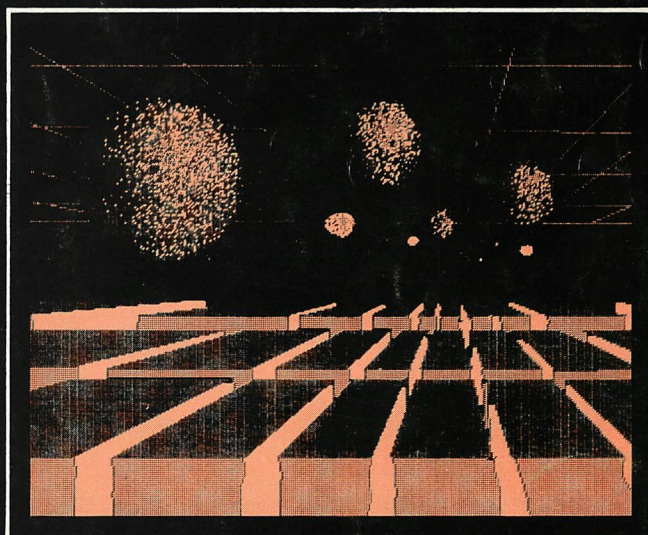
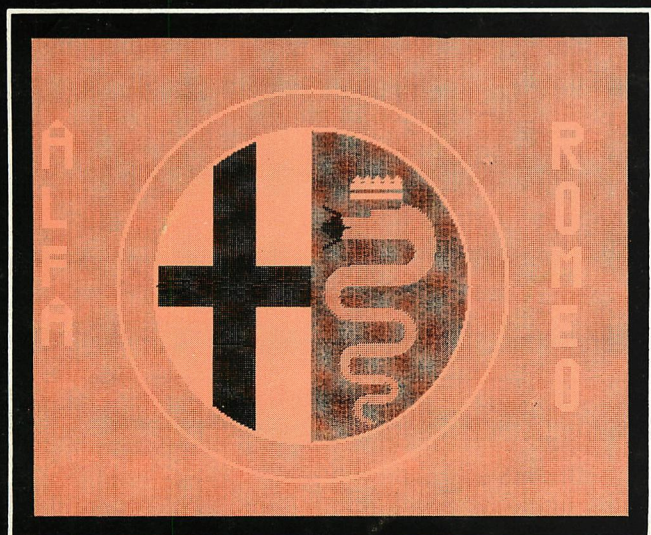
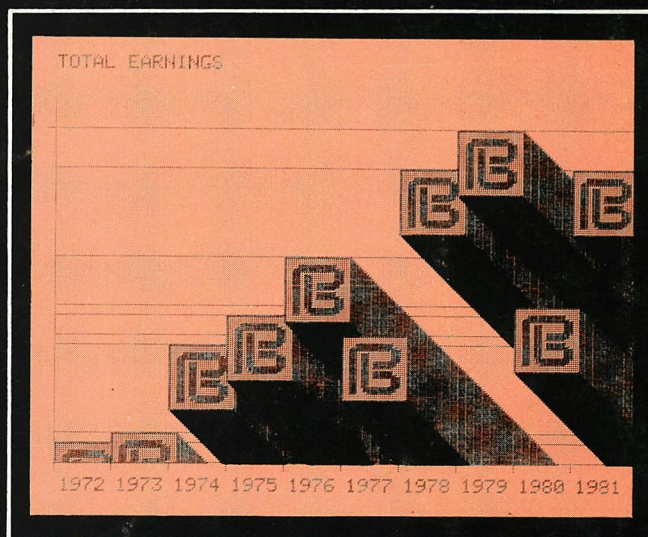


# DAI NAMIC

12

tweemaandelijks tijdschrift september - oktober 1982



**personal computer users club**

een uitgave van dainamic v.z.w.  
verantw. uitgever w. hermans, heide 4 - 3171 westmeerbeek

COLOFON

DAInamic verschijnt tweemaandelijks.  
 abonnementsprijs is inbegrepen in de  
 jaarlijkse contributie : 750 Bfr.

Bij toetreding worden de verschenen  
 nummers van de jaargang toegezonden.

DAInamic redactie :

- Dirk Bonné
- Freddy De Raedt
- Wilfried Hermans
- René Rens
- Jos Schepens
- Roger Theeuws
- Bruno Van Rompaey
- Jef Verwimp

Vormgeving : Ludo Van Mechelen.

U wordt lid door storting van de  
 contributie op het rekeningnr.  
230-0045353-74 van de Generale Bank-  
maatschappij, Leuven, via bankinstel-  
 ling 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  
 Heide 4  
 B 3171 Westmeerbeek  
 België

tel. : 016/69.86.23

Kredietbank Westmeerbeek  
 nr. 406-3016141-33

BTW : 420.840.834

Lidgelden

Bruno Van Rompaey  
 Bovenbosstraat 4  
 B 3044 Haasrode  
 België

tel. : 016/46.10.85

Generale Bankmaatschappij Leuven  
 nr. 230-0045353-74

Inzendingen : Games & Strategy

Frank Druijff  
 's Gravendijkwal 5A  
 NL 3021 EA Rotterdam  
 Nederland

tel. : 010/25.42.75

# DAINAMIC

PERSONAL COMPUTER USERS CLUB

4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

belangrijke ASCII-waarden in DAIPc

functie/symbool	HEX	DEC
back-space	8	8
TAB	9	9
linefeed	A	10
clear screen	C	12
CURSOR UP	10	16
CURSOR DOWN	11	17
CURSOR LEFT	12	18
CURSOR RIGHT	13	19
space-bar	20	32
Ø	30	48
A	41	65
a	61	97
pijltje rechts	89	137
pijltje links	88	136
pijltje boven	5E	94
pijltje onder	8C	140
volle blok	FF	255
verticale lijn	A	10
horizontale lijn	B	11
6 hor.lijnen	1D	29

ASCII - HEX - ASCII CONVERSION TABLE

	MSD	0	1	2	3	4	5	6	7
LSD	000	001	010	011	100	101	110	111	
0	0000	NUL	DLE	SP	0	⊙	P	∕	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	⌘	4	D	T	d	t
5	0101	ENG	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	VS	/	?	O	←	o	DEL

# DAI

Westmeerbeek okt 82

Beste leden,

Het thema van dit nummer is duidelijk de VIDEO SCREEN RAM van onze DAI. We waren ook al maanden aan het zwoegen om MODE 7 & 8 te realiseren, echter zonder afgewerkt resultaat. Komt dan onze vriend LOOIJE te voorschijn met een volledige en diepgaande studie over de VIDEO RAM ... Mooi is dat ! Het verhaal " DAI video screen ram" is de studie, met grondige uitleg over de gebruikte pointers. "Screen driver demo" is een programma dat U toelaat eindeloze combinaties van tekst & grafische modes te creëren. Op p. 261 vindt U de objectcode-dumps voor MODE 7&8. Indien het nog niet duidelijk is hoe U dergelijke programmatuur aan de praat krijgt, lees dan ook eens de beschouwingen op p.266. Jan Boerrigter maakt de informatie over DAI-video screen rond op p.255. Op p. 282 kan U ook lezen dat het FIRMWARE-boek spoedig verkrijgbaar zal zijn.

6 november houden we onze volgende bijeenkomst in de stadsfeestzaal te Aarschot. Deze keer hebben we ook andere verenigingen en handelaars aangesproken. Mogelijk wordt deze MIKROBEL-dag een goede promotie voor het computergebeuren in België en omgeving. U kan de bijgevoegde affiche rustig laten zitten, ophangen in school of kantoor kan natuurlijk ook ! DAI ( sorry : INDATA ), heeft beloofd op 6 november een aantal interessante uitbreidingen voor DAI te tonen, dus : kijk uit ! Alhoewel de programma's en artikels rijkelijk op de redactie binnenstromen , zijn uw bijdragen nog steeds erg welkom.

dank aan alle auteurs, tot 6 november...

Dear members,

In this issue you will find a lot of information about DAI VIDEO SCREEN RAM. Thanks to N.LOOIJE, who offers us MODE 7&8 : 512 x 244 resolution ! Our next meeting is in Aarschot (stadsfeestzaal-city hall) on 6 nov. You can find all participants on the enclosed poster. INDATA will show a lot of new extensions for DAIpc ... We receive a lot of articles and programs, please continue.

see you on 6 nov

Wilfried Hermans

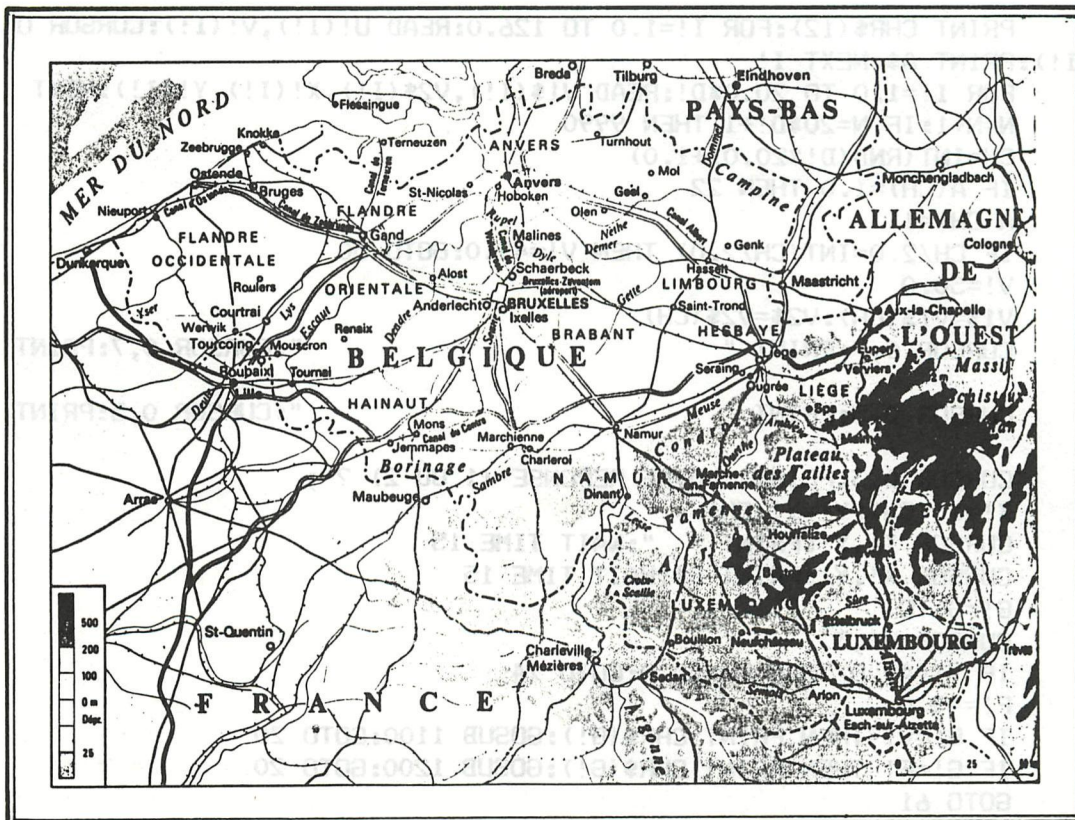
mikrobel  
6 nov 82  
Aarschot

# INHOUD — CONTENTS

231	Remark	Redactiepraatje
232	Bladwijzer	
233	Geographie de la Belgique	J.Moens
234	"	
235	"	
236	Demo IAND-IOR-MOD-IXOR-SHL-SHR	W.Hermans
237	"	
238	Ellen	E.Smet
239	"	
240	"	
241	"	
242	Screen Driver Demo	N.P.Looije
243	"	
244	"	
245	"	
246	"	
247	" + Polygons, Dot, Filling	F.Druijff
248	DAI video screen ram	N.P.Looije
249	"	
250	"	
251	"	
252	"	
253	"	
254	"	
255	Screen constants graphic modes	J.Boerrigter
256	Screen memory management	J.Boerrigter
257	"	
258	Kalenderalgoritmen	P.Lelie
259	"	
260	"	
261	" + 512 x 244 (MODE 7 & 8)	N.P.Looije
262	Fascination	DAInamic France
263	Super Noise Generator	W.De Winter
264	How to extend DAI-BASIC	J.Boerrigter
265	"	
266	Audio Cassette Marker	W.De Winter-Hermans
267	" + Lines	F.Druijff
268	Efficient Circle-drawing	F.Druijff
269	"	
270	"	
271	"	
272	Dessin-Drawing-16 colors-oiseau	F.Wittendal
273	Groeten uit België	R.Rens
274	Hardcopy MODE 0	R.Leuenberger
275	"	
276	"	
277	"	
278	TV-tennis	Luc Maes
279	"	
280	"	
281	INFO-INFO-INFO	
282	DAIpc firmware manual	J.Boerrigter
283	Bootstrap Loader V2	W.De Winter-De Raedt
284	"	
285	"	
286	"	
287	"	
288	" + FGT-check	
289	Conversion APPLE-ATARI-DAI	F.Verstegen
290	"	
291	" + New extensions for DAI + tip	
292	Subroutine cassettebesturing	I.Broekman
293	"	
294	"	

No part of this book may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the publisher.  
 Niets uit deze uitgave mag worden vervoerd, vervoerd en/of openbaar gemaakt door middel van druk, fotocopie, microfilm of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

# belgi-quiz



## program identification

title           ◦ Geographie de la Belgique  
 author          ◦ J.Moens  
 purpose        ◦ Learn the cities of Belgium  
 comment        ◦ First find Belgium !!!

```

1  REM
2  REM -----
3  REM GEOGRAPHIE DE LA BELGIQUE - 5 NIVEAUX
4  REM -----
5  REM COPYRIGHT JACQUES MOENS - FEVRIER 1982
6  REM -----
7  REM
8  PRINT CHR$(12):POKE #75,32:CLEAR 6000:ENVELOPE 0 15
9  DIM A(100.0),U!(126.0),V!(126.0),V1$(100.0),V2$(100.0),X!(100.0),Y!(1
00.0)
10 N=0:P=0:A$=CHR$(127):C$=CHR$(129)+CHR$(130)
11 COLORT 14 0 0 0
12 PRINT :PRINT :PRINT "GEOGRAPHIE DE LA BELGIQUE":PRINT "
-----"
13 PRINT :INPUT "DEGRE DE DIFFICULTE (1 A 5) ";D!
14 IF D!>0.0 AND D!<6.0 THEN 18
15 GOTO 15
  
```

```

18 PRINT CHR$(12):FOR I!=1.0 TO 126.0:READ U!(I!),V!(I!):CURSOR U!(I!),V
!(I!):PRINT A$:NEXT I!
19 FOR I!=1.0 TO 20.0*D!:READ V1$(I!),V2$(I!),X!(I!),Y!(I!):NEXT
20 N=N+1:IF N=20*D!+1 THEN 9990
22 CH=INT(RND(D!*20.0)+1.0)
24 IF A(CH)=1.0 THEN 22
26 A(CH)=1
28 IF CH/2.0=INT(CH/2.0) THEN V!=49.0:GOTO 50
29 V!=50.0
50 V1$=V1$(CH):V2$=V2$(CH)
51 CURSOR 0,7:PRINT " " " :CURSOR 0,7:PRINT "1 ";V
1$
52 CURSOR 0,5:PRINT " " " :CURSOR 0,5:PRINT "2 ";V
2$
53 CURSOR 0,3:PRINT "VOTRE REPONSE (1 OU 2) ?";
60 X!=X!(CH):Y!=Y!(CH)
61 CURSOR X!,Y!:PRINT " " :WAIT TIME 15
62 CURSOR X!,Y!:PRINT C$:WAIT TIME 15
63 G!=GETC
71 CURSOR 26,3
72 IF V!=49.0 THEN F!=50.0:GOTO 74
73 F!=49.0
74 IF G!=V! THEN PRINT CHR$(G!):GOSUB 1100:GOTO 20
75 IF G!=F! THEN PRINT CHR$(G!):GOSUB 1200:GOTO 20
76 GOTO 61
1100 CURSOR 0,1:PRINT "BRAVO":P=P+1
1104 CURSOR 9,1:PRINT P;" POINT";:IF P>1.0 THEN PRINT "S";
1106 PRINT " SUR";N
1110 SOUND 0 0 15 0 FREQ(400.0):WAIT TIME 80:SOUND OFF
1115 CURSOR X!,Y!:PRINT " "
1120 RETURN
1200 CURSOR 0,1:PRINT "FAUX "
1204 CURSOR 9,1:PRINT P;" POINT";:IF P>1.0 THEN PRINT "S";
1206 PRINT " SUR";N
1210 SOUND 0 0 15 0 FREQ(40.0):WAIT TIME 80:SOUND OFF
1215 CURSOR X!,Y!:PRINT " "
1220 RETURN
9020 DATA 5,16,5,17,5,18,6,18,7,19,8,19,9,19,10,20,11,20,12,20,13,21,14,21
,15,21,16,20,17,20,18,20,19,21,20,20,21,20,22,20
9040 DATA 23,20,24,20,25,20,26,21,27,21,28,21,29,21,30,22,31,23,32,22,33,2
2,34,23,35,22,36,22,37,23,38,23,39,22,39,21,40,21,41,21
9060 DATA 42,20,43,20,44,20,45,20,46,20,47,20,48,19,48,18,47,17,46,16,46,1
5,47,15,48,15,49,15,50,15,51,15,52,14,53,14,54,13,54,12
9080 DATA 55,12,56,11,56,10,55,9,54,9,53,8,52,8,51,8,50,7,49,6,48,5,49,4,5
0,3,50,2,49,1,48,1,47,1,46,1,45,1,44,1
9100 DATA 43,1,42,2,41,2,40,2,39,3,38,3,37,3,36,4,36,5,36,6,36,7,35,7,34,7
,33,6,32,5,31,5,30,5,29,5,28,5,27,5
9120 DATA 26,6,27,7,26,8,26,9,25,9,24,10,23,10,22,10,21,10,20,10,19,11,18,
11,17,11,16,11,15,12,14,12,14,13,14,14,13,14,12,14
9140 DATA 11,14,10,14,9,14,8,14,7,15,6,15
9200 DATA "LOUVAIN (LEUVEN)","BRUXELLES (BRUSSEL)",30,15,"ANVERS (ANTWERPE
N)","GAND (GENT)",29,20
9210 DATA "ANVERS (ANTWERPEN)","GAND (GENT)",21,17,"BRUGES (BRUGGE)","OSTE
NDE (OOSTENDE)",14,19
9220 DATA "TONGRES (TONGEREN)","HASSELT",42,16,"MONS","TOURNAI",25,11
9230 DATA "HUY","NAMUR",36,11,"LIEGE","VERVIERS",45,13
9240 DATA "VIRTON","ARLON",48,3,"LOUVAIN (LEUVEN)","BRUXELLES (BRUSSEL)",3
4,15
9250 DATA "BRUGES (BRUGGE)","OSTENDE (OOSTENDE)",10,19,"TONGRES (TONGEREN)
","HASSELT",44,14

```

9260 DATA "MONS", "TOURNAI", 17, 12, "HUY", "NAMUR", 42, 12  
 9270 DATA "LIEGE", "VERVIERS", 49, 12, "VIRTON", "ARLON", 45, 2  
 9280 DATA "NAMUR", "CHARLEROI", 31, 11, "MARCHE", "BASTOGNE", 44, 9  
 9290 DATA "BASTOGNE", "NEUFCHATEAU", 45, 4, "BASTOGNE", "MARCHE", 47, 7  
 9300 DATA "FURNES (VEURNE)", "DIXMUDE (DIKSMUIDE)", 9, 17, "FURNES (VEURNE)",  
 DIXMUDE (DIKSMUIDE)", 7, 17  
 9310 DATA "YPRES (IEPER)", "ROULERS (ROESELAEERE)", 12, 16, "YPRES (IEPER)", "RO  
 ULERS (ROESELAEERE)", 11, 15  
 9320 DATA "EEKLO", "ST-NICOLAS (ST-NIKLAAS)", 26, 19, "TIELT", "BRUGES (BRUGGE)  
 ", 18, 16  
 9330 DATA "ALOST (AALST)", "TERMONDE (DENDERMONDE)", 25, 17, "EEKLO", "GAND (GE  
 NT)", 19, 19  
 9340 DATA "TERMONDE (DENDERMONDE)", "ALOST (AALST)", 26, 16, "COURTRAI (KORTRI  
 JK)", "AUDENARDE (OUDENAARDE)", 14, 15  
 9350 DATA "COURTRAI (KORTRIJK)", "AUDENARDE (OUDENAARDE)", 18, 15, "ATH", "SOIG  
 NIES", 23, 13  
 9360 DATA "ATH", "SOIGNIES", 27, 12, "THUIN", "CHARLEROI", 29, 9  
 9370 DATA "GENK", "MAASEIK", 46, 18, "DINANT", "NAMUR", 36, 9  
 9380 DATA "LIEGE", "WAREMME", 42, 13, "MALINES (MECHELEN)", "WILLEBROECK", 31, 17  
 9390 DATA "CHIMAY", "PHILIPPEVILLE", 32, 8, "WAVRE", "NIVELLES", 33, 14  
 9400 DATA "WAVRE", "NIVELLES", 30, 12, "GENK", "MAASEIK", 44, 17  
 9410 DATA "OSTENDE (OOSTENDE)", "NIEUPOORT (NIEUWPOORT)", 8, 18, "EEKLO", "BRUGE  
 S (BRUGGE)", 19, 19  
 9420 DATA "BRUGES (BRUGGE)", "ZEEBRUGGE", 13, 20, "COURTRAI (KORTRIJK)", "MOUSC  
 RON (MOESKROEN)", 14, 15  
 9430 DATA "COURTRAI (KORTRIJK)", "MOUSCRON (MOESKROEN)", 15, 14, "NINOVE", "GRA  
 MMONT", 26, 15  
 9440 DATA "MARCHE", "ST-HUBERT", 43, 6, "EUPEN", "VERVIERS", 51, 13  
 9450 DATA "NINOVE", "GRAMMONT", 22, 14, "CHIMAY", "PHILIPPEVILLE", 29, 6  
 9460 DATA "MALINES (MECHELEN)", "WILLEBROECK", 29, 17, "TURNHOUT", "TORHOUT", 37  
 , 20  
 9470 DATA "TURNHOUT", "TORHOUT", 12, 17, "TIRLEMONT (TIENEN)", "TONGRES (TONGER  
 EN)", 37, 14  
 9480 DATA "ARLON", "BOUILLON", 40, 3, "CINEY", "ROCHEFORT", 40, 9  
 9490 DATA "LAROUCHE", "ROCHEFORT", 42, 8, "LAROUCHE", "SAINT-HUBERT", 46, 8  
 9500 DATA "BASTOGNE", "MARTELANGE", 47, 4, "STAVELOT", "MALMEDY", 50, 10  
 9510 DATA "STAVELOT", "MALMEDY", 52, 11, "NINOVE", "ALOST (AALST)", 26, 15  
 9520 DATA "HUY", "ANDENNE", 40, 11, "LOKEREN", "ST-NICOLAS (ST-NIKLAAS)", 24, 18  
 9530 DATA "TIRLEMONT (TIENEN)", "ST-TROND (ST-TRUIDEN)", 40, 14, "TIRLEMONT (T  
 IENEN)", "ST-TROND (ST-TRUIDEN)", 37, 14  
 9540 DATA "COUVIN", "CHIMAY", 29, 6, "COUVIN", "CHIMAY", 31, 6  
 9550 DATA "ATH", "LESSINES", 25, 14, "DIEST", "AARSCHOT", 38, 17  
 9560 DATA "DIEST", "AARSCHOT", 36, 16, "ENGHIEN", "SOIGNIES", 27, 13  
 9570 DATA "TOURNAI", "PERUWELZ", 21, 11, "SAINT-VITH", "MALMEDY", 52, 9  
 9580 DATA "THUIN", "BEAUMONT", 28, 8, "LIERRE (LIER)", "MALINES (MECHELEN)", 32,  
 19  
 9590 DATA "LAROUCHE", "HOUFFALIZE", 48, 8, "VIELSALM", "SAINT-VITH", 50, 9  
 9600 DATA "YPRES (IEPER)", "POPERINGE", 8, 15, "DURBUY", "MARCHE", 44, 10  
 9610 DATA "GENK", "OVERPELT", 44, 19, "BEAURAING", "GEDINNE", 38, 7  
 9620 DATA "BEAURAING", "GEDINNE", 38, 5, "DEINZE", "AUDENARDE (OUDENAARDE)", 18,  
 16  
 9630 DATA "TURNHOUT", "GEEL", 38, 19, "RENAIX (RONSE)", "MOUSCRON (MOESKROEN)",  
 18, 14  
 9640 DATA "NIVELLES", "BRAINE-LE-COMTE", 28, 12, "BOOM", "ANVERS (ANTWERPEN)", 2  
 9, 18  
 9650 DATA "CHARLEROI", "GOSSELIES", 31, 12, "SPA", "STAVELOT", 49, 11  
 9660 DATA "MONS", "BINCHE", 27, 10, "VISE", "LIEGE", 46, 14  
 9670 DATA "BOUILLON", "FLORENVILLE", 43, 2, "PERWEZ", "PERUWELZ", 35, 13  
 9680 DATA "MAUBEUGE", "VALENCIENNES", 18, 10, "MAUBEUGE", "VALENCIENNES", 23, 9  
 9690 DATA "LILLE", "ROUBAIX", 12, 13, "LILLE", "ROUBAIX", 11, 12  
 9990 CURSOR 0,1:PRINT "TERMINE":POKE #75,95:END  
 9995 REM PROGRAMME ECRIT PAR JACQUES MOENS  
 9996 REM .....CLOS FONTAINE DES DUCS,6  
 9997 REM .....1310 LA HULPE  
 9998 REM .....T. 02/657.95.60  
 9999 REM \_\_\_\_\_

