW SBTADR
138 ORG SPL+1A60H
139 DW SRCADR
140 ORG SPL+1AE2H
141 DB SYMTOP
142 ORG SPL+1B07H
143 DW TOPADR
144 ORG SPL+1B6CH
145 DB SYMTOP
146 ORG SPL+1BA0H
147 DB SYMTOP
148 ORG SPL+1BF6H
149 DW TOPADR
150 ORG SPL+1CB4H
151 DB SYMTOP
152 ORG SPL+1FBAH
153 DW TOPADR-1H
154 ORG SPL+1FD1H
155 DW TOPADR
156 ORG SPL+2532H
157 DW TOPADR-LBLMAX
158 ENDIF
159 ;
160 IF DEFADR=CHANGD
161 ORG SPL+240H
162 DW 400H
163 ENDIF
164 ;
165 IF MAXADR=CHANGD
166 ORG SPL+234H
167 DW TOPADR
168 ENDIF
169 ;
170 IF MINADR=CHANGD
171 ORG SPL+22BH
172 DW 2ECH
173 ENDIF
174 ;

```

```

175 ;
176 ;This list contains directives recognised only by SPL
177 ;and are not included in the object code. LST and UNL are
178 ;not included here, as they are treated as normal mnemonics.
179 ;
180 IF HIDING=CHANGD
181 ORG SPL+27D3H
182 CPI 20H ;###
183 RZ
184 CPI 30H ;control
185 RZ
186 CPI 40H ;IF
187 RZ
188 CPI 49H ;ELSE
189 RZ
190 CPI 52H ;ENDIF
191 RZ
192 CPI 5BH ;call-macro
193 RZ
194 CPI 64H ;MACRO
195 RZ
196 CPI 6DH ;MEND
197 RET
198 ENDIF
199 ;
200 IF SETPRT=CHANGD
201 ORG SPL+13C5H
202 DB 40H
203 DS 4H
204 DB '='
205 ENDIF
206 ;
207 ORG SPL+1F29H
208 IF ADRPRT=USED
209 CALL ADROUT
210 ELSE
211 NOP
212 NOP
213 NOP
214 ENDIF
215 ;
216 IF DFWORD=CHANGD
217 ORG SPL+111H
218 DB 'H'
219 ENDIF
220 ;
221 IF DFBYT=CHANGD
222 ORG SPL+112H
223 DB 'h'
224 ENDIF
225 ;
226 IF FILECH=CHANGD
227 ORG SPL+186CH
228 DB '&'
229 ORG SPL+18C5H
230 DB '&'
231 ENDIF
232 ;

```

```

233 IF BASGEB=CHANGD
234 ORG SPL+146H
235 DW EIND
236 ENDIF
237 ;
238 IF RGDEFL=CHANGD
239 PUT "d"
240 ORG SPL+10EH
241 DB 58D ; 2 + 58 LINES/PAGE
242 PUT "h"
243 ENDIF
244 ;
245 IF TOFILE=TRUE
246 ORG SPL+131H
247 DW COLDST
248 ENDIF
249 ;
250 ORG SPL+152H
251 IF BASPRG=SAVED
252 DW HEAPRS
253 ELSE
254 DW BASNEW
255 ENDIF
256 ;
257 ;To supress signal tone: 1st byte 0FFH
258 IF SOUND=CHANGD
259 ORG SPL+2A74H
260 DB 4H,0C4H,9H,0AH,0FH,0FFH
261 ENDIF
262 ;
263 IF COLORS=CHANGD
264 ORG SPL+2A62H
265 DB 8H,0H,0CH,0H
266 ENDIF
267 ;
268 ORG SPL+2500H
269 PUT "D"
270 IF WIDTH<78D
271 DW DUMMYR
272 DS 1D
273 DB 6H
274 ELSE
275 DW SPCOUT
276 DS 1D
277 DB 6H
278 ENDIF
279 PUT "H"
280 ;
281 IF TABELS=CHANGD
282 PUT "d"
283 ORG SPL+29FDH ;By BV COMMAND
284 TBEDIT DB 8D,17D,32D,0D ; E
285 TBLIST DB 6D,14D,23D,38D,0D ; L
286 TBHLST DB 6D,13D,21D,31D,46D,0D ; DIVERSE
287 PUT "h"
288 ENDIF
289 ;
290 IF DCR=PRSENT

```

```

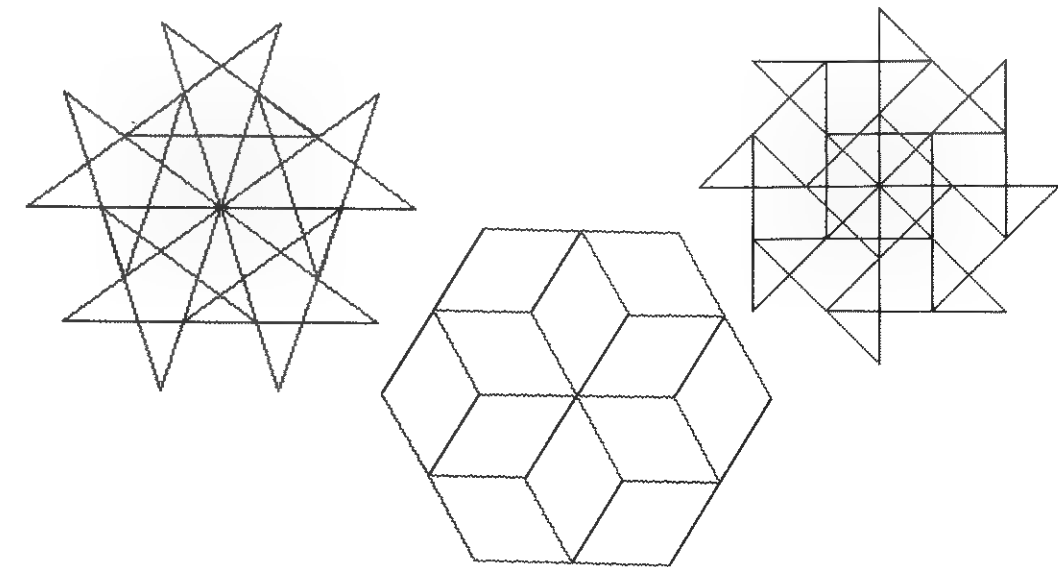
291      ORG      SPL+2821H
292      DW      SPL+2E8CH
293      ORG      SPL+2E8CH
294      LXI H    DCRFT
295      SHLD    DCRFTV
296      LHLD    DCRTBL
297 DCRNWT MVI E  1H
298      XRA A
299      STA     EFSW
300      CALL    LOOKUP
301      CNC     DIFTBL
302      JNC     DCRNWT
303      MOV E,M
304      INX H
305      MOV D,M
306      LXI H    DCRCTR
307      PUSH H
308      LXI H    0H
309      DAD SP
310      SHLD    RSP
311      PUSH D
312      CALL    6TFIL
313      RET
314      ELSE
315      ORG      SPL+2821H
316      DW      SPL+2671H
317      ORG      SPL+2E8CH
318      ENDIF
319 PRTFNC NOP
320 ;
321 ;Below is an example of a software
322 ;interface for a parallel printer
323      IF      PRINTR=PARALL
324      ORG      SPL+20E0H
325      DW      PRINIT
326      ORG      SPL+248FH
327      DW      PRINT
328      ORG      PRTFNC
329 PRINT PUSH PSW
330 HARDOU LDA  0FE02H      ;PORT C BIT 4 IS 'BUSY' SIGNAL
331      ANI  10H          ;PRINTER BUSY ?
332      JNZ  HARDOU      ;YES - TRY AGAIN
333      POP PSW          ;NO - RESTORE DATA IN A
334      STA  0FE00H      ;STORE IN PORT A (PRINTER)
335      PUSH H
336      LXI H    0FE01H      ;PORT B
337      MVI M    70H          ;STROBE - BIT 7 LOW
338      MVI M    0F0H        ;RESET STROBE
339      POP H
340      RET
341 PRINT LXI H    PRTMES      ;Ready? message address
342      CALL    STROUT      ;output to screen
343      CALL    WAITSB     ;wait for [ ] (confirmation)
344      ;                or [break] (stop)
345      MVI A    88H          ;MODE 0,A OUT,B OUT,CH IN,CL OUT
346      STA  0FE03H      ;8255 CONTROL REGISTER
347      LXI H    0FE00H      ;PORT A ADDRESS (DATA)
348      MVI M    0H          ;RESET

```

```

349      INX H          ;PORT B (BIT 7 IS STROBE)
350      MVI M    0F0H      ;RESET STROBE
351      RET
352      ENDIF
353 ;
354      IF      PRINTR=SERIAL
355      ORG      SPL+20E0H
356      DW      INIPRT
357      ORG      SPL+248FH
358      DW      TPRINT
359      ORG      PRTFNC
360 INIPRT LXI H    PRTMES      ;print 'Ready?'
361      CALL    STROUT      ;output to screen
362      CALL    WAITSB     ;wait for confirmation
363      MVI A    BDRATE     ;get baud rate and store
364      STA  BAUDRG        ;in the baud rate register
365      RET
366      ENDIF
367 ;
368      IF      PRINTR=SERIAL?PRINTR=PARALL
369 PRTMES DB  0DH
370      DB  'Ready?'
371      DB  0DH,0H
372 DUM1  PUT  "H"
373 EIND  SET  DUM1
374      ENDIF
375 ;
376      IF      PRINTR="PARALL"!PRINTR="SERIAL
377      ORG      SPL+20E0H
378      DW      HRDERR
379      ORG      SPL+248FH
380      DW      DUMYR
381 EIND  SET  PRTFNC
382      ENDIF
383 ;
384      END

```



# Puissance 5

PAGE 1 -- \*\*\* PUISSANCE 5 [10\*10] \*\*\*

-- FLST V2.2

```
10 REM *** PUISSANCE 5 [10*10] ***
11 REM ~~~~~
12 REM
13 REM --- copied by Di Marco Emmanuel ---
14 REM
16 CLEAR 5000:COLORT 8 0 0 0:MODE 0:PRINT CHR$(12);
18 KKK=0.0
20 GOSUB 9100:REM MENU
25 GOSUB 100:REM INIT JEU
35 GOSUB 600:REM PROLOGUE
40 GOSUB 700:REM PARTIE
45 PRINT " Voulez-vous jouer une autre partie [O/N] ";
48 GOSUB 9000:REM QUI OU NON
50 IF R$="O" OR R$="o" THEN 35
52 CURSOR 0,CURY
55 GOSUB 900:REM END
60 END

100 REM ~~~~~
101 REM *** INITIALISATIONS DU JEU ***
102 REM ~~~~~
120 DIM A(99.0),B(99.0),V(21.0),B$(5.0),A$(9.0)
260 FOR J=0.0 TO 21.0
270 1 V(J)=0.0
280 NEXT J
290 FOR J=0.0 TO 4.0
300 1 READ V(J)
310 NEXT J
330 DATA .01,.03,.5,10,1000
340 FOR J=5.0 TO 20.0 STEP 5.0
350 1 READ V(J)
360 NEXT J
380 DATA .1,2,100,1000000
410 READ B$(0.0),B$(1.0),B$(5.0)
420 DATA ".", "0", "X"
430 FOR J=0.0 TO 9.0
440 1 READ A$(J)
450 NEXT J
460 DATA "00", "10", "20", "30", "40", "50", "60", "70", "80", "90"
470 RETURN

600 REM ~~~~~
610 REM *** PROLOGUE DE LA PARTIE ***
620 REM ~~~~~
630 FOR J=0.0 TO 99.0
640 1 A(J)=0.0:B(J)=0.0
650 NEXT J
660 N=0.0
680 F=0.0
690 RETURN

700 REM ~~~~~
710 REM *** PARTIE ***
715 REM ~~~~~
720 PRINT " Voulez-vous commencer [O/N] ";
```

PAGE 2 -- \*\*\* PUISSANCE 5 [10\*10] \*\*\*

-- FLST V2.2

```
722 GOSUB 9000:REM QUI OU NON
725 CURSOR 0,21:GOSUB 4000:REM CLEAR
726 GOSUB 8000:REM AFFICHAGE
730 IF R$="O" OR R$="o" THEN 770
740 GOSUB 1000:REM PROGRAMME
760 IF F<>0.0 THEN 790
770 GOSUB 2000:REM ADVERSAIRE
780 IF F=0.0 THEN 740
790 IF F=1.0 THEN PRINT " J'ai gagne en ";GOTO 850
792 IF F=(-1.0) THEN PRINT " Vous avez gagne en ";GOTO 850
794 PRINT " Partie nulle apres ";
850 N=INT(N/2.0+0.6)
855 PRINT N;" coups."
860 RETURN

900 REM ~~~~~
910 REM *** END ***
912 REM ~~~~~
915 PRINT " Au revoir !"
920 PRINT :PRINT
930 RETURN

1000 REM ~~~~~
1010 REM *** LE PROGRAMME JOUE ***
1020 REM ~~~~~
1030 FOR J=0.0 TO 99.0:B(J)=0.0:NEXT J
1090 C=1.0:J1=0.0:J2=5.0:K1=0.0:K2=90.0:GOSUB 1500:REM QUINTUPLETS
1100 C=9.0:J1=4.0:J2=9.0:K1=0.0:K2=50.0:GOSUB 1500
1110 C=10.0:J1=0.0:J2=9.0:K1=0.0:K2=50.0:GOSUB 1500
1120 C=11.0:J1=0.0:J2=5.0:K1=0.0:K2=50.0:GOSUB 1500
1170 S=-1.0
1180 Q=0.0
1200 FOR J=0.0 TO 99.0
1210 1 IF A(J)<>0.0 OR B(J)<S THEN 1230
1220 1 IF B(J)>S THEN C=J:S=B(C):Q=1.0:GOTO 1230
1222 1 Q=Q+1.0
1224 1 IF Q*RND(1.0)<1.0 THEN C=J
1230 NEXT J
1235 IF KKK=(-1.0) THEN RETURN
1240 CURSOR 0,9:GOSUB 4000:REM CLEAR
1245 PRINT " Je joue en : ";C
1250 A(C)=5.0
1260 XA=C-10.0*INT(C/10.0)
1262 YA=INT(C/10.0)
1264 GOSUB 8500:REM POSIT
1266 PRINT "O":CURSOR Q,H
1270 N=N+1.0
1280 IF N>100.0 OR (S=0.0 AND N>1.0) THEN F=2.0
1290 IF S>=V(20.0) THEN F=1.0
1300 RETURN

1500 REM ~~~~~
1510 REM *** EXPLORATION DES QUINTUPLETS ***
1515 REM ~~~~~
1520 C4=4.0*C
```

```

1530   FOR J=J1 TO J2
1540   1   FOR K=K1 TO K2 STEP 10.0
1550   2   A=J+K:B=A+C4:Z=0.0
1570   2   FOR P=A TO B STEP C:Z=Z+A(P):NEXT P
1580   2   Q=V(Z)
1590   2   IF Q<>0.0 THEN FOR P=A TO B STEP C:B(P)=B(P)+Q:NEXT P
1600   1   NEXT K
1610   NEXT J
1620   RETURN

2000   REM *****
2001   REM *** L'ADVERSAIRE JOUE ***
2002   REM *****
2007   CURSOR 0,7:GOSUB 4000:REM CLEAR
2010   INPUT " Votre coup [ex:46] ";A:PRINT
2030   IF A<0.0 OR A>99.0 OR A<>INT(A) THEN PRINT " Un entier entre 0 et
C     99 SVP ...";:WAIT TIME 50:GOTO 2007
2040   IF A(A)<>0.0 THEN PRINT " Case deja occupee ...";:WAIT TIME 50:
C     GOTO 2007
2060   A(A)=1.0
2070   XA=A-10.0*INT(A/10.0)
2072   YA=INT(A/10.0)
2074   GOSUB 8500:REM POSIT
2076   PRINT "X":CURSOR Q,H
2080   N=N+1.0
2100   IF N=100.0 THEN F=2.0
2110   IF B(A)>=V(4.0) THEN F=-1.0
2220   RETURN

4000   REM *****
4001   REM *** CLEAR ***
4002   REM *****
4010   H=CURY
4015   FOR T=1.0 TO H
4020   1   PRINT "
C     1   "
4030   NEXT T
4040   CURSOR 0,H
4050   RETURN

8000   REM *****
8001   REM *** AFFICHAGE ***
8002   REM *****
8010   PRINT "           0 1 2 3 4 5 6 7 8 9"
8020   FOR J=0.0 TO 9.0
8030   1   PRINT "           ";A$(J);" ";
8040   1   FOR K=0.0 TO 9.0
8050   2   PRINT B$(A(10.0*J+K));" ";
8060   1   NEXT K
8070   1   PRINT
8080   NEXT J
8090   RETURN

8500   REM *****
8501   REM *** POSIT ***

```

```

8502   REM *****
8505   H=CURY:Q=CURX
8510   CURSOR 18+XA*2,20-YA
8520   RETURN

9000   REM *****
9001   REM *** OUI DU NON ***
9002   REM *****
9005   Q=CURX:H=CURY
9010   INPUT R$:PRINT R$=LEFT$(R$,1)
9020   IF R$<>"0" AND R$<>"o" AND R$<>"N" AND R$<>"n" THEN 9040
9030   RETURN
9040   PRINT " OUI OU NON "
9050   CURSOR Q,H
9060   PRINT "           "
9070   CURSOR Q,H
9080   GOTO 9010

9100   REM *****
9101   REM *** MENU ***
9102   REM *****
9103   PRINT "           PUISSANCE 5 [10*10]"
9105   PRINT "           -----"
9110   PRINT :PRINT
9120   PRINT " Le but du jeu est d'aligner 5 de vos pions, soit:"
9130   PRINT " - horizontalement"
9140   PRINT " - verticalement"
9150   PRINT " - en diagonale"
9160   PRINT :PRINT
9170   PRINT " Vous avez les X et l'ordinateur les O."
9180   PRINT
9220   PRINT " BONNE CHANCE !!!"
9230   PRINT
9240   RETURN

```

---

**HINT**

For users of a COMMODORE FLOPPY DISC DRIVE equipped with a interface developed by J. BOERRIGTER, P. WIEGRS e.a. the following short program can be used to obtain a DIRECTORY page on a printer with parallel connection. Therefore load or type in program, insert floppy ; RUN ; push LEFTCURSORKEY after the page of which the printout is wanted, is displayed.

