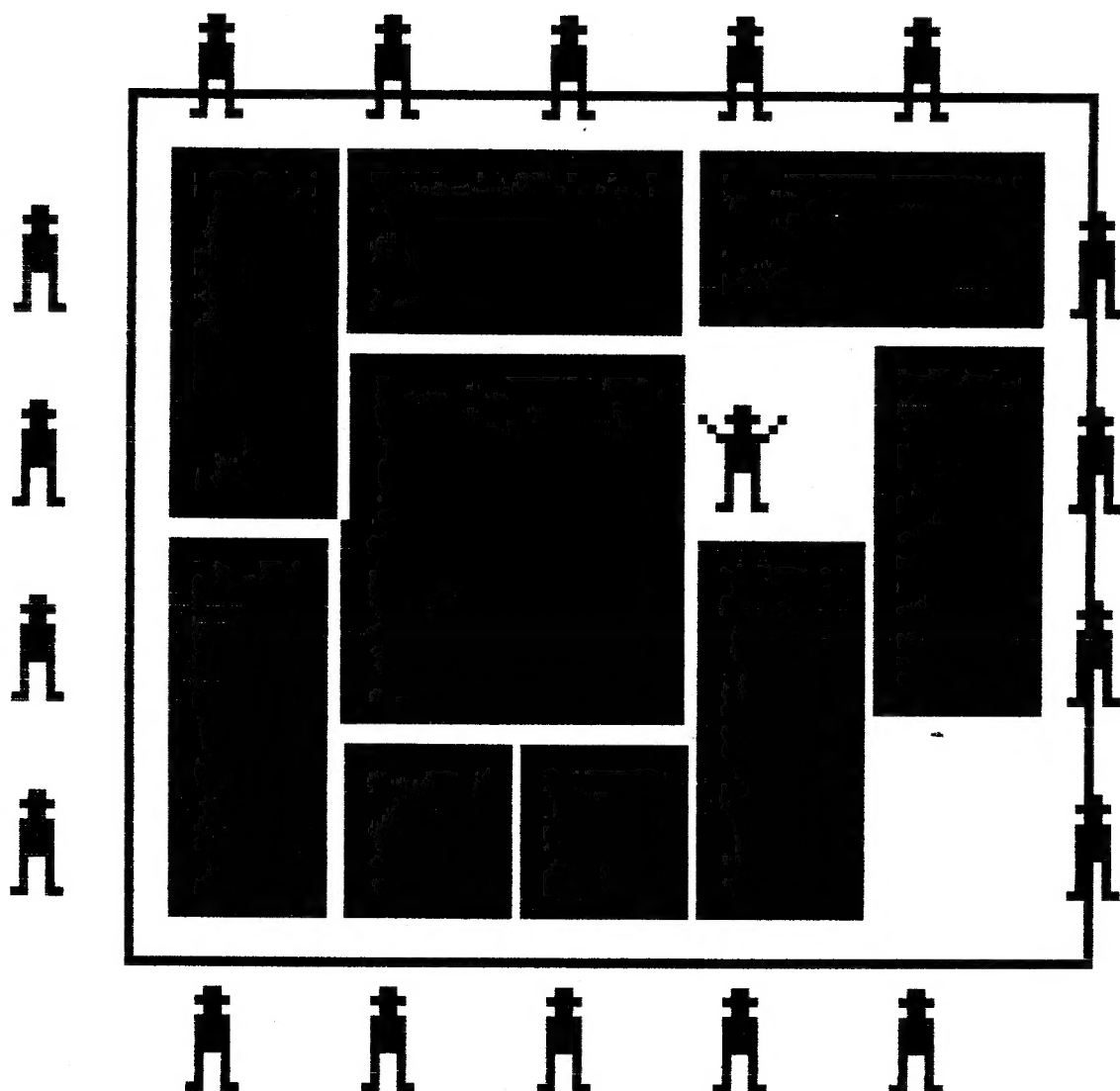


# DAI

## RAMIC

Nr 8 NOV-DEC 1981



GEDRUKTE PERIODIEK verschijnt tweemaandelijks

Verantw. Uitgever : W. HERMANS HEIDE 98 3171 WESTMEERBEEK

COLOFON

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

750 Bfr 50 Gld 50 Dm

Bij toetreding worden de verschenen  
 nummers van de jaargang toegezonden.

DAInamic redactie:

Dirk Bonn 

Freddy De Raedt

Wilfried Hermans

Jules Meulenbergs

Jos Schepens

Roger Theeuws

Bruno Van Rompaey

Jef Verwimp

vormgeving :Ludo van Mechelen

U wordt lid door storting van de  
 contributie op nr406-3016141-33 van  
 KREDIETBANK WESTMEERBEEK, via bank-  
 instelling of POSTGIRO.

Abonnement loopt van januari tot  
 december.

U kan telefonisch contact nemen op  
 nr 016/698623.

correspondentieadres:

DAInamic

Heide 98

3171 WESTMEERBEEK BELGIE

DAInamic verschijnt de eerste week van  
 de pare maanden.

Bijdragen zijn steeds welkom.

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
∅	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

# DAInamic

Naam/Name: \_\_\_\_\_

Datum/Date: \_\_\_\_\_

## PERSONAL COMPUTER USERS CLUB

C L U B I N F O

HOOFDLETTERS AUB  
CAPITALS PLEASE

NAAM/NAME \_\_\_\_\_

VOORNAAM/FIRST NAME \_\_\_\_\_

STRAAT, NUMMER/STREET, NUMBER \_\_\_\_\_

POSTNUMMER, WOONPLAATS/CODE-NUMBER, CITY \_\_\_\_\_

LAND \_\_\_\_\_

TELEFOONNUMMER/PHONE-NUMBER \_\_\_\_\_

GEBOORTEJAAR/BIRTH-YEAR \_\_\_\_\_

DAI GEBRUIKER SINDS/ DAI USER SINCE \_\_\_\_\_

BEROEP/PROFESSION \_\_\_\_\_

COMPUTERACTIVITEITEN, INTERESSE/COMPUTERACTIVITIES, INTEREST \_\_\_\_\_

TYPE CASSETTERECORDER, FLOPPY , DCR \_\_\_\_\_

INTERESSE VOOR 8080 MACHINETAAL / INTEREST IN 8080 MACHINE-LANGUAGE J,N / Y,N

SUGGESTIES IVM DAInamic / SUGGESTIONS FOR DAInamic \_\_\_\_\_

FIRMA + ADRES/ FIRM + ADDRESS \_\_\_\_\_

ADRES WAARNAAR DE NIEUWSBRIEVEN MOETEN VERSTUURD WORDEN/ADDRESS TO SEND THE NEWSLETTERS

DEALER-ADRES/DEALER-ADDRESS \_\_\_\_\_

AARD VAN BETALING/KIND OF PAYMENT

DATUM VAN BETALING/DATE OF PAYMENT

BANKOVERSCHRIJVING/BANK-ORDER	
POSTORDER/ POSTAL ORDER	
CHEQUE / CHECK	
KAS / CASH	

\_\_\_\_\_

Dear members,

With edition 8 we bring you the last issue of 1981. DAI PERSONAL COMPUTER seems to be in the running more than ever, especially in Belgium, where DAIpc was chosen as the best machine by "TESTRAANKOOP", a consumer organisation. Because our club is also in full expansion (close to 600 members), there was need for reorganisation. Please note the new addresses and names for the different services of DAIInamic:

DAInamic subscriptions  
c/o Bruno Van Rompay  
Bovenbosstraat 4  
3044 HAASRODE BELGIUM phone : 016/461085  
the new banc account number for membership fee:  
SOCIETE GENERALE LEUVEN Nr 230-0045353-74  
Please send your info-sheet to this address!  
In despite of inflation, the membership fee will remain the same:  
Benelux : 750 Bfr all other countries : 850 Bfr

DAInamic FIRMWARE (coordination ROM-software)  
c/o Jan Boerrigter  
Fabriliusstraat 15  
6174 RG SWEIKHUIZEN NEDERLAND phone : 4493/2093

DAInamic GAMES & STRATEGY  
c/o Frank Druifjff  
's Gravendijkwal 5A  
3021 EA ROTTERDAM NEDERLAND phone : 010/254275

DAInamic repair service (Nederland)  
c/o Hans Wegman  
Tuinlaan 109  
3195 GJ ROTTERDAM PERNIS tel : 010/386294  
Deze dienst geniet nog niet de medewerking van de firma DAI, maar kan u wel te lange wachttijden besparen!

DAInamic redaction & library  
c/o Wilfried Hermans  
Heide 4 \*\*\* new house number ! \*\*\*  
3171 WESTMEERBEEK BELGIUM phone 016/698623  
Since a few weeks we hire a small workshop, this is open for members on saturday, please contact if you intend to pay us a visit! the address of the workshop: Hoogzand 312 Westmeerbeek.  
In 1982 we intend to release a lot of information about DAI hard & software, contact us if you have discovered interesting details!

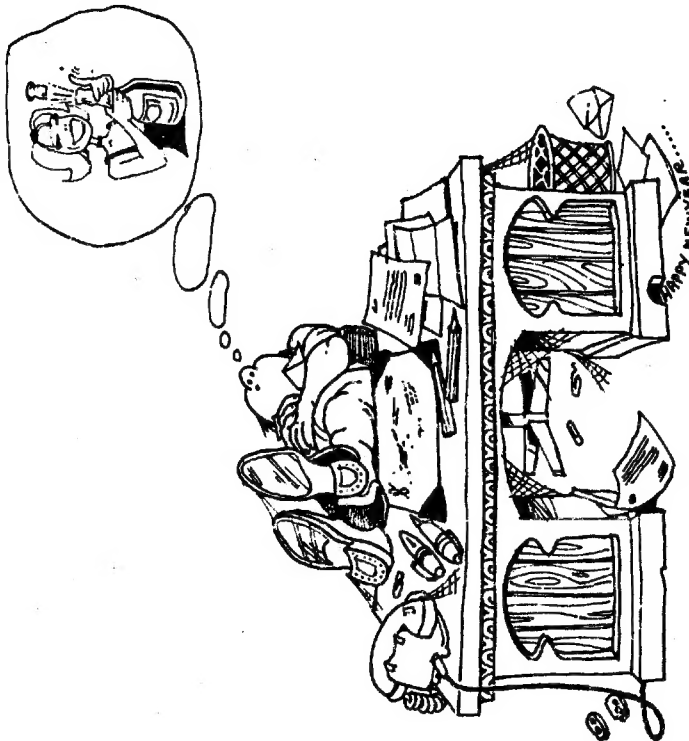
Next issue we will give details information about the following new library tapes: GAMES 6, GAMES 7, TINY PASCAL, GERMAN-ENGLISH GRAMMAR, BASIC TUTOR + 20 DEMO PROGRAMS, SECONDARY EDUCATION 2.

we hope you will enjoy this issue, see you next year  
Wilfried Hermans

BLADWIJZER CONTENTS SOMMAIRE IN DIESEM HEFT

215	The Carpenters Mystery	a not so easy puzzle	L. & W. Hermans
217	NOIS BASICODE	HOBBYSCOOP primer	Th.v. Lieshout
224	Memory Map MODE 3 & 4		F. De Raedt
227	English-Grammar Teacher	a new library tape	W. Hermans
228	Cassette-interface	interface techniques	J. Marchand
229	Cassette-TV		F. de Jong
230	Noise-Random-Paddle	schematics	F. de Jong
231	Cassette-level		Th.v. Lieshout
232	Telex in Basic	read TASS & UPI...	A. De Dauw
	Kraan		Dierckx
233	RTTY / RS232-Parallel	schematics	A. De Dauw
234	Screencopy on MX 80/II	source listing	K. Hoffmann
236	The Hat	graphics	R. Corswandt
	?CHR\$(12) in software	TAB=clear screen	C. Pommerell
237	Letter	DCR-comments	H. Van Cooten
	FOKE-ACTION	speedy animation	W. Hermans
238	Choose the winner of the MATHCHIP 81.....		
239	RASIC-letter		J. Marchand
	Sientje (she will try to kiss you !!)		J. Vewimp
240	Acoustical signal SAVE & LOAD		Th. Wanders
241	Video interface for Teletext tv		Th. Wanders
242	Complementen & Supplementen		Spica
243	Restart routines in DAIPC		J. Boerrigter
244	Memory map (cont) Reset		J. Boerrigter
	DAI Firmware	towards information...	

With many thanks from all of us to the authors, please go on...



```

100 REM THE CARPENTERS MYSTERY w.herhans IMP INT
110 CLEAR 1000:MODE 0:PRINT CHR$(12):COLOR 8 0 8 B
120 CURSOR 5,20:PRINT "THE CARPENTER'S MYS
130 T E R Y"
140 WAIT TIME 20:CURSOR 4,18:D$="Move the red block from
150 upper-left to upper-right"
160 FOR X=0 TO LEN(D$)-1:PRINT MID$(D$,X,1):SOUND 1 0 15 0
170 FREQ(1000,0):WAIT TIME 3:SOUND OFF:WAIT TIME 3:NEXT
180 POKE #B9A6,#CC:POKE #B814,#C8
190 POKE #75,32:CURSOR 5,10:INPUT "WHO PLAYS : COMPUTER [ C ]
200 OR HUMAN [ H ] :";W$
210 IF W$<>"C" AND W$<>"H" THEN 160
220 CURSOR 5,10:PRINT SPC(50)
230 IF W$="C" THEN 500:CURSOR 2,10:PRINT "CURSOR-KEYS to move
240 assistant, SHIFT-CURSOR to move block"
250 IF GETC=0,0 THEN 200
260 GOTO 500
270 X1=A*20:Y1=B*20
280 NOISE 0 15
290 FILL X1+3,Y1+6 X1+6,Y1+9 CO
300 FILL X1+4,Y1+10 X1+5,Y1+12 CO
310 DRAW X1+3,Y1+11 X1+6,Y1+11 CO
320 DRAW X1+3,Y1+12 X1+3,Y1+5 CO
330 DRAW X1+6,Y1+2 X1+6,Y1+5 CO
340 DRAW X1+2,Y1+2 X1+3,Y1+2 CO
350 DRAW X1+6,Y1+2 X1+7,Y1+2 CO
360 SOUND OFF
370 RETURN
380 CO=21:GOSUB 220
390 SOUND 1 0 15 0 FREQ(50,0):SOUND 0 0 15 0 FREQ(5000,0):SOUND
400 0 0 15 2 FREQ(51,0)
410 DRAW X1+2,Y1+9 X1,Y1+11 21
420 WAIT TIME 10
430 CO=SCRN(A*20+7,B*20+7).
440 DRAW X1+2,Y1+9 X1,Y1+11 CO:DRAW X1+7,Y1+9 X1+9,Y1+11 CO
450 SOUND OFF
460 RETURN
470 REM INITIATE
480 MODE 4:COLOR 0 14 3 9
490 FILL 15,15 124,104 21:FILL 16,16 123,103 20
500 FILL 21,61 59,100 22
510 FOR X=1 TO 60:DRAW 60,80 110,80 23:DRAW 110,80 105,75 23:
520 DRAW 110,80 105,85 23
530 DRAW 59+X,61 59+X,100 3:DRAW 20+Y,61 20+X,100 0:NEXT
540 WAIT TIME 100:FILL 16,16 123,103 20
550 DIM F(7,0,6,0)
560 FOR X=1 TO 7:FOR Y=1 TO 6:F(X,Y)=99:NEXT:NEXT
570 F(1,0,1,0)=2:F(2,0,1,0)=2:F(4,0,1,0)=2:F(5,0,1,0)=2
580 F(1,0,3,0)=4:F(3,0,3,0)=1:F(3,0,4,0)=1:F(4,0,3,0)=3:F(4,0,
590 4,0)=3
600 F(3,0,1,0)=0:F(3,0,2,0)=0
610 FOR Y=1 TO 4:FOR X=1 TO 5
620 XP=X*20:YP=Y*20

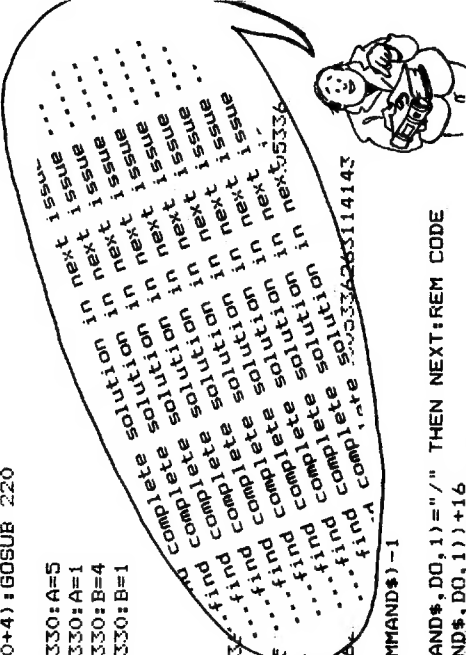
```



```

640 2 IF F(X,Y)=0,0 OR F(X,Y)>5,0 THEN 670:ON F(X,Y) GOSUB 3000,
650 3010,3020,3030
660 CO=23:IF F(X,Y)=4,0 THEN CO=3
670 FILL XP,YP XP+X0,YP+Y0 CO
1000 NEXT:NEXT
1010 IF W$="C" THEN 2000:REM COMPUTER PLAYS
1020 REM MOVE
1030 IF F(4,0,3,0)=4,0 THEN 5000:REM END
1040 IF FLAG=0,0 THEN FLAG=1:A=3:B=3:GOTO 1050:REM FIRST DOT
1050 GETC:IF G=0 THEN 1040
1060 CO=SCRN(A*20+4,B*20+4):GOSUB 220
1070 GOSUB 3040
1080 IF A>5 THEN GOSUB 330:A=5
1090 IF A<1 THEN GOSUB 330:A=1
1100 IF B>4 THEN GOSUB 330:B=4
1110 IF B<1 THEN GOSUB 330:B=1
1120 GOTO 1010
1200 A=3:B=3
2010 COMMAND$="515005152271140413"
2020 COMMAND$=COMMAND$+"...find complete solution in next issue"
2030 "33626311434142705"
2040 26336261422737205"
2050 COMMAND$=COMMAND$+"...find complete solution in next issue"
2060 "27114043625053503"
2070 05150227"
2080 FOR DO=0 TO LEN(COMMAND$)-1
2090 IF MID$(COMMAND$,DO,1)="/" THEN NEXT:REM CODE
2100 G=VAL(MID$(COMMAND$,DO,1))+16
2110 CO=SCRN(A*20+4,B*20+4):GOSUB 220
2120 GOSUB 3040
2130 CO=21:GOSUB 220
2140 WAIT TIME 5:NEXT
2150 GOTO 5000:REM END
2160 X0=18:Y0=18:RETURN
2170 X0=38:Y0=18:RETURN
2180 X0=38:Y0=38:RETURN
2190 REM MOVE CURSOR OR MOVE BLOCK
2200 IF G<16 OR G>23 THEN GOSUB 330:RETURN:REM FALSE KEY
2210 IF G<20 THEN ON G-15 GOTO 3080,3090,3100,3110:GOTO 3120
2220 GOTO 3120:REM MOVE BLOCK
2230 B=B+1:RETURN
2240 B=B-1:RETURN
2250 A=A+1:RETURN
2260 A=A-1:RETURN
2270 SIZE=F(A,B)
2280 IF SIZE=0,0 OR SIZE>4,0 THEN GOSUB 330:RETURN:REM NOT VALID
2290 CO=23:IF SIZE=4 THEN CO=3
2300 ON G-19 GOSUB 3180,3380,3550,3720
2310 SOUND 0 0 15 0 FREQ(1000,0):WAIT TIME 10:SOUND OFF

```



```

3170 RETURN
3180 REM UP
3190 IF B=4 THEN GOSUB 330:RETURN:REM NOT POSSIBLE
3200 IF SIZE=2 OR SIZE=4 THEN IF B>2 THEN GOSUB 330:RETURN
3210 ON SIZE GOTO 3220,3240,3260,3280
3220 IF F(A,B+1.0)<>0.0 THEN GOSUB 330:RETURN
3230 GOTO 3290
3240 IF B>2.0 OR F(A,B+2.0)<>0.0 THEN GOSUB 330:RETURN
3250 GOTO 3290
3260 IF F(A,B+1.0)<>0.0 OR F(A+1.0,B+1.0)<>0.0 THEN GOSUB 330:
RETURN
3270 GOTO 3290
3280 IF B>2.0 OR F(A,B+2.0)<>0.0 OR F(A+1.0,B+2.0)<>0.0 THEN
GOSUB 330:RETURN
3290 GOSUB 3930
3300 FOR Y=0 TO 20:GOSUB 4020
3310 DRAW A*20,B*20+Y A*20+X0,B*20+Y 20
3320 DRAW A*20,B*20+Y0+Y A*20+X0,B*20+Y0+Y 00
3330 NEXT: SOUND OFF
3340 F(A,B+1.0)=F(A,B):ON SIZE GOTO 3350,3350,3360,3360
3350 F(A,B)=0:GOTO 3370
3360 F(A,B)=0:F(A+1.0,B)=0
3370 GOSUB 3950:RETURN

REM DOWN
3380 IF B=1 THEN GOSUB 330:RETURN:REM N.P.
3390 ON SIZE GOTO 3410,3410,3430,3430
3400 IF F(A,B-1.0)<>0.0 THEN GOSUB 330:RETURN
3420 GOTO 3440
3430 IF F(A,B-1.0)<>0.0 OR F(A+1.0,B-1.0)<>0.0 THEN GOSUB 330:
RETURN
3440 GOSUB 3930
3450 FOR Y=0 TO 20:GOSUB 4020
3460 DRAW A*20,B*20+Y0-Y A*20+X0,B*20+Y0-Y 20
3470 DRAW A*20,B*20-Y A*20+X0,B*20-Y 00
3480 NEXT: SOUND OFF
3490 F(A,B-1.0)=F(A,B):ON SIZE GOTO 3500,3510,3520,3530
3500 F(A,B)=0:GOTO 3540
3510 F(A,B+1.0)=0:GOTO 3540
3520 F(A,B)=0:F(A+1.0,B)=0:GOTO 3540
3530 F(A,B+1.0)=0:F(A+1.0,B+1.0)=0
3540 GOSUB 3950:RETURN

REM LEFT
3550 IF A=1 THEN GOSUB 330:RETURN
3560 ON SIZE GOTO 3580,3600,3580,3600
3570 IF F(A-1.0,B)<>0.0 THEN GOSUB 330:RETURN
3580 GOTO 3610
3590 IF F(A-1.0,B)<>0.0 OR F(A-1.0,B+1.0)<>0.0 THEN GOSUB 330:
RETURN
3600 GOSUB 3930
3610 FOR Y=0 TO 20:GOSUB 4020
3620 DRAW A*20-Y+X0,B*20 A*20-Y+X0,B*20+Y0 20
3630 DRAW A*20-Y,B*20 A*20-Y,B*20+Y0 00
3640