## program identification

title : DEMO-program IAND,IOR,MOD,IXOR,SHL,SHR  
author : W.Hermans  
purpose : demonstrates BIT-manipulations in DAI-BASIC  
comment : MOD is not a bit-manipulation

```
10 REM DEMO-PROGRAMMA VOOR IAND, IOR, MOD, IXOR, SHL, SHR
20 REM -----
100 CLEAR 4000:MODE 0:PRINT CHR$(12)
105 POKE #74,0:REM CURSOR IN ALTERNATE COLOURS
110 COLORT 8 0 5 15
120 CURSOR 0,23:PRINT "-----INTEGER BIT OPERATIO
NS-----"
200 REM INPUT OPERATION
210 REM -----
215 POKE #75,0:REM INVISIBLE CURSOR
220 RESTORE:FOR X=1 TO 6:READ A$:CURSOR X*8,20:PRINT A$:CU
RSOR X*8,18:PRINT X:NEXT
225 G=GETC:IF G<>0 THEN 255:REM CLEAR BUFFER
230 CURSOR 5,15:PRINT "OPERATION 1...6"
240 G=GETC:IF G=0 THEN 240
250 IF G<#31 OR G>#36 THEN 225:REM FALSE ENTRY
255 OP=G-#30
260 FOR X=OP*8-2 TO OP*8+5:CURSOR X,20:POKE #76,#FF:CURSOR
X,21:POKE #76,#FF:NEXT:REM COLORED BACKGROUND
300 REM INPUT OPERATORS
310 REM -----
315 POKE #75,255:REM VISIBLE CURSOR
320 CURSOR 5,13:INPUT "OPERATOR 1 (<100000) " ;O1
321 IF O1>100000 THEN 320
322 O1%=STR$(O1):O1%=LEFT$(O1%,LEN(O1%)-2)
324 IF LEN(O1%)<12 THEN O1%=" "+O1%:GOTO 324
330 CURSOR 5,11:INPUT "OPERATOR 2 (<100000) " ;O2
331 IF O2>100000 THEN 330
332 O2%=STR$(O2):O2%=LEFT$(O2%,LEN(O2%)-2)
334 IF LEN(O2%)<12 THEN O2%=" "+O2%:GOTO 334
336 POKE #75,0
340 FOR Y=15 TO 1 STEP -1:CURSOR 0,Y:PRINT SPC(59):NEXT
400 REM OPERATION
410 REM -----
420 ON OP GOSUB 510,520,530,540,550,560
422 R%=STR$(R):R%=LEFT$(R%,LEN(R%)-2)
424 IF LEN(R%)<12 THEN R%=" "+R%:GOTO 424
430 RESTORE:FOR X=1 TO OP:READ OP$:NEXT:REM GET STRING
440 CURSOR 4,12:PRINT O1;" ";OP$;" ";O2;" " "=" ";R
445 G=O1:GOSUB 2000:G1%=BR$:CURSOR 5,10:PRINT BR$;" ";O1$
446 G=O2:GOSUB 2000:G2%=BR$:CURSOR 5,8:PRINT BR$;" ";O2$
447 FOR X=5 TO 39:CURSOR X,7:PRINT CHR$(11);:NEXT
448 G=R:GOSUB 2000:G3%=BR$:CURSOR 5,6:PRINT BR$;" ";R$
450 POKE #BFEE-#86*20,#CC:CURSOR 5,2:PRINT "TYPE SPACE FOR
NEXT OPERATION"
460 G=GETC:IF G<>32 THEN 460
470 GOTO 100
```



```

510 REM IAND
515 R=01 IAND 02:RETURN
520 REM IOR
525 R=01 IOR 02:RETURN
530 REM MOD
535 R=01 MOD 02:RETURN
540 REM IXOR
545 R=01 IXOR 02:RETURN
550 REM SHL
555 R=01 SHL 02:RETURN
560 REM SHR
565 R=01 SHR 02:RETURN
1000 DATA IAND, IOR, MOD, IXOR, SHL, SHR
2000 REM DEC-BIN CONVERSION
2010 BR$="":M=VARPTR(G)
2020 FOR A=M TO M+3
2030 P=PEEK(A)
2040 FOR X=7 TO 0 STEP -1
2050 B=P IAND (2^(X+1))
2060 C=B SHR X:ST$="o":IF C=1 THEN ST$="1"
2070 BR$=BR$+ST$:NEXT
2080 BR$=BR$+" "
2090 NEXT
2100 RETURN

```

```

IAND
----
35355 IAND 35451 = 35355

00000000 00000000 10001010 00011011
00000000 00000000 10001010 01111011
-----
00000000 00000000 10001010 00011011

```

```

IOR
---
23433 IOR 54653 = 57341

00000000 00000000 01011011 10001001
00000000 00000000 11010101 01111101
-----
00000000 00000000 11011111 11111101

```

```

MOD
---
63456 MOD 23 = 22

00000000 00000000 11110111 11100000
00000000 00000000 00000000 00010111
-----
00000000 00000000 00000000 00010110

```

```

SHL
---
3455 SHL 4 = 55280

00000000 00000000 00001101 01111111
00000000 00000000 00000000 00000100
-----
00000000 00000000 11010111 11110000

```

```

IXOR
----
74123 IXOR 3644 = 77751

00000000 00000001 00100001 10001011
00000000 00000000 00001110 00111100
-----
00000000 00000001 00101111 10110111

```

```

SHR
---
3243 SHR 8 = 12

00000000 00000000 00001100 10101011
00000000 00000000 00000000 00001000
-----
00000000 00000000 00000000 00001100

```

## program identification

title : ELLEN

author : E.SMET

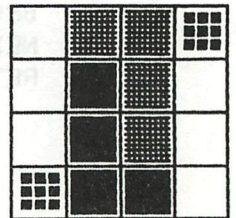
purpose : Position your own "L" and prevent the computer from moving its "L". use cursor-keys & space-bar.

comment : program does POKE in MEMORY instead of using arrays

```

5 MODE 0:PRINT CHR$(12);:COLORT 5 15 5 5:POKE #BA2D,95:CURSOR 0,12
:PRINT " ** ELLEN **"
8 CURSOR 45,4:PRINT " * E. SMET *"
9 IF GETC<>32.0 THEN 9
20 MODE 0:PRINT CHR$(12):CLEAR 5000
25 POKE #74,0:POKE #75,#FF:COLORT 8 0 8 0:GOSUB 3000
30 FOR I=15000 TO 15220:POKE I,0:NEXT I
50 F2=0:RESTORE:FOR X=1 TO 30:READ I:POKE #3A9B+X,I:NEXT X
60 MODE 4A:COLORG 8 14 3 0:GOSUB 30000
61 PRINT CHR$(12);"Moeilijkheidsgraad : 1 = Eenvoudig ":CURSOR 21,2
:PRINT "2 = Moeilijk"
62 FOR X=20 TO 100 STEP 20:DRAW X,20 X,100 0:DRAW 20,X 100,X 0:NEXT
X
63 GOSUB 5000
68 H=GETC-48:IF H<1 OR H>2 THEN 68
69 POKE #3AF9,H:PRINT CHR$(12);"Wenst U te starten (J/N) ?":GOSUB
30000
110 H=GETC:IF H<>74 AND H<>78 THEN 110
120 PRINT CHR$(12):IF H=78.0 THEN 500
240 PRINT CHR$(12);"Ingave L-vorm (cursor=plaats,spatie=ingave)"
241 Z2=0:PRINT "Positie ";:FOR I=1 TO 4
242 CURSOR 10,2:PRINT I:IF I=1.0 THEN GOSUB 7000:GOTO 245
244 GOSUB 7002
245 POKE 15030+I,Z
247 FILL X1-6,Y1-6 X1+10,Y1+10 8:FILL X1-5.5,Y1-5.5 X1+9.5,Y1+9.5 14
248 FOR T=X1-5.5 TO X1+9.5 STEP 5:DRAW T,Y1-5.5 T,Y1+9.5 8:NEXT T
249 FOR T=Y1-5.5 TO Y1+9.5 STEP 5:DRAW X1-5.5,T X1+9.5,T 8:NEXT T
250 NEXT I:X3=0
255 FOR X1=1 TO 4:FOR X2=1.0 TO 4.0:IF PEEK(#3A9E+X1)=PEEK(#3AB6+X2)
THEN X3=X3+1
256 NEXT X2:NEXT X1:IF X3=4 THEN 1220
258 FOR I=#3AFD TO #3B60:POKE I,0:NEXT I
260 FOR X=1.0 TO 4.0:FOR Y=1.0 TO 4.0:IF PEEK(#3AB6+Y)>PEEK(#3AB5+Y)
THEN 280
270 Z=PEEK(#3AB6+Y):POKE #3AB6+Y,PEEK(#3AB5+Y):POKE #3AB5+Y,Z
280 NEXT Y
290 NEXT X
300 FOR X=1.0 TO 4.0:POKE #3ABE+X,PEEK(#3AB6+X):NEXT X:N1=4.0:F1=0.0
:GOSUB 1800:IF M1<>4.0 THEN 1220
370 FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3A9D+X),0:NEXT X
380 FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3AFC+X),1:POKE #3A9E+X,PEEK(#3
AFC+X):NEXT X:POKE #3AA6+PEEK(#3A99),0:POKE #3AA6+PEEK(#3A9A),0
405 POKE #3A99,0:POKE #3A9A,0:GOSUB 5000:Z2=1:PRINT CHR$(12);"Uw inp
ut voor de blokjes ";

```



```

410 FOR S4=1 TO 2
412 GOSUB 7000
414 IF PEEK(#3AA6+Z)<>0.0 THEN 412
415 IF S4=1.0 THEN POKE #3A99,Z:GOSUB 5000:GOTO 440
420 IF Z>PEEK(#3A99) THEN POKE #3A9A,Z:GOSUB 5000:GOTO 440
430 GOTO 412
440 NEXT S4
450 POKE #3AA6+PEEK(#3A99),3:POKE #3AA6+PEEK(#3A9A),3
500 PRINT CHR$(12);"Mijn beurt":FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3A9A+X),0:NEXT X
550 GOSUB 1420:F1=1.0:N1=Z:GOSUB 1800:IF M1=0.0 THEN 1300
560 PRINT "Ik heb";LEFT$(STR$(M1/4.0),2);
570 IF M1>4 THEN PRINT " mogelijkheden ":GOTO 600
580 PRINT " mogelijkheid "
600 X1=0:GOSUB 1370
601 IF PEEK(#3AF9)=1 THEN 650
604 FOR E=0.0 TO M1-4.0 STEP 4.0
608 IF X1<4 THEN 617
610 GOSUB 6000
617 X1=0:FOR F=1.0 TO 4.0:FOR G=1.0 TO 4.0
618 IF PEEK(#3AFC+E+F)=PEEK(#3A9A+G) THEN X1=X1+1
620 IF PEEK(#3AFC+E+F)<>PEEK(#3AA2+G) THEN 640
630 POKE #3ABE+((E/4.0)+1.0),PEEK(#3ABE+((E/4.0)+1.0))+1.0
640 NEXT G:NEXT F:NEXT E
650 GOSUB 1500:Y=(Z-1.0)*4.0:X1=0
700 FOR X=1.0 TO 4.0:POKE #3A9A+X,PEEK(#3AFC+X+Y):POKE #3AA6+PEEK(#3A9A+X),2:NEXT X
750 POKE #3AA6+PEEK(#3A99),0:POKE #3AA6+PEEK(#3A9A),0:POKE #3A99,0:POKE #3A9A,0:GOSUB 5000:PRINT CHR$(12);"Nu zoek ik de blokjes nog ";
760 IF PEEK(#3AF9)=1 THEN 1020
800 FOR I=1.0 TO 4.0:FOR J=1.0 TO 4.0:IF PEEK(#3A9E+I)=PEEK(#3AA2+J) THEN 870
810 NEXT J:NEXT I:GOTO 1020
870 FOR X=1.0 TO 4.0:IF PEEK(#3AA6+PEEK(#3AA2+(X)))>0 THEN 920
880 POKE #3A99,PEEK(#3AA2+X):POKE #3AA6+PEEK(#3A99),3:GOTO 950
920 NEXT X:GOTO 1020
950 FOR X=1.0 TO 4.0:IF PEEK(#3AA6+PEEK(#3AA2+X))>0.0 THEN 1000
970 POKE #3A9A,PEEK(#3AA2+X):POKE #3AA6+PEEK(#3A9A),3:GOTO 1080
1000 NEXT X:GOTO 1050
1020 GOSUB 1610:POKE #3A99,PEEK(#3AFB):POKE #3AA6+PEEK(#3A99),3
1050 GOSUB 1610:POKE #3A9A,PEEK(#3AFB):POKE #3AA6+PEEK(#3A9A),3
1080 FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3A9E+X),1:NEXT X:GOSUB 5000
1090 PRINT:PRINT "En ik controleer de posities "
1100 FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3A9E+X),0:NEXT X:GOSUB 1420:F1=2.0:N1=Z:GOSUB 1800:IF M1=0.0 THEN 1280
1180 FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3A9E+X),1:NEXT X:GOTO 240
1220 PRINT CHR$(12);"Incorrecte ingave ";:GOSUB 5000:GOTO 240
1250 PRINT CHR$(12);"Incorrecte ingave ";:GOSUB 5000:GOTO 405
1280 PRINT CHR$(12);"Ik win ":GOTO 1310
1300 PRINT CHR$(12);"U wint "
1310 PRINT "Voor nog een spel druk spatie";:GOSUB 30000
1320 H=GETC:IF H=0.0 THEN 1320
1330 IF H=32.0 THEN 50
1340 MODE 0:PRINT CHR$(12):COLORT 8 0 15 0:END
1370 FOR X=1.0 TO 16.0:POKE #3ABE+X,0:NEXT X:RETURN
1420 Z=0.0:FOR X=1.0 TO 16.0:IF PEEK(#3AA6+X)>0 THEN 1440
1430 Z=Z+1.0:POKE #3ABE+Z,X
1440 NEXT X:RETURN
1500 Y=0.0:Z=1.0:FOR X=1.0 TO M1/4.0:IF PEEK(#3ABE+X)<Y THEN 1580
1520 IF PEEK(#3ABE+X)>Y THEN 1560
1540 IF RND(1.0)>0.5 THEN 1580
1560 Y=PEEK(#3ABE+X):Z=X
1580 NEXT X:RETURN

```

```

1610 FOR X=1.0 TO 4.0:POKE #3AA6+PEEK(#3A9E+X),0:NEXT X:GOSUB 1420:F1
=2.0:N1=Z:GOSUB 1800:GOSUB 1370
1700 FOR X=1.0 TO M1:POKE #3ABE+PEEK(#3AFC+X),(PEEK(#3ABE+PEEK(#3AFC+
X)))+1.0:NEXT X
1750 FOR X=1.0 TO 4.0:POKE #3ABE+PEEK(#3A9E+X),0:NEXT X:M1=64.0:GOSUB
1500:POKE #3AFB,Z:RETURN
1800 M1=0.0:J=4.0:K=1.0:GOSUB 1880:J=1.0:K=4.0:GOSUB 1880:RETURN
1880 P=0
1890 P=P+1.0:POKE #3ABB,PEEK(#3ABE+P):X=P
1920 X=X+1.0:IF X>N1 THEN 2050
1940 IF PEEK(#3ABE+X)-PEEK(#3ABB)<>J THEN 1920
1950 POKE #3ABC,PEEK(#3ABE+X)
1960 X=X+1.0:IF X>N1 THEN 2050
1980 IF PEEK(#3ABE+X)-PEEK(#3ABC)<>J THEN 1960
1990 POKE #3ABD,PEEK(#3ABE+X)
2000 FOR E=1.0 TO N1:IF ABS(PEEK(#3ABE+E)-PEEK(#3ABB))=K OR ABS(PEEK(
#3ABE+E)-PEEK(#3ABD))=K THEN 2060
2030 NEXT E:GOTO 1890
2050 IF P<N1-2.0 THEN 1890
2055 RETURN
2060 POKE #3ABE,PEEK(#3ABE+E):FOR F=1.0 TO 4.0:IF (PEEK(#3ABA+F))/4.0
<>INT((PEEK(#3ABA+F))/4.0) THEN 2130
2090 FOR G=1.0 TO 4.0:IF PEEK(#3ABA+G)=PEEK(#3ABA+F)+1.0 THEN 2030
2120 NEXT G
2130 NEXT F
2140 FOR Y=1.0 TO 4.0:IF F1=1.0 THEN 2190
2160 IF PEEK(#3ABA+Y)<>PEEK(#3A9E+Y) THEN 2210
2170 NEXT Y:GOTO 2030
2190 IF PEEK(#3ABA+Y)<>PEEK(#3A9A+Y) THEN 2210
2200 GOTO 2170
2210 FOR Y=1.0 TO 4.0:POKE #3AFC+M1+Y,PEEK(#3ABA+Y):NEXT Y:M1=M1+4.0:
GOTO 2030
3000 PRINT CHR$(12);" BBB E L L E N BBB":PRINT "BBBBBBBBBBBBBBBBBBBB":
PRINT :PRINT
3010 PRINT "Strategie en denkvermogen. Dit zijn de factoren waarmee
"
3020 PRINT "in dit spel de winstvoortzetting gevonden wordt."
3030 PRINT "Beide spelers beschikken over een L-vormige figuur. Voor
"
3040 PRINT "de comp.is deze rood, voor U geel. Op het 4x4 bord staan
"
3050 PRINT "buiten deze twee figuren, nog twee blokjes (zwart) welke
"
3060 PRINT "de bewegingsvrijheid belemmeren."
3070 PRINT "De uitdaging van dit spel bestaat erin Uw Å L ü en beide
"
3080 PRINT "blokjes zodanig te plaatsen dat de tegenpartij geen moge-
"
3090 PRINT "lijkheid heeft zijn stukken te plaatsen."
3100 PRINT "Het is hiermee ook duidelijk dat Uw stukken die van de "
3110 PRINT "tegenpartij, nog de geplaatste blokjes mogen overlappen."
3120 PRINT "In het spel kiest U positie dmv de cursor toetsen om deze
"
3130 PRINT "in te geven met de spatie toets."
3134 PRINT "Een kleine tip: Bescherm het centrum ."
3138 PRINT :PRINT "Veel succes."
3140 GOSUB 30000:PRINT :PRINT "Druk spatie om te beginnen."
3150 H=GETC:IF H<>32.0 THEN 3150
3160 RETURN
5000 FOR X=1 TO 16:POKE #3AA6+X,0:NEXT X:POKE #3AA6+PEEK(#3A99),3:POK
E #3AA6+PEEK(#3A9A),3
5010 FOR X=1 TO 4:POKE #3AA6+PEEK(#3A9A+X),2:POKE #3AA6+PEEK(#3A9E+X)
,1:NEXT X
5018 Y1=0:FOR X=1 TO 13 STEP 4:Y1=Y1+20:X1=0
5019 FOR Y=0 TO 3:X1=X1+20:ON PEEK(#3AA6+X+Y)+1.0 GOTO 5020,5030,5040
,5060

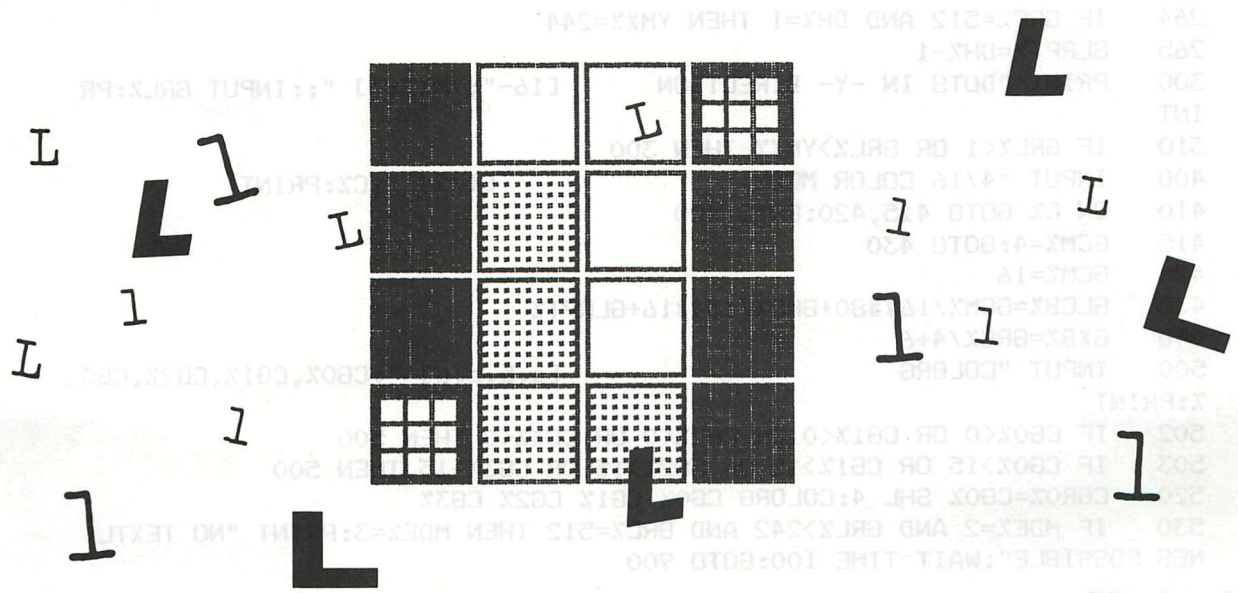
```

ellen

```

5020 C0=8:GOTO 5050
5030 C0=14:GOTO 5050
5040 C0=3
5050 FILL X1+2,Y1+2 X1+18,Y1+18 C0:GOTO 5070
5060 FILL X1+2.5,Y1+2.5 X1+17.5,Y1+17.5 0
5065 FOR Z=X1+2.5 TO X1+17.5 STEP 5:DRAW Z,Y1+2.5 Z,Y1+17.5 8:NEXT Z
5066 FOR Z=Y1+2.5 TO Y1+17.5 STEP 5:DRAW X1+2.5,Z X1+17.5,Z 8:NEXT Z
5070 NEXT Y:NEXT X:RETURN
6000 E=E-4:FOR F=1 TO 4:POKE (#3AFC+E+F),PEEK(#3B00+E+F):NEXT F:RETUR
N
7000 X1=28:Y1=28:Z=1
7002 GOSUB 30000
7005 H=GETC
7010 ON PEEK(#3AA6+Z)+1.0 GOTO 7020,7030,7040,7020
7020 C1=8:GOTO 7060
7030 C1=14:GOTO 7060
7040 C1=3
7060 FILL X1,Y1 X1+3,Y1+3 0:WAIT TIME 4
7070 FILL X1,Y1 X1+3,Y1+3 C1:WAIT TIME 2
7075 IF H=32 THEN 7250
7080 IF H<16 OR H>19 THEN WAIT TIME 3:GOTO 7005
7090 IF (H=16 AND Z<13) OR (H=17 AND Z>4) OR (H=18 AND X1>30) OR (H=1
9 AND X1<80) THEN GOTO 7400
7200 IF H<>32 THEN 7005
7250 IF Z2=1 THEN 7270
7254 IF PEEK(#3AA6+Z)>1.0 THEN 7005
7256 IF I=1.0 THEN RETURN
7258 FOR Z3=I-1 TO 1 STEP -1:IF Z=PEEK(#3AB6+Z3) THEN 7005:NEXT Z3
7260 RETURN
7270 IF PEEK(#3AA6+Z)=2.0 THEN 7005
7300 RETURN
7400 IF PEEK(#3AA6+Z)=3.0 THEN FILL X1,Y1 X1+3,Y1+3 0
7410 ON H-15 GOTO 7420,7430,7440,7450
7420 Y1=Y1+20:Z=Z+4:GOTO 7200
7430 Y1=Y1-20:Z=Z-4:GOTO 7200
7440 X1=X1-20:Z=Z-1:GOTO 7200
7450 X1=X1+20:Z=Z+1:GOTO 7200
30000 H=GETC:H=GETC:H=GETC:RETURN
51000 DATA 1,16,2,3,6,10,7,11,14,15,6,7,10,11,3,2,2,0,0,2,1,0,0,2,1,0,
0,1,1,3
65000 FOR Y=YMAX TO 1 STEP -1
65010 FOR X=0 TO XMAX STEP 2
65020 IF SCRNX(X,Y)=3 THEN DOT X,Y-1 8
65030 NEXT:NEXT

```



## program identification

title : SCREEN DRIVER DEMO  
author : N.P.Looije  
purpose : gives 1001 ways of building up the screen  
comment : results in fascinating screen organisations...

```
10  MODE 0: CLEAR 500: PRINT CHR$(12): COLORT 0 3 3 0: POKE #75,32
15  FOR A%=0 TO 23: POKE #BFEE-A%*#86, #C0+A% MOD 16: NEXT
20  CURSOR 5,18: PRINT "S C R E E N D R I V E R   D E M O N S T R A T
   I O N "
30  CURSOR 12,14: PRINT "THIS PROGRAM SHOWS HOW TO OBTAIN;"
40  CURSOR 10,11: PRINT "64 TEXTMODES WITH VARIABLE LINE SPACING"
50  CURSOR 19,9: PRINT "242.560 GRAPHICMODES"
60  CURSOR 6,7: PRINT "13.899.008 COMBINATIONS OF GRAPHICS AND TEXT"
65  CURSOR 16,5: PRINT "MAXIMUM RESOLUTION 512*244"
70  CURSOR 1,1: PRINT "SDD V4.7
   NPL"
80  IF GETC=0 THEN 80: POKE #75,95: PRINT CHR$(12)
100 INPUT "TEXTMODE/SPLITMODE/GRAPHMODE [1/2/3] "; MDE%: PRINT
110 IF MDE%<1 OR MDE%>3 THEN 100: IF MDE%=1 THEN 600
120 IF MDE%<>2 THEN 150
130 INPUT "TOP/BOTTOM TEXT           [1/2] "; TOP%: PRINT
140 IF TOP%<1 OR TOP%>2 THEN 130
150 INPUT "DOT WIDTH:  0.67/1/2/4   [1/2/3/4] "; DW%: PRINT
151 ON DW% GOTO 170,165,160,155: GOTO 150
155 GRC%=72: SIZE$="4": GOTO 200
160 GRC%=160: SIZE$="2": GOTO 200
165 GRC%=336: SIZE$="1": GOTO 200
170 GRC%=512: SIZE$="0.67"
200 PRINT "DOTS IN -X- DIRECTION   [1-"; GRC%; "]" ": INPUT XMX%: PR
   INT
210 IF XMX%<1 OR XMX%>GRC% THEN 200
250 INPUT "DOT HEIGHT:              [1-16] "; DH%: PRINT
255 IF DH%<1 OR DH%>16 THEN 250
260 YMX%=256/DH%
262 IF DH%>1 THEN YMX%=260/DH%
264 IF GRC%=512 AND DH%=1 THEN YMX%=244
265 GLRPT%=DH%-1
300 PRINT "DOTS IN -Y- DIRECTION   [16-"; YMX%; "]" ": INPUT GRL%: PR
   INT
310 IF GRL%<1 OR GRL%>YMX% THEN 300
400 INPUT "4/16 COLOR MODE           [1/2] "; C%: PRINT
410 ON C% GOTO 415,420: GOTO 400
415 GCM%=4: GOTO 430
420 GCM%=16
430 GLCB%=GCM%/16*#80+GRC%/134*16+GLRPT%
440 GXB%=GRC%/4+6
500 INPUT "COLORG                   [a,b,c,d] "; C60%, C61%, C62%, C63
   %: PRINT
502 IF C60%<0 OR C61%<0 OR C62%<0 OR C63%<0 THEN 500
503 IF C60%>15 OR C61%>15 OR C62%>15 OR C63%>15 THEN 500
520 CGR0%=C60% SHL 4: COLORG C60% C61% C62% C63%
530 IF MDE%=2 AND GRL%>242 AND GRC%=512 THEN MDE%=3: PRINT "NO TEXTLI
   NES POSSIBLE": WAIT TIME 100: GOTO 900
```

```

599 IF MDE%=3 THEN 900
600 LMX%=264
602 IF MDE%=2 THEN LMX%=260
604 IF GRC%=512 THEN LMX%=244:XLNS%=20
605 TMX%=LMX%-GRL%*GLRPT%-GRL%
606 IF TMX%=0 THEN MDE%=3:PRINT "NO TEXTLINES POSSIBLE":WAIT TIME 10
0:GOTO 900
610 NORM%=(TMX%+XLNS%)/11:IF NORM%<1 THEN NORM%=0
620 PRINT NORM%;" NORMAL TEXTLINES POSSIBLE"
630 LEES%=(TMX%+XLNS%)/8:IF LEES%=0 THEN LEES%=1
635 IF LEES%<2 AND GRC%=512 THEN TMX%=0:GOTO 606
640 PRINT LEES%;" READABLE TEXTLINES POSSIBLE"
650 IF LEES%<2 THEN TXL%=2:TLRPT%=0:GOTO 700
660 INPUT "NORMAL,READABLE,VARIABLE TEXTHEIGHT 1/2/3";LZ:PRINT
662 ON LZ GOTO 664,666,668:GOTO 660
664 TMAX%=(TMX%+XLNS%)/11:TLRPT%=10:GOTO 670
666 TMAX%=(TMX%+XLNS%)/8:TLRPT%=7:GOTO 670
668 TMAX%=TMX%:IF GRC%=512 THEN TMAX%=244
670 PRINT "NUMBER OF TEXTLINES      MIN 2 -MAX ";TMAX%;:INPUT TXL%:P
RINT
672 IF TXL%>TMX% OR TXL%<2 THEN 670
680 IF LZ=3 THEN TLRPT%=(TMX%+XLNS%)/TXL%-1:IF TLRPT%>15 THEN TLRPT%
=15
700 INPUT "4/16 COLOR MODE          [1/2] ";CZ:PRINT
710 ON CZ GOTO 712,714:GOTO 700
712 TCM%=4:GOTO 750
714 TCM%=16
750 INPUT "COLORT          [a,b,c,d] ";CT0%,CT1%,CT2%,CT3
%:PRINT
752 IF CT0%<0 OR CT1%<0 OR CT2%<0 OR CT3%<0 THEN 750
754 IF CT0%>15 OR CT1%>15 OR CT2%>15 OR CT3%>15 THEN 750
760 CTX0%=CT0% SHL 4:COLORT CT0% CT1% CT2% CT3%
800 PRINT "THE SCREENDRIVER ONLY SUPPORTS 66 COLUMNS"
810 INPUT "COLUMNS          [66/44/22/11] ";TXC%:PRINT
820 IF TXC%<>66 AND TXC%<>44 AND TXC%<>22 AND TXC%<>11 THEN 810
830 TXLCB%=#40+TCM%/16*#80+TXC%/20*16+TLRPT%:TXB%=2*TXC%+2
900 SCREEN%=#BFFF:SCTOP%=#BFEF
905 LTOT%=GLRPT%*GRL%+GRL%+TLRPT%*TXL%+TXL%
910 EXT%=(260-LTOT%)/23*4
920 SCTOP%=SCTOP%-EXT%:SCREEN%=SCREEN%-EXT%
999 REM *****TEXT*****
1005 IF MDE%<>1 THEN 2000
1010 PRINT CHR$(12):GOSUB 62000
1040 IF TXB%=#86 THEN LIST 1-
1050 FOR A%=CHS% TO CHE%+1 STEP -TXB%
1060 FOR B%=A%-TXB%+5 TO A%-3 STEP 2
1080 POKE B%,#FF*((B%) MOD 4)/2:IF TCM%=16 THEN POKE B%,RND(255)
1090 NEXT:NEXT
1100 IF TXB%=#86 THEN LIST 1-
1990 IF GETC=0 THEN 1990:GOTO 3060
1999 REM *****GRAPH+TEXT*****
2000 IF MDE%<>2 THEN 3000
2020 IF TOP%=1 THEN GOSUB 59000
2030 IF TOP%=2 THEN GOSUB 61000
2040 TRON:GOTO 3040
2999 REM *****GRAPH*****
3000 IF GRC%=512.0 AND GRL%>242.0 THEN 3030
3010 TXL%=2:TXB%=#86:TXLCB%=#70:TCM%=4
3015 FOR A%=#7C TO #7F:POKE A%,PEEK(A#+#22):NEXT
3020 GOSUB 61000:GOTO 3040
3030 GOSUB 60000
3040 FILL 0,0 XMAX,YMAX 21:FILL 1,1 XMAX-1,YMAX-1 20