```

5 CALLM #F000: REM DIR : /P
10 FOR L=0 TO 22 : FOR T = 0 TO 30
20 A = PEEK(#BFE7-134*L-2*T):?CHR$(A);
30 NEXT T
40 PRINT CHR$(10);: CURSOR 0,0 : NEXT L

```

M. Van Schoor



# Cirkelschijfsegment

PAGE 01 -- CIRKELSCHIJSSEGMENT

```

1000 REM +++ DEMO 0044-05-00 ~ V1.0
1010 REM +++ tekenen cirkelschijf segment
1020 REM +++ DAI
1100 CLEAR 2000:MODE 6A
1110 PRINT "Kies: Alleen een cirkelsegment < 1 >"
1120 INPUT " Segment in cirkelschijf < 2 > ";X:PRINT
1130 IF X<1 OR X>2 GOTO 1100
1200 INPUT "Horizontale middelpuntscoördinaat ZMX 10-325 ";ZMX:
PRINT
1300 INPUT "Verticale middelpuntscoördinaat ZMY 10-245 ";ZMY:
PRINT
1400 IF X=2 THEN INPUT "Cirkelschijf in de kleur ZKL
0/3/5/14 ";ZKL1:PRINT
1450 INPUT "Segment in de kleur ZKL 0/3/5/14 ";ZKL2:
PRINT
1500 INPUT "Met de straal ZR! 10-125 ";ZR!:
PRINT
1600 INPUT "Loze hoek in radialen voor segment ZH1! 0- 6 ";ZH1!:
PRINT
1700 INPUT "Volle hoek in radialen voor segment ZH2! 0- 6 ";ZH2!:
PRINT
1800 MODE 6:COLORG 0 3 5 14
1810 IF X=2 THEN ZKL=ZKL1:GOSUB 31612:REM tekenen cirkelschijf
1900 ZKL=ZKL2:GOSUB 31616:REM tekenen cirkelschijfsegment
2000 WAIT TIME 200:GOTO 1100
17500 END

26322 REM %%% tekenen cirkelschijf %%%
26326 IF ZR!<>ZMR! THEN GOSUB 26334
26327 FOR ZQ=0 TO 159:FOR ZN=-1 TO 1 STEP 2
26328 2 DRAW ZMX-ZCX(ZQ),ZMY+ZN*ZCY(ZQ) ZMX+ZCX(ZQ),ZMY+ZN*ZCY(ZQ) ZKL
26329 NEXT ZN:NEXT ZQ
26330 RETURN

26333 REM %% berekening cirkelconstanten %%
26334 ZQ=0:DIM ZCX(159),ZCY(159):REM startroutine
26335 FOR ZA!=0.0 TO PI/2.0 STEP PI/316.0
26336 1 ZCX(ZQ)=ZR!*SIN(ZA!):ZCY(ZQ)=ZR!*COS(ZA!):ZQ=ZQ+1
26337 NEXT ZA!:ZMR!=ZR!:RETURN

26370 REM ##### DIDACOM 0044-05-00 #####
26371 REM #####
26372 REM ##### tekenen cirkelschijf segment #####
26373 REM #####
26374 REM ##### versie 1.0 DAI - basic #####
26375 REM #####
26376 ZS!=1.0/ZR!:FOR ZQ!=ZH1! TO ZH2! STEP ZS!
26377 1 ZCX=ZR!*SIN(ZQ!):ZCY=ZR!*COS(ZQ!)
26378 DRAW ZMX,ZMY ZMX+ZCX,ZMY+ZCY ZKL:NEXT ZQ!:RETURN
31612 GOTO 26326:REM tekenen cirkelschijf 0042-05-00
31616 GOTO 26376:REM tekenen cirkelschijf segment 0044-05-00

```

# Maasgrafieken

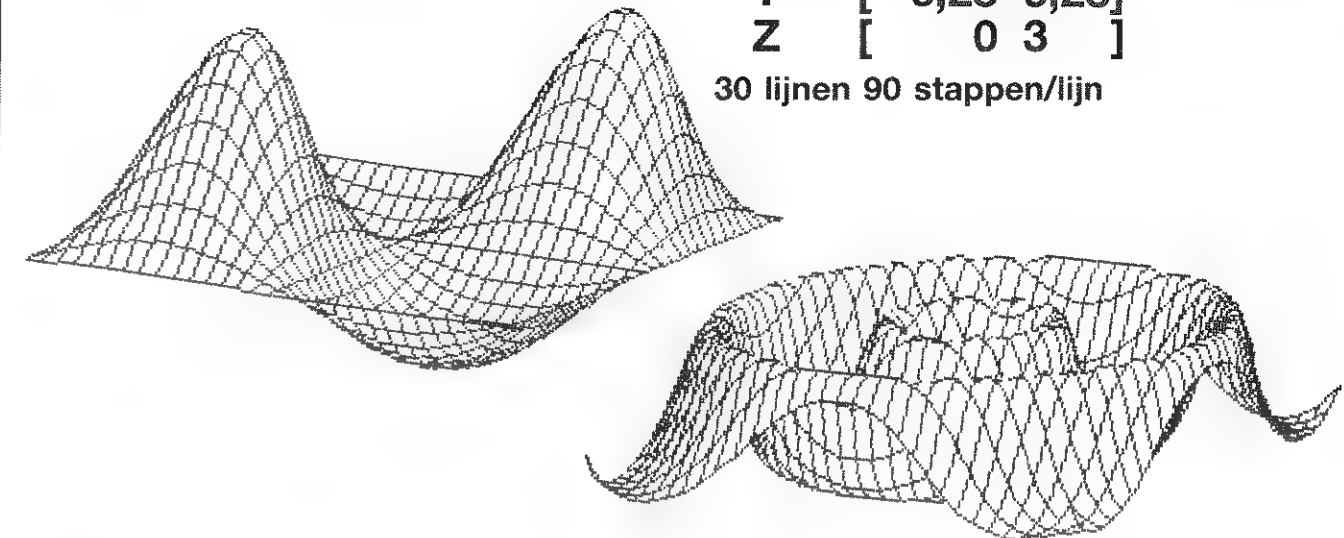
$$Z = \text{EXP}(-\text{SIN}(X) \star \text{SIN}(Y))$$

$$X \quad [-6,28 \quad 6,28]$$

$$Y \quad [-6,28 \quad 6,28]$$

$$Z \quad [ \quad 0 \quad 3 \quad ]$$

30 lijnen 90 stappen/lijn



$$Z = \text{EXP}(-\text{COS}(X \star X + Y \star Y))$$

```

1 REM MAASGRAFIEKEN - TRANSCRIPTIE VAN ANONIEM APPLE II-PROG.
2 REM C.W.A. van Dijk - Kampen
3 REM
10 MODE 0:GOTO 800
20 F=SIN(X)*COS(Y)
30 IF F>Z2 THEN F=Z2
40 IF F<Z1 THEN F=Z1
50 RETURN
100 Y=W0:X=U0:GOSUB 20
110 W4=S2*(W0-W1):X=S4*(U0-U1)-C1*W4+R1:Y=S3*(F-Z1)-S1*W4+R2:
W=1.0:IF P8=0.0 THEN RETURN
120 W6=Y:U6=X:D=1.0:Y0=-S1*W4+R2-1.0:Y9=S6:IF X>S9 THEN
Y9=S5*X+S7
130 IF Y0<Y9 THEN RETURN
140 V2=(Y0-Y9)/EA:FOR K=0.0 TO EA:Y8=Y0-K*V2
150 Y=(R2-Y8)/S1:X=(U6+C1*Y-R1)/S4+U1:Y=Y/S2+W1:GOSUB 20:
F=(F-Z1)*S3+Y8:IF F<W6 THEN W=E0
160 IF F>W6 THEN D=E0
170 IF W+D=E0 THEN 200
180 NEXT K
190 W=1.0
200 Y=W6:X=U6:RETURN
210 GOSUB 100:IF W=V9 THEN 330
220 IF T0=E2 THEN 280
230 W9=W0:W8=W9-W3
240 IF S2*(W9-W8)<EZ THEN 320
250 W0=(W8+W9)/E2:GOSUB 100:IF W<>V9 THEN W9=W0
260 IF W=V9 THEN W8=W0
270 GOTO 240
280 U9=U0:U8=U9-U3
290 U0=(U8+U9)/E2:GOSUB 100:IF W<>V9 THEN U9=U0
300 IF W=V9 THEN U8=U0
310 IF S4*(U9-U8)>=EZ THEN 290
320 W=V9:V9=1.0-W

```

```

330 IF WK>0.0 THEN DRAW XA,YA X,Y 15
340 XA=X:YA=Y:RETURN
350 FOR K1=E0 TO N1:U0=U1+K1*U3:W0=W1:GOSUB 100:XA=X:YA=Y:V9=W:
    FOR K2=E1 TO N2:W0=W1+K2*W3:GOSUB 210:NEXT:NEXT
360 T0=E2:U3=(U2-U1)/N2:W3=(W2-W1)/N1
370 FOR K1=E0 TO N1:W0=W1+K1*W3:U0=U1:GOSUB 100:XA=X:YA=Y:V9=W:
    FOR K2=E1 TO N2:U0=U1+K2*U3:GOSUB 210:NEXT:NEXT
380 PRINT :INPUT " ";A$:MODE 6A:INPUT "DO YOU WISH TO MAKE ANY
    CHANGES";A$:PRINT :IF A$<>"" THEN GOSUB 1060:GOTO 820
390 END

800 REM
810 GOSUB 1000
820 COLORG 0 15 5 12:MODE 6
830 S4=A1/(U2-U1):S2=B1/(W2-W1):S3=C2/(Z2-Z1):C1=0.519615:S1=0.3:
    S5=0.577353:S9=A1-C1*B1+R1
840 S6=R2-S1*B1:U3=(U2-U1)/N1:W3=(W2-W1)/N2:S7=R2-S5*(A1+R1)
850 E0=0.0:E1=1.0:E2=2.0:EZ=0.5:T0=E1
860 GOTO 350

