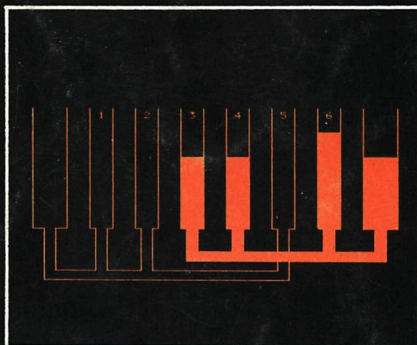
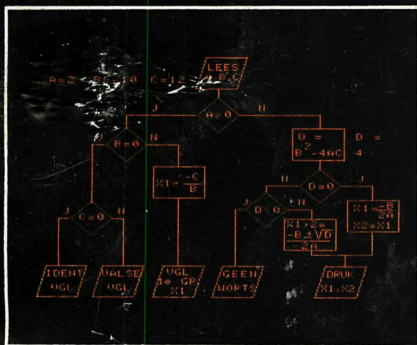


DAI DYNAMIC

tweemaandelijks tijdschrift januari-februari 1982

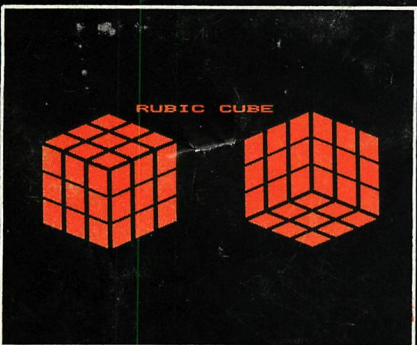


ABCDEF + 0x00
 GHIJKL 0I0K+
 MNOPQR MNO0QX
 VERBORGEN TEKENS

PLAYER SHIPSWH
 1 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 2 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 3 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 4 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 5 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 6 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 7 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 8 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 9 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 10 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 11 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 12 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 13 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 14 13 14 15 16 17 18 19 20 21 22 23 24 25
 15 14 15 16 17 18 19 20 21 22 23 24 25
 16 15 16 17 18 19 20 21 22 23 24 25
 17 16 17 18 19 20 21 22 23 24 25
 18 17 18 19 20 21 22 23 24 25
 19 18 19 20 21 22 23 24 25
 20 19 20 21 22 23 24 25
 21 20 21 22 23 24 25
 22 21 22 23 24 25
 23 22 23 24 25
 24 23 24 25
 25 24 25

a b c d e f a h
 8 8
 7 7
 6 6
 5 5
 4 4
 3 3
 2 2
 1 1
 a b c d e f a h

Player Sargon
 18:44 18:05



Deelverzamelingen van
 N en Z
 $N_0 = N \setminus \{0\}$ $Z_0 = Z \setminus \{0\}$
 $Z_0^+ = \{0, 1, 2, 3, 4, \dots\}$
 $Z_0^- = \{-1, -2, -3, -4, \dots\}$
 OPGEVEN: HUL TH RET
 = drot 1
 C = drot 2
 Q = drot 3

JUIST	FOUT
0	0

Z_0^+ N_0



personal computer users club

een uitgave van dainamic v.z.w., heide 4 - 3171 westmeerbeek belgie

COLOFON

DAInamic verschijnt tweemaandelijks.
 abonnementsprijs is inbegrepen in de
 jaarlijkse contributie : 750 Bfr.
 Bij toetreding worden de verschenen
 nummers van de jaargang toegezonden.

DAInamic redactie :

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

Vormgeving : Ludo Van Mechelen.

U wordt lid door storting van de
 contributie op het rekeningnr.
230-0045353-74 van de Generale Bank-
maatschappij, Leuven, via bankinstel-
 ling of postgiro
 Het abonnement loopt van januari tot
 december.

DAInamic verschijnt de pare maanden.
 Bijdragen zijn steeds welkom.

CORRESPONDENTIE ADRESSEN.

Redactie en software bibliotheek
 Wilfried Hermans
 Heide 4
 B 3171 Westmeerbeek
 België
 tel. : 016/69.86.23

Kredietbank Westmeerbeek
 nr. 406-3016141-33

BTW : 420.840.834

Lidgelden

Bruno Van Rompaey
 Bovenbosstraat 4
 B 3044 Haasrode
 België

tel. : 016/46.10.85

Generale Bankmaatschappij Leuven
 nr. 230-0045353-74

Inzendingen : Games & Strategy

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

tel. : 010/25.42.75

DAINAMIC
 PERSONAL COMPUTER USERS CLUB

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

belangrijke ASCII-waarden in DAIPc

functie/symbool	HEX	DEC
back-space	8	8
TAB	9	9
linefeed	A	10
clear screen	C	12
CURSOR UP	10	16
CURSOR DOWN	11	17
CURSOR LEFT	12	18
CURSOR RIGHT	13	19
space-bar	20	32
∅	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

LSD	MSD	0	1	2	3	4	5	6	7
		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

PERSONAL COMPUTER USERS CLUB

Westmeerbeek feb 82

Beste leden,

Voor de eerste keer stappen we met onze kopij naar de drukker. Gezien de groeiende oplage en omvang is het niet meer mogelijk om het drukwerk intern te verzorgen. De kwaliteit van het drukwerk zal er beslist ook op vooruit gaan. We moeten ons nu wel houden aan afspraken die in de drukkerswereld gebruikelijk zijn, waarvan de voornaamste : aantal pagina's MOD 16 = 0. Vandaar ons voornemen om U telkens 64 pagina's aan te bieden, zonder reclame.

In november hadden we weer het HCC-gebeuren. Door de beschikbare ruimte in de Julianahal ging het er veel rustiger aan toe dan vorig jaar in 't turfschip. Buiten de druk bezochte DAINamic-stand was er deze keer voor DAI-gebruikers nog meer te bekijken : de fraai verzorgde show bij MEMOCOM en het piepkleine standje van vader en zoon De Vries. Hier kregen we sublieme VIDITEL-demonstraties, meer nieuws over dit programma krijgt U verder in dit nummer.

DAInamic was ook vertegenwoordigd op de populaire computerdag in Rosendaal, niet zo druk maar toch erg boeiend.

Onze penningmeester meldt op de volgende pagina's hoe je de contributie voor 1982 kan regelen. Breng dit aub zo vlug mogelijk in orde zodat we onze oplage kunnen vaststellen. Nieuwe leden kunnen nu inschrijven voor "DAInamic 80/81", als er genoeg interesse is kunnen we dit werkje laten drukken. OPRDEP : wie wil de redactie van dit werkje verzorgen ? (neem contact met de redactie, zodat we alle beschikbaar materiaal kunnen opsturen.)

Verder in dit nummer kan je vaststellen dat er voor DAI erg veel uitstekende software verkrijgbaar is. Het is een unieke situatie dat al deze software te verkrijgen is binnen de club, aan redelijke prijzen. Door ruilactiviteiten met buitenlandse gebruikersgroepen wordt het aanbod nog groter.

De vraag naar toestellen blijkt nog steeds de productiecapaciteit van DAI te overtreffen. Het streefcijfer schijnt momenteel 500 eenheden per maand te zijn. Er wordt veel gepraat over het nieuwe toestel dat DAI op de markt zou gaan brengen: separaat toetsenbord, IEEE-interface vervangt het SOUND-gedeelte, ingebouwde kleurenmonitor, een dubbele schijf met dubbele density en .. ingebouwde hard-disc, dat klinkt mooi....afwachten maar !!!

Op zaterdag 10 april houden we onze jaarlijkse bijeenkomst in de lokalen van Tongelsbos, Bosstraat 2 te Tongerlo. Deelnemers of handelaars die ruimte willen reserveren dienen dit voor 1 april aan te vragen.

veel lees- en toetsplezier met nummer 9, tot de volgende keer....

dear members,

For the first time we gave the big job to a printing-business. Size and volume of the newsletter have grown above the capacity of the redactional staff and our machines. We hope this also will improve the quality of our magazine. In the future we will try to offer 64 pages in each issue...

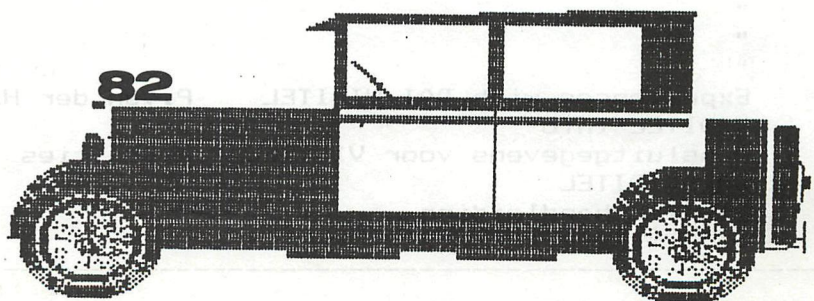
Further on, our treasurer gives information about the membership fee for 1982 and how to pay it... Even if you have local meetings and club-activities, we hope you will renew your subscription, we think it is a good idea to have one european DAI-magazine. (The more ideas, the better our magazine).

We are proud to announce many new software packages, especially the SARGON and VIDITEL, real beauties!

On 10th of april we have our second international meeting in Tongelsbos, in Tongerlo-Belgium
You are welcome !!

we hope you will enjoy this issue, see you in Tongerlo ?

Wilfried Hermans



3	remark	redactiepraatje
4	bladwijzer	
5	membership-subscription	the treasurer
6	membership-subscription	
7	DAI math routines	P.Jongen
8	DAI math routines...how to use	
9	DCE-connections	Jos Schepens
10	Serial-parallel card	A.De Dauw
11	Mailbox	Colin Hards
12	Mailbox programs	
13	Mailbox reply + Getc	R.Rens
14	The most useful entrypoints	Freddy De Raedt
15	"	
16	"	
17	Read	Jos Schepens
18	"	
19	"	
20	Read + How random is Rom?	Jos Schepens-W.Hermans
21	Graphic Demonstration	Markus Sigg
22	"	
23	"	
24	"	
25	Graphic Demonstration + Seikosha screen copy routine	
26	Screen copies Seikosha - MX100	Daniel Theys
27	Catalogus the collections	
28	Catalogus the prices	
29	Catalogus new software : G6 G7 TK2 M3	
30	Catalogus new software	
31	Catalogus secondary education 2	B.Van Rompaey
32	Catalogus "	
33	Conversions(printed on PERIFERIE zip 30)	R.De Lombaert
34	Conversions + meeting on 10 april in Tongerlo (10.00 Hr)	
35	Driehoeksvormen	K.De Bont
36	"	
37	8080 Reference :articles from Interface Age	
38	Eenvoudige Lichtpen + Luxaflex	T.Berkx
39	Schematics : sound + noise	A.F.J.De Jong
40	DAI firmware	J.Boerrigter
41	"	
42	"	
43	Bootstrap loader	Freddy De Raedt
44	"	
45	"	
46	BASIC handboeken-manuals	Jef Verwimp
47	Carpenters solution(s)	
48	Sip Tricks: tape list/crash assistance	Robert Sip
49	Sip Trics	
50	Basic Tutor	W.Hermans
51	"	
52	"	
53	"	
54	"	
55	"	
56	"	
57	"	
58	"	
59	"	
60	"	
61	"	
62	Experiences with DAI VIDITEL	P.van der Hijden
63	VIDITEL-info	
64	Aansluitgegevens voor VIDITEL	H.De Vries
65	DAI-VIDITEL	"
66	VIDITEL handleiding	"

MEMBERSHIP SUBSCRIPTIONS

+++++ A A N D A C H T + + + + + A T T E N T I O N + + + + + + + + +

Dit is het laatste nummer dat we aan de 1981-leden zenden.

Zij die hun lidmaatschapsbijdrage voor 1982 niet betaalden zullen DAInamic niet meer ontvangen.

Deze die één en ander vergeten zijn. Ter herinnering:

Betalingswijzen:

normale post : 750 BF (Belgie en Nederland) 850 BF (andere landen)
luchtpost : 1100 BF op uw overschrijving vermelden "LUCHTPOST"



- a. Kas: Belgisch geld of 50 gulden of 50 DM
- b. cheque: alleen uitschrijven in Belgisch geld
- c. bankoverschrijving op rekeningnummer:

230 - 0045353 - 74

van de Generale Bankmaatschappij Leuven t.a.v. DAInamic

- d. postoverschrijving: gebruik de postrekening van de Generale Bank Leuven. Dit nummer is: →

000 - 0000982 - 12

ALLEEN VOOR POSTOVERSCHRIJ-
VINGEN VANUIT HET BUITENLAND

en schrijf expliciet op uw betalingsformulier:

"OVERBOEKEN OP NUMMER 230-0045353 - 74 VAN DAINAMIC"

De voorraad DAInamic-nieuwsbrieven van 1981 is uitgeput. We zullen weldra de beste artikels in boekvorm bundelen en aanbieden. U kan hierop nu reeds intekenen met onderstaande strook. Opsturen naar: Bruno Van Rompaey

Bovenbosstraat 4 3044 Haasrode België

Alle briefwisseling in verband met uw lidmaatschap (nieuw adres...) opsturen naar hetzelfde adres.

MEMBERSHIP SUBSCRIPTIONS

+++++ ATTENTION +++++ A A N D A C H T +++++

d. Postal-order use the post-accountnumber of the Generale Bank Leuven.

This number is : 000 - 0000982 - 12

If you use this kind of payment write explicit on your order:

"OVERBOEKEN OP NUMMER 230 - 0045353 - 74 VAN DAINAMIC"

This kind of postal order is only allowed for postal-orders
FROM ABROAD

There are no more issues of 1981 available.

Soon we'll collect the best of DAINamic 1981 in one book. You can order this book by subscribing at the bottom of this page. Please send it to

Bruno Van Rompaey

Bovenbosstraat 4

3044 HAASRODE

Belgium



replied 1/7/82

All membership-correspondence (new address...) can be sent to the same address.

inschrijving BEST OF DAINamic 1981

NAAM : _____

STRAAT : _____

WOONPLAATS: _____ LAND : _____

Zend mij _____ exemplaar (exemplaren) van The best of DAINamic.

Eenheidsprijs: 550 BF

Handtekening

MATH ROUTINES

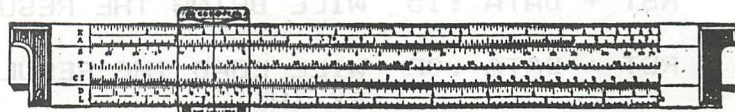
THE DAI MATH. PACKAGE.

THE VARIOUS FUNCTIONS OF THE MATH. PACKAGE ARE CALLED VIA KNOWN METHODE OF:

RST : 4
DATA : XX

THE VALUES OF THE DATA BYTE AND THE FUNCTIONS THEY PERFORM ARE LISTED BELOW. THE VALUES ARE VALLID FOR BOTH WITH AND WITHOUT HARDWARE MATH MODULE. THE RESTART ROUTINE AUTOMATICALLY SELECTS THE CORRECT ENTRY POINTS. THE ADDRESS OF THE ACTUAL CALL ARE ALSO LISTED.

DATA	CALL	CALL+	FUNCTION
:00	EDAA	E474	FLOATING POINT ADD. (X) + (Y)
:03	EDB4	E479	FLOATING POINT SUBTRACT. (X) - (Y)
:06	E0FE	E47E	FLOATING POINT MULTIPLY. (X) * (Y)
:09	E108	E483	FLOATING POINT DIVIDE. (X) / (Y)
:0C	E112	E588	ENTER FROM (HL)
:0F	E11C	E599	GET RESULT IN MEMORY (HL)
:12	E126	E55F	ENTER FROM REGISTERS A, B, C, D
:15	E133	E56F	GET RESULT IN REGISTERS A, B, C, D
:18	E140	E488	ABS(X)
:1B	E14A	E493	CHANGE SIGN. (X)*(-1)
:1E	E443	E498	FLOATING POINT INT(X)
:21	E154	E4A0	FLOATING POINT FRAC(X)
:24	E855	EDA1	(X) ^ (Y)
:27	E745	E4B1	LN(X)
:2A	E667	E4B6	EXP(X) e^(X)
:2D	E870	E4BB	LOGT(X)
:30	E880	E4C0	ALOG(X)
:33	E5F8	E4CC	SQR(X)
:36	E7D2	E4D1	SIN(X)
:39	37D9	E4D6	COS(X)
:3C	E894	E4DB	TAN(X)
:3F	E96C	E4E0	ASIN(X)
:42	E9C1	E4E5	ACOS(X)
:45	E8AC	E4EA	ATAN(X)
:48	E414	E4EF	CONVERT FROM FLOATING TO INTEGER.
:4B	E3DE	E49B	CONVERT FROM INTEGER TO FLOATING.
:4E	E16D	E4F4	INTEGER ADD. (X) + (Y)
:51	E18D	E4F9	INTEGER SUBTRACT. (X) - (Y)
:54	E1AC	E4FE	INTEGER MULTIPLY. (X) * (Y)
:57	E22B	E503	INTEGER DIVIDE. (X) / (Y)
:5A	E238	E508	INTEGER DIVIDE REMAINDER. (X) / (Y)
:5D	E30B	E517	INTEGER ABS(X).
:60	E315	E522	INTEGER CHANGE SIGN. (X) * (-1)
:63	E32E	ED19	(X) IAND (Y)
:66	E345	ED26	(X) IOR (Y)
:69	E35C	ED33	(X) IXOR (Y)
:6C	E373	ED43	COMPLEMENT (X)
:6F	E3A5	ED6C	SHIFT (X) LEFT (Y) PLACES
:72	E398	ED55	SHIFT (X) RIGHT (Y) PLACES



MATH ROUTINES

USING THE MATH. ROUTINES.

BEFORE CALLING ONE OF THE FUNCTIONS THE FIRST OPERAND HAS TO BE ENTERED INTO THE MATH ACCUMULATOR. FOR THIS TWO FUNCTIONS ARE PROVIDED:

A. THE 4 BYTE NUMBER IS PREVIOUSLY LOADED IN THE REGISTERS A-B-C-D (A= MOST SIGNIFICANT)

CALL RST :4
DATA :12 ENTER FROM REGISTERS

B. THE ADDRESS OF THE NUMBER IS LOADED IN REGISTER PAIR (HL).

CALL LXI H,NUMBER
RST :4
DATA :0C ENTER FROM MEMORY

THE SECOND OPERAND (IF NEEDED) IS ALWAYS PASSED TO THE ARITHMETIC BY LOADING ITS ADDRESS IN (HL) FIRST.

LXI H,NUMBR2
RST :4
DATA :4E INTEGER ADD

AFTER ABOVE CALLS NUMBR2 IS ADDED TO NUMBER. THE RESULT STILL RESIDES IN THE ARITHMETIC ACCUMULATOR. NOW WE CAN CALL OTHER FUNCTIONS TO BE PERFORMED ON THIS RESULT OR RETRIEVE THE RESULT.

TO OBTAIN THE RESULT TWO CALLS ARE PROVIDED:

RST + DATA :15 WILL BRING THE RESULT IN REGISTRS A,B,C,D

RST + DATA :0F WILL LOAD THE RESULT IN MEMEORY ADDRESSED BY (HL).

FOLLOWING AN EXAMPLE

START LXI H,INMBR1 INTEGER NUMBER
RST :4
DATA :0C ENTER NUMBER 1
RST :4
DATA :4B CONVERT TO FLOATING POINT
RST :4
DATA :33 SQUARE ROOT
LXI H,RESLT1
RST :4
DATA :0F STORE RESULT IN RESLT1
LXI H,FNMBR FLOATING POINT NUMBER
RST :4
DATA :00 ADD IT TO PREVIOUS RESULT
RST :4
DATA :15 ARITH RESULT IN A,B,C,D
..... ETC. ETC.....

DCE-CONNECT IONS

DCE-BUS CONNECTIONS PERSONAL COMPUTER AND RW-CARDS.

NAME	DESCRIPTION	P.C. PIN#	R.W.C. PIN#	DCE FUNCTION
****	*****	*****	*****	*****
POB0	GIC PORT 0 BIT 0	16	24	DATA BIT 0
POB1	GIC PORT 0 BIT 1	14	26	DATA BIT 1
POB2	GIC PORT 0 BIT 2	12	28	DATA BIT 2
POB3	GIC PORT 0 BIT 3	10	30	DATA BIT 3
POB4	GIC PORT 0 BIT 4	9	29	DATA BIT 4
POB5	GIC PORT 0 BIT 5	11	27	DATA BIT 5
POB6	GIC PORT 0 BIT 6	13	25	DATA BIT 6
POB7	GIC PORT 0 BIT 7	15	23	DATA BIT 7
P1B0	GIC PORT 1 BIT 0	30	12	DEV. ADDR 0
P1B1	GIC PORT 1 BIT 1	31	10	DEV. ADDR 1
P1B2	GIC PORT 1 BIT 2	32	8	DEV. ADDR 2
P1B3	GIC PORT 1 BIT 3	25	7	DEV. ADDR 3
P1B4	GIC PORT 1 BIT 4	24	9	CARD ADDR 0
P1B5	GIC PORT 1 BIT 5	23	11	CARD ADDR 1
P1B6	GIC PORT 1 BIT 6	22	13	CARD ADDR 2
P1B7	GIC PORT 1 BIT 7	21	15	CARD ADDR 3
P2B0	GIC PORT 2 BIT 0	26	18	BUS EXPAND
P2B1	GIC PORT 2 BIT 1	27	17	WR" (NEG)
P2B2	GIC PORT 2 BIT 2	28	16	RD" (NEG)
P2B3	GIC PORT 2 BIT 3	29	14	NOT USED
P2B4	GIC PORT 2 BIT 4	20	19	NOT USED
P2B5	GIC PORT 2 BIT 5	19	20	NOT USED
P2B6	GIC PORT 2 BIT 6	18	21	NOT USED
P2B7	GIC PORT 2 BIT 7	17	22	NOT USED
EXINTR	EXTERNAL INTERRUPT	6	4	EXINTR
IN7	PARALLEL INPUT BIT 7	5	3	IN7
EXRESET"	EXTERNAL RESET (NEG)	7	5	EXRESET"
+12V	+12 V DC	2	2	+12V
+5V	+5 V DC	1	31	+5V
-5V	-5 V DC	3	1	-5V
GND	GND (0V DC)	4	6	GND
INTR	8080 INTR-PIN 14	33	NA	NON EXISTENT
IN7	PARALLEL INPUT BIT 7	34	NA	NON EXISTENT
NC	NOT CONNECTED	8	NA	NON EXISTENT

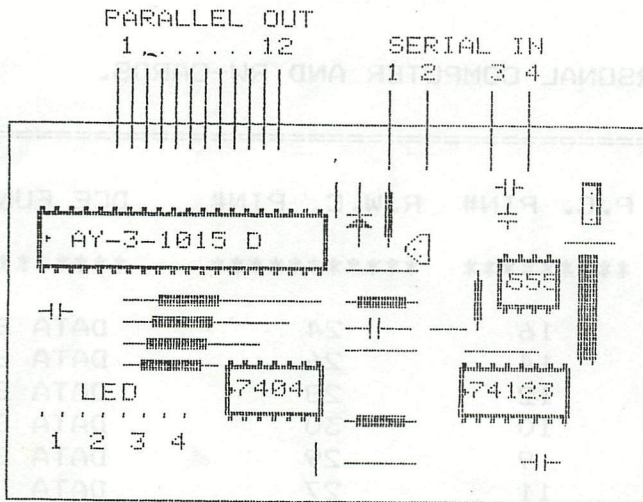
 STOP PRESS

I am designing a 8 inch floppy for my DAI, can someone help ?
 Peter Jongen Zeemanhof 25 2871 JW Schoonhoven Nederland

Which English family, preferably in London or other town where language courses
 can be followed, is willing to have Just van Dunne, aged 17, as a paying guest
 for three weeks in July or August 1982.
 Please reply to : M. van Dunne, Hoflaan 70, 3062 JJ Rotterdam Netherlands.

SERIAL - PARALLEL INTERFACE

FOR EPSON, SEIKO GP-80, MICROLINE, CENTRONICS



LED CODE :

- 1 = PARITY ERROR
- 2 = FRAME ERROR
- 3 = OVERRUN ERROR
- 4 = DATA READY

DE DAUW A. WALLENHOF 93, B2770 NIEUKERKEN TEL031/770676

PIN CONNECTIONS :

Parallel cable (from left to right) : Serial cable (from L to R) :

- 1 - brown = busy
- 2 - red = AKNLG (n.c.)
- 3 - orange = data 8
- 4 - yellow = data 7
- 5 - green = data 6
- 6 - blue = data 5
- 7 - violet = data 4
- 8 - grey = data 3
- 9 - white = data 2
- 10 - black = data 1
- 11 - brown = strobe
- 12 - red = ground

- 1 = 0 volt PC
- 2 = serial in
- 3 = busy out
- 4 = + 12 volt.

EPSON MACHINE ROUTINE FOR SCREEN COPY < CALLM#300 >

>D300 3AF

```

0300 C5 D5 E5 F5 F3 3A 40 00 F5 E6 3F F6 80 32 40 00
0310 32 06 FD FB 21 00 00 4C CD 84 EB 3E F9 80 E6 FB
0320 C6 07 4F 13 D5 C5 3E 1B CD 97 03 3E 4E CD 97 03
0330 3E 00 CD 97 03 3E 1B CD 97 03 3E 41 CD 97 03 3E
0340 08 CD 97 03 7B CD 97 03 7A CD 97 03 3E 01 F5 CD 84
0350 CD 97 03 7B CD 97 03 7A CD 97 03 3E 01 F5 CD 84
0360 EB D1 DA 78 03 FE 08 7A 17 0D D2 5D 03 EE FF CD
0370 97 03 C1 C5 23 C3 5B 03 C1 3E 0A CD 97 03 79 D6
0380 08 4F 21 00 00 D1 D2 24 03 F1 F3 32 40 00 32 06
0390 FD FB F1 E1 D1 C1 C9 F5 3A 00 FD E6 08 CA 98 03
03A0 3A 03 FF E6 10 CA A0 03 F1 32 06 FF C9 FF FF FF
    
```

Technicaldata :

- needs no software
- speed adjustable from 50 to 9600 baud
- 1 or 2 stop-bits
- used for grafical printers EPSON (MX80-II,82,100), SEIKO GP80, MICROLINE a.o.
- error-codes : 1)parity error (red led)
2)frame error (red led)
3)overrun error (red led)
4)data ready (green led)

On the card , there is a 20 Kohm trimmer. You can adjust the clock speed by turning on the trimmer. By extremely temperature variations , it may be necessary to adjust the trimmer. The exact frequency is about 153600 Hz. The card works correctly, only when the green LED is flashed , and the middle red LED stay out.

5 December 1981

50, Marlingford Way,
Easton,
Norwich,
Norfolk, NR5 9HB.
England.

W. Hermans,
DAInamic,
Heide 98,
3171 Westmeerbeek,
Belgium,

MAILBOX

Dear Mr. Hermans,

Firstly, happy Christmas and new year to you. Please accept my appologies for writing to you in English. Thank you for sending me DAINamic No. 3 7, and recently DAINamic No. 8.