```

```

3050 MXY=YMAX:MXX=XMAX:F=MXY/MXX:XM2%=MXX/2:YM2%=MXY/2:X2%=XM2%*XM2%
3051 FOR X%=0 TO XM2% STEP 2:Y%=SQR(X2%-X%*X%)
3052 DRAW XM2%-X%,YM2%-Y%*F XM2%+X%,YM2%+Y%/F 21:DRAW XM2%-X%,YM2%+Y%
*F XM2%+X%,YM2%-Y%*F 21
3054 IF GETC<>0 THEN 3060
3055 NEXT:XX1%=XX%/8*8:GRL1%=GRL%/8*8
3056 IF GETC<>0 THEN 3060:A%=RND(XX1%/8*7):B%=RND(GRL1%/8*7)
3057 FILL A%,B% A%+XX%/8,B%+GRL%/8 RND(GCM%)+20*(1-GCM%/16):GOTO 305
6
3060 TROFF:POKE #80,#FF:POKE #81,#BF:POKE #82,#EF:POKE #83,#BF
3070 MODE 0:PRINT CHR$(12):END
9999 GOTO 9999
59000 MODE 1A: IF GCM%=4 THEN MODE 2A
59030 FFB%=SCTOP%-TXL%*TXB%-GRL%*GXB%*-#20
59040 SCRBOT%=FFB%+1
59060 GRE%=SCTOP%-TXL%*TXB%-(GRL%-GAL%)*GXB%*-#10
59070 CHS%=SCTOP%
59080 CHE%=SCTOP%-TXL%*TXB%
59090 SCE%=SCTOP%-TXL%*TXB%-(GRL%-GAL%)*GXB%*-#20
59100 CRSOR%=CHS%*-#8
59110 LNSTR%=CHS%
59120 SCTOP%=SCTOP%-TXL%*TXB%*-#10
59130 GAE%=GRE%
59140 IF TCM%=4 THEN GOSUB 64000
59150 IF TCM%=16 THEN GOSUB 64100
59160 IF GCM%=4 THEN GOSUB 64200:GOSUB 63600
59170 IF GCM%=16 THEN GOSUB 64300:GOSUB 63700
59200 GOTO 61200
59999 REM *****UNSPLIT MODE
60000 MODE 2: IF GCM%=16 THEN MODE 1
60010 FFB%=SCTOP%-GRL%*GXB%*-#10
60020 SCRBOT%=FFB%+1
60040 GRE%=SCTOP%-GRL%*GXB%
60060 GAE%=SCTOP%-GRL%*GXB%
60070 SCE%=SCTOP%-GRL%*GXB%*-#10
60100 IF GCM%=4 THEN GOSUB 63000:GOSUB 63600
60110 IF GCM%=16 THEN GOSUB 63100:GOSUB 63700
60200 POKE #80,SCREEN% MOD 256:POKE #81,SCREEN%/256
60220 POKE #82,SCTOP% MOD 256:POKE #83,SCTOP%/256
60230 POKE #84,FFB% MOD 256:POKE #85,FFB%/256
60250 POKE #88,GRE% MOD 256:POKE #89,GRE%/256
60270 POKE #8C,GAE% MOD 256:POKE #8D,GAE%/256
60280 POKE #8E,SCE% MOD 256:POKE #8F,SCE%/256
60310 POKE #94,XX% MOD 256:POKE #95,XX%/256
60320 POKE #96,GRL%:POKE #98,GXB%
60800 CCB%=#40:D%=0:E%=0: IF GCM%=16 THEN D%=#FF:E%=CG1%+CGR0%:CCB%=CCB
%+CG0%
60820 FOR A%=SCTOP% TO GRE%+1 STEP -GXB%
60830 POKE A%,GLCB%:POKE A%-1,CCB%
60832 FOR B%=A%-GXB%+2 TO A%-2 STEP 2
60834 POKE B%,D%:POKE B%-1,E%
60840 NEXT: NEXT
60900 RETURN
60999 REM *****SPLIT MODE
61000 MODE 1A: IF GCM%=4 THEN MODE 2A
61010 CRSOR%=SCTOP%-(GRL%-GAL%)*GXB%*-#18
61020 LNSTR%=SCTOP%-(GRL%-GAL%)*GXB%*-#10
61030 FFB%=SCTOP%-TXL%*TXB%-GRL%*GXB%*-#20
61040 SCRBOT%=FFB%+1
61060 GRE%=SCTOP%-(GRL%-GAL%)*GXB%
61070 CHS%=SCTOP%-(GRL%-GAL%)*GXB%*-#10
61080 CHE%=SCTOP%-TXL%*TXB%-(GRL%-GAL%)*GXB%*-#10
61090 SCE%=SCTOP%-TXL%*TXB%-(GRL%-GAL%)*GXB%*-#20

```



```

61100 IF GCM%=4 THEN GOSUB 63000
61110 IF TCM%=4 THEN GOSUB 63200:GOSUB 63400
61120 IF GCM%=16 THEN GOSUB 63100
61130 IF TCM%=16 THEN GOSUB 63300:GOSUB 63500
61200 POKE #72,CRSOR% MOD 256:POKE #73,CRSOR%/256
61210 POKE #78,LNSTR% MOD 256:POKE #79,LNSTR%/256
61220 POKE #7A,(LNSTR%+TXB%-6) IAND #FF
61230 POKE #80,SCREEN% MOD 256:POKE #81,SCREEN%/256
61240 POKE #82,SCTOP% MOD 256:POKE #83,SCTOP%/256
61250 POKE #84,FFB% MOD 256:POKE #85,FFB%/256
61270 POKE #88,GRE% MOD 256:POKE #89,GRE%/256
61280 POKE #8A,CHS% MOD 256:POKE #8B,CHS%/256
61290 POKE #8C,CHE% MOD 256:POKE #8D,CHE%/256
61300 POKE #8E,SCE% MOD 256:POKE #8F,SCE%/256
61330 POKE #94,XX% MOD 256:POKE #95,XX%/256
61340 POKE #96,GRL% MOD 256:POKE #98,GXB%
61380 POKE #2A5,SCRBOT% MOD 256:POKE #2A6,SCRBOT%/256
61440 CCB%=#40:D%=0:E%=0:IF GCM%=16 THEN D%=#FF:E%=CGRO%+CG1%:CCB%=CCB
%+CG0%
61450 FOR A%=SCTOP% TO GRE%+1 STEP -GXB%
61460 POKE A%,GLCB%:POKE A%-1,CCB%
61470 FOR B%=A%-GXB%+2 TO A%-2 STEP 2
61480 POKE B%,D%:POKE B%-1,E%
61490 NEXT:NEXT
61600 CCB%=#40:IF TCM%=16 THEN CCB%=CCB%+CTO%
61610 FOR A%=CHS% TO CHE%+1 STEP -TXB%
61620 POKE A%,TXLCB%:POKE A%-1,CCB%
61630 FOR B%=A%-TXB%+2 TO A%-2 STEP 2
61640 POKE B%,#20:POKE B%-1,TCM%/16*RND(255)
61650 NEXT:NEXT
61990 RETURN
61999 REM *****TEXTMODE
62000 MODE 0
62010 FFB%=SCTOP%-TXL%*TXB%+#10
62032 SCRBOT%=FFB%+1
62100 CHE%=SCTOP%-TXL%*TXB%
62120 SCE%=SCTOP%-TXL%*TXB%+#10
62130 CHS%=SCTOP%
62140 CRSOR%=SCTOP%-8
62150 LNSTR%=SCTOP%
62170 POKE #72,CRSOR% MOD 256:POKE #73,CRSOR%/256
62180 POKE #78,LNSTR% MOD 256:POKE #79,LNSTR%/256
62190 POKE #7A,(LNSTR%-TXB%+6) IAND #FF
62200 POKE #80,SCREEN% MOD 256:POKE #81,SCREEN%/256
62210 POKE #82,SCTOP% MOD 256:POKE #83,SCTOP%/256
62220 POKE #84,FFB% MOD 256:POKE #85,FFB%/256
62250 POKE #8A,CHS% MOD 256:POKE #8B,CHS%/256
62260 POKE #8C,CHE% MOD 256:POKE #8D,CHE%/256
62270 POKE #8E,SCE% MOD 256:POKE #8F,SCE%/256
62280 POKE #2A5,SCRBOT% MOD 256:POKE #2A6,SCRBOT%/256
62400 IF TCM%=4 THEN GOSUB 63800:GOSUB 63400
62410 IF TCM%=16 THEN GOSUB 63900:GOSUB 63500
62500 REM TEXT
62505 CCB%=#40:IF TCM%=16 THEN CCB%=CCB%+CTO%
62510 FOR A%=CHS% TO CHE%+1 STEP -TXB%
62520 POKE A%,TXLCB%:POKE A%-1,CCB%
62530 FOR B%=A%-2 TO A%-TXB%+1 STEP -2
62550 POKE B%-1,TCM%/16*RND(256):POKE B%,#20
62560 NEXT:NEXT
62570 RETURN

```

```

63000 REM *****GR 4 KL HEADER
63010 FOR A%=SCTOP%+1 TO #BFFF STEP 4
63020 POKE A%,0:POKE A%+1,0:POKE A%+3,#36
63030 POKE A%+2,PEEK(#A1-(A%-SCTOP%-1)/4 MOD 4)
63040 NEXT:RETURN
63100 REM *****GR 16 KL HEADER
63110 FOR A%=SCTOP%+1 TO #BFFF STEP 4
63120 POKE A%,CGR0%:POKE A%+1,#FF:POKE A%+3,#B6
63130 POKE A%+2,PEEK(#A1-(A%-SCTOP%-1)/4 MOD 4)
63140 NEXT:RETURN
63200 REM *****4 KL INTER
63210 FOR A%=CHS%+1 TO CHS%+16 STEP 4
63220 POKE A%,0:POKE A%+1,0:POKE A%+3,#30
63230 POKE A%+2,PEEK(#7F-(A%-CHS%-1)/4)
63240 NEXT:RETURN
63300 REM *****16 KL INTER
63310 FOR A%=CHS%+1 TO CHS%+16 STEP 4
63320 POKE A%,CTX0%:POKE A%+1,#FF:POKE A%+3,#B0
63330 POKE A%+2,PEEK(#7F-(A%-CHS%-1)/4)
63340 NEXT:RETURN
63400 REM *****4 KL TX+SPLIT TRAILER
63410 FOR A%=CHE% TO CHE%-15-EXT% STEP -4
63420 POKE A%-3,0:POKE A%-2,0:POKE A%,#3F
63430 POKE A%-1,PEEK(#7C+(CHE%-A%)/4 MOD 4)
63440 NEXT:RETURN
63500 REM *****16 KL TX+SPLIT TRAILER
63510 FOR A%=CHE% TO CHE%-15-EXT% STEP -4
63520 POKE A%-3,CTX0%:POKE A%-2,#FF:POKE A%,#BF
63530 POKE A%-1,PEEK(#7C+(CHE%-A%)/4 MOD 4)
63540 NEXT:RETURN
63600 REM *****GR 4 KL TRAILER
63610 FOR A%=GAE% TO GAE%-15-EXT% STEP -4
63620 POKE A%-3,0:POKE A%-2,0:POKE A%,#3F
63630 POKE A%-1,PEEK(#9E+(GAE%-A%)/4 MOD 4)
63640 NEXT:RETURN
63700 REM *****GR 16 KL TRAILER
63710 FOR A%=GAE% TO GAE%-15-EXT% STEP -4
63720 POKE A%-3,CGR0%:POKE A%-2,#FF:POKE A%,#BF
63730 POKE A%-1,PEEK(#9E+(GAE%-A%)/4 MOD 4)
63740 NEXT:RETURN
63800 REM *****TX 4 KL HEADER
63810 FOR A%=SCTOP%+1 TO #BFFF STEP 4
63820 POKE A%,0:POKE A%+1,0:POKE A%+3,#36
63830 POKE A%+2,PEEK(#7F-(A%-SCTOP%-1)/4 MOD 4)
63840 NEXT:RETURN
63900 REM *****TX 16 KL HEADER
63910 FOR A%=SCTOP%+1 TO #BFFF STEP 4
63920 POKE A%,CTX0%:POKE A%+1,#FF:POKE A%+3,#B6
63930 POKE A%+2,PEEK(#7F-(A%-SCTOP%-1)/4 MOD 4)
63940 NEXT:RETURN
64000 REM *****GR 4 KL INTER
64010 FOR A%=SCTOP%+1 TO SCTOP%+16 STEP 4
64020 POKE A%,0:POKE A%+1,0:POKE A%+3,#30
64030 POKE A%+2,PEEK(#A1-(A%-SCTOP%-1)/4)
64040 NEXT:RETURN
64100 REM *****GR 16 KL INTER
64110 FOR A%=SCTOP%+1 TO SCTOP%+16 STEP 4
64120 POKE A%,CGR0%:POKE A%+1,#FF:POKE A%+3,#B0
64130 POKE A%+2,PEEK(#A1-(A%-SCTOP%-1)/4)
64140 NEXT:RETURN

```

```

64200 REM *****SPL TX 4 KL HEADER
64210 FOR A%=CHS%+1 TO #BFFF STEP 4
64220 POKE A%,0:POKE A%+1,0:POKE A%+3,#36
64230 POKE A%+2,PEEK(#7F-(A%-CHS%-1)/4 MOD 4)
64240 NEXT:RETURN
64300 REM *****SPL TX 16 KL HEADER
64310 FOR A%=CHS%+1 TO #BFFF STEP 4
64320 POKE A%,CTX0%:POKE A%+1,#FF:POKE A%+3,#B6
64330 POKE A%+2,PEEK(#7F-(A%-CHS%-1)/4 MOD 4)
64340 NEXT:RETURN

```

```

10 MODE 6: CLEAR 5000: REM POLYGONS F.H. DRUIJFF 1/82
20 A=3+RND(10): IF RND(10)>8 THEN A=A+RND(15)
30 R!=40.0+RND(80.0): DIM B(A),C(A): P!=(PI+PI)/A
40 FOR I=1 TO A: H=R!*COS((I-1)*P!): B(I)=168+H
50 H=R!*SIN((I-1)*P!): C(I)=128+H: NEXT
60 COLORG 0 5 10 RND(15)+1: FOR J=1 TO A: FOR I=J TO A
70 DRAW B(J),C(J) B(I),C(I) 23: NEXT: NEXT
80 WAIT TIME 100: GOTO 10

```

```

10 REM FALLING DOT / F.H. DRUIJFF 1/82
20 MODE 2: COLORG 0 5 14 3: DRAW 0,64 71,64 21
30 FOR I=0 TO 71: DOT I,63 22: DOT I,64 20
40 FOR J=62 TO 0 STEP -1: DOT I,J 22: DOT I,J+1 20: NEXT
50 DOT I,0 23: NEXT: END

```

```

10 REM FILLING SCREEN WITH DOTS / F.H. DRUIJFF 1/82
20 MODE 2: COLORG 0 5 10 15: X=0: Y=0
100 DOT X,Y 21: X=X+1: IF X>71 THEN X=71: GOTO 410
110 Y=Y+1: IF Y<=64 GOTO 100: Y=64: GOTO 200
200 DOT X,Y 21: X=X+1: IF X>71 THEN X=71: GOTO 310
210 Y=Y-1: IF Y>=0 GOTO 200: Y=0: GOTO 100
300 DOT X,Y 21: X=X-1: IF X<0 THEN X=0: GOTO 210
310 Y=Y-1: IF Y>=0 GOTO 300: Y=0: GOTO 400
400 DOT X,Y 21: X=X-1: IF X<0 THEN X=0: GOTO 110
410 Y=Y+1: IF Y<=64 GOTO 400: Y=64: GOTO 300

```

## program identification

**title** : DAI VIDEO SCREEN RAM  
**author** : N.P.Looije  
**purpose** : explains how DAI Screen is build up  
**comment** : The study did result in MODE 7/8 and more ...

HOEK VAN HOLLAND

10 SEPTEMBER 1982

BESTE DAI VRIENDEN,

Hierbij doe ik jullie meerdere programma's toekomen.  
Twee hiervan zijn aanvullingen op het artikel over  
240\*528 resolutie in het DAInamicblad nummer 10, nl. MODE 7  
en MODE 8 met demonstraties en een uitleg over de pointers.  
Na voornoemd artikel te hebben gelezen was ik meteen enthousiast om dit eens uit te proberen. Het programma ingetypt maar helaas wat bleek het scherm werd wel geinitialiseerd maar verder kon ik er niets mee aanvangen.

Ik ben toen eens gaan uitzoeken hoe dat kwam. En zo ook het beeldscherm en de controlewoorden uit gaan zoeken aan de hand van het DAI manual.

Na wat uitzoeken kwam ik tot de volgende opzet;

SCHERMOPBOUW MODE 2A na COLORT 0 1 2 3 COLORG 0 1 2 3

N.B. Alle gegevens zijn vanaf de TOP van het scherm!.

#BFFF

\*SCREEN-----I

I 36 80-00 00 UNITCOLORMODE/COLORG 0 ===== I

I 36 91-00 00 ' ' /COLORG 1 =HEADER= I

I 36 A2-00 00 ' ' /COLORG 2 ===== I

I 36 B3-00 00 ' ' /COLORG 3 I

\*SCTOP-----I

I 03 40-00 I

I totaal 53 [GRL-GAL] lijnen GRAPHICS met #18 [GBX] bytes/lijn I

\*GRE-----I

I 30 80-00 00 UNITCOLORMODE/COLORT 0 I

I 30 91-00 00 ' ' /COLORT 1 ===== I

I 30 A2-00 00 ' ' /COLORT 2 =INTERMEDIATE= I

I 30 B3-00 00 ' ' /COLORT 3 ===== I

\*CHS-----I

I 7A 40-20 00 (66\*[20 00]) I

I totaal 4 [TXL] lijnen TEKST met #86 [TXB] bytes/lijn I

\*CHE-----I

I 3F 80-00 00 UNITCOLORMODE/COLORT 0 I

I 3F 91-00 00 ' ' /COLORT 1 ===== I

I 3F A2-00 00 ' ' /COLORT 2 =TRAILER= I

I 3F B3-00 00 ' ' /COLORT 3 ===== I

\*SCE/GTS-----I

I 03 40-00 I

I totaal 12 [GAL] opgeschoven lijnen (van bovenaan het scherm) \*FFB I

\*GTE-----I

#B8B7

#BFFF-#BFF0 deze lijn bevat de HEADER. Voor mode 2A is dit;  
 #BFFF-#BFFD 36 80 00 00 dit heeft als gevolg:  
 #36 is het MODEBYTE 'HIGH adress'  
 bit 7,6 [00] displaycontrol : 4 colour graphics  
 bit 5,4 [11] resolutioncontrol: 528 dots per lijn  
 bit 3,2,1,0 [0110] linerepeatcount : nog 6\* herhalen  
 #80 is het COLOURTYPEBYTE 'LOW adress'  
 bit 7 [1] maakt een kleurverandering mogelijk  
 bit 6 [0] zet deze lijn in 'unit colour mode'  
 zodat de twee databytes zoveel keer  
 worden herhaald als bepaald door de  
 resolutioncontrolbits.  
 bit 5,4 [00] geven aan dat kleur register 0 wordt  
 bepaald uit 'COLORG' door bits 3-0.  
 Iedere keer als de HIGH en LOW databits  
 0 en 0 zijn wordt de kleur uit bits 3-0  
 gebruikt, dit blijft zo totdat kleur 0  
 anderd wordt in een ander colourtype-  
 byte d.m.v. bit 7 en 5,4.  
 bit 3,2,1,0 [0000] selecteren kleur 0 uit 16 kleuren  
 #00 is de HIGH BYTE 'HIGH adress'  
 bit 7-0 [00000000] selecteert voor alle dots in combinatie  
 met de low byte kleur 0 uit COLORG  
 #00 is de LOW BYTE 'LOW adress'  
 bit 7-0 [00000000] zie HIGH adres  
 deze 2 bytes worden door de 'unit color  
 mode' nog 65\* herhaald.

Op gelijke wijze worden de volgende 12 bytes afgehandeld in  
 de 'unit color mode'. Dit zijn dus totaal 28 lege lijnen.  
 De bits 3-0 van #BFFA, #BFF6 en #BFF2 bevatten respectieve-  
 lijk de kleuren 1,2 en 3 uit de 'COLORG' registers de bits  
 5 en 4 bepalen welk nummer.

Deze 28 lege lijnen komen in deze vorm in alle 4 kleuren-  
 modes voor. In de 16 kleurenmodes gebeurt iets soortgelijks,  
 hier zijn alleen de bits die de 4 kleurenmodedata bevatten  
 vervangen door 16 kleurendata. Dit houdt o.a in dat kleur-  
 veranderingen niet direkt plaatsvinden na COLORG.

De intermediate en trailer (linerepeatcount resp. 0 en #F)  
 werken op dezelfde wijze maar dan met data voor COLORT. Het  
 COLORG werkt op de HEADER.

Het COLORT op de INTERMEDIATE en TRAILER.

De linemodebytes voor de graphics en de tekst bevatten de  
 data voor de displaycontrol, resolutioncontrol, linerepeat-  
 count en de lengte van de lijnen. De colourtypebyte staat  
 neutraal, bit 7=0 (geen kleurverandering) en bit 6=1 (geen  
 'unit colourmode').

Informatie hierover is uitvoerig beschreven in het manual  
 onder PROGRAMMABLE GRAPHICS GENERATOR.

#### MAXIMALE RESOLUTIE - 512\*244 MODE 7 & 8

In het manual kunnen we lezen dat het geheugen voor het  
 beeldscherm maximaal 32K byte kan innemen van #4000-#BFFF.  
 Een rekensommetje leert dan dat bij 528 dots per lijn er  
 #86 (134) bytes per lijn zijn en dus maximaal 32768/134=244  
 lijnen mogelijk zijn. Dus meer dan 240.

De controlewoorden moeten dan zijn;

#### MODEBYTE

bit 7,6 [00] voor 4 kleurengraphics  
 [10] voor 16 kleurengraphics  
 bit 5,4 [11] voor 528 dots per lijn  
 bit 3,2,1,0 [0000] voor 1 lijn dus niet herhalen

## COLORTYPEBYTE

bit 7 [0] bit 5-0 negeren  
bit 6 [1] geen 'unit color mode'  
bit 5-0 [XXXXXX] maakt niet uit

Wij vinden dus voor MODE 7 #B0 #40 en voor MODE 8 #30 #40.  
MODE 7 opstarten via MODE 1,3 of 5, MODE 8 via MODE 2,4 of 6.

Vanaf #BFEF met een stap van #86 worden deze bytes 244\* ingevuld. Hierna moet een trailer (linerepeatcount #F) komen die identiek is aan de header en die dient om de rest van het beeld te vullen met lege lijnen. Als we dit niet doen krijgen we een warboel onder in het beeld.

Als dit gebeurt is is het scherm geïntialiseerd. Maar dan kan je er nog niet mee werken. Dan moeten de pointers voor de graphische functies worden ingevuld bij CURRENT STATE OF SCREEN, CHARACTERMODE en START OF SCREEN.

Na een tijdje experimenteren met het invullen van de diverse pointers bleek dat de DOT, DRAW, FILL en SCRIN() uitstekend bleken te werken. Zelfs de foutmeldingen zoals COLOR NOT AVAILABLE en OFF SCREEN worden gegeven XMAX blijkt (hardwarematig) beperkt tot 511 en YMAX=243.

De nieuwe pointers zijn;

80/81 SCREEN	#BFFF	90/91 GTE	XXXX
82/83 SCTOP	#BFEF	92/93 GAS/GTS	XXXX
84/85 FFB	#4027	94/95 GRC	#0200
86/87 GRR	XXXX	96 GRL	# F4
88/89 GRE	#4037	97 GAL	# XX
8A/8B CHS	#4027	98 GXB	# 86
8C/8D GAE/CHE	#4037		
8E/8F SCE	#4027		

## HOE DE POINTERS TE BEREKENEN

Zonder het ROM te hebben bestudeerd ben ik tot de volgende invulling van de pointers gekomen. Met hulp van de memory-rymap V4.4 heb ik de functies van het scherm getest.