1000 A1=182.0:B1=182.0:C2=136.0:R1=96.0:R2=55.0:N1=15.0:N2=30.0:
    EA=10.0:U1=-6.28:U2=6.28:W1=-6.28:W2=6.28:Z1=0:Z2=3.0:
    DIM A$(3.0)
1010 PRINT CHR$(12)
1020 PRINT "PROGRAM PLOTS F AS FUNCTION OF x AND y"
1030 PRINT "STANDARD FUNCTION IS:"
1040 LIST 20-20
1050 PRINT "ANSWER TO QUESTIONS WITH NO=RETURN AND YES=ANY OTHER
    KEY"
1060 PRINT "DO YOU WISH TO CHANGE THE FUNCTION";:INPUT A$:PRINT
1070 IF A$<>"" THEN PRINT "INSERT YOUR FUNCTION AT LINE 10":
    PRINT "TYPE RUN TO CONTINUE":END
1080 PRINT "DO YOU WISH TO CHANGE"
1090 PRINT TAB(6);"THE ORIGINS OR THE LENGTHS OF THE AXES ";:
    INPUT A$:PRINT
1100 IF A$="" THEN 1130
1110 PRINT "DEF. SCREEN X- AND Y-COORDINATES OF ORIGIN ";R1;R2:
    INPUT R1,R2:PRINT
1120 PRINT "SCREEN LENGTHS OF X-,Y- AND Z-AXIS ";A1;B1;C2:
    INPUT A1,B1,C2:PRINT
1130 PRINT "DO YOU WISH TO CHANGE THE INTERVALS X, Y OR Z ";:
    INPUT A$:PRINT :IF A$="" THEN 1170
1140 PRINT "BEGIN AND END POINT OF X-INTERVAL ";U1;U2:
    INPUT U1,U2:PRINT
1150 PRINT "BEGIN AND END POINT OF Y-INTERVAL ";W1;W2:
    INPUT W1,W2:PRINT
1160 PRINT "BEGIN AND END POINT OF Z-INTERVAL ";Z1;Z2:
    INPUT Z1,Z2:PRINT
1170 PRINT "DO YOU WISH TO CHANGE THE NUMBER OF LINES OR THEN
    NUMBER OF PLOTSTEPS PER LINE (DEFAULT=15,30) ":INPUT A$:?
1180 IF A$="" THEN 1200
1190 INPUT "NUMBER OF LINES, NUMBER OF PLOTSTEPS PER LINE";N1,N2
1200 INPUT "DO YOU WISH TO OMIT THE HIDDEN LINES (0=NO, 1=YES)";
    P8:PRINT
1210 IF P8=1.0 THEN PRINT "DO YOU WISH TO CHANGE THE PRECISION OF
    THE VISIBILITY-DETECTION ROUTINE (DEFAULT= 10 STEPS)":
    INPUT A$:PRINT
1220 IF P8=1.0 THEN IF A$<>"" THEN INPUT "NUMBER OF STEPS TO BE
    USED FOR VISIBILITY - DETECTION";EA:PRINT
1230 RETURN

```

# International DAICLIC Club

## EDITO 7 :

Dans ce numéro, vous trouverez la suite tant attendue de l'article d'IDC BORDEAUX sur le Minitel, vous pourrez également donner la parole à votre DAI grâce à l'article de Pascal Janin qui nous décrit son synthétiseur vocal ! Un petit montage amusant à réaliser pendant les vacances ...

Vous apprendrez aussi la création du club EMC dont les activités autour de l'ordinateur Atari ST sont désormais opérationnelles ! Nous avons besoin de volontaires pour les différentes sections que nous voulons créer. Contactez la rédaction pour de plus amples renseignements.

Le serveur "MN2 PARIS" est opérationnel depuis fin mai ! Nous espérons que vous serez nombreux à vous y connecter, car du succès de ce serveur dépendra de la qualité et de la quantité des services offerts. Nous vous rappelons que si vous n'êtes pas encore l'heureux possesseur d'un modem, vous pouvez en commander par correspondance auprès de la rédaction (voir DAICLIC précédent) .

Suite aux grèves ayant affecté le service postal, certains membres n'ont reçu leur "Daiclic" numéro 6 qu'après un retard pouvant atteindre un mois ! Nous tenons à vous signaler que TOUTES les revues ont été envoyés le 30 avril (le cachet de la poste faisant foi ...).

Plusieurs réclamations nous ont appris que certains d'entre vous ont reçu à leur grand étonnement une deuxième fois un "Daiclic" numéro 5 à la place du numéro 6. Après enquête, nous nous sommes aperçu que dans la livraison des numéros 6 s'étaient glissés une quantité indéterminée de numéros 5 !!! Les membres dans ce cas sont priés de nous renvoyer ce numéro "en trop" et nous nous empresserons de leur expédier le numéro 6.

Toujours dans un souci d'améliorer la qualité de votre revue, la plupart des articles et schémas de ce numéro, ont été imprimés à l'aide d'une ImageWriter 2 (l'imprimante du Mac Intosh). Nous tenons particulièrement à remercier M.Giacomo Infantino pour l'aide qu'il nous a ainsi apportée dans la réalisation de la revue.

Le logiciel "Mailing List" est maintenant disponible sur disquette Commodore VC1541 au prix de 1750 FB, avis aux amateurs...

IDC vous souhaite de passer d'agréables vacances (bien méritées).

La rédaction.

## EUROPEAN MICRO CLUB (: EMC ASBL)

Nous avons le plaisir de vous annoncer la naissance du petit frère d'IDC **EMC** !!

**EMC** a été créé pour répondre à un besoin de plus en plus insistant de la part de nos membres: l'ouverture à d'autres machines que le **DAI**. En effet, depuis quelques mois (et même plus longtemps pour certains), beaucoup d'entre-vous changent d'ordinateur pour diverses raisons.

Le fonctionnement d'**EMC** sera "calqué" sur celui d'**IDC**, il proposera à peu de choses près les mêmes services: revues, réseaux, programmes, réunions, rencontres internationales, etc. La principale différence tient dans le fait qu'il n'y a pas de limitation théorique dans le nombre de marques d'ordinateur !

Cependant, chaque nouvelle section, avant d'être créée sera soigneusement étudiée et préparée. **EMC ATARI** est la première section. Cette section tourne autour de l'**ATARI ST (130 / 260 / 520 / 1040 / 2080, ...)** et est déjà opérationnelle au niveau régional ! **EMC LIEGE** propose en effet déjà des réunions (en parallèle aux réunions d'**IDC LIEGE** !), une programmathèque, une bibliothèque, un service d'achat à prix clubs (matériel **ATARI**, disquettes, modems, etc.) !

Les prochaines sections prévues sont: **EMC PC (IBM PC et compatibles)**, **EMC MACINTOSH**, **EMC AMIGA**, ....

Les responsables de **EMC** sont les mêmes que ceux d'**IDC**, vous ne serez donc pas trop dépaysés !... Pour tout renseignement sur "**EMC**" ou "**EMC ATARI**", contactez Christian POELS à la rédaction.

Pour ceux qui n'ont pas l'intention de changer d'ordinateur, qu'ils se rassurent: **IDC** continuera de leur fournir normalement les services auxquels ils sont maintenant habitués! Le **DAI** est une machine qui mérite amplement de continuer à être exploitée et utilisée. Ses caractéristiques n'ont rien à envier à la plupart des ordinateurs "actuels" et continue d'être utilisé dans certains domaines professionnels, tels la vidéo, l'enseignement, etc !

EMC A.S.B.L.

## PETITES ANNONCES

- Vends **DCR** + une quarantaine de cassettes **DCR** (: 700 programmes) + cassettes audio: 10000 FB. Vends **DAI** "colérique": prix à débattre. Benoit Gortz, Rue de l'Athénée 12, B-6000 Charleroi
- Vends carte **PAL DAI**: 2500 FB. Christian POELS (rédaction).

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

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

## A L'ATTENTION DE TOUS LES CLUBS ASSOCIES

### Remarque

Nous sommes très heureux de vous annoncer que grâce à notre bien aimé Sébastien DUBOURG *Gloria in Excelcis DEO*, notre programme didactique sur l'émulation MINITEL est enfin terminé. Il est cependant trop long pour être publié in extenso dans **DAICLIC** mais il sera envoyé sur simple demande à tout membre le désirant (n'oubliez pas l'enveloppe timbrée pour la réponse...) Vous verrez ça vaut le déplacement.

**96 pages de listing source S.P.L.**

**15 Kb de pur langage machine !**

Profitez-en pour suivre avec plus d'attention notre série d'articles sur le MINITEL !! Nous n'oublierons pas de commenter les multiples astuces ayant contribué à ce logiciel dans des articles annexes SOYEZ ATTENTIFS !

De très bonnes nouvelles !!

### Du côté du matériel

Grâce à un de nos adhérents enseignant, M. IMMER Alain, 6 Hameau de GALAU, F-47520 Le Passage d'AGEN (53 96 48 05), il vous est possible d'acheter les imprimantes que le club IDC BORDEAUX vous recommandait déjà. **MANNESMANN TALLY 80 S** et **85** à des prix encore plus bas, à la CAMIF, (sérieux dans l'après-vente), et équipées d'interfaces de type centronics, dont vous savez maintenant équiper votre **DAI** depuis le super montage des frères LEGRY dans **DAICLIC 5** (sinon achetez l'interface RS 232 !)

MANNESMANN TALLY MT 80S = 2520 FRF

MANNESMANN TALLY MT 85 = 3670 FRF

Et rappelons les livres proposés

Le secteur bibliothèque du club se développe grâce à la coopération d'un de nos membres qui a traduit de l'anglais en français, de façon très professionnelle, le complet du **KEN-DOS**

En raison du don bénévole de ce travail au club, ce livre vous est offert à prix coûtant pour 40FF port compris. **MANUEL KEN-DOS en FRANCAIS**

Vu le nombre des demandes, une réédition du '**TOS COMMENTE**' a été faite. Que tous ceux qui n'avaient pu l'obtenir l'an dernier me réécrivent. Le prix lui n'a pas augmenté: 150FF port compris (= p.c.).

IDC BORDEAUX  
s/c DELANNAY Bruno  
Res les Acacias Bt. B3  
Avenue de SAIGE  
33600 PESSAC - FRANCE

## IDC LIEGE NEWS

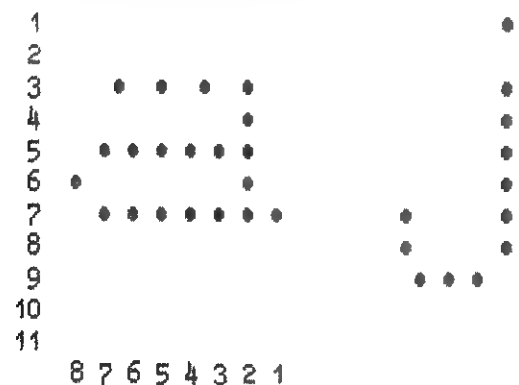
Chers Amis

**IDC Liège** vient de terminer la programmation d'une **EPROM** pour le générateur de caractères. Elle contient 255 caractères dont les 127 premiers sont compatibles avec les imprimantes **EPSON** et **STAR** .. Elle ne perturbera pas vos programmes elle est aussi compatible avec l'**EPROM** du **DAI**

Sont disponibles avec cette **EPROM**

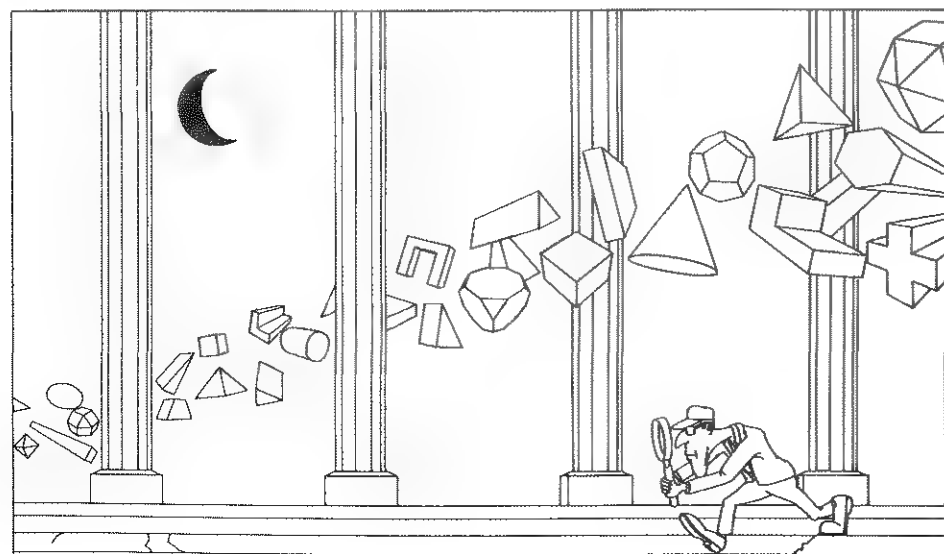
- les caractères accentués (au clavier avec le traitement de texte) é è à ù ç ê à ù î ô,
- les caractères graphiques des imprimantes **EPSON** (128 à 159)
- d'autres caractères graphiques pointillés ..
- des caractères mathématiques tels que racine carrée, somme, intégrale, chiffres en exposants et en indices,
- les lettres grecques

Une dizaine d'emplacements sont libres pour vos propres caractères. Nous pouvons vous les programmer. Il suffit de nous communiquer leur dessin dans une grille de 8 x 11 (cf manuel français p. 124)



Pour avoir cette **EPROM**, il suffit de verser 500 FB à Ph. Rasquin au compte 001-1845627-84 pour la Belgique. Pour l'étranger il faut envoyer un mandat international à Ph Rasquin, Rue de Saivelette 89, 4510 Belgique. Mentionnez clairement vos nom et adresse !

Le responsable **IDC LIEGE**,  
Ph. Rasquin



## "PRESENTEZ VOUS QUI DISAIS" (OU L'EDITO D'UN CLUB QUI SE RENGAGE POUR LA DEUXIEME ANNEE CONSECUTIVE DANS L'AVENTURE DU DAICLIC)

Déjà le deuxième numéro de cette année et nous ne vous avons toujours pas dit bonjour ! Qu'a cela ne tienne ! Permettez nous de vous présenter les buts de notre club. Ils sont multiples et non exhaustifs ! Sachez cependant que fonctionnent déjà pleinement notre bibliothèque de logiciels (la softthèque et sa fameuse liste contenant d'une façon compacte et précise tous les renseignements sur les programmes qu'elle contient), notre service de documentation, notre service achats de matériels, sans oublier bien sur l'édition de programmes à des conditions défiant toutes concurrence.

### *To our English friends*

We are very glad that the French Dai users and those whose who speak the Shakespeare's language can now communicate via an commun issue. If you are interested in our club and its services (soft library, hardware section, programm diffision, etc.), don't hesitate to contact us. We will be very happy to answer you !

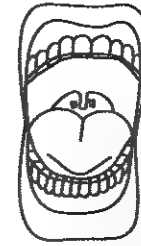
### *Coucou, nous revoilà en Français !*

Cette année nous inaugurerons deux nouveaux services.

La première est une surprise, et, comme toutes les surprises doit se faire attendre un tout petit peut. Mais nous croyons ne pas nous tromper en disant qu'elle rejouira tout les Français daiistes muni d'une certaine interface... Patience donc jusqu'au prochain numéro.

La seconde est en fait l'aboutissement d'une des raisons de la création de D.C.A. et s'est concrétisé grâce à la participation exemplaire de certains de nos membres (vous savez donc se qu'il vous reste à faire les autres ?...) Ainsi par **DCA hard**, voulons nous permettre à chacun, même s'il ne connaît rien à "l'art du fer à souder" ou n'a pas un détaillant de composants électroniques dans ses relations, d'étendre les possibilités matérielles de son ordinateur favori.

*Pour DCA, Philippe Casier*



## Hello, DAI's speaking !

Pascal JANIN, F-73 LA MOTTE SERVOLEX

Eh oui ! Voici enfin ce dont je vous avais parlé depuis trop longtemps déjà, et à propos de quoi je m'abstenais obstinément (ndrl:!) de vous donner de plus amples informations. Vous avez sans doute deviné ce dont il s'agit: le synthétiseur vocal !

Passons tout d'abord quelques points en revue.