```

```

3650 NEXT: SOUND OFF
3660 F(A-1.0,B)=F(A,B):ON SIZE GOTO 3670,3680,3690,3700
3670 F(A,B)=0:GOTO 3710
3680 F(A,B)=0:F(A,B+1.0)=0:GOTO 3710
3690 F(A+1.0,B)=0:GOTO 3710
3700 F(A+1.0,B)=0:F(A+1.0,B+1.0)=0
3710 GOSUB 3950:RETURN

REM RIGHT
3720 IF A=5 THEN GOSUB 330:RETURN
3730 ON SIZE GOTO 3750,3770,3810,3790
3740 IF F(A+1.0,B)<>0.0 THEN GOSUB 330:RETURN
3750 GOTO 3820
3760 IF F(A+1.0,B)<>0.0 OR F(A+1.0,B+1.0)<>0.0 THEN GOSUB 330:
RETURN
3770 GOTO 3820
3780 IF A>4.0 OR F(A+2.0,B)<>0.0 OR F(A+2.0,B+1.0)<>0.0 THEN
GOSUB 330:RETURN
3790 GOTO 3820
3800 IF A>4.0 OR F(A+2.0,B)<>0.0 THEN GOSUB 330:RETURN
3810 GOSUB 3930
3820 FOR Y=0 TO 20:GOSUB 4020
3830 DRAW A*20+Y+X0,B*20 A*20+Y+X0,B*20+Y0 00
3840 DRAW A*20+Y,B*20 A*20+Y,B*20+Y0 20
3850 NEXT
3860 F(A+1.0,B)=F(A,B):ON SIZE GOTO 3880,3890,3900,3910
3870 F(A,B)=0:GOTO 3920
3880 F(A,B)=0:F(A,B+1.0)=0:GOTO 3920
3890 F(A,B)=0:GOTO 3920
3900 F(A,B)=0:F(A,B+1.0)=0
3910 GOSUB 3950:RETURN
3920 ON F(A,B) GOSUB 3000,3010,3020,3030
3930 RETURN
3940 FOR Y=1 TO 4
3950 FOR X=1 TO 5
3960 IF F(X,Y)=2.0 THEN F(X,Y+1.0)=99
3970 IF F(X,Y)=3.0 THEN F(X+1.0,Y)=99
3980 IF F(X,Y)=4.0 THEN F(X+1.0,Y)=99:F(X,Y+1.0)=99:F(X+1.0,Y+
1.0)=99
3990 NEXT: NEXT
4000 RETURN
4010 FOR Y=1 TO 10
4020 SOUND 0 0 15 0 FREQ(300.0+Y*10.0-SIZE*50.0):RETURN
4030 FOR Y=1 TO 10
4040 COLORG 0 0 3 0: SOUND 1 0 15 2 FREQ(1000.0):WAIT TIME 10
4050 COLORG 0 5 3 9: SOUND 1 0 15 2 FREQ(600.0):WAIT TIME 10
4060 NEXT: SOUND OFF
4070 GOTO 100:REM AGAIN

```

playing the game, solution...

The object of the game is to move the RED block from upper-left to upper-right: You can move your little assistant with the CURSOR-keys. You can move a block if the assistant is in the lower-left corner of the block. (this could be modified but will make the program even longer!) If your assistant is in position you move the block with SHIFT-CURSOR. The SOLUTION is entered on lines 2010-2030 encoded in COMMAND\$, following this table: 0 = assistant up 1 = ass down 2 = ass left 3 = ass right 4 = block up 5 = block down 6 = block left 7 = block right. We needed 223 characters (moves) to solve the mystery! Please send your solution if you can make it faster! We will publish COMMAND\$ in next issue, in the mean time... start moving.

1) INLEIDING

Stonds enige tijd zendt de NOS in het programma HOBBYSCOOP elke week een BASIC programma uit. De informatie wordt uitgezonden in de vorm van het normale cassette signaal, dwz zeggen inhoud, seriele standaard en baudrate zoals gebruikt voor SAVE en LOAD. Vermits deze kenmerken specifiek zijn voor een bepaald type computer, was elke NOS uitzending slechts bruikbaar voor een beperkt aantal hobbyisten. Daarom ook werd elke week uitgezonden voor een ander type computer.

Om dit probleem te vermijden werd gezocht naar een signaal-formaat dat eender welke microcomputer kan genereren en inlezen. Nu is het zo dat de meeste microcomputers een programma saven in het formaat waarin een basic programma intern opgeslagen wordt. Dit formaat is eigen aan elk type computer en het is dan ook niet geschikt als universele codering.

Daarom wordt in de NOS BASICODE het programma in de vorm van een "LISTING" overgedragen. Dus een gewone LIST wordt uitgevoerd, maar ipv het programma te printen op het scherm, wordt elk karakter door een speciale routine omgezet in een seriele signaalvorm.

Het speciale aan dit signaal is dat we vooraf alle kenmerken afspreken waaraan het moet voldoen. Dus dit signaal blijft identiek, eender met welk type computer het opgewekt is. Tevens zorgen we ervoor dat dit signaal opneembaar is op een gewone cassette recorder = uit te zenden langs de radio. Op deze wijze hebben we een NOS BASICODE signaal verkregen. Nu maken we voor elke microcomputer ook nog een routine die het omgekeerde doet, dwz BASICODE terug omzetten in het specifieke interne basic formaat. Indien we het ontvangen BASICODE signaal inputten in deze routine dan kunnen alle uitgezonden programma's ingelezen worden door elk type microcomputer.

2) DEFINITIE VAN DE NOS BASICODE

Het programma wordt overgedragen in de vorm waarin het werd ingetypt of waarin het door LIST wordt getoond. Alle letters en cijfers worden in ASCII voorgesteld met het 8ste bit = 1. Spaties in het programma mogen worden onderdrukt behalve in strings en REM's. Elke regel afsluiten met 'CR'. Het programma wordt voorafgegaan door het ASCII teken 'start of text' (STX = #02) en afgesloten door een 'end of text' (ETX = #03).

Als laatste volgt een checksum die het resultaat is van de bitsgewijze exclusieve OR van alle voorgaande bytes.

Serieele code :  
 baudrate : 1200  
 opbouw : 1 startbit logisch 0  
 7 ASCII bits (LSB eerst)  
 1 MSB logisch 1  
 2 stopbits logisch 1

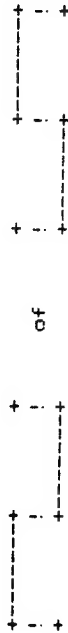
Voorbeeld :

ASCII 'X' = #58 of binair voorgesteld : 0 1 0 1 1 0 0 0  
 of met least significant bits eerst : 0 0 0 1 1 0 1 0

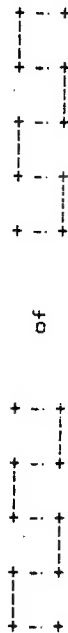


Toonmodulatie :

een 0 bit = 1 volle periode van 1200 Hz (= 1/1.2 ms)



een 1 bit = 2 volle perioden van 2400 Hz (= 1/1.2 ms)



Begin en einde tape signaal :

leader : 3 seconden 2400 Hz (stopbit)

trailer : 3 seconden 2400 Hz (stopbit)

3) NOS BASICODE EN DAI

Voor de DAI microcomputer kan het genereren en terug omzetten van de BASICODE geheel software gebeuren. Dwz de bestaande input/output voor cassette wordt gebruikt.

Dit is in tegenstelling met de andere microcomputers, die naast de software routines meestal ook nog extra hardware nodig hebben. Als de NOS een programma uitzendt, dan verbindt u de cassette-recorder met de tuner/versterker en maakt een opname van het programma en bijbehorende uitleg (best zal u deze opname iets overmoduleren). Daarna wordt de cassette recorder op de normale wijze verbonden met de DAI.

Voor eerst laadt u het machinetaalprogramma Basicode read/write. Dit programma is reeds uitgezonden door de NOS (in DAI formaat). Indien u dit gemist heeft, dan kan u het altijd intypen vanaf de bijgevoegde hex dump.

Als de pointers niet mee gesaved zijn, vergeet dan niet de heappointer aan te passen :

\*KUT >S29B -00 -06 >B \*CLEAR xxxx of \*NEW

U kunt nu verder handelen zoals beschreven in bijgevoegd basic uitleg programma.

Opmerking : in tegenstelling met de gewone DAI LOAD wordt door de basicode leesroutine geen NEW gedaan. Dit laat het mergen van basicprogramma's toe. Indien dit niet gewenst is geef dan een NEW voor het laden.

4) BESLUIT

Ter dokumentatie volgen achter dit artikel hex dump en sourcecode van het machine taal programma en een LIST van het uitlegprogramma. Dit artikel werd gebaseerd op de ingezonden tekst van dhr Th v Lieshout, Postgalei 5, 1687 VP WOGNUM, tel 02297-2648. Hij maakte deel uit van de werkgroep die de NOS BASICODE ontworpen heeft en hij is ook de originator van de mlp vertaal-programma voor DAI.

Leden die de NOS BASICODE gebruiken mogen altijd hun ervaringen opsturen naar de redactie (bv in de vorm van een kort artikel voor de nieuwsbrief).

HEX-DUMP OF BOOTSTRAP BASICODE

```

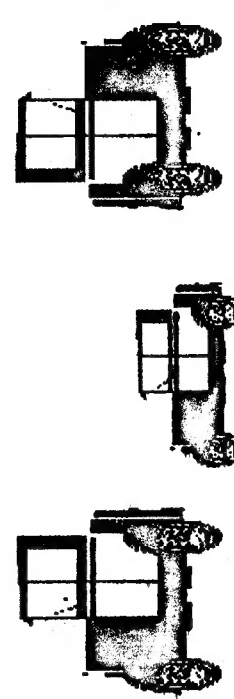
1 PRINT CHR$(12)
10 PRINT "Dit machinetaal programma maakt het mogel ) BASIC "
20 PRINT "programma's uit te wisselen met andere uf "
30 PRINT "In een werkgroep is er een code afgesproken waaraan "
40 PRINT "alle bekende systemen zullen voldoen."
50 PRINT "Voorlopig zijn er een aantal beperkingen betreffende "
60 PRINT "de te gebruiken statements."
70 PRINT "Zo zijn de volgende opdrachten uit den boze i:"
80 PRINT "POKE,PEEK,CURX,CURY,CURSOR,SOUND,ENVELOPE,NOISE, "
90 PRINT "FRE,VARPTR,DOT, DRAW,FILL,MODE,IN,OUT,CALLM."
100 PRINT "Er zijn ook een aantal statements die niet direct "
110 PRINT "uitwisselbaar zijn doch na een kleine verandering wel:"
120 PRINT "bv LEFT$(A$,X) waar X bij de DAI telt vanaf 0"
130 PRINT "enkele andere systemen tellen daarentegen vanaf 1"
140 PRINT "hetzelfde geldt voor MID$ en RIGHT$"
150 PRINT "Ook GETC wordt wel eens anders genoemd :GET"
160 PRINT "Als U ergens leest HOME kunt U dit vervangen door CHR$(12)"
170 PRINT "Bij enkele systemen is het bij een PRINT opdracht "
180 PRINT "niet nodig met aanhalingstekens te sluiten : "
190 PRINT "Dit geeft bij de DAI dan een foutmelding (syntax err)"
200 PRINT "waarna men dan in de EDIT mode dit kan herstellen"
210 PRINT "Ook is de regellengte beperkt tot MAX 64 karakters"
205 PRINT
210 PRINT "DRUK DE SPATIEBALK IN VOOR VERDER LEZEN"
220 A=GETC:IF A<>32.0 GOTO 220
230 PRINT CHR$(12)
240 PRINT " Nu dan enkele aanwijzingen voor het "
250 PRINT "NOS BASICODE machinetaal programma"
260 PRINT
270 PRINT "Dit programma is in te lezen in machinetaal : "
280 PRINT "kies UTILITY), toets R(lead) en return,start tape"
290 PRINT "indien dit niet automatisch gebeurt."
300 PRINT "Als het programma ingelezen is dan staat dit van"
310 PRINT "#29B tot #527 :de pointers worden dus meegeladen."
311 PRINT "Het is nuttig na het inlezen NEW in te tikken"
315 PRINT
320 PRINT "DE NOS BASICODE IS IN TE LEZEN DMV CALLM 750"
330 PRINT
340 PRINT "DE NOS BASICODE IS WEG TE SCHRIFVEN DMV: "
350 PRINT " CALLM 1000:LIST:CALLM 1250"
355 PRINT
360 PRINT "het verdient aanbeveling de cassette voor de leader"
370 PRINT "(=aanloopstrook )te starten"
380 PRINT "het programma wordt op checksom gecontroleerd : "
390 PRINT "bij een fout verschijnt er een F op de plaats van de cursor"
400 PRINT "dit is te verhelpen door POKE #75,#5F"
410 PRINT "Een listing verschijnt in het Oktobernummer(1981)"
420 PRINT "van het blad DATABUS (uitgeverij KLUMER)"
430 PRINT
435 PRINT " DRUK OP DE SPATIEBALK VOOR VERDER LEZEN"
440 A=GETC:IF A<>32.0 GOTO 440
445 PRINT CHR$(12)
450 PRINT "voor de fijnproevers zijn er nog enkele features"
460 PRINT "tw: Bij het wegschrijven is het volgende mogelijk"
470 PRINT "bv CALLM 1000:LIST 68-134:LIST 567-785:CALLM1250"
480 PRINT "Ook is het mogelijk slechts een deel in te lezen"
490 PRINT "U kunt het inlezen dan onderbreken door BREAK in te "
495 PRINT "drukken en daarna CALLM 904"
500 PRINT "in dat geval wordt de checksom niet gecontroleerd"
510 PRINT "Programma: Th v Lieshout te Wognum "

```

```

0298 27 05 00 01 27
02A0 06 28 06 29 06 50 B3 C5 E8 00 00 00 00 00 00 00 00 00
02B0 00 00 00 00 00 00 00 00 00 00 00 00 35 32 37 0D BA 02
02C0 BA 02 01 00 00 C3 B8 D2 C3 F1 D2 D4 C3 27 D4 C3 25
02D0 D3 C3 40 D3 C3 45 D4 C3 A2 D3 C9 00 00 C9 00 00
02E0 C3 B4 DD C9 00 00 24 24 24 3C 24 18 75 07 E5 D5
02F0 C5 21 40 00 36 28 21 75 00 36 FF 2A A3 02 24 22
0300 EC 02 22 C6 03 0E 00 1E 00 FB 06 FF 16 01 F3 26
0310 FF 2E 03 2C FA 09 03 3A 00 FD E6 80 BA CA 13 03
0320 57 7D FE 08 DA 2A 03 7C 17 67 7C 07 F6 F1 FE FF
0330 CA 3C 03 FE F5 CA 48 03 67 C3 11 03 7B FE 00 CA
0340 11 03 78 F6 80 C3 4E 03 29 29 29 78 E6 7F 1F 4F
0350 21 04 F0 1C DA 13 03 FB 7B E6 80 1E 00 C2 79 03
0360 79 AB 4F 78 FE 03 DA 74 03 2A EC 02 77 23 22 EC
0370 02 C3 09 03 1E 80 C3 09 03 79 E6 7F BB 21 75 00
0380 CA 88 03 36 46 C3 8A 03 36 5F 21 40 00 36 30 21
0390 9D 03 22 E1 02 21 96 02 36 01 C3 C2 03 E5 D5 C5
03A0 2A EC 02 EB 2A C6 03 23 22 C6 03 7A BC 7E C2 C2
03B0 03 7B 8D 7E C2 C2 03 21 96 02 36 00 21 B4 DD 22
03C0 E1 02 C1 D1 E1 C9 76 07 E5 D5 C5 21 31 01 36 03
03D0 2A A3 02 24 22 EC 02 22 C6 03 3E C3 32 DD 02 21
03E0 F5 03 22 DE 02 C3 EB 03 C3 C8 03 21 E0 04 3E B2
03F0 36 83 C3 FA 03 E5 D5 C5 F6 80 47 21 E0 04 AE 77
0400 2A EC 02 70 23 22 EC 02 78 C3 C2 03 E5 D5 C5 21
0410 31 01 36 00 21 E0 04 46 2A EC 02 36 83 23 70 22
0420 EC 02 21 40 00 36 28 11 00 00 21 06 FD F3 CD 03
0430 05 CD AD 04 7A FE 65 C2 2E 04 CD FF 04 2A C6 03
0440 46 23 22 C6 03 21 06 FD 11 00 00 CD 20 05 CD C3
0450 04 0E 01 CD FB 04 78 0F 47 DC E5 04 D4 D5 04 0C
0460 79 FE 09 FA 56 04 CD 06 05 CD AD 04 CD 08 05 CD
0470 AD 04 2A EC 02 EB 2A C6 03 13 7A BC C2 ED 04 7B
0480 BD C2 3D 04 21 06 FD 11 00 00 CD FB 04 CD AD 04
0490 7A FE 39 CD 03 05 FA BD 04 FB 21 40 00 36 30 3E
04A0 C9 32 DD 02 21 00 00 22 DE 02 C3 C2 03 CD CD 04
04C0 CD 04 C9 CD CD 04 CD 16 05 CD CD 04 C9 13 7B E6
04D0 01 F6 20 77 C9 CD 20 05 CD C3 04 00 00 00 C9
04E0 FC CD C3 0C 04 CD FF 04 CD AD 04 37 C9 00 7F C3
04F0 3D 04 00 00 00 00 00 00 00 00 C9 CD F6 04 C9 CD
0500 F2 04 C9 CD F9 04 00 CD FF 04 C9 CD FA 04 CD 06
0510 05 C9 CD 0B 05 C9 CD 0B 05 CD 12 05 CD F6 04 C9
0520 CD 19 05 CD F7 04 C9

```





002	TITL	NOS BASICODE RD	TH V LIESHOUT, WOGNUM
003	ORG	:2EC	
004	PUSH H		
005	PUSH D		
006	PUSH B		
007	PUSH 0		
008	LXI	H,:0040	CASS MOTOR 1 ON
009	MVI	M,:28	
010	LXI	H,:75	CURSOR BLACK
011	MVI	M,:FF	END SYMB TABLE
012	LHLD	:2A3	RESERVE 256BYTES
013	INR	H	
014	SHLD	PTRIN	
015	SHLD	PTROUT	
016	MVI	C,:00	CHECKSUM
017	MVI	E,:100	
018	EI		C=CHECKSUM
019	MVI	B,:FF	R=BYTE
020	MVI	D,:01	1ST RND NOT EQU
021	DI		H=SHIFTRG
022	MVI	H,:FF	L=UNITS OF 49US
023	MVI	L,:03	E=BITCOUNTER
024	INR	L	INCR TIMECNTR
025	JM	START	D=MEM CASS INP
026	LDA	:FD00	CASS INTERFACE
027	ANI	:80	MASK
028	CMF	D	EDU?
029	JZ	TCOUNT	N CHANGE NXT RND
030	MOV	D,A	MODIFY D
031	MOV	A,L	TIMECNTR IN ACCU
032	CFI	:0B	t<312 us?
033	JC	SHORT	t<312us
034	MOV	A,H	INPBLUFF IN ACCU
035	RAL		SHIFT,SET 0
036	MOV	H,A	
037	MOV	A,H	
038	RLC		
039	ORI	:F1	SHIFT
040	CPI	:FF	SET 1
041	JZ	BIT=1	COMP FF=2400HZ
042	CPI	:F5	GOTO BIT=1
043	JZ	BIT=0	COMP F5=1200HZ
044	MOV	H,A	GOTO BIT=0
045	JMP	PRESET	MOD INP BUFF
046	MOV	A,E	NOT YET CORRECT
047	CPI	:00	
048	JZ	PRESET	BYTE INTO ACCU
049	MOV	A,B	