Het kan dus onvolledig zijn maar het werkt voor alle modes!

De plaats van de pointers kunt u zien in de schetsen.

Hieronder vindt u de belangrijke pointers voor het ontwerpen van een eigen mode. Met een korte beschrijving en hoe ze te berekenen. Zij zijn gegeven voor;

- A. Tekstmode [MODE 0]
- B. Graphische mode [MODE 1-8]
- C. Splitmode tekst onderaan [MODE 1A-8A]
- D. Splitmode tekst bovenaan [MODE 1B-8B]

Enige hulpwaarden;

HDRL=headerlengte (normaal #10)

INTL=intermediatelengte (normaal #10)

TRAL=trailerlengte (normaal #10)

TXL =aantal lijnen tekst

TXB =aantal bytes per tekstlijn (#86)

mag ook #5A, #2E of #18 zijn maar de screendriver kan alleen met #86 werken.

## #72/#73 CURSOR

Positie van de cursor op beeldscherm. Bij het zetten van de cursor in een nieuwe text- of splitmode moet men deze op 8 bytes na de linemodebyte komen. In een graphicmode wordt de cursorpositie op 0000 gezet.

\* A.+C.+D. >> CHS-8

\* B. >> 0000

## #78/#79 LNSTR - LiNeStArT

Linemodebyte van de cursorlijn.

\* A.+C.+D. >> CHS

\* B. >> don't care

## #7A LNEND - LiNeEND

Low byte van het eind van de cursorlijn. Gebruikt om het einde van de lijn te controleren.

\* A.+C.+D. >> (CHS+#80) IAND #FF

\* B. >> don't care

#### #80/#81 SCREEN

Top van het scherm, wordt gebruikt voor de cursorpositie en voor memorycheck tijdens modewisseling.

\* A.+B.+C.+D. >> #BFFF

#### #82/#83 SCTOP - SCreenTOP

Deze pointer wijst naar de eerste byte (linemodebyte) van de graphics, direct na de header of intermediate. De naam SCTOP is in dit geval wat ongelukkig gekozen, deze pointer had beter GRS (GRaphicsStart) kunnen heten omdat deze pointer ook naar ergens middenin het schermgeheugen kan wijzen. Wordt gebruikt voor COLORG en tekencommando's.

\*A. >> don't care

\*B.+C. >> #BFFF-HDRL

\*D. >> #BFFF-HDRL-TXL\*TXB-INTL

#### #84/#85 FFB - FirstFreeByte

Deze pointer wijst naar de eerste byte na het beeldscherm.

\*A. >> #BFFF-HDRL-TXL\*TXB-TRAL

\*B. >> #BFFF-HDRL-GRL\*GXB-TRAL

\*C.+D. >> #BFFF-HDRL-TXL\*TXB-INTL-(GRL-GAL)\*GXB-TRAL

#### #86/#87 GRR -GRaphicsRolled

Deze pointer wordt gebruikt in splitmodes. Hij geeft aan vanwaar het scherm is opgeschoven om ruimte te maken voor de tekstregels. Het gedeelte boven GRR tot aan SCTOP is naar net onder het zichtbare scherm verplaatst. Omdat opschuiven niet mogelijk is in een eigen mode zijn deze pointers van weinig waarde.

\*A. >> don't care

\*B.+C.+D.>> #BFFF-HDRL-GAL\*GXB or don't care

#### #88/#89 GRE - GRaphicsEnd

Deze pointer wijst 1 byte na het einde van het voor graphics gebruikte beeldschermgeheugen. Gebruikt in tekenfuncties.

\*A. >> don't care

\*B. >> #BFFF-HDRL-GRL\*GXB

\*C. >> #BFFF-(GRL-GAL)\*GXB

\*D. >> #BFFF-HDRL-TXL\*TXB-INTL-(GRL-GAL)\*GXB

#### 8A/8B CHS - CHaracterStart

Deze pointer wijst naar de eerste byte (linemodebyte) van de tekst. Wordt o.a. gebruikt in PRINT en COLORT commando's.

\*A.+D. >> #BFFF-HDRL

\*B. >> #BFFF-HDRL-GRL\*GXB or don't care

\*C. >> #BFFF-HDRL-(GRL-GAL)\*GXB-INTL

#### #8C/#8D GAE/CHE GraphicsArchiveEnd/CHaracterEnd

In een tekst of splitmode wijst deze pointer 1 byte na het tekstgedeelte. In een unsplitmode wijst deze pointer 1 byte na de trailer. Wordt o.a. voor PRINT gebruikt

A. >> #BFFF-HDRL-TXL\*TXB

B. >> #BFFF-HDRL-GRL\*GXB-INTL

C. >> #BFFF-HDRL-(GRL-GAL)\*GXB-INTL-TXL\*TXB

D. >> #BFFF-HDRL-TXL\*TXB

#### #8E/#8F SCE - SScreenEnd

Deze pointer wijst 1 byte na het einde van het scherm dus dus direkt na de trailer. Wordt o.a. voor COLORG en COLORT gebruikt.

- A. >> #BFFF-HDRL-TXL\*TXB-TRAL
- B. >> #BFFF-HDRL-GRL\*GXB-TRAL
- C.+D. >> #BFFF-HDRL-GRL\*GXB-INTL-TXL\*TXB-TRAL

#### #90/#91 GTE - GraphicsTemporary save areaEnd

In een graphicmode wijst deze pointer naar het einde van het gebied waar het opgeschoven gedeelte graphics zich bevindt.

- A. >> don't care
- B.+C.+D. >> #BFFF-HDRL-(GRL+GAL)\*GXB or don't care

#### #92/#93 GAS/GTS - GraphicsArchiveStart

/GraphicsTemporary save areaStart

In een graphicmode wijst deze pointer naar de start van het gebied waar het opgeschoven gedeelte graphics zich bevindt.

- A. >> don't care
- B. >> #BFFF-HDRL-TXL\*TXB-GRL\*GXB or don't care
- C.+D. >> #BFFF-HDRL-GRL\*GXB

#### #94/#95 GRC - GGraphicColumns

Deze pointer geeft het aantal dots in de X richting aan. De waarde moet minimaal 1 zijn en maximaal zoveel als de linemodebyte bepaald. Bijvoorbeeld voor very high resolution =#86 bytes/lijn is het  $8 * (\#86 / 2 - 3) = \#200 = 512$  voor low resolution #18 bytes/lijn is het  $8 * (\#18 / 2 - 3) = \#48$  72. Gebruikt door tekencommando's.

- A. >> don't care
- B.+C.+D. >> GRC or XMAX+1

#### #96 GRL - GGraphicLines

Deze pointer geeft het aantal dots dus het aantal lijnen Y richting. De waarde moet minimaal 16 en maximaal 256 zijn. Gebruikt door tekencommando's.

- A. >> don't care
- B.+C.+D. >> GRL=YMAX+1

#### #97 GAL - GraphicArchiveLines

Deze pointer geeft het aantal opgeschoven of op te schuiven lijnen aan. Bij het zelf ontwerpen van een mode kan men de waarde 0 gebruiken.

- A.+B.+C.+D. >> don't care

#### #98 GXB GraphicXBytes

Het aantal bytes per lijn in de gebruikte graphische of splitmode. Gebruikt in tekencommando's.

- A. >> don't care
- B.+C.+D. >> Als bepaald door linemodebytes graphics  
resp. #86, #5A, #2E, #18 voor MODE 7/8, 5/6, 3/4, 2/1.

#### #99/#9A GREQ - GraphicsEndOld

Vorig einde of graphics. Behoeft men niet in te vullen bij het ontwerpen van een mode.

- A.+B.+C.+D. >> don't care



#9B/#9C CHSO - CHAracterStartOld  
 Vorig einde characters. Behoeft men niet in te vullen bij  
 het ontwerpen van een mode.  
 A.+B.+C.+D.>> don't care

#2A5/2A6 SCR BOT - SCREenBOTtom  
 Hier staat de eerste byte van het schermgeheugen, wordt o.a.  
 gebruikt voor testen op OUT OF SPACE FOR MODE.  
 A. >> #BFFF-HDRL-TXL\*TXB-TRAL+1  
 B. >> #BFFF-HDRL-GRL\*GXB-TRAL+1  
 C.+D. >> #BFFF-HDRL-GRL\*GXB-INTL-TXL\*TXB-TRAL+1

Indeling van de verschillende modes;  
 Met de plaats van de pointers.

SCREEN-----I	SCREEN-----I
I     HEADER     I	I     HEADER     I
CHS-----I	SCTOP-----I
I	I
I     TEKST     I	I     GRAPHICS    I
I	I
CHE-----I	GRE-----I
I     TRAILER    FFB I	I     TRAILER    FFB I
SCE-----I	SCE-----I

SCREEN-----I	SCREEN-----I
I     HEADER     I	I     HEADER     I
CHS-----I	SCTOP-----I
I	I
I     TEKST     I	I     GRAPHICS    I
I	I
CHE-----I	GRE-----I
I     INTERMEDIATE I	I     INTERMEDIATE I
SCTOP-----I	CHS-----I
I	I
I     GRAPHICS    I	I     TEKST     I
I	I
GRE-----I	CHE-----I
I     TRAILER    FFB I	I     TRAILER    FFB I
SCE-----I	SCE-----I

Dan volgen hier de waardes voor de HEADER, INTERMEDIATE  
 en TRAILER na een COLORT w x y z en of COLORG w x y z.  
 De HEADER en de INTERMEDIATE bevatten de kleuren van het  
 gedeelte waar zij boven staan (graphics COLORG/tekst COLORT)  
 De trailer bevat de kleuren van waar hij onderstaat

HEADER           4 KLEUREN  
 36 8w 00 00 36 9x 00 00 36 Ay 00 00 36 Bz 00 00  
 INTERMEDIATE   4 KLEUREN  
 30 8w 00 00 30 9x 00 00 30 Ay 00 00 30 Bz 00 00  
 TRAILER           4 KLEUREN  
 3F 8w 00 00 3F 9x 00 00 3F Ay 00 00 3F Bz 00 00  
 HEADER           16 KLEUREN  
 B6 8w w0 FF B6 9x w0 FF B6 Ay w0 FF B6 Bz w0 FF  
 INTERMEDIATE   16 KLEUREN  
 B0 8w w0 FF B0 9x w0 FF B0 Ay w0 FF B0 Bz w0 FF  
 TRAILER           16 KLEUREN  
 BF 8w w0 FF BF 9x w0 FF BF Ay w0 FF BF Bz w0 FF

OPMERKINGEN

Bij een charactermode worden door LIST,?CHR\$(12) en UT de  
 tekstlijnen teruggebracht naar de normale lijnafstand.

In 16 kleurentekst wordt door deze *commando's* ook het scherm weer teruggezet naar 4 kleuren, bij de scroll van het beeld wordt de onderste lijn in 4 kleurenmode afgedrukt. Bij onderste tekstlijn wordt altijd een normale regelafstand geforceerd.

Bij het experimenteren met de MODES 8A en 7A bleek dat deze niet ingesteld worden bij een BREAK omdat de screendriver deze modes niet kent. Dit zal dan door een kunstgreep moeten gebeuren nl. door 240 lijnen graphics en 4 lijnen tekst te nemen met een linerepeatcount van 0. Bij een BREAK forceert de screendriver dan 1 lijn met een linerepeatcount van #A. Door ?CHR\$(12) wordt dan de linerepeatcount van alle tekstlijnen op #A gezet.

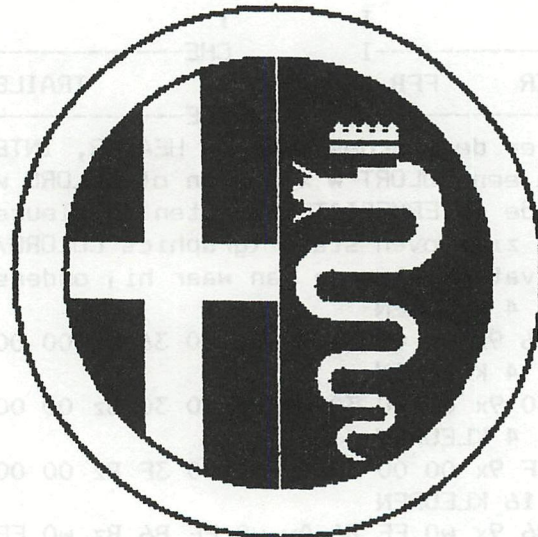
Bij de volle 244 lijnen graphics is dus raadzaam in een programma het volgende op te nemen bijv. 65335 IF GETC=0 THEN 65335:MODE 0

De dots in MODE 7/8 zijn niet vierkant [ $Y=ca.1.5*X$ ] in de praktijk blijkt deze verhouding bijv. bij het trekken van een cirkel goed te voldoen.

Het nadeel van een iets lagere verticale resolutie wordt ruim gecompenseerd door de hogere horizontale resolutie bijv. in grafieken.

Met vriendelijke groeten,  
N.P. Looije,  
Paludanushof 22,  
3151 CM Hoek van Holland.  
Telefoon 01747-2920

A  
L  
F  
A



R  
O  
M  
E  
O

SCREEN MEMORY MANAGEMENT  
SCREEN CONSTANTS GRAPHIC MODES  
=====

RAM	POINTER	MODE 0	MODE 1/2	MODE 1A/2A	MODE 3/4	MODE 3A/4A	MODE 5/6	MODE 5A/6A
0080/81	1st byte screen RAM	0	0	0	0	0	0	0
0082/83	Points after header	10	10	10	10	10	10	10
0084/85	1st free RAM byte	C80	638	860	177C	19A4	5A20	5C48
0086/87	Top rolled area	0	130	130	460	460	F88	F88
0088/89	End graphics area	0	628	508	176C	131C	5A10	4A98
008A/8B	Start character area	10	628	518	176C	132C	5A10	4AAB
008C/8D	End character area End archive area	CA0	860	730	19A4	1534	5C48	4CC0
008E/8F	End screen	C80	638	740	177C	1554	5A20	4CD0
0090/91	End area used changing mode	0	748	748	18BC	18BC	6988	6988
0092/93	Start archive area Start temp.save area	0	740	628	1554	176C	4CD0	5A10
0094/95	Number of hor. blobs	0	48	48	A0	A0	150	150
0096	Number of graph. lines	0	41	41	82	82	0	0
0097	Number of saved graphic lines	0	0C	0C	18	18	2C	2C
0098	Number bytes/line	0	18	18	2E	2E	5A	5A

All constants given are offset of top of screen address (#BFFF for a 48K machine)

DAI FIRMWARE  
=====

Do you have problems concerning certain matters which you believe are caused by DAI monitor ?

Did you find some useful application of routines available in the firmware ?

Maybe they are worth to be published. Please write or phone me.

Most of the articles I published are based on your problems or questions. In any case, you will get an answer as soon as possible.

Jan Boerrigter  
Fabritiusstraat 15  
NL-6174 RG Sweikhuizen  
tel. 04493/2093

## SCREEN MEMORY MANAGEMENT

=====

This article describes the method used to organise the screen memory by the DAI-firmware.

The handbook gives a lot of information about the several screen modes and the way they are fitted into the screen memory. The way of changing from one mode to another is not described very clear.

In an 'all-graphic' mode, the screen consists of:

- A1. Header.
- A2. Screen colour data.
- A3. Trailer.

The header and trailer areas are blanking lines at the beginning and the end of the screen.

In 'split-graphics' modes, the screen organisation is as follows:

- B1. Header.
- B2. Picture area corresponding to the bottom part of A2.
- B3. Middle.
- B4. Character area.
- B5. Trailer.
- B6. Archive area, containing the upper part of A2, which is not displayed in B2.

When changing from split-mode to all-graphic-mode and visa-versa, the archive area is temporarily moved to the temporary save area, which starts at A3.

When changing from all-graphics to split:

- Top of screen into temporary save area.
- Bottom part graphics shifted upwards.
- Top part graphics from temporary save area into archive area.
- Set up character screen.

When changing from split to all-graphics:

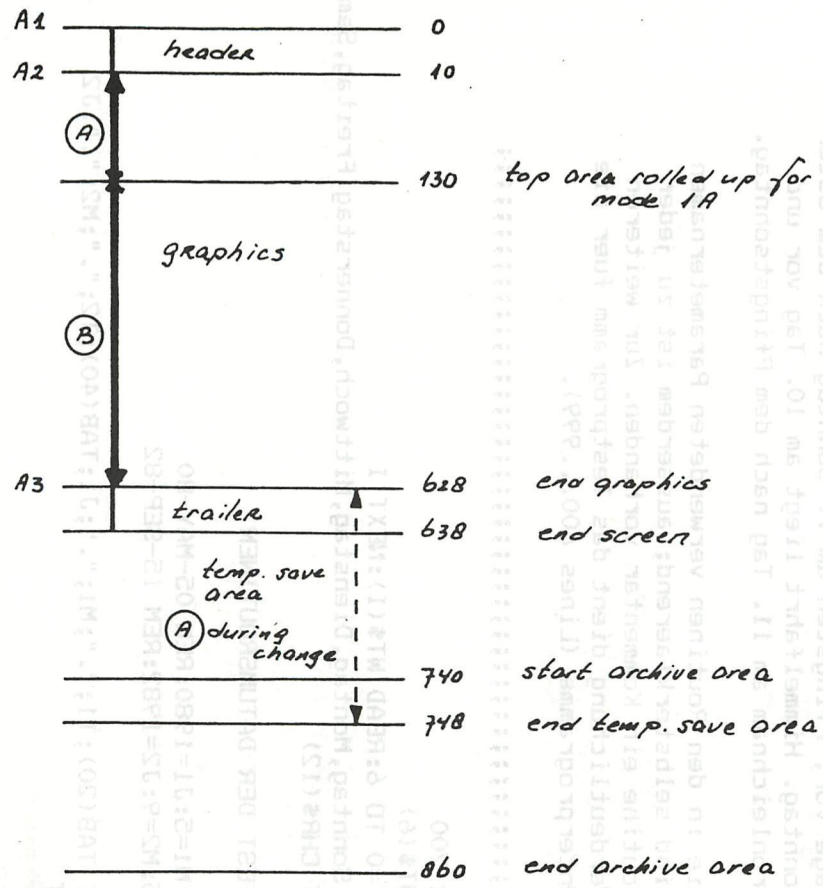
- Archive area into temporary save area.
- Bottom part graphics shifted downwards.
- Temporary save area into top of screen.

The annexed drawing shows the screen memory organisation in a diagram.

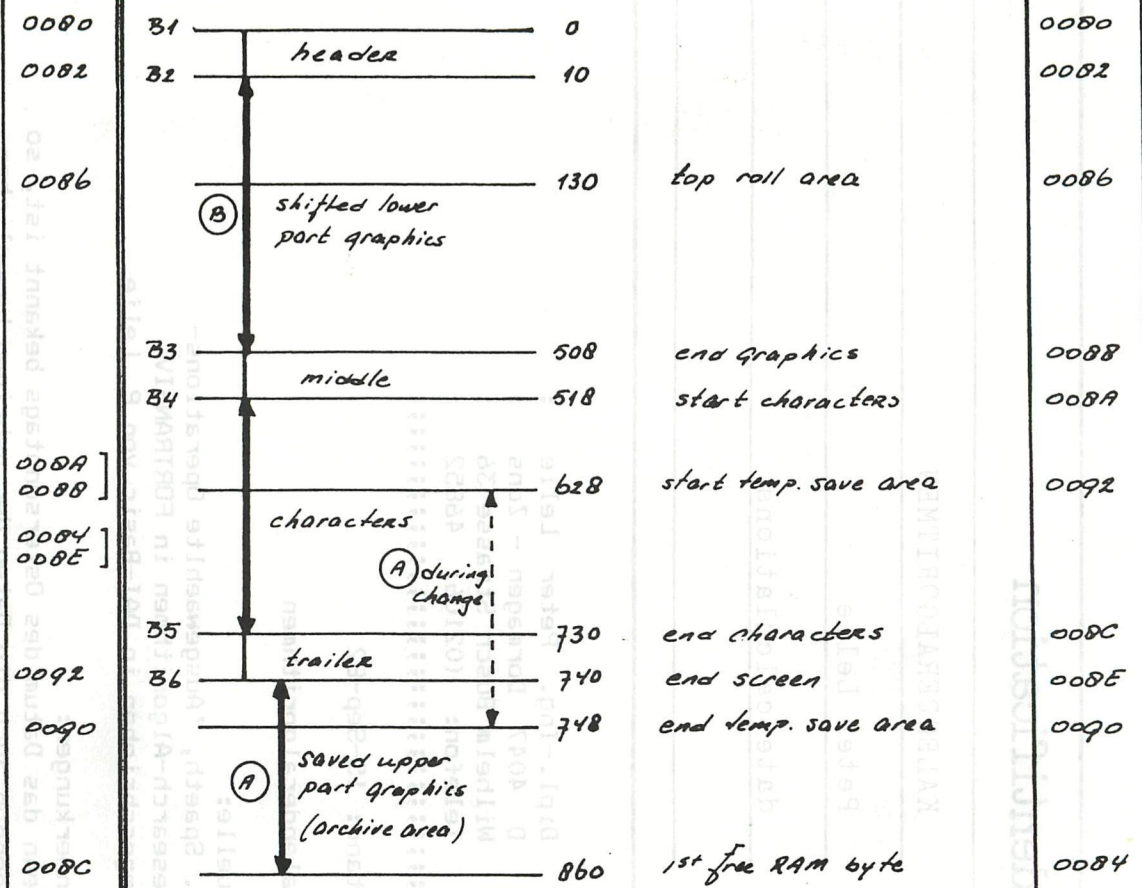
The numbers at the right side of the lines are offsets from the address of the top of the screen (#BFFF for 48K). These values are stored in RAM pointers (addresses in separate columns) during the set-up of a particular screen mode.

For each mode, these constants are retrieved from ROM. The annexed table gives all the constants as they can be found in ROM-bank 2 on the addresses 2E030 - 2E0C2. Vectors to these constant tables are located on addresses 2E59A - 2E5A5.

(C) Jan Boerrigter - August, 1982



MODE 1



MODE 1A

SCREEN RAM ORGANISATION in all-graphics mode and in split mode (and during mode change)

## program identification

title       :    KALENDERALGORITMEN  
author      :    Peter Lelie  
purpose     :    date-calculations  
comment     :    

```
10  REM ;   Dipl.-Ing. Peter Lelie ;
11  REM ;   D 4047 Dormagen - Zons ;
12  REM ;   Wilhelm Busch Strasse 36 ;
13  REM ;   Telefon: (02106) 46852 ;
14  REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
15  REM
16  REM Stand: 15-Sep-82
17  REM
20  REM Kalenderalgorithmen
21  REM
22  REM Quelle:
23  REM H. Spaeth, 'Ausgewaehlte Operations-
24  REM Research-Algorithmen in FORTRAN-IV'
25  REM Umgeschrieben in DAI-Basic von P. Lelie
26  REM
27  REM Anmerkungen:
28  REM Wenn das Datum des Ostersonntags bekannt ist, so
29  REM koennen auch die Daten der uebrigen beweglichen
30  REM Feiertage berechnet werden: Fastnacht liegt 48
31  REM Tage vor, Pfingsten am 7. Sonntag nach dem Oster-
32  REM sonntag. Himmelfahrt liegt am 10. Tag vor und
33  REM Fronleichnam am 11. Tag nach dem Pfingstsonntag.
34  REM
35  REM Die in den Routinen verwendeten Parameternamen
36  REM sind selbsterklaerend; ausserdem ist zu jeder
37  REM Routine ein Kommentar vorhanden. Zur weiteren
38  REM Verdeutlichung dient das Testprogramm fuer die
39  REM Unterprogramme (Lines 600...999).
40  REM
50  REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
60  REM
510 CLEAR 200
520 DIM WT$(6)
530 FOR I=0 TO 6:READ WT$(I):NEXT I
540 DATA Sonntag, Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag
550 PRINT CHR$(12)
590 REM
600 REM TEST DER DATUMSROUTINEN
602 REM
610 T1=5:M1=5:J1=1980:REM 05-MAY-80
620 T2=15:M2=9:J2=1982:REM 15-SEP-82
622 REM
630 PRINT TAB(20);T1;". ";M1;". ";J1;TAB(40);T2;". ";M2;". ";J2
635 PRINT
```

```

640 TAG=T1:MONAT=M1:JAHR=J1:GOSUB 1500:W1=WT
642 GOSUB 1800:JT1=JT
644 GOSUB 1000:JD1=JD
650 TAG=T2:MONAT=M2:JAHR=J2:GOSUB 1500:W2=WT
652 GOSUB 1800:JT2=JT
654 GOSUB 1000:JD2=JD
660 PRINT "WOCHENTAG";TAB(21);WT$(W1);TAB(41);WT$(W2)
665 PRINT
670 PRINT "JAHRESTAG";TAB(20);JT1;TAB(40);JT2
680 PRINT
690 PRINT "JULIANTAG";TAB(20);JD1;TAB(40);JD2
700 PRINT
705 JJ=JD2-JD1
710 PRINT "DIFFERENZ";TAB(40);JJ
720 PRINT
730 JD=JD1+JJ/2
740 GOSUB 1600
750 PRINT "DATUM DES TAGES";TAB(20);JD;" :";TAB(40);TAG;" .";MONAT;" .";JAHR
760 PRINT
770 GOSUB 1800
780 PRINT "DESSEN NUMMER IM JAHR";JAHR;" :";TAB(40);JT
790 PRINT
800 GOSUB 1200
810 PRINT "OSTERN IM JAHR ";JAHR;" :";TAB(40);TAG;" .";MONAT;" .";
820 PRINT :PRINT :PRINT
999 GOTO 999
1000 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1001 REM SUBROUTINE JD
1002 REM
1010 REM EINEM (SINNVOLLEN) DATUM DES GREGORIANISCHEN
1020 REM KALENDERS WIRD EINDEUTIG EINE NATUERLICHE ZAHL,
1030 REM DIE DER TAGNUMMER IM JULIANSCHEN KALENDER ENT-
1040 REM SPRICHT, DERART ZUGEORNET, DASS DIE DIFFERENZ
1050 REM DER ZWEI DATEN ZUGEORNETEN ZAHLEN DIE ANZAHL
1060 REM DER ZWISCHEN IHNEN LIEGENDEN TAGE ERGIBT.
1070 REM
1080 REM ENTER WITH: TAG, MONAT, JAHR = DATUM
1090 REM RETURN WITH: JD = JULIANTAGNUMMER
1095 REM
1100 J=(MONAT-14)/12
1110 L=JAHR+J+4800
1120 JD=TAG-32075+1461*L/4+367*(MONAT-2-12*J)/12-3*((L+100)/100)/4
1130 RETURN
1200 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1201 REM SUBROUTINE OSTERN
1202 REM
1210 REM FUER EIN VORGEGEBENES JAHR WERDEN MONAT UND
1220 REM TAG DES OSTERSONNTAGS BERECHNET.
1230 REM
1240 REM ENTER WITH: JAHR = JAHR FUER OSTERN
1250 REM RETURN WITH: MONAT, TAG = OSTERSONNTAG-DATUM IN JAHR
1252 REM
1260 GN=(JAHR MOD 19)+1
1270 IF JAHR<=1582 THEN 1350
1280 C=JAHR/100+1
1290 GC=(3*C)/4-12
1300 CC=(C-16-(C-18)/25)/3
1310 ED=(5*JAHR)/4-GC-10
1320 E=((11*GN+19+CC-GC) MOD 30)+1
1330 IF (E=25 AND GN>11) OR E=24 THEN E=E+1
1340 GOTO 1370