- La bête en question ne jacasse convenablement qu'en anglais; amis francophones, ne soyez pas déçus pour autant: en faisant abstraction (ndrl: quelle obstination...) de son déplorable accent anglais, il est néanmoins capable de vous sussurer à l'oreille quelques bonnes phrases en français, mais étant donné que cette satanée langue d'outre-manche est plus "causée" que notre belle langue française, les résultats obtenus avec la première sont nettement plus convaincants qu'avec la seconde...

- La réalisation de ce petit montage, d'un prix de revient ridicule face aux plaisirs qu'il vous procurera, est à la portée de n'importe quel informaticien en herbe sachant manipuler avec dextérité un petit fer à souder (maxi 40 W !)...

- La bête se connecte en principe avec le lecteur DCR; pour le Ken-DOS, je n'ai pas vérifié, mais à mon avis il doit falloir débrancher au préalable tout l'équipement externe connecté au bus DCE (lecteur + alimentation) sans néanmoins toucher à la carte interne sur le X-bus.

Pour finir, l'article entier sera réparti sur 2 DAICLIC, d'une part pour maintenir les lecteurs en haleine, d'autre part, pour vous permettre de monter d'abord la partie électronique et non pas vous étouffer sous une montagne de documentation donnée en un seul bloc.

Allez, assez bavardé, passons à la...

## 1ère PARTIE: LA DESCRIPTION TECHNIQUE

La pièce maîtresse de la carte électronique est un petit processeur vocal mis récemment sur le marché "amateur" de l'électronique. Cette petite bête à 28 pattes, nommée SPO 256, travaille sur le principe des phonèmes. Chaque mot parlé couramment est décomposé en une suite de phonèmes envoyés successivement au SPO 256 qui les reproduit par un haut-parleur précédé d'un petit ampli. Pour la langue anglaise, il suffit de 60 phonèmes pour reproduire pratiquement tout son vocabulaire; pour la langue française, il en faut un peu moins de 40 pour reproduire également presque tout le Larousse! Il est également utile de préciser que le SPO 256 n'admet pas de changement de registre (voix aiguë ou grave) ni de changement d'accent! Les anglais ou français du sud et du nord y sont tous sur un pied d'égalité. La deuxième partie de l'article sera d'ailleurs entièrement consacrée à l'exploitation "pratique" de cette carte. Mais n'anticipons pas trop...

Le schéma électronique est "relativement" simple:

## DCA hard

Les cartes vont sont proposées sous deux formes. La première, ou version kit (VK), comprend le circuit imprimé percé, les composants, et la notice de montage. La deuxième, ou version intégrale (VI), comprend la carte entièrement montée et testée. Pour certaines cartes nécessitant un minimum d'habileté dans la soudure ou de connaissances électronique, nous vous conseillons fortement la version intégrale (dans ce cas la carte est signalée par \*\*\*). A noter qu'aucunes cartes n'est livrées avec boîtier, certaines cartes pouvant en effet se regrouper ensemble. Enfin il faut rajouter aux prix indiqués (moins la remise éventuelle de 10% pour les membres de DCA) les frais d'envoi variant ci-dessous (nous consulter dans le cas d'une commande groupée).

\$ : envoi léger, France : 3,90 FF  
\$\$ : envoi lourd, France : 12,60 FF  
\$\$\$ : envoi léger, Europe : 10,30 FF  
\$\$\$\$ : envoi lourd, Europe : 28,80 FF

**ANANUM** : Faites l'acquisition de 8 données numériques et 8 données analogiques continues avec cette carte, début d'une série d'extension du bus DCE. Remarque : \*\*\*

Prix : 378 FF en VK, 398 FF en VI  
+ \$\$ ou \$\$\$\$

**DAISPOT** : Vous rêviez de vous faire un chenillard pro ou de commandez jusqu'à 8 canaux 220v simultanés (jusqu'à 24 avec 3 cartes). C'est chose faites à présent avec cette carte entièrement isolée du secteur.

Prix : 370 FF en VK, 390 FF en VI  
+ \$\$ ou \$\$\$\$

**ELECTROFUS** : En cas de défaillance de l'alimentation il arrive fréquemment que du 22v arrive sur les circuits TTL ce qui est mortel pour eux. Cette carte, placée à la place du fusible du Dai, le complète avantageusement.

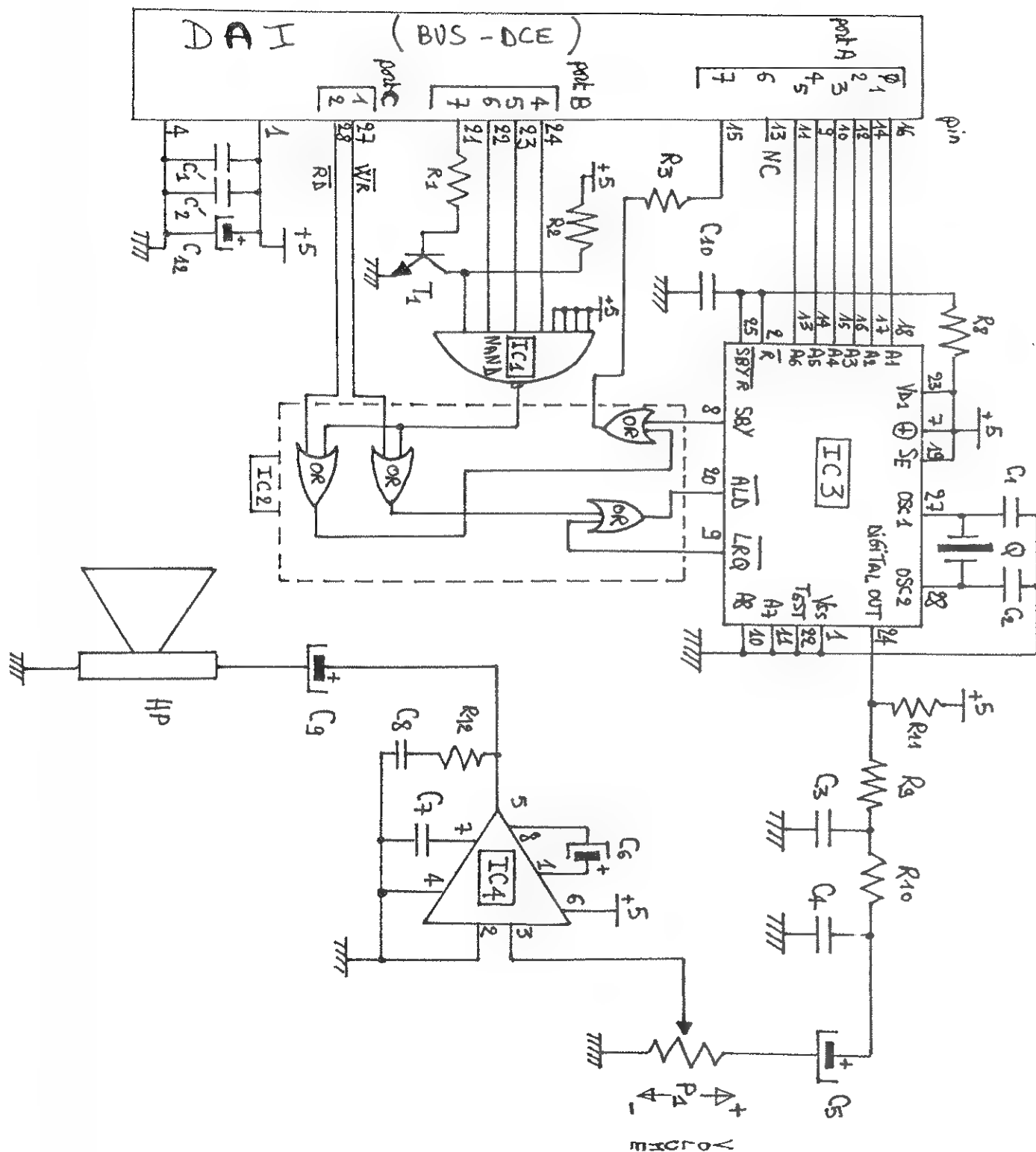
Prix : 45 FF en VK, 50 FF en VI  
+ \$ ou \$\$\$

**INTERFACE JOYSTICK** : Avec cette carte vous pourrez remplacer vos manettes de jeux par des manettes type atari et autres spectravideo, ce qui est beaucoup plus efficace pour les jeux.

Prix : 30 FF en VK, 35 FF en VI  
+ \$ ou \$\$\$

Nous avons essayé d'obtenir les prix les plus bas pour les composants des montages, cependant si vous connaissez des fournisseurs particulièrement avantageux, n'hésitez pas à nous contacter.

DCA hard  
c/o Philippe Casier  
31ter rue de Paris  
92190 MEUDON  
FRANCE



et valide ou non la lecture ou l'écriture de données pour le SPO 256. Celui-ci génère 3 signaux importants:

- . **LRQ** (Load ReQuest, pin 9) est à l'état 0 lorsque le SPO 256 est prêt à recevoir d'autres données.
- . **SBY** (Stand BY, pin 8) est à l'état 1 lorsque le SPO 256 a fini d'exécuter la précédente donnée.
- . **ALD** (Address Load, pin 20) doit être "pulsé" à 0 lorsqu'une donnée a été envoyée vers le SPO 256 pour qu'il l'accepte.

La combinaison du décodage et des 3 signaux précités plus **WR** et **RD** venant du bus **DCE**, permet de contrôler efficacement le SPO 256. A noter que l'envoi d'une donnée et la lecture de l'état du SPO 256 ("prêt" ou "en cours d'exécution") se font sur la même adresse (**OUT** #70... ou **INP** (#70)). Pratique, n'est-ce pas ?...

Il est à noter également que le **DAI** s'informe de l'état du SPO 256 par la lecture du bit 7 du port 7 choisi pour la carte: lorsque ce bit 7 est à 1, on peut envoyer une nouvelle donnée. L'exploitation par logiciel est la suivante:

- OUT** #70, donnée (de 0 à 64)
- WAIT** #70, #80 (attend que ce fameux bit 7 passe à 1)
- NEXT** donnée

Noter aussi la résistance de 82 K (valeur déterminée après plusieurs essais) en série entre la porte **OR** et le bit 7 du port A: elle évite, lors de l'envoi d'une donnée, la destruction de la porte logique en cas de conflit de signaux en limitant l'intensité de court-circuit à une valeur tout-à-fait supportable pour la porte **OR** ( $I = U/R = 5V/82000 \text{ Ohms} = 60 \text{ micro-A}$ !). Il ne faut surtout pas baisser cette valeur de 82 KOhms (destruction de la porte) ni l'augmenter (le niveau logique n'est plus pris en compte). Enfin, le seul transistor de toute la carte permet de réaliser une porte inverseuse à moindres frais pour le décodage par le 74LS30.

Ces quelques explications concernent le montage en "amont" du SPO 256, passons maintenant au côté "aval".

Une fois la donnée reçue, le SPO 256 fait entendre sa "voix" par sa pin 24 sur laquelle vient se greffer un double filtre passe-bas passif pour donner au signal une forme moins "brute" (plus agréable à l'oreille). Viennent ensuite un potentiomètre pour le réglage du volume, puis un petit ampli de 0.5 W environ, tout à fait classique et d'une qualité à toute épreuve pour ce prix, à la sortie duquel vient se greffer l'inévitable (et indispensable) haut-parleur. Il est important de souligner que la qualité de la parole dépendra en grande partie du haut-parleur choisi: un petit modèle pour magnétophone (5cm - 80hms - 0.3 W) donnera des résultats plutôt décevants car sa petite taille ne permet qu'une reproduction médiocre des graves dont la "voix" humaine fait un usage intensif. Aussi, je vous conseille vivement d'opter pour la "taille au-dessus", à savoir un haut-parleur d'au moins 10cm en 80hms, de puissance respectable (1W, c'est le pied... et c'est pas ruineux !)

Encore un mot sur le quartz de SPO 256 (car en effet, il lui en faut un pour ses divers filtres digitaux, ROM internes, etc): je préconise une valeur de 3.2768 MHz car c'est la plus facile à trouver et la moins chère! Mais toute valeur comprise entre 3 MHz et 4 MHz peut tout aussi bien convenir (plus la valeur de la fréquence est élevée, plus la voix est aigüe...).

Pour de plus amples renseignements sur le SPO 256 lui-même, vous pouvez toujours vous référer au numéro 81 (mars 85) de la revue **ELEKTOR**, pages 3-38 et suivantes.

**Une dernière mise en garde:**

Le circuit intégré spécialisé utilisé ici doit pouvoir se trouver à peu près partout (chez TANDY par exemple). Mais attention: sous la référence SPO 256 se cachent 2 (et plus) types de circuits:

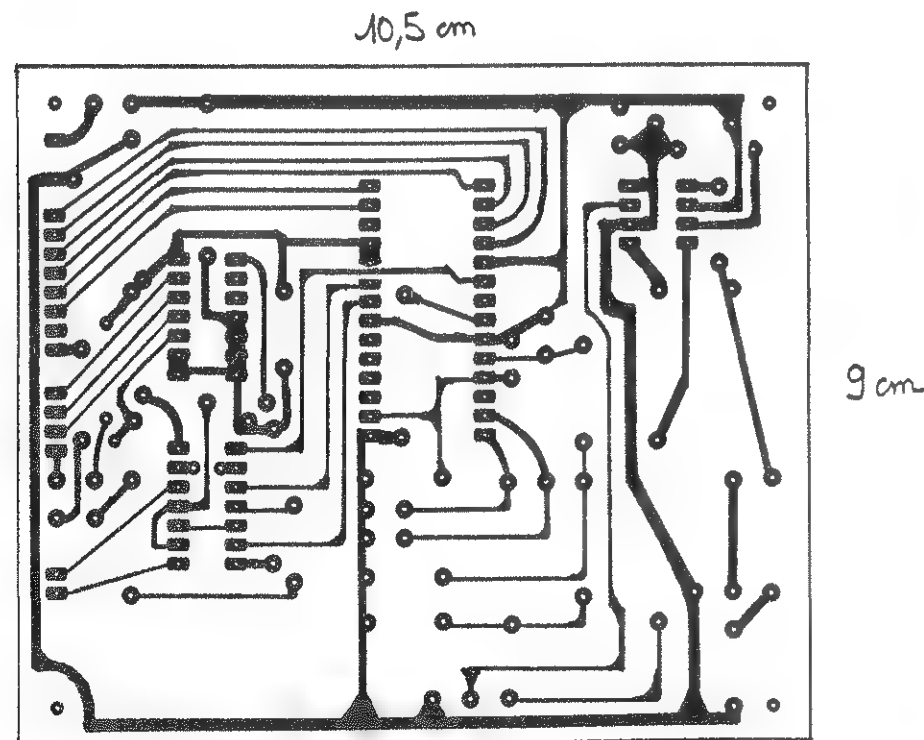
- le SPO 256 "tout court" qui est vendu avec une ROM contenant son vocabulaire à un prix inférieur à 100 FF (en général, il est bradé car personne n'en veut plus: son vocabulaire est fixé à l'avance et il ne fonctionne pas sur le principe des phonèmes). **A ECARTER !!!**

- le SPO 256 ALx: où x prend une valeur entre 0 et 9; à chaque x correspond une langue différente. Tout autre type que x=2 est introuvable, le plus courant (et le seul distribué !) est le SPO 256 AL2 qui parle, vous l'avez deviné, l'anglais. C'est cette seule et unique version qui fonctionne avec la carte, n'en achetez pas d'autre !

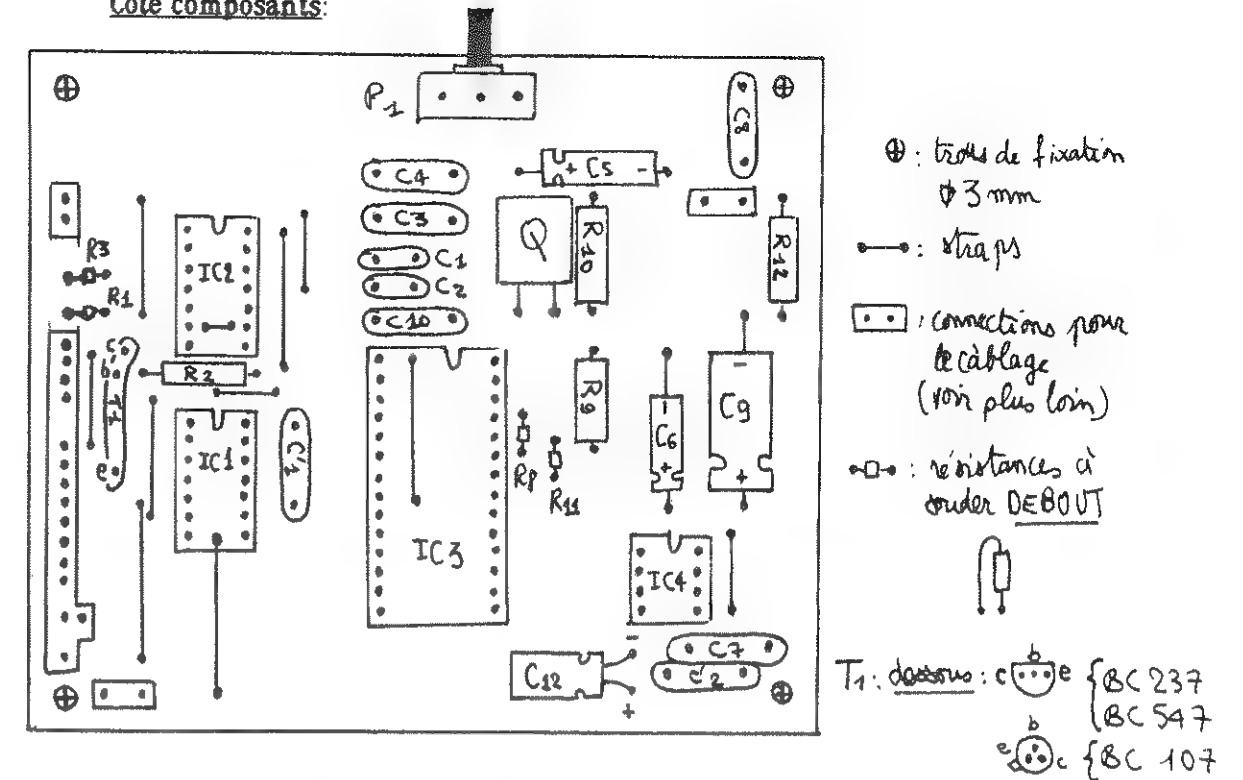
Donc: retenez bien son nom: SPO 256 AL2.

Voici maintenant les plans du circuit imprimé que je vous propose:

Côté cuivre:

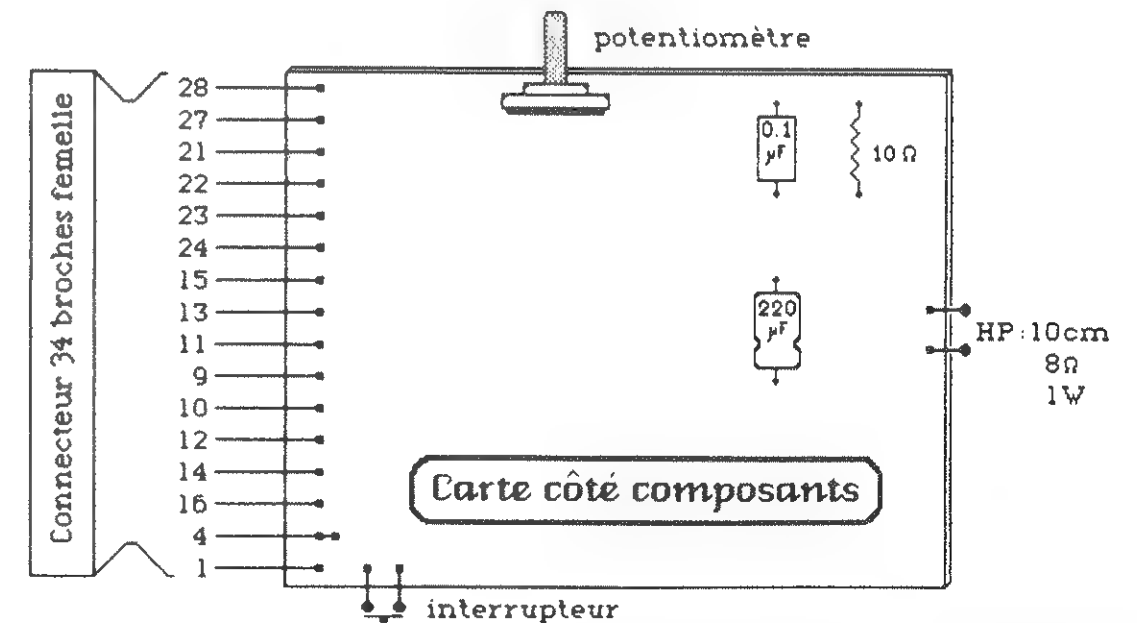


Côté composants:



Peu de remarques à faire si ce n'est que je vous conseille vivement d'étamer le circuit imprimé (question de longévité). Attention au transistor T1 dont la patte "émetteur" devra être pliée (à la pince !) plus loin que les 2 autres.

Câblage:



Pour le câblage du haut-parleur et de l'interrupteur, utiliser du fil simple de petite section. Par contre, pour le câblage vers le bus DCE, il faudra se procurer un connecteur type SUB-D 34 broches femelles que l'on sertira d'un côté sur du câble en nappe 34 conducteurs au pas de 1.27 (très facile à trouver chez n'importe quel dealer DAI...) et que l'on dénudera de l'autre avec toutes les précautions d'usage pour ces 34 conducteurs

extrêmement fragiles. Comme tous ne sont pas utilisés, couper proprement ceux qui restent et en isoler le bout par du ruban adhésif (simple .. mais efficace !).

#### Essais le moment crucial !

Bien revérifier une dernière fois toutes les orientations des circuits intégrés, condensateurs Eteindre alors le DAI, mettre en place le connecteur dans la prise bus DCE, et rallumer le DAI. Si rien ne claque ou ne chauffe sur la carte, c'est bien parti. On doit alors observer un infame gargouillis dans le haut-parleur, et ce, quelque soit la position du potentiomètre de volume: ce sont tous les parasites qu'induit le DAI, et contre cela, il n'y a aucun remède (voir plus loin) ! Vérifions si tout marche bien: mettez le volume à fond, puis en mode direct, faites.

OUT #70,xx (RETURN) ; avec 5 <= xx <= 63

Un son **DOIT** se faire entendre. Il s'arrête tout seul, ou bien dès que l'on fait:

OUT #70,yy (RETURN) ; avec 0 <= yy <= 4

Si c'est le cas... Félicitations ! Patientez dès lors jusqu'au prochain article... en rentrant le programme de démonstration suivant:

#### IMP FPT

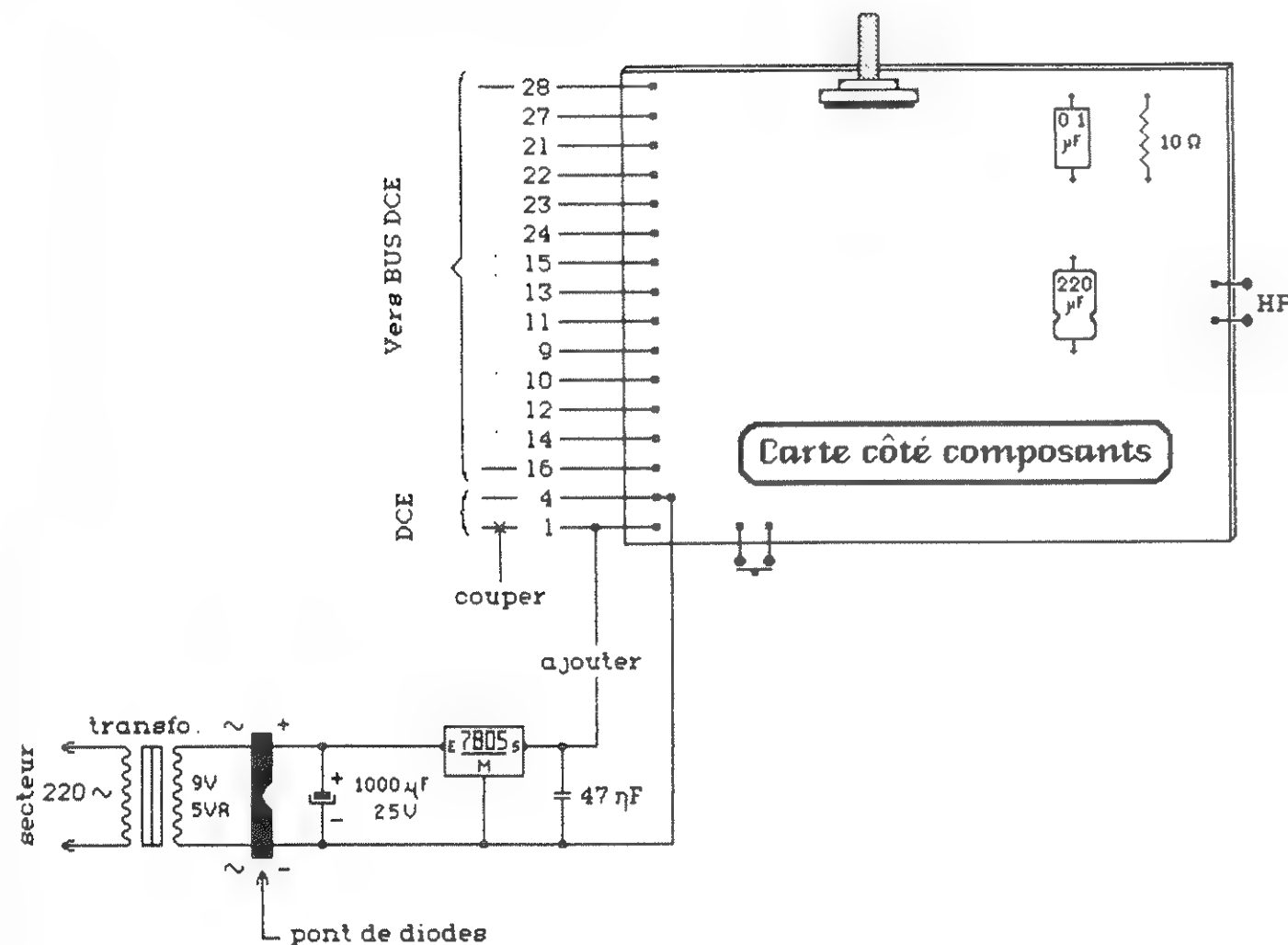
```
1 PRINT CHR$(12);TAB(17);" DEMO <<< EXPEND SPK >>>"
2 PRINT TAB(17);"*****"
3 POKE #75,32
9 CLEAR 10000
10 DIM DIZ%(9,0,9,0),UN%(9,0,6,0),SPEC%(12,0,10,0)
20 FOR I%=1 TO 9:FOR J%=0 TO 6:READ UN%(I%,J%):NEXT:NEXT
21 FOR K%=1 TO 9:FOR L%=0 TO 9:READ DIZ%(K%,L%):NEXT:NEXT
22 FOR M%=1 TO 12:FOR N%=0 TO 10:READ SPEC%(M%,N%):NEXT:NEXT
23 GOTO 1000
29 REM ----- UNITES
30 DATA 46,15,11,0,0,0,3
31 DATA 13,31,0,0,0,0,3
32 DATA 29,14,19,0,0,0,3
33 DATA 40,40,58,0,0,0,3
34 DATA 40,40,6,35,0,0,3
35 DATA 55,12,12,2,41,55,3
36 DATA 55,7,7,35,12,11,3
37 DATA 20,2,13,0,0,0,3
38 DATA 11,24,6,11,0,0,3
39 REM ----- DIZAINES
40 DATA 13,7,7,11,0,0,0,0,3
41 DATA 13,48,7,7,11,1,2,13,19,3
42 DATA 29,52,1,2,13,19,0,0,0,3
43 DATA 40,58,2,13,19,0,0,0,0,3
44 DATA 40,40,12,1,40,40,2,13,19,3
45 DATA 55,55,12,1,41,55,2,13,19,3
46 DATA 55,55,15,35,12,11,2,13,19,3
47 DATA 20,2,13,19,0,0,0,0,0,3
48 DATA 11,6,11,2,13,19,0,0,0,3
49 REM ----- 11>19
50 DATA 12,45,7,7,35,12,11,0,0,0,3
51 DATA 13,48,7,7,45,35,0,0,0,0,3
52 DATA 29,51,1,2,13,19,11,0,0,0,3
53 DATA 40,58,1,2,13,19,11,0,0,0,3
54 DATA 40,12,40,1,2,13,19,11,0,0,3
55 DATA 55,55,12,2,41,55,2,13,19,11,3
56 DATA 55,55,15,35,12,11,2,13,19,11,3
```