050	0343	F680	JMP	:80	SET "1" IN BYTE
051	0345	C34E03	DAD	H	**IRRELEVANT**
052	0348	29	DAD	H	*INSTRUCTIONS*
053	0349	29	DAD	H	**DELAY!****
054	034A	29	DAD	H	BYTE INTO ACCU
055	034B	78	MOV	A,B	SET "0" IN BYTE
056	034C	E67F	ANI	:7F	
057	034E	1F	RAR		MODIFY BYTE
058	034F	47	MOV	B,A	CLEAR SHIFTRG.
059	0350	2104F0	LXI	H,:F004	INCR BITCNTR
060	0353	1C	INR	E	N STARTBIT ON CY
061	0354	DA1303	JC	TCOUNT	
062	0357	FB	EI		
063	0358	7B	MOV	A,E	BITCNTR IN ACCU
064	0359	E680	ANI	:80	CLR EXP. ENDFLG
065	035B	1E00	MVI	E,:00	CLR BITCOUNTER
066	035D	C27903	JNZ	STOTEX	ENDFLAG SET
067	0360	79	MOV	A,C	CHECKSUM IN ACCU
068	0361	AB	XRA	B	EXOR W B
069	0362	4F	MOV	C,A	STORE IN C
070	0363	78	MOV	A,B	BYTE IN ACCU
071	0364	FE03	CPI	:03	END OF TEXT?
072	0366	CA7403	JZ	ENDFLG	JUMP ENDFLAG
073	0369	2AEC02	LHLD	PTRIN	INPOINTER IN HL
074	036C	77	MOV	M,A	STORE A IN BUFF
075	036D	23	INX	H	INCR INPOINTER
076	036E	22EC02	SHLD	PTRIN	STORE INPOINTER
077	0371	C30903	JMP	START	NEXT
078	0374	1E80	MVI	E,:80	SET ENDFLAG
079	0376	C30903	JMP	START	NEXT
080	0379	79	MOV	A,C	CHECKSUM IN ACCU
081	037A	E67F	ANI	:7F	MASK 8th BIT
082	037C	88	CMF	B	COMPARE W BYTE
083	037D	217500	LXI	H,:75	:75=ADDR CURSOR
084	0380	CAB803	JZ	GOOD	"F" IN CURSOR
085	0383	3646	MVI	M,:46	"-" IN CURSOR
086	0385	C38A03	JMP	LAST	CASS MOTOR 1 OFF
087	0388	365F	MVI	M,:5F	
088	038A	214000	LXI	H,:0040	DINC (INCOM DATA)
089	038D	3630	MVI	M,:30	INP. FROM NEXT
090	038F	219D03	LXI	H,:NEXT	INSW IN H&L
091	0392	22E102	SHLD	:2E1	TURN INSW ON
092	0395	219602	LXI	H,:296	
093	0398	3601	MVI	M,:01	
094	039A	C3C203	JMP	RET	
095	039D	E5	PUSH	H	NEXT
096	039E	D5	PUSH	D	
097	039F	C5	PUSH	B	
098	03A0	2AEC02	LHLD	PTRIN	INPOINTER IN H&L
099	03A3	EB	XCHG		EXCHANGE W D&E
100	03A4	2AC603	LHLD	PTROUT	OUTPOINTER IN HL
101	03A7	23	INX	H	INCR OUTPOINTER
102	03AB	22C603	SHLD	PTROUT	MODIFY OUTPOINTER
103	03AB	7A	MOV	A,D	
104	03AC	BC	CMF	H	COMP H W D (MSB)
105	03AD	7E	MOV	A,M	M INTO ACCU
106	03AE	C2C203	JNZ	RET	NOT EQU

```

PAGE 03      NDS BASICODE RD+WR 10SEP81
107 03B1 7B      MOV A,E
108 03B2 8D      CMP L W E (LSB)
109 03B3 7E      MOV A,M
110 03B4 C2C203  JNZ RET
111 03B7 219602  LXI H,:296
112 03BA 3600    MVI M,:00
113 03BC 21B4DD  LXI H,:DDB4
114 03BF 22E102  SHLD :2E1
115 03C2 C1     POP B
116 03C3 D1     POP D
117 03C4 E1     POP H
118 03C5 C9     RET
119
120 *          PTROUT EQU :3C6
121 *          *          *          *          *          *          *          *          *          *
122 *          *          *          *          *          *          *          *          *          *
123 *          *          *          *          *          *          *          *          *          *
124 *          *          *          *          *          *          *          *          *          *
125 *          *          *          *          *          *          *          *          *          *
126 *          *          *          *          *          *          *          *          *          *
127 *          *          *          *          *          *          *          *          *          *
128 *          *          *          *          *          *          *          *          *          *
129 *          *          *          *          *          *          *          *          *          *
130 03CB E5      CSTART
131 03C9 D5      PUSH D
132 03CA C5      PUSH B
133 03CB 213101  LXI H,:131
134 03CE 3603    MVI M,:03
135 03D0 2AA302  LHL D :2A3
136 03D3 24     INR H
137 03D4 22EC02 SHLD PTRIN
138 03D7 22C693 SHLD PTROUT
139 03DA 3EC3    MVI A,:C3
140 03DC 2DD002 STA :2DD
141 03DF 21F503  LXI H,START1
142 03E2 22DE02 SHLD :2DE
143 03E5 C3EB03 JMP JUMP1
144 03EB C3CB03 JMP CSTART
145 03EB 21E004 LXI H,CHECKS
146 03EE 3EB2   MVI A,:B2
147 03F0 36B3   MVI M,:B3
148 03F2 C3FA03 JMP START2
149 03F5 E5     PUSH H
150 03F6 D5     PUSH D
151 03F7 C5     PUSH B
152 03F8 F680  ORI :80
153 03FA 47     MOV B,A
154 03FB 21E004 LXI H,CHECKS
155 03FE AE     XRA M
156 03FF 77     MOV M,A
157 0400 2AEC02 LHL D PTRIN
158 0403 70     MOV M,B
159 0404 23     INX H
160 0405 22EC02 SHLD PTRIN
161 0408 78     MOV A,B
162 0409 C3C203 JMP RET
163 040C E5     TBLEND PUSH H

```

```

PAGE 04      NOS B.      CODE RD+WR 10SEP81
164 040D D5     COMP L W E (LSB)
165 040E C5     M INTO ACCU
166 040F 213101 LXI H,:131
167 0412 3600   NOT EQU
168 0414 21E004 INSW INTO H&L
169 0417 46     SET INSW ON KEYB
170 0418 2AEC02 SUBROUT. KEYB.
171 041B 36B3   HLD :2E1
172 041D 23     POP B
173 041E 70     POP D
174 041F 22EC02 LHL D PTRIN
175 0422 214000 LXI H,:40
176 0425 3628   MVI M,:28
177 0427 110000 LXI D,:0000
178 042A 2106FD LXI H,:FD06
179 042D F3     DI
180 042E CD0305 LEADER
181 0431 CDAD04 CALL BITONE
182 0434 7A     MOV A,D
183 0435 FE39   CPI :39
184 0437 C22E04 JNZ LEADER
185 043A CDF004 CALL DELO59
186 043D 2AC603 LHL D PTROUT
187 0440 46     MOV B,M
188 0441 23     INX H
189 0442 22C603 SHLD PTROUT
190 0445 2106FD LXI H,:FD06
191 0448 110000 LXI D,:0000
192 044B CD2005 CALL DEL312
193 044E CDC304 CALL BITZER
194 0451 0E01   MVI C,:01
195 0453 CDFB04 CALL DELO43
196 0456 78     MOV A,B
197 0457 0F     RRC
198 0458 47     MOV B,A
199 0459 DCE504 CC IN$1
200 045C DAD504 CNC IN$0
201 045F 0C     INR C
202 0460 79     MOV A,C
203 0461 FE09   CPI :09
204 0463 FA2604 JM ASCI1
205 0466 CD0605 CALL DELO90
206 0469 CDAD04 CALL BITONE
207 046C CD0805 CALL DEL144
208 046F CDAD04 CALL BITONE
209 0472 2AEC02 LHL D PTRIN
210 0475 EB     XCHG
211 0476 2AC603 LHL D PTROUT
212 0479 13     INX D
213 047A 7A     MOV A,D
214 047B BC     CMP H
215 047C C2ED04 JNZ IN$DAT
216 047F 7B     MOV A,E
217 0480 BD     CMP L
218 0481 C23D04 JNZ DATA
219 0484 2106FD LXI H,:FD06
220 0487 110000 LXI D,:0000

```

```

164 040D D5     COMP L W E (LSB)
165 040E C5     M INTO ACCU
166 040F 213101 LXI H,:131
167 0412 3600   NOT EQU
168 0414 21E004 INSW INTO H&L
169 0417 46     SET INSW ON KEYB
170 0418 2AEC02 SUBROUT. KEYB.
171 041B 36B3   HLD :2E1
172 041D 23     POP B
173 041E 70     POP D
174 041F 22EC02 LHL D PTRIN
175 0422 214000 LXI H,:40
176 0425 3628   MVI M,:28
177 0427 110000 LXI D,:0000
178 042A 2106FD LXI H,:FD06
179 042D F3     DI
180 042E CD0305 LEADER
181 0431 CDAD04 CALL BITONE
182 0434 7A     MOV A,D
183 0435 FE39   CPI :39
184 0437 C22E04 JNZ LEADER
185 043A CDF004 CALL DELO59
186 043D 2AC603 LHL D PTROUT
187 0440 46     MOV B,M
188 0441 23     INX H
189 0442 22C603 SHLD PTROUT
190 0445 2106FD LXI H,:FD06
191 0448 110000 LXI D,:0000
192 044B CD2005 CALL DEL312
193 044E CDC304 CALL BITZER
194 0451 0E01   MVI C,:01
195 0453 CDFB04 CALL DELO43
196 0456 78     MOV A,B
197 0457 0F     RRC
198 0458 47     MOV B,A
199 0459 DCE504 CC IN$1
200 045C DAD504 CNC IN$0
201 045F 0C     INR C
202 0460 79     MOV A,C
203 0461 FE09   CPI :09
204 0463 FA2604 JM ASCI1
205 0466 CD0605 CALL DELO90
206 0469 CDAD04 CALL BITONE
207 046C CD0805 CALL DEL144
208 046F CDAD04 CALL BITONE
209 0472 2AEC02 LHL D PTRIN
210 0475 EB     XCHG
211 0476 2AC603 LHL D PTROUT
212 0479 13     INX D
213 047A 7A     MOV A,D
214 047B BC     CMP H
215 047C C2ED04 JNZ IN$DAT
216 047F 7B     MOV A,E
217 0480 BD     CMP L
218 0481 C23D04 JNZ DATA
219 0484 2106FD LXI H,:FD06
220 0487 110000 LXI D,:0000

```

```

278 04FA C9          DEL000 RET          DEL000 CALL          DEL016
279 04FB CDF604     DEL043  CALL          DEL043 RET          DEL016
280 04FE C9          DEL059  CALL          DEL059 RET          DEL032
281 04FF CDF204     DEL121  CALL          DEL121 CALL          DEL004
282 0502 C9          DEL090  NOP           DEL090 NOP           DEL004
283 0503 CDF904     DEL086  CALL          DEL086 CALL          DEL059
284 0506 00         DEL144  CALL          DEL144 CALL          DEL000
285 0507 CDF004     DEL171  CALL          DEL171 CALL          DEL090
286 050A C9          DEL171  CALL          DEL171 CALL          DEL144
287 050B CDF0A0     DEL417  CALL          DEL417 CALL          DEL144
288 050E CD0605     DEL246  CALL          DEL246 CALL          DEL171
289 0511 C9          DEL312  CALL          DEL312 CALL          DEL016
290 0512 CD0B05     RET          RET          RET          DEL246
291 0515 C9          DEL312  CALL          DEL312 CALL          DEL012
292 0516 CD0B05     RET          RET          RET          RET
293 0519 CD1205     DEL417  CALL          DEL417 CALL          DEL144
294 051C CDF604     DEL246  CALL          DEL246 CALL          DEL171
295 051F C9          RET          RET          RET          DEL016
296 0520 CD1905     DEL312  CALL          DEL312 CALL          DEL246
297 0523 CDF704     RET          RET          RET          DEL012
298 0526 C9          RET          RET          RET          RET
299 0527              END          END          END          END

```

```

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

```

```

ASCII 0456          BIT=0 0348          BIT=1 033C          BITONE 04AD
BITZER 04C3         BYTE 034E          CHECKS 04E0          CHKSUM 0357
COLDST 02EE        CSTART 03C8         DATA 043D          DEL000 04FA
DEL004 04F9        DEL008 04FB         DEL012 04F7         DEL016 04F6
DEL028 04F3        DEL032 04F7         DEL043 04FB         DEL059 04FF
DEL086 0507        DEL090 0506         DEL121 0503         DEL144 050B
DEL171 0512        DEL246 0519         DEL312 0520         DEL417 0516
ENDFLG 0374        FALSE 0383         GOOD 038B          INSO 04D5
INS1 04E5          INSDAT 04ED        JUMP1 03EB         LAST 038A
LEADER 042E        LONG 0327          NEXT 039D          PRESET 0311
PTRIN 02EC         PTROUT 03C6        RET 03C2           RETURN 0369
SHORT 032A         START 0309         START1 03F5        START2 03FA
STOTEX 0379        TBLEND 040C       TCCOUNT 0313      TOGGLE 04CD
TRAILR 048D

```



```

221 04BA CDFB04     TRAILR  CALL          DEL043
222 04BD CDAD04     TRAILR  CALL          BITONE
223 0490 7A         MOV      A,D
224 0491 FE39      CPI      :39
225 0493 CD0305     CALL    DEL121
226 0496 FABD04     JM      TRAILR
227 0499 FB        EI
228 049A 214000     LXI     H,:0040
229 049D 3630      MVI     M,:30
230 049F 3EC9      MVI     A,:C9
231 04A1 32DD02     STA     :2DD
232 04A4 210000     LXI     H,:0000
233 04A7 22DE02     SHLD   :2DE
234 04AA C3C203     JMP
235 04AD CDCD04     BITONE  CALL
236 04B0 CD1205     CALL    DEL171
237 04B3 CDCD04     CALL    TOGGLE
238 04B6 CD1205     CALL    DEL171
239 04B9 CDCD04     CALL    TOGGLE
240 04BC CD1205     CALL    DEL171
241 04BF CDCD04     CALL    TOGGLE
242 04C2 C9        RET
243 04C3 CDCD04     BITZER  CALL
244 04C6 CD1605     CALL    DEL417
245 04C9 CDCD04     CALL    TOGGLE
246 04CC C9        RET
247 04CD 13        TOGGLE  INX
248 04CE 7B        MOV      A,E
249 04CF E601      ANI     :01
250 04D1 F620      ORI     :20
251 04D3 77        MOV      M,A
252 04D4 C9        RET
253 04D5 CD2005     INSO   CALL    DEL312
254 04D8 CDC304     CALL    BITZER
255 04DB 00        NOP
256 04DC 00        NOP
257 04DD 00        NOP
258 04DE 00        NOP
259 04DF C9        RET
260              EQU     :4E0
261              ORG     :4E2
262 04E2 C30C04     JMP     TBLEND
263 04E5 CDF004     CALL    DELO59
264 04E8 CDAD04     CALL    BITONE
265 04EB 37        STC
266 04EC C9        RET
267 04ED 00        INSDAT NOP
268 04EE 7F        MOV     A,A
269 04EF C33D04     JMP     DATA
270 04F2 00        NOP
271 04F3 00        DELO32
272 04F4 00        DELO2B
273 04F5 00        NOP
274 04F6 00        DELO16
275 04F7 00        DELO12
276 04F8 00        DELO0B
277 04F9 00        DELO04

```

TIME= 3 SEC

OUTPUT POINTER  
BACK

18.00  
 SBT 2 LUBB. 18.30 Liedes ut de lipa leden: - Mis-  
 Een nieuwe dag (Lohan Vermeiren). - Mis-  
 trech bleef toch Mevrouw (Johnny Blanc). -  
 Laat me nog even blijven (Kitty Prinz). Ik hou  
 van Nederland (Wissas Albert). - Mischele  
 Tere (Liesbeth de Vries). - Mischele  
 Twee vrienden (D. Smaal). - Orkestconcert  
 (Therese Steinmetz). - Meise (Wisk en Roel).  
 Als ik met vakantie ga (Jan Blaaser).  
 BRT 3 18.00 Nieuw - 18.15 Uitsluitend - 18.30  
 Het Uitsluitend van de Dinsdagavond.  
 Kerk. - Bron van de Dinsdagavond voor  
 trompet, hoorn en trombone. Revue.  
 RTBF 1 18.05 Sportmagazine - 18.25 Kato-  
 leek godsdienstige uitzending  
 RTBF 2 18.30-19.30 Verkoopslijstenpro-  
 gram  
 RTBF 3 18.05 Autrement dit.  
 MIL 1 18.05 Akkoord - 18.30 Hobbycoop.  
 Programma over stereofonie, elektronica,  
 foto en film, zendamateurs, DX-ers en Lucht-  
 leiding  
 HL 2 18.00 Nieuw - 18.10 Brood & Spelen -  
 o.a. Wet er niet in het boekje staat.  
 18.40-18.10 Liturgie en kerkmusiek met  
 recente liedliederen besproken en platen-  
 gese uit de muziekwereld.  
 WDR 3 18.05-19.30 Pasconnc. nr 11, Mozart:  
 Schiedstahed op. 54, Brahms: Sinfonie nr 6,  
 Dvorak.

```

**DELYXX=DELAY**
*****XX US*****
**CALL+RET=DELAY**
*****27 US*****

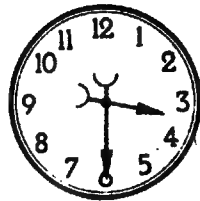
```



```

1 PRINT CHR$(12):REM CLEAR SCHERM
2 PRINT :PRINT :PRINT
3 PRINT "HET VOLGENDE PROGRAMMA IS AFKOMSTIG VAN EEN"
4 PRINT :PRINT "D A I PERSONAL COMPUTER":PRINT
5 PRINT "DIT PROGRAMMA IS BEDOELD OM HET CONVERSIE"
6 PRINT "PROGRAMMA - NOS BASICODE - TE TESTEN"
7 PRINT :PRINT "D I G I T A L E K L O K":PRINT
8 PRINT "BASIC,HARD-&SOFTWARE KORTOM ALLE WARE:"
9 PRINT "TH.V.LIESHOUT"
10 PRINT "POSTGALEI 5"
11 PRINT "1687 VP WOGNUM (N.H.)"
12 CLEAR 700:REM RES. GEHEUGENRUINTE VOOR ARRAYS
13 FOR I=1.0 TO 35000.0:NEXT
14 PRINT CHR$(12):REM CLEAR SCHERM
15 PRINT "TIJDVERTRAGING PROEFONDERVINDELIJK VASTSTELLEN"
16 PRINT
20 INPUT "TIJDVERTRAGING (0-64000)";T:PRINT :PRINT
30 INPUT "12 OF 24 UURS KLOK (12 OF 24)";D:PRINT
35 PRINT
40 INPUT "AANTAL SECONDEN PER STAP (4-60) ";C:PRINT
41 PRINT
45 PRINT "AANTAL KARAKTERS AFHANKELIJK VAN MERK PC"
46 PRINT
50 INPUT "AANTAL KARAKTERS PER REGEL";E:PRINT :PRINT
60 E$=" ":IF E<60.0 THEN E$=" "
70 INPUT "TIJD (UREN,MINUTEN,SECONDEN)";U,M,S:PRINT
80 PRINT CHR$(12):REM CLEAR SCHERM
90 DIM A$(6.0),B$(9.0,6.0)
100 A$(1.0)="111010111111011111001111111111"
110 A$(2.0)="101010001001101100100001101101"
120 A$(3.0)="101010111111111111111011111111"
130 A$(4.0)="101010100001001001101001101001"
140 A$(5.0)="111010111111001111110011111111"
150 A$(6.0)="000000000000000000000000000000"
200 N=INT(U/10.0):J=1.0:IF N=0.0 GOTO 220
210 GOSUB 1000
220 N=U-INT(U/10.0):J=2.0:GOSUB 1000
230 M=INT(M/10.0):J=3.0:GOSUB 1000
240 M=M-INT(M/10.0):J=4.0:GOSUB 1000
250 S=INT(S/10.0):J=5.0:GOSUB 1000
260 S=S-INT(S/10.0):J=6.0:GOSUB 1000
300 PRINT CHR$(12)
310 FOR I=1.0 TO 4.0:PRINT :NEXT
320 FOR I=1.0 TO 9.0:J=1.0:K=I:IF I>1.0 AND I<5.0 THEN J=2.0
330 IF I=5.0 THEN J=3.0
340 IF I>5.0 AND I<9.0 THEN J=4.0
350 IF I=9.0 THEN J=5.0
360 IF I>5.0 THEN K=6.0
370 PRINT E$;B$(J,1.0);E$;B$(J,2.0);E$;E$;B$(J,3.0);
380 PRINT E$;B$(J,4.0);E$;E$;B$(K,5.0);E$;B$(K,6.0)
390 NEXT
400 FOR I=1.0 TO T:NEXT
410 S=S+C:IF S>=60.0 THEN S=S-60.0:M=M+1.0
420 IF M>=60.0 THEN M=M-60.0:U=U+1.0
430 IF U>=D THEN U=U-D:FOR I=1.0 TO 9.0:B$(I,1.0)=" " :NEXT
440 GOTO 200
1000 FOR I=1.0 TO 5.0
1010 B$(I,J)=MID$(A$(I),3*I,J)
1020 IF B$(I,J)="111" THEN B$(I,J)="XXXXX"
1030 IF B$(I,J)="010" THEN B$(I,J)=" X "
1040 IF B$(I,J)="100" THEN B$(I,J)="X "
1050 IF B$(I,J)="001" THEN B$(I,J)=" X"
1060 IF B$(I,J)="101" THEN B$(I,J)="X X"
1070 IF B$(I,J)="011" THEN B$(I,J)=" XX"
1080 NEXT
1090 RETURN

```



## NOS BASICODE

HET VOLGENDE PROGRAMMA IS AFKOMSTIG VAN EEN

D A I PERSONAL COMPUTER

DIT PROGRAMMA IS BEDOELD OM HET CONVERSIE  
PROGRAMMA - NOS BASICODE - TE TESTEN

D I G I T A L E K L O K

BASIC,HARD-&SOFTWARE KORTOM ALLE WARE:  
TH.V.LIESHOUT  
POSTGALEI 5  
1687 VP WOGNUM (N.H.)