```

```

1350 ED=(5*JAHR)/4
1360 E=((11*GN-4) MOD 30)+1
1370 TAG=44-E
1380 IF TAG<21 THEN TAG=TAG+30
1390 TAG=TAG+7-((ED+TAG) MOD 7)
1400 IF TAG<=31 THEN 1440
1410 MONAT=4
1420 TAG=TAG-31
1430 GOTO 1450
1440 MONAT=3
1450 RETURN
1500 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1501 REM SUBROUTINE WT
1502 REM
1510 REM ZU EINEM VORGEgebenEN DATUM WIRD DER
1520 REM ZUGEHÖRIGE WOCHENTAG BERECHNET:
1530 REM WT=0:SONNTAG ... WT=6:SAMSTAG
1540 REM
1550 REM ENTER WITH: TAG, MONAT, JAHR = DATUM
1560 REM RETURN WITH: WT = NR. DES WOCHENTAGS
1565 REM
1570 L=MONAT+10
1580 M=(MONAT-14)/12+JAHR
1590 WT=((13*(L-(L/13)*12)-1)/5+TAG+77+5*(M-(M/100)*100)/4+M/400-(M/100)*2) MOD 7
1595 RETURN
1600 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1601 REM SUBROUTINE DJ
1602 REM
1610 REM EINER NATUERLICHEN ZAHL JD, DIE DER TAGNUMMER
1620 REM IM JULIANSCHEN KALENDER ENTSPRICH, WIRD EIN
1630 REM DATUM IM GREGORIANSCHEN KALENDER ZUGEOEDNET.
1640 REM
1650 REM ENTER WITH: JD = TAGNUMMER
1660 REM RETURN WITH: TAG, MONAT, JAHR = KALENDERDATUM
1670 REM
1670 L=JD+68569
1670 N=4*L/146097
1670 L=L-(146097*N+3)/4
1670 JAHR=4000*(L+1)/1461001
1670 L=L-1461*JAHR/4+31
1670 MONAT=80*L/2447
1670 TAG=L-2447*MONAT/80
1670 L=MONAT/11
1670 MONAT=MONAT+2-12*L
1670 JAHR=100*(N-49)+L+JAHR
1670 RETURN
1800 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1801 REM SUBROUTINE JT
1810 REM
1820 REM FUER EIN VORGEgebenES DATUM WIRD DIE NUMMER
1830 REM DES TAGS IM ENTSPRECHENDEN JAHR BERECHNET
1840 REM
1850 REM ENTER WITH: TAG, MONAT, JAHR = DATUM
1860 REM RETURN WITH: JT = TAGNUMMER
1870 REM
1870 L=(MONAT+10)/13
1870 JT=TAG+3055*(MONAT+2)/100-2*L-91
1870 JT=JT+(1-(JAHR-(JAHR/4)*4+3)/4+(JAHR-(JAHR/100)*100+99)/
1870 100-(JAHR-(JAHR/400)*400+399)/400)*L

```



```

1930 RETURN
2000 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2001 REM SUBROUTINE DT
2010 REM
2020 REM FUER EIN VORGEGEBENES JAHR UND DIE NUMMER JT
2030 REM EINES TAGES IN DIESEM JAHR WERDEN DIE RESTLICHEN
2040 REM BESTANDTEILE DES ZUGEHORIGEN DATUMS, ALSO
2050 REM MONAT UND TAG, BERECHNET (1900<JAHR<2100).
2060 REM
2070 REM ENTER WITH: JAHR, JT = JAHR UND JAHRESTAG
2080 REM RETURN WITH: MONAT, TAG = RESTLICHES DATUM IN JAHR
2090 REM
2100 L=0
2110 IF (JAHR/4)*4=JAHR THEN L=1
2120 K=0
2130 IF JT>59+L THEN K=2-L
2140 J=JT+K+91
2150 MONAT=(J*100)/3055
2160 TAG=J-(MONAT*3055)/100
2170 MONAT=MONAT-2
2180 RETURN
2199 REM ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

## 512x244

MODE 7 :

```

0300 F5 C5 D5 E5 3E 08 EF 18 2A 9E 00 3E 0F A5 87 87
0310 87 87 6F 7C E6 0F 85 0E FF 21 EF BF 1E F4 16 42
0320 36 B0 2B 36 40 2B 71 2B 77 15 C2 25 03 2B 1D C2
0330 1E 03 21 6B 03 11 84 00 06 15 7E 12 13 23 05 C2
0340 3A 03 21 27 40 22 A5 02 06 10 21 F0 BF 11 28 40
0350 7E 12 13 23 05 C2 50 03 3E BF 32 2B 40 32 2F 40
0360 32 33 40 32 37 40 E1 D1 C1 F1 C9 27 40 77 B0 37
0370 40 27 40 37 40 27 40 77 56 EF 56 00 02 F4 2C 86

```

MODE 8 :

```

0300 F5 C5 D5 E5 3E 0A EF 18 21 EF BF 1E F4 16 84 36
0310 30 2B 36 40 2B 36 00 15 C2 14 03 2B 1D C2 0D 03
0320 21 59 03 11 84 00 06 15 7E 12 13 23 05 C2 28 03
0330 21 27 40 22 A5 02 06 10 21 F0 BF 11 28 40 7E 12
0340 13 23 05 C2 3E 03 3E 3F 32 2B 40 32 2F 40 32 33
0350 40 32 37 40 E1 D1 C1 F1 C9 27 40 77 B0 37 40 27
0360 40 37 40 27 40 77 56 EF 56 00 02 F4 2C 86 B0 37

```

These machine language routines will offer you MODE 7 or MODE 8 after CALLMH300. The screen will be initiated and the screen pointers will be adapted. From that moment on, BASIC will be able to use DOT, DRAW, FILL, SCRN, XMAX, YMAX... in the ULTRA-HIGH resolution. The normal SPLIT-mode is not possible, so if you want to return to TEXT-mode, use : MODE 0, if BASIC will try to go to split-mode this will result in garbage on your screen.

with many thanks to N.P.LOOIJE

# program identification

title : FASCINATION  
 author : contribution DAInamic France  
 purpose : generates graphic animations  
 comment :

```

17000 MODE 0:POKE #75,32:PRINT CHR$(12):CLEAR 1000:DIM C$(15),T(64):PRINT " FASCINATION...":PRINT :PRINT
17001 PRINT " -Retour a ce menu:----- M."
17002 PRINT " -Retour au programme principal:- P."
17003 PRINT " -Acceleration:----- ";CHR$(136);"."
17004 PRINT " -Ralentissement:----- ";CHR$(137);"."
17005 PRINT " -Marche Avant:----- ";CHR$(94);"."
17006 PRINT " -Marche Arriere:----- ";CHR$(140);"."
17008 FOR TZ=1 TO 64:T(TZ)=13.1:NEXT
17009 T(2)=T(2)+3.8:T(8)=T(8)+4.5:T(16)=T(16):T(64)=T(64)
17010 C$(0)="Noir":C$(1)="Bleu Roi":C$(2)="Rose":C$(3)="Rouge":C$(4)="Kaki":C$(5)="Vert":C$(6)="Chair"
17011 C$(7)="Mauve":C$(8)="Gris":C$(9)="Azur":C$(10)="Orange":C$(11)="Rose Saumon"
17012 C$(12)="Bleu Ciel":C$(13)="Vert Clair":C$(14)="Jaune":C$(15)="Blanc"
17015 FOR TZ=0 TO 15:COURSOR 33+12*INT(TZ/8.0),12-(TZ MOD 8):PRINT TZ;"=";C$(TZ):NEXT
17020 POKE #75,95:COURSOR 0,12:INPUT " Couleur fascinante no 1";BZ:BZ=ABS(INT(BZ)):IF BZ>15 THEN 17000
17021 PRINT :INPUT " Couleur fascinante no 2";NZ:NZ=ABS(INT(NZ)):IF NZ>15 THEN 17000
17022 PRINT :INPUT " Couleur fascinante no 3";MZ:MZ=ABS(INT(MZ)):IF MZ>15 THEN 17000
17024 COURSOR 0,8:PRINT SPC(25):COURSOR 0,8:INPUT " Nombre de spirales ";LIZ:Z=LIZ
17025 IF Z=1 THEN 17030
17026 Z=Z/2:IF FRAC(Z)=0 THEN 17025
17027 GOTO 17024
17030 PRINT :PRINT :INPUT " Densite fascinante (1 a 10)";DEZ
17031 IF DEZ<1 OR DEZ>10 THEN 17000
17032 KL=T(LIZ)/DEZ
17033 MODE 6:IY=5:VZ=0:COLORG VZ BZ NZ MZ
17035 LIX=LIZ*(2+1.25*INT(LIZ/64)+1.5*INT(LIZ/32)+0.5*INT(LIZ/16)+INT(LIZ/8)-INT(LIZ*0.25)):LIYZ=4*LIZ/LIX:KL=2*KL/LIX
17040 FOR TZ=1 TO LIYZ STEP 2:DZ=TZ*YMAX/LIYZ
17045 FOR YZ=1 TO LIX STEP 2:CCZ=YZ*XMAX/LIX:RZ=0
17047 GOSUB 17055:NEXT:NEXT:GOTO 17070
17055 FOR X=0.0 TO KL STEP 0.1:WZ=BZ:BZ=NZ:NZ=MZ:MZ=WZ:SZ=RZ*SIN(X):CZ=RZ*COS(X)
17058 DRAW SZ+CCZ,CZ+DZ CZ+CCZ,-SZ+DZ WZ:DRAW CZ+CCZ,DZ-SZ CCZ-SZ,-CZ+DZ WZ
17060 DRAW -SZ+CCZ,DZ-CZ CCZ-CZ,SZ+DZ WZ:DRAW -CZ+CCZ,SZ+DZ SZ+CCZ,CZ+DZ WZ
17065 RZ=RZ+DEZ:NEXT:RETURN
17070 WAIT TIME IZ:AZ=GETC:IF AZ=0 THEN ON FZ GOTO 17098,17099
17080 IF AZ=16 THEN FZ=1:GOTO 17090
17082 IF AZ=17 THEN FZ=2:GOTO 17090
17084 IF AZ=18 THEN IZ=IZ-1:IF IZ<=0 THEN IZ=0:GOTO 17070
17086 IF AZ=19 THEN IZ=IZ+1:GOTO 17070
17088 IF AZ=ASC("P") THEN RETURN
17090 IF AZ=ASC("M") THEN 17000
17095 GOTO 17070
17098 COLORG VZ BZ NZ MZ:ZL=MZ:MZ=BZ:BZ=NZ:NZ=ZL:GOTO 17070
17099 COLORG VZ BZ NZ MZ:ZL=MZ:MZ=NZ:NZ=BZ:BZ=ZL:GOTO 17070
  
```

mikrobel

6 nov 82

Aarschot

# **INFORMATIEVE COMPUTERBEURS**

**IN DE STADSFEESTZAAL**

**TE AARSCHOT**

**VAN 10 TOT 19 UUR**

**met :**

**ACORN - ATARI - BBC - CIRIS - DAI - ITT 2020 - APPLE II  
ITT 3030 - PHILIPS P2000 - SCHOOL & COMPUTER  
DAInamic - DIDACOM - DIDAI SOFT - PERCOSOFT  
UITGEVERIJ VAN IN - UITGEVERIJ KLUWER  
IMAGE VIDEO - INDATA - COMPAC - CONFIRMEX**

**VRIJE TOEGANG**

**RUIME PARKING**

een organisatie van DAI namic vzw



## SUPER NOISE GENERATOR (SNG)

\*\*\* THIS IS A MUST FOR EVERY GAME-FREAK !!!!!

**now**

Soon available from DAInamic Belgium, a sound extension for your DAI.

By adding a simple PC-board (dimensions 9 x 5 CM), containing 4 IC's, 17 resistors, 9 condensators, and connecting 12 wires into the DAI, you can make your DAI more attractive.

With " SNG " is it possible to create new noise effects, going from a realistic explosion to a starting rocket, and a lot more of unexpected effects. You can use this SNG to make your games and animations more attractive.

All these new noise commands are controlled by the normally not used NOISE X 1 to NOISE X 7 volume selects.

The new commands are;

- NOISE X 0 : SNG reset, noise off
- NOISE X 1 : continuous, low noise
- NOISE X 2 : continuous, variable noise (sweep)
- NOISE X 3 : continuous, high noise
- NOISE X 4 : high, short decay noise
- NOISE X 5 : high, long decay noise
- NOISE X 6 : low, short decay noise
- NOISE X 7 : low, long decay noise
- NOISE X 8 to NOISE X 15 : normal DAI noise

It is also possible now, with a simple POKE, to switch on and off the sound from your cassetterecorder, ( music, explanations, talk, and so on ), and play this back through your TV set.

NOTE:

FOR ALL THESE MODIFICATIONS THERE ARE NO CHANGES NECESSARY ON THE DAI PC-BOARD, NOR SPECIAL SOFTWARE REQUIRED !!!!!

COPYRIGHT R. CORSWANDT / W. DE WINTER 1982



price : 2200 Bfr (complete kit)

## HOW TO EXTEND THE DAI-BASIC

=====

Several times I have been asked if it is possible to add some useful commands to the BASIC inside the DAI-ROM's. Well, to start with, it is not possible simply to add commands to the existing ones. Then the ROM contents must be changed on several locations, including shifting of program parts to other areas. So let us forget about this (very nice) idea, and let's see if another possibility exists.

First of all, we have to deal with two problems:

1. Where the additional software (machine language routines) can be placed.
2. What kind of routines are required.

Let us start with the memory problem.

1. The socket for the math.chip is not available for other purposes because it is exclusively wired for the math.chip AMD9511.
2. An additional ROM can be placed on a small printed circuit board on the X-bus. Then the address area F700-F7FF can be used without big problems. But this solution has already been used for the Memocom digital recorder.
3. The address range F900-FAFF is not used. Theoretically, it could be possible to use it for additional ROM on the X-bus (evt. combined with the Memocom ROM).  
The address-decoder IC45 (see DAI-schematics sheet 5) has outputs for decoded chip select signals F9 and FA. These outputs are not used and not wired, but could be used with some additional wiring. The X-bus has 2 not-used pins, which could be available for this purpose.
4. Last possibility: Locate the machine language routines in RAM.

Now the software part of the BASIC extension.

1. Let us follow the method Memocom used for the MDCR. New BASIC commands are written in a REM-statement in the BASIC program. Before this REM-statement, a CALLM command is written, calling a routine which checks the REM-statement for a known (new) BASIC command. If found, this command is executed by its appropriate execution routine.
2. So we need:
  - a. A routine, which tests if the text in a REM-statement is a (new) BASIC command.
  - b. A list in which all new commands are written.
  - c. Execution routines for this new commands.

How will it look like ?

In the BASIC, we will find for instance:

```
10 CALLM (TESTREM): REM COUNT
```

'TESTREM' is the address of the m.l.routine for testing the REM-statement; COUNT is one of our new BASIC commands.

'TESTREM' will perform:

Test if the next command on the BASIC textline is 'REM' (token A9; registerpair BC points to it). If not, the routine can be aborted.

If it is a REM-statement, the string in this statement must be

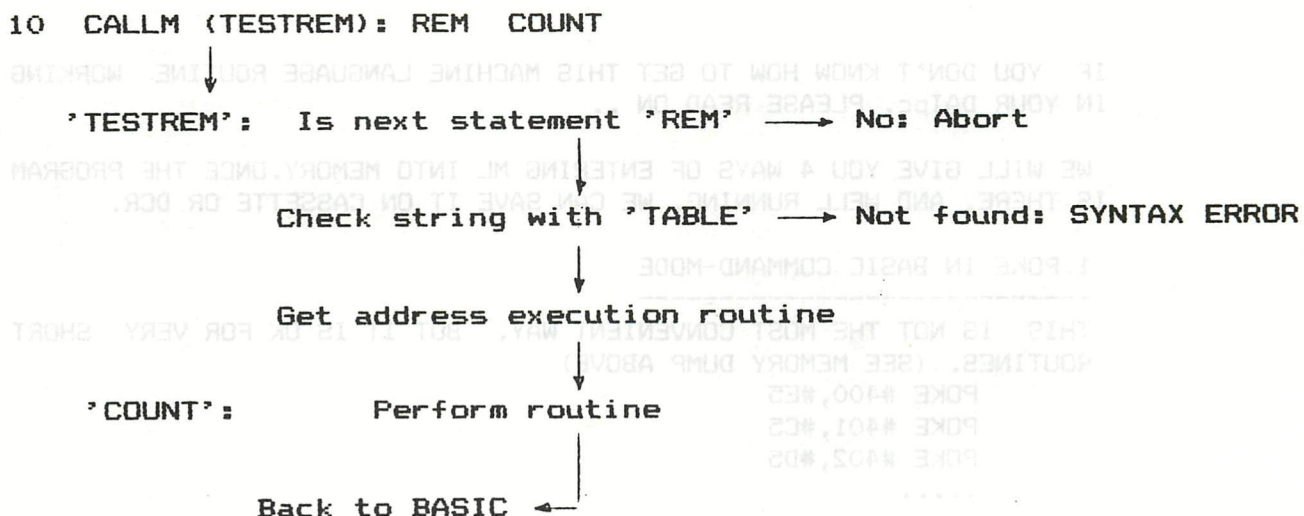
checked. In the textline, direct after the token, follows the length of the string and then the string itself. The string must be compared with a list of the new commands. If found, the execution routine belonging to this command must be executed. If not found, a syntax error can be displayed (via a jump to address DA0B).

So we need a table with the new BASIC instructions. The format of the table is as follows:

TABLE:	<length>	<string>	<address>
e.g. for 'COUNT':	05	length	
	43	C	
	4F	O	
	55	U	
	4E	N	
	54	T	
	XXXX	Addr	execution routine
	00	End	of table

The 'end of table' byte is required to let the routine know that all strings have been checked, and the command is not found.

Again everything in a flow diagram:



As you see, this is just some philosophy about 'how to do it'. Lack of time made it impossible to write a sound program. This is up to you, if you urgently need extension of the DAI-BASIC. And don't forget to publish the results !!

(C) Jan Boerrigter - July 1982

## program identification

title : AUDIO CASSETTE MARKER  
author : W.De Winter & W.Hermans  
purpose : To indicate the end of files on audio tapes.  
(recorder should be in record position)  
comment : start with CALLM H400, exit by hard reset.

```
0400 E5 C5 D5 F5 F3 3A 20 00 32 06 FD CD 11 04 C3 0B
0410 04 3A 4B 04 2F E6 3F 4F 3E 21 32 06 FD CD 43 04
0420 3E 20 32 06 FD CD 43 04 0D C2 18 04 3A 4B 04 3D
0430 C2 35 04 3E 14 32 4B 04 C9 0E FF CD 43 04 0D C2
0440 3B 04 C9 3A 4B 04 3D C2 46 04 C9 14 C2 44 04 C9
```

IF YOU DON'T KNOW HOW TO GET THIS MACHINE LANGUAGE ROUTINE WORKING IN YOUR DA1pc, PLEASE READ ON ..

WE WILL GIVE YOU 4 WAYS OF ENTERING ML INTO MEMORY.ONCE THE PROGRAM IS THERE, AND WELL RUNNING, WE CAN SAVE IT ON CASSETTE OR DCR.

### 1. POKE IN BASIC COMMAND-MODE

THIS IS NOT THE MOST CONVENIENT WAY, BUT IT IS OK FOR VERY SHORT ROUTINES. (SEE MEMORY DUMP ABOVE)

```
POKE #400,#E5
POKE #401,#C5
POKE #402,#D5
.....
.....
POKE #44F,#C9
```

### 2. POKE IN BASIC PROGRAM-MODE

```
10 POKE #400,#E5
20 POKE #401,#C5
.....
500 POKE #44F,#C9
```

THIS IS ALREADY BETTER, WE CAN SAVE OUR BASIC PROGRAM ON CASSETTE BEFORE EXECUTING THE ML-ROUTINE.

NOTE : DON'T FORGET TO ADAPT THE HEAP-POINTER :  
POKE #29C,#5: CLEAR 1000



### 3. POKE IN BASIC PROGRAM-MODE FROM DATA-LINES

THIS IS A VERY CONVENIENT WAY IF THE BLOCK OF ML IS CONTINUOUS , AS IN OUR ROUTINE.

```
10 REM IMPLIED INTEGER !
20 FOR X=#400 TO #44F
30 READ V:POKE X,V
40 NEXT
100 DATA #E5,#C5,#D5,#F5,#F3,.....
140 DATA .....#44,#04,#C9
```

TAKE THE SAME PRECAUTIONS : SAVE THE BASIC PROGRAM ON TAPE BEFORE 'RUN', AND ADAPT HEAP POINTER.

### 4. ENTER THE ML IN UTILITY

```
TYPE : UT <RETURN>
TYPE : S400 <SPACE> (THE OLD CONTENTS OF #400
ARE DISPLAYED)
TYPE : E5
TYPE : <SPACE> (#401)
TYPE : C5
ETC. UNTIL 44F = C9
```

CHECK THE CONTENTS WITH DISPLAY:  
D400 44F <RETURN>

SAVE ON TAPE BEFORE GO OR CALLM#400:  
W400 44F 400 44F AUDIO CASSETTE MARKER CALLM#400

!-----!  
THIS IS THE TITLE ON CASSETTE/DCR

```
10 GOTO 100:REM LINES / F.H. DRUIJFF 1/82
20 IF P=0 THEN Q=Q-D:IF Q<0 THEN P=P-Q:Q=0:T=T+1:IF T=4 GOTO 110:RETURN
30 IF P=335 THEN Q=Q+D:IF Q>255 THEN P=P-Q+255:Q=255:RETURN
40 IF Q=0 THEN P=P+D:IF P>335 THEN Q=P-335:P=335:RETURN
50 IF Q=255 THEN P=P-D:IF P<0 THEN Q=Q+P:P=0
60 RETURN
70 P=A:Q=B:D=6:GOSUB 20:A=P:B=Q
80 P=X:Q=Y:D=5:GOSUB 20:X=P:Y=Q
90 DRAW A,B X,Y 21:GOTO 70
100 MODE 6:COLORG 0 1 3 14:A=0:B=255:X=15:Y=255:GOTO 70
110 IF GETC=0 GOTO 110:END
```

## program identification

title : EFFICIENT CIRCLE-DRAWING  
author : F.Druijff  
purpose : circle-routines for different applications  
comment :

>>>>>>> PROGRAMMEER - TECHNIEKEN <<<<<<<<

In het nu voor U liggende artikel wil ik nader ingaan op het tekenen van cirkels. Bij veel programma's zijn cirkels of ellipsen nodig en het tekenen hiervan levert soms nogal wat problemen op. Er zijn echter vele wegen die naar Rome leiden.

Vantevoren wil ik nu reeds stellen dat het niet zo is dat er een beste methode bestaat. De ene methode programmeert snel en werkt traag de andere is lastig te programmeren maar is snel.

En dan zijn er nog varianten die veel of weinig geheugen vragen.

Maar laten we beginnen; de cirkel die we wensen te tekenen zal steeds een cirkel zijn die het middelpunt heeft in het beeldmidden en een straal van 100.

De meest voor de hand liggende manier om deze cirkel te tekenen:

```
10 MODE 6
20 FOR I=0.0 TO 2.0*PI STEP PI/180.0
30 DOT XMAX/2+100*SIN(I),YMAX/2+100*COS(I) 22:NEXT
```

We weten dat er 180 graden in PI radialen zitten dus nemen we als stapgrootte  $\pi/180$ , precies een graad. Als we echter RUN geven blijkt dat onze cirkel gaten heeft. We kunnen natuurlijk de stapgrootte verkleinen maar dit gaat ten koste van de snelheid dus moeten we het wel verantwoord verkleinen.

We bereiden, als de cirkel een straal heeft van honderd dan is de omtrek  $200\pi$ . En dat is ruim 628, dus zal de stapgrootte zo moeten zijn dat elk van deze punten aan bod komt (en niet meer dan eens). Hieruit volgt stapgrootte is  $2\pi/200\pi$  dus  $\pi/315$  ofwel 0.01 dus nog eenvoudiger  $1/\text{straal}$ .

Eventueel om bij ongelukkige afrondingen goed te zitten :  $0.9/\text{straal}$ .

Er zitten nog wel meer punten ter verbetering in het programma.

Als eerste noem ik het feit dat werken met een variabele sneller gaat dan met een constante, dus moet :

```
10 MODE 6:FOR I=0.0 TO 2.0*PI STEP PI/315.0
20 R=100.0:DOT XMAX/2+R*SIN(I),YMAX/2+R*COS(I) 22:NEXT
```

sneller werken, maar ik maak elke keer R gelijk aan honderd en zo gaat mijn toch al minieme winst verloren. Wel essentieel is het feit dat ik de DAI ruim 600 maal XMAX en YMAX laat opzoeken en die dan vervolgens elke keer door twee deel. We krijgen na deze verbetering het volgende programma:

```
10 MODE 6:MX=XMAX/2.0:MY=YMAX/2.0:R=100.0
20 FOR I=0 TO 2.0*PI STEP 1.0/R:DOT MX+R*SIN(I),MY+R*COS(I) 22:NEXT
```

Ons valt nu op dat er nu  $XMAX/2.0$  staat in plaats van  $XMAX/2$  zoals hiervoor. Dit komt omdat de DOT instructie nu eenmaal integers verwacht en dus constanten als deze 2 gewoon 2 laat en er niet 2.0 van maakt. Bij de verwerking echter wordt de  $XMAX/2$  wel omgezet in floating point om bij de  $R*\text{SIN}(I)$  geteld te kunnen worden; het resultaat wordt dan weer integer gemaakt om "gedot" te kunnen worden. Het ligt voor de hand om het programma nu te gaan versnellen door alles in integer te zetten. De eerste poging daartoe mislukt echter, omdat  $\text{SIN}(I)$  en  $\text{COS}(I)$  nu vrijwel altijd nul zijn. Daarbij zal ook de loopvariabele  $I$  steeds nul blijven. Eerst even nadenken en komen we op het volgende programma:

```
IMPINT
10  MODE 6:MX=XMAX/2:MY=YMAX/2:R!=100.0
20  FOR I!=0.0 TO PI+PI STEP 1.0/R!:X=R!*SIN(I!):Y=R!*COS(I!)
30  DOT MX+X,MY+Y 22:NEXT
```

En dit werkt inderdaad sneller. Is het U opgevallen? Juist  $PI+PI$  is sneller dan  $2.0*PI$  maar dat zet hier geen zoden aan de dijk, het combineren van regel 20 en 30 zal wel iets uitmaken.

Maar ons programma blijft toch traag, logisch we laten 628 keer een sinus en een cosinus uitrekenen. We bedenken dat linker en rechterdeel elkaars spiegelbeeld zijn en komen zo tot:

```
10  MODE 6:MX=XMAX/2:MY=YMAX/2:R!=100.0
20  FOR I!=0.0 TO PI STEP 1E-2:X=R!*SIN(I!):Y=COS(I!)
30  DOT MX+X,MY+Y 22:DOT MX-X,MY+Y 22:NEXT
```

Dit blijkt goed en vlot te werken en dan willen we vanzelfsprekend verder versnellen door andere symmetrieën ook te gebruiken.

We gaan in regel 20 maar tot  $PI/2.0$  en regel 30 wordt