```
57 DATA 20,1,2,13,19,11,0,0,0,0,3
58 DATA 11,6,11,1,2,13,19,11,0,0,3
59 REM ----- CENT
60 DATA 57,15,15,11,0,33,39,7,7,0,21
61 REM ----- MILLE
62 DATA 29,24,32,43,15,0,11,21,0,0,3
63 REM ----- MILLION
64 DATA 16,12,12,45,49,15,11,0,0,0,3
65 REM ----- FIN DES PHONEMES
1000 POKE #75,95
1001 PRINT "DESIREZ-VOUS ENTENDRE TOUS LES CHIFFRES (O/N) ?";
1002 G% =GETC:IF G% =0 GOTO 1002:IF G% =ASC("N") THEN PRINT :PRINT :GOTO 1070
1005 PRINT :PRINT :PRINT "ECOUTEZ LES CHIFFRES ( en Anglais ) .. ";
1010 FOR I%=0 TO 9:FOR J%=0 TO 6
1011 OUT #70,UN%(I%,J%):WAIT #70,*80:NEXT:NEXT
1020 FOR L%=0 TO 9:OUT #70,DIZ%(1,0,L%):WAIT #70,*80:NEXT
1030 FOR M%=1 TO 9:FOR N%=0 TO 10
1031 OUT #70,SPEC%(M%,N%):WAIT #70,*80:NEXT:NEXT
1040 FOR K%=2 TO 9:FOR L%=0 TO 9
1041 OUT #70,DIZ%(K%,L%):WAIT #70,*80:NEXT:NEXT
1050 FOR M%=10 TO 12:FOR N%=0 TO 10
1051 OUT #70,SPEC%(M%,N%):WAIT #70,*80:NEXT:NEXT
1060 PRINT :PRINT :PRINT :PRINT
1070 Y% =CURY
1071 CURSOR 0,Y%:PRINT SPC(59);CURSOR 0,Y%:INPUT " VOTRE NOMBRE (Max :
2147483646 ; Min:1) ":NB%
1072 IF NB% <1 GOTO 1071:IF NB% >=1E6 GOTO 1075
1073 NA% =NB%:GOSUB 2000:GOTO 1071
1075 NA% =INT(NB%/1E6):GOSUB 2000
1076 FOR K%=0 TO 10:OUT #70,SPEC%(12,0,K%):WAIT #70,*80:NEXT
1077 NA% =INT(NB% MOD 1E6):GOSUB 2000
1078 GOTO 1071
2000 N1% =INT(NA%/1E5):IF N1% =0.0 GOTO 2010
2001 A% =1
2005 FOR I%=0 TO 6:OUT #70,UN%(N1%,I%):WAIT #70,*80:NEXT
2006 FOR I%=0 TO 10:OUT #70,SPEC%(10,0,I%):WAIT #70,*80:NEXT
2010 N2% =INT(NA%/10000.0) MOD 10.0:IF N2% =0.0 GOTO 2020
2011 A% =1
2012 IF N2% =1 GOTO 5000
2013 FOR I%=0 TO 9:OUT #70,DIZ%(N2%,I%):WAIT #70,*80:NEXT
2020 N3% =INT(NA%/1000.0) MOD 10.0:IF N3% =0.0 GOTO 2030
2021 A% =1
2022 IF N2% =1 GOTO 2030
2023 FOR I%=0 TO 6:OUT #70,UN%(N3%,I%):WAIT #70,*80:NEXT
2030 IF A% =1 THEN A% =0:FOR I%=0 TO 10:OUT #70,SPEC%(11,0,I%):WAIT #70,*80:NEXT
2040 N4% =INT(NA%/100.0) MOD 10.0:IF N4% =0.0 GOTO 2050
2041 FOR I%=0 TO 6:OUT #70,UN%(N4%,I%):WAIT #70,*80:NEXT
2042 FOR I%=0 TO 10:OUT #70,SPEC%(10,0,I%):WAIT #70,*80:NEXT
2050 N5% =INT(NA%/10.0) MOD 10.0:IF N5% =0.0 GOTO 2060
2051 IF N5% =1.0 GOTO 6000
2052 FOR I%=0 TO 9:OUT #70,DIZ%(N5%,I%):WAIT #70,*80:NEXT
2060 IF NA% MOD 10 =0 THEN RETURN
2070 FOR I%=0 TO 6:OUT #70,UN%(NA% MOD 10.0,I%):WAIT #70,*80:NEXT
2080 RETURN
5000 IF INT(NA%/1000.0) MOD 10.0 =0.0 GOTO 2013
5010 FOR I%=0 TO 10:OUT #70,SPEC%(INT(NA%/1000.0) MOD 10.0,I%):WAIT #70,*80:
NEXT
5020 GOTO 2020
6000 IF NA% MOD 10 =0 GOTO 2052
6010 FOR I%=0 TO 10:OUT #70,SPEC%(NA% MOD 10.0,I%):WAIT #70,*80:NEXT:RETURN
```



Ce programme restitue tout nombre **ENTIER** rentré au clavier sous forme parlée... Ce n'est qu'un avant-goût de ce qui vous attend au prochain numéro !

**PS:** Pour tous ceux que les parasites rendent dingues, il est possible de les supprimer en passant pas une alimentation **EXTERIEURE** au **DAI**; les modifications du câblage et les plans suivent...



**Liste des composants pour alimentation extérieure:**

- 1 transformateur 220V/9V - 5VA
- 1 pont de diodes 1A
- 1 condensateur électrolytique 1000µF - 25V
- 1 régulateur 7805 (positif 1A - 5V)
- 1 condensateur 47nF

**Liste générale des composants:**

R1	resistance	3.3 k	1/2 W
R2		4.7 k	
R3		82 k	
R8		100 k	
R9		33 k	
R10		33 k	
R11		56 k	
R12		10 ohm	
P1	potentio	10 k log	
C1	condensateur	22 pF	
C2		22 pF	
C3		22 nF	
C4		22 nF	
C5		10 µF 16 V	
C6		10 µF 16 V	
C7		100 nF	
C8		100 nF	
C9		220 µF 16 V	
C10		100 nF	
C12		100 µF 16 V	
C'1		100 nF	
C'2		47 nF	
Q	quartz	3.2768 Mhz	
IC1		SN 74LS30 (14)	
IC2		SN 74LS32 (14)	
IC3		SPD 255 AL2 (28)	
IC4		LM 386 (8)	
T1		NPN (BC107)	
- HP haut-parleur 8Ω 1W Ø10 cm			
- 1 connecteur SVBD 34 broches femelles a sertir			
- du cable-nappe 34 conducteurs au pas de 1.27 mm			
- fil simple			
- soudure			
- 1 interrupteur simple			
1 circuit 2 positions			

**SI CA NE MARCHE OU'A MOITIE ?...**

Pour la connexion au bus DCE, deux cas peuvent se présenter:

1. Si la carte de parole est connectée avec d'autres extensions sur le bus DCE (ex: **Mémocom, Ken-DOS,...**), IL SE PEUT que ces extensions brouillent le signal **READY**, c-à-d se superposent à celui-ci qui devient alors indétectable car trop faible... Alors passez au paragraphe qui suit en allant jusqu'à 10 Kohms, voire 8,2 Kohms MINIMUM pour R3. Si la carte ne marche pas mieux, vous DEVREZ déconnecter du bus DCE toutes les cartes extérieures, sauf "synthé de parole" (évidemment !), pour pouvoir exploiter cette dernière...

2. Si la carte est connectée SEULE au bus DCE, armez-vous de patience et ABAISSEZ la valeur de R3 tout en essayant périodiquement **SPEECH COUNTER**, et ce jusqu'à une valeur minimale de 15 Kohms. A partir d'une certaine valeur (en principe, aux alentours de 47 Kohms), le programme se met subitement à fonctionner: le mal est alors réparé !!!

### 2ème Partie

Après une longue interruption, l'article sur le **MINITEL** se poursuit donc par l'explication ingrate d'une des parties les plus sujettes aux exceptions et donc aux difficultés de gestion, en vue de son imitation pour parfaire notre émulateur **MINITEL**.

Ici ne seront évoqués que les paramètres en rapport avec les caractères alphanumériques. Nous commencerons le graphisme la prochaine fois (nanère...).

### **ATTENTION !!**

Dans le **DAICLIC 4** ont débuté les explications, et cet article se voulait suffisamment complet pour vous proposer les moyens matériels (interface) et logiciels pour commencer à tâter un peu du sujet par vous même. Les deux programmes (**MINITEL 1** et **2**) qui devaient se trouver dans cet article ont finalement été publiés plus tard dans le **DAICLIC 6**. Ils ont bien sûr été envoyés à tous ceux qui nous en avaient fait la demande avant la parution du numéro 6. En plus ce mois-ci, je vous annonce la naissance d'un utilitaire de développement encore plus performant pour étudier votre **MINITEL**, entièrement en assembleur et commenté par M. Sébastien DUBOURG. Ce programme, comme tous nos logiciels, est disponible et entièrement gratuit (demandez **MINITEL 3** !!).

Pour les personnes qui ne sont pas satisfaites de la petite interface décrite, le CLUB I.D.C. Bordeaux diffuse une interface un peu plus performante qui vous permettra de brancher sur la même **RS-232**, votre **MINITEL**, et votre imprimante, avec possibilité de commutation de l'un à l'autre aussi bien par un simple interrupteur manuel que par une prise sur une des deux interfaces K7, ce qui autorisera la commutation automatique par SOFT !!!

Vous en trouverez le plan dans un très prochain **DAICLIC**.

### Préambule

Dans cet article, je parle indifféremment de recevoir des séquences ou d'envoyer des séquences, en effet, ne perdez pas de vue que vous pouvez considérer vos relations avec le **MINITEL** sous deux angles complémentaires :

- celui de l'utilisateur qui ne sait que recevoir des séquences d'attributs.
- celui de l'émulateur qui a construit "l'interface" décrite dans le précédent article et qui va envoyer tous les codes qu'il veut grâce à son **DAI** et aux programmes que I.D.C. Bordeaux tient à la disposition de ses adhérents.

### Les attributs de GROUPE

#### Mémorisation des attributs de groupe

Prenons un exemple.

Le **MINITEL** reçoit un attribut pour faire changer la couleur du fond: l'ancienne couleur de fond étant noire, la nouvelle couleur de fond, rouge. La séquence correspondante reçue par le **MINITEL** est alors **1B 23 20 51** (cf. **DAICLIC 4**...).

En théorie, cette commande change la couleur du fond sur l'ensemble de l'écran

En pratique:

- Cette commande demande à être validée
- La validation s'effectue par la réception d'un caractère espace que nous qualifierons d'espace validant !!!
- Tous les caractères reçus avant cet espace validant sont affichés avec l'ancienne couleur de fond, car le changement d'attribut de couleur de fond n'a pas encore été confirmé par l'espace validant (enfonçons le clou !!).
- Après réception de l'espace validant, l'attribut, bien qu'étant en théorie valable sur tout l'écran, ne reste actif que sur la fin de la longueur de la ligne où l'espace validant a été reçu et que sur cette partie de l'écran !!
- Pour une ligne différente, que ce soit la suivante après avoir atteint naturellement l'extrémité de la ligne où nous étions, ou que ce soit n'importe quelle autre ligne atteinte grâce aux commandes de déplacement du curseur, le fond garde la couleur qu'il avait avant la demande de changement d'attribut "couleur de fond" comme si de rien n'était !! (CIEL !!)
- Si sur cette ligne suivante, une nouvelle commande d'attribut de groupe est reçue, autre qu'un changement de couleur de fond (bien sûr !!), par exemple l'attribut de groupe 'lignage' (commande de soulignement si on est en mode texte), alors dès réception d'un espace "validant", nous aurons non seulement obtention du 'lignage' mais en plus le changement de couleur du fond et nous verrons notre fond rouge ressortir des enfers !!! (HELLS).

Ces quelques remarques sont généralisables non seulement à toutes les demandes de changement de couleur, mais à toutes les demandes de changement (gestion) des attributs de groupe (cf. article précédent).

### POURQUOI ???

Parce que:

Tout se passe comme si le **MINITEL** "mémorisait" les différents attributs de groupe en cours pour une page d'écran (ce que nous devons faire dans notre programme d'émulation en créant un "buffer" "attributs de groupe") et que ces différents attributs sont remis à jour pour la ligne en cours, à partir de la position courante du curseur, lors de chaque réception d'un espace validant.

### Qu'est ce qu'un ESPACE VALIDANT ?!

Tout se passe comme si le **MINITEL** distinguait deux catégories différentes du caractère espace. En réalité c'est le même code qui est utilisé et la différence entre ces deux espaces ne tient qu'à leur position dans le flot des données : le premier espace reçu après une séquence d'attributs de groupe est dit **VALIDANT**. Son code est le même que l'espace normal (: CHR\$(#20)) et il est affiché à l'écran comme (ou presque .. cf. bien plus tard..) un espace normal, faisant donc avancer le curseur d'une case

Reprenons notre exemple:

Le **MINITEL** reçoit une séquence de demandes d'attributs de changement de couleur de fond en rouge, le **MINITEL** interprétera le prochain espace trouvé dans le flot des données, comme validant. Avant réception de cet espace validant le **MINITEL** présente un