TIJDVERTRAGING PROEFONDERVINDELIJK VASTSTELLEN

TIJDVERTRAGING (0-64000)?40000

12 OF 24 UURS KLOK (12 OF 24)?24

AANTAL SECONDEN PER STAP (4-60) ?5

AANTAL KARAKTERS AFHANKELIJK VAN MERK PC

AANTAL KARAKTERS PER REGEL?60

TIJD (UREN,MINUTEN,SECONDEN )?9,12,35

XXXXX	X	XXXXX	XXXXX	XXXXX
X X	X	X	X	X
X X	X	X	XXXXX	XXXXX
X X	X	X	X	X
XXXXX	X	XXXXX	XXXXX	XXXXX
X	X	X		
X	X	X		
X	X	X		
XXXXX	X	XXXXX		

XXXXX	X	XXXXX	X X	XXXXX
X X	X	X	X X	X X
X X	X	X	XXXXX	X X
X X	X	X	X X	X
XXXXX	X	XXXXX	X	XXXXX
X	X	X		
X	X	X		
X	X	X		
XXXXX	X	XXXXX		

CONTROL BYTE

MEMORY-MAP MODE 4

COLOR BYTE

TWO BYTES NOT USED BY BASIC

TWO BYTES NOT USED BY BASIC

0- 8- 16- 24- 32- 40- 48- 56- 64- 72- 80- 88- 96- 104- 112- 120- 128- 136- 144- 152-

129	BFEF	BFEF	BFEF	BFE9	BFE7	BFE5	BFE3	BFE1	BFD9	BFD7	BFD5	BFD3	BFD1	BFCF	BFCD	BFCB	BFC9	BFC7	BFC5	BFC3
128	BFC1	BFC0	BFB8	BFB9	BFB7	BFB5	BFB3	BFB1	BFA9	BFA7	BFA5	BFA3	BFA1	BF9F	BF9D	BF9B	BF99	BF97	BF95	
127	BF93	BF92	BF8F	BF8D	BF8B	BF89	BF87	BF85	BF83	BF81	BF7F	BF7D	BF7B	BF79	BF77	BF75	BF73	BF71	BF6F	BF6D
126	BF65	BF64	BF61	BF5F	BF5D	BF5B	BF59	BF57	BF55	BF53	BF51	BF4F	BF4D	BF4B	BF49	BF47	BF45	BF43	BF41	BF3F
125	BF37	BF36	BF33	BF31	BF2F	BF2D	BF2B	BF29	BF27	BF25	BF23	BF21	BF1F	BF1D	BF1B	BF19	BF17	BF15	BF13	BF11
124	BF09	BF08	BF05	BF03	BF01	BFFF	BEFD	BEFB	BEF9	BEF7	BEF5	BEF3	BEF1	BEED	BEED	BEE9	BEE7	BEE5	BEE3	BEE1
123	BED8	BEDA	BED7	BED5	BED3	BED1	BECF	BECD	BECB	BEC9	BEC7	BEC5	BEC3	BEC1	BEBF	BEBD	BEBB	BEB9	BEB7	BEB5
122	BEAD	BEAC	BEA9	BEA7	BEA5	BEA3	BEA1	BE9F	BE9D	BE9B	BE99	BE97	BE95	BE93	BE91	BE8F	BE8D	BE8B	BE89	BE87
121	BE7F	BE7E	BE7B	BE79	BE77	BE75	BE73	BE71	BE6F	BE6D	BE6B	BE69	BE67	BE65	BE63	BE61	BE5F	BE5D	BE5B	BE59
120	BE51	BE50	BE4D	BE4B	BE49	BE47	BE45	BE43	BE41	BE3F	BE3D	BE3B	BE39	BE37	BE35	BE33	BE31	BE2F	BE2D	BE2B
119	BE23	BE22	BE1F	BE1D	BE1B	BE19	BE17	BE15	BE13	BE11	BE0F	BE0D	BE0B	BE09	BE07	BE05	BE03	BE01	BDFD	BDFB
118	BDF5	BDF4	BDF1	BDEF	BDED	BDEB	BDE9	BDE7	BDE5	BDE3	BDE1	BDDF	BDDD	BDDB	BDD9	BDD7	BDD5	BDD3	BDD1	BDCF
117	BDC7	BDC6	BDC3	BDC1	BDBF	BDBD	BDBB	BDB9	BDB7	BDB5	BDB3	BDB1	BDAF	BDAE	BDA8	BDA6	BDA4	BDA2	BDA0	BD9F
116	BD99	BD98	BD95	BD93	BD91	BDF8	BDF6	BDF4	BDF2	BDF0	BDE8	BDE6	BDE4	BDE2	BDDF	BDDD	BDDB	BDD9	BDD7	BDD5
115	BD68	BD6A	BD67	BD65	BD63	BD61	BDF5	BDF3	BDF1	BDE9	BDE7	BDE5	BDE3	BDE1	BDDF	BDDD	BDDB	BDD9	BDD7	BDD5
114	BD3D	BD3C	BD39	BD37	BD35	BD33	BD31	BD2F	BD2D	BD2B	BD29	BD27	BD25	BD23	BD21	BD1F	BD1D	BD1B	BD19	BD17
113	BDF0	BDFE	BDF8	BDF9	BDF7	BDF5	BDF3	BDF1	BDF9	BDF7	BDF5	BDF3	BDF1	BDF9	BDF7	BDF5	BDF3	BDF1	BDF9	BDF7
112	BCE1	BCE0	BCE8	BCE9	BCE7	BCE5	BCE3	BCE1	BCE9	BCE7	BCE5	BCE3	BCE1	BCE9	BCE7	BCE5	BCE3	BCE1	BCE9	BCE7
111	BCB3	BCB2	BCAF	BCAD	BCAB	BCA9	BCA7	BCA5	BCA3	BCA1	BC9F	BC9D	BC9B	BC99	BC97	BC95	BC93	BC91	BC8F	BC8D
110	BC85	BC84	BC81	BC7F	BC7D	BC7B	BC79	BC77	BC75	BC73	BC71	BC6F	BC6D	BC6B	BC69	BC67	BC65	BC63	BC61	BC5F
109	BC57	BC56	BC53	BC51	BC4F	BC4D	BC4B	BC49	BC47	BC45	BC43	BC41	BC3F	BC3D	BC3B	BC39	BC37	BC35	BC33	BC31
108	BC29	BC28	BC25	BC23	BC21	BC1F	BC1D	BC1B	BC19	BC17	BC15	BC13	BC11	BC0F	BC0D	BC0B	BC09	BC07	BC05	BC03
107	B8FB	B8FA	B8F7	B8F5	B8F3	B8F1	B8EF	B8ED	B8EB	B8E9	B8E7	B8E5	B8E3	B8E1	B8DF	B8DD	B8DB	B8D9	B8D7	B8D5
106	B8CD	B8CC	B8C9	B8C7	B8C5	B8C3	B8C1	B8BF	B8BD	B8BB	B8B9	B8B7	B8B5	B8B3	B8B1	B8AF	B8AD	B8AB	B8A9	B8A7
105	B89F	B89E	B89B	B899	B897	B895	B893	B891	B88F	B88D	B88B	B889	B887	B885	B883	B881	B87F	B87D	B87B	B879
104	B871	B870	B86D	B86B	B869	B867	B865	B863	B861	B85F	B85D	B85B	B859	B857	B855	B853	B851	B84F	B84D	B84B
103	B843	B842	B83F	B83D	B83B	B839	B837	B835	B833	B831	B82F	B82D	B82B	B829	B827	B825	B823	B821	B81F	B81D
102	B815	B814	B811	B80F	B80D	B80B	B809	B807	B805	B803	B801	B7FF	B7FD	B7FB	B7F9	B7F7	B7F5	B7F3	B7F1	B7EF
101	B7E7	B7E6	B7E3	B7E1	B7DF	B7DD	B7DB	B7D9	B7D7	B7D5	B7D3	B7D1	B7CF	B7CD	B7CB	B7C9	B7C7	B7C5	B7C3	B7C1
100	B7B9	B7B8	B7B5	B7B3	B7B1	B7AF	B7AD	B7AB	B7A9	B7A7	B7A5	B7A3	B7A1	B79F	B79D	B79B	B799	B797	B795	B793
99	B78B	B78A	B787	B785	B783	B781	B77F	B77D	B77B	B779	B777	B775	B773	B771	B76F	B76D	B76B	B769	B767	B765
98	B75D	B75C	B759	B757	B755	B753	B751	B74F	B74D	B74B	B749	B747	B745	B743	B741	B73F	B73D	B73B	B739	B737
97	B72F	B72E	B72B	B729	B727	B725	B723	B721	B71F	B71D	B71B	B719	B717	B715	B713	B711	B70F	B70D	B70B	B709
96	BA01	BA00	B9FD	B9FB	B9F9	B9F7	B9F5	B9F3	B9F1	B9EF	B9ED	B9EB	B9E9	B9E7	B9E5	B9E3	B9E1	B9DF	B9DD	B9DB
95	B9D3	B9D2	B9CF	B9CD	B9CB	B9C9	B9C7	B9C5	B9C3	B9C1	B9BF	B9BD	B9BB	B9B9	B9B7	B9B5	B9B3	B9B1	B9AF	B9AD
94	B9A5	B9A4	B9A1	B99F	B99D	B99B	B999	B997	B995	B993	B991	B98F	B98D	B98B	B989	B987	B985	B983	B981	B97F
93	B977	B976	B973	B971	B96F	B96D	B96B	B969	B967	B965	B963	B961	B95F	B95D	B95B	B959	B957	B955	B953	B951
92	B949	B948	B945	B943	B941	B93F	B93D	B93B	B939	B937	B935	B933	B931	B92F	B92D	B92B	B929	B927	B925	B923
91	B91B	B91A	B917	B915	B913	B911	B90F	B90D	B90B	B909	B907	B905	B903	B901	B8FF	B8FD	B8FB	B8F9	B8F7	B8F5
90	B8ED	B8EC	B8E9	B8E7	B8E5	B8E3	B8E1	B8DF	B8DD	B8DB	B8D9	B8D7	B8D5	B8D3	B8D1	B8CF	B8CD	B8CB	B8C9	B8C7
89	B8BF	B8BE	B8BB	B8B9	B8B7	B8B5	B8B3	B8B1	B8AF	B8AD	B8AB	B8A9	B8A7	B8A5	B8A3	B8A1	B89F	B89D	B89B	B899
88	B891	B890	B88D	B88B	B889	B887	B885	B883	B881	B87F	B87D	B87B	B879	B877	B875	B873	B871	B86F	B86D	B86B
87	B863	B862	B85F	B85D	B85B	B859	B857	B855	B853	B851	B84F	B84D	B84B	B849	B847	B845	B843	B841	B83F	B83D
86	B835	B834	B831	B82F	B82D	B82B	B829	B827	B825	B823	B821	B81F	B81D	B81B	B819	B817	B815	B813	B811	B80F
85	B807	B806	B803	B801	B7FF	B7FD	B7FB	B7F9	B7F7	B7F5	B7F3	B7F1	B7EF	B7ED	B7EB	B7E9	B7E7	B7E5	B7E3	B7E1
84	B7D9	B7D8	B7D5	B7D3	B7D1	B7CF	B7CD	B7CB	B7C9	B7C7	B7C5	B7C3	B7C1	B7BF	B7BD	B7BB	B7B9	B7B7	B7B5	B7B3
83	B7AB	B7AA	B7A7	B7A5	B7A3	B7A1	B79F	B79D	B79B	B799	B797	B795	B793	B791	B78F	B78D	B78B	B789	B787	B785
82	B77D	B77C	B779	B777	B775	B773	B771	B76F	B76D	B76B	B769	B767	B765	B763	B761	B75F	B75D	B75B	B759	B757
81	B74F	B74E	B74B	B749	B747	B745	B743	B741	B73F	B73D	B73B	B739	B737	B735	B733	B731	B72F	B72D	B72B	B729
80	B721	B720	B71D	B71B	B719	B717	B715	B713	B711	B70F	B70D	B70B	B709	B707	B705	B703	B701	B6FF	B6FD	B6FB
79	B6F3	B6F2	B6EF	B6ED	B6EB	B6E9	B6E7	B6E5	B6E3	B6E1	B6DF	B6DD	B6DB	B6D9	B6D7	B6D5	B6D3	B6D1	B6CF	B6CD

LINENUMBER (Y)

CONTROL BYTE

MEMORY-MAP MODE 4

COLOR BYTE

2

TWO BYTES NOT USED BY BASIC

TWO BYTES NOT USED BY BASIC

0- 8- 16- 24- 32- 40- 48- 56- 64- 72- 80- 88- 96- 104- 112- 120- 128- 136- 144- 152-

78	B6C5	B6C4	B6C1	B6BF	B6BD	B6BB	B6B9	B6B7	B6B5	B6B3	B6B1	B6AF	B6AD	B6AB	B6A9	B6A7	B6A5	B6A3	B6A1	B69F	B69D	B69B	B699
77	B697	B696	B693	B691	B68F	B68D	B68B	B689	B687	B685	B683	B681	B67F	B67D	B67B	B679	B677	B675	B673	B671	B66F	B66D	B66B
76	B669	B668	B665	B663	B661	B65F	B65D	B65B	B659	B657	B655	B653	B651	B64F	B64D	B64B	B649	B647	B645	B643	B641	B63F	B63D
75	B63B	B63A	B637	B635	B633	B631	B62F	B62D	B62B	B629	B627	B625	B623	B621	B61F	B61D	B61B	B619	B617	B615	B613	B611	B60F
74	B60D	B60C	B609	B607	B605	B603	B601	B5FF	B5FD	B5FB	B5F9	B5F7	B5F5	B5F3	B5F1	B5EF	B5ED	B5EB	B5E9	B5E7	B5E5	B5E3	B5E1
73	B5DF	B5DE	B5DB	B5D9	B5D7	B5D5	B5D3	B5D1	B5CF	B5CD	B5CB	B5C9	B5C7	B5C5	B5C3	B5C1	B5BF	B5BD	B5BB	B5B9	B5B7	B5B5	B5B3
72	B5B1	B5B0	B5AD	B5AB	B5A9	B5A7	B5A5	B5A3	B5A1	B59F	B59D	B59B	B599	B597	B595	B593	B591	B58F	B58D	B58B	B589	B587	B585
71	B583	B582	B57F	B57D	B57B	B579	B577	B575	B573	B571	B56F	B56D	B56B	B569	B567	B565	B563	B561	B55F	B55D	B55B	B559	B557
70	B555	B554	B551	B54F	B54D	B54B	B549	B547	B545	B543	B541	B53F	B53D	B53B	B539	B537	B535	B533	B531	B52F	B52D	B52B	B529
69	B527	B526	B523	B521	B51F	B51D	B51B	B519	B517	B515	B513	B511	B50F	B50D	B50B	B509	B507	B505	B503	B501	B4FF	B4FD	B4FB
68	B4F9	B4F8	B4F5	B4F3	B4F1	B4EF	B4ED	B4EB	B4E9	B4E7	B4E5	B4E3	B4E1	B4DF	B4DD	B4DB	B4D9	B4D7	B4D5	B4D3	B4D1	B4CF	B4CD
67	B4CB	B4CA	B4C7	B4C5	B4C3	B4C1	B4BF	B4BD	B4BB	B4B9	B4B7	B4B5	B4B3	B4B1	B4AF	B4AD	B4AB	B4A9	B4A7	B4A5	B4A3	B4A1	B49F
66	B49D	B49C	B499	B497	B495	B493	B491	B48F	B48D	B48B	B489	B487	B485	B483	B481	B47F	B47D	B47B	B479	B477	B475	B473	B471
65	B46F	B46E	B46B	B469	B467	B465	B463	B461	B45F	B45D	B45B	B459	B457	B455	B453	B451	B44F	B44D	B44B	B449	B447	B445	B443
64	B441	B440	B43D	B43B	B439	B437	B435	B433	B431	B42F	B42D	B42B	B429	B427	B425	B423	B421	B41F	B41D	B41B	B419	B417	B415
63	B413	B412	B40F	B40D	B40B	B409	B407	B405	B403	B401	B3FF	B3FD	B3FB	B3F9	B3F7	B3F5	B3F3	B3F1	B3EF	B3ED	B3EB	B3E9	B3E7
62	B3E5	B3E4	B3E1	B3DF	B3DD	B3DB	B3D9	B3D7	B3D5	B3D3	B3D1	B3CF	B3CD	B3CB	B3C9	B3C7	B3C5	B3C3	B3C1	B3BF	B3BD	B3BB	B3B9
61	B3B7	B3B6	B3B3	B3B1	B3AF	B3AD	B3AB	B3A9	B3A7	B3A5	B3A3	B3A1	B39F	B39D	B39B	B399	B397	B395	B393	B391	B38F	B38D	B38B
60	B389	B388	B385	B383	B381	B37F	B37D	B37B	B379	B377	B375	B373	B371	B36F	B36D	B36B	B369	B367	B365	B363	B361	B35F	B35D
59	B35B	B35A	B357	B355	B353	B351	B34F	B34D	B34B	B349	B347	B345	B343	B341	B33F	B33D	B33B	B339	B337	B335	B333	B331	B32F
58	B32D	B32C	B329	B327	B325	B323	B321	B31F	B31D	B31B	B319	B317	B315	B313	B311	B30F	B30D	B30B	B309	B307	B305	B303	B301
57	B2FF	B2FE	B2FB	B2F9	B2F7	B2F5	B2F3	B2F1	B2EF	B2ED	B2EB	B2E9	B2E7	B2E5	B2E3	B2E1	B2DF	B2DD	B2DB	B2D9	B2D7	B2D5	B2D3
56	B2D1	B2D0	B2CD	B2CB	B2C9	B2C7	B2C5	B2C3	B2C1	B2BF	B2BD	B2BB	B2B9	B2B7	B2B5	B2B3	B2B1	B2AF	B2AD	B2AB	B2A9	B2A7	B2A5
55	B2A3	B2A2	B29F	B29D	B29B	B299	B297	B295	B293	B291	B28F	B28D	B28B	B289	B287	B285	B283	B281	B27F	B27D	B27B	B279	B277
54	B275	B274	B271	B26F	B26D	B26B	B269	B267	B265	B263	B261	B25F	B25D	B25B	B259	B257	B255	B253	B251	B24F	B24D	B24B	B249
53	B247	B246	B243	B241	B23F	B23D	B23B	B239	B237	B235	B233	B231	B22F	B22D	B22B	B229	B227	B225	B223	B221	B21F	B21D	B21B
52	B219	B218	B215	B213	B211	B20F	B20D	B20B	B209	B207	B205	B203	B201	B1FF	B1FD	B1FB	B1F9	B1F7	B1F5	B1F3	B1F1	B1EF	B1ED
51	B1EB	B1EA	B1E7	B1E5	B1E3	B1E1	B1DF	B1DD	B1DB	B1D9	B1D7	B1D5	B1D3	B1D1	B1CF	B1CD	B1CB	B1C9	B1C7	B1C5	B1C3	B1C1	B1BF
50	B1BD	B1BC	B1B9	B1B7	B1B5	B1B3	B1B1	B1AF	B1AD	B1AB	B1A9	B1A7	B1A5	B1A3	B1A1	B19F	B19D	B19B	B199	B197	B195	B193	B191
49	B18F	B18E	B18B	B189	B187	B185	B183	B181	B17F	B17D	B17B	B179	B177	B175	B173	B171	B16F	B16D	B16B	B169	B167	B165	B163
48	B161	B160	B15D	B15B	B159	B157	B155	B153	B151	B14F	B14D	B14B	B149	B147	B145	B143	B141	B13F	B13D	B13B	B139	B137	B135
47	B133	B132	B12F	B12D	B12B	B129	B127	B125	B123	B121	B11F	B11D	B11B	B119	B117	B115	B113	B111	B10F	B10D	B10B	B109	B107
46	B105	B104	B101	B0FF	B0FD	B0FB	B0F9	B0F7	B0F5	B0F3	B0F1	B0EF	B0ED	B0EB	B0E9	B0E7	B0E5	B0E3	B0E1	B0DF	B0DD	B0DB	B0D9
45	B0D7	B0D6	B0D3	B0D1	B0CF	B0CD	B0CB	B0C9	B0C7	B0C5	B0C3	B0C1	B0BF	B0BD	B0BB	B0B9	B0B7	B0B5	B0B3	B0B1	B0AF	B0AD	B0AB
44	B0A9	B0A8	B0A5	B0A3	B0A1	B09F	B09D	B09B	B099	B097	B095	B093	B091	B08F	B08D	B08B	B089	B087	B085	B083	B081	B07F	B07D
43	B07B	B07A	B077	B075	B073	B071	B06F	B06D	B06B	B069	B067	B065	B063	B061	B05F	B05D	B05B	B059	B057	B055	B053	B051	B04F
42	B04D	B04C	B049	B047	B045	B043	B041	B03F	B03D	B03B	B039	B037	B035	B033	B031	B02F	B02D	B02B	B029	B027	B025	B023	B021
41	B01F	B01E	B01B	B019	B017	B015	B013	B011	B00F	B00D	B00B	B009	B007	B005	B003	B001	FFFF	AFDD	AFDB	AFD9	AFD7	AFD5	AFD3
40	AFD1	AFD0	AFED	AFEB	AFE9	AFE7	AFE5	AFE3	AFE1	AFDF	AFDD	AFDB	AFD9	AFD7	AFD5	AFD3	AFD1	AFCF	AFCD	AFCB	AFC9	AFC7	AFD5
39	AFD3	AFD2	AFBF	AFBD	AFBB	AFB9	AFB7	AFB5	AFB3	AFB1	AFAF	AFAD	AFAB	AFA9	AFA7	AFA5	AFA3	AFA1	AF9F	AF9D	AF9B	AF99	AF97
38	AF95	AF94	AF91	AF8F	AF8D	AF8B	AF89	AF87	AF85	AF83	AF81	AF7F	AF7D	AF7B	AF79	AF77	AF75	AF73	AF71	AF6F	AF6D	AF6B	AF69
37	AF67	AF66	AF63	AF61	AF5F	AF5D	AF5B	AF59	AF57	AF55	AF53	AF51	AF4F	AF4D	AF4B	AF49	AF47	AF45	AF43	AF41	AF3F	AF3D	AF3B
36	AF39	AF38	AF35	AF33	AF31	AF2F	AF2D	AF2B	AF29	AF27	AF25	AF23	AF21	AF1F	AF1D	AF1B	AF19	AF17	AF15	AF13	AF11	AF0F	AF0D
35	AF0B	AF0A	AF07	AF05	AF03	AF01	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF	AEEF
34	AEDD	AEDC	AED9	AED7	AED5	AED3	AED1	AECF	AECD	AECB	AEC9	AEC7	AEC5	AEC3	AEC1	AEBF	AEBD	AEBB	AEB9	AEB7	AEB5	AEB3	AEB1
33	AEBF	AEBE	AEBB	AEB9	AEB7	AEB5	AEB3	AEB1	AE9F	AE9D	AE9B	AE99	AE97	AE95	AE93	AE91	AE8F	AE8D	AE8B	AE89	AE87	AE85	AE83
32	AE81	AE80	AE7D	AE7B	AE79	AE77	AE75	AE73	AE71	AE6F	AE6D	AE6B	AE69	AE67	AE65	AE63	AE61	AE5F	AE5D	AE5B	AE59	AE57	AE55
31	AE53	AE52	AE4F	AE4D	AE4B	AE49	AE47	AE45	AE43	AE41	AE3F	AE3D	AE3B	AE39	AE37	AE35	AE33	AE31	AE2F	AE2D	AE2B	AE29	AE27
30	AE25	AE24	AE21	AE1F	AE1D	AE1B	AE19	AE17	AE15	AE13	AE11	AE0F	AE0D	AE0B	AE09	AE07	AE05	AE03	AE01	ADFF	ADFD	ADFB	ADF9
29	ADF7	ADF6	ADF3	ADF1	ADEF	ADED	ADEB	ADE9	ADE7	ADE5	ADE3	ADE1	ADDF	ADDD	ADDB	ADD9	ADD7	ADD5	ADD3	ADD1	ADCF	ADCD	ADCB
28	ADC9	ADC8	ADC5	ADC3	ADC1	ADBF	ADBD	ADBB	ADB9	ADB7	ADB5	ADB3	ADB1	ADAF	ADAD	ADAB	ADA9	ADA7	ADA5	ADA3	ADA1	AD9F	AD9D