```
30  DOT MX+X,MY+Y 22:DOT MX+X,MY-Y 22, DOT MX-X,MY+Y 22:DOT MX-X,MY-Y 22:NEXT
```

We worden nu pas goed enthousiast en gebruiken ook diagonalen als symmetrieën. We krijgen:

```
10  MODE 6:MX=XMAX/2:MY=YMAX/2:R!=100.0
20  FOR I!=0.0 TO PI/4.0 STEP 1/R!:X=R!*SIN(I!):Y=R!*COS(I!)
30  DOT MX+X,MY+Y 22:DOT MX+X,MY-Y 22:DOT MX-X,MY+Y 22:DOT MX-X,MY-Y 22
40  DOT MX+Y,MY+X 22:DOT MX+Y,MY-X 22:DOT MX-Y,MY+X 22:DOT MX-Y,MY-X 22:NEXT
```

En alhoewel dit programma redelijk snel werkt zal het programmeren ervan velen (terecht vaak) weerhouden om het zo te doen.

We gaan eens nadenken of we die trage sinus en cosinus wel nodig hebben.

We kunnen vaststellen dat inderdaad een van de twee omzeild kan worden maar dat dit vermoedelijk toch weer niet simpel is te programmeren.

We slaan een nieuwe weg in. De stelling van Pythagoras is de basis voor een nieuwe aanpak. Redenerend vanuit het middelpunt is een punt op de cirkel met coördinaten  $x$  en  $y$  zodanig gelegen dat  $x^2+y^2$  gelijk moet zijn aan  $r^2$ . En aangezien we bij elke  $x$ -waarde twee  $y$ -waarden nodig hebben beginnen we met

```
10  MODE 6:XM=XMAX/2:YM=YMAX/2:R=100
20  FOR X=-R TO R:Y=SQR(R*R-X*X)
30  DOT MX+X,MY+Y 22:DOT MX+X,MY-Y 22:NEXT
```

Dit blijkt echter niet correct te werken. Links en rechts zijn vele punten overgeslagen, dit is logisch bij deze x-waarden zijn meerdere y-waarden. Als we echter niet alleen gebruik maken van een boven/onder-symmetrie maar ook van een links/rechts symmetrie? Nee, we kunnen dat wel doen maar dan zullen nog steeds links en rechts punten ontbreken. We zien dat horizontaal alles juist getekend is, dus als we de cirkel voor de helft op bovenstaande manier tekenen en de andere helft verkrijgen door verwisselen van x en y coördinaten zijn we er. Regel 10 blijft gelijk

```
20 FOR I=-R/2 TO R/2:J=SQR(R*R-I*I)
30 DOT MX+I,MY+J 22:DOT MX+I,MY-J 22
40 DOT MX+J,MY+I 22:DOT MX-J,MY+I 22:NEXT
```

Maar ook dit werkt niet bevredigend. We denken weer na: halve cirkel betekent niet van  $-R/2$  tot  $R/2$  maar van  $-45$  tot  $+45$  graden om het zo te zeggen. Dit betekent dat we de grens moeten leggen bij  $+45$  of  $-45$  en niet  $50$ . Deze  $45$  wordt gevonden (voor andere cirkels nodig) door de straal van de cirkel te delen door wortel van 2. Maken we nu ook gebruik van symmetrie komen we tot:

```
10 MODE 6:MX=XMAX/2:MY=YMAX/2:R=100
20 FOR I=0 TO R/SQR(2):J=SQR(R*R-I*I)
30 DOT MX+I,MY+J 22:DOT MX+I,MY-J 22:DOT MX-I,MY+J 22:DOT MX-I,MY-J 22
40 DOT MX+J,MY+I 22:DOT MX+J,MY-I 22:DOT MX-J,MY+J 22:DOT MX-J,MY-I 22:NEXT
```

Dit programma werkt snel vooral op de machines zonder math-chip duidelijk een verbetering maar ook weer lastig te programmeren. Zeker voor kleine cirkels zijn vorige methodes net zo goed zo niet beter. Viel U trouwens ook op dat ik  $R*R$  doe en niet  $R^2$ ? Het is in ieder geval sneller en werkt tenminste ook bij negatieve getallen.

We denken weer eens wat verder na over ons probleem. Als we nu meerdere cirkels moeten tekenen zal het waarschijnlijk de moeite lonen de sinus en cosinuswaarden op te slaan in een array en die dan telkens weer te gebruiken. Voordelen zijn evident: sneller de bedoelde waarde beschikbaar eventueel al vermenigvuldigd met de straal en omgezet in integer. Zijn de stralen verschillend kan er bij iets kleinere straal toch nog wel wat tijdswinst geboekt worden, is de straal minder dan de helft kunnen we ook arrayelementen overslaan en dan nog grotere winst boeken.

De nadelen zijn jammer genoeg ook aanwezig: we zijn een aanzienlijk stuk geheugen kwijt (waarvan meer dan de helft nutteloos gevuld met nullen) en we moeten wachten totdat de tabel klaar is. Toch maar eens proberen.

```
10 MODE 6A: CLEAR 8000: DIM X(3,159), Y(3,159)
20 MX=XMAX/2:MY=YMAX/2:R!=100.0
30 FOR A!=0.0 TO PI+PI STEP PI/316.0
40 X=R!*SIN(A!):X=MX+X:Y=R!*COS(A!):Y=Y+MY
50 X(P,Q)=X:Y(P,Q)=Y:Q=Q+1
60 IF Q=159 THEN Q=0:P=P+1:PRINT P
70 NEXT:MODE 6
80 FOR P=0 TO 3:FOR Q=0 TO 158:DOT X(P,Q),Y(P,Q) 22:NEXT:NEXT
```

Opmerkingen bij dit programma: de PRINT P is overbodig en dient alleen als indicator, we gebruiken hier een twee dimensionale array voor X en een voor Y dit is beslist sneller dan een drie dimensionale array. Zorgt er trouwens altijd voor om de indices van een array altijd integer te laten zijn; opzoektijden worden tot viermaal zo lang als U floating point gebruikt.

Het ligt voor de hand om de verbeteringen van zoeven nu ook op dit programma los te laten. D.w.z. slechts een kwart van de gegevens uitrekenen en de punten vinden door weer spiegelingen toe te passen. Het onderstaande programma doet dit en werkt ook de +/- nog eens slim af. bestudeert U het maar eens en controleer de werking.

```

10  MODE 6A: CLEAR 8000: DIM X(159), Y(159)
20  MX=XMAX/2: MY=YMAX/2: R!=100.0
30  FOR A!=0.0 TO PI/2.0 STEP PI/316.0
40  X(Q)=R!*SIN(A!): Y(Q)=R!*COS(A!): Q=Q+1: NEXT
50  MODE 6
60  FOR Q=0 TO 159: FOR M=-1 TO 1 STEP 2: FOR N=-1 TO 1 STEP 2
70  DOT MX+M*X(Q), MY+N*Y(Q) 22: NEXT: NEXT: NEXT

```

Als volgende stap kunnen we natuurlijk het opbouwen van de array mbv de methode Pythagoras doen maar dat laat ik aan de lezer over.

We kunnen nu nog denken aan het niet gebruiken van de sinus en cosinus maar een benadering daarvan. Zie hiervoor nummer 10 het artikel van T. Berkx, maar bedenk wel dat deze methode alleen sneller is voor machines zonder math.chip. Dit is de reden dat ik het niet uitgewerkt heb.

De volgende mogelijkheid heb ik ook niet uitgewerkt maar zal zeker voor kleine cirkels goed bruikbaar zijn: Geef de benodigde waarden op in een aantal DATA's of lees ze in vanuit een array. Dit laatste zal alleen voor MDCR en floppy bezitters interessant zijn.

Nu tot besluit wil ik de cirkel wel eens echt snel tekenen.

We zullen dit doen door niet meer puntje voor puntje de cirkel te tekenen maar steeds kleine lijntjes te tekenen. De punten die verbonden moeten worden kunnen we op een van de hiervoor besproken methodes bepalen.

```

10  MODE 6: MX=XMAX/2: MY=YMAX/2: R!=100.0: XO=MX: YO=MY+100
20  FOR I!=0.0 TO PI+PI STEP PI/15.0: X=R!*SIN(I!): Y=R!*COS(I!)
30  DRAW XO, YO MX+X, MY+Y 22: XO=MX+X: YO=MY+Y: NEXT

```

Vindt U dat de cirkel te hoekig wordt dan dient U de STEP aan te passen.

Het kan nodig zijn iets meer dan rond te gaan omdat anders het laatste lijntje niet meer getrokken wordt.

Deze methode kan dan weer gecombineerd worden met het uit een array lezen van de gegevens en zo bereiken we een snelwerkende maar lastig te programmeren methode.

We krijgen nu het volgende programma:

```

10  MODE 6: CLEAR 2000: MX=XMAX/2: MY=YMAX/2: R!=100.0: DIM X(30), Y(30)
20  FOR A!=0.0 TO 2.06 STEP PI/15.0
30  X=R!*SIN(A!): X(P)=MX+X: Y=R!*COS(A!): Y(P)=MY+Y: P=P+1: NEXT
40  FOR I=0 TO 29: DRAW X(I), Y(I) X(I+1), Y(I+1) 22: NEXT

```

Zelfs aan dit programma zijn nog piepkleine zaken bij te schaven: twee variabelen X en Y kunnen door een vervangen worden. In plaats van nieuwe variabele I kunnen we P gebruiken, het komt echter leesbaarheid niet ten goede en maakt niet uit voor snelheid alleen voordeel voor geheugen (8 bytes). Het combineren van regel 20 en 30 zal wel iets uitmaken in snelheid.

We hebben het al de tijd gehad over cirkels met straal 100, kleinere cirkels kunnen soms beter op een van de eerder besproken methodes behandeld worden. Vervolgens wil ik nog opmerken dat als we de hele cirkelschijf willen hebben we niet een stralenbundel vanuit het middelpunt naar de rand moeten tekenen want dan krijgen we door afrondingen zeker bij grote cirkels punten die niet meegenomen worden. Remedie hiertegen is vanzelfsprekend het verkleinen van de stapgrootte (bestaft met langere werktijd) of en dat is veel beter door lijnen van rand tot rand te trekken.

Deze lijntrekkerij moet dan wel horizontaal geschieden en niet vertikaal. Dit vanwege de beeldopbouw bij de DAI. Ik laat het aan de lezer over dit te programmeren.

Frank H. Druijff

```

*LIST
1  REM * DESSIN - DRAWING - 16 COLORS *
2  REM * you can change the messages; they are all in line 60 *
3  REM * vous pouvez conserver une image de la maniere suivante : *
4  REM * you can save the picture with following procedure : *
5  REM * 1) BREAK, 2) UT+RETURN, 3)W63B8 SPACE BFFF SPACE NAME OF FILE *
6  REM * pour relire - for reloading : 1) MODE 5A 2) UT+RETURN 3) R+RETURN *
7  REM * move without drawing : select same color as the background *
8  REM * deplacement sans trace : selectionner la couleur du fond *
9  REM * to reveal your position - pour connaitre votre position : "S" *
10 COLORT 15 0 0 0:MODE 5A:COLORG 0 15 0 0
20 PRINT CHR$(12);:FOR I%=0.0 TO 9.0:FILL I%*16+11,0 I%*16+20,9 I%
30 PRINT I%;" ";:NEXT:PRINT " ";
40 FOR I%=10 TO 15:FILL I%*16+11,0 I%*16+20,9 I%
50 PRINT CHR$(I%+55);" ";:NEXT
60 PRINT :PRINT "TRES GROS = T, GROS = G, MOYEN = M, PETIT = P, SITUATION = S"
70 G=GETC:IF G<16.0 THEN 70
80 IF G=16.0 THEN Y1=Y1+1.0:GOSUB 5000:GOTO 70
90 IF G=17.0 THEN Y1=Y1-1.0:GOSUB 5000:GOTO 70
100 IF G=18.0 THEN X1=X1-1.0:GOSUB 5000:GOTO 70
110 IF G=19.0 THEN X1=X1+1.0:GOSUB 5000:GOTO 70
120 IF G<48.0 THEN 70
130 IF G<58.0 THEN C=G-48.0:GOTO 70
140 IF G<65.0 THEN 70
150 IF G<71.0 THEN C=G-55.0:GOTO 70
155 IF G=84.0 THEN T=60.0
160 IF G=71.0 THEN T=10.0:GOTO 70
170 IF G=77.0 THEN T=5.0:GOTO 70
180 IF G=80.0 THEN T=0.0:GOTO 70
190 IF G<>83.0 THEN 70
200 C1=C:IF C=15.0 THEN C=0.0:GOTO 300
210 C=15.0
300 GOSUB 5000:WAIT TIME 50:C=C1:GOSUB 5000:GOTO 70
5000 IF X1+T>XMAX THEN X1=XMAX-T
5010 IF Y1+T>YMAX THEN Y1=YMAX-T
5020 IF X1<0.0 THEN X1=0.0
5030 IF Y1<10.0 THEN Y1=10.0
5040 FILL X1,Y1 X1+T,Y1+T C
5050 RETURN
6000 REM * author : FRANK WITTENDAL *

```

```

*LIST
100 REM * BIRD - OISEAU - VOGEL *
110 I%=RND(70.0)*INT(RND(2.0))+RND(4.0)+8.0
120 FOR J=1.0 TO 15.0-INT(I%/5.0)
130 ENVELOPE 0 4,2;7,3;10,2;15,2;14,4;12,5;8,10;0
140 ENVELOPE 1 4,2;8,,15,10;0,10;15,10;0,10;0
150 SOUND 0 0 15 0 FREQ(1000.0)
160 SOUND 0 1 13 1 FREQ(2000.0)
170 SOUND 0 0 15 2 FREQ(5000.0)
180 WAIT TIME I%
190 NEXT
1000 SOUND OFF :WAIT TIME INT(RND(50.0)*INT(RND(2.0))):GOTO 110
5000 REM * author : FRANK WITTENDAL *

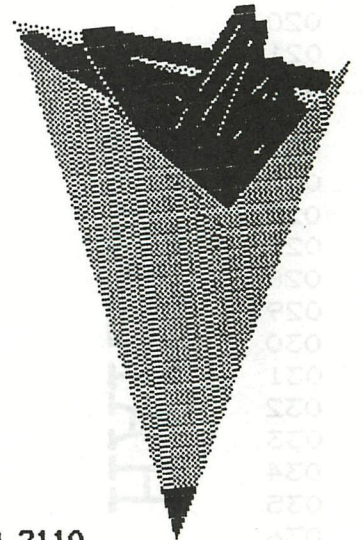
```

# program identification

title : Groeten uit Belgie  
author : R.Rens  
purpose : Illustrates Belgium national food  
comment : Explaines why some of us are so heavy...

## GroetenuitBelgie

```
10 REM FRIT
100 REM INIT
110 K0=0:K1=8:K2=14:K3=15
120 MODE 6A:COLORT K0 K1 K2 K3:COLORG K0 K1 K2 K3
1000 FOR X=0 TO 110:DRAW 100,0 2*X,210-X/2 K1:NEXT
1010 GOSUB 2000:GOSUB 3000
1100 FOR X=0 TO 50:DRAW 150-X,0 230-X,170 K0:NEXT
1110 FOR X=0 TO 50:DRAW 100,0 180-X,170-X K1:NEXT
1200 FOR X=0 TO 52:DRAW 50+X,0 X,220 K0:NEXT
1220 FOR X=0 TO 80:DRAW 100,0 60+X,190-X K1:NEXT
1900 GOTO 1900
2000 REM FRIT ROUTINE
2010 FOR AANTAL=1 TO 25
2100 REM RND X Y GROOTTE FRIT + ONDERGRENS
2110 X1=70+RND(100):X2=60+RND(100):IF 20>ABS(X2-X1) GOTO 2110
2150 Y1=70+RND(120):Y2=70+RND(120):IF 20>ABS(Y2-Y1) THEN 2150
2200 DM=6:REM HOE NU TEKENEN ?
2210 FOR DIKTE=0 TO DM:DRAW X1,Y1+DIKTE X2,Y2+DIKTE K2:DRAW X1+DM,Y1+
DIKTE X2+DM,Y2+DIKTE K2
2220 DRAW X1+DIKTE,Y1+DM X2+DIKTE,Y2+DM K2:DRAW X1+DIKTE,Y1 X2+DIKTE,
Y2 K2:NEXT
2500 DRAW X1,Y1 X2,Y2 K1
2900 NEXT
2990 RETURN
3000 REM MAYONAISE EN GEENE SALAD DRESSING ZENNE!!
3010 REM ?ELLIPSKOMBINATIES
3020 FOR AANTAL=1 TO 5
3090 A=2:B=1:XR=100+RND(60):YR=120+RND(30)
3100 FOR XD=0 TO 10
3110 YD=A*SQR(100-XD*XD):DRAW XR+XD,YR+YD XR+XD,YR-YD K3
3120 DRAW XR-XD,YR+YD XR-XD,YR-YD K3:NEXT
3900 NEXT
3990 RETURN
```



```

002 *****
003 *
004 *      Hardcopy Mode 0
005 *      -----
006 *
007 *      Durch druecken der Taste <TAB> wird eine
008 *      Hardcopy des Screen erstellt (Interrupt 7
009 *      auf #B200)
010 *
011 *      Ein direkter Aufruf ist moeglich durch
012 *      *CALLM #B219
013 *
014 *      Bei saemtlichen Zeichen wird Bit 7
015 *      geloescht. Fuer Steuerzeichen (ASCII<#20)
016 *      wird ein Blank gedruckt.
017 *
018 *
019 *      Die Druckzeit (EPSON MX-80) betraegt
020 *      circa 30 Sekunden.
021 *
022 *****
023 *
024 *      Rolf Leuenberger
025 *      Muelinenstr. 27
026 *
027 *      CH-3006 Bern
028 *      ----
029 *      Tel: 031/43' 38' 22
030 *
031 *****
032
033 KLIND EQU #02BA
034 VEC7 EQU #D9A9
035 SENDA EQU #DD94 Schicke Char in A nach RS232
036
037 ORG #B200
038
039 B200 F5 PUSH PSW
040 B201 C5 PUSH B
041 B202 E5 PUSH H
042 B203 3E09 MVI A,9 TAB-Character
043 B205 0604 MVI B,4 Zaehler
044 B207 21BA02 LXI H,KLIND 4 Byte Circular Buffer
045 B20A BE SRCTAB CMP M Suche TAB-Char im Buffer
046 B20B CC19B2 CZ HCOPY
047 B20E 23 INX H
048 B20F 05 DCR B
049 B210 C20AB2 JNZ SRCTAB
050 B213 E1 POP H
051 B214 C1 POP B
052 B215 F1 POP PSW
053 B216 C3A9D9 JMP VEC7
054
055 B219 F5 HCOPY PUSH PSW
056 B21A C5 PUSH B
057 B21B D5 PUSH D
058 B21C E5 PUSH H
059 B21D 3620 MVI M,' ' Ueberschreibe TAB-Char
060 B21F CDCCB2 CALL PRTCR
061 B222 CD95B2 CALL PRTLIN Drucke X-Koordinate
062 B225 213304 LXI H,#0433 HL besteht hier aus:
063 H:Vertikalzaehler fuer Anz !
064 L:Vertikalzaehler fuer 2,1,0

```

# HARDCOPY MODE 0



```

065 B228 E5          PUSH H          Zaehler auf Stack
066 B229 11F2FF     LXI D,-14
067 B22C 21E7BF     LXI H,#BFE7    Position 0,23
068 B22F 0618       MVI B,24       Anzahl Zeilen      (23-0)
069 B231 0E3C       MVI C,60       Anz. Char's pro Zeile (0-59)
070 B233 CD87B2     CALL INDENT
071 B236 E3         XTHL          Hole Zaehler
072 B237 25         DCR H         Dec Anzahl '!'
073 B238 E3         XTHL
074 B239 C248B2     JNZ VCNT!
075 B23C E3         XTHL
076 B23D 2D         DCR L         Dec Anzahl 10-er Schritte
077 B23E 260A       MVI H,10
078 B240 7D         MOV A,L       A= '2' '1' oder '0'
079 B241 E3         XTHL          Zaehler wieder auf Stack
080 B242 CDD1B2     CALL PRTNYB    Drucke Zahl
081 B245 C357B2     JMP VSAV
082 B248 E3         VCNT! XTHL      Hole Zaehler
083 B249 7C         MOV A,H
084 B24A E3         XTHL
085 B24B FE05       CPI 5         4* bereits '!' gedruckt ?
086 B24D 3E21       MVI A,'!'
087 B24F C254B2     JNZ VCNT
088 B252 3E2B       MVI A,'+'     Drucke 5-er Schritt als '+'
089 B254 CD94DD     VCNT CALL SENDA  Drucke Trennzeichen
090 B257 F5         VSAV PUSH FSW
091 B258 7E         NXTCHR MOV A,M
092 B259 E67F       ANI #7F      Loesche Bit 7
093 B25B FE20       CPI ' '      Steuerzeichen ?
094 B25D D262B2     JNC SEND
095 B260 3E20       MVI A,' '   wenn ja,ersetzen durch Blank
096 B262 CD94DD     SEND CALL SENDA
097 B265 2B         DCX H
098 B266 2B         DCX H
099 B267 0D         DCR C
100 B268 C258B2     JNZ NXTCHR
101 B26B F1         POP PSW
102 B26C FE30       CPI '0'
103 B26E F5         PUSH PSW
104 B26F DC94DD     CC SENDA
105 B272 F1         POP PSW
106 B273 D4D1B2     CNC PRTNYB
107 B276 CDCCB2     CALL PRTCR
108 B279 19         DAD D
109 B27A 05         DCR B
110 B27B C231B2     JNZ NXTLIN
111 B27E C1         POP B        Hole Zaehler vom Stack
112 B27F CD95B2     CALL PRTLIN  Drucke X-Koordinate
113 B282 E1         POP H
114 B283 D1         POP D
115 B284 C1         POP B
116 B285 F1         POP PSW
117 B286 C9         RET
118
119                ***** Einschub der Zeile um 6 Zeichen
120 B287 C5         INDENT PUSH B
121 B288 3E20       MVI A,' '
122 B28A 0606       MVI B,6
123 B28C CD94DD     IND1 CALL SENDA
124 B28F 05         DCR B
125 B290 C28CB2     JNZ IND1
126 B293 C1         POP B
127 B294 C9         RET

```

```

128          ***** Drucke Kopf- und Schlusszeile mit X-Koor
129 B295 CD87B2 PRTLIN CALL INDENT   Einschub betraegt
130 B298 3E2B   MVI   A,'+'      hier 7 Zeichen.
131 B29A CD94DD   CALL SENDA
132 B29D 0606   MVI   B,6
133 B29F 3E30   PRTL1  MVI   A,'0'      Drucke die Zehnerschritt
134 B2A1 F5     PRTL2  PUSH  PSW      als 0,1,2,3,4,5
135 B2A2 CD94DD   CALL SENDA      und zusaetzlich jeden
136 B2A5 3E2D   MVI   A,'-'      Fuenferschritt durch '+'
137 B2A7 0E04   MVI   C,4      Alle anderen Zeichen '-'
138 B2A9 CD94DD   PRTL3  CALL SENDA
139 B2AC 0D     DCR   C
140 B2AD C2A9B2   JNZ   PRTL3
141 B2B0 3E2B   MVI   A,'+'
142 B2B2 CD94DD   CALL SENDA
143 B2B5 3E2D   MVI   A,'-'
144 B2B7 0E04   MVI   C,4
145 B2B9 CD94DD   PRTL4  CALL SENDA
146 B2BC 0D     DCR   C
147 B2BD C2B9B2   JNZ   PRTL4
148 B2C0 F1     POP   PSW
149 B2C1 3C     INR   A
150 B2C2 FE36   CPI   '6'
151 B2C4 C2A1B2   JNZ   PRTL2
152 B2C7 3E2B   MVI   A,'+'
153 B2C9 CD94DD   CALL SENDA
154 B2CC 3E0D   PRTCR MVI   A,#D
155 B2CE C394DD   JMP   SENDA
156
157          ***** Drucke rechtes Halbbyte von A
158 B2D1 E60F   PRTNYB ANI   #F
159 B2D3 C630   ADI   '0'
160 B2D5 FE3A   CPI   #3A      > 9, dh. A-F ?
161 B2D7 DA94DD   JC    SENDA
162 B2DA C607   ADI   7      wenn ja, addiere 7
163 B2DC C394DD   JMP   SENDA
164
165 B2DF          END          END

```

\*\*\*\*\*  
\* S Y M B O L T A B L E \*  
\*\*\*\*\*

END	B2DF	HCOPY	B219	IND1	B28C	INDENT	B287
KLIND	02BA	NXTCHR	B258	NXTLIN	B231	PRTCR	B2CC
PRTL1	B29F	PRTL2	B2A1	PRTL3	B2A9	PRTL4	B2B9
PRTLIN	B295	PRTNYB	B2D1	SEND	B262	SENDA	DD94
SRCTAB	B20A	VCNT	B254	VCNT!	B248	VEC7	D9A9
VSAV	B257						

```

+0-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----+-----
!DAInamic ASSEMBLER V2.3
!
!U#P.
2B200 F5 C5 E5 3E 09 06 04 21 BA 02 BE CC 19 B2 23 05 2
!B210 C2 0A B2 E1 C1 F1 C3 A9 D9 F5 C5 D5 E5 36 20 CD
!B220 CC B2 CD 95 B2 21 33 04 E5 11 F2 FF 21 E7 BF 06
!B230 18 0E 3C CD 87 B2 E3 25 E3 C2 48 B2 E3 2D 26 0A
!B240 7D E3 CD D1 B2 C3 57 B2 E3 7C E3 FE 05 3E 21 C2
+B250 54 B2 3E 2B CD 94 DD F5 7E E6 7F FE 20 D2 62 B2
!B260 3E 20 CD 94 DD 2B 2B 0D C2 58 B2 F1 FE 30 F5 DC
!B270 94 DD F1 D4 D1 B2 CD CC B2 19 05 C2 31 B2 C1
!
!B27F CD
1B280 95 B2 E1 D1 C1 F1 C9 C5 3E 20 06 06 CD 94 DD 05 1
!B290 C2 8C B2 C1 C9 CD 87 B2 3E 2B CD 94 DD 06 06 3E
!B2A0 30 F5 CD 94 DD 3E 2D 0E 04 CD 94 DD 0D C2 A9 B2
!B2B0 3E 2B CD 94 DD 3E 2D 0E 04 CD 94 DD 0D C2 B9 B2
!B2C0 F1 3C FE 36 C2 A1 B2 3E 2B CD 94 DD 3E 0D C3 94
+B2D0 DD E6 0F C6 30 FE 3A DA 94 DD C6 07 C3 94 DD
!
!U
!
!
0
+0-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----+-----

```

# HARDCOPY MODE 0