etat d'attente de validation d'attribut de groupe" grâce à un drapeau ( "flag" pour les anglophiles) comme s'il possédait une variable interne se mettant à 1 ou à 0 selon qu'il attend ou non une validation des attributs de groupe.

Ceci implique que nous devons aussi créer une variable afin de caractériser notre buffer d'attributs de groupe pour un écran donné selon que l'espace validant attendu aura été reçu ou non

Après réception de cet espace, la "variable interne" du **MINITEL** est mise à jour et l'ensemble des attributs de groupe en cours est mis à jour pour l'écran considéré pour nous, le fond de l'écran prendra la couleur rouge à partir de la position du curseur au moment de la réception de l'espace validant, et jusqu'à la fin de la ligne. Les espaces suivants sont considérés comme normaux

#### Nota Bene:

Cette "attente" interne d'espace validant du **MINITEL** ne l'empêche en rien de continuer à dialoguer avec ses périphériques (monde extérieur, **DAI**, écran, clavier).

C'est l'état à 0 ou 1 de la "variable interne" du **MINITEL** qui lui permet de savoir si l'espace reçu est validant ou normal. Chaque demandé d'attribut de groupe remet la variable dans son état "attente" et chaque réception d'espace la remet dans son état "libre".

Toujours dans notre exemple si nous nous déplaçons sur une autre ligne (dont le fond est encore noir) grâce aux instructions 'curseur', et que nous envoyons une séquence d'attributs de lignage: **1B 23 20 5A**, alors le **MINITEL** place sa "variable interne" en état d'attente d'espace validant. Après réception de cet espace, le **MINITEL** va vérifier tous les attributs de groupe en fonction pour cette page d'écran et les caractères suivants qui vont être affichés sur cette ligne seront non seulement soulignés mais aussi avec une couleur de fond rouge !!

Mais si vous désirez seulement avoir des caractères soulignés et que vous vouliez conserver la couleur de fond d'origine noire, il vous faudra rajouter autre chose :

#### La méthode "très boeuf":

Effacer tout l'écran par un **CHR\$(12)** ce qui efface aussi tout état d'attente et les anciens attributs de groupe en rigueur pour l'ancienne page. Inconvénient: ça efface carrément tout !! (sauf certains attributs 'protocollaires' tel le scroll par exemple [cf. plus tard]); éteignez votre **MINITEL** c'est plus rapide.

#### La méthode "un peu moins" (tout juste veau):

Annuler l'attribut gênant par son attribut inverse, ici préciser "couleur de fond noire" + "lignage" + espace validant (un seul nécessaire bien sûr si vous avez suivi).

#### Une autre bizarrerie.

Tous les attributs de groupe sont affichés sans restriction sur la totalité de la fin de la ligne où ils ont été validés ... sauf l'attribut "couleur de fond" !

La couleur de fond, si elle rencontre une ligne "discontinue", c'est-à-dire une ligne où elle a été validée mais où les caractères ne sont pas contigus (des mots séparés par des "espaces"), mais où ces espaces ont été obtenus non pas par pression sur la barre d'espace, mais par une commande de déplacement de curseur (comme les touches fléchées de notre **DAI** en mode éditeur), la couleur de fond donc s'arrête au bord du "trou" entre deux mots et ne se propagera jusqu'au prochain "trou" que si le trou qui arrête sa progression est bouché, même par un espace !!! (ce qui ne change rien à l'aspect visible de l'écran...).

Ceci nous oblige, si nous voulons une émulation permettant ces effets qui peuvent être mis à profit pour un semblant d'animation, à considérer TROIS sortes différentes d'espaces affichables alors que l'écran de notre **DAI** n'en connaît qu'un seul (pour mémoire, espace "normal" + "validant" + "trou" !!!)

#### **Bon courage**

Sebastien DUBOURG  
et Bruno DELANNAY  
IDC Bordeaux



#### **DAI QUI RIT (EXCLUSIVITE DAICLIC !):**

Claudius MORIN, CAEN (F)

#### Codes graphiques au clavier

Le truc est le même que pour reconfigurer le clavier en AZERTY: il suffit de recopier en mémoire vive le tableau des codes ASCII qui figure en MEM (\*E8C5 - \*E934), par exemple, à partir de \*6000. Pour cela, faire sous **UTility**, **MEC85 E934 6000**. Ensuite, par la commande **S**, changer les codes désirés, par exemple en attribuant des codes graphiques aux touches peu utilisées (éviter d'altérer les touches numériques !). Il reste à valider cette nouvelle table, en déviant le vecteur qui pointe normalement sur \*E8C5. Ce vecteur réside en \*2F9. Pour cela, faire par exemple: **POKE \*2F9,0 : POKE \*2FA,\*60** (attention : les deux pokes sur la même ligne, sous peine de surprises !).

Il est intéressant d'affecter aux touches choisies les codes de caractères semi-graphiques complémentés à 128 (code + 128). Cela ne change rien à la représentation à l'écran, mais permet de visualiser les codes 12 et 13 (**FF**, **CR**) qui, normalement, provoquent une action à l'écran, et ne sont pas visibles en tant que caractères. Ce subterfuge permet de les incorporer directement dans des instructions **PRINT** ou **REM**. La seule précaution à prendre consiste à ne pas oublier que le mode **EDIT** refuse les caractères dont le code est supérieur à 128. Ils ne sont donc plus visibles dans ce mode, mais ils ne sont pas perdus pour autant. Prudence donc, si vous modifiez des lignes susceptibles d'en contenir. Pour récupérer un clavier normal, il faut restaurer le vecteur de clavier: **POKE \*2F9,\*C5 : POKE \*2FA,\*E8** (toujours sur la même ligne !).

# Les maux du DAI : le clavier (4)

## Les MAUX du DAI

(I.D.C Bordeaux)

### Serie le clavier (4)

#### 2eme Partie le Nouveau clavier ou PLACE AUX JEUNES

Ce type de clavier est beaucoup plus simple à dépanner ... quand on a des touches de rechange !! En effet le clavier est constitué de touches indépendantes les unes des autres, qui peuvent être dessoudées séparément. Mais si vous êtes isolé de tout magasin pourvu de ce type de touches, il peut être bon de savoir rénovier les contacts

Il ne faut cependant pas se leurrer sur l'efficacité de cette méthode qui, si elle permet une complète rémission en cas d'accident aigu (comme la renverse d'un bol de chocolat !!!) est loin d'égaliser la solution radicale qui est de changer la touche malade. D'autant que le démontage de la partie interne de la touche est long et délicat

Dans le texte, les mots entre " font référence aux annotations des schémas. Bien entendu, le démontage commence par l'enlèvement tous les branchements, prises, câbles (j'ai l'impression de me répéter ?) ainsi que le capot du DAI. Ensuite les cabochons de touche seront enlevés avec beaucoup moins de précautions que pour l'ancien modèle (cf. article précédent) car le plastique de ces touches est bien plus solide que l'ancien. Puis vous retournez le circuit imprimé de façon à avoir le côté soudures au-dessus (et les touches en dessous). Alors vous dessoudez les pattes (pins) [2 par touche] des touches incriminées. Une fois les pattes bien dégagées vous remettez le circuit imprimé dans sa position normale et vous extrayez la touche en appuyant sur les 'cliquets', à droite et à gauche du 'corps de touche', à l'aide de petites lames de tournevis, de façon à libérer la touche de la 'plaque métallique noire' qui maintient les touches ensemble.

Vous avez donc maintenant sur votre plan de travail les touches à réparer. Vous allez pouvoir commencer le démontage et pénétrer au coeur de ce mécanisme petit à petit ...

Il faut d'abord enlever le 'fond de touche'. Pour cela, ne pas tirer dessus !!, mais plutôt se munir d'une pince avec laquelle vous saisissez une des deux 'pins de fixation' dans sa partie alors visible (=partie 'à souder') puis vous lui imprimez une rotation comme si vous vouliez la dévisser tout en maintenant SOLIDAIRE, avec l'autre main, le 'fond de touche' du 'corps de touche' car il est INDISPENSABLE de retirer entièrement les deux 'pins de fixation' avant d'écartier le 'fond de touche' du 'corps de touche' sous peine d'arracher les 'pattes de l'interrupteur' de la partie 'contact' que vous verrez plus bas. Oh ! surprise effectivement la pin se dévisse faisant apparaître sa partie 'pas de vis'. Vous opérez de même avec la deuxième pin et vous pouvez à ce moment enlever le 'fond de touche'.

**ATTENTION:** Repérez très ATTENTIVEMENT l'orientation interne des éléments contenus dans le 'corps de touche' car vous allez devoir tous les enlever puis (accessoirement) les remettre.

Ne perdez pas le ressort 'poussoir' qui se sauve tout seul à cet instant et videz l'intérieur du 'corps de touche' en poussant sur le 'poussoir'. Apparaissent alors les parties nommées 'pièce intermédiaire' dont l'ergot s'insère dans l'encoche du 'poussoir'. Cette 'pièce intermédiaire' sert de surface de glissement pour le 'ressort contact' qui a une forme de E renversé et qui s'appuie sur le 'contact'.

Ce 'contact' est une petite pièce en plastique noir qui comporte à sa partie inférieure deux pattes en cuivre qui sont les contacts de l'interrupteur proprement dit. Sa face appuyée sur le 'ressort contact' est en caoutchouc noir souple tandis que les côtés et la face

opposée (qui possède un trou à son coin supérieur droit) sont en plastique noir dur (partie nommée 'cuvelage')

De toutes ces pièces démontées, seul le 'contact' est délicat. Donc après avoir nettoyé toutes ces pièces, vous les mettez bien en ordre de façon à les remonter dans le bon sens !! et vous vous consacrez au démontage du 'contact'.

Cependant, avant d'aller plus loin il vous faut être convaincu de l'utilité de ce démontage qui n'est pas sans risque. En effet si vous aviez seulement des problèmes mécaniques laissant le 'contact' intact (touche qui ne s'enfonce pas ou qui ne remonte pas) il est préférable (de jardin) de ne pas démonter ce petit 'contact' et de se contenter du bon nettoyage des autres éléments que vous avez sur la table (de lapin)

Si le mal a pénétré la partie 'contact', par le petit trou (par exemple du chocolat ou de l'eau ou toute autre substance susceptible d'oxyder les contacts : travail en atmosphère humide, salée ...) il est indispensable d'aller nettoyer la partie vitale du 'contact'.

Pour le démonter, décollez doucement le cadre en 'scotch transparent' colle sur la face en 'caoutchouc souple' et qui la maintient sur le 'cuvelage en plastique dur'. Vous découvrez le réel interrupteur de la touche constitué de trois couches successives à l'intérieur du 'cuvelage', qui sont du plus profond au plus superficiel : la première couche est une première épaisseur en cuivre se terminant par la '1ère patte de l'interrupteur' comportant en son centre deux petits 'reliefs' la deuxième couche est une épaisseur en 'isolant' évidée en son centre en forme de cadre et enfin la troisième couche est une deuxième épaisseur de cuivre se terminant par la '2ème patte de l'interrupteur' comportant en son centre deux 'avancées'.