LINENUMBER (Y)

CONTROL BYTE

MEMORY-MAP MODE 4

COLOR BYTE

3

TWO BYTES NOT USED BY BASIC

TWO BYTES NOT USED BY BASIC

0- 8- 16- 24- 32- 40- 48- 56- 64- 72- 80- 88- 96- 104- 112- 120- 128- 136- 144- 152-

27	AD9B	AD9A	AD97	AD95	AD93	AD91	AD8F	AD8D	AD8B	AD89	AD87	AD85	AD83	AD81	AD7F	AD7D	AD7B	AD79	AD77	AD75	AD73	AD71	AD6F	
26	AD6D	AD6C	AD69	AD67	AD65	AD63	AD61	AD5F	AD5D	AD5B	AD59	AD57	AD55	AD53	AD51	AD4F	AD4D	AD4B	AD49	AD47	AD45	AD43	AD41	
25	AD3F	AD3E	AD3B	AD39	AD37	AD35	AD33	AD31	AD2F	AD2D	AD2B	AD29	AD27	AD25	AD23	AD21	AD1F	AD1D	AD1B	AD19	AD17	AD15	AD13	
24	AD11	AD10	AD0D	AD0B	AD09	AD07	AD05	AD03	AD01	ACFF	ACFD	ACFB	ACF9	ACF7	ACF5	ACF3	ACF1	ACEF	ACED	ACEB	ACE9	ACE7	ACE5	
23	ACE3	ACE2	ACDF	ACDD	ACDB	ACD9	ACD7	ACD5	ACD3	ACD1	ACCF	ACCD	ACCB	ACC9	ACC7	ACC5	ACC3	ACC1	ACBF	ACBD	ACBB	ACB9	ACB7	
22	ACB5	ACB4	ACB1	ACAF	ACAD	ACAB	ACA9	ACA7	ACA5	ACA3	ACA1	AC9F	AC9D	AC9B	AC99	AC97	AC95	AC93	AC91	AC8F	AC8D	AC8B	AC89	
21	AC87	AC86	AC83	AC81	AC7F	AC7D	AC7B	AC79	AC77	AC75	AC73	AC71	AC6F	AC6D	AC6B	AC69	AC67	AC65	AC63	AC61	AC5F	AC5D	AC5B	
20	AC59	AC58	AC55	AC53	AC51	AC4F	AC4D	AC4B	AC49	AC47	AC45	AC43	AC41	AC3F	AC3D	AC3B	AC39	AC37	AC35	AC33	AC31	AC2F	AC2D	
19	AC2B	AC2A	AC27	AC25	AC23	AC21	AC1F	AC1D	AC1B	AC19	AC17	AC15	AC13	AC11	AC0F	AC0D	AC0B	AC09	AC07	AC05	AC03	AC01	ABFF	
18	ABFD	ABFC	ABF9	ABF7	ABF5	ABF3	ABF1	ABEF	ABED	ABEB	ABE9	ABE7	ABE5	ABE3	ABE1	ABDF	ABDD	ABDB	ABD9	ABD7	ABD5	ABD3	ABD1	
17	ABCF	ABCE	ABCB	ABC9	ABC7	ABC5	ABC3	ABC1	ABBF	ABBD	ABBB	ABB9	ABB7	ABB5	ABB3	ABB1	ABAF	ABAD	ABAB	ABA9	ABA7	ABA5	ABA3	
16	ABA1	ABA0	AB9D	AB9B	AB99	AB97	AB95	AB93	AB91	AB8F	AB8D	AB8B	AB89	AB87	AB85	AB83	AB81	AB7F	AB7D	AB7B	AB79	AB77	AB75	
15	AB73	AB72	AB6F	AB6D	AB6B	AB69	AB67	AB65	AB63	AB61	AB5F	AB5D	AB5B	AB59	AB57	AB55	AB53	AB51	AB4F	AB4D	AB4B	AB49	AB47	
14	AB45	AB44	AB41	AB3F	AB3D	AB3B	AB39	AB37	AB35	AB33	AB31	AB2F	AB2D	AB2B	AB29	AB27	AB25	AB23	AB21	AB1F	AB1D	AB1B	AB19	
13	AB17	AB16	AB13	AB11	AB0F	AB0D	AB0B	AB09	AB07	AB05	AB03	AB01	AAFF	AAFD	AAFB	AAF9	AAF7	AAF5	AAF3	AAF1	AAEF	AAED	AAEB	
12	AAE9	AAE8	AAE5	AAE3	AAE1	AADF	AADD	AADB	AAD9	AAD7	AAD5	AAD3	AAD1	AACF	AACD	AACB	AAC9	AAC7	AAC5	AAC3	AAC1	AABF	AABD	
11	AABB	AABA	AAB7	AAB5	AAB3	AAB1	AAAF	AAAD	AAAB	AAA9	AAA7	AAA5	AAA3	AAA1	AA9F	AA9D	AA9B	AA99	AA97	AA95	AA93	AA91	AA8F	
10	AA8D	AA8C	AA89	AA87	AA85	AA83	AA81	AA7F	AA7D	AA7B	AA79	AA77	AA75	AA73	AA71	AA6F	AA6D	AA6B	AA69	AA67	AA65	AA63	AA61	
9	AA5F	AA5E	AA5B	AA59	AA57	AA55	AA53	AA51	AA4F	AA4D	AA4B	AA49	AA47	AA45	AA43	AA41	AA3F	AA3D	AA3B	AA39	AA37	AA35	AA33	
8	AA31	AA30	AA2D	AA2B	AA29	AA27	AA25	AA23	AA21	AA1F	AA1D	AA1B	AA19	AA17	AA15	AA13	AA11	AA0F	AA0D	AA0B	AA09	AA07	AA05	
7	AA03	AA02	A9FF	A9FD	A9FB	A9F9	A9F7	A9F5	A9F3	A9F1	A9EF	A9ED	A9EB	A9E9	A9E7	A9E5	A9E3	A9E1	A9DF	A9DD	A9DB	A9D9	A9D7	
6	A9D5	A9D4	A9D1	A9CF	A9CD	A9CB	A9C9	A9C7	A9C5	A9C3	A9C1	A9BF	A9BD	A9BB	A9B9	A9B7	A9B5	A9B3	A9B1	A9AF	A9AD	A9AB	A9A9	
5	A9A7	A9A6	A9A3	A9A1	A99F	A99D	A99B	A999	A997	A995	A993	A991	A98F	A98D	A98B	A989	A987	A985	A983	A981	A97F	A97D	A97B	
4	A979	A978	A975	A973	A971	A96F	A96D	A96B	A969	A967	A965	A963	A961	A95F	A95D	A95B	A959	A957	A955	A953	A951	A94F	A94D	
3	A94B	A94A	A947	A945	A943	A941	A93F	A93D	A93B	A939	A937	A935	A933	A931	A92F	A92D	A92B	A929	A927	A925	A923	A921	A91F	
2	A91D	A91C	A919	A917	A915	A913	A911	A90F	A90D	A90B	A909	A907	A905	A903	A901	ABFF	ABFD	ABFB	ABF9	ABF7	ABF5	ABF3	ABF1	
1	ABEF	ABEE	ABEB	ABE9	ABE7	ABE5	ABE3	ABE1	ABDF	ABDD	ABDB	ABD9	ABD7	ABD5	ABD3	ABD1	ABCF	ABCD	ABCB	ABCB	ABC9	ABC7	ABC5	ABC3
0	ABC1	ABC0	ABBD	ABBB	ABB9	ABB7	ABB5	ABB3	ABB1	ABAF	ABAD	ABAB	ABA9	ABA7	ABA5	ABA3	ABA1	AB9F	AB9D	AB9B	AB99	AB97	AB95	

LINENUMBER (Y)

```

900 REM THE PROGRAM TO PRINT THIS TABLE
1000 PRINT CHR$(15);:REM CONDENSED ON MX-80
1010 FOR LZ=0 TO 129:X%=#BFEB-46*LZ:LN%=129-LZ
1020 IF LNZ<100 THEN PRINT " ";
1030 IF LNZ<10 THEN PRINT " ";
1040 PRINT LN%; " ";
1050 FOR YZ=0 TO 1:PRINT HEX$(X%-YZ); " ";:NEXT
1060 FOR YZ=X%-4 TO X%-44 STEP -2
1070 PRINT HEX$(YZ); " ";:NEXT:PRINT :NEXT

```

- 1 - Word Order in Sentences with Auxiliary Verbs
- 2 - The Indefinite and the Definite Article
- 3 - The Simple Present (einfache Zeitform Gegenwart)
- 4 - Das "s" in der 3. Person Singular
- 5 - Word Order of Adverbs of Indefinite Time
- 6 - Possessive Adjectives (besitzanzeig. Fuerwoerter)
- 7 - The Cardinal Numbers (Grundzahlen)
- 8 - Negative Questions (Verneinte Fragen)
- 9 - Word Order in "yes or no questions"
- 10 - Negative Sentences (Verneinte Aussage)
- 11 - Adverbials of Time and Place (Wortstellung)
- 12 - Questions with Question words and "do-does"
- 13 - Berufsbezeichnungen mit "a"
- 14 - Prepositions in Adverbials
- 15 - What colour...What a...

- 1 - Fragen und Kurzwantworten mit >did<
- 2 - Verneinung mit >did<
- 3 - Fragen mit Fragewoertern und >did<
- 4 - Hoefliche Fragen
- 5 - Bindewoerter
- 6 - >some< und >any<
- 7 - Die Ordnungszahlen
- 8 - Das Datum
- 9 - >something/anything< >somebody/anybody<
- 10 - >can/cannot< >could/could not<
- 11 - Das Umstandswort
- 12 - Kurzfragen
- 13 - >stop smoking< >start cooking<
- 14 - >they taste wonderful<
- 15 - Der Bedingungssatz
- 16 - Relativsaetze
- 17 - Verhaeltniswoerter in Umstandsbestimmungen

- 1 - "you"
- 2 - Future with "going to"
- 3 - "certainly"
- 4 - Comparison of Adjectives with "er/est"
- 5 - Comparing Things
- 6 - The "s" and the "of" Genetive
- 7 - Present Perfect (Regular Forms)
- 8 - Word Order Present Perfect
- 9 - Present Perfect (Irregular Forms)
- 10 - "whose"
- 11 - "must not / need not / may"
- 12 - Past Tense (Regular Forms)
- 13 - "was / were" and "had"

- 1 - "is" (Long Form and Short Form)
- 2 - Short Answers
- 3 - The s-Genetive (Besitzfall)
- 4 - The Interrogative Pronouns (Fragefuerwoerter)
- 5 - The Personal Pronouns (persoenlichen Fuerwoerter)
- 6 - "have got" (Long Form and Short Form)
- 7 - Present Continuous (Verlaufsform)
- 8 - Expressions of Quantity with "of" (Mengenangaben)
- 9 - The Imperative and the Negative Imperative
- 10 - Time by the clock (Uhrzeit)
- 11 - The Plural of Nouns (Mehrzahl der Hauptwoerter)
- 12 - "there is / there are"
- 13 - Demonstrative Adjectives "this-that-these-those" (hinweisende Fuerwoerter)
- 14 - Auxiliary Verbs "can / must" (Hilfszeitwoerter)



?  
 BUONGIORNO  
 SALUT BUENOS DIAS OLA  
 BONJOUR GOOD MORNING  
 GRUETZI SALVE BONSOIR  
 GUTEN TAG



SOME REMARKS ABOUT CASSETTE INTERFACING

1. DAI cass. output level more high

In the DAInamic-newsletter of april 1981 the DA1pc cassette-interface was first published (page 45). The scheme probably was based on DA1po versions delivered before spring 1981. In the newer versions (REV.4 on main board) the 1k2-resistor of the output circuit is replaced by an 2k7-resistor (fig.1). This brings the output on a higher signal-level, though the output-impedance is rising too.

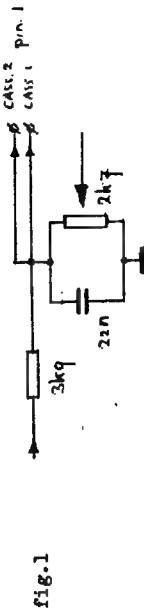


fig.1

2. Deminishing of motor-control current and indicating motor-running

It may happen that the cassette-motor does not stop at the end of a LOAD- or SAVE-procedure or a BREAK despite usage of remote control. This is caused by the motorcurrent which is to be switched by a relais in the DA1pc (RL 1 and RL 2 according to Dessart, DAInamic page 207). DA1 uses different types for the relais: CELDUC, HAMLIN and maybe others. In stead of replacing the present relais by types for higher-power-switching a small and lower-cost operation on the low-cost cassette-recorder might be the solution. Figure 2 shows a circuit which will be present, at least with great resemblance, in most low-cost cassette-recorders.

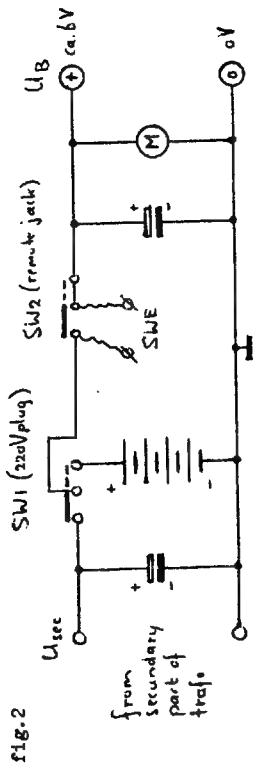


fig.2

SW 1 is switching the power source between batteries and 220 V and is set by the 220 V-plug. SW2 is set open by plugging in the remote-jack. Power-control is then handed over to the external switch SWE, which is, in case of direct connecting, the DAI-relais. The current to be switched by the relais can be reduced by building in a quasi-power-stabilisation circuit into the recorder according to fig 3. At the same time a LED can be inserted for visual signalling motor-running.

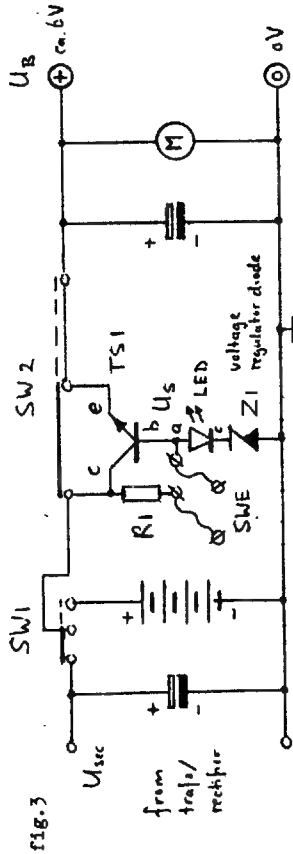
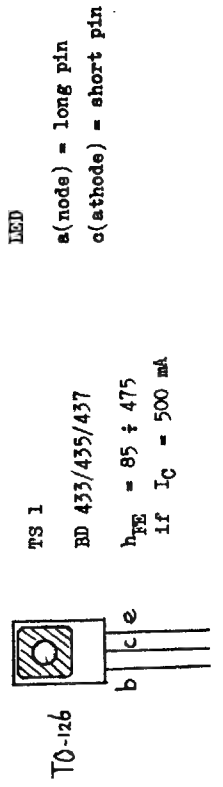


fig.3

Care should be taken in choosing the values of Z1 and RL. U<sub>G</sub> is not allowed to sink too much, as in that case the motor-speed will go under the necessary value. In an Audio Somic CT-226 recorder a choice of 10 Ω for R1 and 4V7 for Z1 with a red LED did suffice. This, however, only if the LS-stage-power-usage, which is dependant on the set output-volume, is not maximal. Notice that the stabilisation was chosen to be not complete, because of too much loss of voltage at U<sub>G</sub>. The reduction in current to be switched by the DAI-relais depends on the h<sub>FE</sub>-factor of the used transistor and U<sub>S</sub>. Some experimenting with values will most likely result in a good solution.

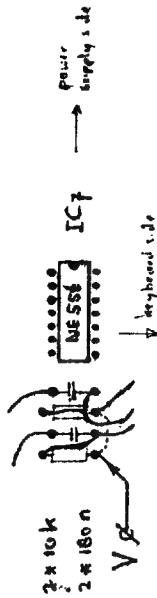
fig.4 component details



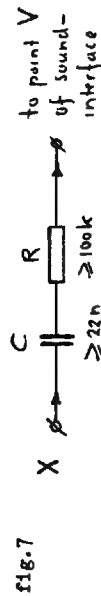
2. Sound from cassette on TV-loudspeaker

For searching the precise start or end of a certain file on cassette, it would be convenient to be able to hear the sound coming from the cassette-recorder without disconnecting the external-speaker (ear phone) plug. This feature is realised through a simple operation in the DA1pc. Figure 5 shows the sound- and noise-interface of the DA1-pc. The end of the circuit exists of 3 opamps of IC 14. One of them (IC 14a) provides for a sound-output for the video-signal, which consists of a mix of the two stereo-channels. Point V is the place where the sound from the cassette-recorder safely can be mixed with other signals for the TV-loudspeaker. On the main board of the DA1pc this point is found left of IC 7 (NE 556). Figure 6 explains the position of this point.

fig.6 top/front view of main board at the back side

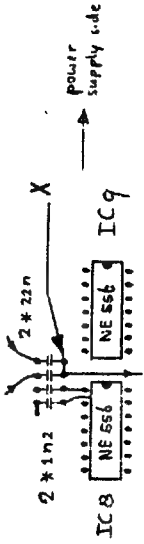


The cassette-sound-signal can be taken from the input-signal in the cassette-interface (DAInamic newsletter page 45), or from a cassette-signal-clipper (e.g. the one suggested by DA1-club Eurocontrol Beek page 129). Both methods can be realised by means of a connection according to fig. 7.



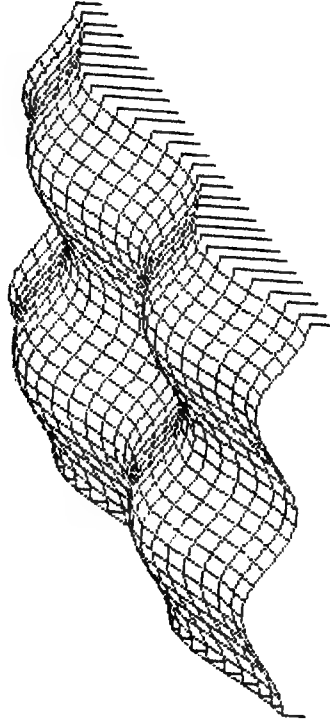
If no signal-clipper is used, point X can be found on the main board somewhat backside of the middle between IC8 and IC9 (fig.8).

fig.8 top/front view of main board at the back side



In case of usage of the Eurocontrol-clipper point X can be the collector of the BC 108-transistor. A resistor (e.g. 220 Ω) must be inserted between the + 12 V-line and X. Here too some experimenting with values of the resistors will bring about the objected goal: cassette-sound on the TV-loudspeaker with a reasonable volume.