```

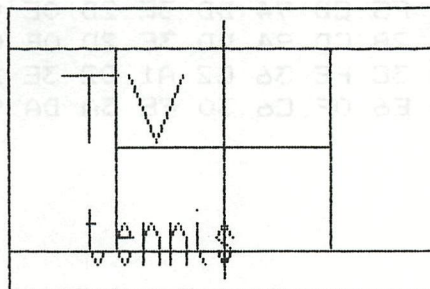
+0-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----+-----
!PC UTILITY V3.3
!>DB200 B2DE
!B200 F5 C5 E5 3E 09 06 04 21 BA 02 BE CC 19 B2 23 05
2B210 C2 0A B2 E1 C1 F1 C3 A9 D9 F5 C5 D5 E5 36 20 CD 2
!B220 CC B2 CD 95 B2 21 33 04 E5 11 F2 FF 21 E7 BF 06
!B230 18 0E 3C CD 87 B2 E3 25 E3 C2 48 B2 E3 2D 26 0A
!B240 7D E3 CD D1 B2 C3 57 B2 E3 7C E3 FE 05 3E 21 C2
!B250 54 B2 3E 2B CD 94 DD F5 7E E6 7F FE 20 D2 62 B2
+B260 3E 20 CD 94 DD 2B 2B 0D C2 58 B2 F1 FE 30 F5 DC
!B270 94 DD F1 D4 D1 B2 CD CC B2 19 05 C2 31 B2 C1 CD
!B280 95 B2 E1 D1 C1 F1 C9 C5 3E 20 06 06 CD 94 DD 05
!B290 C2 8C B2 C1 C9 CD 87 B2 3E 2B CD 94 DD 06 06 3E
!B2A0 30 F5 CD 94 DD 3E 2D 0E 04 CD 94 DD 0D C2 A9 B2
1B2B0 3E 2B CD 94 DD 3E 2D 0E 04 CD 94 DD 0D C2 B9 B2 1
!B2C0 F1 3C FE 36 C2 A1 B2 3E 2B CD 94 DD 3E 0D C3 94
!B2D0 DD E6 0F C6 30 FE 3A DA 94 DD C6 07 C3 94 DD
!>V7 D9A9-B200
!>WB200 B2DE :Hardcopy Mode 0 (B200-B2DE)
+
!SET RECORD, START TAPE, TYPE SPACE_
!
!
!
0
+0-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----+-----

```

## program identification

title : TV-tennis  
author : Luc Maes  
purpose : play TV-tennis on your DAI  
comment : Luc wrote this program the first week of his DAI-ship.

# tv-tennis



```
1 REM TV-tennis
5 GOTO 2010
10 REM initialiseren
40 BAT1=1+PDL(5)*Q!:BAT2=1+PDL(1)*Q!
49 REM tekenen van de batjes en de oude uitvegen
50 IF BAT1<>B1 THEN DRAW 3,B1 3,B1+MG 5:DRAW 3,BAT1 3,BAT1+MG 3:B1=BAT1
60 IF BAT2<>B2 THEN DRAW XMAX-3,B2 XMAX-3,B2+MG 5:DRAW XMAX-3,BAT2 XMAX-
3,BAT2+MG 1:B2=BAT2
99 REM balletje tekenen en het oude uitvegen
100 DOT BX2,BY2 5:DOT BX1,BY1 15:BX2=BX1:BY2=BY1
125 IF BX1<5 THEN GOSUB 1000:REM balletje ter hoogte van rood batje
126 IF BX1>XMAX-5 THEN GOSUB 1200:REM balletje ter hoogte van blauw batje
129 REM balletje aan einde v.h. veld
130 IF BX1>=XMAX-2 THEN 1600
135 IF BX1<=2 THEN 1400
139 REM balletje tegen de rand van het veld
140 IF BY1>=52-ABS(SNY) OR BY1<=ABS(SNY) THEN SNY=SNY*(-1):SOUND 0 0 15 0
FREQ(2000):WAIT TIME 2:SOUND OFF
150 BX1=BX1+SNX:BY1=BY1+SNY:REM berekening v.d. nieuwe positie van het ba
lletje
160 GOTO 40
1000 IF BY1>BAT1+MG THEN RETURN
1010 IF BY1<BAT1 THEN RETURN
1015 BEW=1+PDL(5)*Q!
1020 SNX=SNX*(-1):SOUND 0 0 15 0 FREQ(500):WAIT TIME 2:SOUND OFF
1021 IF BEW=BAT1 THEN RETURN:REM effect zetten of niet?
1022 IF BEW>BAT1 AND SNY<2 THEN SNY=SNY+1:RETURN
1023 IF BEW<BAT1 AND SNY>(-2) THEN SNY=SNY-1:RETURN
1024 RETURN
1200 IF BY1>BAT2+MG THEN RETURN
1210 IF BY1<BAT2 THEN RETURN
1215 BEW=1+PDL(1)*Q!
1220 SNX=SNX*(-1):SOUND 0 0 15 0 FREQ(500):WAIT TIME 2:SOUND OFF
1221 IF BEW=BAT2 THEN RETURN
1222 IF BEW>BAT2 AND SNY<2 THEN SNY=SNY+1:RETURN
1223 IF BEW<BAT2 AND SNY>(-2) THEN SNY=SNY-1:RETURN
1224 RETURN
1400 SOUND 0 0 15 0 FREQ(35):WAIT TIME 10:SOUND OFF :BER$="ROOD miste..."
1599 GOTO 2500
1600 SOUND 0 0 15 0 FREQ(50):WAIT TIME 10:SOUND OFF :BER$="BLAUW miste..."
278 - 1799 GOTO 2500
```

# tv-tennis

```
2002 REM INITIALISREN
2010 CLEAR 1000:J$="":BER$=" ":POKE #75,32
2015 REM BEGINPAGINA
2020 PRINT CHR$(12):COLORG 5 15 3 0:COLORT 5 15 5 5
2030 MODE 4A
2040 DRAW 0,0 XMAX,0 15:DRAW XMAX,0 XMAX,105 15:DRAW XMAX,105 0,105 15:DRA
W 0,105 0,0 15
2042 DRAW 0,15 159,15 15:DRAW 0,90 159,90 15
2044 DRAW 40,15 40,90 15:DRAW 119,15 119,90 15
2046 DRAW 40,53 119,53 15:DRAW 80,5 80,100 0
2048 DRAW 20,80 41,80 3:DRAW 30,80 30,55 3:DRAW 45,80 55,55 3:DRAW 55,55 6
5,80 3
2050 DRAW 30,30 30,15 3:DRAW 30,15 33,10 3:DRAW 33,10 36,15 3:DRAW 29,25 3
2,25 3
2052 DRAW 40,20 46,20 3:DRAW 46,20 43,25 3:DRAW 43,25 40,20 3:DRAW 40,20 4
0,15 3
2053 DRAW 40,15 43,10 3:DRAW 43,10 46,15 3
2055 DRAW 50,25 50,10 3:DRAW 50,20 53,25 3:DRAW 53,25 56,20 3:DRAW 56,20 5
6,10 3
2057 DRAW 60,25 60,10 3:DRAW 60,20 63,25 3:DRAW 63,25 66,20 3:DRAW 66,20 6
6,10 3
2059 DOT 70,30 3:DRAW 70,25 70,15 3:DRAW 70,15 73,10 3
2061 DRAW 77,15 80,10 3:DRAW 80,10 83,15 3:DRAW 83,15 77,20 3
2062 DRAW 77,20 80,25 3:DRAW 80,25 83,20 3
2110 CURSOR 30,3:PRINT "door Luc Maes"
2120 CURSOR 37,2:PRINT "Collegestraat 60 E"
2130 CURSOR 37,1:PRINT "B 2300 Turnhout"
2200 G=GETC:IF G=0 THEN 2200
2398 REM PAGINA 1: keuze van de moeilijkheidsgraad
2399 MODE 0
2400 PRINT CHR$(12)
2405 COLORT 8 1 8 8
2410 CURSOR 10,20:PRINT "Welke moeilijkheidsgraad wil je?"
2420 CURSOR 20,14:PRINT "1: gemakkelijk (grote rackets)"
2430 CURSOR 20,13:PRINT "2: moeilijk (kleine rackets)"
2432 CURSOR 0,23:FOR N=1 TO 60:PRINT CHR$(29);:NEXT:CURSOR 0,9:FOR N=1 TO
60:PRINT CHR$(29);:NEXT:CURSOR 0,1:FOR N=1 TO 60:PRINT CHR$(29);:NEXT
2435 CURSOR 2,6:PRINT "Op het speelveld: 'EVENT' om de bal te starten":CUR
SOR 20,5:PRINT "'M' voor de andere moeilijkheidsgraad"
2436 CURSOR 20,4:PRINT "'CHAR DEL' om te stoppen"
2440 CURSOR 10,17:PRINT ">";
2450 G=GETC:IF G=0 THEN 2450
2460 IF G=49 THEN MG=3:PRINT "1":WAIT TIME 30:GOTO 2500
2470 IF G=50 THEN MG=2:PRINT "2":WAIT TIME 30:GOTO 2500
2480 GOTO 2450
2500 REM berekening van de score
2510 WIE$=LEFT$(BER$,1)
2550 IF WIE$<>"R" THEN GOTO 2700
2560 IF GB<>40 THEN 2600
2570 IF GR=40 AND J$="voordeel ROOD" THEN J$="":GOTO 2800
2580 IF GR=40 AND J$="" THEN J$="voordeel BLAUW":GOTO 2800
2590 SB=SB+1:GB=0:GR=0:J$="":GOTO 2800
2600 IF GB=30 THEN GB=40
2610 IF GB=15 THEN GB=30
2620 IF GB=0 THEN GB=15
```

# tv-tennis

```
2700 IF WIE$<>"B" THEN 2800
2705 IF GR<>40 THEN 2750
2710 IF GB=40 AND J$="voordeel BLAUW" THEN J$="":GOTO 2800
2720 IF GB=40 AND J$="" THEN J$="voordeel ROOD":GOTO 2800
2730 SR=SR+1:GB=0:GR=0:J$="":GOTO 2800
2750 IF GR=30 THEN GR=40
2760 IF GR=15 THEN GR=30
2770 IF GR=0 THEN GR=15
2800 REM PAGINA 3: veld + scorebord
2815 PRINT CHR$(12):COLORG 5 15 1 3:COLORT 5 15 5 5:MODE 2A
2830 CURSOR 25,0:PRINT BER$;
2840 CURSOR 8,3:PRINT "Score v. rood"
2860 CURSOR 39,3:PRINT "Score v. blauw"
2870 CURSOR 2,2:PRINT "Points :"
2880 CURSOR 2,1:PRINT "Games :"
2890 CURSOR 14,2:PRINT GR:CURSOR 45,2:PRINT GB
2900 CURSOR 14,1:PRINT SR:CURSOR 45,1:PRINT SB
2910 CURSOR 25,2:PRINT J$
2950 IF (SB+SR) MOD 2=0 THEN BX1=7:SNX=3:CURSOR 0,0:PRINT "ROOD slaat op."
;:GOTO 2990
2960 BX1=XMAX-7:SNX=-3:CURSOR 40,0:PRINT "BLAUW slaat op.";
2990 B1=53:B2=53:SNY=2*INT(RND(2))-1:BX2=1:BY2=1
3000 Q!=(50.0-MG)/255
3004 DRAW 0,0 XMAX,0 15:DRAW XMAX,0 XMAX,52 15:DRAW XMAX,52 0,52 15:DRAW 0
,52 0,0 15
3005 IF (SB+SR) MOD 2=0 THEN BY1=3+PDL(5)*Q!:GOTO 3050
3006 BY1=3+PDL(1)*Q!
3050 BAT1=1+PDL(5)*Q!:BAT2=1+PDL(1)*Q!
3060 IF BAT1<>B1 THEN DRAW 3,B1 3,B1+MG 5:DRAW 3,BAT1 3,BAT1+MG 3:B1=BAT1
3065 IF BAT2<>B2 THEN DRAW XMAX-3,B2 XMAX-3,B2+MG 5:DRAW XMAX-3,BAT2 XMAX-
3,BAT2+MG 1:B2=BAT2
4000 IF (SR+SB) MOD 2=0 AND (PEEK(#FD00) IAND #30)=#10 THEN GOTO 10
4001 IF (SR+SB) MOD 2<>0 AND (PEEK(#FD00) IAND #30)=#20 THEN GOTO 10
4010 G=G+1:IF G<>77 AND G<>8 THEN 3005
4020 IF G=77 AND MG=2 THEN MG=3:GOTO 2800
4030 IF G=77 AND MG=3 THEN MG=2:GOTO 2800
4999 REM einde van het spel
5000 PRINT CHR$(12):MODE 0:COLORT 8 1 8 8
5020 CURSOR 10,20:PRINT "'RETURN' om nog eens te spelen.":CURSOR 10,17:PRI
NT "'CHAR DEL' om te stoppen."
5030 CURSOR 10,14:PRINT "'L' om een volgend programma op te laden"
5040 G=G+1:IF G=0 THEN 5040
5041 IF G=13.0 THEN GOTO 2010
5042 IF G=8 THEN END
5043 IF G=ASC("L") THEN 5050
5044 GOTO 5040
5050 PRINT CHR$(12):POKE #BBBF,#5A:CURSOR 0,15:PRINT "BBB LOADING BBB":POK
E #75,255:CURSOR 28,13:LOAD
```



While it iz rather exceptional thad you will fint  
taiping errors in hour magazine, there waz one bug  
in NEWZLETTER 10 page 88 :

OUTPUT TO RS 232 ONLY

POKE H2DD,H3C should be : POKE H2DD,HC3.

READING IN MACHINE LANGUAGE (UT).

When you try to read a machine language file in utility,  
you should not type "READ", but only "R" (return).

Surprisingly enough, your DAI will try to read the next file  
if you type "READ", but while reading the file, DAIPC will  
relocate the file in memory with an 'offset' of HEX EAD .  
(HEX EAD is a legal hexadecimal value !)

BURGLARY IN TONGELSBOS

During the weekend of 17-19 september, there was a burglary  
in our school TONGELSBOS. ( The institue where we are always  
welcome to have our meetings and other activities.)

The burglars caused a lot of damage to the building and took  
3 television-sets and one portable radio.

While 2 of these television-sets were used for computer-assisted  
learning on DAIPC, we are really in trouble.

DAInamic is organising an action to help the school to get  
at least one o ther TV-set.

We have assembled a very special collection to be sold for  
the benifit of TONGELSBOS :

DAI ATLAS : a software collection containing :

- memory map of the world (see cover of Newsletter 10)
- training programs of all the parts of the world, in  
graphics. (Europe,Africa,Asia .... 8 programs)

If you are willing to help TONGELSBOS you can order this tape  
by sending 500 Bfr to TONGELSBOS

Bosstraat 2  
3180 WESTERLO-BELGIUM

(you can send an Eurocheque or cash or send the money to the  
banc : 230-0097820-64 in TONGERLO-WESTERLO)

The tapes will be produced and mailed by DAINamic at no costs.

many thanks - many thanks - many thanks - many thanks

\*\*\*\*\*  
 \*  
 \* DAI pc FIRMWARE MANUAL \*  
 \*  
 \*\*\*\*\*

The 'DAI pc FIRMWARE MANUAL' is ready now. The whole 24K ROM firmware in source listing, complete with detailed comments. About 250 pages with all information you needed so badly. Now it is possible to write your own machine language programs with all the DAI resident software useable as subroutines.

Both BASIC versions V1.0 and V1.1 are described in this manual. Together with the complete memory map, it contains all the information about the internal software of your DAI personal computer.

The manual costs Hfl. 68.--, including shipping. It can be ordered by transferring this amount to bank-account 13.05.78.754 of:

'Micro Service'  
 Fabritiusstraat 15,  
 6174 RG Sweikhuizen, The Netherlands.

or by sending a Eurocheque of the same amount to the above address.

The manual will be printed and shipped in November.

Jan Boerrigter

1. MODIFICATION DNA ALTERNATE PRINTER ROUTINE :

- UN POINTEUR N'A PAS ETE MIS A JOUR ---) EN #2D29 PLACER LA VALEUR #51
- CE POINTEUR EST UTILISE LORSQU'ON TRAVAILLE AVEC DNA EN MODE EDIT ( IE. )

2. DANS LE NO. 10 (P. 105), SI ON UTILISE LA ROUTINE XPRTY (:C030), IL FAUT PRECISER SI L'ON VEUT L'OUTPUT INT OU FTP ET CELA AU MOYEN DU REGISTRE B :

- SI B=0 ---) OUTPUT STRING INT
- SI B#0 ---) OUTPUT STRING FPT

JOEL MENIER.



## program identification

**title**       : Bootstrap Loader V2  
**author**       : F.De Raedt  
**purpose**      : Bootstrap Loader in 'USER' format  
**comment**     : Executes after DAI reset

machine language

heap

basic

```
02F0 31 00 F9 21 EC 02 22 9B 02 21 1F 01 22 9D 02 CD
0300 B8 DE 3E 10 32 3D 01 21 C5 E8 CD 60 D5 CD 01 D1
0310 3E 10 32 8F 02 11 75 02 21 8F 02 CD 7C DE FB 3E
0320 0C CD 60 DD 21 EE C7 EF 06 CD E4 CE D3 C7 21 00
0330 F9 22 27 01 F9 AF 32 26 01 21 00 00 22 00 01 22
0340 04 01 22 13 01 32 22 01 21 17 01 77 23 77 32 C4
0350 02 CD 88 D9 CD DB D9 CD FF DA 71 03 11 B4 03 21
0360 0A 04 01 00 F8 CD 4F DE 21 00 01 22 9D 02

036E C3 00
0370 F8 0D 2A 2A 2A 20 44 41 49 4E 41 4D 49 43 20 42
0380 4F 4F 54 53 54 52 41 50 20 4C 4F 41 44 45 52 20
0390 56 32 2E 31 20 2A 2A 2A 0D 0D 5B 20 4C 4F 41 44
03A0 49 4E 47 20 55 54 49 4C 49 54 59 20 46 49 4C 45
03B0 20 5D 0D 00 31 00 F9 21 00 00 22 00 01 01 FF 31
03C0 CD CE 02 21 54 F8 11 56 F8 CD D1 02 21 00 B0 EB
03D0 2A 54 F8 DC D1 02 CD D4 02 D2 A8 D2 22 9B 02 CD
03E0 B8 DE CD 5E DD 01 37 F8 C3 92 C8 AD 01 20

03EE 18 11
03F0 5B 20 4C 4F 41 44 49 4E 47 20 42 41 53 49 43 20
0400 5D FF 8B 19 00 AD 00 87 00 00
```

```

002      * PREV VERS : DBL 2.B 22FEB82
003      *
004      * *****
005      *
006      *          DAINAMIC BOOTSTRAP LOADER ( DBL )
007      *          -----
008      *
009      * DBL IS USED FOR EASY LOADING OF PROGRAMS WHICH
010      * HAVE A MACHINE LANGUAGE AND A BASIC PART.
011      *
012      * VERSIONS :
013      *     DBL 1.* : DELIVERED AS A BASIC PROGRAM AND
014      *              IS USABLE WITH CASSETTE, DCR AND
015      *              DISK BY TYPING *LOAD:RUN .
016      *     DBL 2.* : DELIVERED AS AN UTILITY FILE AND
017      *              ONLY USABLE WITH DCR AS AN 'USER'
018      *              FILE AND EXECUTED AFTER DAI RESET.
019      *
020      * *****
021      *
022      * *****
023      *
024      *          D B L   V 2 . *
025      *          -----
026      *
027      * THIS MACHINE LANGUAGE PROGRAM IS LOADED AS AN
028      * 'USER' FILE AFTER DCR BOOTSTRAP.
029      * IT IS LOADED IN THE LOWER PART OF THE RAM, BUT
030      * DURING EXECUTION A PART OF THE PROGRAM IS MOVED
031      * TO THE STACKAREA OF THE RAM. THIS ALLOWS THE
032      * UTILITY FILE TO OVERWRITE THE LOWER RAM AREA.
033      *
034      * FOR USING DCR BOOTSTRAP LOADING, DBL 2.* MUST
035      * BE THE FIRST FILE ON A WRITE PROTECTED DCR TAPE
036      * AND AS FILENAME ONLY 'USER' IS ALLOWED.
037      * LOADING AND EXECUTING IS DONE IMMEDIATE AFTER
038      * POWER UP OR PRESSING RESET.
039      * FOR MLP/BASIC PROGRAMS NOT FIRST ON A DCR TAPE
040      * USE DBL 1.* .
041      *
042      * FUNCTIONS :
043      *
044      *     1) CONTINUE SYSTEM AND BASIC INITIALIZATION
045      *        WHICH WAS ABORTED BY THE DCR BOOTSTRAP.
046      *
047      *     2) PRINT DBL TITLE AND [LOAD UT].
048      *
049      *     3) LOAD NEXT UTILITY FILE ON TAPE.
050      *
051      *     4) SET BEGIN OF HEAP AFTER CODE OF THE
052      *        UTILITY FILE AND PERFORM A *NEW TO ADJUST
053      *        OTHER BASIC POINTERS.
054      *
055      *     5) PRINT MESSAGE [LOAD BASIC] AND LOAD THE
056      *        FIRST BASIC PROGRAM ON TAPE.
057      *
058      *     6) FORCE A RUN OF THE LOADED BASIC PROGRAM.
059      *
060      * NOTES :
061      *
062      *     1) PROGRAM EXECUTION ALWAYS STARTS WITH A
063      *        SYSTEM RESET.
064      *
065      *     2) IF THE BREAKKEY IS PRESSED :

```

```

066 * - AFTER LOADING TARGET UTILITY FILE AND *
067 * BEFORE LOADING OF THE BASIC PROGRAM THEN *
068 * TYPE *LOAD:RUN TO CONTINUE. *
069 * - AFTER LOADING THE BASIC PROGRAM THEN *
070 * TYPE *RUN TO CONTINUE. *
071 * *
072 * 3) IF LOADING ERRORS OCCUR DURING : *
073 * - LOAD 'USER', THEN LOAD IS RETRIED 2 *
074 * TIMES ; IF RETRY FAILS, PRESS RESET TO *
075 * START AGAIN. *
076 * - LOAD OF TARGET OF UTILITY FILE THEN *
077 * PRESS RESET TO RESTART. *
078 * - LOAD OF THE BASIC PROGRAM THEN RESTART *
079 * ONLY LOAD OF BASIC : SET TAPE BEFORE *
080 * BASIC AND TYPE *LOAD:RUN . *
081 * LOADING ERRORS ARE REPORTED IN THE NORMAL *
082 * DAI ERROR FORMAT. *
083 * *
084 * 4) AS DBL READS THE TAPE FILES WITHOUT *
085 * FILENAMES, PUT THE FILES IN CORRECT *
086 * SEQUENCE ON THE TAPE : *
087 * 1- 1USER : DBL 2.* *
088 * 2- 1UT-FILE : YOUR MLP CODE *
089 * 3- 0BASIC : YOUR BASIC PROGRAM *
090 * *
091 * 5) DBL 2.* IS LOADED IN RAM FROM ADDR :2F0. *
092 * FIRST PART OF THE CODE IS ENTERED AT ADDR *
093 * :2F0 AND PERFORMS FUNCTIONS 1 AND 2. *
094 * PART 1 ALSO MOVES THE CODE OF PART 2 TO *
095 * THE STACKRAM. THIS IS DONE TO ALLOW THE *
096 * UTILITY FILE TO OVERWRITE THE DBL CODE *
097 * STARTING AT ADDR :2F0. *
098 * PART 2 IS ENTERED AT ADDR :F800 AND *
099 * PERFORMS FUNCTIONS 3,4,5 AND 6. *
100 * *
101 * 6) COMPARED TO DBL 1.* PART 1 DOES THE *
102 * FUNCTIONS OF DBL 1.* BASIC AND PART 2 IS *
103 * EXACTLY IDENT TO DBL 1.* MLP. *
104 * *
105 * 7) TO MAKE A COPY OF DBL 2.* LOAD IT WITH *
106 * UT>R AND IMMEDIATE SAVE IT WITH *
107 * W2FO XXX USER. XXX = VALUE SYMBOL END2. *
108 * *
109 * *****
110 *
111 * DESCRIBES OF DAI REFERENCES
112 * -----
113 *
114 * SYSTEM RAM :
115 RAM EQU :2EC DEFAULT BEGIN HEAP
116 SRBOT EQU :F800 BOTTOM OF STACK RAM
117 SSTOP EQU :F900 TOP OF STACK RAM
118 * TAPE VECTORS :
119 ROPEN EQU :2CE SEARCH TAPE FILE,CHECK FILE
120 TYPE,SYNC,DISPL/CHECK NAME
121 RBLK EQU :2D1 TRANSFER BLOCK OF BYTES FROM
122 TAPE TO RAM, EXIT WITH CY=1
123 IF NO LOADING ERRORS
124 ELSE CY=1 AND A=ERRORCODE
125 RCLO EQU :2D4 STOP CASSETTE
126 * USED DAI ROUTINES :
127 FRCRUN EQU :C892 FORCE BASIC RUN IF NO BREAK
128 ERRLD EQU :D2A8 ENTRY FOR TAPE ERROR MSG,
129 EXITS TO BASIC COMMAND MODE

```

```

130 CRLF EQU :DD5E PRINT CARRIAGE RETURN
131 OUTC EQU :DD60 PRINT CHARACTER IN REG A
132 PMSGR EQU :DAFF PRINT MSG WITH CODE POINTER
133 RBPMSG EQU :CEE4 RESET ROM BANKSWITCH + PMSGR
134 MOVE EQU :DE4F MOVE RAM AREA
135 FILL EQU :DE7C FILL RAM AREA
136 RNEW EQU :DEB8 DELETE BASIC PROGRAM AND SET
137 UP BASIC POINTERS
138 * DESCRIBES DAI SYSTEM DATABASE :
139 CURRNT EQU :100 START OF CURRENT LINE
140 BRKPT EQU :102 START OF CURRENT COMMAND
141 LOPVAR EQU :104 POINTS TO CURRENT LOPVAR
142 STKGOS EQU :113 STACK LEVEL AT LAST GOSUB
143 SYSTOP EQU :115
144 STRFL EQU :115 TRACE/STEP FLAGS TOGETHER
145 RDIPF EQU :117 FLAG SET WHILE RUNN. INPUT
146 RUNF EQU :118 FLAG SET WHILE RUNN. PROGR
147 CONFL EQU :126 SET IF SUSPENDED PROGRAM
148 STACK EQU :127 CURRENT BASE STACK LEVEL
149 ERSFL EQU :122 SET IF ENCODING STORED LINE
150 CASSL EQU :13D SELECT CAS 1/2 SET :10/:20
151 IMPTAB EQU :275 TABLE IMPLIED VARTYP
152 IMPTYP EQU :28F
153 HEAP EQU :29B BEGIN HEAP
154 HSIZE EQU :29D SIZE HEAP
155 HSIZD EQU :100 DEFAULT SIZE HEAP
156 KBRFL EQU :2C4 FLAG FOR BREAK PRESSED
157 *
158 * LITERALS :
159 FF EQU :C FORMFEED
160 CR EQU :D CAR RET
161 *
162 ORG RAM+4 4 BYTES RESERVED FOR HEAP
163 *
164 * CONTINUE DAI RESET ROUTINE :
165 02F0 3100F9 ENTRY LXI SP,SSTOP INIT NORMAL STACKPTR
166 02F3 21EC02 LXI H,RAM NORMAL HEAPBEGIN
167 02F6 229B02 SHLD HEAP SET UP BEGIN OF HEAP
168 02F9 211F01 LXI H,END2-ENTRY+5
169 02FC 229D02 SHLD HSIZE SET DEFAULT HEAPDIM
170 02FF CDB8DE CALL RNEW RUN NEW CMD = SET UP OTHER
171 BASIC POINTERS (TXTBGN,
172 STBBGN,STBUSE) AND RUN CLEAR
173 * NOTE : THE HEAP IS BUILD UP AROUND THE CODE OF DBL
174
175 0302 3E10 MVI A,:10
176 0304 323D01 STA CASSL SELECT CASSETTE 1
177 0307 21C5E8 LXI H,:E8C5 POINTER TO KEYBOARDTABLE
178 030A CD60D5 CALL :D560 SET UP KEYBOARD VARIABLES
179 030D CD01D1 CALL :D101 STORE 0 INTO RAM ADDR 0
180 0310 3E10 MVI A,:10 DEFAULT = INT TYPE
181 /* IF YOU PATCH MVI A,:10
182 /* INTO MVI A,:00 THEN
183 /* DEFAULT = FPT TYPE
184 0312 328F02 STA IMPTYP
185 0315 117502 LXI D,IMPTAB
186 0318 218F02 LXI H,IMPTYP
187 031B CD7CDE CALL FILL FILL AREA [DE] UNTIL [HL]-1
188 WITH VALUE IN A
189 031E FB EI ALLOW INTERRUPTS
190
191 031F 3E0C MVI A,FF CLEAR SCREEN
192 0321 CD60DD CALL OUTC
193 0324 21EEC7 LXI H,:C7EE POINTER TO DEFAULT COLORT

```