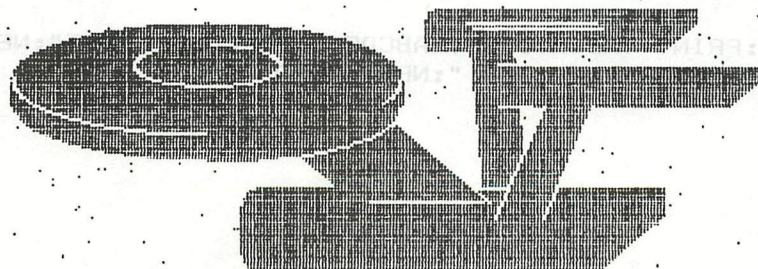
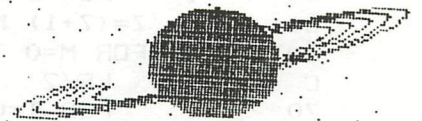
I am writing to you in the hope that you can give me some advice in connection with a program problem that I cannot solve. I am presently writing a banking program where data is held in arrays, these being 'SAVEA' on tape so that later this data can be 'LOADA' back into the main program to be brought up to date. I have condensed the problem and this is illustrated in the program "PROBLEM" saved after this letter. The fault occurs when using large ARRAYS and makes the program run slow. The program I have sent you can be split into three parts :-

1. FOR NEXT loop
2. Fill ARRAY A# (RND Characters)
3. FOR NEXT loop

- ① The routine in the first FOR NEXT loop is identical with the routine in the second, however the second loop runs much slower than the first. Why does this happen and can you tell me how to avoid this ?
- ② I have also been experimenting with altering the line control address bytes (from #7A to #77) to obtain 34 lines per screen page, which is useful. I have tried to change the screen from 4 colour to 16 colour characters by altering the line control bytes from #7A to #FA, as implied on page 16/17 of the DAI P.C. manual, but with no sucess. Do you know how to have more than 4 colour characters on one character line ?
- ③ I have enjoyed reading DAINamic very much, but would like to know if there are any English translations available ? I had DAINamic 3 translated, but it was very expensive and I can not afford to have others done. Could you let me know if anyone in England is thinking of organising something like this ?

Yours sincerely,

Colin Hards



```

T
1 REM ..... PROGRAM TO DEMONSTRATE PROBLEM .....
10 CLEAR 10000: DIM A$(3.0,3.0,3.0,3.0)
100 REM .....routine to fill screen.....
110 PRINT CHR$(12):FOR LZ=1 TO 23:FOR CZ=0 TO 58
120 PRINT CHR$(127);:NEXT:PRINT :NEXT:PRINT CHR$(12)
200 REM .....routine to fill ARRAY.....
210 FOR AZ=0 TO 3:FOR BZ=0 TO 3:FOR DZ=0 TO 3:FOR EZ=0 TO 3
220 A$(AZ,BZ,DZ,EZ)=CHR$(INT(RND(26.0)+65.0))
230 PRINT " A$( ";AZ;" ";BZ;" ";DZ;" ";EZ;" )=";A$(AZ,BZ,DZ,EZ)
240 NEXT EZ:NEXT DZ:NEXT BZ:NEXT AZ
300 REM .....routine to fill screen.....
310 PRINT CHR$(12):FOR LZ=1 TO 23:FOR CZ=0 TO 58
320 PRINT CHR$(127);:NEXT:PRINT :NEXT:PRINT CHR$(12)
400 REM .....Comment.....
410 PRINT " NOTE THE TIME TAKEN TO RUN LINES 100-120, COMPARED"
420 PRINT " WITH LINES 300-320, NEARLY 4x AS LONG. THE SAME DELAY"
430 PRINT " OCCURS WHEN USING 1,2,OR 3 DIMENTIONED ARRAYS TO A"
450 PRINT " SMALLER DEGREE. CAN YOU SUGGEST A SOLUTION SO THAT"
460 PRINT " THIS DOES NOT OCCUR ( WHILE KEEPING THE FILLED ARRAY )?"

```

A SOLUTION....

```

T
1 REM ..... PROGRAM TO DEMONSTRATE PROBLEM .....
10 CLEAR 10000:T$=CHR$(127): DIM A$(3.0,3.0,3.0,3.0)
100 REM .....routine to fill screen.....
110 PRINT CHR$(12):FOR LZ=1 TO 23:FOR CZ=0 TO 58
120 PRINT T$;:NEXT:PRINT :NEXT:PRINT CHR$(12)
200 REM .....routine to fill ARRAY.....
210 FOR AZ=0 TO 3:FOR BZ=0 TO 3:FOR DZ=0 TO 3:FOR EZ=0 TO 3
220 A$(AZ,BZ,DZ,EZ)=CHR$(INT(RND(26.0)+65.0))
230 PRINT " A$( ";AZ;" ";BZ;" ";DZ;" ";EZ;" )=";A$(AZ,BZ,DZ,EZ)
240 NEXT EZ:NEXT DZ:NEXT BZ:NEXT AZ
300 REM .....routine to fill screen.....
310 PRINT CHR$(12):FOR LZ=1 TO 23:FOR CZ=0 TO 58
320 PRINT T$;:NEXT:PRINT :NEXT:PRINT CHR$(12)
400 REM .....Comment.....
410 PRINT " NOTE THE TIME TAKEN TO RUN LINES 100-120, COMPARED"
420 PRINT " WITH LINES 300-320, NEARLY 4x AS LONG. THE SAME DELAY"
430 PRINT " OCCURS WHEN USING 1,2,OR 3 DIMENTIONED ARRAYS TO A"
450 PRINT " SMALLER DEGREE. CAN YOU SUGGEST A SOLUTION SO THAT"
460 PRINT " THIS DOES NOT OCCUR ( WHILE KEEPING THE FILLED ARRAY )?"

```

PAGE 1 -- 16 COLOR IN MODE 0 ~ HARDWARE CHECK -- FLST V2.2

```

5 REM 16 COLOR IN MODE 0 ~ HARDWARE CHECK
10 MODE 0:PRINT CHR$(12):COLORT 8 0 8 0:POKE #75,ASC("!"):O=
C (PEEK(#2A6) IAND #F0) SHL 8
20 A$=
C "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz012345
C 67"
30 FOR L=0 TO 22:C=O+#FEF-#86*L:POKE C-1,#48:POKE C,#FA
40 1 FOR D=C-3 TO C-133 STEP -2:POKE D,(Z SHL 4)+8
50 Z=(Z+1) MOD 16:NEXT:PRINT A$:NEXT
60 FOR M=0 TO 9:A$="## DAI PERSONAL COMPUTER ##":LE=LEN(A$):M=
C 1 LE/2
70 1 FOR T=0 TO M:PRINT TAB(29-T);MID$(A$,M-T,T+T):NEXT
80 1 FOR A=0 TO 2:PRINT TAB(29);"A":NEXT:PRINT TAB(27);"AAAAA":
C 1 PRINT :PRINT
100 NEXT:FOR A=7 TO 18:POKE O+#FEF-#86*A,#F0+J:J=(J+1) MOD 16:
C NEXT
110 FOR A=0 TO 23:PRINT TAB(A*1.5);"ABCDEFGHIJKLMNOIPQRST":NEXT:
C FOR A=16 TO 23:PRINT " " :NEXT:GOTO 110

```

Dear Mr. Hards...

problem no 1

Your second loop is so much slower, because BASIC needs some free space in the HEAP to execute the CHR\$()-command. (ref : problems mentioned in the program FORMAT-LISTING newsletter 2 p.17).
 If a lot of space in the HEAP is already used by your array, the search for free space takes longer ... that's all.
 We can solve this by using T\$=CHR\$(127) as in line 10.

problem no 2

You could have more than 24 lines of mode 0, but you will have to arrange :
 - initialisation of the screen ram
 - use your own screen handler for mode 0

problem no 3

In the handbook on p.17 we find the following table :
 bits 7&6 of line control byte:

00	four colour graphics
01	four colour characters
10	sixteen colour graphics
11	sixteen colour characters

In the earlier machines, the last data (#FA in control byte) only gave the effect of useless garbage on the screen. This was due to some wrong layout of the circuit board.

You can check your circuit with the program "16 COLOR IN MODE 0. We will publish the modifications in our next issue, together with the schematics of the video circuit as supplied by Mr DE HOOG and Mr VAN LEEUWEN.

G=GETC:G=GETC:G=GETC

GETC

1000 G=GETC: WAIT TIME 5: IF G=0 GOTO 1000

This line is mostly used, when waiting for a GETC-command. The WAIT TIME is recommended in the Dai-handbook, before otherwise, when you're pressing a key, this key is to be scanned more than once.

But most people, when pressing a key, look to the screen and wait for a feedback. So they are filling up the key-memory -this holds the first tree keys-. By then it is possible to see the expected routine 4 times.

To avoid this we have to empty the key-memory, every time, before asking a decision to the user. You can do this by using the function GETC 3-times as a dummy. So the key-memory will be emptied.

1000 G=GETC:G=GETC:G=GETC

1010 G=GETC: IF G=0 GOTO 1010

J#L.
PAGE 01 DAI INTERFACE MODULE 1 30JAN82

```

002 *****
003 *
004 * FOLLOWING ASSEMBLER LISTING CONTAINS EQU'S AND *
005 * DESCRIPTIONS FOR USE OF DAI ROM ROUTINES. *
006 * FORMAT : *
007 * LABEL EQU :XXXX SHORT FUNCTION *
008 * * ENTRY : PARAMETERS TO FUCTION *
009 * * FUNC : PERFORMED ACTION *
010 * * EXIT : RETURNED VALUES *
011 * * CORRUPTED REGISTERS *
012 *
013 *****
014 *
015 * NOTES :
016 * -----
017 *
018 * 1) UNLESS OTHERWISE STATED INFO IS CORRECT FOR
019 * BOTH ROM VERSIONS (BASIC 1.0 AND 1.1).
020 *
021 * 2) A REGISTER IS CORRUPTED WHEN IT'S VALUE MAY BE
022 * CHANGED IN AN UNCONTROLLED WAY.
023 * IF THIS IS NOT ALLOWED, PRESERVE THEM DURING
024 * CALL (PUSH X, CALL LABEL, POP X).
025 *
026 * 3) REGISTERS AND FLAGS WHICH ARE NOT MENTIONED IN
027 * EXIT ARE NOT ALTERED OR NOT USED IN THE ROUTINE
028 *
029 * 4) UNLESS OTHERWISE STATED ALL ROUTINES RESIDES IN
030 * ROM WITH NO BANKSWITCHING. SO THEY ARE ALWAYS
031 * ACCESSIBLE.
032 *
033 * 5) ABBREVIATIONS :
034 * PSW : A,F TOGETHER
035 * A : ACCUMULATOR
036 * F : ALL FLAGS
037 * BC : B,C TOGETHER
038 * DE : D,E TOGETHER
039 * HL : H,L TOGETHER
040 *
041 * 6) FLAGS :
042 * +-----+-----+-----+-----+-----+-----+-----+-----+
043 * ! S ! Z ! O ! AC ! O ! P ! 1 ! CY !
044 * +-----+-----+-----+-----+-----+-----+-----+
045 * CY : CARRY
046 * P : PARITY
047 * AC : AUX CARRY
048 * Z : ZERO
049 * S : SIGN
050 *
051 *
052 * PART 1 : GENERAL SUPPORT ROUTINES
053 * -----
054 *
055 COMPDE EQU :DE14 DOUBLE BYTE COMPARE

```

```

056      * ENTRY : 2 16 BIT VALUES IN DE AND HL
057      * FUNC  : COMPARE VALUES, SET FLAGS
058      * EXIT  : IF DE < HL THEN Z = 0 AND CY = 0
059      *          IF DE = HL THEN Z = 1 AND CY = 0
060      *          IF DE > HL THEN Z = 0 AND CY = 1
061      *          A, OTHER FLAGS CORRUPTED
062      *
063      SUBDE EQU :DE1A      DOUBLE BYTE SUBTRACTION
064      * ENTRY : 2 16 BIT VALUES IN DE AND HL
065      * FUNC  : SUBTRACT DE FROM HL, SET FLAGS
066      * EXIT  : RESULT IN HL
067      *          FLAGS AS FOR COMPDE
068      *          IF DE > HL THEN RESULT IN '2' COMPLEMENT
069      *
070      SUBDED EQU :D790     DOUBLE BYTE SUB + DECREMENT
071      * ENTRY : IDEM SUBDE
072      * FUNC  : IDEM SUBDE BUT RESULT IS DECREMENTED BY 1
073      * EXIT  : IDEM SUBDE ( '1' COMPLEMENT IN NEGATIVE )
074      *
075      NEGHL EQU :DE26     DOUBLE BYTE "2" COMPLEMENT
076      * ENTRY : 16 BIT VALUE IN HL
077      * FUNC  : TAKES '2' COMPLEMENT OF HL
078      * EXIT  : RESULT IN HL
079      *
080      ADDA EQU :DE30      ADD SINGLE TO DOUBLE BYTE
081      * ENTRY : 8 BIT VALUE IN A, 16 BIT VALUE IN HL
082      * FUNC  : ADD 8 BIT TO 16 BIT ( A + HL )
083      * EXIT  : RESULT IN HL
084      *
085      ADDMI EQU :DE39     SKIP A STRING
086      * ENTRY : POINTER TO A STRING IN HL
087      * FUNC  : FETCH FIRST BYTE STRING (= LENGTH STRING)
088      *          INCREMENT POINTER ( HL + 1 )
089      *          ADD LENGTH TO POINTER
090      * EXIT  : POINTER AFTER STRING IN HL
091      *
092      MULA EQU :DEBF      MULTIPLY 16 BIT WITH 8 BIT
093      * ENTRY : 8 BIT VALUE A, 16 BIT VALUE IN HL
094      * FUNC  : MULTIPLY 16 BIT WITH 8 BIT
095      *          REPORT OVERFLOW
096      * EXIT  : IF CY = 0 THEN RESULT IN HL
097      *          IF CY = 1 THEN OVERFLOW OCCURED
098      *
099      DELAY EQU :DE41     FIXED DELAY
100     * ENTRY : -
101     * FUNC  : RUN A FIXED DELAY LOOP
102     * EXIT  : IF INTERRUPTS ENABLED THEN
103     *          RETURN AFTER APPROX 750 MS
104     *          ELSE
105     *          RETURN AFTER 660 MS
106     *
107     MOVE EQU :DE4F      MOVE BLOCK OF MEMORY
108     * ENTRY : RAM BLOCK FROM DE TILL ( HL - 1 )
109     *          TARGET BEGIN IN BC
110     * FUNC  : MOVE RAM BLOCK TO NEW BEGIN
111     *          ( ALLOW OVERLAY OLD AND NEW BLOCK )
112     * EXIT  : MOVED RAM BLOCK
113     *          BC, DE AND HL CORRUPTED

```

```

114      *
115      FILL      EQU      :DE7C      FILL BLOCK OF MEMORY
116      * ENTRY : RAM BLOCK FROM DE TILL ( HL - 1 )
117      *          8 BIT VALUE IN A
118      * FUNC   : FILL RAM BLOCK WITH 8 BIT VALUE
119      * EXIT   : DE POINTS AFTER BLOCK
120      *          A, F CORRUPTED
121      *
122      ALFA      EQU      :DE02      TEST A..Z
123      * ENTRY : ASCI VALUE IN A ( MOST SIGN BIT = 0 )
124      * FUNC   : TEST ASCI VALUE FOR A,B,C,...,Z
125      * EXIT   : CY SET IF TRUE, OTHER F CORRUPTED
126      *
127      ALFNUM    EQU      :DE09      TEST 0..9 OR A..Z
128      * ENTRY : IDEM ALFA
129      * FUNC   : TEST ASCI VALUE FOR 0,1,...,9,A,B,...,C
130      * EXIT   : IDEM ALFA
131      *
132      NUMER     EQU      :DE0D      TEST 0..9
133      * ENTRY : IDEM ALFA
134      * FUNC   : TEST ASCI VALUE FOR 0,1,2,...,9
135      * EXIT   : IDEM ALFA
136      *
137      POPRET    EQU      :C14D      POP REGS AND RETURN
138      * ENTRY : REGISTER AND RETURN ADDRESS ON STACK
139      *          THIS ADDRESS CAN ONLY BE USED WITH A JMP,
140      *          NEVER WITH CALL
141      * FUNC   : POP H, POP D, POP B, POP PSW, RET
142      * EXIT   : ALL REGISTERS RESTORED FROM STACK
143      *          RETURNED TO CALLING SUBROUTINE
144      *
145      RSTART    EQU      :C80C      HARDBREAK TO BASIC
146      * ENTRY : JMP OR CALL
147      * FUNC   : PERFORM ***BREAK
148      *          START UP BASIC COMMAND MODE
149      * EXIT   : STACK POINTER SET TO :F900, NO RETURN
150      *          POSSIBLE
151      *
152      * PART 2 : CHARACTER INPUT/OUTPUT ROUTINES
153      * -----
154      *
155      GETC      EQU      :D&BE      INPUT A CHARACTER
156      * ENTRY : KEYBOARD SET TO FULL SCAN OR BREAK SCAN
157      *          ( KNSCAN SET TO 0 OR :FF )
158      *          INPUT SWITCH INSW SET TO 0 OR 1
159      * FUNC   : IF KNSCAN = 0 GET 1 CHAR FROM KEYBOARD
160      *          FIFO OR FROM RS232 INPUT DEVICE
161      *          IF KNSCAN = :FF THEN ONLY BREAK KEY
162      *          PRESSED CAN BE DETECTED
163      * EXIT   : CY SET IF BREAK KEY PRESSED
164      *          A = ASCI VALUE INPUTTED CHARACTER
165      *          A = 0, Z = 1 IF NO CHARACTER INPUTTED
166      *          OTHER F CORRUPTED
167      KNSCAN    EQU      :2B9       SET TO SCAN FOR BREAK ONLY
168      INSW      EQU      :296       INPUT SWITCH
169      * 0 = DAI KEYBOARD , 1 = RS232 CONNECTED KEYBOARD
170      * INSW IS SET UP TO 0 IF FIRST CHARACTER AFTER
171      * IS INPUTTED ON THE DAI KEYBOARD.

```

TO BE CONTINUED..



Steeds meer tijdschriften beginnen met regelmatige DAI rubrieken of op z'n minst toch een artikeltje over de DAI P.C.. Onlangs verschenen er artikels i.v.m. de DAI in o.a. :

- 1) * Test Aankoop
- 2) * Databus
- 3) * HCC Nieuwsbrief
- 4) * L'Ordinateur Individuel
- 5) * Personal Computer World

1) Opmerkelijk is het artikel dat in Test Aankoop van November verscheen. Onder de titel "Help, een computer in huis" worden 11 huiscomputers getest nl. ATOM ACORN, APPLE II Europlus, COMMODORE PET 4016N, DAI, ITT 2020, SHARP MZ-80K, SINCLAIR ZX 80, SINCLAIR ZX 81, TANDY mod.1/lev.2, ROCKWELL AIM 65-1415 en SHARP PC 1211.

De 3 beste kopen zijn volgens TA de SINCLAIR ZX 81 als eerste kennismaking met microcomputers, de DAI als beste huiscomputer en tenslotte de TANDY TRS 80 voor kleinere budgetten.

I.v.m. de DAI citeren we letterlijk

"de DAI bleek de beste huiscomputer van de 11 geteste modellen, wie aan dit toestel z'n vrije tijd wil besteden zal wel 46.741 fr. moeten neertellen. Maar daar krijg je dan ook heel wat voor: 48 K geheugen, 16 kleuren, een muzieksynthesizer, goede tot zeer goede Basic en in/output;"

Commentaar is voor de DAI fanaat overbodig.

2) In DATABUS van 19 OKT 1981 zijn 2 artikels belangrijk voor DAI-gebruikers.

- a) BASICODE voor DAI door Th. van LIESHOUT.
- b) Wat is wat in BASIC (vergelijking tussen 6 dialecten)

a) BASICODE is voor de meesten wellicht geen onbekende meer, toch willen we in het kort even trachten samen te vatten waar het om gaat. De cassette lees- en schrijfroutines zijn voor bijna alle populaire microcomputers verschillend. Dit maakt het uitwisselen van standaard BASIC programmas zeer moeilijk. De NOS was geconfronteerd met een dergelijk probleem voor de Hobbyscoop uitzendingen. Tijdens die uitzendingen worden soms BASIC prgr's de ether ingestuurd. Om geen 10 versies van hetzelfde prgr. te moeten uitzenden werd een nieuwe communicatiestandaard opgesteld.

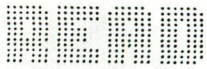
Voor de verschillende microcomputers moet dan een soort vertaalprgr geschreven worden. Feitelijk zijn dit nieuwe cassette in & out routines. Voor de DAI werden deze prgr's geschreven door dhr. van Lieshout die zeer zeker knap werk heeft geleverd.

Enkele opmerkelijke eigenschappen van de BASICODE prgr's:

de BREAK-toets blijft actief ook tijdens het inlezen, dit in tegenstelling met de gewone DAI LOAD. Het gedeelte van het prgr. dat dan al ingelezen is kan geLIST en geRUNd worden.

In het artikel worden ook enkele optionele hardware modificaties voor de cassette in- & output gegeven. We hebben echter geen enkel probleem ondervonden met onze standaard interfaces.

b) Het tweede artikel in DATABUS, door J. van EGDOM geschreven vergelijkt de BASIC-dialecten van de Apple II, DAI, Exidy Sorcerer, PET/CBM, P2000 en TRS 80. Een dergelijk overzichtje is o.a. zeer nuttig wanneer men BASICODE prgr's wil schrijven. Deze moeten natuurlijk in "standaard BASIC" geschreven worden.



3) In de laatste nieuwsbrief van de HCC (nr. 33) is eveneens een prgr. voor de DAI gepubliceerd. Onder de hoofding DAI-Hernummersen wordt een zogenaamde renumber, in BASIC geschreven door Th. M. BOS, voorgesteld. Een goed hernummeringsprogramma hernummers niet alleen de regelnummers maar ook alle verwijzingen naar regelnummers in bvb. GOTO linenr., GOSUB linenr., IF condition THEN linenr., ON expression GOTO linenr.1, linenr.2,..., enz., enz... Het prgr is zo'n 58 lijnen lang en modificeert volgens dhr. Bos GOTO, GOSUB, IF THEN GOTO en IF THEN GOSUB. IF THEN linenr. worden niet hernummerd. Bij dit alles willen we toch enige kritische opmerkingen plaatsen.

a) Het feit dat ON GOTO en ON GOSUB alsook IF THEN niet worden aangepast maken van dit prgr. geen "veilig" prgr.. Alle prgr's zullen dus niet correct hernummerd worden.

b) Het prgr. kan op vele manieren verbeterd worden. Vooreerst moeten alle variabelen geïnitieerd worden. Voor BASIC V1.0 is dit niet nodig daar ook een RUN linenr. alle variabelen op 0 zet, doch met BASIC V1.1 is dit wel nodig. Dan worden de veranderlijken alleen geïnitieerd door een RUN zonder linenr..

c) Indien slechts een deel van het prgr. moet hernummerd worden dan moeten toch ALLE referenties naar lijnummers nagekeken en eventueel aangepast worden. Op het eerste zicht gebeurt dit niet.

d) Het prgr. zal makkelijk "crashen" door onbestaande lijnummers op te geven voor bvb. oude laatste lijn. Een iets veiliger manier om het prgr. te doen stoppen: 64120 IF LO%(Z%) >= L2% THEN 64160

e) Er kunnen met het voorgestelde prgr. niet meer dan 256 lijnen hernummerd worden. Deze beperking vloeit voort uit de maximale dimensie van array's. Een mogelijke manier om dit op te lossen is door PEEK en POKE.

f) Een trucje om het prgr. te versnellen:

bvb. lijn 64100:PO2%=L3%-PO1%*256:... kan vervangen worden door:PO2%=L3% MOD 256:...

Men wint ongeveer 1 sec per 1000 dergelijke statements.

g) De oude startlijn wordt bij het begin gevraagd doch er wordt niets mee gedaan: het hernummeren begint van in het begin.

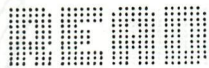
Besluit: een prgr. dat met de grootste omzichtigheid moet gebruikt worden! Het is misschien beter dan niets maar 2 maal nakijken van het resultaat is zeker nodig.

Het is overigens zeer moeilijk om een redelijke renumber te schrijven voor de DAI: door de zeer compacte stockering van de prgr's in het geheugen is het moeilijk te weten hoe lang een expression bvb is.

Moge deze opmerkingen niet verhinderen dat er in de HCC nieuwsbrief^w nog DAI prgr's zouden verschijnen!

4) Op pag.171 in L'Ordinateur Individuel staat een klein prgr. geschreven door Christian FOELS voor de DAI. Het prgr. is slechts 8 lijnen lang en bootst het effect van een generiek na.

Het kan best vergeleken worden met een wit blad papier dat op een gedrukte tekst gelegd is en dan langzaam naar beneden wordt geschoven zodat de letters van boven naar beneden zichtbaar worden. Dit wordt bekomen door wat te "spelen" met de screen controlewoorden. Het effect mag zeker gezien worden.



5) Het laatste tijdschrift dat we deze maand zullen bespreken is Personal Computer World, afgekort PCW. Hierin verschijnt sedert kort een regelmatige rubriek verzorgd door Alan SUTCLIFFE. PATTERNS is de naam van de rubriek die geïllustreerd wordt met voorbeeldjes op de DAI. In het nummer van november wordt nagegaan wat de kwaliteit is van de willekeurige getallen door de DAI gegenereerd. Op de vraag "HOW RANDOM IS RANDOM?" tracht dhr. Sutcliffe te antwoorden door test-prgr's voorgesteld door het NBS (National Bureau of Standards uit de USA) op de DAI te laten lopen. Achter deze testprogrammas steekt een diepe theoretische statistische achtergrond. Het is echter niet de bedoeling deze hier uit de doeken te doen. De besluiten zijn op zichzelf al interessant genoeg. De test op de gemiddelde waarde van de willekeurige getallen tussen 0 en 1 is zonder meer bevredigend zowel voor de hardware als software toevalsgenerator.

Met de verdelingsfunctie ligt het wel anders. Theoretisch zou deze een rechte moeten benaderen. Voor de software rnd-generator is dit ook zo. De hardware rnd-generator blijkt echter veel slechter te zijn op dit gebied, althans toch op het toestel van Alan Sutcliff. Hier bleken sterke pieken bij 0, 1 en 0.5 op te treden! Het zou interessant zijn te weten of dit op alle DAI's het geval is. Het programma om de verdelingsfunctie te tekenen is terug te vinden in dat nummer van PCW.