#### Comment ça marche ? :

Quand vous appuyez sur une touche, le poussoir s'enfonce comprimant le 'ressort poussoir' (qui remonte la touche quand vous cessez d'appuyer dessus). Lorsque le poussoir s'enfonce il entraîne avec lui la 'pièce intermédiaire' qui comprime le 'ressort contact'. Etant comprimé, le 'ressort contact' transmet cette pression au 'contact' grâce à la jambe centrale de son E renversé qui appuie au centre de la face en 'caoutchouc souple' du 'contact'. La face en 'caoutchouc souple' se laisse déprimer grâce au trou de la face opposée du 'contact' sur sa face 'cuvelage' en 'plastique dur' qui laisse ainsi échapper l'air de l'intérieur du 'contact'. L'enfoncement de la face en 'caoutchouc souple' appuie sur les 'deux avancées' de la feuille en cuivre solidaire de la '2ème patte de l'interrupteur' qui viennent toucher les 'deux reliefs' de la feuille en cuivre solidaire de la '1ère patte de l'interrupteur', (dont ils étaient séparés par le cadre 'isolant' à l'état de repos), fermant ainsi le contact !!.

Pour rénovier le contact défaillant, il suffit de nettoyer les 'deux avancées' et les 'deux reliefs'. Il n'est pas superflu d'en profiter pour enlever toutes les crasses qui ont pu pénétrer le contact.

Le remontage s'effectue dans le sens inverse du démontage sans problème majeur.

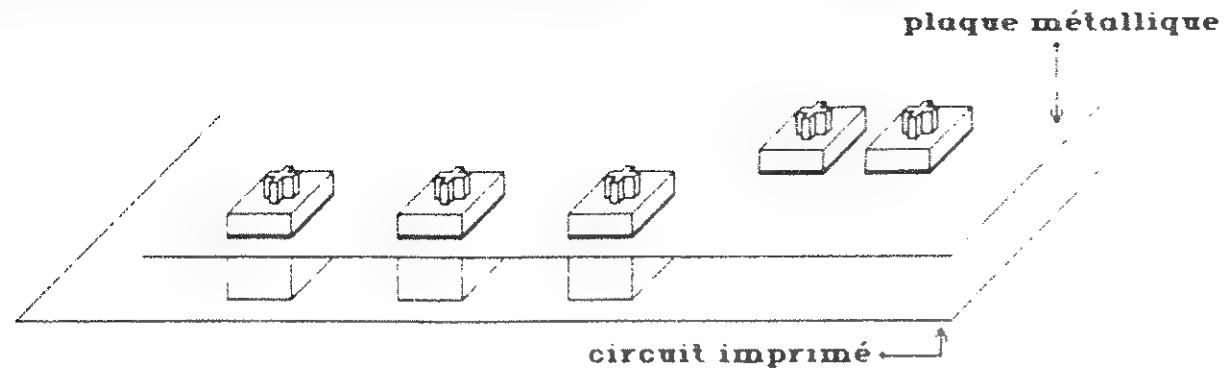
N'oubliez pas que cette démarche qui peut vous apparaître évidente à la lecture de ces lignes a nécessité de longues heures d'interrogations afin de trouver la VOIE et présenter un démontage qui autorise ensuite le remontage !!!

Il est formellement déconseillé de faire ce démontage aux parkinsoniens, aux éthyliques, aux myopes ou gens atteints de strabisme

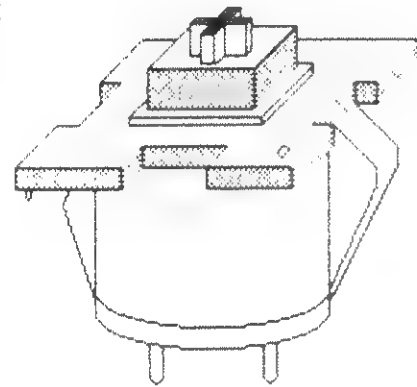
Le président d'IDC Bordeaux  
Delannay Bruno

# Clavier du DAI - nouveau modèle

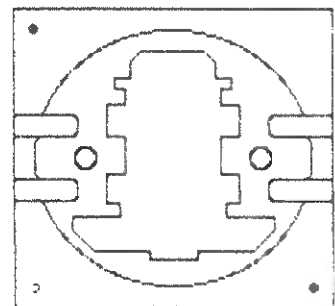
Vue générale du clavier avec touches (perspective cavalière)



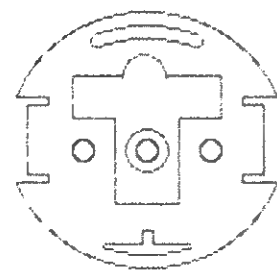
Vue générale d'une touche



Corps de touche - vue interne

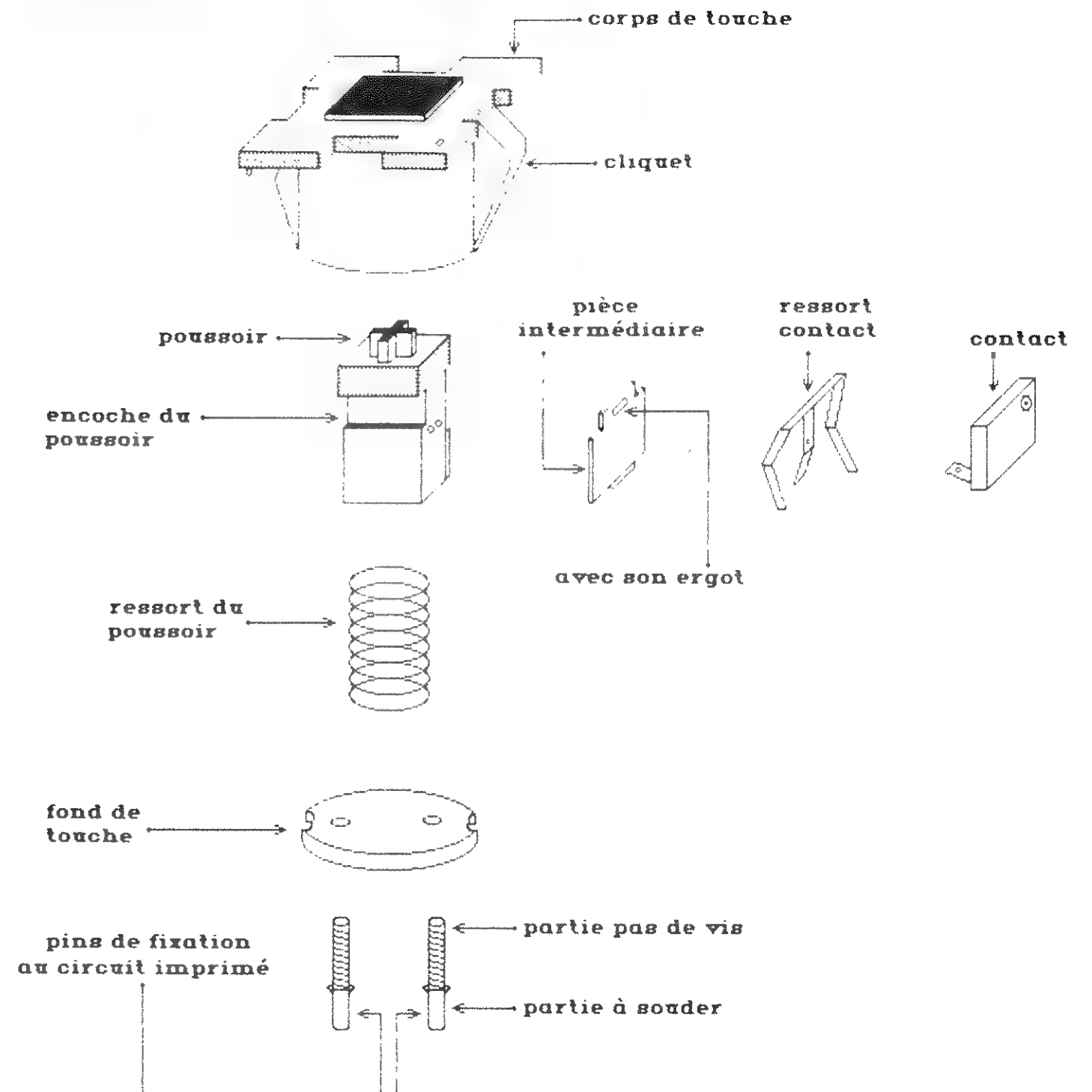


Fond de touche - vue interne

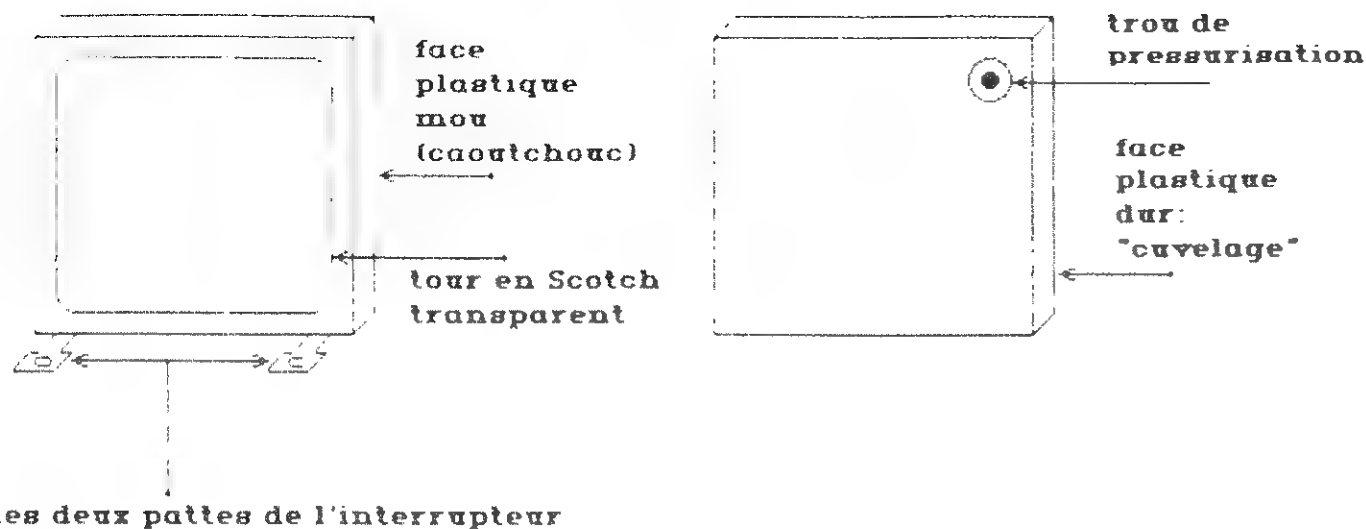


# Clavier du DAI - nouveau modèle

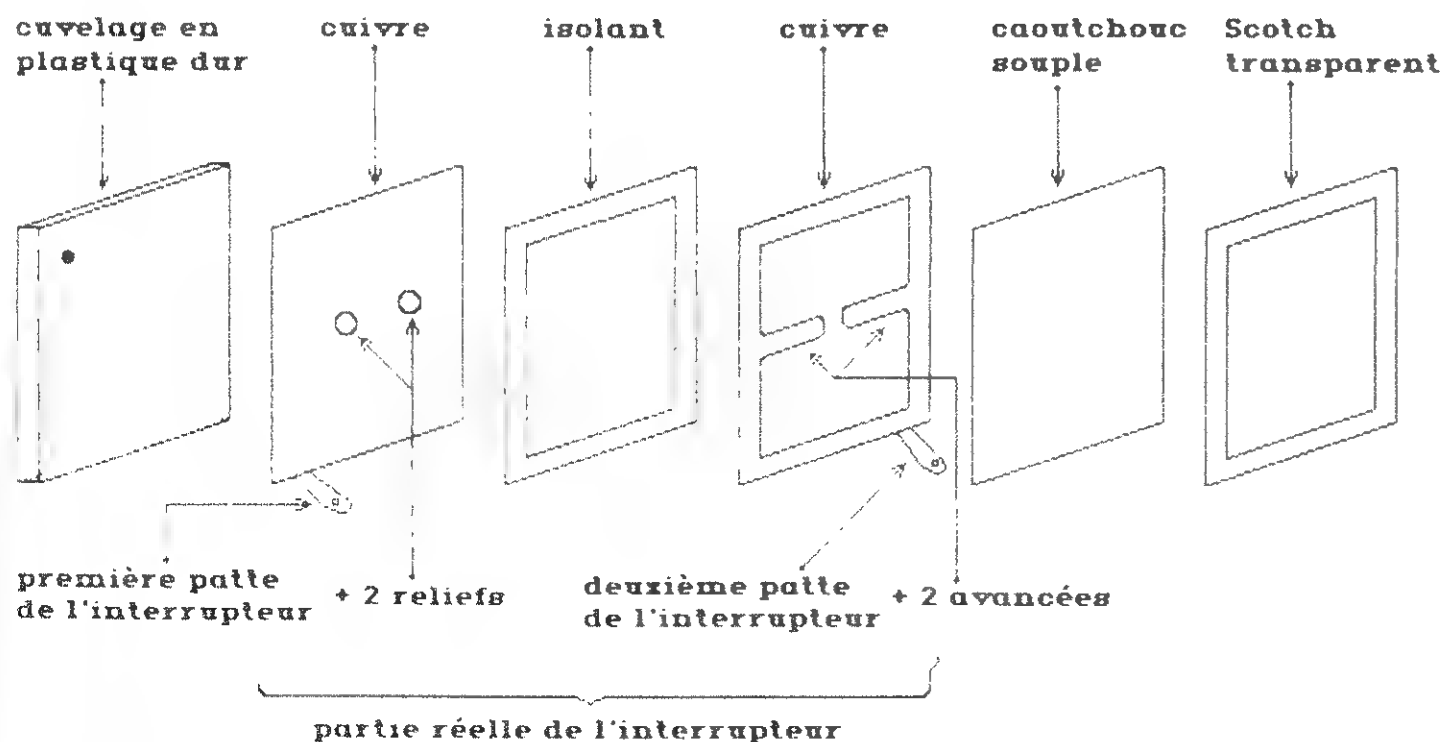
Vue éclatée



Le contact - de face et de dos



Le contact - vue éclatée



# Interruptions et communications

## Interruptions et communication



### ou comment utiliser les ressources (cachées ?) du DAI

Les fans de la communication (modem, imprimantes ou minitel) sont ou seront un jour ou l'autre confrontés au problème du manque de temps.

Prenons un exemple concret : le minitel, objet chéri par les membres d' IDC BORDEAUX. Les liaisons DAI-MINTEL se font à une vitesse de 1200 bauds, ce qui laisse entre chaque réception d'un caractère par le DAI à peu près 8 millisecondes. 8 millisecondes, c'est beaucoup pour la majorité des tâches que doit accomplir un émulateur minitel. Cependant, un simple *scrolling* de l'écran, même en mode 0, prend largement plus de temps que ces 8 millisecondes (Si vous ne me croyez pas, essayez et vous verrez que certains caractères ne sont pas pris en compte lors d'un *scrolling*).

Pour résoudre ces problèmes de communication, étant donné que la RS 232 du DAI ne possède pas de broche indiquant que l'ordinateur n'est pas disposé à recevoir un caractère, plusieurs solutions sont envisageables :

1) la solution du boeuf : mise au point par le célèbre mathématicien *Augustin Leboeuf* (1707-1783). Elle consiste à s'arranger pour qu'aucune tâche ne dépasse le temps fatidique (8 ms. dans le cas de notre bien-aimé minitel). C'est la méthode qui est utilisée dans "utilitaire minitel 3" distribué gratuitement (sauf les frais de transport) par IDC BORDEAUX. Dans ce programme les *scrollings* de l'écran sont évités, quand l'écran est plein on revient à la première ligne !

2) l'utilisation de l'interruption TV numéro 7 : toutes les 20 millisecondes une interruption 7 est effectuée. On peut l'employer pour ses propres besoins en mettant dans le vecteur d'interruption 7, l'adresse de sa propre routine qui scrutera l'entrée de la RS232 et qui, si un caractère vient d'être reçu le stockera dans un buffer d'attente (Cf. le programme d'horloge dans le hand-book pour l'utilisation de cette interruption). Il ne faut bien-sûr pas oublier, à la fin de sa propre routine, de faire un JUMP à l'adresse qui se trouvait initialement dans le vecteur d'interruption 7. Le principal inconvénient de cette méthode est ce temps de 20 milliseconde qui dans le monde de la communication inter-ordinateur fait figure d'une éternité, et qui ne permet que des communications à 300 bauds maximum.

3) l'utilisation d'un timer : la méthode est la même que celle exposée ci-dessus (nous reviendrons sur l'utilisation des timers), mais elle permet de s'affranchir de cette durée parfois trop longue de 20 ms. Il ne faut bien sûr pas oublier de recharger le timer à chaque fois et de mettre à 1 le bit correspondant du registre de masque.

4) l'utilisation des interruptions de communication : de loin la méthode la plus élégante. Cette méthode est détaillée dans la suite de l'article.

Avant de poursuivre, imprégnons notre esprit des saintes paroles de la Bible, j'ai nommé le célèbre **Firmware Manual**. Dans la rubrique "TICC : TIMMER + INTERRUPT CONTROLLER 5501: FFOO-FFFF", on voit qu'un seul circuit le 5501 gère à la fois la RS232 et les interruptions. On y apprend en outre que les interruptions 4 et 5 correspondent à la réception et à la transmission de données sur la RS232 (seule l'interruption relative à la réception présente un intérêt pour nous).

Mais comment ça marche ? C'est très simple ! ... Dès qu'un caractère est reçu sur la RS232, une interruption 4 a lieu. En fait pour que l'interruption ait effectivement lieu, il faut que les deux conditions suivantes soient réalisées :

1) Le bit 3 du registre de commande du 5501 (adresse : FFF4) doit être mis à 1 (bit 3 = interrupt acknowledge enable). Ceci est automatiquement réalisé lors de l'initialisation du DAI.

2) Le bit 4 du registre de masque des interruptions doit être à 1

A noter: Les interruptions du 5501 sont *bufferisées*, c'est à dire que si lors de l'interruption, le 8080 est dans le mode "interruptions interdites" (ordre assembleur DI), la demande d'interruption demeurera jusqu'à ce que le 8080 autorise à nouveau les interruptions (ordre assembleur EI). L'interruption sera alors exécutée.

Comment exploiter cette interruption : d'une part chaque caractère reçu est stocké dans un buffer. D'autre part une boucle lit les caractères du buffer et les traite au choix de l'utilisateur. Pour plus de clarté, passons tout de suite à l'application.

#### étude détaillée du programme

1) Le buffer: le buffer est une zone mémoire servant au stockage provisoire des caractères reçus sur la RS232. Il fonctionne en liaison avec deux pointeurs: un pointeur d'entrée (BUF2) et un pointeur de sortie (BUF1).

La routine INRBUF incrémente le pointeur de buffer contenu dans le registre HL du 8080. 2 cas se présentent :

-a) le pointeur pointe initialement une case mémoire au milieu du buffer, dont l'adresse est A, alors la case mémoire suivante dans le buffer est naturellement celle qui porte l'adresse A+1.

-b) le pointeur pointe la dernière case mémoire du buffer, la case mémoire suivante dans le buffer ne peut être alors choisie comme précédemment. La case suivante sera alors la première case mémoire du buffer.

BUF2 va pointer la case mémoire dans le buffer où l'on va entrer un caractère.

BUF1 pointe la position dans le buffer où l'on va lire un caractère

Le rôle de ces deux pointeurs étant défini, les commentaires du programme et, tout de même, un peu d'attention doivent permettre de comprendre le fonctionnement des routines OUTB et INB.

2) la gestion des interruptions : la mise en place de l'interruption 4 est faite par la routine COMON. L'action de cette routine est la suivante:

a) Mise à 1 du bit 4. La gestion du registre de masque est similaire à celle de POROM et PORO (adresse F006 et 40), il ne faut pas oublier de paker le registre de masque (FFF8) et sa copie en RAM (5F)!

b) Mise en place de l'adresse de notre propre routine (RECEPT) dans le vecteur d'interruption 4.

Il faut prendre la précaution d'interdire les interruptions (DI), afin d'éviter qu'une interruption puisse intervenir avant que toutes les opérations ci-dessus soient terminées (cela pourrait faire planter le DAI).

Ceci étant fait, dès qu'un caractère est reçu, une interruption 4 a lieu: cela signifie que le programme en cours d'exécution (ici c'est notre boucle de lecture -voir plus loin-) est interrompu et que la routine située à l'adresse 20H est exécutée. Cette routine en RAM est automatiquement mise en place lors de l'initialisation du DAI:

20	00	NOP		;Aucune action.
21	E5	PUSH H		;Sauvegarde le registre HL.
22	2A6A00	LHLD	V4	;HL est chargé avec l'adresse
				;contenue dans le vecteur 4.
25	E9	PCHL		;Effectue un JUMP à cette adresse.

Note : en fin de compte la routine dont l'adresse est dans V4 (ici RECEPT) est exécutée. Cette routine doit impérativement laisser intacts tous les registres du 8080. Il est alors prudent de sauvegarder (PUSH) tous les registres au début de notre propre routine (sauf HL qui est déjà sauvé par la routine ci-dessus) et de les restituer à la fin (POP).

Ceci étant dit, la routine RECEPT ne pose plus de difficulté.

3) Le programme principal: il est tout bête ! C'est une boucle qui lit les caractères du buffer et les affiche sur l'écran, c'est tout ! Dès que 50 caractères ont été affichés sur une même ligne, on passe à la ligne suivante.

Cette méthode de gestion de buffer peut s'appliquer à tout problème relatif à la communication informatique, alors bon amusement ! ... En attendant de voir comment nous l'avons mise en pratique dans notre émulateur MINITEL.

Sébastien Dubourg  
le 15/03/86

pour IDC BORDEAUX  
Président : Bruno Delannay  
Résidence : Les Acacias bat. B3  
Avenue de Saige 33600 PESSAC

TITL 'TEST INTERRUPTION DE COMMUNICATION par S.Dubourg'