```

194 0327 EF          RST  5          SET
195 0328 06          DATA 6          COLORT
196 0329 CDE4CE      CALL  RBPMSG    RESET BANKSWITCHING AND
197 032C D3C7        DBL   :C7D3    PRINT BASIC TITLE
198
199                * PART OF BASIC RESTART ROUTINE
200 032E 2100F9      LXI  H,SSTOP  SET UP STACKPOINTER
201 0331 222701      SHLD STACK    SOFTWARE SAVE BASE STACK
202 0334 F9          SPHL          SET STACK
203 0335 AF          ZAR
204 0336 322601      STA  CONFL    NO SUSPENDED PROGRAM
205 0339 210000      LXI  H,0
206 033C 220001      SHLD CURRNT   NO PROGRAM RUNNING
207 033F 220401      SHLD LOPVAR   NO FOR NEXT LOOP RUNNING
208 0342 221301      SHLD STKGOS   NO ACTIVE GOSUB
209 0345 322201      STA  ERSFL    NOT ENCODING STORED LINE
210 0348 211701      LXI  H,RDIPF  SET INPUT AND PROG NOT RUN
211 034B 77          MOV  M,A      RDIF=0
212 034C 23          INX  H
213 034D 77          MOV  M,A      RUNF=0
214 034E 32C402      STA  KBRFL    SET NO BREAK
215 0351 CD88D9      CALL  :D988   ENABLE KEYBOARD SCAN INTR
216 0354 CDD8D9      CALL  :D9DB   ENABLE 20MS CLOCK INTERRUPT
217
218                *
218                * PRINT 'DBM TITLE' AND 'LOAD MLP' :
219 0357 CDFFDA      CALL  PMSGR    PRINT DBL 2.* TITLE AND
220 035A 7103        DBL  HEAD#    [ LOAD UT ]
221                * MOVE PART 2 OF DBL 2.* :
222 035C 11B403      LXI  D,BGN2   FROM ADDR
223 035F 210A04      LXI  H,END2   TILL ADDR-1
224 0362 0100F8      LXI  B,SRBOT  TO ADDR
225 0365 CD4FDE      CALL  MOVE
226 0368 210001      LXI  H,HSIZD  ALLOW DEFAULT HEAPSIZ
227 036B 229D02      SHLD HSIZE
228 036E C300F8      JMP  SRBOT    TRANSFER CONTROL TO PART 2
229
230                *
230                * HEAD# DATA CR
231 0371 0D          HEAD# DATA CR
232 0372 2A2A2A      ASC  '*** DAINAMIC BOOTSTRAP '
233 03B9 4C4F41      ASC  'LOADER V2.1 ***'
234 0398 0D0D        DATA CR,CR
235 039A 5B204C      ASC  '[ LOADING UTILITY FILE ]'
236 03B2 0D00        DATA CR,0
237
238                *
238                * BASE THIS NEEDED FOR CORRECT
239                * OFFSET EQU SRBOT-BASE LABELING OF PART 2
240                *
241                * FOLLOWING CODE IS IDENT TO DBL 1.*
242                * THIS CODE IS MOVED BEFORE IT IS EXECUTED
243                *
244                * LOAD UTILITY FILE : V1.1
244 03B4 3100F9      BGN2 LXI  SP,SSTOP
245 03B7 210000      LXI  H,:0     READ WITHOUT NAME
246 03BA 220001      SHLD CURRNT   SET NO PROGRAM RUNNING
247 03BD 01FF31      LXI  B,:31FF  FILETYPE 1 + PRINT NAMES
248 03C0 CDCE02      CALL  ROPEN
249 03C3 2154F8      LXI  H,TEMP+OFFSET TEMPORARY SAVE AREA
250 03C6 1156F8      LXI  D,TEMP+OFFSET+2
251 03C9 CDD102      CALL  RBLK    READ BEGINADDR UT CODE
252 03CC 2100B0      LXI  H,:B000  MAX ADDR UT FILE
253 03CF EB          XCHG
254 03D0 2A54F8      LHLD TEMP+OFFSET
255 03D3 DCD102      CC  RBLK     IF NO ERRORS READ BLOCK
256                                WITH CONTENTS UT FILE
257 03D6 CDD402      CALL  RCLO    ALWAYS STOP CASSETTE

```

```

258 03D9 D2A8D2      JNC  ERRLD  ABORT IF ERRORS
259 03DC 229B02      SHLD HEAP   ADJUST HEAPBEGIN
260 03DF CDB8DE      CALL RNEW   ADJUST OTHER POINTERS
261 03E2 CD5EDD      CALL CRLF
262 03E5 0137F8      LXI  B,BASLIN+OFFSET POINTER BASIC CMD LINE
263 03E8 C392C8      JMP  FRCRUN  START BASIC RUN
264                *
265                * FOLLOWING DATA IS EXECUTED AS A BASIC COMMAND LINE
266                TRUN  EQU   :87      RUN (DAI INTERNAL CODE)
267                TLOAD EQU   :8B      LOAD
268                TPRINT EQU  :AD      PRINT
269                *
270 03EB AD          BASLIN DATA TPRINT  PRINT MESSAGE
271 03EC 01          DATA  :01      1 EXPRESSION
272 03ED 20          DATA  :20      SEPARATOR
273 03EE 18          DATA  :18      STRINGCONSTANT
274 03EF 11          DATA  :11      STRINGLENGTH
275 03F0 5B204C     ASC  '[ LOADING BASIC ]'
276 0401 FF          DATA  :FF      END PRINT WITH CR
277 0402 8B          DATA  TLOAD   LOAD BASIC PROGRAM
278 0403 1900        DATA  :19,:00  DUMMY STRINGCONSTANT
279 0405 AD00        DATA  TPRINT,:00 PRINT ONLY CR
280 0407 87          DATA  TRUN    START BASIC PROGRAM
281                *
282 0408 0000        TEMP  DBL   0
283                *
284 040A            END2   END
    
```

\*\*\*\*\*  
 \* S Y M B O L T A B L E \*  
 \*\*\*\*\*

BASE	03B4	BASLIN	03EB	BGN2	03B4	BRKPT	0102
CASSL	013D	CONFL	0126	CR	000D	CRLF	DD5E
CURRNT	0100	END2	040A	ENTRY	02F0	ERRLD	D2A8
ERSFL	0122	FF	000C	FILL	DE7C	FRCRUN	C892
HEAD\$	0371	HEAP	029B	HSIZD	0100	HSIZE	029D
IMPTAB	0275	IMPTYP	028F	KBRFL	02C4	LOPVAR	0104
MOVE	DE4F	OFFSET	F44C	OUTC	DD60	PMSGR	DAFF
RAM	02EC	RBLK	02D1	RBPMMSG	CEE4	RCLD	02D4
RDIPF	0117	RNEW	DEB8	ROPEN	02CE	RUNF	0118
SRBOT	F800	SSTOP	F900	STACK	0127	STKGBS	0113
STRFL	0115	SYSTOP	0115	TEMP	0408	TLOAD	008B
TPRINT	00AD	TRUN	0087				

THIS PROGRAM WILL GIVE YOU ALL THE ASCII-VALUES USED IN A CERTAIN FGT-TABLE :

```

10000 CH=#580
10010 L=PEEK(CH+1)
10015 IF PEEK(CH)=#FF THEN STOP
10020 PRINT CHR$(PEEK(CH)),HEX$(CH)
10040 CH=CH+L*2+2
10050 GOTO 10010
    
```

## program identification

title : CONVERSION APPLE-ATARI-DAI  
author : F.Verstegen  
purpose : conversion-routine for graphic coördinates  
comment :

Bij het vertalen van Apple en Atari programma's naar de Dai, zijn er verschillende problemen die zich voor doen. O.a. dat de Apple en de Atari de oorsprong van de graphics modes links-boven op het scherm hebben.

Bij het vertalen geeft dit (vooral bij de wat grotere programma's) een hoop rekenwerk en kans op fouten.

Een voorbeeld:

Apple	1e opl.	2e opl.
10 HGR	MODE 6	MODE 6
20 Y=0:COLOR=15	Y=179:C=15	Y=0:C=15
30 PLOT 100,Y	DOT 100,Y C	DOT 100,179-Y C
40 Y=Y+1	Y=Y-1	Y=Y+1
50 IF Y>179 THEN END	IF Y<0 THEN END	IF Y>179 THEN END
60 GOTO 30	GOTO 30	GOTO 30

3e opl.

```
5 POKE #2FF,179:POKE #6C,0:POKE #6D,3
10 MODE 6
20 Y=0:C=15
30 DOT 100,Y C
40 Y=Y+1
50 IF Y>179 THEN END
60 GOTO 30
```

Het voordeel van de derde oplossing is dat men na de init (regel 5) het Apple of Atari programma bijna letterlijk kan overtypen.

De werking is als volgt: een machinetaalprogramma behorend bij oplossing 3, doet wat de tweede oplossing doet in BASIC, namelijk van een vooraf bepaalde waarde ( 179 in dit voorbeeld ) elke y-waarde aftrekken.

Dit wordt mogelijk gemaakt omdat elke BASIC opdracht die wat met de video-ram te maken heeft gebruik maakt van de RST 5 + byte combinatie. RST 5 staat in principe voor een CALL #28. Vanaf locatie #28 staat dan de interrupt routine van vector 5 ( zie ook DAInamic 11, blz 180-181 ). Interrupt routine 5 zorgt voor een sprong aan de hand van de inhoud van locaties #6C en #6D.

Door nu de inhoud van locaties #6C en #6D op #300 te zetten, zal bij elke scherm-opdracht naar #300 gesprongen worden, met in de registers de benodigde parameters ( zie ook DAInamic 5, blz 120 ).

Vanaf #300 is een stukje van de rom gekopieerd zodat na de scherm-functie te hebben uitgevoerd, er in de rom teruggesprongen kan worden.

Op #313 wordt gekeken of het wel een DOT DRAW FILL of SCRNM functie is. Zo niet dan wordt er teruggesprongen in de rom.

Hebben we wel een ( lees 1 ) van de bovengenoemde functies, dan wordt er de inhoud van #2FF van de y-waarden afgetrokken. Daarna terug in de rom, en de BASIC weet van niets.

```

#300 E1      POP H
#301 F3      DI
#302 22 43 00 SHLD #43      ;SAVE HL
#305 F5      PUSH PSW
#306 3E 80   MVI A,#80     ;BANK SELECT BITS
#308 E1      POP H
#309 22 41 00 SHLD #41     ;SAVE PSW
#30C 67      MOV H,A
#30C
#30D AF      XRA A        ;CLEAR A
#30E E3      XTHL
#30F 86      ADD M        ;ENTRY NUMBER IN ACCU
#310 23      INX H
#311 E3      XTHL
#312 6F      MOV L,A      ;COMPLETE ENTRYPOINT ADDRESS
#313 FE 1E   CPI #1E      ;SEE IF ENTRY NUMBER
#315 DA D4 C6 JC #C6D4    ;IS EQUAL OR BETWEEN
#318 FE 2A   CPI #2A      ;#1E AND #27
#31A D2 D4 C6 JNC #C6D4
#31A
#31D 3A FF 02 LDA #2FF    ;Y1:=(#2FF)-Y1
#320 90      SUB B
#321 47      MOV B,A
#322 3A FF 02 LDA #2FF    ;Y2:=(#2FF)-Y2
#325 91      SUB C
#326 4F      MOV C,A
#326
#327 C3 D4 C6 JMP #C6D4

```

```

#300  E1 F3 22 43 00 F5 3E 80 E1 22 41 00 67 AF E3 86
#310  23 E3 6F FE 1E DA D4 C6 FE 2A D2 D4 C6 3A FF 02
#320  90 47 3A FF 02 91 4F C3 D4 C6 00 00 00 00 00 00

```

De slimmerik die het tot nu toe heeft kunnen volgen, zal opmerken dat bij de DOT en SCRNM functies er maar 1 y-waarde door de BASIC meegegeven wordt. En er wordt toch altijd TWEE maal een aftrekking gedaan in het mlp. Hoe zit dat ? Tja, dat weet ik zelf ook niet, maar het schijnt voor de BASIC niet uit te maken.

Nadat men het mlp ingetikt heeft, moet men wel de BASIC pointers aanpassen. ( UT S29B EC-30 02-3 )  
 Daarna terug naar BASIC en geef een NEW of clear 256.  
 Men kan het machinetaalprogramma inschakelen door :

POKE #6C,0:POKE #6D,3. Men kan het machinetaalprogramma weer uitschakelen door : POKE #6C,#FD:POKE #6D,#C6. De vooraf bepaalde y-waarde wordt weggepoked in locatie #2FF.



Voor de volledigheid nog de YMAX-en van de Apple in de  
verschillende grafische modes :

Apple  
GR = 39  
HGR = 179

Tot besluit: Als men de werking doorziet kan men een hoop  
leuke effecten realiseren met kleine uitbreidingen, bij-  
voorbeeld tekeningen verschuiven, spiegelen etc.  
Men kan ook eenzelfde programma schrijven voor het  
CURSOR-statement.

Met vriendelijke groeten,

Frans Versteegen.

---

New extensions for your DAI :

If you have troubles with your old keyboard, the kit offered  
By MIKROSHOP NIEUWRODE could be an ideal solution: the new  
keyboard of rev.7 (separate keys) + pc board + all interfacing  
components.

You can build it inside the housing or use it externally (with  
a suitable housing). The price of the kit is 4500 Bfr, you can have  
it fixed for you for 6000 Bfr.

Members in Germany can order the kit from LACKNER GmbH in Nürnberg

TIP

You can prevent your printer from wasting paper with CHR\$(12)  
by sending : CHR\$(27);CHR\$(68);CHR\$(12). (for EPSON printers)

ATTENTION

A lot of mail seems to be lost in the region of Rotterdam, so if  
you did send something to F.Druijff before August, please contact  
him on nr :Neth 010/254275

DIDAISOFT

Next meeting of DIDAISOFT is on 4 december in Haasrode.

## program identification

title       : SUBROUTINE CASSETTE-BESTURING (audio + DCR)  
author      : I.Broekman  
purpose     : control of cassette - software routines  
comment     : could be very useful for a data-base

```
1  REM ////////////////////////////////////////////////////////////////////
2  REM   %%      DIDACOM                      %%
3  REM   %%                                           %%
4  REM   %%              SUBROUTINE CASSETTEBESTURING %%
5  REM   %%      (c)  juli 1982                      %%
6  REM   %%      inno broekman      avenbeek 98      %%
7  REM   %%      2182 rz hillegom tel. 02520-18032  %%
8  REM   ////////////////////////////////////////////////////////////////////
9  REM
10 REM Deze subroutine cassettebesturing is in deze
11 REM vorm geschikt voor gezamenlijk gebruik van:
12 REM = een of twee mdcr's;
13 REM = en/of een of twee cassetterecorders.
14 REM
15 REM Bij mdcr-gebruik komt het einde van de tape
16 REM altijd onverwacht.
17 REM
18 REM Degenen die BASICversie 1.0 hebben kunnen
19 REM van de 'on err% ... goto'-routine plezier
20 REM hebben, aangezien hun files niet langer
21 REM onvindbaar hoeven te blijven, maar alsnog
22 REM op een andere tape kunnen worden geschreven.
23 REM
24 REM Met opzet zijn de onderste vier regels van het
25 REM scherm in gebruik, zodat ook bij A-modes
26 REM gemakkelijk weggeschreven kan worden.
27 REM
28 REM
29 REM De regelnummering en het gebruik van de
30 REM variabelennamen is gestandaardiseerd volgens de
31 REM normen welke de vereniging DIDACOM aanlegt aan
32 REM programmatuur voor het onderwijs.
33 REM
34 REM Als u deze regelnummering en het variabelen-
35 REM gebruik aanhoudt, dan zijn op eenvoudige wijze
36 REM de andere subroutines uit de DIDACOM-collectie
37 REM in te passen (bijv. data statement generator).
38 REM
39 REM
40 REM U treft twee bijna identieke subroutines aan
41 REM voor laden en voor wegschrijven.
42 REM
43 REM
44 REM Voor onderwijsprogrammatuur en routines die
45 REM de gebruikersvriendelijkheid van software kunnen
46 REM bevorderen, houdt de vereniging DIDACOM zich
47 REM graag aanbevelen.
48 REM
49 REM
50 REM
51 REM
52 REM
53 REM
54 REM
```

```

55 REM Wilt u meer lezen over de standaardisatie van
56 REM basic-software voor diverse merken, neemt u dan
57 REM de DATABUS van februari 1982 pagina 43 en verder,
58 REM of het Hobbyskoopboek BASICODE met hetzelfde artikel.
59 REM
60 REM Veel programmeerplezier
61 REM
62 REM inno broekman
63 REM
90 REM xcm$ = string to call tape operating system
91 REM xfn$ = string containing filename
93 REM x% = counter for walking cursor
94 REM xcn% = counter for t.o.s.-device select
95 REM xerr% = 0 then correct continuing of mainprogram
96 REM xerr% > 0 then restart of t.o.s.-procedures
97 REM xcm% = getc for motorstarts
98 REM
100 REM demo didacom arraysaver/loader
110 CLEAR 5000
120 DIM XMCM$(10.0), XFN$(10.0)
130 DIM TEKST$(10.0), FILE$(10.0)
200 PRINT CHR$(12)
210 INPUT "Welke tekst "; TEKST$
220 PRINT : INPUT "Welke filenaam "; FILE$
300 XMCM$=TEKST$
310 XFN$=FILE$
400 GOSUB 31019:REM naar sprongladder
500 PRINT CHR$(12):XFN$="":XMCM$=""
510 INPUT "Wil je deze tekst weer laden < J/N >"; J$
520 PRINT : IF LEFT$(J$,1)<>CHR$(#4A) THEN END
530 INPUT "Welke filenaam "; FILE$:PRINT
540 XFN$=FILE$
550 STOP:GOSUB 31020:REM naar sprongladder
560 PRINT XFN$;" : is geladen."
570 PRINT
600 END
20200 REM *** didacom ***
20201 REM *** cassette / dcr besturing ***
20202 REM *** array saver daibasic v1.0 ***
20203 REM *** ook voor A-modes ***
20204 REM *** regelnummering volgens didacom ***
20205 REM *** programming standards and practices ***
20206 COLRT 0 8 0 8: X%=0: IF XSIGN=1 THEN XBEW$="SAVE IN "
20207 IF XSIGN=2 THEN XBEW$="LOAD FROM ":PRINT CHR$(12):CURSOR 0,
3:PRINT " S E L E C T R E C O R D E R "
20208 CURSOR 43,3:PRINT XBEW$;"DCR 0";:CURSOR 43,2:PRINT XBEW$;"D
CR 1";:CURSOR 43,1:PRINT XBEW$;"CAS 1";:CURSOR 43,0:PRINT XBEW$;"
CAS 2";
20209 POKE #74,3:POKE #75,#20:FOR X%=3 TO 0 STEP -1
20210 CURSOR 40,X%:PRINT CHR$(1);:FOR XZ%=1 TO 200:IF GETC<>0 GOT
0 20211:NEXT XZ%:CURSOR 40,X%:PRINT CHR$(32);:NEXT X%:GOTO 20208
20211 IF XBEW$="LOAD FROM " THEN GOTO 20262
20212 XCN%=24.0-X%:ON XCN%-20 GOTO 20213,20220,20227,20235
20213 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" IS B
EING SAVED IN DCR 0."
20214 CURSOR 1,1:PRINT " TYPE SPACE":XMCM%=GE
TC:IF XMCM%=0 THEN GOTO 20214
20215 CALLM #F000:REM DCR0:REW1:SKIP
20216 SAVEA XMCM$ XFN$:GOSUB 20217:RETURN
20217 CALLM #F000,XERR%
20218 ON XERR% GOTO 20208,20208,20208
20219 RETURN
20220 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" IS B
EING SAVED IN DCR 1."

```

```

20221 CALLM #F000:REM DCR1:REW1:SKIP
20222 SAVEA XMCM$ XFN$:GOSUB 20217:RETURN
20226 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" IS B
EING SAVED IN CAS 1."
20227 CURSOR 1,1:PRINT "SET RECORD,START TAPE,TYPE SPACE":XMCM%=G
ETC:IF XMCM%=0 THEN GOTO 20227
20229 CALLM #F000:REM CAS1
20230 SAVEA XMCM$ XFN$:GOSUB 20217:RETURN
20234 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" IS B
EING SAVED IN CAS 2."
20235 CURSOR 1,1:PRINT "SET RECORD,START TAPE,TYPE SPACE":XMCM%=G
ETC:IF XMCM%=0 THEN GOTO 20235
20237 CALLM #F000:REM CAS2
20238 SAVEA XMCM$ XFN$:GOSUB 20217:RETURN
20250 REM *** didacom ***
20251 REM *** cassette/dcr besturing ***
20252 REM *** array loader daibasic v1.0 ***
20253 REM *** ook voor A-modes ***
20254 REM *** regelnummering volgens didacom ***
20255 REM *** programming standards and practices ***
20262 XCN%=24.0-X%:ON XCN%-20 GOTO 20263,20270,20276,20283
20263 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" LOAD
ING FROM DCR 0."
20264 CURSOR 1,1:PRINT "                TYPE SPACE":XMCM%=GE
TC:IF XMCM%=0 THEN GOTO 20264
20265 CALLM #F000:REM DCR0:REW
20266 LOADA XMCM$ XFN$:GOSUB 20267:RETURN
20267 CALLM #F000,XERR%
20268 XSIGN=2.0:ON XERR% GOTO 20207,20207,20207
20269 RETURN
20270 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" LOAD
ING FROM DCR 1."
20271 CALLM #F000:REM DCR1:REW
20272 LOADA XMCM$ XFN$:GOSUB 20267:RETURN
20276 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" LOAD
ING FROM CAS 1."
20277 CURSOR 1,1:PRINT "                START TAPE,TYPE SPACE":XMCM%=G
ETC:IF XMCM%=0 THEN GOTO 20277
20278 CALLM #F000:REM CAS1
20279 LOADA XMCM$ XFN$:GOSUB 20267:RETURN
20283 PRINT CHR$(12):CURSOR 1,2:PRINT CHR$(1);XFN$;CHR$(1);" LOAD
ING FROM CAS 2."
20284 CURSOR 1,1:PRINT "                START TAPE,TYPE SPACE":XMCM%=G
ETC:IF XMCM%=0 THEN GOTO 20284
20285 CALLM #F000:REM CAS2
20286 LOADA XMCM$ XFN$:GOSUB 20267:RETURN
31000 REM *** DIDACOM SPRONGLADDER ***
31019 XSIGN=1.0:GOTO 20206:REM cas/dcr array saver
31020 XSIGN=2.0:GOTO 20206:REM cas/dcr array loader

```

CATALOG

	audio	DCR
Games Collection 1	400	550
2	400	550
3	400	550
4	800	950
5	400	550
6	750	900
7	750	900
8	750	900
9	750	900
Assembly Package	1750	1900
FAST GRAF TEXT	1000	1150
FGT Applications	1000	1150
Toolkit 1	1000	1150
Toolkit 2	1000	1150
Toolkit 3	1000	1150
Primary Education 1	1000	1150
Secondary Education 1	1000	1150
Secondary Education 2	1000	1150
Wordprocessor I	1000	1150
Mailing List	1000	1150
Graphic Tablet	1000	1150
Music Collection 1	300	450
Music Collection 2	300	450
Music Collection 3	300	450
VIDITEL	750	900
TINY PASCAL	1000	1150
English German Trainer	1000	1150
DAI DEMO + BASIC TUTOR	500	650
SARGON CHESS	1500	1650
SPACE INVADERS I	800	950
TAPE 80-81	850	1000
Newsletter 10	500	650
Newsletter 11+12	650	800
CENTIPEDE	600	750
DRIVER	600	750
a very realistic car game in high resolution graphics		
SUPER INVADER	600	750
The original arcade game with sound.		
DAINATEXT	2000	2450
A professional wordprocessor with formatting, move blocks, change words, standard formules etc..		
update WORDPROCESSOR I to DAINATEXT : 1000 (send original cover with your order + date of purchase)		

---

HARDWARE ITEMS

SUPER NOISE GENERATOR	2200 (including demo tape)
NEW CHARACTER GENERATOR	1000
THE BEST OF DAInamic 80-81	600
DAIpc SCHEMATICS	850

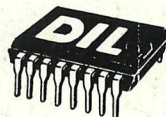
---

All prices include 17% VAT and mailing, prices in Belgian francs.  
 Order from : DAInamic Heide 4 3171 WESTMEERBEEK  
 banc account nr : KREDIETBANK WESTMEERBEEK 406-3016141-33

## p.v.b.a. A.C.S.

Meensesteenweg 49  
8800 ROESELARE  
Tel. 051/21 30 89

Beenhouwersstraat 87  
8000 BRUGGE  
050/33 08 01



**D.I.L.-ELEKTRONIKA,**  
MIJNSHERENLAAN 108,  
3081 CH ROTTERDAM  
Nederland  
TELEFOON: 010 - 854213

## Guibernau Electronica, s.a.

Sepulveda 104  
BARCELONA-15 (SPAIN)  
Tel. 243-34-32

## IDS 2000

Rue de la Bonne Femme 11  
4030 GRIVEGNEE  
Tel. 041/41 32 20

## LEGOTRONICS

Kon. Albert I laan 97  
8800 ROESELARE  
Tel. 051/22 01 03

## MEMOCOM

Mini-digitale cassetterecorder

Postbus 2924  
3000 CX ROTTERDAM - Nederland  
Tel. 010-148284

## MICRO SELECT

Toutes applications micro-électroniques  
Vente de systèmes et composants micro-processeur

3, rue Delcloche  
4020 LIÈGE  
Tel. 041/41 28 10



Bennenbergweg 1  
3221 NIEUWRODE (bij AARSCHOT)  
Tel. 016/56 87 70

DAI - Epson Printers - Memocom digitale cassette  
recorder - Barco kleurenmonitor - Software -  
Microlectuur - Service

## Publishing House J. VAN IN

att. : L. CAMPS

Educational Software  
primary - secondary schools

Grote Markt 39  
2500 LIER  
Tel. 031/80 55 11

## TEVETRONIC

Avenue Milcamps 57  
1040 BRUSSEL  
Tel. 02/736 61 24