Zo dit was het voor deze maand. Laat ons hopen dat de interesse voor de DAI in dezelfde mate blijft toenemen, het kan de DAI-users alleen maar ten goede komen!

In Kilobaud Microcomputing van december 1981 wordt de DAI eveneens vernoemd. Naar mijn weten is het de eerste maal dat in een Amerikaans tijdschrift gewag wordt gemaakt van het bestaan van de DAI. Het betreft geen uitgebreide test o.i.d. doch wel een zeer bondige samenvatting van de belangrijkste kenmerken.

In Engeland is PCW (personal computer world) waarschijnlijk het tijdschrift dat de meeste aandacht besteedt aan de DAI. Dit voornamelijk door de artikels van Alan Sutcliffe (de reeks "PATTERNS"). Ook deze maand (januari 1982) waren er weer twee programma's voor de DAI bij.

Het eerste lost een domino vraagstuk op. Dit naar aanleiding van de twee vorige artikels over Ken Knowlton, een artist die de computer aanwendt voor het ontwerpen en maken van zijn grafische kunststukken. Zo maakt hij onder andere digitale beelden met domino's. De vraag stelt zich dan natuurlijk op hoeveel manieren een aantal domino-sets op een $n \times m$ rechthoek kunnen geplaatst worden. Voor een meer nauwkeurige omschrijving van het probleem wordt verwezen naar PCW van december 1981 en januari 1982.

Het tweede programma is een soort SGT (standard graf text of slow graf text dit in vergelijking met FGT).

Zoals u al kunt vermoeden kan men met dit programma lettertekens in grafische mode op het scherm plaatsen.

Er wordt in dat programma geen gebruik gemaakt van DOT of DRAW command's (zoals in de oorspronkelijke DAINAMIC graf-text en zoals te vinden is in de DAI-manual) daar deze te traag zijn.

In plaats daarvan worden de noodzakelijke bitcombinaties rechtstreeks in video-ram ge-POKE-d. Het programma waarvan de listing in PCW staat kan alleen 10*" R " op het scherm schrijven doch het programma kan gemakkelijk uitgebreid worden tot het hele alfabet en andere te definiëren symbolen.

DAI GRAPHICS

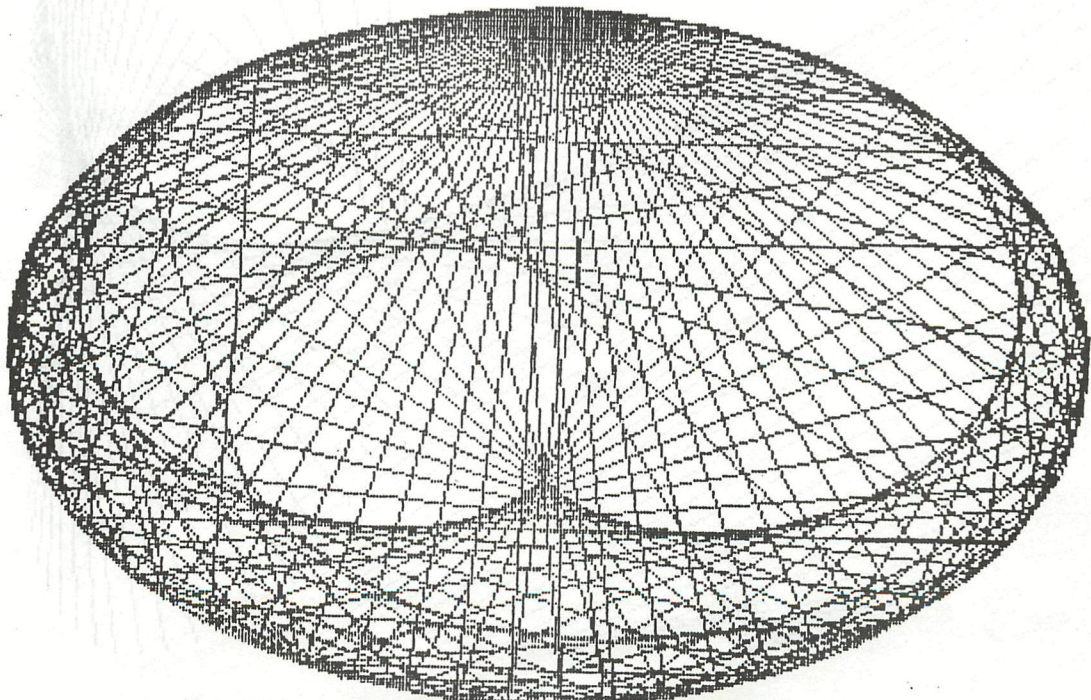
```

1      REM 'GRAPHIC~DEMONSTRATION +++ Markus Sigg 7/81 +++'
5      POKE #FFF5,#10:POKE #131,1
10     CLEAR 3000:P!=2*PI:R=RND(-RND(0)):COLORT 8 2 0 0:MODE 0:
C      PRINT CHR$(12);"
C      DAI-GRAPHIK-DEMONSTRATIONSPROGRAMM":PRINT
20     PRINT "Tastenbelegung:":PRINT
30     PRINT " 'A' Zufallslinien           'K' Rechteckschlangen 1"
32     PRINT " 'B' Kaleidoskop             'L' Rechteckschlangen 2"
34     PRINT " 'C' Ellipsengraphik 1"
36     PRINT " 'D' Ellipsengraphik 2"
38     PRINT " 'E' Lissajous 1"
40     PRINT " 'F' Lissajous 2"
42     PRINT " 'G' Polygone"
44     PRINT " 'H' Sternzeichner"
46     PRINT " 'I' n-Eck-Dreher"
48     PRINT " 'J' Quaderzeichner":PRINT
50     PRINT " 'TAB' andere Zufallsfarben":PRINT " 'SPACE'
C      neuer Start des gesamten Programmes"
60     PRINT " '2,4,6' Graphikmodusaenderung + neuer Start des":
C      PRINT "           jeweiligen Teilprogrammes"
70     PRINT " 'P' Druckerausgabe (MODE 6)":PRINT :PRINT
C      "Welches Teilprogramm wuenschen Sie?";
80     POKE #2C3,0:G=GETC:IF G<65 OR G>76 THEN 80:PRINT CHR$(G):
C      PRINT :PRINT "In welchem Modus soll Programm ";CHR$(G);"
C      laufen?";
90     M=GETC:IF M<>50 AND M<>52 AND M<>54 THEN 90:MM=9:C1=8:C2=1:
C      C3=3:C4=5:GOSUB 60000:ON G-64 GOTO 1000,2000,3000,4000,5000,
C      6000,7000,8000,9000,10000,11000,12000

1000    REM ' Zufallslinien
1010    X1=RND(XMAX):Y1=RND(YMAX):DRAW XO,YO X1,Y1 21:DRAW XO,YO X1,
C      Y1 22:XO=X1:YO=Y1:GOSUB 61000:IF MM=32 THEN 1:GOTO 1010

2000    REM ' Kaleidoskop
2010    C=21+RND(3):X1=RND(XMAX):Y1=RND(YMAX)
2020    DRAW XO,YO X1,Y1 C:DRAW XMAX-XO,YO XMAX-X1,Y1 C:DRAW XO,
C      YMAX-YO X1,YMAX-Y1 C:DRAW XMAX-XO,YMAX-YO XMAX-X1,YMAX-Y1 C:
C      XO=X1:YO=Y1:GOSUB 61000:IF MM=32 THEN 1:GOTO 2010

```

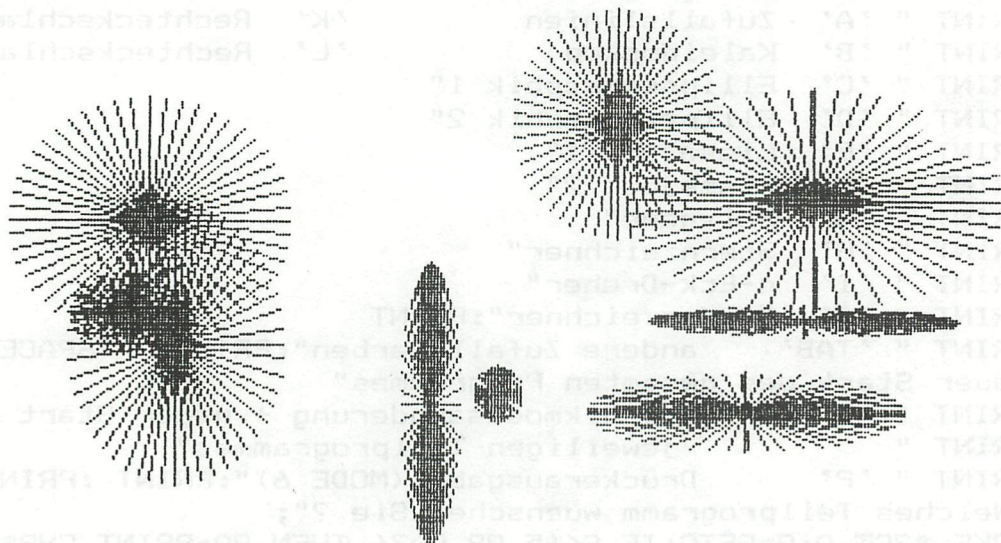


GRAPHICS

```

3000 REM ' Ellipsengraphik 1
3010 FOR I!=0.1 TO P! STEP 0.1:X0=X:Y0=YMAX-1:FOR J!=I! TO P!+I:
C 2 STEP I!:X1=X+X*SIN(J!):Y1=Y+Y*COS(J!):DRAW X0,Y0 X1,Y1 21:
C 2 X1:Y0=Y1
3020 GOSUB 61000:IF MM=M THEN 3010:IF MM=32 THEN 1:NEXT:NEXT:GOTO
C 3010

```



```

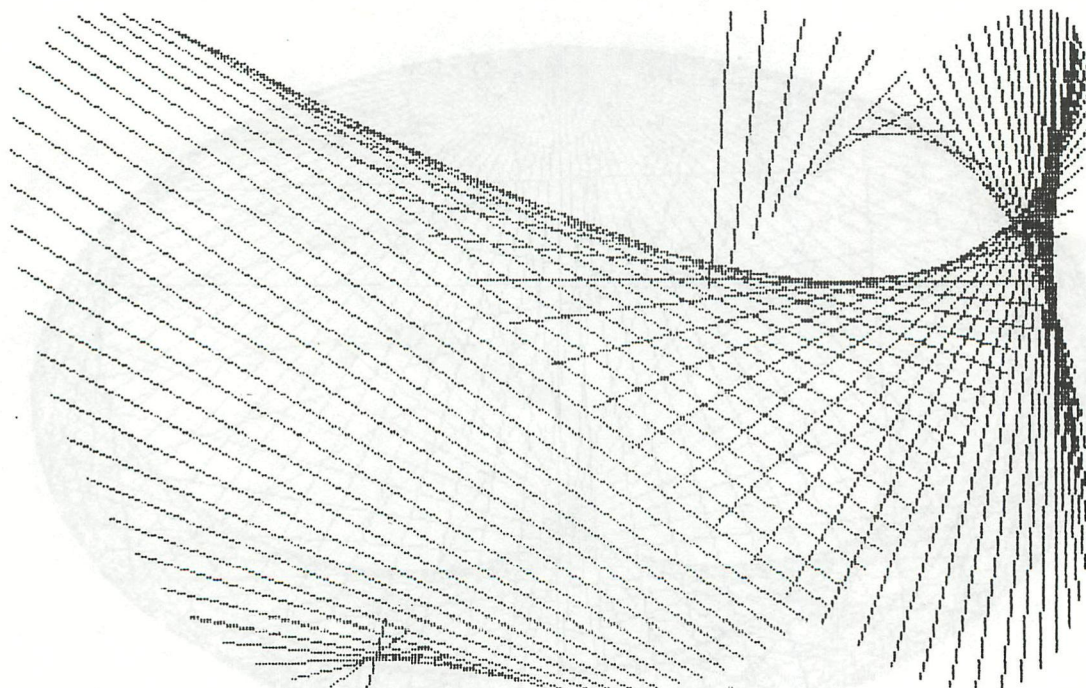
4000 REM ' Ellipsengraphik 2
4010 RX=RND(YMAX/4):RY=RND(YMAX/4):X=RX+RND(XMAX-2*RX):Y=RY+
C RND(YMAX-2*RY):C=21+RND(3)
4020 FOR I!=0 TO PI STEP 0.12:DX=RX*SIN(I!):DY=RY*COS(I!):DRAW X+
C 1 DX,Y+DY X-DX,Y-DY C:GOSUB 61000:IF MM=M THEN 4010:IF MM=32
C THEN 1:NEXT:GOTO 4010

```

```

5000 REM ' Lissajous 1
5010 DIM X(2),Y(2):FOR I=0 TO 2:X(I)=RND(XMAX):Y(I)=RND(YMAX):
C NEXT:FI!=RND(PI):V=RND(X):W=RND(Y):A=RND(10):B=RND(10)
5020 FOR X!=0 TO P! STEP S!:X0=X+V*SIN(A*X!):Y0=Y+W*SIN(B*X!+FI!):
C 1 FOR I=0 TO 2:DRAW X0,Y0 X(I),Y(I) 21+I:NEXT:GOSUB 61000:IF
C MM=M THEN 5010:IF MM=32 THEN 1:NEXT:GOTO 5020

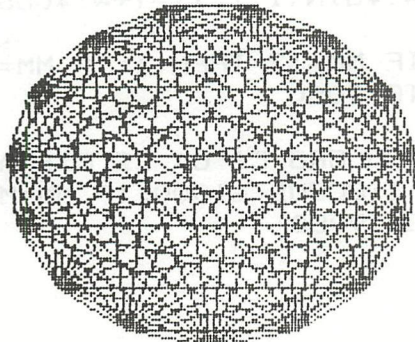
```



GRAPHICS

```

6000 REM ' Lissajous 2
6010 V1!=RND(P!):V2!=RND(P!):V3!=RND(P!):V4!=RND(P!):W1!=RND(P!):
C W2!=RND(P!):W3!=RND(P!):W4!=RND(P!)
6020 FOR I!=0 TO P! STEP S!:DRAW X+X*SIN(I!*V1!+W1!),Y+
C 1 Y*SIN(I!*V2!+W2!) X+X*SIN(I!*V3!+W3!),Y+Y*SIN(I!*V4!+W4!) 21+
C 1 3*I!/P!
6030 GOSUB 61000:IF MM=M THEN 6010:IF MM=32 THEN 1:NEXT:GOTO 6020
    
```

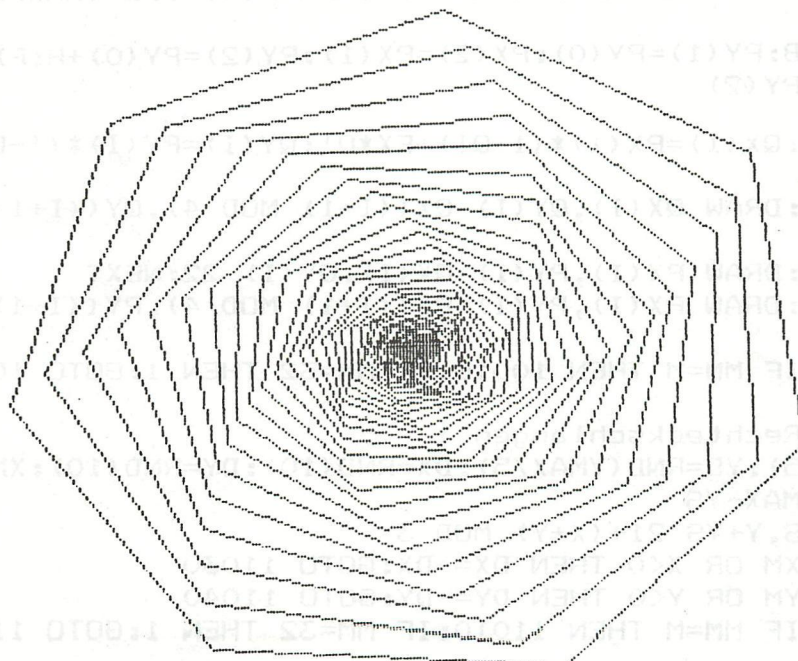


```

7000 REM ' Polygone
7010 INPUT "Zahl der Ecken ";N:IF N<1 THEN 7010
7020 GOSUB 60000:DIM X(N),Y(N):N!=P!/N:FOR I=1 TO N:X(I)=X+
C Y*SIN(I*N!):Y(I)=Y+Y*COS(I*N!):NEXT
7030 FOR I=1 TO N-1:FOR J=I+1 TO N:DRAW X(I),Y(I) X(J),Y(J) 21:
C GOSUB 61000:IF MM=M THEN 7010:IF MM=32 THEN 1:NEXT:NEXT:GOTO
C 7030
    
```

```

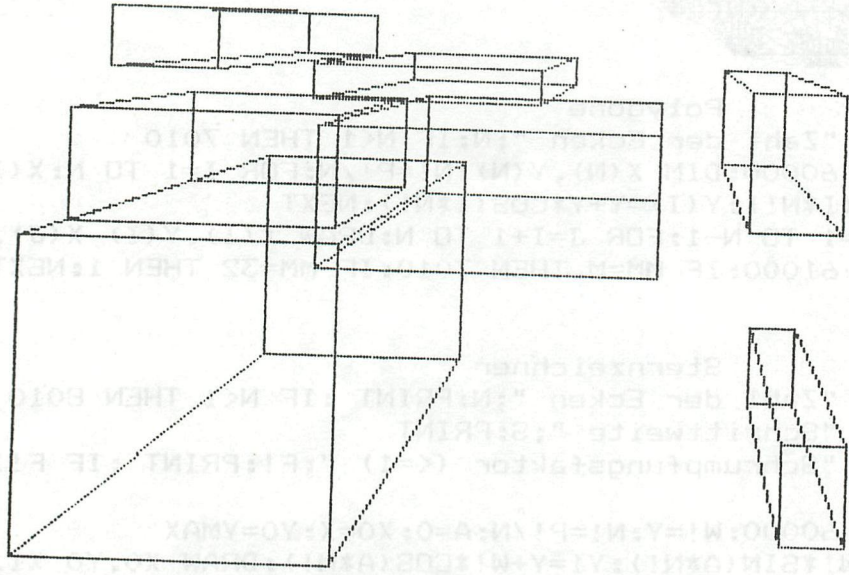
8000 REM ' Sternzeichner
8010 INPUT "Zahl der Ecken ";N:PRINT :IF N<1 THEN 8010
8020 INPUT "Schrittweite ";S:PRINT
8030 INPUT "Schrumpfungsfaktor (<=1) ";F!:PRINT :IF F!>1 THEN
C 8030
8040 GOSUB 60000:W!=Y:N!=P!/N:A=0:X0=X:Y0=YMAX
8050 X1=X+W!*SIN(A*N!):Y1=Y+W!*COS(A*N!):DRAW X0,Y0 X1,Y1 21:X0=
C X1:Y0=Y1:A=(A+S) MOD N:W!=W!*F!:GOSUB 61000:IF MM=M THEN
C 8010:IF MM=32 THEN 1:GOTO 8050
    
```



GRAPHICS

```

9000 REM ' n-Eck-Dreher
9010 INPUT "Zahl der Ecken ";N:PRINT :IF N<1 THEN 9010
9020 INPUT "Drehwinkel (Grad) ";D!:PRINT :D!=D!*PI/180:DD!=D!
9030 INPUT "Schrumpfungsfaktor (<=1) ";F!:PRINT :IF F!>1 THEN
C 9030:PRINT "( 'RETURN' = Zeichnen beenden )";:WAIT TIME 100
9040 GOSUB 60000:W!=Y:S!=P!/N
9050 XO=X+W!*SIN(DD!):YO=Y+W!*COS(DD!):C=21+(C+1) MOD 3:FOR I!=
C 1 DD!+S! TO P!+DD! STEP S!:X1=X+W!*SIN(I!):Y1=Y+W!*COS(I!):
C 1 DRAW XO,YO X1,Y1 C:XO=X1:YO=Y1
9060 1 GOSUB 61000:IF MM=M THEN 9010:IF MM=32 THEN 1:IF MM=13 THEN
C 9070:NEXT:W!=W!*F!:DD!=DD!+D!:GOTO 9050
9070 FOR I=20 TO 2 STEP -1
9080 1 COLORG C1 C2 C3 C4:WAIT TIME I:COLORG C1 C4 C2 C3:WAIT TIME
C 1 I:COLORG C1 C3 C4 C2:WAIT TIME I:GOSUB 61000:IF MM=M THEN
C 1 9010:IF MM=32 THEN 1:IF I=2 THEN 9080
9090 NEXT
    
```



```

10000 REM ' Quaderzeichner
10010 FX=RND(XMAX):FY=RND(YMAX):DIM PX(3),PY(3),QX(3),QY(3)
10020 PX(0)=RND(0.9*XMAX):PY(0)=RND(0.9*YMAX)
10030 B=RND(XMAX-PX(0))/2+XMAX/20:H=RND(YMAX-PY(0))/2+YMAX/20:Q!=
C RND(0.2)+0.2
10040 PX(1)=PX(0)+B:PY(1)=PY(0):PX(2)=PX(1):PY(2)=PY(0)+H:PX(3)=
C PX(0):PY(3)=PY(2)

10050 FOR I=0 TO 3:QX(I)=PX(I)*(1-Q!)+FX*Q!:QY(I)=PY(I)*(1-Q!)+
C FY*Q!:NEXT
10060 FOR I=0 TO 3:DRAW QX(I),QY(I) QX((I+1) MOD 4),QY((I+1) MOD
C 4) 21:NEXT
10070 FOR I=0 TO 3:DRAW PX(I),PY(I) QX(I),QY(I) 22:NEXT
10080 FOR I=0 TO 3:DRAW PX(I),PY(I) PX((I+1) MOD 4),PY((I+1) MOD
C 4) 23:NEXT
10090 GOSUB 61000:IF MM=M THEN 10010:IF MM=32 THEN 1:GOTO 10020

11000 REM ' Rechteckschlangen 1
11010 XS=RND(XMAX/5):YS=RND(YMAX/5):DX=RND(10):DY=RND(10):XM=
C XMAX-XS:YM=YMAX-YS
11020 FILL X,Y X+XS,Y+YS 21+(X+Y) MOD 3
11030 X=X+DX:IF X>XM OR X<0 THEN DX=-DX:GOTO 11030
11040 Y=Y+DY:IF Y>YM OR Y<0 THEN DY=-DY:GOTO 11040
11050 GOSUB 61000:IF MM=M THEN 11010:IF MM=32 THEN 1:GOTO 11020
    
```



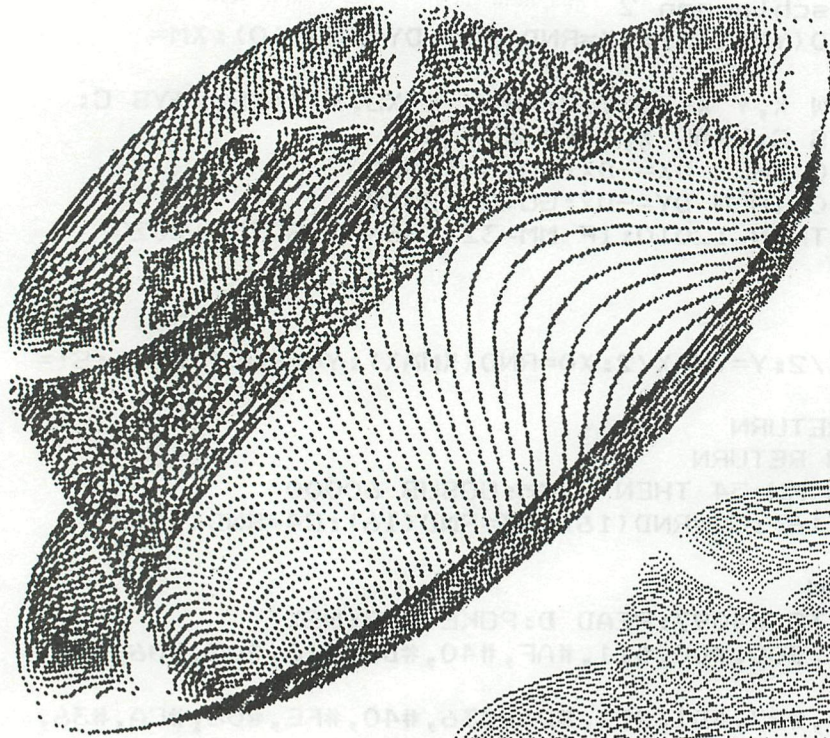
```

12000 REM ' Rechteckschlangen 2
12010 XS=RND(XMAX/5):YS=RND(YMAX/5):DX=RND(10):DY=RND(10):XM=
C XMAX-XS:YM=YMAX-YS
12020 C=21+(X+Y) MOD 3:DRAW X,Y X+XS,Y C:DRAW X+XS,Y X+XS,Y+YS C:
C DRAW X+XS,Y+YS X,Y+YS C:DRAW X,Y+YS X,Y C
12030 X=X+DX:IF X>XM OR X<0 THEN DX=-DX:GOTO 12030
12040 Y=Y+DY:IF Y>YM OR Y<0 THEN DY=-DY:GOTO 12040
12050 GOSUB 61000:IF MM=M THEN 12010:IF MM=32 THEN 1:GOTO 12020
60000 IF M=50 THEN MODE 2
60010 IF M=52 THEN MODE 4
60020 IF M=54 THEN MODE 6
60030 IF PFL=0 THEN X=XMAX/2:Y=YMAX/2:X0=RND(XMAX):Y0=RND(YMAX):S!=
C 5E-2
60040 COLORG C1 C2 C3 C4:RETURN
61000 MM=GETC:IF MM=0 THEN RETURN
61010 IF MM=50 OR MM=52 OR MM=54 THEN M=MM:GOSUB 60000
61020 IF MM=9 THEN C1=RND(16):C2=RND(16):C3=RND(16):C4=RND(16):
C COLORG C1 C2 C3 C4
61030 IF MM<>80 THEN RETURN
62000 RESTORE:FOR K=#4000 TO #4097:READ D:POKE K,D:NEXT
62010 DATA #C5,#0E,#F8,#21,#00,#00,#11,#AF,#40,#D5,#16,#07,#06,#80,
C #C5,#79
62020 DATA #82,#4F,#D5,#EF,#27,#D1,#C1,#DA,#36,#40,#FE,#08,#CA,#36,
C #40,#79
62030 DATA #FE,#FC,#C2,#2B,#40,#7A,#FE,#04,#DA,#36,#40,#AF,#3E,#80,
C #5A,#1F
62040 DATA #1D,#C2,#2F,#40,#80,#47,#15,#C2,#0E,#40,#78,#D1,#12,#13,
C #23,#7A
62050 DATA #FE,#42,#C2,#09,#40,#11,#AF,#40,#21,#B0,#40,#06,#01,#1A,
C #BE,#C2
62060 DATA #5A,#40,#04,#CA,#67,#40,#23,#C3,#4E,#40,#78,#FE,#03,#DA,
C #79,#40
62070 DATA #FE,#0D,#C2,#67,#40,#05,#2B,#3E,#1C,#CD,#94,#DD,#78,#CD,
C #94,#DD
62080 DATA #1A,#CD,#94,#DD,#E5,#D1,#C3,#81,#40,#D5,#E1,#7E,#CD,#94,
C #DD,#13
62090 DATA #23,#23,#7C,#FE,#42,#C2,#4B,#40,#3E,#0D,#CD,#94,#DD,#3E,
C #F9,#81
62100 DATA #4F,#FE,#F5,#C2,#03,#40,#C1,#C9
63000 C1=8:C2=8:C3=8:C4=8
63010 COLORG 8 0 8 8:GOSUB 63100:C2=C:GOSUB 63200
63020 COLORG 8 8 0 8:GOSUB 63100:C3=C:GOSUB 63200
63030 COLORG 8 8 8 0:GOSUB 63100:C4=C:GOSUB 63200
63040 PRINT CHR$(12);"Drucken <J/N> ?"
63050 G=GETC:IF G=0 THEN 63050
63060 IF G=74 THEN POKE #131,0:PRINT CHR$(#14);CHR$(8):PFL=1:GOSUB
C 60000:PFL=0:CALLM #4000:POKE #131,1
63070 RETURN
63100 PRINT CHR$(12);"Diese Linien drucken <J/N> ?":C=8
63110 G=GETC:IF G=0 THEN 63110:IF G=74 THEN C=0
63120 RETURN
63200 PRINT CHR$(12);"Bisheriges Gesamtbild ^"
63210 COLORG C1 C2 C3 C4:WAIT TIME 50:RETURN

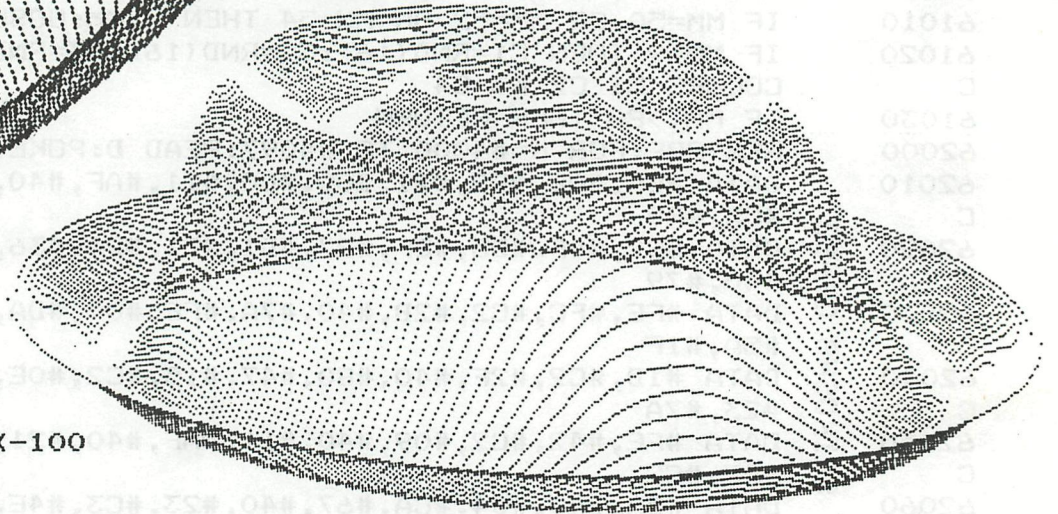
```