Fred de Jong  
Dermigstraat 23  
6371 VX Schaesberg  
the Netherlands







TELEX IN BASIC

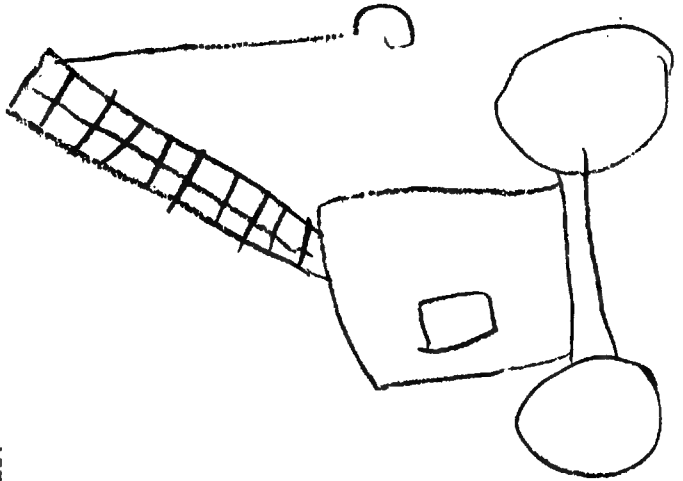
```

1  MODE 0:PRINT CHR$(12)
2  COLORT 7 0 7 7
3  PRINT "DATA INLEZEN....."
4  GOSUB 195:REM POKE ASCII LABEL
5  PRINT CHR$(12)
6  ENVELOPE 0 15
7  BD=45.45:REM BAUD RATE
8  SOUND 0 0 1 0 FREQ(BD*16.0)
9  P%=1.0
10 POKE #FE03,#BB:REM CONTROL WORD
11 FOR I=0.0 TO 12.0:POKE (#BFEF-#B6*I),#SA:NEXT
12 CURSOR 0,21:PRINT "TELEX IN BASIC"
13 PRINT :PRINT "~~~~~";
14 POKE #BEE2,#D1
15 CURSOR 0,17:PRINT " 45.45 BAUD"
16 POKE #BDD6,#D0
17 WAIT TIME 100
18 IF PEEK(#FE02)<>16 THEN 65:REM TEST INPUT.
19 PRINT CHR$(12)
20 IF PEEK(#FE02)<>16 THEN 100
21 D7=PEEK(#FE01):REM GET BAUDOT CHAR.
22 IF D7=27.0 THEN P%=0.0:GOTO 100:REM FIGS
23 IF D7=31.0 THEN P%=1.0:GOTO 100:REM LETS
24 IF D7=2.0 THEN 100:REM LF
25 PRINT CHR$(B%(P%,D%));
26 IF PEEK(#FE02)<>16 THEN 100:GOTO 170:REM TEST INPUT
27 GOTO 100
28 CLEAR 500
29 DIM B%(1,0,31.0)
30 FOR X%=0.0 TO 31.0:READ A%B%(0.0,X%)=A%:NEXT X%
31 FOR X%=0.0 TO 31.0:READ A%B%(1.0,X%)=A%:NEXT X%
32 PRINT
33 RETURN
34 DATA 0,#33,0,#2D,#20,#27,#38,#37,#0D,#24,#34,#27
35 DATA #2C,#21,#3A,#28,#35,#2B,#29,#32,#23,#36,#30
36 DATA #31,#39,#3F,#26,#2E,#2F,#3B,0
37 DATA 0,#45,0,#41,#20,#53,#49,#55,#0D,#44,#52,#44
38 DATA #4E,#46,#43,#48,#54,#5A,#4C,#57,#48,#59,#50
39 DATA #51,#4F,#42,#47,0,#4D,#58,#56,0

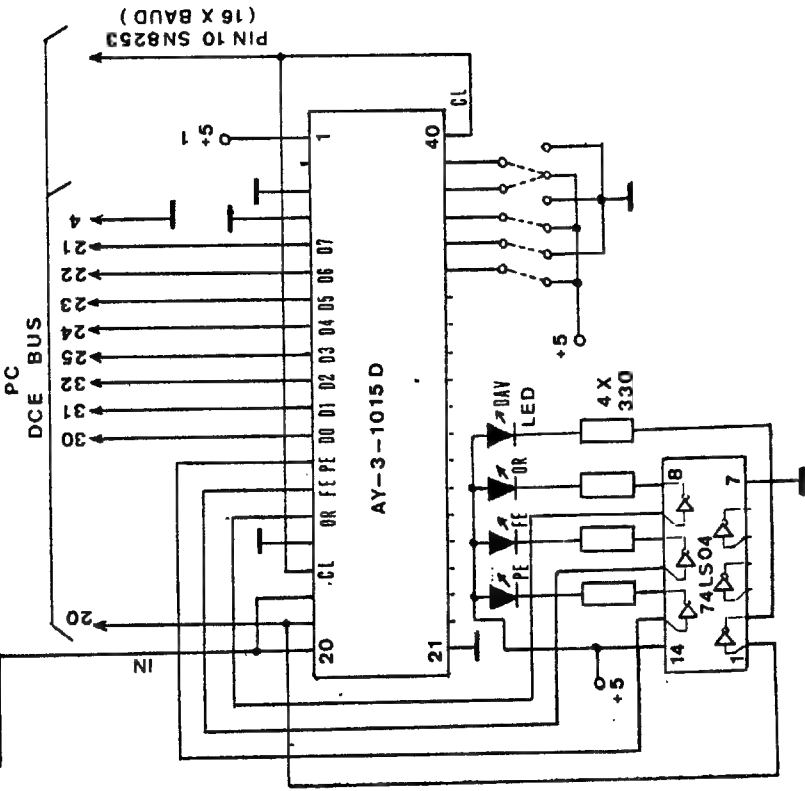
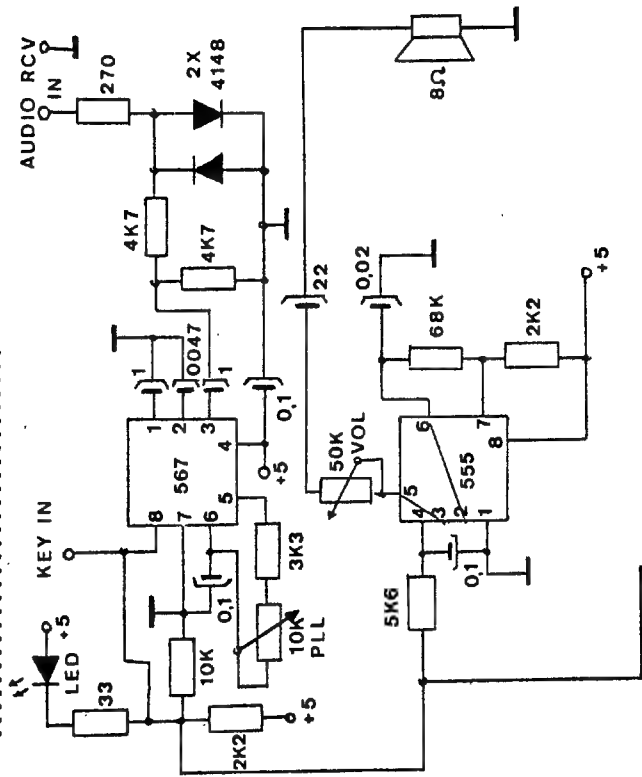
```

The RS232-parallel interface card is available (assembled & tested) at the price of 1500 Bfr (+ 700 Bfr for cables & connectors). Please contact A.De Dauw for detailed information:

A.De Dauw  
Wallehof 93  
2770 Nieuwkerken-Waas  
BELGIUM tel : 031/770676

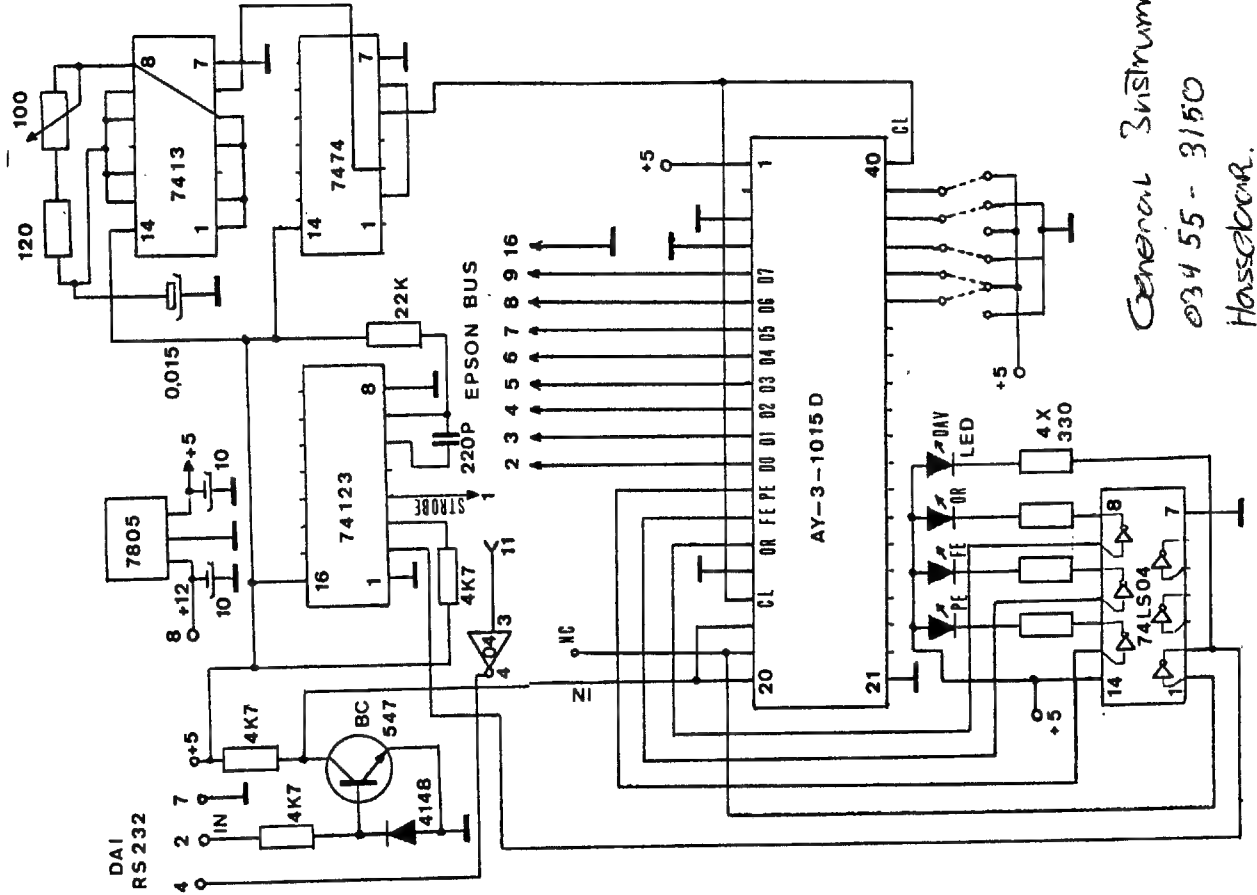


RTTY - CONVERTOR



DAI-RS232 SERIAL TO EPSON-PARALLEL INTERFACE CARD

- No software required
- Up to 9600 baud



General Instruments  
 03455 - 3150  
 Hasselbar.



Klaus Hoffmann  
Tulbeckstr.17  
8000 München 2  
Tel.089/505610

Liebe DAINAMIC-Redaktion

als Antwort auf Eure Frage auf Seite 154 sende ich Euch ein Assemblerprogramm zur Screencopy auf MX 80/II. Für Ausgabe in 960-Punkt-Mode muß #4B in Location #331 durch #4C ersetzt werden.

Ich könnte eine deutsche Übersetzung von DAINAMIC jeweils ca. 2-3 Wochen nach Erscheinen der holländischen Ausgabe erstellen, falls Interesse bei den deutschsprachigen Lesern besteht.

300 C5	PUSH 0	378 2C	INR L
301 D5	PUSH 2	379 77	MOV M;A
302 E5	PUSH 4	37A 3A AB 3	WTLO:LDA OUTP
303 F5	PUSH 6	37D BD	CMP L
304 F3	DI	37E CA 7A 3	JZ WTLO
305 3A 40 0	LDA #40	381 22 AC 3	SHLD INP
308 F5	PUSH M	384 3E D	MVI A;#D
309 E6 3F	ANI #3F	386 32 3 FE	STA #FE03
30B F6 B0	ORI #B0	389 E1	POP H
30D 32 40 0	STA #40	38A C9	RET
310 32 6 FD	STA #FD06	38B F5	INTSV;PUSH M
313 FB	EI	38C 2A AB 3	LHLD OUTP
314 21 0 0	LXI H;0	38F 7C	MOV A;H
317 4C	MOV C;H	390 BD	CMP L
318 CD 84 EB	CALL #E8B4	391 C2 9C 3	JNZ PUT
31B 3E F9	MVI A;#F9	394 3E C	MVI A;#C
31D 80	ADD B	396 32 3 FE	STA #FE03
31E E6 FB	ANI #FB	399 C3 A7 3	JMP INTEND
320 C6 7	ADI 7	39C 2C	PUT;INR L
322 4F	MOV C;A	39D 26 4	MVI H;4
323 13	INX D	39F 7E	MOV A;M
324 D5	LINE:PUSH D	3A0 32 0 FE	STA #FE00
325 C5	PUSH B	3A3 7D	MOV A;L
326 3E D	MVI A;#D	3A4 32 AB 3	STA OUTP
328 CD 74 3	CALL BFOUT	3A7 F1	INTEND;POP M
32B 3E 1B	MVI A;#1B	3AB E1	POP H
32D CD 74 3	CALL BFOUT	3A9 FB	EI
330 3E 4B	MVI A;#4B	3AA C9	RET
332 CD 74 3	CALL BFOUT	3AB 0	OUTP;DATA1 0
335 7B	MOV A;E	3AC 0 4	INP;DATA2 #400
336 CD 74 3	CALL BFOUT	3AE F5	INIT;PUSH M
339 7A	MOV A;D	3AF E5	PUSH H
33A CD 74 3	CALL BFOUT	3B0 21 8B 3	LXI H;INTSV
33D 3E 1	LOPC;MVI A;1	3B3 22 66 0	SHLD #66
33F F5	LOOP:PUSH M	3B6 21 74 3	LXI H;BFOUT
340 CD 84 EB	CALL #E8B4	3B9 22 DE 2	SHLD #2DE
343 D1	POP D	3BC 3E A0	MVI A;#A0
344 DA 5A 3	JC NEXT	3BE 32 3 FE	STA #FE03
347 FE B	CPI B	3C1 3E FF	MVI A;#FF
349 7A	MOV A;D	3C3 32 1 FE	STA #FE01
34A 17	RAL	3C6 3E C3	MVI A;#C3
34B D	DCR C	3C8 32 DD 2	STA #2DD
34C D2 3F 3	JNC LOOP	3CB 21 0 5	LXI H;#500
34F EE FF	XRI #FF	3CE 22 9B 2	SHLD #29B
351 CD 74 3	CALL BFOUT	3D1 E1	POP H
354 C1	POP B	3D2 F1	POP M
355 C5	PUSH B	3D3 C9	RET
356 23	INX H		
357 C3 3D 3	JMP LOPC	SYMBOL TABLE	
35A C1	NEXT;POP B	INP 3AC	PUT 39C
35B 79	MOV A;C	LOOP 33F	INIT 3AE
35C D6 B	SUI B	OUTP 3AB	BFOUT 374
35E 4F	MOV C;A		
35F 21 0 0	LXI H;0	WTLO 37A	INTSV 38B
362 D1	POP D	LOPC 33D	LINE 324
363 D2 24 3	JNC LINE	INTEND 3A7	NEXT 35A
366 F1	POP M		
367 F3	DI		
368 32 40 0	STA #40	2EC F5	PUSH PSW
36B 32 6 FD	STA #FD06	2ED 3A 2 FE	LDA FE02
36E FB	EI	2F0 A7	AA A ANA A
36F F1	POP 6	2F1 F2 ED 2	JP 2ED
370 E1	POP 4	2F4 F1	POP PSW
371 D1	POP 2	2F5 32 0 FE	STA FE00
372 C1	POP 0	2FB C9	RET Next Byte:FF
373 C9	RET		
374 E5	BFOUT;PUSH H		
375 2A AC 3	LHLD INP		

## 1. Anschluss des MX80 (Centronics-Schnittstelle) an den DCE-Bus

	MX80	DCE
(19)	STROBE	C7 (17)
(20)	DATA1	A0 (16)
.....		
(27)	DATA8	A7 (15)
(28)	ACKNLG	C6 (18)

Der 8255 wird in MODE 1 betrieben: POKE #FE03,#A0. Port B und ein Teil der Leitungen von Port C bleiben frei.

Es gibt zwei Ansteuerungsmöglichkeiten:

- Polling:** Hierbei wartet das Ausgabeprogramm, bis der Output-buffer leer ist (Bit 7 von Port C low), dann wird das Byte in das Ausgaberegister geschrieben. Diese Technik wird im Programmbeispiel 1 angewendet.
- Interruptcontrol:** Die Ausgaberroutine BFOUT speichert die Daten in einem Circular Buffer (FIFO) mit 255 Bytes. Die Interruptserviceroutine INTSV gibt die Daten aus, während der Rechner weiterrechnet, wodurch sich ein größerer Throughput ergibt. Für Interruptcontrol wird der EXINTR (Stack) verwendet. Dazu muß Bit C3 (DCE 29) mit EXINTR (DCE 6) über eine Logikschaltung oder einen Schalter verbunden (beim Power-on oder Reset muß EXINTR low bleiben!).

## 2. Beschreibung des Screencopy - Programmes:

Location #300-303 und #36F-372: Register save

Location #304-313 und #366-36E: Memory Bank Switch für Screen - ROM und Restore.

Nun geht's los: In #318 werden XMAX und YMAX abgefragt, in #318-325 YMAX auf ein Vielfaches von 8 abgerundet und mit XMAX auf dem Stack gesaved. In #326-33A werden die Steuerzeichen für hochauflösende Graphik ausgegeben. Die Schleife LOOP (#33F-34C) sammelt 8 Bit von der Screen (Abbruch durch Shift to Carry), CPI 8 und XRI #FF (#34F) bestimmen, welche Farben ausgedruckt werden. Die Ausgabezeilen werden durch 'OFF SCREEN' (#344 JC NEXT) beendet, die gesamte Subroutine durch Overflow von Y (#35C und #363).

Die Routine BFOUT (#374) speichert die Daten in den Circular buffer, updated den Pointer INP und setzt das Interrupt-enable Bit des 8255.

INP=OUTP bedeutet Buffer empty, INP+1=OUTP Buffer full. Die Interruptserviceroutine entnimmt die Characters aus dem Buffer und cleart das Int.enable-Bit if Buffer empty.

Die Hilfsroutine INIT (#3AE) initialisiert die Interruptvektoren, Hardware und die Schnittstelle für Print und List (POKE #131,3:LIST:POKE #131,1). Aufruf: CLEAR 1000 UT,R (read SCRCOPY in),B,CALLM #3AE,CLEAR xxx.

Das kurze Basicprogramm zeigt den Aufruf von SCRCOPY. Der Drucker arbeitet etwas ungleichmäßig, weil der Rechner langsamer als der Drucker ist.

```
5   REM PROGRAMMBEISPIEL 1: PRINTERSTEUERUNG UEBER DCE-BUS OHNE INTERRUPTS
10  POKE #29B,#0:POKE #29C,3:CLEAR 1000
30  FOR I=#2EC TO #2F8:READ H:POKE I,H:NEXT
40  POKE #FE03,#A0:POKE #FE01,#FF
50  POKE #2DD,#C3:POKE #2DE,#EC:POKE #2DF,2
60  DATA #F5,#3A,2,#FE,#A7,#F2,#ED,2,#F1,#32,0,#FE,#C9

29000  REM PROGRAMM FUER SCREENCOPY
30000  MODE 6A: DRAW 0,0 XMAX,YMAX 15
30005  DRAW 0,YMAX XMAX,0 15
30010  POKE #131,3:PRINT CHR#(#1B)+"A"+CHR#(8)
30020  CALLM #300:PRINT CHR#(#1B)+"2":POKE #131,1
```



```

5 REM GRAFIC 'THE HAT'
7 REM r.corswandt 9/81
8 REM
10 MODE 6:COLORG 12 Ø 15 Ø
12 REM HORIZT. POS. - VERT. POS.
20 P=165.Ø : Q=14Ø.Ø
30 XP=144.Ø : XR=1.5*3.14159
40 YP=56.Ø : YR=1.Ø : ZP=64.Ø
50 XF=XR/XP : YF=YP/YR : ZF=XR/ZP
60 FOR ZI = -Q TO Q-1.Ø
70 IF ZI < (-ZP) OR ZI > ZP GOTO 15Ø
80 ZT=ZI*XP/ZP : ZZ=ZI
90 XL=INT (Ø.5+SQR(XP*XP-ZT*ZT))
100 FOR XI = -XL TO XL
110 XT=SQR(XI*XI+ZT*ZT)*XF : XX=XI
120 YY=(SIN(XT)+Ø.4*SIN(3.Ø*XT))*YF
130 GOSUB 17Ø
140 NEXT XI
150 NEXT ZI
160 GOTO 16Ø : REM ENDE
170 X1=XX+ZZ+P
180 Y1=YY-ZZ+Q
190 DOT X1,Y1 15
200 IF Y1=Ø GOTO 22Ø
210 DOT X1,Y1-1 Ø
220 RETURN

```

Ein kleines Grahic-Programm "DER HUT"  
 Um eine anderen Schatteneffekt zu erzeugen können schwarz und weiß in Zeile Nr.190 und Nr. 210 vertauscht werden.  
 Drei zusätzliche Grahic-Figuren kann man erhalten, wenn die beiden SINUS-Funktionen in Zeile Nr.120 durch COS ersetzt werden. Verändert man nur eine SIN-Funktion ist es nötig XP,YP und ZP etwas zu verkleinern 144 auf 124 ,56 auf 46 usw. da die Figur sonst zu groß werden würde.

Rainer Corswandt ,Lüdenscheid,Deutschland/W

Le programme suivant permet de réaliser l'effacement de l'écran du DAI en pressant simplement sur la touche "TAB". Appelé tous les 20ms par l'Interrupt 7, il cherche si dans les 4 octets qui contiennent les derniers codes donnés par le contrôle du clavier, il y a un 9, c.à d. le ASCII-Code pour "TAB". Si tel est le cas, il change cet octet en hex.1Ø (pour éviter une seconde exécution, hex.1Ø étant un code "inoffensif"), puis il efface l'écran par RST 5/Ø3 avec hex.ØC dans l'accumulateur.  
 Comme ceci pose le curseur sur la position Ø de la 23e ligne, il est nécessaire de presser "RETURN" avant de commencer à introduire des commandes ou des lignes de programme en BASIC.

Listing en Assembler:

```

3ØØ F5 PUSH PSW
3Ø1 E5 PUSH H
3Ø2 D5 PUSH D
3Ø3 3E Ø9 MVI A,Ø9 ASCII "TAB"
3Ø5 21 BA Ø2 LXI H,KLIND cf. NEWSLETTER N°6
3Ø8 16 Ø4 MVI D,Ø4 Compteur
3ØA BE BOUCLE CMP M ((HL))=9?
3ØB CA 19 Ø3 BOUCLE CA 19 Ø3
3ØE 23 INX H
3ØF 15 DCR D
31Ø G2 ØA Ø3 JNZ BOUCLE
313 DL SORTIE POP D
314 EI POP H
315 FI POP PSW
316 C3 A9 D9 JMP INTR7 Routine normale
319 36 1Ø MVI M,1Ø ASCII:Touche ↑
31B 3E ØC MVI A,ØC ASCII-Code pour effacement
31D EF RST 5
31E Ø3 DATA Ø3
31F C3 13 Ø3 JMP SORTIE

```

Symboles: BOUCLE : 3ØA KLIND : 2BA  
 SORTIE : 313 INTR7 : D9A9  
 EFFACE : 319

En langage machine:

```

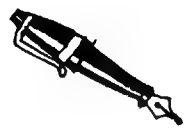
Ø3ØØ F5 E5 D5 3E Ø9 21 BA Ø2 16 Ø4 BE CA 19 Ø3 23 15
Ø31Ø C2 ØA Ø3 D1 E1 F1 C3 A9 D9 36 1Ø 3E ØC EF Ø3 C3
Ø32Ø 13 Ø3

```

Cette fonction est activée en mode UTILITY par:  
 >V7 D9A9-3ØØ2

N.B.: Contrairement à la solution proposée au numéro 7 par M.Dessart, celle-ci est aussi applicable en UTILITY.

Claude Pommerell



BESTE DAINAMIC VRIENDEN,

BIJGAAND DOE IK JULLIE TOEKOMEN EEN DCR CASSETTE MET DAAROP :

1. DEZE BRIEF.
2. HET DAI DATA BASE PROGRAM MANUAAL.
3. HET DAI DATA BASE PROGRAMMA.

IEDER VAN BOVENSTAANDE ITEMS IS TWEE KEER OP DE TAPE VASTGELEGD ZODAT INLEZEN HOPELIJK ZONDER PROBLEMEN ZAL VERLOPEN.

MIJN ERVARINGEN MET DE DCR ZIJN 100% UITSTEKEND. HET IS EEN VERADEMING MET DIT MEDIUM TE KUNNEN WERKEN VANWEGE SNELHEID ( VERBIJSTEREND WANNEER MEN DE GEMONE COMPACT CASSETTE GEWEND IS ) HET VOLLEDIG ONTBREKEN VAN LOADING EN SAVING ERRORS EN DE VOLLEDIGE SOFTWARE BESTURING.