```

; Sebastien Dubourg / IDC BORDEAUX le 9 fevrier 1986
;
;*****
;* CONSTANTES DU PROGRAMME *
;*****
@=0400 DEBUT EQU 400H ;Adresse de debut de programme.
ORG DEBUT
;
;*****
;* Adresses relatives au entree sortie *
;*****
@=FFF8 MASQUE EQU 0FFF8H ;Registre de masque du 5501.
@=005F RM EQU 5FH ;Copie en Ram du registre
; ;de masque.
@=006A V4 EQU 6AH ;Vecteur 4 d'interuption.
@=FFFD INPUT EQU 0FFFDH ;Port d'entree RS232 du 5501
@=FF05 BAUD EQU 0FF05H ;Rgistre de vitesse RS232.
;
;*****
;* Sous-programmes en ROM *
;*****
@=DE14 COMP EQU 0DE14H ;Compare HL et DE (HL-DE).
;
;***** TEST COMMUNICATION *****
;
3E88 TEST1 MVI A 88H ;1200 bauds/1 stop bit.
3205FF STA BAUD
21B304 LXI H BUFDEB ;Entree et sortie du
22ADD04 SHLD BUF1 ;buffer au debut du buffer.
22AF04 SHLD BUF2
CD8204 CALL COMON ;Met en service l'interupt.4
; ;pour la reception RS232.
0632 LOOP08 MVI B 32H ;nb max de car. sur 1 ligne.
CD5604 LOOP07 CALL OUTB ;Sort un car du buffer,si
DA1304 JC LOOP07 ;celui-ci n'est pas vide.
F680 ORI 80H ;Desactive les car. de
; ;controle.
EF RST 5 ;Affiche ce caracter sur
03 DB 3H ;l'ecran.
05 DCR B ;DCR le compteur de car.
C21304 JNZ LOOP07 ;Si c'est la fin de la ligne
3E0D MVI A 0DH ;passe a la ligne suivante.
EF RST 5
03 DB 3H
C31104 JMP LOOP08
;
;*****
;* Incremente HL en tant que pointeur de Buffer *
;*****
F5 INRBUF PUSH PSW
05 PUSH D
11B308 LXI D BUFFIN ;Si HL pointe la fin du

```

TEST INTERRUPTION DE COMMUNICATION par S.Dubourg

```

CD14DE CALL COMP ;Buffer, la nouvelle adresse
D1 POP D ;est celle du debut du
DA3904 JC LINRB1 ;buffer..
21B304 LXI H BUFDEB
F1 POP PSW
C9 RET
23 LINRB1 INX H
F1 POP PSW
C9 RET
;
;*****
;* Entre le contenu de l'accumulateur dans le buffer *
;*****
F5 INB PUSH PSW
2AAF04 LHLD BUF2 ;HL=adresse nouvelle d'
CD2804 CALL INRBUF ;entree du Buffer.
EB XCHG ;Si elle est egale a celle
2AAD04 LHLD BUF1 ;de la sortie du buffer,c'
CD14DE CALL COMP ;est que le buffer est plein
CA5404 JZ LINB1 ;exit sans rien changer.
F1 POP PSW ;Sinon,sauvegarder la
EB XCHG ;nouvelle adresse du
22AF04 SHLD BUF2 ;pointeur de sortie du buffer,
77 MOV M,A ;et le car. dans le buffer.
C9 RET
;
F1 LINB1 POP PSW
C9 RET
;
;*****
;* Sort un car. du buffer et le met dans l'accumulateur *
;*****
2AAF04 OUTB LHLD BUF2 ;Si le pointeur d'entree et
EB XCHG ;le pointeur de sortie du
2AAD04 LHLD BUF1 ;buffer sont les meme,c'est
CD14DE CALL COMP ;que le buffer est vide;
C26504 JNZ LIQUB1 ;mettre alors le carry a 1
37 STC ;pour indique que le buffer
C9 RET ;et vide.
;
CD2804 LIQUB1 CALL INRBUF ;Sinon.HL=add de la sortie,
7E MOV A,M ;A=caractere lu dans le buf.
B7 ORA A ;Met le carry a 0.
22ADD04 SHLD BUF1 ;sauvegarde le nouveau
C9 RET ;pointeur de sortie de buf.
;
;*****
;* Poke le caractere recu sur la RS232 dans le buffer *
;*****
;
;Appelle par l'interruption 4 liee a la reception sur RS232.
;
F3 RECEPT DI ;Interdit les inrteruptions
F5 PUSH PSW ;pour eviter les STACK
C5 PUSH B ;OVERFLOW et sauvegarde tous
D5 PUSH D ;les registres.

```



# BASIC : caractères couleur

```
TEST INTERUPTION DE COMMUNICATION par S.Dubourg

3AF0FF LDA INPUT ;Lit le car. recu sur RS232.
B7 ORA A ;Si c'est 0 ne pas le
CA7C04 JZ LIREC ;prendre en compte, sinon le
CD3C04 CALL INB ;rentrer dans le buffer.
D1 LIREC POP D ;Recupere tous les registres
C1 POP B ;et reprendre l'execution
F1 POP PSW ;du programme principal.
E1 POP H
FB EI
C9 RET

;
;*****
;* Mise en marche de la routine de reception *
;*****
;
F3 COMON DI
3A5F00 LDA RM ;Autorise l'interruption 4
E610 ANI 10H ;(reception sur RS232).
325F00 STA RM
32F8FF STA MASQUE
2A6A00 LHLD V4 ;Vecteur 4 = pgm de
22B104 SHLD RV4 ;reception, et sauvegarde l'
216E04 LXI H RECEPT ;ancien vecteur 4 (routines
226A00 SHLD V4 ;mathematiques).
FB EI
C9 RET

;
;*****
;* Arrete la routine de reception *
;*****
;
F3 COMOFF DI
3EC4 MVI A 0C4H ;Interdit l'interruption 4.
325F00 STA RM
32F8FF STA MASQUE
2AB104 LHLD RV4 ;Recupere l'ancienne adresse
226A00 SHLD V4 ;du vecteur 4 (math.).
FB EI
C9 RET

;
;
BUF1 DS 2H ;Pointeur de sortie de buff.
BUF2 DS 2H ;Pointeur d'entree de buff.
RV4 DS 2H ;Zone de sauvegarde du
; vecteur 4 d'origine.
;
;
@=0400 BUFLON EQU 400H ;Longueur du buffer.
BUFDEB DS BUFLON ;Debut du buffer.
@=0883 BUFFIN EQU BUFLON+BUFDEB ;Fin du buffer.
;
IDCBX END
```

```
1 REM CARATERES EN COULEUR EN MODE 16 COULEURS
2 REM LA VARIABLE A REPRESENTE LA COULEUR DE TEXTE = 0
3 REM LA COULEUR DE FOND EST 8
4 LET A=#8
7 REM METTRE LA LIGNE EN MODE 16 COULEURS
10 REM LIGNE 2
11 POKE #B4F1,*FA
15 REM
25 REM METTRE SUR LA LIGNE 2. LA COULEUR ET LE CARACTERE CHOISIS
26 REM
27 REM
28 REM BOUCLE POUR CHAQUE POSITION DE LA LIGNE 2
29 FOR X=#B4F0-132 TO #B4F0-2 STEP 2
30 REM
31 REM METTRE LA COULEUR DU TEXTE
32 POKE X,A
33 REM
34 REM METTRE LE CARACTERE ,*41 = A
35 POKE X+1,*41
36 REM
37 REM AUGMENTER DE 1 LA COULEUR DU TEXTE
38 REM SI DEPASSEMENT REMETTRE A ZERO
39 REM
40 A=A+16:IF A>240 THEN LET A=#0
41 REM
42 REM POSITION SUIVANTE POUR LA LIGNE
43 NEXT X
44 REM
```

## ACCELEREZ LA VITESSE DE LECTURE D'UN PROGRAMME SUR CASSETTE AUDIO

Vous connaissez peut-être l'astuce qui consiste à accélérer la vitesse de votre cassette pour relire vos programmes beaucoup plus vite que la normale.

Cette méthode comporte cependant un inconvénient majeur : ceci n'est possible qu'après lecture de la première partie du programme (le LEADER). Il n'est donc pas possible de relire l'intégralité d'un programme standard à grande vitesse.

Pour rendre cela possible, il suffit d'augmenter la période du LEADER. La valeur de cette période dépend de deux données stockées aux adresses #2E6 et #2E7. A l'initialisation de la machine ces données prennent comme valeur : #24. Il faut tout simplement augmenter ces valeurs : à #80 par exemple.

En résumé, il vous suffit de taper POKE #2E6,#80 et POKE #2E7,#80 avant d'enregistrer un programme. Vous pourrez ensuite, moyennant une petite "bidouille" sur votre cassette (pour en augmenter la vitesse de lecture), relire votre programme à grande vitesse.

Laurent LEGRY

# AMD 9511 : BUG

## UN BUG DANS L'AMD 9511

Claude PICARD, CHALON SUR SAONE (F)

L'instruction **INOT** ne fonctionne pas correctement, quand le processeur arithmétique **AMD 9511** est en service sur le **DAI**. En effet, **INOT 0 = -1**, alors que l'**AMD** répond **INOT 0 = 1**. Pour corriger cette erreur, il convient de dévalider l'**AMD** quand on emploie **INOT**. Ceci se fait très simplement par **POKE #D4,0**. On peut alors employer **INOT**, qui est normalement évalué par le basic **DAI**. Ce calcul effectué, on peut alors remettre l'**AMD** en service, par **POKE #D4,#7B**.

Si cette manoeuvre vous ennuie, il reste la solution de tenir compte du bug, dans la mesure où l'on sait que:

sans **AMD** **INOT A -- A - 1**  
avec **AMD** **INOT A -- A + 1**

L'adresse **#D4** contient le déplacement (0 ou **#7B**) dans la table des sauts aux fonctions mathématiques. Attention donc, toute autre valeur dans **#D4** "plante" irrémédiablement le système, aucun calcul ne pouvant plus être effectué correctement. Par exemple, la procédure de reconnaissance de la présence de l'**AMD**, décrite dans le manuel en français du **DAI** (p. 141) est à éviter... si on a un programme en mémoire ! Il est plus simple et moins dangereux de lire **#D4**, pour être fixé. Si le doute persiste, il reste la ressource de lever le capot: non seulement un co-processeur arithmétique ne passe pas inaperçu dans une machine, mais en plus sa voracité en alimentation électrique le distingue aisément: c'est le seul circuit intégré qui brûle franchement le doigt, au bout de quelques instants de fonctionnement !

# Errata

## CARTE CENTRONICS : ERRATUM

Une petite erreur s'est glissée dans l'article sur la carte CENTRONICS.

En effet, la figure qui représente les deux faces du circuit imprimé et qui donne le brochage de la carte doit être légèrement modifiée. Il faut tout simplement inverser les appellations **DTR1** et **DTR2**.

Laurent LEGRY

# jaargang '85 gemist?

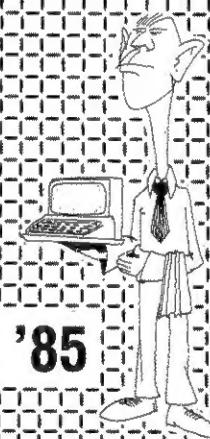
EEN UITGAVE VAN MSX-CLUB BELGIE - NEDERLAND

# MSX

p.a. MOTTAART 20 3170 HERSELT 014/ 54 59 74

## VOLUME

### JAARGANG '85



Alle programma's en artikels uit **MSX CLUB MAGAZINE** jaargang '85 zijn nu verkrijgbaar in een handig jaarboek.

136 bladzijden

uren lees- en programmeerplezier !

## prijs :

jaarboek '85 :	385 Fr/FI 20
verzamelcassette :	600 Fr/FI 33
jaarboek + cassette :	900 Fr/FI 50
3 1/2 inch floppy :	900 Fr/FI 50
jaarboek + 3 1/2 inch floppy :	1.200 Fr/FI 67

bespaar Uzelf uren tikwerk en frustratie : de verzamelcassette bevat alle programma's welke in het boek gepubliceerd zijn.

# MSX CLUB magazine

## VOLUME

### JAARGANG '85

1985

#### INHOUDSTAFEL

- Spaceman
- Blokdiagram
- Getallenschrijver
- MSX-monitor
- Seawar
- Coördinatenzoeker
- MSX-logo
- Soundmaster
- Computer adventure
- Sprite-maker
- Potpourri
- Gon Fies
- Para
- Drijfzand
- Music box
- Kleurpolloden
- Basic Wordprocessor
- Time
- Kamertje tekenen
- Lintograph
- Adressenbestand
- Doolhof

**MSX-club**

p.a. Mottaart 20 - 3170 Herselt  
Tel. 014/ 54 59 74