SCREENCOPY SEIKOSHA GP-80

SCREENCOPY OF SEIKOSHA GP-80 from Daniel Thies



SEIKOSHA GP-80



MX-100



SCREENCOPY OF SEIKOSHA GP-80 from Daniel Theys

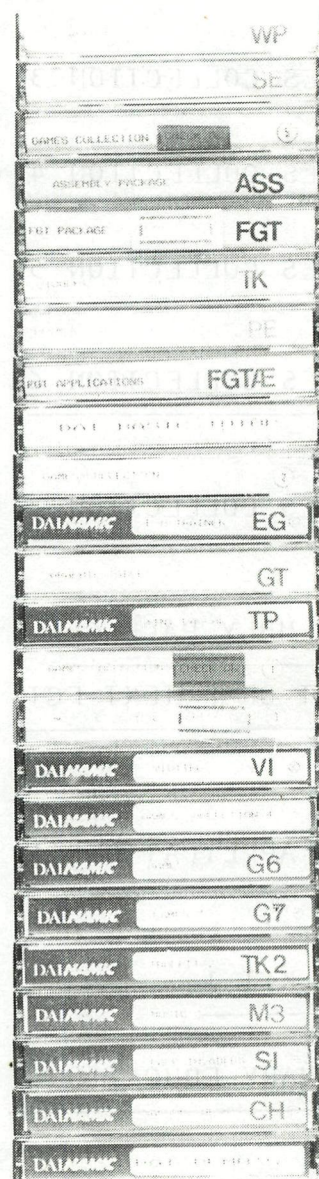
CATALOGUS LIBRARY

<u>GAMES COLLECTION 1</u>	Yathzee, Awari, Submarine, Kanonspel, Othello, Startrek, Reactietest, Lunar landing, Vier op een rij
<u>GAMES COLLECTION 2</u>	Surround, Break-out, Amazing, Kim Cache, Mastermind, Geitenspel, Space invaders, Hannibal 2000, Towers of Hanoi
<u>GAMES COLLECTION 3</u>	Backgammon, Barricade, Moeras, Snake, Poisoned cake, Invasion, Robots, Traffic test
<u>GAMES COLLECTION 4 (ml)</u>	Football, The Car, Break-out, Gompy, Space invaders
<u>GAMES COLLECTION 5</u>	Atomic Attack, Zich-zagger, Rat maze, Dom dam, Tic-tac-toe, Life(ml), Katz&Mause
<u>GAMES COLLECTION 6</u>	Rubic Cube(ml), 3-dim Maze, Bridge-building, Gomuku, Verborgen tekens, City Bomber, Space Game
<u>GAMES COLLECTION 7</u>	Blackjack, Androidenkaefig, Booby-traps, Communicerende vaten, Carpenters Mystery, Breinkraker, Auto-course
<u>ASSEMBLY PACKAGE</u>	Assembler-Editor-Loader-Disassembler
<u>FAST GRAF TEXT FGT</u>	Table Creator, Word Game, Pictures+Minuscules, Shadow, DAI Character set, Greece, Trigisch, Math Symbols, Morse, Russian, Heavy style, Computer characters + source FGT
<u>FGT APPLICATIONS 1</u>	Horner alg., 555 Quick design, Math comp, TV-tennis, Superwurm, Mastermind, Clock training, FGT-paddles
<u>TOOLKIT 1</u>	Renumber, Format listing, Data statements generator, Labeljumps + sources
<u>PRIMARY EDUCATION 1</u>	Universal Math-trainer, Clock, Math comp, Technical reading, visual discrimination, missing letter, pictographic reading
<u>SECONDARY EDUCATION 1</u>	Matrixproduct, 3X3 Determinant, Triangle-Algorithm, Volume of box, Area-computing, $Y=A.SIN(B.X+C)$, Examen sine, cylinder, Horner alg., Darboux-sums
<u>SECONDARY EDUCATION 2</u>	Bewerkingen met functies, functie-onderzoek, 1e en 2e afgeleide, getallenleer+animatie, demo getallenleer, oefeningen getallenleer, Bord van Galton, meetkundige plaats, staande golf, lopende golf
<u>TOOLKIT 2</u>	BASIC with abbreviations, Disassembler-relocator, screen copy MX 80/MX 100/SEIKOSHA auto-linenummer, basicode, Cross reference

CATALOGUS LIBRARY

Nieuwe prijzen vanaf 15/2/82 New prices

TAPE	PRIJS	CLUB-PRIJS
Games Collection 1	600	400
Games Collection 2	600	400
Games Collection 3	600	400
Games Collection 4	1200	800
Games Collection 5	600	400
Games Collection 6 ***	900	750
Games Collection 7 ***	900	750
Assembly Package	2250	1750
Fast Graf Text (FGT)	1400	1000
FGT Applications 1	1400	1000
Toolkit 1	1400	1000
Primary Education 1	1400	1000
Secondary Education 1	1400	1000
Secondary Education 2 ***	1400	1000
Wordprocessor	1400	1000
Mailing List	1400	1000
Graphic Tablet	1400	1000
Music Collection 1	500	300
Music Collection 2	500	300
Music Collection 3 ***	500	300
VIDITEL ***	900	750
TINY PASCAL ***	1400	1000
English-German trainer ***	1400	1000
DAI-DEMO'S + BASIC TUTOR ***	1000	500
SARGON CHESS ***	2000	1500
SPACE INVADERS ***	1000	800
Toolkit 2 ***	1400	1000



*** new package

all prices include VAT & Mailing, Belgian francs

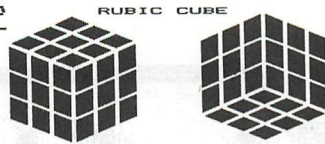
all packages also available on DCR (+100 FR /tape)

GAMES COLLECTION 6

RUBIC'S CUBE

author : F.Druijff

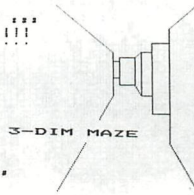
A beautiful simulation of the mystery cube, with full colour graphics in machine language. Let him turn random, from keyboard or from a table ... This is the best cube available on microcomputer !!!



3-DIM MAZE

author : J.Visser

Try to escape out of the maze ... up to 256 rooms. This is a superfast maze with many options.



BRIDGE BUILDING

author : J.Marchand

Help the Joellies cross the river by manipulating 4 bridges. Fast and frustrating ,beautiful sound & graphics.

GOMUKU

author : R.Sip

Try to beat the computer's strategy ...

VERBORGEN TEKENS (HIDDEN SHAPES)

author : G.Goethals

Watch carefully and try to remember



BOMBER

author : J.Van Dunne

Try to hit the city-buildings, splendid graphics

SPACE GAME

author : J.Visser

Try to destroy the enemy ships, 3 play levels

note : G6 & G7 exclusive copyright for Germany :

TOOLKIT 2

CROSS REFERENCE

Variables at10s + complete line reference

BASIC WITH ABBREVIATIONS

All BASIC keywords with 2 characters only ! You can use your own list of key-words

DISASSEMBLER-RELOCATOR

SCREEN COPY ROUTINES

Screen copy routines for MX-80 II, MX82, MX100, SEIKOSHA

AUTO-LINENUMBER

Automatic linenummer generation step 10, with on/off switch

BASICODE

The complete BASICODE-routines + sample files

GAMES COLLECTION 7

BLACKJACK

author : J.Visser

The casino card game, up to 3 players + banc Realistic presentation of cards and symbols Get your money ready...

ANDROIDENKAEFIG

author : K.Peter

The androids are dangerous, look out for high tension. Animated graphics, realistic sound effects.

COMMUNICERENDE VATEN

author : G.Goethals

Which tank is not OK ? Watch and think, time is limited !!!

CARPENTERS MYSTERY

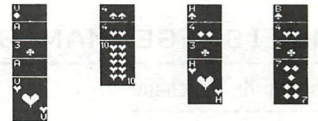
author : W.Hermans

Solve the puzzle ... or let the computer do it !

BREINKRAKER

author : W.Nijland

Try to arrange the blocks, misleading simple ...



COURSE

author : Dufour & co

Drive your car across the screen avoid to crash !

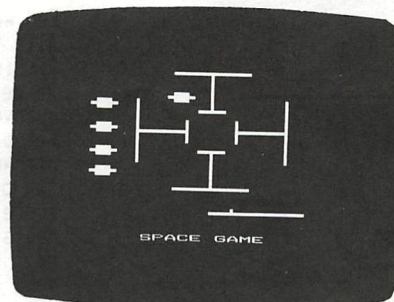
BLACKJACK

DAI-CLUB Germany
c/o D.Sommer
Overwegstr 30 City House
4650 GELSENKIRCHEN



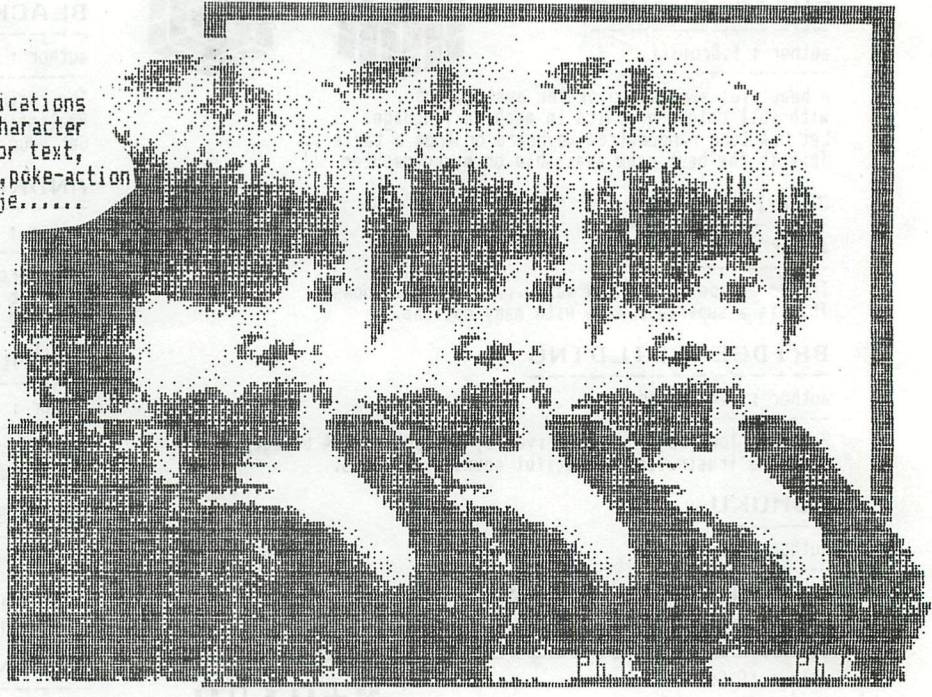
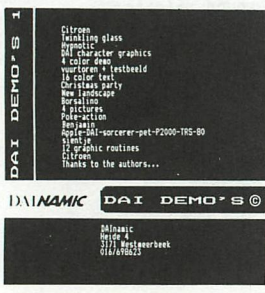
MUSIC COLLECTION 3

Radinerie, Pop corn, Slowrock, Borsalino, Zweistimmige invention, Presto sonate



DAI-DEMO'S + BASIC TUTOR

20 beautiful demoprograms showing all kind of applications for DAIPC : Citroen, Twinkling glass, Hypnotic, DAI character graphics, 4 color demo, vuurtoren + testbeeld, 16 color text, Christmas party, new landscape, borsalino, 4 pictures, poke-action benjamin, apple-DAI-sorcerer-pet-P2000-TRS-80, sientje..... special for beginners, super value !!!!

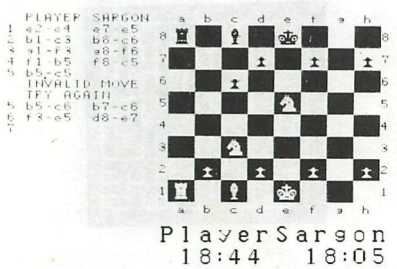


ENGLISH-GERMAN GRAMMAR TRAINER

author : Mr Marchand

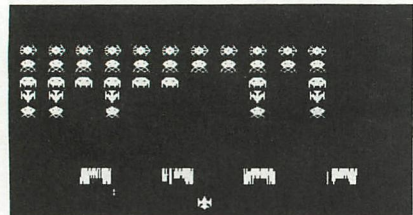
Learn english,german or both !
A C-60 tape was needed to carry 5 programs with hundreds of exercices,
more details in Newsletter 8.

DAI PERSONAL COMPUTER USERS CLUB



SARGON-CHESS

You have been waiting for it : here it is !!
More than 8K of machine language, 6 levels of play,
plays black or white, with chess clock, castling, position analysis..
and the best chess graphics you have ever seen



SPACE INVADER

The DAI S U P E R INVADER.....
features superb mode 6-graphics, nail-biting tension and
hilarious antics by the moon creatures ...

VIDITEL

Een wereld van informatie met dit programma, de telefoonmodem ... en uw DAI.
Alle VIDITEL-faciliteiten komen op uw DAI tot leven.
Dit is versie 3.0, met meer dan 30 nieuwe commando's.
De gevraagde plaatjes kunnen bewaard worden en later weer
opgevraagd. (+ 8 sec per plaatje op cassette).
Het programma wordt geleverd met een 50-tal voorbeeldplaatjes,
nieuwe collecties zullen regelmatig verkrijgbaar zijn.
Kan ook zonder modem gebruikt worden om plaatjes te creeren, op te slaan
en te bekijken.
Meets the german and english standard for BILDSCHIRMTEXT & VIEWDATA.
noot : wie de vorige versie kocht op de HCC-dag kan deze ruilen
voor versie 3.0 (de tape opsturen als bewijs)



1. Bewerkingen met functies

Dit programma laat toe twee functies met domein R in te voeren en biedt nadien volgend menu aan:

1. constructie van $f_1 + f_2$ en $f_1 - f_2$
2. constructie van $f_1 \cdot f_2$ en $f_1 : f_2$
3. constructie van $\sqrt{f_1}$ en $\sqrt{f_2}$.

De basisfuncties f_1 en f_2 worden ook steeds getekend zodat de verbanden tussen f_1 en f_2 en de bewerkingen hierop van het scherm kunnen worden afgelezen.

De demo-functies zijn: $f_1: y = \frac{x}{2} + 2$ en $f_2: y = 2 - x$

Het standaard-interval is : X-as: $[-6, 6]$ Deze intervallen kunnen vanuit
Y-as: $[-5, 5]$ het programma gewijzigd worden.

Met space-bar terug naar menu.

2. Functie-onderzoek: didactisch hulpprogramma

Dit programma laat toe een functie in te voeren en vraagt nadien naar het domein en de asymptoten. De leerling voert zijn antwoord in. Het programma schetst de grafiek van de ingevoerde functie en houdt rekening met de door de gebruiker ingevoerde informatie in verband met asymptoten en domein. Indien hierin fouten zitten.... Op deze wijze is het programma een didactisch hulpprogramma bij de studie van functies.

De demo-functie is $y = \cos x + \frac{1}{3} \cos 3x + \frac{1}{5} \cos 5x + \frac{1}{7} \cos 7x$

Assenstel met ijk zoals in vorig programma te wijzigen.

3. Functie met 1e en 2e afgeleide

Dit programma laat toe een functie in te voeren en tekent terzelfder tijd de grafiek van deze functie en van haar eerste en tweede afgeleide functie.

De demo-functie is:

$$y = \frac{(x+5)(x+3)(x-6)(x-1)(x-3)}{162}$$

4. Getallenleer + animatie

Speelse aanbrenge van enkele getallenverzamelingen en hun deelverzamelingen.

Als de animatie ophoudt:space-bar. Het programma eindigt met 6 oefeningen op de geleerde theorie. Met sound.

5. Demo getallenleer

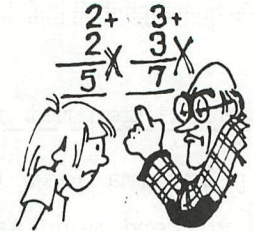
Oefeningen op de bewerkingen met de verzamelingen: $\mathbb{N}, \mathbb{Z}, \mathbb{N}_0, \mathbb{Z}_0, \mathbb{Z}_0^+, \mathbb{Z}^-, \emptyset, \{a\} \dots$

6. Oefeningen getallenleer: uitbreiding van vorig programma

Dit programma laat de leraar toe zelf oefeningen te construeren.

7. Bord van Galton : mode 6

simulatie van het bord van Galton in mode 6
 aantal vallende kogeltjes ingeven; druk op TAB-toets
 teller geeft aantal gevallen kogeltjes aan
 na laatste kogel: met space-bar naar frequentieverdeling
 space-bar: nieuwe run



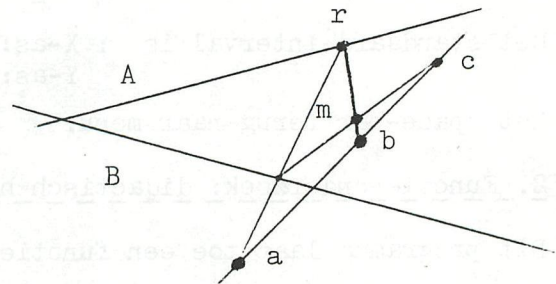
8. Bord van Galton : mode 4

Analoog met programma 7 - zonder FGT-programma - geen frequentieverdeling

9. Demo: meetkundige plaats

Punt per punt wordt de meetkundige plaats van m gegenereerd. Vooraf 5-maal space-bar; nadien automatisch.

Gegeven : | twee rechten A en B
 | 3 collineaire punten : a, b en c
 | r loopt op A



Gevraagd: meetkundige plaats van m

10. staande golf

Simulatie van een staande golf: aantal buiken in te stellen tussen 1 en 4

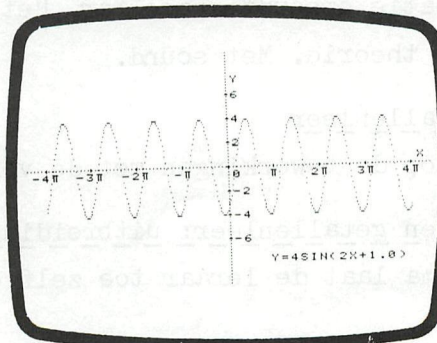
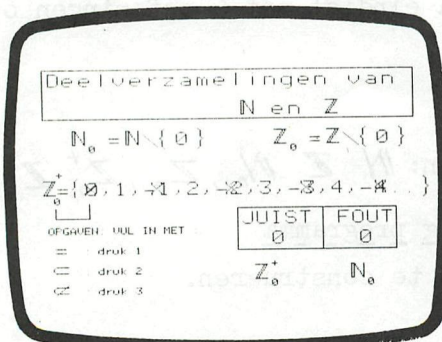
11. Lopende golf

Simulatie van een lopende golf

Alle programma's uitgezonderd nr.8, nr.10 en nr.11 maken gebruik van het FGT-programma met de tabel met de wiskundige symbolen. Dit programma staat vooraan op de tape en moet worden ingeladen vanuit UTILITY.

Indien, nadat programma 10 of 11 gelopen heeft opnieuw een programma wordt uitgevoerd dat gebruik maakt van de FGT-tabel moet deze vooraf opnieuw worden ingelezen, omdat de programma's 10 en 11 een deel van deze tabel vernietigen.

Met dank aan de heren B. Van Rompaey en K. Esveld



CONVERSIONS

```

10 PRINT CHR$(12):PRINT "MENU:      CONVERSIONS":PRINT
12 PRINT "Pour choisir votre programme, "
14 PRINT "appuyez sur la touche indiquee et puis sur RETURN. ":PRINT
20 PRINT "CONVERSIONS: DEGRES CENTIGRADES-DEGRES FAHRENHEIT 1"
25 PRINT "                DEGRES FAHRENHEIT-DEGRES CENTIGRADES 2"
30 PRINT "                MONNAIES ETRANGERES                    3"
35 PRINT "                UNE BASE QUELCONQUE EN BASE 10          4"
40 PRINT "                BASE 10 EN UNE BASE QUELCONQUE          5"
70 INPUT A
75 ON A GOTO 100,200,300,500,750
80 PRINT "Ce programme n'existe pas. Recommencez. ":GOTO 70
100 PRINT CHR$(12):PRINT "CONVERSION DEGRES CENTIGRADES-DEGRES FAHRENHEIT":PRI
    NT
110 INPUT "Temperature en degres Centigrades: ";C$:PRINT
120 F%=9*C$/5+32
130 PRINT "      ";C$; " DEGRES CENTIGRADES =";F%; " DEGRES FAHRENHEIT":PRINT
140 B=1.0:GOTO 1000
200 PRINT CHR$(12):PRINT "CONVERSION DEGRES FAHRENHEIT-DEGRES CENTIGRADES":PRI
    NT
210 INPUT "Temperature en degres Fahrenheit: ";F$:PRINT
220 C%=(F%-32)*5/9
230 PRINT "      ";F$; " DEGRES FAHRENHEIT =";C%; " DEGRES CENTIGRADES":PRINT
240 B=2.0:GOTO 1000
300 PRINT CHR$(12):PRINT "CONVERSION DE MONNAIES ETRANGERES":PRINT
310 INPUT "Introduisez le cours du jour: ";C:PRINT
320 INPUT "      Somme a convertir: ";S:PRINT
330 PRINT "      Unite: F pour FRANCS"
340 INPUT "      ET pour MONNAIES ETRANGERES ";U$:PRINT
350 IF U$="F" THEN 380
360 IF U$="ET" THEN 400
370 PRINT "Precisez F ou ET uniquement":GOTO 340
380 ET=S/C
390 PRINT S; " F =";ET; " ET":B=3.0:GOTO 1000
400 F=S*C
410 PRINT S; " ET =";F; " F":B=3.0:GOTO 1000
500 PRINT CHR$(12):PRINT "CONVERSION BASE 10 EN UNE BASE QUELCONQUE":PRINT
510 DIM A(15.0)
520 B$="0123456789ABCDEF"
530 INPUT "Donnez la base: ";C:PRINT
540 INPUT "Donnez le debut et la fin de la table: ";D:INPUT F:PRINT
550 FOR I%=D TO F:PRINT
560   GOSUB 630
570   PRINT I%:TAB(12);
580   FOR L=J TO 1.0 STEP -1.0
590     PRINT MID$(B$,A(L),1);
600   NEXT L
610 NEXT I%
620 PRINT :B=4.0:GOTO 1000
630 I1=I%
640 J=1.0
650 Q=INT(I1/C)
660 R=I1-Q*C
670 I1=Q
680 A(J)=R
690 J=J+1.0
700 IF Q>=C THEN 650
710 A(J)=Q
720 RETURN
750 PRINT CHR$(12.0):PRINT "CONVERSION D'UNE BASE QUELCONQUE EN BASE 10":PRINT

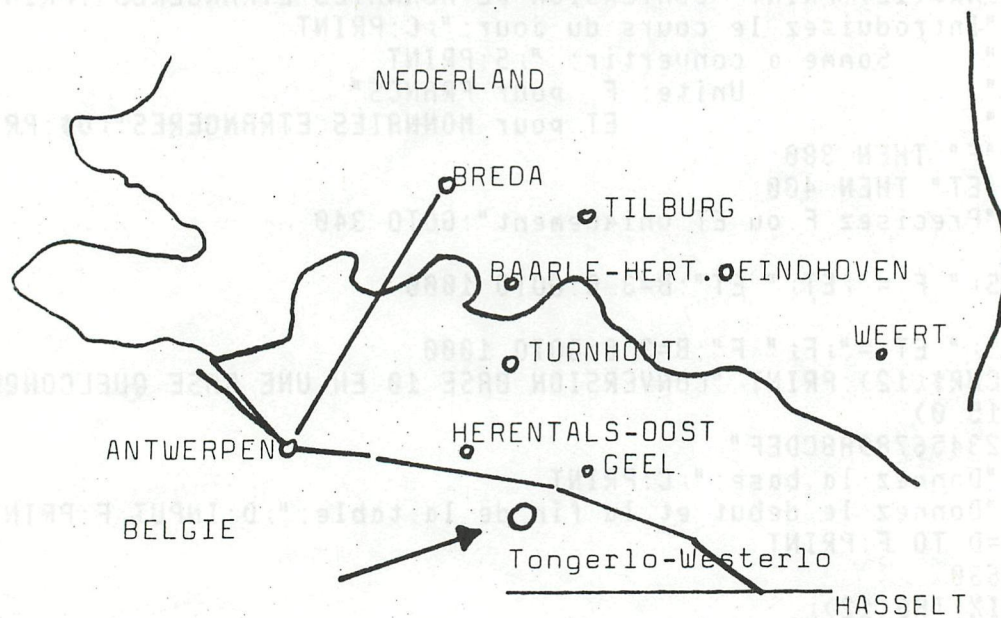
```

```

760 INPUT "Nombre a convertir:";U$:PRINT
770 INPUT "Base utilisee:";C:PRINT
780 N=LEN(U$)
790 D=0.0:R=0.0:N=N-1.0:N1=N:B$="0123456789ABCDEF"
800 FOR L=0.0 TO N
810 M$=MID$(U$,N1,1)
820 N1=N1-1.0
830 FOR J=0.0 TO 15.0
840 IF M$=MID$(B$,J,1) THEN M=J:D=1.0
850 NEXT J
860 IF M<C AND D=1.0 THEN 880
870 PRINT "Valeur fausse. Donnez de nouvelles valeurs":GOTO 760
880 P=M*C^L
890 R=R+P
900 NEXT L
910 PRINT "Valeur decimale=";R
920 B=5.0:GOTO 1000
1000 INPUT " ENCORE UNE CONVERSION (OUI/NON)";A$
1010 PRINT :IF A$="OUI" THEN 1040
1020 IF A$="NON" THEN 10
1030 PRINT " REPONDEZ OUI OU NON":GOTO 1000
1040 ON B GOTO 110,210,310,500,760
*

```

¹⁰
¹⁰
zaterdag ~~11~~ april DAInamic-bijeenkomst zaterdag ~~11~~ april



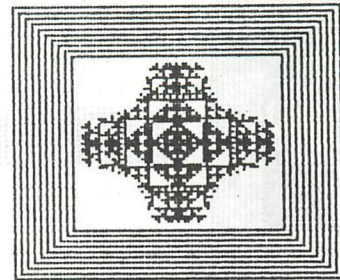
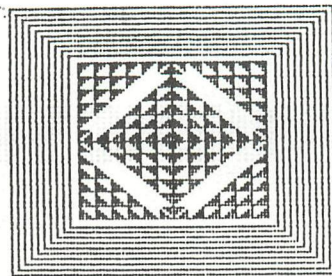
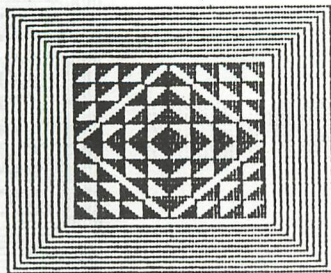
Enkele reistips:

vanuit OOST- en WESTVLAANDEREN:autostrade tot ANTWERPEN,dan AUTOSTRAD E ANTWERPEN-HASSELT,afrit HERENTALS-OOST ,dan richting WESTERLO.