MIJN WELGEMEENDE COMPLIMENTEN AAN MEMOCOM ! SLECHTS EEN ENKEL GELUID VAN KRITIEK : HEXFILES KUNNEN NIET INGELEZEN WORDEN VANAF COMPACT CASSETTES WANNEER DE DCR AAN DE DAI IS GEKOPPELD MAAR VERGEN ENIGE SOFTWARE AANPASSINGEN.

DE BASICODE UITZENDING VAN HOBBIYSCOOP HEB IK OP COMPACT CASSETTE OPGENOMEN EN ZONDER PROBLEMEN INGELEZEN. PRACHTIG DAT DE DAI GEEN HARDWARE MODIFICATIES BEHOEFT VOOR BASICODE ZOALS ALLE ANDERE PERSONAL COMPUTERS.

BESTUDERING VAN DATABUS ARTIKEL INZAKE BASICODE MAAKT MIJ ERG OPTIMISTISCH OVER DE DAI SOFTWARE TOEKOMST. HET ARTIKEL OVER DE DAI IN DE MINI/MICRO COMPUTER GELEZEN HEBBENDE LIJKT HET MIJ ZINVOEL VANUIT DAINAMIC COMMENTAAR TE ZENDEN NAAR DIT TIJDSCHRIFT O.A. OVER DCR ONTWIKKELING.

ZELF HEB IK BELANGSTELLING VOOR EEN DAI MANUAAL MET MEER INFORMATIE OVER HARD/SOFTWARE EN DE ZINSNEDEN IN VOORNOED BLAD DAT IN SEPTEMBER 1981 EEN NIEUW MANUAAL UITKOMT ROEPT BIJ MIJ DIRECT VRAGEN OP WAAROM DAINAMIC HIEROVER (NOG) NIETS PUBLICEERT.

GAARNE REAKTIE IN DAINAMIC.

MOCHT BIJGAANDE SOFTWARE IN AANMERKING KOMEN VOOR DE DAINAMIC SOFTWARE BIBLIOTHEEK DAN ONTVANG IK GRAAG DE TOOLKIT MET RENUMBER MOGELIJKHEID. TEVENS HEB IK BELANGSTELLING VOOR WORD-PROCESSOR PROGRAMMA WAARVAN IK HOOP DAT IN EEN VAN DE KOMENDE DAINAMICS MEER INFORMATIE OVER MOGELIJKHEDEN VAN DIT PROGRAMMA TE VINDEN ZULLEN ZIJN.

BIJ CONVERSIE VAN MIJN COMPACTCASSETTE SOFTWARE NAAR DCR FORMAAT IS MIJN FGT SOFTWARE DE MIST INGEGAAN (LOWER CASE FGT INCLUSIEF ) ZODAT IK GRAAG NOG EEN KOPIE ONTVANG ( WELKE KOSTEN ? )

IK ZIE MET SPANNING UIT NAAR DE VOLGENDE DAINAMIC OMDAT IK ELKE LETTER VAN JULLIE UITSTEKENDE BLAD TWEE KEER SPEL. HOUDEN ZO !!!

HOPELIJK TOT ZIENS IN UTRECHT EN DE GROETJES

HARRY VAN COOTEN

*a dcr cassette is smaller, Harry!*

*no problems!*

*we agree.*

*only if the file is from #298...., in that case :*

```
*CAS
*UT
>5296 02-00
>R
>Z5
>GF2F2
and save on dcr
with:
>4298 xxx
```

*the article was just a translation from "Computing Today.."*

*we didn't see it!*

*Toolkit is on his way...*

*...on his way, no costs*

*We think one is enough*

*see you soon, Harry! Wilfried*

POKE-ACTION

DAI-BASIC offers wonderful DOT, DRAW & FILL commands. This makes programming graphics very easy, but to illustrate that one can do fine tricks with POKE in the VIDEO-RAM, I wrote the following lines of program. To assist you in this way of graphics-programming, we publish the MEMORY-MAP of MODE 4.

Next time we will work out MODE 2 and/or MODE 6. You can use these maps for the 16-color modes, but then you have to poke into 2 bytes to illuminate some dots. We POKE in ODD addresses, this has result on COLOR register COLOR X X X X.

The action-subroutine is in front of the program (3-6) to gain speed. Please compare the program in INTEGER & FLOATING POINT!

- 20-40 : creatures in position
- 50-70 : one creature down
- 74-78 : sorry, I couldn't stop him....
- 80-100 : back to your position (too high)
- 110-120 : back in the row!
- 130 : clear screen, again....

```
1 REM POKE-ACTION w. hermans *** IMP INT ***
2 GOTO 10:REM SKIP ACTION ROUTINE
3 POKE X%,238:POKE X%+46,34:POKE X%+92,34
4 POKE X%+138,34:POKE X%+184,58:POKE X%+230,255
5 POKE X%+276,223:POKE X%+322,127:POKE X%+368,60:POKE X%+414,0
6 RETURN
10 MODE 4:COLOR 8 0 0 0
20 FOR X%=#B9D7 TO #B9FF STEP 4
30 GOSUB 3
40 NEXT
50 FOR X%=#B9FB TO #ABBB STEP -46
60 GOSUB 3
70 NEXT
74 NOISE 0 15
75 FOR X%=#A99F TO #AB99 STEP -46:POKE X%,#80:WAIT TIME 2
76 POKE X%,0:NEXT:POKE #AB99,#80
78 WAIT TIME 10:SOUND OFF
80 FOR X%=#ABBB TO #BB99 STEP 46
90 POKE X%-46,0:GOSUB 3
100 NEXT
110 FOR X%=#BB99 TO #B9FB STEP -46
120 GOSUB 3:NEXT
130 WAIT TIME 50:MODE 1:GOTO 10
```

WHO WILL WIN THE MATHCHIP AM9511 ?

This list contains the articles/authors of DAInamic Newsletter 80/81.

You can assign 10 points to 3 articles/authors.

- eg : Peeters J. 6 points assume 253 points
- Camby JC 3 points assume 199 points
- Bonny M. 1 points assume 56 points

Assigning more than 10 points is not legal.

Send your votes to DAInamic before 1 jan 1982.

You can even win your own prize by assuming the results in exact order, close enough to the total points of the winners.

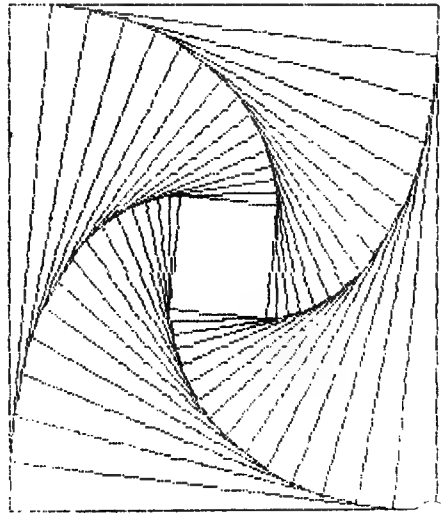
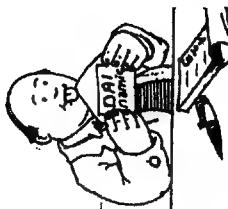
All entries can be controlled on our next meeting. (in april?)

the competitors:

Issue	page	article/author	
0/1	4	GETALCONVERSIES	J. VERDONK
0/1	7	DCE-BUS	F. DE RAEDT
0/1	11	VIER OP FFFN RIJ	W. HERMANS
0/1	15	RAKLISPEL	H. BAKKER
0/1	21	4 COLOR DEMO	W. HERMANS
0/1	24	CITROEN	W. HERMANS
0/1	28	PADDLE-EVENT	F. DE RAEDT
2	5	VARIABLEN IN DAI-BASIC	W. HERMANS
2	14	DIGITAAL-ANALOOG CLOCK	H. VAN COOTEN
2	17	FORMAT-LISTING	W. HERMANS
2	18	COMPUTER-MAGAZINES	J. SCHEPENS
2	29	LIST-PRINT	J. DESSART
3	8	VERKENNINGSTOCHTEN IN ML	W. HERMANS
3	12	BLUE MOON	G. UYTTERHOLVEN
3	22	TUBULAR BELLS	R. SIP
3	25	VIDEO-TEXT	H. VAN COOTEN
3	30	VARIABLEN ATLAS	F. DE RAEDT
3	32	CENTRONICS-INTERFACE	CATTIAERT/DE DAUW
3	35	BARRICADE	F. DRUYFF
4	46	FLASH	F. DE RAEDT
4	47	PRIME NUMBERS	B. VAN ROMPAEY
4	67	STORY OF FGT	W. HERMANS
5	83	DAI AAN DE MONITOR	H. BAKKER
5	99	TIMING OF MATH-CHIP	J. SCHEPENS
5	110	TIMERS	HERMANS/DE RAEDT
5	116	GROSSEN ZAHL/STATISTIEK	MEYER

5	118	FASING KEYBOARD	W. HERMANS
5		DATA STATEMENTS GENERATOR	I. BROEKMAN
5		POWER-ON INITIALISATION	J. BOERRIGTER
5		128 CASSETTE CLIPPERS	W. DE LEEUW
			C. DE BONT
			Co Beek
6	133	BASIC CALL OF FGT/BASIC LOAD	ASSINK D.
6	138	BOOK REVIEW	I. BROEKMAN
6	139	WCC LAUSANNE	P. V. D. HIJZEN
6	141	MEMORY MAP	J. BOERRIGTER
6	151	WILHELMUS	F. DRUIJFF
6	152	TUERME VON HANOI	MEYSTRE A.
6	156	VICJDF GRAADS POLYNOMEN	C. VAN DIJK
7	180	8080 SIMULATOR	W. HERMANS
7	189	HFAP-STORY	J. BOERRIGTER
7	190	INTEGERS/FPT IN DAI-FORMAT	B. VAN ROMPAEY
7	205	BOOK REVIEW	I. BROEKMAN
7	205	16 COLORS IN 4-COLOR MODE	J. MOL
7	206	IMP STR	T. KREBS
7	207..	KEYBOARD PCB-LAYOUT	J. DESSART
7	211	TALK EDITOR	SIP/HERMANS
7	214	CIJFFERTABEL	C. VAN DIJK KAMPEN

You can also vote for programs from collection tapes and for all articles from NEWSLETTER 8. LET THE BEST WIN....



```

5 REM si Je
10 REM hoord en neuw
12 COLORG 0 15 0 0
15 MODE 6
20 R=60.0:FOR X=-R TO R:Y=SQR(R*R-X*X)
30 DOT X+XMAX/2,Y+YMAX/2 15:DOT X+XMAX/2,-Y+YMAX/2 15:NEXT
40 FOR X=-6.0 TO 6.0:Y=SQR(6.0*6.0-X*X):DRAW X+167,Y+127 X+167,
C -Y+127 15:NEXT
50 FOR X=0.0 TO 30.0:Y=SQR(30.0*30.0-X*X):DOT X/2+150,-Y+150 15:
C NEXT
55 FOR X=-30.0 TO 0.0:Y=SQR(30.0*30.0-X*X):DOT X/2+185,-Y+150
C 15:NEXT

60 REM rechter oor
70 FOR X=-30.0 TO 0.0:Y=SQR(30.0*30.0-X*X)
80 DOT X/2+107,-Y+YMAX/2 15:NEXT

90 REM linker oor
100 FOR X=0.0 TO 30.0:Y=SQR(30.0*30.0-X*X)
110 DOT X/2+227,-Y+YMAX/2 15:NEXT

120 REM haar
130 FOR X=3.0*PI/2.0 TO 5.0*PI/2.0 STEP 0.1:S=80.0
140 FOR X1=3.2 TO 4.2 STEP 0.2
150 DOT XMAX/2+S*SIN(X)*X1/4.0,YMAX/2+S*COS(X)*X1/4.0 15
160 NEXT:NEXT

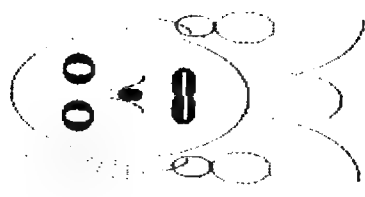
190 REM ogen
200 FOR X=-15.0 TO 15.0:Y=SQR(15.0*15.0-X*X)
210 DRAW X+145,Y/2+153 X+145,-Y/2+153 15
220 DRAW X+190,Y/2+153 X+190,-Y/2+153 15:NEXT
240 FOR X=-6.0 TO 6.0:Y=SQR(6.0*6.0-X*X)
250 DRAW X+145,Y+153 X+145,-Y+153 22
260 DRAW X+190,Y+153 X+190,-Y+153 0:NEXT
270 DOT 145,153 15:DOT 190,153 15

280 REM mond
290 FOR X=-15.0 TO 15.0:Y=SQR(15.0*15.0-X*X)
292 DRAW X+155,Y/2+100 X+155,-Y/2+100 22
294 DRAW X+179,Y/2+100 X+179,-Y/2+100 22
300 DRAW X+155,Y/2.5+100 X+155,-Y/2.5+100 15
310 DRAW X+179,Y/2.5+100 X+179,-Y/2.5+100 15:NEXT
320 FOR X=0.0 TO 1.0:DRAW 145,98+X 190,99+X 0:NEXT

330 REM oorringen
350 FOR R=0.0 TO 1.35 STEP 1.35:FOR X=0.0 TO 2.0*PI STEP 0.1:S=
C 10.0
360 DOT 100+100*R+S*SIN(X),95+S*COS(X) 15
370 DOT 100+100*R+S*SIN(X)*3.0/2.0,70+S*COS(X)*3.0/2.0 15
380 NEXT:NEXT

390 REM hals
400 FOR X=0.0 TO 60.0:Y=SQR(60.0*60.0-X*X)
410 DOT X+90,-Y+70 15:NEXT
420 FOR X=-60.0 TO 0.0:Y=SQR(60.0*60.0-X*X)
430 DOT X+245,-Y+70 15:NEXT
440 FOR X=-25.0 TO 25.0:Y=SQR(25.0*25.0-X*X)
460 DOT X+167,-Y+45 15:NEXT
500 PRINT CHR$(12):CURSOR 23,2:PRINT "S I E N T J E "
510 COLORG 0 15 15 0:WAIT TIME 30:COLORG 0.15 0 0:WAIT TIME 200
520 GOTO 510

```



```

5 REM letter from J. Marchand
10 PRINT CHR$(12):CLEAR 1000:COLORT 1 14 1 1:POKE #75,32:T%=
C #47
12 ENVELOPE 0 15:FOR R%=0 TO 6:READ D$:NEXT R%
15 FOR P%=0 TO 7:POKE #BF69-#86*P%,T%:NEXT
17 PRINT "FROM"
19 POKE #BE59,68:POKE #BE57,65
20 PRINT "I DAI"
22 PRINT " TO"
23 PRINT " TO"
25 PRINT
26 POKE #BC41,68:POKE #BC3F,65
27 PRINT "I DAI":PRINT
29 FOR LZ=0 TO 20
31 CURSOR 0,19:PRINT CHR$(92);:WAIT TIME 2:CURSOR CURX-1,CURY:
C 1 PRINT CHR$(32);
32 PRINT " ";CHR$(47);:WAIT TIME 2:CURSOR CURX-1,CURY:PRINT
C 1 CHR$(32)
33 CURSOR 0,17:PRINT CHR$(47);:WAIT TIME 2:CURSOR CURX-1,CURY:
C 1 PRINT CHR$(32);
34 PRINT " ";CHR$(92);:WAIT TIME 2:CURSOR CURX-1,CURY:PRINT
C 1 CHR$(32):CURSOR 0,15
36 GOSUB 540:NEXT
37 FOR T%=47 TO #3F STEP -1:FOR P%=0 TO 7:POKE #BF69-#86*P%,
C T%:NEXT:CURSOR 0,23:WAIT TIME 2:NEXT T%
39 SOUND OFF :WAIT TIME 50:PRINT CHR$(12):COLORT 7 0 7 7:A$=
C "Can I get:C$="(Y or N)"
40 B$="Okay then send me ":GOSUB 500
41 PRINT TAB(15);"Velbert 16.9.81":PRINT
42 PRINT TAB(2);"Beste Dainamic-specialisten !"
44 PRINT TAB(2);"Thank you for your quick answer."
46 PRINT :PRINT TAB(2);"But I don't know"
47 PRINT TAB(2);"Which programmes I may get.":CURSOR 2,CURY-4:
C PRINT "Go on by typing any KEY !"
48 G%-GETC:IF G%=0 GOTO 48:PRINT CHR$(12):GOSUB 500
49 RESTORE
50 READ P$:IF P$="NO" GOTO 110
55 PRINT TAB(2);A$;" "P$;" "C$;:INPUT " ";X$;:PRINT
60 IF X$="Y" THEN PRINT :GOTO 100
65 GOTO 50
100 PRINT :PRINT TAB(2);P$;" "P$;" "":GOTO 140
110 PRINT :PRINT TAB(2);B$;"what you like !"
120 PRINT TAB(2);"(No programmes from NEWSLETTERS)"
140 PRINT TAB(2);"Thank you very much."
150 PRINT TAB(2);"With kind regards"
152 PRINT TAB(10);"Yours sincerely"
154 PRINT TAB(16);"Jean Marchand"
160 CURSOR 2,CURY-2:END
200 DATA "FGT + G4","FGT + half of G4","FGT","G4","half of G4",
C "G1","NO"
500 FOR P%=0 TO 14:POKE #BF69-#86*P%,#67:NEXT:RETURN
540 READ F,W:IF F=0 AND W=0 THEN SOUND OFF :RETURN
550 SOUND 0 0 15 0:FREQ(F):SOUND 2 0 15 1 FREQ(2.0*F):SOUND 1 0
C 15 2 FREQ(4.0*F):WAIT TIME W:RETURN
600 DATA 198,1,151,1,209,1,198,1,313,1,264,9,0,0
610 DATA 198,1,151,1,209,1,198,1,352,1,297,9,0,0
620 DATA 396,1,313,1,264,1,297,1,247,1,264,9,0,0

```



## AKOESTISCH SIGNAAL BIJ EINDE LOAD EN SAVE OPDRACHTEN.

Door de schakeling uit fig 1 in de computer te bouwen, wordt de gebruiker dmv. een akoestisch signaal erop attent gemaakt dat de computer klaar is met een LOAD of SAVE opdracht. Er wordt gebruik gemaakt van een IC met 2 monostabiele multivibrators (MVB), nl de SN 74LS123 (zie fig 2). In fig 1 is MVB nr 1 gebruikt voor cassette 1. Wil men ook een akoestisch signaal voor cassette 2 dan moet MVB nr 2 met dezelfde componenten overeenkomstig als bij MVB nr 1 aangesloten worden.

### De werking:

Door de CLEAR en de B inputs (pin 3 en 2) via R4 aan de +5V te lassen is de schakeling zodanig ingesteld, dat de MVB triggert op een neersaande flank (van HOOG naar LAAG) welke aangeboden wordt op de A input (pin 1). In rust is de A input LAAG omdat de A input via R3 aan de GND ligt. Volgt er nu een LOAD of SAVE opdracht voor cassette 1 dan komt relais 1 in en wordt de A input HOOG. Dit HOOG worden van input A geschied echter met een zekere tijdsvertraging om ongewenste triggering door contactdender van het relais te voorkomen. R1 en C3 zorgen voor deze tijdsvertraging. Aan het einde van een LOAD of SAVE opdracht voor cassette 1 schakelt relais 1 weer in de rusttoestand. De condensator C3 zal zich nu ontladen via R3. Wanneer de spanning over C3 het LAAG niveau bereikt heeft, triggert de MVB. De Qnot uitsans (pin 14) - welke in rust HOOG is - wordt sedurende de alarmtijd LAAG en de buzzer geeft een akoestisch signaal. De timings componenten C1 en R2 zorgen voor een alarmtijd van ongeveer 1 seconde. De buzzer mag niet meer dan 16 mA trekken als hij aangestuurd wordt omdat anders de uitsans van het IC te zwaar belast wordt. Wil men een buzzer gebruiken die meer stroom trekt, dan kan deze aangesloten worden zoals in fig 3.

### De inbouw:

Het geheel aan componenten past op een experimenteerprintje van 3 bij 3 cm. Op de computerprint zit tussen de CAS 2 Plus en de PDL 1 Plus een gat. Hierop kan het printje met een (plastic) afstandsbusje gemonteerd worden. De buzzer kan met dubbelzijdig plakband geplakt worden op de RS 232 connector. De +5V is te vinden op de zekering naast de voeding (zie fig 4). De GND kan aangesloten worden op een van de schroeven v.h. deksel v.d. voeding. De relais aansluitingen kunnen afgetapt worden aan de achterkant v.d. CAS 1 Plus (pin 2 en 3).

Aan te raden is de schakeling alleen in te bouwen wanneer men geen gebruik maakt van de remote aansluiting van de cassette recorder, omdat anders de voedingsspanningen v.d. computer en de cassetterecorder met elkaar in aanraking komen. NB: Wanneer de computer ingeschakeld wordt geeft de buzzer ook een akoestisch signaal.

11-10-1981

Th. Wanders  
Molenlaan 16  
2394 AS Hazerswoude NL

### PADDLE ROUTINE IN MACHINE LANGUAGE

```
MVI A,x          x= paddle nr 0...5
CALL : EBC6
LDA  : DB        value 0...255 from software acc
```

1. +12 V aansluiten op punt 1 van de video plug, via 'n schakelaar.  
( +12 V is aanwezig op het printje waar de videoplug op gemonteert is. Connector punt 5 A 7).
2. Video signaal van de computer aansluiten op punt 2 van de t.v. videoplug (via coax kabel).
3. Afscherming van de coax kabel aansluiten op punt 3 van de t.v. video plug.