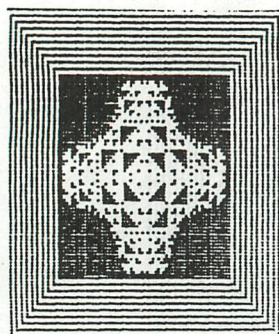
vanuit LIMBURG:autostrade tot GEEL-WEST,dan richting WESTERLO.

vanuit BRABANT,zuidelijk BELGIE:richting LEUVEN,AARSCHOT,WESTERLO. NEDERLAND

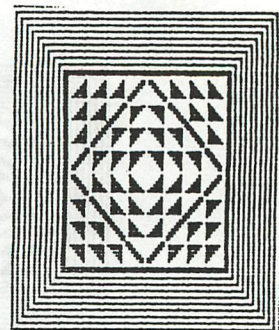
over BREDA:autobaan BREDA-ANTWERPEN,autobaan ANTWERPEN-HASSELT, afrit HERENTALS-OOST(OLEN),dan richting WESTERLO
over EINDHOVEN:TURNHOUT-GEEL-WESTERLO



```
100 REM *** DRIEHOEKSVORMEN
110 REM *** GESCHREVEN DOOR : CREATIVE COMPUTING
120 REM ***** MORRISTOWN, NEW JERSEY
130 REM **** GEWIJZIGD DOOR : CORNEEL DE BONT
140 REM ***** 24-11-1981
150 MODE 0:PRINT CHR$(12):COLORT 5 15 5 5:POKE #75,32
160 POKE #BBBF,#5A:CURSOR 0,15:PRINT "DRIEHOEKSVORMEN"
170 CURSOR 8,13:PRINT "*****"
180 PRINT TAB(8);"* *- * - *- *- *- *- *- *- *- *- *- *"
190 PRINT TAB(8);"* WILLEKEURIGE GRAFISCHE VOORSTELLINGEN *"
200 PRINT TAB(8);"* >>>****=====*****<<<< *"
210 PRINT TAB(8);"* GEBASEERD OP DIVERSE DRIEHOEKSVORMEN. *"
220 PRINT TAB(8);"* *- *- *- *- *- *- *- *- *- *- *- *- *"
230 PRINT TAB(8);"*****"
240 CLEAR 8000
250 DIM P(32.0,32.0)
260 FOR B1=1.0 TO 30.0:FOR B2=1.0 TO 30.0
270 P(B1,B2)=0.0:NEXT B2:NEXT B1
280 CURSOR 10,5:PRINT "DRUK OP DE SPATIEBALK OM TE BEGINNEN ";
290 IF GETC<>32.0 THEN 290
300 MODE 4A:COLORT 12 0 0 0:COLORG 12 9 3 0:XX=80.0:YY=50.0
310 FOR X=50.0 TO 32.0 STEP -2.0:FILL XX-X,YY-X XX+X,YY+X 0
320 FILL XX-X+1,YY-X+1 XX+X-1,YY+X-1 12:NEXT X
330 PRINT CHR$(12):FILL XX-30,YY-30 XX+30,YY+30 12
340 PRINT "DIT PROGRAMMA GEEFT U DIVERSE TEKENINGEN, ALLEN GEBASEERD"
350 PRINT "OP DRIEHOEKSVORMEN. WELKE REPETERENDE BAZISVORM KIEST U ?"
360 PRINT "ENKEL, DUBBEL OF VIERVOUDIG (E, D, V) ? ";:POKE #75,95
370 G=GETC:WAIT TIME 3:IF G=0.0 THEN 370
380 IF G=ASC("E") THEN O=1.0:O$="ENKELE":GOTO 420
390 IF G=ASC("D") THEN O=2.0:O$="DUBBELE":GOTO 420
400 IF G=ASC("V") THEN O=3.0:O$="VIERVOUDIGE":GOTO 420
410 GOTO 370
420 PRINT CHR$(12)
430 CURSOR 10,3:PRINT "U KOOS DUS VOOR ";O$;" DRIEHOEKSVORMEN."
440 CURSOR 10,2:PRINT "HOEVEEL TIENDE WILT U ZWART LATEN (2-9)";
450 G=GETC:IF G<ASC("2") OR G>ASC("9") THEN 450
460 Q=G-48.0:Q%=INT(Q):PRINT Q%:POKE #75,32
470 T=30.0
480 ON O GOTO 510,720,860
490 REM *** CREER EN PRINT EEN DRIEHOEKSVORM
500 REM *** BOUW EEN ENKELE DRIEHOEK OPTIE 1
510 CURSOR 31,1:PRINT "30"
520 FOR R=1.0 TO T:FOR C=1.0 TO T
530 IF R=1.0 OR C=1.0 THEN 550
540 P(R,C)=P(R,C-1.0)+P(R-1.0,C):GOTO 560
550 P(R,C)=1.0
560 NEXT C
570 CURSOR 30,0:R%=INT(R):PRINT R%:NEXT R
580 REM *** PRINT TEKENING UIT
590 FOR R=1.0 TO T:FOR C=1.0 TO T
600 IF P(R,C)=0.0 THEN 640
610 IF (P(R,C)/Q)=INT(P(R,C)/Q) THEN 640
620 DOT XX+C-1,YY+R-1 0:DOT XX-C+1,YY+R-1 0
630 DOT XX+C-1,YY-R+1 0:DOT XX-C+1,YY-R+1 0
640 NEXT C:NEXT R
```



IN I EENENSVORMEN



```
650 CURSOR 0,1:PRINT SPC(59):CURSOR 0,0:PRINT SPC(59);
660 CURSOR 10,0:POKE #75,95
670 PRINT "WILT U NOG EEN TEKENING ZIEN (J/N) ?";
680 G=GETC:WAIT TIME 3:IF G=0.0 THEN 680
690 IF G=ASC("J") THEN POKE #75,32:GOTO 330
700 IF G=ASC("N") THEN PRINT CHR$(12):PRINT :PRINT TAB(22);:END
710 GOTO 680
720 REM *** BOVEN LINKER HELFT OPTIE 2
730 CURSOR 21,1:PRINT "30";TAB(41);"1":Z=T:N=Z
740 FOR R=1.0 TO N:FOR C=1.0 TO Z-1.0
750 IF (R-1.0)*(C-1.0)=0.0 THEN 770
760 P(R,C)=P(R,C-1.0)+P(R-1.0,C):GOTO 780
770 P(R,C)=1.0
780 NEXT C:Z=Z-1.0:CURSOR 20,0:R%=INT(R):PRINT R%;:NEXT R
790 REM *** ONDER RECHTER HELFT OPTIE 2
800 Z=N:N=2.0:FOR R=Z TO 1.0 STEP -1.0:FOR C=Z TO N STEP -1.0
810 IF (R-Z)*(C-Z)=0.0 THEN 830
820 P(R,C)=P(R,C+1.0)+P(R+1.0,C):GOTO 840
830 P(R,C)=1.0
840 NEXT C:N=N+1.0:CURSOR 40,0:R%=INT(R)
850 PRINT R%;" ";:NEXT R:GOTO 590
860 REM *** BOVEN LINKER HELFT OPTIE 3
870 CURSOR 11,1:PRINT "15          15";TAB(41);"16          16"
880 M=Q:Y=T:Z=INT(Y/2.0):B5=Z*2.0:Z1=Z
890 Z2=Z1:Z3=Z2:X4=Z3:X5=X4
900 FOR I=1.0 TO Z1:FOR J=1.0 TO Z
910 IF (J-1.0)*(I-1.0)=0.0 THEN 930
920 P(I,J)=P(I,J-1.0)+P(I-1.0,J):GOTO 940
930 P(I,J)=1.0
940 NEXT J:Z=Z-1.0:CURSOR 10,0:I%=INT(I)
950 PRINT I%;:NEXT I:N=Z1
960 REM *** BOVEN RECHTER HELFT OPTIE 3
970 FOR I=1.0 TO Z1:FOR J=Y TO X5+1.0 STEP -1.0
980 IF I=1.0 OR J=Y THEN 1000
990 P(I,J)=P(I,J+1.0)+P(I-1.0,J):GOTO 1010
1000 P(I,J)=1.0
1010 NEXT J:X5=X5+1.0:CURSOR 20,0:I%=INT(I)
1020 PRINT I%;:NEXT I:N=Z2
1030 REM *** ONDER LINKER HELFT OPTIE 3
1040 FOR I=Y TO X4+1.0 STEP -1.0:FOR J=1.0 TO Z2
1050 IF J=1.0 OR I=Y THEN 1070
1060 P(I,J)=P(I,J-1.0)+P(I+1.0,J):GOTO 1080
1070 P(I,J)=1.0
1080 NEXT J:Z2=Z2-1.0:CURSOR 40,0:I%=INT(I)
1090 PRINT I%;:NEXT I:N=Z3
1100 REM *** ONDER RECHTER HELFT OPTIE 3
1110 FOR I=Y TO N+1.0 STEP -1.0:FOR J=Y TO Z3+1.0 STEP -1.0
1120 IF J=Y OR I=J THEN 1140
1130 P(I,J)=P(I+1.0,J)+P(I,J+1.0):GOTO 1150
1140 P(I,J)=1.0
1150 NEXT J:Z3=Z3+1.0:CURSOR 50,0:I%=INT(I)
1160 PRINT I%;:NEXT I:GOTO 590
```

- 8080 2707 EPROM programming software assembled listing December 1978 p 104
- 8080 A tabcounter for your edit buffer assembled listing August 1978 p 151
- 8080 An 8080 binary tape monitor assembled listing February 1978 p 153
- 8080 An Intel hex-format paper tape monitor assembled listing March 1978 p 162
- 8080 Blockade—VDM-1 video game assembled listing with object code November 1977 p 167
- 8080 Cassette Operating System assembled subroutine listings April 1977 p 129
- 8080 Convert Motorola 6800 hex format to Intel format assembled listing with object code May 1977 p 109
- 8080 Conway's Game of Life assembled listing for Processor Technology's video display board May 1977 p 138
- 8080 Copy/Sort/Search data manipulation assembled listing September 1978 p 136
- 8080 Cromemco Dazzler graphics interface driver assembled listing January 1978 p 153
- 8080 Diablo printer drive routine assembled listing July 1977 p 25
- 8080 Dr. Wang's tiny BASIC assembled listing December 1976 p 89
- 8080 Generalized string sorting routine BASIC and 6 assembled listings November 1978 p 131
- 8080 I/O driver for PERSCI 1070 intelligent disk controller assembled listing September 1977 p 153
- 8080 IAPSTM International ASCII Publication Standard conversion assembled listing May 1978 p 79
- 8080 IAPSTM International ASCII Publication Standard conversion assembled load-create-verify-dump listings June 1978 p 148
- 8080 Intel hex format paper-tape punch program assembled listing with object code July 1977 p 155
- 8080 Intelligent terminal program assembled listing with object code September 1977 p 75
- 8080 Interval timer design 5 assembled routines January 1978 p 127
- 8080 Intruded driven floppy disk controller for the S-100 bus assembled listing May 1978 p 152
- 8080 Livermore BASIC Interpreter Part I BASIC and assembled plot functions December 1976 p 115
- 8080 Livermore BASIC Interpreter Part II assembled listing January 1977 p 97
- 8080 Livermore BASIC Interpreter Part III assembled floating point math package listing February 1977 p 104
- 8080 Livermore BASIC Interpreter Part IV assembled octal debugging listing March 1977 p 121
- 8080 Look—byte look-up program assembled listing May 1978 p 167
- 8080 Memory catalog program assembled listing May 1978 p 170
- 8080 Memory object code search routine assembled routine February 1977 p 121
- 8080 Monitor initialization and printer control 2 assembled Daisywheel listings October 1978 p 111
- 8080 Morse Code generator assembled listing October 1978 p 89
- 8080 Octal Monitor Program assembled listing February 1977 p 13
- 8080 Piranha—game assembled listing with object code December 1977 p 166
- 8080 Polymorphic Ideaboard software 2 assembled routines May 1977 p 98
- 8080 Punch and read Intel formatted tape assembled listing December 1977 p 152
- 8080 Random Number Generator Program assembled listing with a BASIC Chi-square subroutine February 1977 p 100
- 8080 Robot's random programming approach assembled listing April 1978 p 158
- 8080 Text editor for XEK and PTC Co Assemblers assembled listing October 1978 p 140
- 8080 Video Chase—a game assembled listing with object code October 1977 p 167
- 8080A Resident Operating System 1K SOL assembled listing January 1977 p 90
- 8080/8085/8086 boeken
 D2 8080A/8085 ASSEMBLY LANGUAGE PROGRAMMING (Leventhal/Osborne)
 D3 8080 PROGRAMMING FOR A LOGIC DESIGN (Osborne/Osborne)
 D4 8080 SOFTWARE GOURMET GUIDE & COOKBOOK (Findley/Scelbi)
 D5 EDITOR/ASSEMBLER SYSTEM FOR 8080/8085 BASED COMPUTERS (Weller)
 D6 DEBUG: AN 8080 INTERPRETIVE DEBUGGER (Titus/Sams)
 D7 KF2DOS: A FLOPPY DISK OPERATING SYSTEM FOR THE 8080 (Wellus)
 D8 8080 GALAXY GAME (Edwards/Scelbi)
 D9 8080 STANDARD EDITOR (Edwards/Scelbi)
 D10 8080 STANDARD ASSEMBLER (Edwards/Scelbi)
 D11 8080 STANDARD MONITOR (Edwards/Scelbi)
 D12 THE 8086 PRIMER (Morse/Hayden)
 D13 TEA: AN 8080/8085 CO-RESIDENT EDITOR/ASSEMBLER (Titus/Sams)
 D16 THE 8086 BOOK (Rector & Alexy/Osborne)

```

2 REM TITEL MENU MET LICHTPEN V0.0
4 REM DATUM 26-12-81
6 REM BRON (AUTEUR) M.J.BERKX
8 REM OPSLAG BAND NR.16; CODE LP1;NR A1
10 MODE 0:PRINT CHR$(12):COLORT 0 8 15 0
15 REM *****OPBOUW VAN MENU*****
20 CURSOR 6,20:PRINT "Eerste keuze"
25 POKE #BE4D+1,#B:POKE #BE4D-3,#B
30 CURSOR 6,15:PRINT "Tweede keuze"
35 POKE #BBAF+1,#B:POKE #BBAF-3,#B
40 CURSOR 6,10:PRINT "Derde keuze"
45 POKE #B911+1,#B:POKE #B911-3,#B
60 K1=0.0:K2=0.0:K3=0.0
65 CURSOR 3,3:PRINT "Plaats de lichtpen TUSSEN DE STREEPJES voor Uw keuze"
70 WAIT TIME 200
80 REM ***** DOORLOPEN VAN MENU MET WIT BLOKJE*****
100 FOR N=1.0 TO 10.0
110 POKE #BE4D-3,#FF:POKE #BE4D-1,#FF:FOR T=0.0 TO 100.0:NEXT
112 IF PEEK(#FD00) IAND #20=#20 THEN GOSUB 500
115 POKE #BE4D-1,0:POKE #BE4D-3,0
120 POKE #BBAF-3,#FF:POKE #BBAF-1,#FF:FOR T=0.0 TO 100.0:NEXT
122 IF PEEK(#FD00) IAND #20=#20 THEN GOSUB 600
125 POKE #BBAF-1,0:POKE #BBAF-3,0
130 POKE #B911-3,#FF:POKE #B911-1,#FF:FOR T=0.0 TO 100.0:NEXT
132 IF PEEK(#FD00) IAND #20=#20 THEN GOSUB 700
135 POKE #B911-1,0:POKE #B911-3,0
160 NEXT
165 REM *****RESULTAAT VAN KEUZE OF GEEN KEUZE*****
170 PRINT CHR$(12):CURSOR 10,12:PRINT "GEEN KEUZE TE MAKEN":END
500 PRINT CHR$(12):CURSOR 10,12:PRINT "U MAAKTE KEUZE EEN":END
510 RETURN
600 PRINT CHR$(12.0):CURSOR 10,12:PRINT "U MAAKTE KEUZE TWEE":END
610 RETURN
700 PRINT CHR$(12):CURSOR 10,12:PRINT "U MAAKTE KEUZE DRIE":END
710 RETURN
1000 POKE #BE4D+1,#A:POKE #BE4D-3,#A

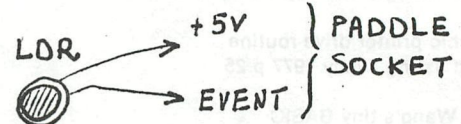
```

EEN VAN DE
LICHTPEN

```

*
*
*
*
*LOAD:LIST
10 MODE 0
20 PRINT CHR$(12):CURSOR 10,12
30 PRINT "PAK ME DAN, ALS JE KAN!"
40 WAIT TIME 200
50 MODE 4
60 COLORG 0 5 10 15
90 FOR N=1.0 TO 5.0
100 X1=10.0+RND(XMAX-20.0):Y1=10.0+RND(YMAX-20.0)
110 FILL X1,Y1 X1+10,Y1+10 15
120 WAIT MEM #FD00,#10
130 X2=10.0+RND(XMAX-20.0):Y2=10.0+RND(YMAX-20.0)
140 FILL X2,Y2 X2+10,Y2+10 15
150 FILL X1,Y1 X1+10,Y1+10 0
160 WAIT MEM #FD00,#10
170 FILL X2,Y2 X2+10,Y2+10 0
180 NEXT
190 K=K+1.0
195 ON K GOTO 200,210,220,230,240
200 PRINT "JE KRIJGT ME TOCH NIET TE PAKKEN!":WAIT TIME 200:GOTO 50
210 PRINT "IK BEN LEKKER SNELLER!":WAIT TIME 200:GOTO 50
220 PRINT "VOLHOUDEN! JE HAD ME BIJNA!":WAIT TIME 200:GOTO 50
230 PRINT "EEN BEETJE VLUgger EN JE HEBT ME!":WAIT TIME 200:GOTO 50
240 PRINT "HOU MAAR OP IK BEN STEEDS SNELLER DAN JIJ!":END

```



```

*
*
*
*LOAD:LIST
1 REM TITEL "LUXAFLEX"
2 REM BRON/AURTEUR: M.J. BERKX (27-12-81)
3 REM OPSLAG: BAND NR.14; CODE P01
5 COLORT 8 0 0 0
6 READ C
7 IF C<0.0 THEN END
10 FOR A=1.0 TO 23.0:FOR B=1.0 TO 60.0:PRINT CHR$(C);:NEXT B:PRINT :NEXT A
15 FOR P=1.0 TO 3.0
20 K=#F
25 FOR N=0.0 TO 23.0 100 POKE #BFEF-N*#86,#70+K
30 POKE #BFEF-N*#86,#70+K 110 WAIT TIME 2
40 WAIT TIME 2 120 NEXT N
50 NEXT N 125 WAIT TIME 20
60 K=#17-K 130 NEXT P
80 WAIT TIME 20 140 GOTO 6
90 FOR N=23.0 TO 0.0 STEP -1.0 150 DATA 1,2,6,15,29,22,-1

```

LUXAFLEX

VOLUME CONTROL

MIXING AND IMPEDANCE TRANSFORMATION

from programmable interval timer 8253-5
IC36 out 0 pin 10

adr. #FDx4 lower nibble

out 1 pin 13

adr. #FDx4 higher nibble

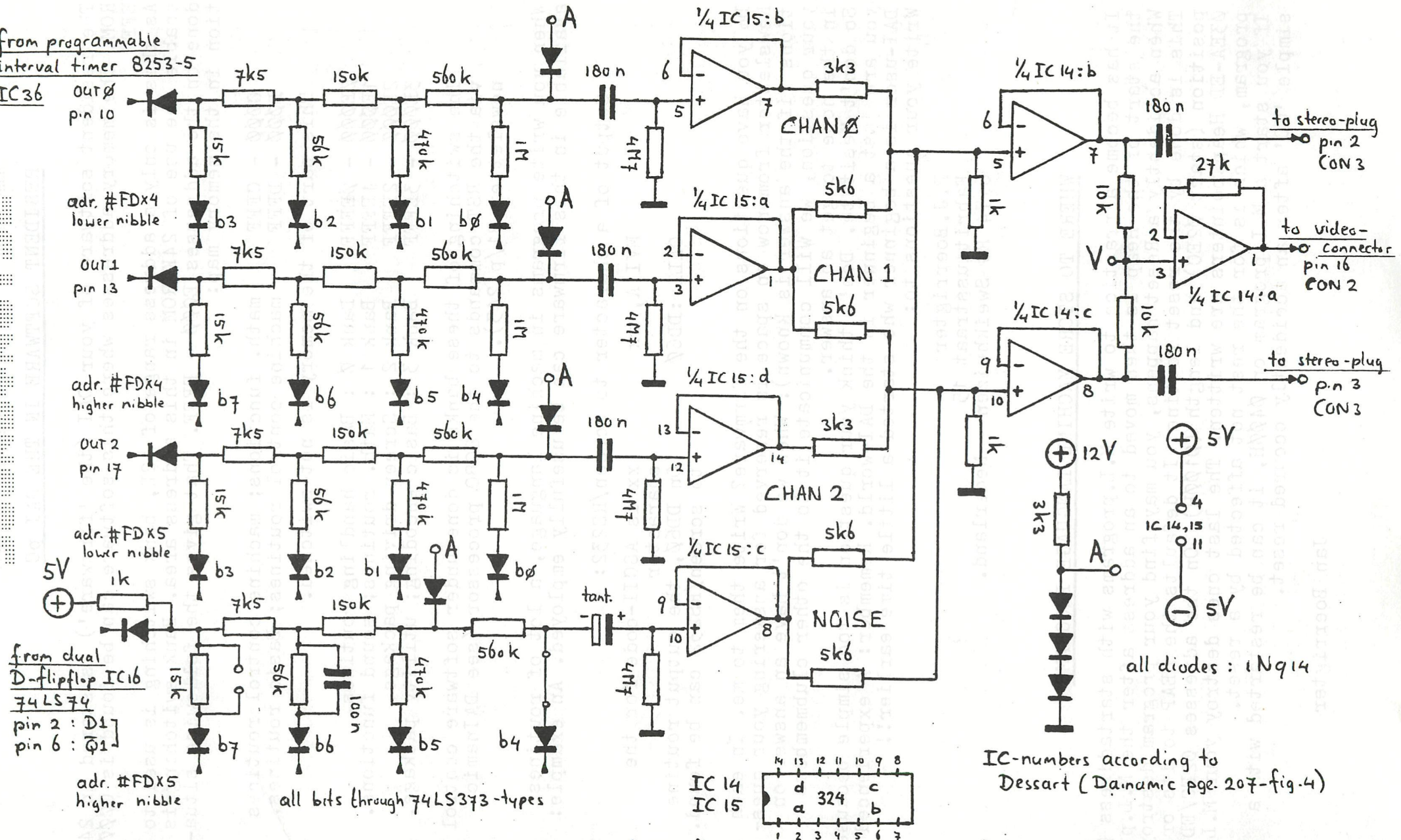
out 2 pin 17

adr. #FDx5 lower nibble

from dual D-flipflop IC16
74LS74
pin 2 : D1
pin 6 : Q1

adr. #FDx5 higher nibble

all bits through 74LS373-types



IC-numbers according to Dessart (Dainamic pge. 207-fig.4)

fig.5 : SOUND + NOISE INTERFACE DAI P.C. (REV.4)

A.F.J. de Jong 01-10-81

RESIDENT SOFTWARE IN THE DAI pC

RESIDENT SOFTWARE IN THE DAI pC

The resident software of your DAI (the 'firmware') is stored in 24K ROM. The memory addresses where this software can be found is C000 - EFFF.

As this is only a address range of 12K, bank switching is used to enable the use of 24K ROM in this address area. Bank switching is done on the addresses E000 - EFFF. That gives the following situation in the memory map:

C000 - CFFF math. functions; machine control routines
D000 - DFFF machine control routines; Basic routines.

This part of the memory is not-switched.

0E000 - 0EFFF Bank 0 : Basic handling routines.
1E000 - 1EFFF Bank 1 : Math. routines; sound functions.
2E000 - 2EFFF Bank 2 : Screen driving package.
3E000 - 3EFFF Bank 3 : Basic encoding; utility package.

The switching of these banks is done under software control via the RST commands to the 8080 processor (see DAInamic newsletter 81/p.243).

When you write programs in machine language, a lot of routines available in this firmware can be usefully employed. An example:

Output of a character to screen/RS232:

```
MVI A, xx                    xx is ASCII-code for the
                           character
CALL :DD60                   On DD60, the output routine
                           to screen/RS232 can be found.
```

Do you have questions on the firmware? Write them to me. In each Newsletter from now on space is reserved for answering your questions (if the answer is known). When we don't have an answer on your question, we will communicate it to the other clubmembers in the hope to get an answer.

So don't hesitate. Don't think your question is too simple because you are just a beginner in the DAI-world. Remember: A experienced DAI-user is a beginner who started a little time earlier!!

Write your questions to:

B.J.Boerrigter
Fabritiusstraat 15
6174 RG Sweikhuizen - Nederland.

WHERE TO STORE MACHINE LANGUAGE PROGRAMS

It has become a practice to write M.L.programs with startaddress 0300, The start of the Heap is then moved to an address after the M.L.program. When accidentally a Reset happens, you may find your program destroyed. This is done by the reset routine: It defaults the HEAP to its original position (start 02EC) and length (0100H). On the addresses 02EC/ED and 03EA-ED Heap pointers are written. The last ones destroy your M.L. program, which is for the rest not affected by a reset.

If you start a M.L.program on 0400H, it can be restarted with a simple 'GO' after an accidentally occurred reset.

Jan Boerrigter

SAVING and LOADING of STRING ARRAYS.

Especially in word processing programs, string arrays (e.g. A\$(N)) are used to store the data. These arrays can be saved on tape via the Basic instruction SAVEA.

During saving (and/or loading) of these arrays, several problems may occur, as for instance a 'OUT OF MEMORY' error report.

This article will describe what happens during saving and loading of string arrays. Read before what is written in Newsletter 1981, page 188,189 about string arrays and the Heap.

In the Heap, string arrays are stored in two parts: 1. the address where the strings can be found, and 2. the string itself. Strings are stored in the sequence they are assigned, so they do not have a fixed location inside the Heap. Before saving a string array, it is necessary to get all the data of the particular array together. Via the next program, the whole sequence is demonstrated.

```

10 CLEAR 2500
20 DIM A$(5.0),B$(5.0),N$(5.0),M$(5.0)
30 FOR N=0.0 TO 5.0:READ A$(N):READ B$(N):NEXT
40 FOR N=0.0 TO 5.0
50 FOR X=0.0 TO 9.0
60 N$(N)=N$(N)+A$(N):M$(N)=M$(N)+B$(N)
70 NEXT X:NEXT N
80 STOP
90 SAVEA N$ "TEST 1"
100 STOP
110 SAVEA M$ "TEST 2"
120 END
130 DATA A,G,B,H,C,I,D,J,E,K,F,L

```

```

200 CLEAR 2500
210 DIM N$(5.0),M$(5.0)
220 PRINT "START TAPE"
230 LOADA N$ "TEST 1"
240 PRINT "START TAPE"
250 LOADA M$ "TEST 2"
260 END

```

The two string arrays N\$(5) and M\$(5) are filled with the contents of 10 times a letter. N\$(0)="AAAAAAAAAA"; M\$(5)="LLLLLLLLLL".

When the program stops in line 80, the contents of the Heap is as follows:

02EC	00 0E 01 05	
02F0	2D 03 33 03 39 03 3F 03 45 03 4B 03 00 0E 01 05	string addresses
0300	30 03 36 03 3C 03 42 03 48 03 4E 03 00 0E 01 05	
0310	5C 03 7F 03 97 03 AF 03 C7 03 DF 03 00 0E 01 05	
0320	68 03 8B 03 A3 03 BB 03 D3 03 EB 03 00 01 41 00	
0330	01 47 00 01 42 00 01 48 00 01 43 00 01 49 00 01	A\$(n), B\$(n)
0340	44 00 01 4A 00 01 45 00 01 4B 00 01 46 00 01 4C	
0350	89 89 46 46 46 46 46 46 46 46 46 00 0A 41 41 41	
0360	41 41 41 41 41 41 41 00 0A 47 47 47 47 47 47 47	
0370	47 47 47 89 89 46 46 46 46 46 46 46 46 00 0A	
0380	42 42 42 42 42 42 42 42 42 42 00 0A 48 48 48 48	
0390	48 48 48 48 48 48 00 0A 43 43 43 43 43 43 43 43	N\$(n), M\$(n)
03A0	43 43 00 0A 49 49 49 49 49 49 49 49 49 49 00 0A	
03B0	44 44 44 44 44 44 44 44 44 44 00 0A 4A 4A 4A 4A	
03C0	4A 4A 4A 4A 4A 4A 00 0A 45 45 45 45 45 45 45 45	
03D0	45 45 00 0A 4B 4B 4B 4B 4B 4B 4B 4B 4B 4B 00 0A	
03E0	46 46 46 46 46 46 46 46 46 46 00 0A 4C 4C 4C 4C	
03F0	4C 4C 4C 4C 4C 4C 88 B6	

In the Heap, the addresspointers to the strings can be found, and the particular strings themselves. All strings are found in the Heap in the sequence they are loaded. For instance parts of strings can be found from line 60 in the program, where the N\$ and the M\$ increase every time.

When now line 90 is runned, the following happens. The ROM-software dictates the DAI to order all the data belonging to the string N\$. This is done in the free RAM space! If line 90 has been executed, you will find directly after the symboltable (ending on 0E27) all the string data in a neat order.

```

OE28 | 00 0C 0A 41 41 41 41 41
OE30 | 41 41 41 41 41 0A 42 42 42 42 42 42 42 42 42
OE40 | 0A 43 43 43 43 43 43 43 43 43 43 0A 44 44 44 44
OE50 | 44 44 44 44 44 44 0A 45 45 45 45 45 45 45 45
OE60 | 45 0A 46 46 46 46 46 46 46 46 46 46 46

```

The same happens after executing line 110. But no additional RAM space is used; the data of N\$ is overwritten by the data of M\$:

```

OE28 | 00 0C 0A 47 47 47 47 47
OE30 | 47 47 47 47 47 0A 48 48 48 48 48 48 48 48 48
OE40 | 0A 49 49 49 49 49 49 49 49 49 49 0A 4A 4A 4A 4A
OE50 | 4A 4A 4A 4A 4A 4A 0A 4B 4B 4B 4B 4B 4B 4B 4B
OE60 | 4B 0A 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C

```

Maybe now is clear why a 'OUT OF MEMORY' error report occur when a string array is to be saved on tape. When a large Heap is cleared, the program itself uses a lot of memory space, and the stringarray contains a lot of data, the free RAM space may be too small to contain all the string data to be saved.

When LOADING string arrays from tape, the same sequence is used. The data, read from tape, is stored in the free RAM space above textbuffer and symbol table.

When all reading has been done, the ROM-firmware initiates the re-organisation of the Heap: the string data is moved from the free RAM space into the Heap and the address pointers are updated.

This is the reason the DAI uses some time after reading string arrays from tape. You remember: after reading a string array (cursor do not flash), the cursor starts flashing, but the DAI is still busy for some time.

It is a failure in the software design that the cassette motor is not switched off after the end of the reading, but after the organisation of the Heap.

Final conclusion: When you want to use string arrays for saving data on tape, be careful in the memory space you use. For both saving and reading, you need almost the same memory space in the free RAM space as you need in the Heap for the particular string array. Every obstruction against this rule is fined with a 'OUT OF MEMORY' error report.

So take care!

Jan Boerrigter/jan.'82

MINI C BOOTSTRAP LOADER

J#L.
PAGE 01 DBL 1.1 30DEC81



002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055

```
*****  
*  
*      DAINAMIC  BOOTSTRAP  LOADER   ( DBL )  
*      -----  
*  
*      DBL IS USED FOR EASY LOADING OF PROGRAMS WHICH  
*      HAVE A MACHINE LANGUAGE AND A BASIC PART.  
*  
*      VERSIONS :  
*      DBL 1.* : DELIVERED AS A BASIC PROGRAM AND  
*                IS USABLE WITH CASSETTE, DCR AND  
*                DISK BY TYPING *LOAD:RUN .  
*      DBL 2.* : DELIVERED AS A UTILITY FILE AND  
*                ONLY USABLE WITH DCR AS A 'USER'  
*                FILE AND EXECUTED AFTER DAI RESET.  
*  
*****  
*  
*      D B L   V 1 . *  
*      -----  
*  
*      THIS MACHINE LANGUAGE PROGRAM IS LOADED IN THE  
*      STACKAREA OF THE RAM BY A BASIC PROGRAM.  
*      CONTROL IS TRANSFERED TO IT WITH A CALLM #FOOO.  
*  
*      FUNCTIONS :  
*  
*      1) INIT STACKPOINTER.  
*  
*      2) LOAD THE FIRST UTILITY FILE ON TAPE.  
*  
*      3) SET BEGIN OF HEAP AFTER CODE OF THE  
*          UTILITY FILE AND PERFORM A *NEW TO ADJUST  
*          OTHER BASIC POINTERS.  
*  
*      4) PRINT MESSAGE [LOAD BASIC] AND LOAD THE  
*          FIRST BASIC PROGRAM ON TAPE.  
*  
*      5) FORCE A RUN OF THE LOADED BASIC PROGRAM.  
*  
*      NOTES :  
*  
*      1) TO START WITH A CLEAN SYSTEM, DO A RESET  
*          BEFORE STARTING THE BOOTSTRAP LOADING  
*  
*      2) IF THE BREAKKEY IS PRESSED :  
*          - BEFORE CONTROL IS TRANSFERED TO THIS MLP  
*            ( = BEFORE OR DURING POKING OF THE CODE )  
*            THEN TYPE *RUN TO RESTART  
*          - AFTER LOADING OF THE UTILITY FILE  
*            ( = AFTER FUNCTION 1,2,3 )  
*            THEN TYPE *LOAD:RUN TO CONTINUE  
*  
*****
```

```

056 * - AFTER LOADING THE BASIC *
057 * THEN TYPE *RUN TO START THE PROGRAM *
058 * *
059 * 3) IF LOADING ERRORS OCCUR DURING : *
060 * - LOAD OF BASIC PROGRAM DBL OR *
061 * LOAD OF UTILITY FILE *
062 * THEN RESTART THE WHOLE PROCESS *
063 * - LOAD OF THE BASIC PROGRAM THEN RESTART *
064 * ONLY LOAD OF BASIC : SET TAPE BEFORE *
065 * BASIC AND TYPE *LOAD:RUN *
066 * LOADING ERRORS ARE REPORTED IN THE NORMAL *
067 * DAI ERROR FORMAT. *
068 * *
069 * 4) AS DBL READS THE TAPE FILES WITHOUT *
070 * FILENAMES, PUT THE FILES IN CORRECT *
071 * SEQUENCE ON THE TAPE : *
072 * 1- ODBL 1.* : BASIC TO POKE DBL IN RAM *
073 * 2- 1UT-FILE : YOUR MLP CODE *
074 * 3- OBASIC : YOUR BASIC PROGRAM *
075 * *
076 * ***** *
077 * *
078 * DESCRIBES OF DAI REFERENCES *
079 * ----- *
080 CURRNT EQU :100 START OF CURRENT LINE
081 HEAP EQU :29B BEGIN HEAP
082 SRBOT EQU :F800 BOTTOM OF STACK RAM
083 SSTOP EQU :F900 TOP OF STACK RAM
084 ROPEN EQU :2CE SEARCH TAPE FILE, SYNC, CHECK
085 FILE TYPE, DISPL/CHECK NAME
086 RBLK EQU :2D1 TRANSFER BLOCK OF BYTES FROM
087 TAPE TO RAM, EXIT :
088 IF NO LOADING ERRORS
089 THEN CY=1
090 ELSE CY=0 AND A=ERRORCODE
091 RCLO EQU :2D4 STOP CASSETTE
092 FORCEB EQU :CB92 FORCE BASIC RUN IF NO BREAK
093 ERRLD EQU :D2A8 ENTRY FOR TAPE ERROR MSG,
094 EXITS TO BASIC COMMAND MODE
095 CRLF EQU :DD5E PRINT A CAR RETURN
096 RNEW EQU :DEB8 DELETE BASIC PROGRAM AND SET
097 UP BASIC POINTERS
098 *
099 * ORG SRBOT
100 *
101 F800 3100F9 ENTRY LXI SP, SSTOP INIT STACK POINTER
102 F803 210000 LXI H, :0 READ WITHOUT NAME
103 F806 220001 SHLD CURRNT SET NO PROGRAM RUNNING
104 F809 01FF31 LXI B, :31FF FILETYPE 1 + PRINT NAMES
105 F80C CDCE02 CALL ROPEN
106 F80F 2154F8 LXI H, TEMP TEMPORY SAVE AREA
107 F812 1156F8 LXI D, TEMP+2
108 F815 CDD102 CALL RBLK READ BEGINADDR UT CODE
109 F818 2100B0 LXI H, :B000 MAX ADDR UT FILE
110 F81B EB XCHG
111 F81C 2A54F8 LHLD TEMP
112 F81F DCD102 CC RBLK IF NO ERRORS READ BLOCK
113 WITH CONTENTS UT FILE

```

```

114 F822 CDD402      CALL  RCLO      ALWAYS STOP CASSETTE
115 F825 D2A8D2      JNC   ERRLD     ABORT IF ERRORS
116 F828 229B02      SHLD  HEAP      ADJUST HEAPBEGIN
117 F82B CDB8DE      CALL  RNEW      ADJUST OTHER POINTERS
118 F82E CD5EDD      CALL  CRLF
119 F831 0137F8      LXI  B,BASLIN  POINTER BASIC CMD LINE
120 F834 C392C8      JMP   :C892     START BASIC RUN
121                  *
122                  * FOLLOWING DATA IS EXECUTED AS A BASIC COMMAND LINE
123                  TRUN  EQU   :87      RUN (DAI INTERNAL CODE)
124                  TLOAD EQU   :8B      LOAD
125                  TPRINT EQU  :AD      PRINT
126                  *
127 F837 AD          BASLIN DATA TPRINT PRINT MESSAGE
128 F838 01          DATA :01      1 EXPRESSION
129 F839 20          DATA :20      SEPARATOR
130 F83A 18          DATA :18      STRINGCONSTANT
131 F83B 11          DATA :11      STRINGLENGTH
132 F83C 5B204C     ASC   '[ LOADING BASIC ]'
133 F84D FF          DATA :FF      END PRINT WITH CR
134 F84E 8B          DATA TLOAD     LOAD BASIC PROGRAM
135 F84F 1900        DATA :19,:00  DUMMY STRINGCONSTANT
136 F851 AD00        DATA TPRINT,:00 PRINT ONLY CR
137 F853 87          DATA TRUN      START BASIC PROGRAM
138                  *
139 F854 0000        TEMP  DBL   0
140                  *
141 F856              END

```

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

```

BASLIN F837      CRLF DD5E      CURRNT 0100      ENTRY  F800
ERRLD  D2A8      FORCEB C892     HEAP   029B     RBLK   02D1
RCLO   02D4      RNEW  DEB8     ROPEN  02CE     SRBOT  F800
SSTOP  F900      TEMP  F854     TLOAD  008B     TPRINT 00AD
TRUN   0087

```

```

100 REM <<< DAINAMIC BOOTSTRAP LOADER V1.1 30DEC81 >>>
110 REM /* THIS PROGRAM LOADS A BOOTSTRAP MLP IN RAM
120 REM /* THE MLP LOADS A UTILITY FILE AND A BASIC PROGRAM
130 REM /* BEGIN OF HEAP IS ADJUSTED AFTER THE MLP
140 REM /* THE BASIC PROGRAM IS ENTERED WITH A RUN
150 REM /* CONTENTS TAPE : DBL, UT-FILE, BASIC PROGRAM
160 REM /* SET TAPE BEFORE DBL, TYPE *LOAD:RUN, START CASSETTE
200 MODE 0:PRINT CHR$(12):COLORT 8 0 0 8
210 POKE #75,#5F: CLEAR #100
220 PRINT "*** DAINAMIC BOOTSTRAP LOADER V1.1 ***":PRINT
230 READ BGNADDR,NMBR
240 FOR ADDR=BGNADDR TO BGNADDR+NMBR-1
250 READ BYTE:POKE ADDR,BYTE
260 NEXT:READ C:IF C<>#FFFF THEN PRINT "DATA READ ERROR":PRINT :END
270 PRINT "[ LOADING UTILITY FILE ]"
280 CALLM BGNADDR
300 REM /* MLP BYTES
310 DATA #F800,86
320 DATA #31,#00,#F9,#21, #00,#00,#22,#00, #01,#01,#FF,#31, #CD,#CE,#02,#21
330 DATA #54,#F8,#11,#56, #F8,#CD,#D1,#02, #21,#00,#B0,#EB, #2A,#54,#F8,#DC
340 DATA #D1,#02,#CD,#D4, #02,#D2,#A8,#D2, #22,#9B,#02,#CD, #B8,#DE,#CD,#5E
350 DATA #DD,#01,#37,#F8, #C3,#92,#C8,#AD, #01,#20,#18,#11, #5B,#20,#4C,#4F
360 DATA #41,#44,#49,#4E, #47,#20,#42,#41, #53,#49,#43,#20, #5D,#FF,#8B,#19
370 DATA #00,#AD,#00,#87, #00,#00,#FFFF

```

BASIC HANDBOEKEN-MANUALS

Voorbeelden en informatie vind je ondermeer in volgende boeken :

- I DAI personal computer manual - DAI microcomputer engineering company.
- II Basic voor je personal computer - Stichting BASICned Hollandsche Rading.
- III Cursusboek bij Teleaccursus MICROPROCESSORS 2 - Stichting Teleac Utrecht.
- IV User's manual for level 1 Basic (TRS-80)
- V Handleiding voor Basic level 2 "

		I	II	III	IV	V		I	II	III	IV	V
01	CHECK	74		75			47	WAIT	68		271	
02	CLEAR	78		191		4/2	48	UT	71		191	
03	COLORG	82		280			49	ABS	96	239	113	93 7/1
04	COLORT	82		279			50	ACOS	96		123	
05	CONT	77			53	2/3	51	ALOG	96			
06	CURSOR	87		283			52	ASC	96	215	157	5/3
07	DATA	71	110	92	85	3/8	53	ASIN	96		122	
08	DIM	78	158	100		4/3	54	ATN	96		123	7/1
09	DOT	83	249	281			55	CHR \$	97	209	159	5/4
10	DRAW	83	250	281			56	COS	97		120	7/2
11	EDIT	61		43		9/1	57	CURX	88		283	
12	END	64	53	75	13	4/4	58	CURY	88		283	
13	ENVELOPE	92		284			59	EXP	97		124	7/2
14	FILL	83	250	282			60	FRAC	97		115	
15	FOR NEXT	64	134	67	45	4/8	61	FRE	79			5/5
16	GPSUB	65	227	81	81	4/6	62	FREQ	93		285	
17	GOTO	66	60	40	29	4/5	63	GETC	72		301	
18	IF GOTO	66		59	29		64	HEX \$	97		191	
19	IF THEN	66	99	52	29	4/12	65	INP	69		261	8/4
20	IMP	48					66	INT	98	90	113	69 7/3
21	INPUT	72	64	34	33	3/7	67	LEFT \$	98	221	155	5/6
22	LET	79	47	31	18	4/4	68	LEN	98	221	154	5/6
23	LIST	62	63	44	13	2/4	69	LOG	98		124	7/3
24	LOAD	74		74			70	LOGT	98		124	
25	LOADA	75		107			71	MID \$	99	219	156	5/6
26	MODE	80	249	280			72	PDL	70			
27	NEW	63	51	42	7		73	PEEK	70		266	8/5
28	NEXT	65	145	71	45	4/8	74	PI	99		121	
29	NOISE	93		285			75	RIGHT \$	99	221	156	5/7
30	ON GOSUB	66		84	78	4/7	76	RND	99	87	116	99 7/3
31	ON GOTO	67	97	77	78	4/6	77	SCRN	87		283	
32	OUT	69		261		8/4	78	SGN	100		115	7/4
33	POKE	70		267		8/5	79	SIN	100		120	7/4
34	PRINT	73	62	32	9	3/1	80	SPC	100	249	160	
35	READ	73	118	94	85	3/9	81	SQR	100		116	7/7
36	REM	77	98	45		4/12	82	STR \$	100	213	158	5/7
37	RESTORE	73	217	98	87		83	TAB	101	248	160	61
38	RETURN	67	232	81	82	4/6	84	TAN	101		121	7/4
39	RUN	63	66	45	9		85	VAL	101	212	159	5/8
40	SAVE	75		74			86	VARPTR	79		270	8/8
41	SAVEA	76		107			87	XMAX	87		282	
42	SOUND	92		284			88	YMAX	87		282	
43	STOP	68	104	81	56	4/5	89	MOD	47			
44	TALK	94					90/91	IOR/IAND	104		251	
45	TROFF	78				2/5	92/93	IXOR/INOT	104		251	
46	TRON	78				2/5	94/95	SHL/SHR	102		263	

CARPENTERS SOLUTION



```
T
2010 COMMAND$="515005152271140413603625050336263141143422737227372050
053362631141436050271404114041227270"
2020 COMMAND$=COMMAND$+"051503633605227124336263362605227371406111404
1270051500515"
2030 COMMAND$=COMMAND$+"360361211427050503631411427372273705250053362
6311431406316005150227"
```

```
2010 COMMAND$="51500515227114041360362505/033626314114342273722737205005"
2020 COMMAND$=COMMAND$+"33626311434142700515036/1404114041270051503624/22737205336263362614227372053636163114041272270051500515"
2030 COMMAND$=COMMAND$+"271140436250535036263141142737227370525005336263114143636005150227"
```

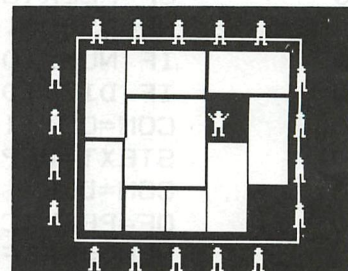
```
2010 COMMAND$="5051150522711404136360502503362631411434227273372720500533626311
4143605027141400414/"
2020 COMMAND$=COMMAND$+"1227270051503633605227214336263362605227371406111404127
00505115053611404/"
2030 COMMAND$=COMMAND$+"2735052503362631411427372273705250053362631141436360051
50227"
```

*

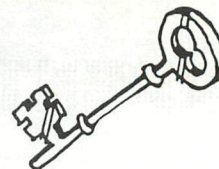
```
2010 COMMAND$="5150051522711404136036250503362631431142427372273720500533626311
4143605027140411404136005150027352227372143362"
2020 COMMAND$=COMMAND$+"6336260522714363631140412722700515005152711436050530626
3141142737227370525005336263114143636005150227"
```

```
51500515227114041360362505033626314311424273722737205005336263114143605027140411
40413600515002735222737214336263362605227143636311404127227005150051527114360508
306263141142737227370525005336263114143636005150227
211
```

```
3299 REM CORRECTIONS AS SUGGESTED BY K..VERBEKE
3300 FOR Y=0 TO 19:GOSUB 4020
3320 DRAW A*20,B*20+Y0+Y+1 A*20+X0,B*20+Y0+Y+1 CO
3450 FOR Y=0 TO 19:GOSUB 4020
3470 DRAW A*20,B*20-Y-1 A*20+X0,B*20-Y-1 CO
3620 FOR Y=1 TO 20:GOSUB 4020
3630 DRAW A*20+Y+1+X0,B*20 A*20-Y+X0+1,B*20+Y CO
3830 FOR Y=1 TO 20:GOSUB 4020
3840 DRAW A*20+Y-1,B*20 A*20+Y-1,B*20+Y0 CO
```



```
T
3299 REM CORRECTIONS AS SUGGESTED BY M.STREICHER
3310 DRAW A*20,B*20+Y-1 A*20+X0,B*20+Y-1 20
3460 DRAW A*20,B*20+Y0-Y+1 A*20+X0,B*20+Y0-Y+1 20
3630 DRAW A*20-Y+X0+1,B*20 A*20-Y+X0+1,B*20+Y0 20
3850 DRAW A*20+Y-1,B*20 A*20+Y-1,B*20+Y0 20
```



Un petit programme qui intéressera les possesseurs d'une imprimante.
Vous avez sûrement déjà remarqué qu'avec un "check" le titre des programmes n'apparaît pas sur l'imprimante.

Ce programme vous permettra de lister les titres des programmes contenus sur une cassette digitale (pour un DCR, tapez d'abord DCR(REW) ou CAS au choix).

Quand vous voulez arrêter l'impression, tapez espace.