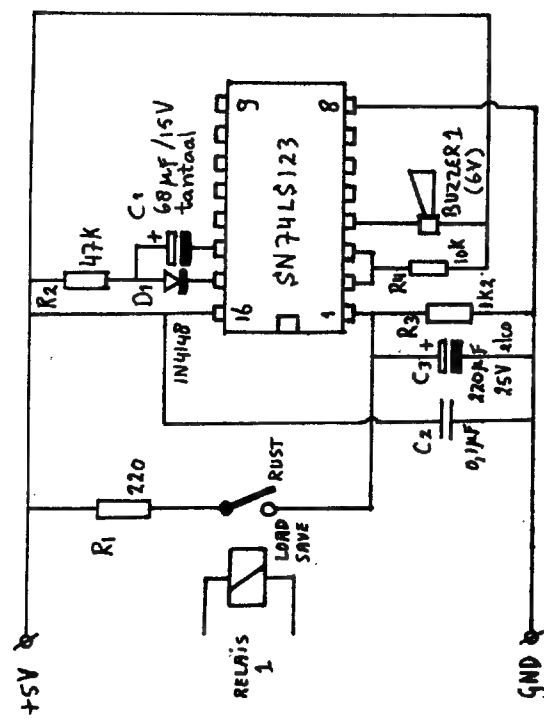


FIG. 1

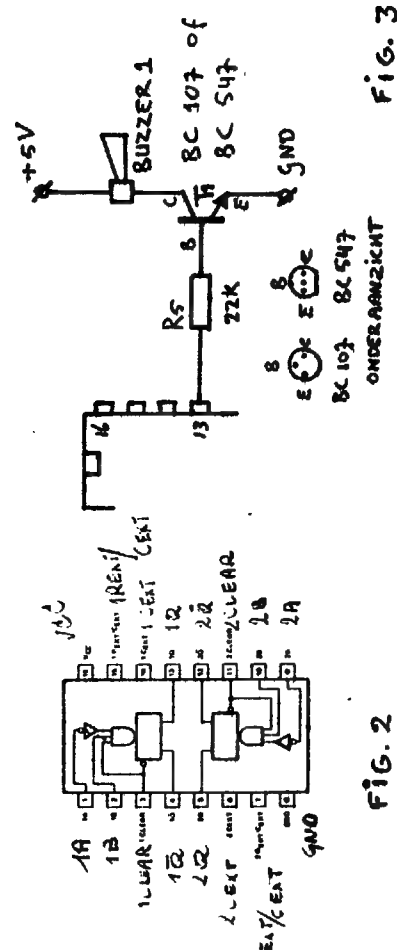


FIG. 2

FIG. 3

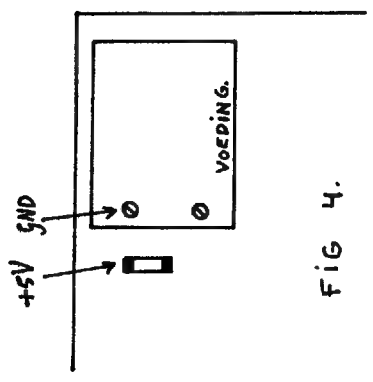
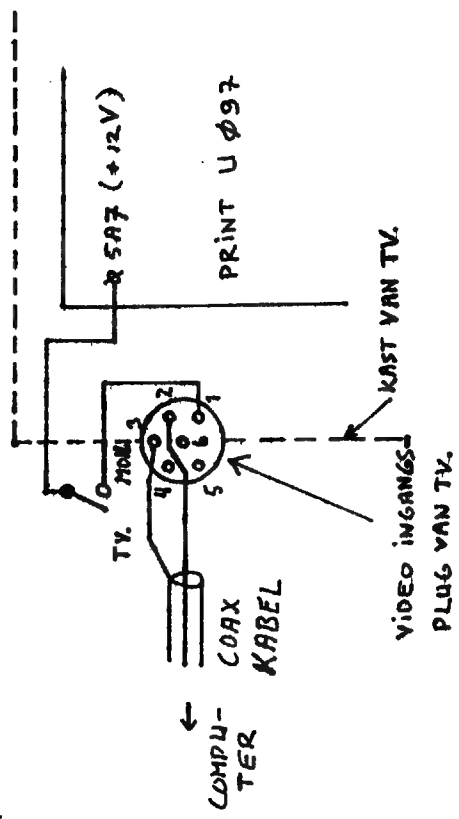


FIG. 4.



11 Oktober 1981

```

10  MODE 6:COLORG B 0 0 0
20  AX=0:AY=0:BX=XMAX:BY=YMAX
90  FOR X=1 TO 32
100  DRAW AX,AY BX,AY 21
110  DRAW BX,AY BX,BY 21
120  DRAW BX,BY AX,BY 21
130  DRAW AX,BY AX,AY 21
140  AX=AX+8:AY=AY+8:BX=BX-8:BY=BY-8
150  NEXT
160  GOTO 160
    
```

Th. Wanders  
Molenlaan 16  
2394 AS Hazerswoude NL.

```

10 REM COMPLEMENTEN EN SUPPLEMENTEN
11 REM UIT CREATIVE COMPUTING SEPT 1980
12 REM AUTEUR R.CARLSON, USA
65 PRINT CHR$(12)
67 CLEAR 3000
70 DIM R(100),A$(8,0),B$(10,0),C$(12,0),Q$(3,0)
80 PRINT :PRINT " Dit programma helpt bij het oplossen
C van "
90 PRINT :PRINT " wiskundeproblemen met complementen
C "
91 PRINT :PRINT " en supplementen . "
100 PRINT :PRINT " Typ JA in als je met de computer wilt
C werken. "
110 PRINT :PRINT " of NEE als je een aantal opgaven wilt
C "
119 PRINT
120 INPUT Q$
130 IF LEFT$(Q$,1)="J" THEN T=1.0
131 IF LEFT$(Q$,1)<>"J" THEN T=0.0
132 IF T=0.0 GOTO 150
139 PRINT
140 PRINT :INPUT " Hoeveel sommen wil je ? ":N
150 PRINT CHR$(12)

160 REM DEF FNA(A,B,C)=(A+B*C*90)/(B+1)
170 REM DE FUNCTIE A ZAL DE VERGELIJKING OPLOSSEN DIE
171 REM DE WAARDEN VAN A,B,C GEBRUIKT.
180 FOR M=1.0 TO N
190 A=INT(60.0*RNDR(M))+1.0
200 IF RND(M)<0.5 THEN A=-A

210 REM A GAAT VAN -60 TOT +60
220 B=INT(4.0*RNDR(M))+1.0

230 REM B=1,2,3,4
240 C=INT(2.0*RNDR(M))+1.0

250 REM D=1 OF 2 VOOR HET COMPLEMENT OF HET SUPPLEMENT
260 R(M)=(A+B*C*90.0)/(B+1.0)

270 REM R(M) BEVAT HET HUIDIGE ANTWOORD
280 IF C=1.0 THEN C$=" complement ":IF C<>1.0 THEN C$="
C supplement "
290 IF SGN(A)=1.0 THEN A$=" meer "
291 IF SGN(A)<1.0 THEN A$=" minder "
300 ON B GOTO 310,330,350,370
310 B$=" "
320 GOTO 380
330 B$=" tweemaal "
340 GOTO 380
350 B$=" driemaal "
360 GOTO 380
370 B$=" viermaal "
380 PRINT :PRINT " Som ";M; ". ";
389 PRINT

```

```

390 PRINT :PRINT " Een hoek is ";ABS(A);" graden ";A$;
391 PRINT :PRINT " dan ";B$;" zijn ";C$;" ".:PRINT
400 PRINT :PRINT " Wat is de hoek ? "
410 PRINT
420 K=0.0
430 IF T<>1.0 THEN 680
440 PRINT
450 PRINT :INPUT " Typ nu de uitkomst in ! ":A1
460 PRINT
470 IF ABS(A1-R(M))>1E-3 THEN 510
479 PRINT
480 PRINT " Inderdaad, het antwoord is ";R(M)
490 PRINT
500 GOTO 680
510 IF A1<>C*90.0-R(M) THEN 550
519 PRINT
520 PRINT " Probeer de ";C$
530 GOTO 450

540 REM K IS VOOR VOORTGANGSCONTROLE
550 K=K+1.0
560 ON K GOTO 570,590,620,650
570 PRINT :PRINT " Weet je, dat hoeken ";C$;" zijn. ":;
C PRINT CHR$(8);" als ze samen ";C*90.0;" zijn . "
580 GOTO 450
590 PRINT :PRINT " Probeer deze vergelijking eens . "
600 PRINT :PRINT " X= ";B; "(";C*90.0; "-X) +";A
610 GOTO 450
620 PRINT :PRINT " Je vergelijking vereenvoudigt dit : "
630 PRINT :PRINT B+1.0;"X= ";B/C*90.0;" + ";A
640 GOTO 450
650 PRINT :PRINT " Het juiste antwoord is ";R(M)
660 PRINT :PRINT " Probeer nog een probleem. "
670 PRINT
679 IF T=0.0 GOTO 150
680 NEXT M
690 PRINT
700 PRINT
710 IF T=1.0 THEN 770
720 PRINT :INPUT " Wil je de antwoorden ?":Q$
730 IF LEFT$(Q$,1)="N" THEN PRINT " Typ dan de uitkomst in
C ! "
735 INPUT A1
740 FOR M=1.0 TO N
750 PRINT :PRINT " Som ";M; ". Het antwoord is ";R(M);
C graden. "
755 PRINT :PRINT " Is er veel verschil tussen de twee
C uitkomsten ? "
760 NEXT M
770 END
775 PRINT :PRINT

```

## RESTART ROUTINES IN THE DAI PC.

1. The 8080 microprocessor in the DAI knows 8 instruction codes that are one-byte CALL instructions: RST 0 through RST 7. In many computer systems, these instructions are used in combination with interrupts. This article will describe the way in which the DAI uses these interrupts.
2. RST 1, RST 4, RST 5.

These 3 restart instructions are used for bank switching. In the memory area E000-EFFF the DAI uses 4 banks of each 4K ROM 'in parallel'. Via bits 6 and 7 of output port FDC6, one of these banks is selected. Normally, bank 0 is switched on, but via software instructions one of the other banks can be activated. Therefore, the RST 1, RST 4 and RST 5 instruction codes are used. These instructions are followed by one data byte.

When the program counter encounters one of these RST instructions, it goes to the interrupt vector routines in the area C000-C03F. The interrupt vector address from the area 0062-0071 is loaded, and the program counter is set to this address.

The routines which are found on the vector addresses prepare the selection of the required ROM bank:

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

Via the general ROM bank switching routine on address C6CF the selected ROM bank is activated.

The data byte after the RST instruction indicates which address in the particular ROM bank has to be jumped to. It is an offset to the startaddress E000.

Example: RST 5, data 18: ROM bank 2, address E018.  
There a jump to the screen mode changing routine can be found.

When switching to another ROM bank, the previous selection is saved in memory. On return from the switched bank, the old bank select is restored again.
3. The other Restart instructions.

All other RST instructions are used on interrupt base. The interrupts are generated by the timers in the 5501 Timer and Interrupt controller.
4. RST 7, Clock interrupt.

The 20 ms page blanking signal for the TV is used as clock signal. Each time this clock interrupt is present, the program counter is set on 0038. Via the interrupt vector routine, the program counter is set on address D9A9.

The RST 7 routine on this address enables only stack interrupts and checks the contents of timer 01BE/F. Each time when a RST 7 interrupt is present, this counter is decremented. As long as it is not zero, nothing happens.

When this timer is zero, then on each RST 7 interrupt the clock timer 01C0 is decremented. Again, nothing happens when it is not zero.

But when the clock timer is also zero, a RST 5, data 12 routine is activated.

This routine flashes the cursor according the information in the pointers 0074-0077 (see memory map). After changing the contents of the screen location pointed by the cursor, the old interrupt mask is restored and the program returns from interrupt to its normal sequence.

### 5. RST 6, Keyboard interrupt service.

Each time an interrupt from timer 4 is present, the program counter is set to D578 via the interrupt vector routine on address 0030.

The RST 6 routine reloads timer 4 and enables only clock and stack interrupts.

The keyboard counter 01C1 is decremented on each RST 6 interrupt. When the result is not zero, the routine is aborted. Else, the keyboard counter is reloaded and a keyboard scan is performed (the GETC routine).

On exit, the original interrupt mask is restored again.

### 6. RST 3, Sound interrupt.

On an interrupt from timer 3, the interrupt vector routine on address 0018 load D755 into the program counter.

This RST 3 routine enables clock and sound interrupts only. Timer 3 is reloaded and ROM bank 1 is selected.

Now the program continues on address EE6E in bank 1, which is the Sound program.

On exit, the old ROM bank and the old interrupt mask is restored again.

### 7. RST 2, Stack interrupt.

When stack overflow occurs, an RST 2 interrupt is the result. Via address 0010 in the interrupt vector routine area, the program counter is loaded with D9E2.

The RST 2 routine resets the stackpointer on F900. The running of inputs and the encoding of stored lines is disabled. The input is returned to the keyboard and the timers for sound and keyboard interrupts are reloaded.

Then the error messages 'STACK OVERFLOW' is printed.

### 8. RST 0, Utility.

The RST 0 interrupt is used only by the Utility program. On this interrupt, the program counter is set on C000.

The vector address, required in this interrupt vector routine, is only present after a Z2 or a Z3 command in utility. Then location 0062/63 is loaded with EB5D, the startaddress of the RST 0 routine in ROM bank 3.

The RST 0 interrupt is caused by timer 0; it is used in the LOOK routine in utility.

On a RST 0 interrupt, all CPU registers are saved in the RAM area 0053-005E. Then the program continues on a address which is given by the LOOK routine and indicates the next instruction to be performed.

The program checks this instruction. If it is a CALL or a RST instruction, then the next address is saved too.



Then a check is performed to see if the next instruction address is within the frame given by the LOOK window. When the result is positive, the contents of all registers, including stack pointer, flags and program counter, is displayed on the screen. On exit, the timer 0 is reloaded, the interrupt mask set and - among other instructions - the CPU registers are restored again.

Because the program runs now under RST 0 interrupts, it runs much slower than in normal runtime!

Jan Boerrigter - okt. '81

#### ADDITIONAL INFORMATION ON MEMORY MAP

After working out the whole Utility program (12K on EA00-EFFF in ROM bank 3), the following updates on the memory map are available.

0045-0046 Not used.  
 0047 Used to store EI/DI instructions in RST 0.  
 0048/49 High address look window.  
 004A/4B Low address look window.  
 004C-4F Store current instruction of traced program.  
 0050 Flag for LOOK: FF First time L addr laddr haddr  
                   OO only L or L laddr haddr  
 0051/52 Address current instruction (I)  
 0053 Contents accumulator (A)  
 0054 Contents flag register (F)  
 0055 Contents register B  
 0056 Contents register C  
 0057 Contents register D  
 0058 Contents register E  
 0059 Contents register H  
 005A Contents register L  
 005B/5C Contents stackpointer (SP)  
 005D/5E Contents program counter (P).  
           Points to next instruction to be performed.  
 0060 Init. value for TICC (FC after Z2 instruction).  
 0061 Init. value for GIC (1B after Z2 instruction).  
 0062/63 I0USA: By Z2 instruction set to EB5D.

... IF YOU ACCIDENTLY HIT 'RST0' ...

In the last Newsletter (page 134) a routine is given to save the pointers of the HEAP, the textbuffer and the symboltable in the RAM-area at the beginning of each BASIC program.  
 The RAM addresses 0045 - 004E are used for this purpose.

This is a very useful method, although some reservation must be made.  
 The same RAM-area is also used by the DAI Utility routines to save the contents of CPU registers etc. Because normally BASIC programs and the Utility routine are not used together, it doesn't give any problem. Only if you want to look in the particular RAM area (via UT Display e.g.), you will find complete different values!!

So take care.

Jan Boerrigter.

#### D A I FIRMWARE

Jan Boerrigter has done a big job on disassembling the DAI BASIC ROMS. If you discovered some information that might be interesting, please contact our DAI-ware detective. Those who can assist in exploring the most useful routines, will receive a free copy of the "DAI ROM BOOK". This is what you can expect for the moment:

1. On tapes you will find part of the internal software of the DAI personal computer. (ROMS V1.0)  
 The files on tape are source files, which can be used together with the DAI-ware assembler program. By means of this assembler, the comments can be easily updated.  
 2. These source files are the result of long and intensive study of the DAI firmware. They are not yet complete, there is also no guarantee that all comments are free of failures. Therefore we need your help. This is a beginning. And something is already better than nothing at all.  
 On the long run, DAI-ware users club plans to produce (in bookform) the total list of useful routines. But getting this software (24K) complete is a very time consuming business. Therefore, the assistance of anybody studying the "brains & soul" of DAI-ware is required. This tape(s) may be a help and a start for it.  
 Did you find out some not yet discovered routines in your DAI? Or do you find some failures in the comments on these source files? Please send it (hardcopy, hand/machine written or on tape) to the address below.  
 With all the single pieces added together, one day the yig-saw puzzle called 'DAI FIRMWARE' may be solved. And then we are able, as DAI-ware Users Club, to start producing a final printed version, available for anybody.  
 3. Concerning the files on tape:  
 Each block of 4K ROM is divided into 8 blocks of about 1/2 K, thus preventing too long datablocks on tape.  
 Before loading the first block into the Assembler, the buffers must be initialised on 16K dimension.  
 Each block must be handled separately, clearing the buffer contents in between, otherwise the buffer dimension may be insufficient.

Conclusion:

1. if you think you can jump into the 8080-jungle and you could provide some help in this job, please contact Jan Boerrigter:  
 Fabritiusstr 15  
 6174 RG SWEIKHUIZEN  
 NEDERLAND  
 tel 4493/2093
2. if you want to know everything about DAI-BASIC but you are not able to assist: wait for the publication of the book.
3. if you are only interested in the most useful routines you will find a lot of them in our Newsletter.

```

0 zwart           alle adressen in HEXvorm!
1 blauw
2 d.rood          29B-29C   start heap           131,0   output scrnt+
3 rood            29D-29E   size heap           RS232
4 paars           29F-2A0   start text buffer   131,1   screen only
5 groen           2A1-2A2   start symbol table  131,2   edit buffer
6 d.bruin         2A3-2A4   end of symbol table 135,2   read from
7 l.bruin         2A5-2A6   bottom screen ram   edit buffer
8 grijs
9 blauw
10 oranje
11 rose           75         cursor symbol       MODE    XMAX    YMAX
12 l.blauw        74         cursor mode         1/2    71     64
13 l.groen        72-73     cursor position     3/4    159    129
14 geel
15 wit            40,28     cass motor 1 ON
                    40,18     cass motor 2 ON
                    40,30     1 and 2 OFF
                    MERGE
                    °CLEAR XXX
                    °LOAD"A"
                    °EDIT BREAK/BREAK
                    °LOAD"B"
                    °POKE 135,2

COLORG R1 R2 R3 R4
        20 21 22 23
16 :R2*R1 R4*R3
17 :R1*R2 R3*R4      32K 7XXX
18 :R3*R1 R4*R2      12K 2XXX
19 :R1*R3 R2*R4      8K 1XXX

                    IMP INT *** IMP FPT
                    °IMP FPT
                    °CLEAR XXXX
                    °EDIT BREAK/BREAK
                    °IMP INT
                    °POKE 135,2

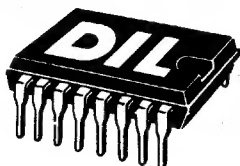
LIJN   CTRL COLOR   LIJN CTRL COLOR
23     BFEF BFEE    11    B9A7 B9A6
22     BF69 BF68    10    B921 B920
21     BEE3 BEE2     9     B89B B89A
20     BE5D BE5C     8     B815 B814
19     BDD7 BDD6     7     B78F B78E
18     BD51 BD50     6     B709 B708
17     BCCB BCCA     5     B683 B682
16     BC45 BC44     4     B5FD B5FC
15     BBBF BBBE     3     B577 B576
14     BB39 BB38     2     B4F1 B4F0
13     BAB3 BAB2     1     B46B B46A
12     BA2D BA2C     0     B3E5 B3E4

                    CTRL&COLOR BYTES IN A-MODE
                    MODE CTRL COLOR LIJN
                    1A/2A BAE7 BAE6 3
                    BA61 BA60 2
                    B9DB B9DA 1
                    B955 B954 0
                    3A/4A ACD3 ACD2 3
                    AC4D AC4C 2
                    ABC7 ABC6 1
                    AB41 AB40 0
                    5A/6A 7557 7556 3
                    74D1 74D0 2
                    744B 744A 1
                    73C5 73C4 0

FD00 b2 page signal   FF00 ser.inp.buff
      b3 serial out rdy FF01 b0-6 keyb.inp.
      b4 right paddle   b7 in7 DCE
      b5 left paddle    FF02 Interr.reg.
      b6 random data    FF03 b1 frame error
      b7 cass. input    b2 overrun error
FD01 Trigger paddle    b3 rec.buf.loaded
FD04 0-3 volume ch.1(0) b4 trans.buf.empty
      4-7 volume ch.2(1)
FD05 0-3 volume ch.3(2) FF04 COMMAND REGISTER
      4-7 volume noise  FF05 BAUD RATE REGISTER
FD06 b0 cass.out      FF06 ser.out buf.
      b1/2 paddle select FF07 keyb.output
      b3 paddle enable  FF08 interr.mask reg.
      b4 cass motor 1
      b5 cass motor 2
      b6/7 ROM BANK SWITCH

                    TEST EVENT
                    PEEK(°FD00) IAND 32
                    PEEK(°FD00) IAND 16
                    PEEK(°FD00) IAND 48
                    CH 0 FC00/FC01
                    CH 1 FC02/FC03
                    CH 2 FC04/FC05
                    STATUS FC06/FC07

```

**D.I.L.-ELEKTRONIKA**

Mijnsherenlaan 108, 3081 CH Rotterdam

**ALLE DOE-HET-ZELF ELEKTRONIKA - TECHN. TIJDSCHRIFTEN EN -BOEKEN**

LEGOTRONICS

Middenstraat 8

8800 ROESELARE BELGIE

tel. 051/207878

ORDIMAX

Rue de la Bonnefemme 11

4030 GRIVEGNEE BELGIE

MULTISOFT

Rue Bague 25

75015 PARIS FRANCE

7838837

TELEC

Steenstraat 40

9711 GP GRONINGEN NEDERLAND

MSB R. NEDELA

MARKSTRASSE 3

POSTFACH 1420

D7778 MARKDORF GERMANY

COMPAC

Plaats 25

2513 AD DEN HAAG NEDERLAND

HCC NEDERLAND hobby computer club

Prinsenhof 11

2641 RN PIJNACKER

NEDERLAND

DAI BRUSSEL

Raketstraat 60

1130 BRUSSEL BELGIE

02/2166010

HCC BELGIE

Borkelstraat 51

2120 SCHOTEN BELGIE

031/589674

DAI NEDERLAND

Van Vollenhovenstraat 15A

3016 BE ROTTERDAM NEDERLAND

010/361288

Stichting BASICned

Tolakkerweg 81

3739 JJ HOLL. RADING NEDERLAND

DIDACOM computers&amp;onderwijs

p/a I. BROEKMAN AVENBEECK 98

2182 RZ HILLEGOM

2520/18032 NEDERLAND