PAGE 1 -- CASSETTE LIST -- FLST V2.2

```

5      REM CASSETTE LIST
10     MODE 0:PRINT CHR$(12);:POKE #131,1:AD$=""
30     FOR A=1 TO 21:READ B:AD$=AD$+CHR$(B):NEXT
40     AD=PEEK(VARPTR(AD$)) IOR (PEEK(VARPTR(AD$)+1) SHL 8)+1
100    INPUT "CASSET'S NAME";CN$:PRINT
120    POKE #131,0:PRINT CN$:FOR A=0 TO LEN(CN$)-1:PRINT "-";:NEXT:
C      PRINT :CPT=1
1000   PRINT CHR$(12);:CALLM AD:PRINT :PRINT CPT;" - ";
1030   FOR A=#BFE7 TO #BF85 STEP -2:PRINT CHR$(PEEK(A));:NEXT
1035   IF GETC<>0 THEN 2000
1040   CPT=CPT+1:GOTO 1000
2000   POKE #131,1:END
10000  DATA #F5,#C5,#D5,#E5,#01,#40,#00,#11,#B1,#80,#21,#9E,#E6,
C      #CD,#CE,#02,#E1,#D1,#C1,#F1,#C9

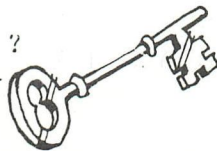
```

PAGE 1 -- WHAT TO DO IF A CRASH OCCURS.... -- FLST V2.2

```

1      REM WHAT TO DO IF A CRASH OCCURS....
2      INPUT "START OF HEAP (#2EC)";START:PRINT :DEP=START
10     COM=DEP:DIZ=0
20     OF=PEEK(COM):NUM=(PEEK(COM+1) SHL 8) IOR PEEK(COM+2):IF
C      NUM>2000.0 OR NUM<1.0 THEN DEP=DEP+1:IF DEP<#A000 THEN 10
40     IF NUM/10=INT(NUM/10.0) THEN DIZ=DIZ+1
50     IF DIZ>10 THEN 70
60     COM=COM+1+OF:GOTO 20
70     STEXT=DEP
80     COM=DEP
90     OF=PEEK(COM):NUM=(PEEK(COM+1) SHL 8) IOR PEEK(COM+2):IF
C      NUM>#FFFF OR NUM=0 OR OF=0 THEN 120
110    COM=COM+1+OF:GOTO 90
120    COM=COM+1:FINP=COM
130    A=PEEK(COM) IAND #F:IF A=0 THEN 200
140    COM=COM+A+1:B=PEEK(COM) IAND #F:COM=COM+B+1:GOTO 130
200    ESY=COM+1
210    PRINT "HEAP: ";HEX$(STEXT-START)
220    PRINT "!!! START OF TEXT: ";HEX$(STEXT);" FIRST LINE=
C      ";(PEEK(DEP+1) SHL 8) IOR PEEK(DEP+2)
230    PRINT "END OF TEXT: ";HEX$(FINP)
240    PRINT "END OF SYMBOL TABLE: ";HEX$(ESY)

```

Beaucoup d'entre vous ont sûrement déjà été "ennuyés" par des "resets intempestifs".

Ces accidents, qu'ils soient dus à la machine ou à l'homme-qui-programme-la-machine, arrivent toujours au début d'un programme (vous savez, après la 400ème ligne environ).

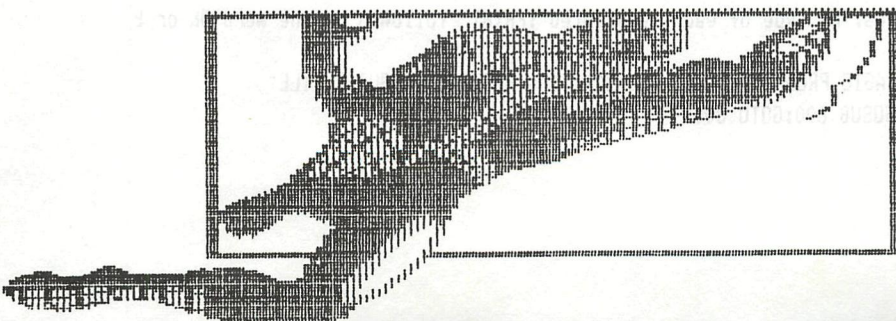
Voici donc un petit programme qui vous permettra de "récupérer" à 99,999 % des cas un programme "planté" (à condition qu'il ne se soit pas auto-détruit).

Voici la procédure à suivre :

- 1 - Ne pas paniquer ni jurer.
- 2-- Faire reset (si ce n'est pas déjà fait).
- 3 - Taper UT
Z3
S29B 02-00 EC-A0
B
NEW
- 4 - Charger le programme "What to do when a crash occurs".
- 5 - Faire "RUN".
- 6 - A la question "START OF HEAP?", taper- 2EC pour un programme normal;
-un autre chiffre si la mémoire était protégée par du langage machine
ex : 900 pour FGT.
- 7 - Prier.
- 8 - Le programme donne : HEAP
START OF TEXT
START OF SYMBOL TABLE
END OF SYMBOL TABLE
- 9 - Remplacer les pointeurs de 29B et 2A4 en n'oubliant pas d'inverser LSB et MSB.
- 10- Revenir au basic.
- 11- Faire "LIST".
- 12- Si rien ne se passe, jurer un bon coup!



Robert SIP.



```

5 CLEAR 100
6 POKE #75,32
10 ENVELOPE 0 15:ENVELOPE 1 15:MODE 0:PRINT CHR$(12);:COLORT 8 8 8:F#= " format : "
110 PRINT " *** D A I B A S I C T U T O R ***":GOSUB 600:PRINT
140 PRINT " select a number ... (2 digits)":GOSUB 600
150 PRINT " 01 CHECK 02 CLEAR 03 COLORG 04 COLORT 05 CONT"
160 PRINT " 06 CURSOR 07 DATA 08 DIM 09 DOT 10 DRAW"
170 PRINT " 11 EDIT 12 END 13 ENVELOPE 14 FILL 15 FOR NEXT"
180 PRINT " 16 GOSUB 17 GOTO 18 IF GOTO 19 IF THEN 20 IMP"
190 PRINT " 21 INPUT 22 LET 23 LIST 24 LOAD 25 LOADA"
200 PRINT " 26 MODE 27 NEW 28 NEXT 29 NOISE 30 ON GOSUB"
210 PRINT " 31 ON GOTO 32 OUT 33 POKE 34 PRINT 35 READ"
220 PRINT " 36 REM 37 RESTORE 38 RETURN 39 RUN 40 SAVE"
230 PRINT " 41 SAVEA 42 SOUND 43 STOP 44 TALK 45 TROFF"
240 PRINT " 46 TRON 47 WAIT 48 UT 49 ABS 50 ACDS"
250 PRINT " 51 ALOG 52 ASC 53 ASIN 54 ATN 55 CHR#"
260 PRINT " 56 COS 57 CURX 58 CURY 59 EXP 60 FRAC"
270 PRINT " 61 FRE 62 FREQ 63 GETC 64 HEX# 65 INP"
280 PRINT " 66 INT 67 LEFT# 68 LEN 69 LOG 70 LOG1"
290 PRINT " 71 MID# 72 PDL 73 PEEK 74 PI 75 RIGHT#"
300 PRINT " 76 RND 77 SCRN 78 SGN 79 SIN 80 SPC"
310 PRINT " 81 SQR 82 STR# 83 TAB 84 TAN 85 VAL"
320 PRINT " 86 VARPTR 87 XMAX 88 YMAX 89 MOD 90 IOR"
330 PRINT " 91 IAND 92 IXOR 93 INOT 94 SHL 95 SHR";
332 POKE #BEE2,204:POKE #BDD6,200:POKE #BFEE,205
340 COLORT 8 0 12 0
405 M#="":FOR X=1.0 TO 2.0
420 G=GETC:IF G=0.0 THEN 420
430 IF G<48.0 OR G>57.0 THEN 420:M#=M#+CHR$(G):WAIT TIME 3
440 SOUND 1 0 15 0 FREQ(400.0*X):G=0.0:WAIT TIME 3:G=0.0:SOUND OFF :NEXT:M=VAL(M#):IF M<1 THEN 405
460 P%=#BFEE+10-(5*#86)-(INT(N/5.0))*#86-(INT(M MOD 5.0)*24.0):IF M MOD 5.0=0.0 THEN P%=P%+14
470 FOR X=P%-18 TO P%+2 STEP 2:POKE X,##F:NEXT:WAIT TIME 100:MODE 0:PRINT CHR$(12);
510 ON M GOTO 1100,1200,1300,1400,1500,1600,1700,1800,1900,2000
520 ON M-10 GOTO 2100,2200,2300,2400,2500,2600,2700,2800,2900,3000
530 ON M-20 GOTO 3100,3200,3300,3400,3500,3600,3700,2500,3900,4000
540 ON M-30 GOTO 4100,4200,4300,4400,1700,4600,1700,2600,4900,5000
550 ON M-40 GOTO 5100,5200,5300,5400,5500,5500,5700,5800,5900,6000
560 ON M-50 GOTO 6100,6200,6000,6000,6500,6600,6700,6800,6900,7000
570 ON M-60 GOTO 7100,5200,7300,7400,7500,7600,7700,7700,7900,7900
580 ON M-70 GOTO 7700,8200,8300,8400,7700,8600,8700,8800,8600,9000
590 ON M-80 GOTO 9100,9200,9300,6600,9200,9600,3600,3600,9900,10000
595 IF M>90 GOTO 10000
600 FOR X=1.0 TO 60.0:PRINT CHR$(11);:NEXT:PRINT :RETURN
800 CURSOR 30,1:PRINT "spacebar to continue ..."
830 G=GETC:IF G<>32.0 THEN 830
840 GOTO 10
850 CURSOR 30,1:PRINT "spacebar to continue...";
860 G=GETC:IF G=0.0 THEN 860
870 RETURN
900 POKE #BFEE,198:POKE #BE5C,200:GOSUB 600:PRINT TAB(25);A#:GOSUB 600:RETURN

```

CHECK

```

1100 A#="CHECK":GOSUB 900:PRINT "The CHECK command scans a cassette tape or disc and "
1120 PRINT "examines all the files.The type of each is printed":PRINT "followed by the word Ok or B AD."
1130 PRINT :PRINT TAB(20);"0 = BASIC PROGRAM":PRINT TAB(20);"1 = MACHINE LANGUAGE FILE"
1140 PRINT TAB(20);"2 = ARRAY":GOSUB 600:GOTO 800

```

CLEAR

```
1200 A#="CLEAR":GOSUB 900:PRINT F#:TAB(25);"10 CLEAR 4000"  
1217 PRINT TAB(25);"CLEAR 4000":GOSUB 600  
1220 PRINT "Resets all variables to 0 or null string, and returns"  
1230 PRINT "all space assigned to arrays. The size of the HEAP "  
1240 PRINT "is set to the number specified by the CLEAR statement."  
1250 PRINT "minimum = 4 , maximum = 32767"  
1255 PRINT "If you get 'OUT OF STRING SPACE', the HEAP is too small":GOSUB 600:GOTO 800
```

COLORG

```
1300 A#="COLORG":GOSUB 900:PRINT F#:TAB(20);"10 COLORG 0 5 10 15 (default values)"  
1317 PRINT TAB(20);"COLORG 1 3 5 8":GOSUB 600  
1320 PRINT "Sets the 4 colours available in 4 COLOR modes (2,4,6)"  
1330 PRINT "If the screen is already in 4 COLOR mode,"  
1340 PRINT "then the colour change will be immediate"  
1345 PRINT "The first colour is background colour":GOSUB 600:GOSUB 850  
1360 MODE 2A:PRINT CHR$(12);:COLORG 0 1 3 5  
1370 FOR X=0.0 TO 10.0:FOR Y=0.0 TO 10.0:FILL Y#4,X#4 Y#4+2,X#4+2 21+RND(3.0):NEXT: NEXT  
1375 PRINT CHR$(12);TAB(10);"COLORG 0 1 3 5":GOSUB 850  
1377 PRINT CHR$(12);TAB(10);"COLORG 1 5 12 14":COLORG 1 5 12 14:GOSUB 850  
1379 PRINT CHR$(12);TAB(10);"COLORG 5 0 1 13":COLORG 5 0 1 13:GOTO 800
```



COLORT

```
1400 GOSUB 600:PRINT TAB(25);"COLORT":GOSUB 600  
1410 PRINT "Sets up 4 colours for character mode (MODE 0)"  
1420 PRINT "First colour is background colour":PRINT "Second colour is character colour"  
1430 PRINT "Colours nrs 3 and 4 are used if you POKE #FF"  
1435 PRINT "into the colour byte of the character":GOSUB 600  
1450 PRINT F#:TAB(25);"10 COLORT 8 0 12 0":PRINT TAB(25);"COLORT 8 0 12 0":GOSUB 600:PRINT :PRINT  
1460 FOR X=#BC68 TO #BC7A STEP 2.0:POKE X,#FF:NEXT:LIST 1460:GOSUB 850  
1465 CURSOR 5,5:PRINT "COLORT 12 0 13 1 ":COLORT 12 0 13 1:GOSUB 850  
1480 CURSOR 5,5:PRINT "COLORT 5 0 7 0 ":COLORT 5 0 7 0:GOTO 800
```

CONT

```
1500 A#="CONT":GOSUB 900:PRINT "Continues a BASIC program execution "  
1520 PRINT "with the next statement following ":PRINT "the STOP statement or BREAK position"  
1540 PRINT "You can't CONT if you used UT,EDIT"  
1550 PRINT "You can LIST or PRINT variables and then CONT":GOSUB 600:GOTO 800
```

CURSOR

```
1600 A#="CURSOR":GOSUB 900:PRINT "Moves the CURSOR to the X,Y position"  
1620 PRINT "given in the statement":PRINT " maximum X = 59 , maximum Y = 23":GOSUB 600:GOSUB 850  
1630 CURSOR 10,4:PRINT CHR$(1);"CURSOR 10,4":GOSUB 850:CURSOR 25,10:PRINT CHR$(1);"CURSOR 25,10":GOSUB 850  
1645 CURSOR 40,15:PRINT CHR$(1);"CURSOR 40,14":GOTO 800
```

DATA

READ

RESTORE

```
1700 A#="DATA":GOSUB 900:PRINT "Specifies information to be used by READ"  
1720 PRINT "DATA must give the same type of information":PRINT "as READ is asking for "  
1735 PRINT "RESTORE points back to the first DATA-information":GOSUB 600:RESTORE  
1750 DATA 999,VIS,75,FELLOW,10000,HERE YOU ARE
```

```

1760 LIST 1750:LIST 1770:PRINT :PRINT
1770 FOR X%=1 TO 3:READ A#,A#:PRINT TAB(10);A#,A#:NEXT
1772 P%=#BFEE-(#86#8):P1%=P%-30:P2%=P%-58-#86
1774 FOR X%=P1% TO P1%+10 STEP 2:POKE X%,#FF:NEXT
1776 FOR X%=P2%+2 TO P2%+10 STEP 2:POKE X%,#FF:NEXT:GOTO 800

```

DIM

```

1800 A#="DIM":GOSUB 900:PRINT "DIM allocates space for arrays          A(250)"
1802 PRINT "arrays can have more than one suscript eg:A$(10,5,20)"
1803 PRINT "the HEAP is used to store the arrays":PRINT "EDIT or RUN clears the arrays":GOSUB 600:G
OTO 800

```

DOT

```

1900 A#="DOT":GOSUB 900:PRINT "Places a dot of colour C at the position X,Y"
1912 PRINT "The size of the DOT will depend upon the current MODE"
1914 GOSUB 600:PRINT :PRINT F#:TAB(25);"DOT X,Y C":GOSUB 600:GOSUB 850
1920 COLORG 0 1 3 5:MODE 1A:PRINT CHR$(12):CURSOR 25,3:PRINT "MODE 1A":GOSUB 1980
1930 MODE 3A:PRINT CHR$(12):CURSOR 25,3:PRINT "MODE 3A":GOSUB 1980
1940 MODE 5A:PRINT CHR$(12):CURSOR 25,3:PRINT "MODE 5A":GOSUB 1980:GOTO 800
1980 GOSUB 600:A%=10:B%=10:IF M=10 THEN 1990
1981 FOR X%=1 TO 15
1982 DOT A#,B% X#:CURSOR 5,1:PRINT "DOT ";A#;" ";B%;X#;" "
1983 A%=A%+4:B%=B%+3:GOSUB 850:NEXT:RETURN
1990 FOR X%=1 TO 15
1992 DRAW A#,B% A%+5,B% X#:CURSOR 5,1:PRINT "DRAW ";A#;" ";B%;A%+5;" ";B%;X#;" "
1994 A%=A%+4:B%=B%+3:GOSUB 850:NEXT:RETURN

```

DRAW

```

2000 A#="DRAW":GOSUB 900:PRINT "Draws a line in colour C between X1,Y1 and X2,Y2"
2020 PRINT "There is no restriction on the order of the coordinates":GOSUB 600
2040 PRINT F#:TAB(25);"DRAW X1,Y1 X2,Y2 C":GOSUB 600:GOSUB 850:GOTO 1920

```

EDIT

```

2100 A#="EDIT":GOSUB 900:PRINT "the program has stopped"
2120 PRINT "You are in command now":PRINT "Go to EDIT MODE with : EDIT 2110-2190 <return>"
2140 PRINT "Use the cursorkeys to make modifications to this text"
2150 PRINT "If you end with BREAK BREAK,":PRINT "Then no modifications are effective"
2170 PRINT "If you want some modifications then do BREAK SPACE"
2180 PRINT "To see the results type LIST 2110-2190 <return>"
2190 PRINT "Type RUN <return> to continue the program":GOSUB 600:END

```

END

```

2200 A#="END":GOSUB 900:PRINT "Terminates the execution of a BASIC program."
2220 PRINT "An 'END PROGRAM' message is displayed"
2230 PRINT "Type RUN <return> to start again":GOSUB 600:END

```

ENVELOPE

```

2300 A#="ENVELOPE":GOSUB 900:PRINT "format : ENVELOPE nr (V,T);(V,T);(V,T);V"
2320 PRINT "          ENVELOPE nr (V,T);(V,T);(V,T);":GOSUB 600
2330 PRINT "nr is an expression in the range 0 to 1"

```

```

2335 PRINT "V is a volume level in the range 0 to 15"
2340 PRINT "T is the time for volume V in the range 1 to 254"
2345 PRINT "T is in units of 3.2 milliseconds":GOSUB 600:GOSUB 850
2355 MODE 6A:COLORG 0 15 12 14:FOR X%=21 TO 23:A%=100
2361 PRINT CHR$(12);:PRINT "ENVELOPE 1 ";
2362 Y%=(X%-20)*50:Q%=Y%:DRAW 100,Y% 235,Y% X%
2364 V%=RND(16.0):T%=RND(30.0)
2365 Z%=Y%+V%*2:DRAW A%,Z% A%+T%,Z% X%:DRAW A%,Z% A%,Q% X%:Q%=Z%
2367 PRINT "(";V%";";T%";")";:SOUND 1 0 V% 0 FREQ(800.0):WAIT TIME T%
2370 A%=A%+T%:IF A%+20>=XMAX-100 THEN GOSUB 850:GOTO 2380
2372 GOTO 2364
2380 NEXT:SOUND OFF :GOTO 800

```

FILL

```

2400 A#="FILL":GOSUB 900:PRINT "format : FILL X1,Y1 X2,Y2 C":GOSUB 600
2420 PRINT "Fills the rectangle with opposite corners "
2425 PRINT "There is no restriction on the order of the points":GOSUB 600:GOSUB 850
2440 COLORG 8 1 3 5:MODE 3A:FOR X%=0 TO 15:FILL 0,X%*6 XMAX,X%*6+4 X%
2465 CURSOR 4,2:PRINT "FILL 0,";X%*6;XMAX;",";X%*6+4;X%:" :GOSUB 850
2477 IF X%=0.0 THEN FILL 0,0 XMAX,YMAX 0
2480 NEXT:GOTO 800

```

FOR NEXT

```

2500 A#="FOR NEXT":GOSUB 900:PRINT "To set up a loop a number of times"
2520 PRINT "A STEP value is optional":GOSUB 600:GOSUB 850
2540 PRINT CHR$(12);:A#="SOME EXAMPLES...":GOSUB 900:LIST 2560:GOSUB 600
2560 FOR X%=1 TO 10:PRINT X%:NEXT:GOSUB 850
2562 PRINT CHR$(12);:GOSUB 600:LIST 2564:GOSUB 600
2564 FOR X%=10 TO 200 STEP 10:PRINT X%:NEXT:GOSUB 850
2567 PRINT CHR$(12);:GOSUB 600:LIST 2569:GOSUB 600
2569 FOR X%=30 TO 13 STEP -1:FOR Y%=5 TO 20:PRINT X%+Y%::NEXT:PRINT :NEXT:GOTO 800

```

GOSUB RETURN

```

2600 A#="GOSUB":GOSUB 900:PRINT F#;TAB(25);"40 GOSUB 1000":GOSUB 600
2620 PRINT "Branches to the specified linenumber.":PRINT "When a return statement is encountered th
e next statement"
2624 PRINT "executed is the statement following the GOSUB."
2626 PRINT "To be safe,only leave the subroutine with RETURN":GOSUB 600:LIST 2630-2640
2629 PRINT :PRINT :PRINT CHR$(1);:LIST 2670-2674
2630 FOR X=1.0 TO 3.0
2634 CURSOR 1,1:INPUT "type a name <return>";A#
2635 GOSUB 2670:REM to the subroutine
2640 NEXT:GOTO 800
2670 CURSOR CURX-LEN(A#),CURY-1
2672 FOR X=LEN(A#)-1.0 TO 0.0 STEP -1.0:PRINT MID$(A#,X,1);
2674 NEXT:RETURN

```

GOTO

```

2700 A#="GOTO":GOSUB 900:PRINT F#;TAB(25);"20 GOTO 1000":GOSUB 600
2720 PRINT "Branches to the specified linenumber":GOSUB 600:GOTO 800

```



IF GOTO IF THEN

```
2800 A#="IF THEN / IF GOTO":GOSUB 900
2805 PRINT F#;TAB(25);"20 IF A=5 THEN SOUND OFF"
2806 PRINT TAB(25);"40 IF A=5 THEN 100":PRINT TAB(25);"60 IF A=5 GOTO 100":GOSUB 600
2808 PRINT "If the condition is true,execute following statement"
2809 PRINT "IF THEN can be followed by a statement or a linenumber"
2810 PRINT "IF GOTO must be followed by a linenumber":GOSUB 600:PRINT "attention":PRINT "-----"
2830 PRINT "with IF GOTO lnr. and IF THEN lnr.,if the condition is false":PRINT "the next statement
on the same line is executed"
2850 PRINT "This does not apply for IF THEN GOTO":GOSUB 600:GOTO 800
```

IMP

```
3000 A#="IMP":GOSUB 900:PRINT F#;TAB(25);"IMP INT":PRINT TAB(25);"IMP INT M-Z"
3007 PRINT TAB(25);"IMP FPT":PRINT TAB(25);"IMP FPT L-0":PRINT TAB(25);"IMP STR A-D":GOSUB 600
3012 PRINT "If the IMP statement is never used,":PRINT "DAIpc is im IMPLIED FLOATING POINT"
3016 PRINT "In this situation A=A! I=I! Z=Z!":PRINT "I%, A%, Z% are integer and distinct from I,
A, Z"
3020 PRINT "I%, A%, Z% are string variables":GOSUB 600
3030 PRINT "note the different formats ":PRINT TAB(10);"FPT";TAB(25);"INT";TAB(40);"STR"
3040 FOR X%=1 TO 5:PRINT TAB(10);A%;TAB(25);A%;TAB(40);A%
3050 A=A+1.23456E5:A%=A%+12345669:A%=A#+CHR$(65+RND(20.0)):NEXT:GOTO 800
```

INPUT

```
3100 A#="INPUT":GOSUB 900:PRINT F#;TAB(25);"20 INPUT A"
3115 PRINT TAB(25);"30 INPUT A#":PRINT TAB(25);"40 INPUT B,B#,B%"
3125 PRINT TAB(25);"50 INPUT 'HOW MANY TIMES';I":PRINT TAB(25);"60 INPUT 'WHAT'S YOUR NAME';NAME#":
GOSUB 600
3150 PRINT "Requests data from the terminal to be typed in."
3160 PRINT "Data typed in must be of the same type as the variable(s)":GOTO 800
```

LET

```
3200 A#="LET":GOSUB 900:PRINT F#;TAB(25);"LET W=1234"
3220 PRINT TAB(25);"LET Q#='?CROCODILE'":PRINT TAB(25);"M=9876"
3240 PRINT TAB(25);"HOME#='DAInamic'":GOSUB 600
3250 PRINT "Assigns a value to a variable":PRINT "LET is optional":GOSUB 600:GOTO 800
```

LIST

```
3300 A#="LIST":GOSUB 900:PRINT F#;TAB(15);"LIST :      display the entire program"
3315 PRINT TAB(15);"LIST 100";TAB(30);"display line 100"
3320 PRINT TAB(15);"LIST 100-";TAB(30);"display from 100 to the end"
3325 PRINT TAB(15);"LIST 100-110";TAB(30);"display from 100 to 110"
3330 PRINT TAB(15);"LIST -100";TAB(30);"display from start to 100"
3335 PRINT TAB(15);"30 LIST 999";TAB(30);"with linenumber":GOSUB 600
3340 PRINT "Display can be made to pause by pressing any key"
3350 PRINT "Pressing spacebar will continue listing":GOSUB 600:GOTO 800
```

LOAD

```
3400 A#="LOAD":GOSUB 900:PRINT F#;TAB(25);"LOAD":PRINT TAB(25);"LOAD:RUN"
3420 PRINT TAB(25);"10 LOAD":PRINT TAB(25);"10 LOAD 'TEST'":GOSUB 600
3445 PRINT "When BASIC encounters a LOAD command,":PRINT "It loads the first program from tape"
3449 PRINT "The old program is deleted ":PRINT "and the new one starts when loading is finished":GO
SUB 600:GOTO 800
```

LOADA

```
3500 A#="LOADA":GOSUB 900:PRINT F#;TAB(25);"10 LOADA A# 'APPLE'"
3517 PRINT TAB(25);"20 LOADA M#":PRINT TAB(25);"LOADA M# 'TRIAL':":GOSUB 600
3520 PRINT "Loads an array from tape/disc":GOTO 800
```



MODE

XMAX

YMAX

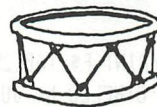
```
3600 A#="MODE":GOSUB 900:PRINT F#;TAB(25);"MODE 5A"
3607 PRINT TAB(25);"100 MODE 3":GOSUB 600:COLORG 0 1 3 12
3610 PRINT "Sets up display format":PRINT "There are 13 modes available :":GOSUB 600
3625 PRINT "MODE GRAPHIC RESOLUTION";TAB(30);"4/16 COLOR REQUIRED SPACE":PRINT
3627 PRINT " XMAX YMAX"
3630 PRINT " 0 24 x 60 characters";TAB(30);"both 3.5K "
3632 PRINT " 1 72 x 65";TAB(30);"16 1.5K"
3636 PRINT " 2 72 x 65";TAB(30);"4 1.5K"
3640 PRINT " 3 160 x 130";TAB(30);"16 5.8K"
3644 PRINT " 4 160 x 130";TAB(30);"4 5.8K"
3648 PRINT " 5 336 x 256";TAB(30);"16 22.8K"
3652 PRINT " 6 336 x 256";TAB(30);"4 22.8K"
3654 PRINT :PRINT " 1A,2A,3A,4A,5A,6A same as 1,2,3,4,5,6 but split MODE":GOSUB 600:GOSUB 850
3660 PRINT CHR$(12):PRINT TAB(25);"MODE 1A":MODE 1A:GOSUB 3690:REM 16 C
3662 PRINT CHR$(12):PRINT TAB(25);"MODE 2A":MODE 2A:GOSUB 3695:REM 4C
3664 PRINT CHR$(12):PRINT TAB(25);"MODE 3A":MODE 3A:GOSUB 3690:PRINT CHR$(12):PRINT TAB(25);"MODE 4
A ":MODE 4A:GOSUB 3695
3666 PRINT CHR$(12):PRINT TAB(25);"MODE 5A":MODE 5A:GOSUB 3690:PRINT CHR$(12):PRINT TAB(25);"MODE 6
A":MODE 6A:GOSUB 3695
3670 GOTO 800
3690 CURSOR 2,1:PRINT "XMAX =";XMAX;" YMAX =";YMAX;" "
3692 FOR X%=0 TO YMAX STEP 2:DRAW 0,X% XMAX,X% RND(16.0):NEXT:GOSUB 850:RETURN
3695 CURSOR 2,1:PRINT "XMAX =";XMAX;" YMAX =";YMAX;" "
3696 FOR X%=1 TO 10:M%=XMAX/20:FILL XMAX-3*M%,X%M% XMAX,X%M%+M%/2 RND(3)+21:NEXT
3697 FOR X%=0 TO XMAX STEP 3:DRAW 0,0 X%,3*YMAX/4 RND(3.0)+21.0:NEXT:GOSUB 850:RETURN
```

NEW

```
3700 A#="NEW":GOSUB 900:PRINT "Deletes current BASIC program and resets all variables":GOSUB 600:GO
TO 800
```

NOISE

```
3900 A#="NOISE":GOSUB 900:PRINT F#;TAB(25);"NOISE E V"
3915 PRINT TAB(25);"NOISE OFF":GOSUB 600:PRINT "Gives white noise on both channels"
3927 PRINT "following ENVELOPE E(0/1) with volume V":GOSUB 600:GOSUB 850
3930 PRINT CHR$(12):LIST 3935-3940
3935 ENVELOPE 0 15,5;0,15;
3937 NOISE 0 15:WAIT TIME 200
3940 NOISE OFF
3945 GOSUB 850:CURSOR 0,15:GOSUB 600:LIST 3955-3960
3955 ENVELOPE 1 15
3957 FOR Z%=1 TO 10:FOR X%=5 TO 15
3958 NOISE 1 X%:WAIT TIME 5
3960 NEXT:NEXT:NOISE OFF
3970 GOTO 800
```



ON GOSUB

```
4000 A#=" ON GOSUB":GOSUB 900:PRINT F#;TAB(25);": ON A GOSUB 100,150,200,250,...":GOSUB 600
4030 CURSOR 0,15:PRINT "If A=1 then GOSUB 100 ( GOTO 100)"
4035 PRINT "If A=2 then GOSUB 150":PRINT "If A=3 then GOSUB 200"
4040 PRINT "If A=4 then GOSUB 250":GOSUB 600:PRINT "If A<=0 or A>number of line numbers"
4060 PRINT "then the following statement is executed":GOTO 800
```

ON GOTO

```
4100 A#=" ON GOTO ":GOSUB 900:PRINT F#;TAB(25);"ON A GOTO 100,150,200,250"
4120 GOSUB 600:GOSUB 850:GOTO 4030
```

OUT

```
4200 A#="OUT":GOSUB 900:PRINT F#;TAB(25);"OUT A,B":GOSUB 600
4220 PRINT "Sends the number in variable B to the DCE-bus"
4230 PRINT "The 8255 IC is used following the DCE-concept":GOSUB 600:GOTO 800
```

POKE

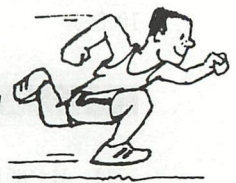
```
4300 A#="POKE":GOSUB 900:PRINT F#;TAB(25);"POKE #BF00,#FF"
4315 PRINT TAB(25);"POKE A,B":PRINT TAB(25);"POKE 40000,216":GOSUB 600
4330 PRINT "Stores value B into memory location A":GOSUB 600:LIST 4350:GOSUB 850
4350 POKE #B935,65
4360 GOSUB 850:CURSOR 0,10:LIST 4370:GOSUB 850:CURSOR 0,8
4370 POKE #B932,#FF
4380 GOTO 800
```

PRINT

```
4400 A#="PRINT":GOSUB 900:PRINT F#;TAB(25);"PRINT A,B,C"
4420 PRINT TAB(25);"PRINT":PRINT TAB(25);"PRINT 'TOTAL ' ; T"
4440 PRINT TAB(25);"? TOTAL":GOSUB 600:LIST 4450-4470
4450 PRINT "4450":FOR X%=1 TO 5:PRINT X%;NEXT
4460 PRINT "4460":FOR X%=1 TO 5:PRINT X%;NEXT:PRINT
4470 PRINT "4470":FOR X%=1 TO 5:PRINT X%,NEXT
4480 GOTO 800
```

REM

```
4600 A#="REM":GOSUB 900:PRINT F#;TAB(25);"350 INPUT A#:REM ASK NAME of player":GOSUB 600
4610 PRINT "Allows comments inside BASIC programs"
4620 PRINT "REM statements are not executed":GOSUB 600:GOTO 800
```



RUN

```
4900 A#="RUN":GOSUB 900:PRINT F#;TAB(25);"RUN":PRINT TAB(25);"RUN 100":GOSUB 600
4930 PRINT "Starts execution of a BASIC program in memory":PRINT "All variables and arrays are cleared":GOTO 800
```

SAVE

```
5000 A#="SAVE":GOSUB 900:PRINT F#;TAB(25);"SAVE"
5020 PRINT TAB(25);"SAVE 'OTHELLO by JUL KABAS':PRINT TAB(25);"SAVE M#":GOSUB 600
5040 PRINT "Saves on cassette or disc the current program in memory"
5050 PRINT "You get the message : SET RECORD,START TAPE,TYPE SPACE":GOTO 800
```

SAVEA

```
5100 A#="SAVEA":GOSUB 900:PRINT F#;TAB(25);"SAVEA A# 'TUNE"
5115 PRINT TAB(25);"10 SAVEA A# 'TUNE':PRINT TAB(25);"10 SAVEA A#":GOSUB 600
5130 PRINT "Saves an array on tape or disc":GOSUB 600:GOTO 800
```


SOUND FREQ

```

5200 A$="SOUND":GOSUB 900:PRINT :PRINT F#;TAB(11);" SOUND 1 0 5 0 FREQ(440)"
5230 FOR X%=18.0 TO 5.0 STEP -1.0:CORSOR 19,X%:PRINT CHR#(10);:NEXT
5235 PRINT "Oscillator number : 0 , 1 or 2"
5240 FOR X%=18.0 TO 8.0 STEP -1.0:CORSOR 21,X%:PRINT CHR#(10);:NEXT
5245 PRINT "Envelope number : 0 or 1"
5247 FOR X%=18.0 TO 11.0 STEP -1.0:CORSOR 23,X%:PRINT CHR#(10);:NEXT
5249 PRINT "Volume 0.....15"
5251 FOR X%=18.0 TO 14.0 STEP -1.0:CORSOR 25,X%:PRINT CHR#(10);:NEXT
5253 PRINT "Mode 0,1,2,3"
5255 FOR X%=18.0 TO 17.0 STEP -1.0:CORSOR 27,X%:PRINT CHR#(10);:NEXT
5257 PRINT "Frequency in Hertz":GOSUB 850:PRINT CHR#(12);
5259 A$="SOUND 2":GOSUB 900
5260 SOUND 1 0 15 0 FREQ(440.0)
5261 LIST 5260:GOSUB 5299
5262 SOUND 1 0 15 1 FREQ(440.0)
5264 LIST 5262:GOSUB 5299
5266 SOUND 1 0 15 2 FREQ(31.0):SOUND 1 0 15 2 FREQ(15000.0)
5268 LIST 5266:GOSUB 5299:GOTO 800
5299 WAIT TIME 100:SOUND OFF :RETURN

```



STOP

```

5300 A$="STOP":GOSUB 900:PRINT F#;TAB(25);"100 STOP":GOSUB 600
5320 PRINT "Basic suspends execution of the program":PRINT "and enters the command mode"
5340 PRINT "To continue type CONT":GOSUB 600:GOTO 800

```

TALK

```

5400 A$="TALK":GOSUB 900:PRINT F#;TAB(25);"TALK #400":GOSUB 600
5420 PRINT " By building up a table of parameters in memory"
5430 PRINT " and sending this information to the oscillators"
5440 PRINT " by the TALK command,":PRINT " It is possible to create vocal sounds":GOTO 800

```

TROFF TRON

```

5500 A$="TRON TROFF":GOSUB 900:PRINT F#;TAB(25);"10 TRON":PRINT TAB(25);"100 TROFF":GOSUB 600
5515 TRON
5520 PRINT "When BASIC encounters a TRON command,"
5530 PRINT "all lines in execution are displayed"
5540 PRINT "until a TROFF command is given"
5550 TROFF
5555 GOSUB 600:PRINT "TRACE WAS ON for linenumbers 5520-5540":GOTO 800

```

WAIT

```

5700 A$="WAIT":GOSUB 900
5710 PRINT F#;TAB(25);"WAIT I,J,K (1)":PRINT TAB(25);"WAIT MEM I,J,K (2)"
5715 PRINT TAB(25);"WAIT TIME I (3)":PRINT TAB(25);"K is optional":GOSUB 600
5720 PRINT "(1) Read status of DCE port I,exclusive OR's with K"
5730 PRINT " and AND's the result with J until result equals J":GOSUB 600
5740 PRINT "(2) As (1), but I is a memory location"
5750 PRINT " example : event check :WAIT MEM #F000,#10":GOSUB 600
5760 PRINT "(3) Delays program execution for a time I"
5780 PRINT " Time is in units of 20 milliseconds":GOSUB 600:GOTO 800

```

UT

```

5800 A#="UT":GOSUB 900:PRINT F#;TAB(25);"UT (command model)":GOSUB 600
5820 PRINT "Calls the Machine Language Monitor":PRINT "with this facilities:";TAB(30);"L LOOK"
5840 PRINT TAB(30);"D DISPLAY":PRINT TAB(30);"G GO":PRINT TAB(30);"F FILL"
5850 PRINT TAB(30);"S SUBSTITUTE":PRINT TAB(30);"M MOVE":PRINT TAB(30);"X EXAMINE"
5860 PRINT TAB(30);"V VECTOR EXAMINE":PRINT TAB(30);"R READ":PRINT TAB(30);"W WRITE"
5870 PRINT TAB(30);"Z INITIALISE":GOSUB 600:GOTO 800

```

ABS

```

5900 A#="ABS":GOSUB 900:PRINT F#;TAB(25);"A=ABS(B)":GOSUB 600:PRINT :PRINT
5920 FOR X%=1 TO 10:A%=RND(7657):IF RND(1)<0.5 THEN A%=A%*(-1)
5940 PRINT TAB(10);"A = ";A%;TAB(22);CHR$(10);TAB(25);" ABS(A) = ";ABS(A%);NEXT:GOTO 800

```

ACOS ASIN ATN

```

6000 A#="ASIN ACOS ATN":GOSUB 900:PRINT F#;TAB(25);"A=ACOS(B) A=ASIN(B) A=ATN(B)":GOSUB 600
6020 PRINT "Returns arc sine,cosine,tangent of argument":PRINT "Result is between -PI/2 and PI/2":G
GOSUB 600
6030 PRINT "argument";TAB(15);"arc sine";TAB(30);"arc cosine";TAB(45);"arc tangent":GOSUB 600
6040 FOR X%=1 TO 10:A=RND(1):IF RND(1)<0.5 THEN A=A*(-1)
6050 PRINT A;TAB(15);ASIN(A);TAB(30);ACOS(A);TAB(45);ATN(A);NEXT:GOTO 800

```

ALOG

```

6100 A#="ALOG":GOSUB 900:PRINT F#;TAB(25);"A=ALOG(B)":GOSUB 600
6120 PRINT "Returns antilog base 10 of argument":GOSUB 600
6130 PRINT :PRINT :FOR X%=1 TO 10:A=RND(7)
6140 PRINT TAB(3);"argument = ";A;TAB(25);CHR$(10);TAB(28);"antilog = ";ALOG(A);NEXT:GOTO 800

```

ASC

```

6200 A#="ASC":GOSUB 900:PRINT F#;TAB(25);"A = ASC(X#)":GOSUB 600
6220 PRINT "Returns the integer ASCII value of the first character"
6230 PRINT "of the string X":GOSUB 600
6235 FOR X%=0 TO 3:PRINT TAB(X%*15);"str";TAB(X%*15+5);"asc";:NEXT:PRINT :GOSUB 600
6240 FOR X%=48 TO 91 STEP 4:FOR Y%=0 TO 3
6260 PRINT TAB(Y%*15);CHR$(34);CHR$(X%+Y%);CHR$(34);TAB(Y%*15+5);X%+Y%;
6270 NEXT:PRINT :NEXT:GOTO 800

```

CHR\$

```

6500 A#="CHR$":GOSUB 900:PRINT F#;TAB(25);"A#=CHR$(B)":GOSUB 600
6520 PRINT "Returns a character whose value is B":GOSUB 600:GOTO 6235

```

COS SIN TAN

```

6600 A#="SIN COS TAN ":GOSUB 900:PRINT F#;TAB(25);"A = COS(B) A = SIN(B) A=TAN(B)":GOSUB 600
6620 PRINT "Returns the (co)sine ,tangent of the argument":GOSUB 600
6630 PRINT :PRINT "argument";TAB(15);"sine";TAB(30);"cosine";TAB(45);"tangent":GOSUB 600
6635 FOR X%=1 TO 10:A=RND(2*PI):IF RND(1)<0.5 THEN A=A*(-1)
6640 PRINT A;TAB(15);SIN(A);TAB(30);COS(A);TAB(45);TAN(A);NEXT:GOTO 800

```

CURX

```

6700 A#="CURX":GOSUB 900:PRINT F#;TAB(25);"A=CURX":GOSUB 600
6720 PRINT "Sets A to the X position of the CURSOR":GOSUB 600;GOSUB 850
6730 FOR X%=0 TO 59:CURSOR X%,10:PRINT CHR$(255):CURSOR 25,5:PRINT "CURY = ";X%:SOUND 1 0 15 0 FREQ
(31+X%*10)
6740 WAIT TIME 2:SOUND OFF :WAIT TIME 10:CURSOR X%,10:PRINT CHR$(32):NEXT:GOTO 800

```

CURY

```

6800 A#="CURY":GOSUB 900:PRINT F#;TAB(25);"A = CURY":GOSUB 600
6820 PRINT "Sets A to the Y position of the cursor":GOSUB 600;GOSUB 850
6830 PRINT CHR$(12);:FOR X%=1 TO 23:CURSOR 5,X%:PRINT CHR$(255)
6840 SOUND 1 0 15 0 FREQ(800):WAIT TIME 2:SOUND OFF
6850 CURSOR 40,10:PRINT " CURY = ";X%:WAIT TIME 10:CURSOR 5,X%:PRINT CHR$(32):NEXT:GOTO 800

```

EXP

```

6900 A#="EXP":GOSUB 900:PRINT F#;TAB(25);"A = EXP(X)":GOSUB 600
6920 PRINT "Returns the value 'e' (2.71828) to the power X":GOSUB 600
6930 FOR X%=1 TO 10:A=RND(20):IF RND(1)<0.5 THEN A1%=A#(-1)
6940 PRINT TAB(5);" X = ";A;TAB(25);CHR$(10);TAB(27);" e ^ X = ";EXP(A):NEXT:GOTO 800

```

FRAC

```

7000 A#="FRAC":GOSUB 900:PRINT F#;TAB(25);"A = FRAC(X)":GOSUB 600
7020 PRINT "Returns the floating point fractional part of the argument":GOSUB 600
7030 PRINT :PRINT :FOR X%=1 TO 10:A=RND(10000):IF RND(2)<1 THEN A=A#(-1)
7040 PRINT TAB(2);" argument = ";A;TAB(25);CHR$(10);TAB(27);" fraction = ";FRAC(A):NEXT:GOTO 800

```

FRE

```

7100 A#="FRE":GOSUB 900:PRINT F#;TAB(25);"B = FRE":GOSUB 600
7210 PRINT " Gives the number of bytes currently unused by BASIC":GOSUB 600;GOTO 800

```

GETC

```

7300 A#="GETC":GOSUB 900:PRINT F#;TAB(25);"G=GETC":GOSUB 600
7320 PRINT "G is set to the ASCII value of the character typed":PRINT "on the keyboard":GOSUB 600
7330 PRINT :PRINT :PRINT "Hit some keys...":PRINT :FOR X%=1 TO 10
7334 G%=GETC:IF G%=0 THEN 7334
7340 PRINT TAB(5);" you typed ";CHR$(G%);" ASCII value = ";G%:NEXT:GOTO 800

```

HEX\$

```

7400 A#="HEX$":GOSUB 900:PRINT F#;TAB(25);"A#HEX$(X)":GOSUB 600
7420 PRINT "Returns a string of characters representing ":PRINT "the hexadecimal value of the number X":GOSUB 600
7430 PRINT :PRINT :FOR X%=1 TO 10:A%=RND(65000)
7440 PRINT TAB(5);" value = ";A%;TAB(25);CHR$(10);TAB(27);" HEX$ = #";HEX$(A%):NEXT:GOTO 800

```

INP

```

7500 A#="INP":GOSUB 900:PRINT F#;TAB(25);"A= INP(31)":GOSUB 600
7520 PRINT "Reads the byte present in the DCE-bus CARD 3 PORT 1":PRINT "and assigns it to the variable A":GOSUB 600;GOTO 800

```

INT

```

7600 A#="INT":GOSUB 900:PRINT F#;TAB(25);"A=INT(B)":GOSUB 600
7620 PRINT "Returns the largest integral value less than":PRINT "or equal to the argument B":GOSUB
600
7630 FOR X%=1 TO 10:B=RND(1000)
7640 PRINT TAB(5);"value = ";B:TAB(25);CHR$(10);TAB(27);" integer = ";INT(B):NEXT:GOTO 800

```

LEFT\$

MID\$

RIGHT\$

LEN

```

7700 A#="LEFT$ MID$ RIGHT$ LEN":GOSUB 900
7710 PRINT F#;TAB(25);"B#=LEFT$(A#,3)":PRINT TAB(25);"B#=MID$(A#,3,2)"
7720 PRINT TAB(25);"B#=RIGHT$(A#,4)":PRINT TAB(25);"B=LEN(A#)":GOSUB 600
7730 PRINT "Returns a number of characters from B#":GOSUB 600:A#="SOUNDGENERATOR"
7750 CURSOR 0,13:PRINT "LEN(A#)=14":CURSOR 40,13:PRINT A#
7755 CURSOR 20,12:PRINT "LEFT$(A#,5) = ";TAB(40);:FOR X%=1 TO 5:PRINT CHR$(94);:NEXT
7757 CURSOR 40,11:PRINT LEFT$(A#,5):CURSOR 40,9:PRINT A#
7765 CURSOR 20,8:PRINT "MID$(A#,3,6) = ";TAB(43);:FOR X%=1 TO 6:PRINT CHR$(94);:NEXT
7766 CURSOR 43,7:PRINT MID$(A#,3,6):CURSOR 40,5:PRINT A#
7775 CURSOR 20,4:PRINT "RIGHT$(A#,4) = ";TAB(50);:FOR X%=1 TO 4:PRINT CHR$(94);:NEXT
7777 CURSOR 50,3:PRINT RIGHT$(A#,4):GOTO 800

```

LOG

LOGT

```

7900 A#="LOG LOGT":GOSUB 900:PRINT F#;TAB(25);"A=LOG(X)":PRINT TAB(25);"A=LOGT(X)":GOSUB 600
7930 PRINT " LOG returns the natural logarithm of the argument":PRINT " LOGT calculates the logarit
hm base 10":GOSUB 600
7950 FOR X%=1 TO 10:A%=RND(100000):PRINT TAB(1);" argument = ";A%;TAB(20);" LOG = ";LOG(A%);TAB(40)
;" LOGT = ";LOGT(A%):NEXT:GOTO 800

```

PDL

```

8200 A#="PDL":GOSUB 900:PRINT F#;TAB(25);"A=PDL(2)":GOSUB 600
8210 PRINT "Sets A to a number between 0 and 255,":PRINT "representing the position of potm. nr 2":
GOSUB 600:GOSUB 850
8230 MODE 6A:PRINT CHR$(12);:PRINT "you controll some of these lines..."
8235 FOR Y%=0 TO 5:FOR X%=1 TO 50
8240 A%=PDL(Y%):CURSOR Y%*10,2:PRINT "PDL";Y%;"=";A%;DRAW Y%*50,0 Y%*50,A% 23:WAIT TIME 5:DRAW Y%*5
0,0 Y%*50,A% 20
8250 NEXT:NEXT:GOTO 800

```

PEEK

```

8300 A#="PEEK":GOSUB 900:PRINT F#;TAB(25);"A=PEEK(B)":GOSUB 600
8310 PRINT "Sets A equal to the contents of adress B":GOSUB 600:GOTO 800

```

PI

```

8400 A#="PI":GOSUB 900:PRINT "Returns the floating point value 3.14159":GOTO 800

```

RND

```

8600 A#="RND":GOSUB 900:PRINT F#;TAB(25);"A=RND(100)":PRINT TAB(25);"A=RND(1)*100":PRINT TAB(25);"A
=RND(-1)":PRINT TAB(25);"A=RND(0)":GOSUB 600
8610 PRINT "Generates a software or hardware (RND(0)) random number":PRINT :FOR X%=1 TO 10:PRINT TA
B(10);"RND(100)=";TAB(20);RND(100):NEXT:GOTO 800

```

SCRN

```

8700 A#="SCRN":GOSUB 900:PRINT F#;TAB(25);"A=SCRN(X,Y)":GOSUB 600
8710 PRINT "Sets A equal to the colour number of the screen":PRINT "at coordinate X,Y":GOSUB 600
8720 GOSUB 850:MODE 1A:PRINT CHR$(12):FOR X%=0 TO 52 STEP 2:DOT 30,X% RND(15):NEXT
8730 FOR X%=0 TO 52 STEP 2:A%=SCRN(30,X%):DRAW 0,X% 25,X% A%:DRAW 35,X% XMAX,X% A%
8740 CURSOR 9,2:PRINT "SCRN (";30;",";X%;" )= ";A%;" " :WAIT TIME 50:NEXT:GOTO 800

```

SGN

```

8800 A#="SGN":GOSUB 900:PRINT F#;TAB(25);"A=SGN(X)":GOSUB 600
8810 PRINT "Sets A to 1 if X>0 , 0 if X=0 , -1 if X<0":GOSUB 600:GOTO 800

```

SPC

```

9000 A#="SPC":GOSUB 900:PRINT F#;TAB(25);"PRINT SPC(10);...":GOSUB 600
9010 PRINT "Returns a string of a number of spaces":GOSUB 600
9020 PRINT :LIST 9030:PRINT
9030 FOR X%=1 TO 6:PRINT TAB(RND(10));CHR$(10);R%=RND(40):PRINT SPC(R%);CHR$(10);"SPC(";R%;" )":PRI
NT :NEXT:GOTO 800

```

SQR

```

9100 A#="SQR":GOSUB 900:PRINT F#;TAB(25);"A=SQR(B)":GOSUB 600
9110 PRINT "Gives the square root of the argument":GOSUB 600
9120 PRINT TAB(20);"argument";TAB(40);"square root":GOSUB 600
9130 FOR X%=1 TO 10:B%=RND(100000):B=SQR(B%):PRINT TAB(20);B%;TAB(40);B:NEXT:GOTO 800

```

STR#**VAL**

```

9200 A#="STR# VAL":GOSUB 900:PRINT F#;TAB(25);"A#="STR#(X)":PRINT TAB(25);"A=VAL(A#)":GOSUB 600
9210 PRINT "STR# returns a string which is the ASCII representation of x"
9220 PRINT "VAL returns the fpt value of the number in the string":GOSUB 600:GOTO 800

```

TAB

```

9300 A#="TAB":GOSUB 900:PRINT F#;TAB(25);"PRINT TAB(30);...":GOSUB 600
9310 PRINT "Moves the cursor to the desired column":GOSUB 600
9320 FOR Y%=15 TO 3 STEP -1:A%=RND(40):CURSOR A%,Y%:PRINT CHR$(255);"TAB(";A%;" )":WAIT TIME 50:NEXT
:GOTO 800

```

VARPTR

```

9600 A#="VARPTR":GOSUB 900:PRINT F#;TAB(25);" A=VARPTR(B)":PRINT TAB(25);"A=VARPTR(B(3,4))":GOSUB 6
00
9610 PRINT "Variable A is set to the memory address of B":GOSUB 600:GOTO 800

```

MOD

```

9900 A#="MOD":GOSUB 900:PRINT F#;TAB(25);"A = X MOD B":GOSUB 600
9910 PRINT "A equals the remaining part of X/B":GOSUB 600
9920 PRINT :FOR X%=1 TO 10:A%=RND(100):B%=RND(6)+1:PRINT TAB(20);A%;" MOD";B%;" =";A% MOD B%;NEXT:G
OTO 800

```

EXPERIENCES WITH DAI VIDITEL

VIDITEL

At the Utrecht meeting of HCC I bought the DAI VIDITEL program as developed by Hans de Vries c.s. and requested for a Viditel subscription from PTT.

In the weeks that followed I had to wait for my Viditel modem to be delivered by PTT as a part of the Viditel subscription. But the DAI VIDITEL program could be used already. In its local editing mode viditel-like screen lay-outs can be composed. This offers an easy to handle alternative for the preparation of text frames to be copied onto colour slides !

The modem came within three weeks. In that period I also assembled the connection cable, so the Viditel computer could be called immediately.

It is a pleasure for me to report a smoothly operating DAI VIDITEL program. By the fact that the complete DAI keyboard is used the potentials of Viditel are much greater for DAI users than for people using a special Viditel tv-set. By the so called Vidibus function I could send a letter with my first impressions to the designers of DAI Viditel. Note, that this way of corresponding is cheaper than sending real mail !

Another facility made possible through DAI VIDITEL and the PTT modem is a connection with the Viewdata computer of NOVA Automation Consultants. This private viewdata system is accessible for HCC members (free of charge). They can both access it for retrieving stored information pages, but also for storing their own information pages onto it. Further, it offers program development and processing functions which are available in combination with the viewdata functions.

According to my 'software philosophy' programs should be seen as 'tools': simple products which perform a single well-defined task which results into output that can be processed further by another tool. Following this philosophy I would like to offer the following suggestions to the designers of DAI VIDITEL:

1. do NOT add any more bells and whistles to the program.
2. change the program into a module; i.e. make it possible to call the program as a module from within another module or basic program.

It is my strong opinion that by following these suggestions the designers of DAI VIDITEL will be able to provide the DAI users with a powerful tool that they can extend themselves according to a variety of wishes. This can lead to a stream of ideas and applications for which the DAI VIDITEL module is crucial. It frees the designers from the implementation and management of a never ending stream of 'improvements'. The latter in fact better can be programmed by other people, thus enabling the designers to devote themselves to more advanced problems.

Supplement to DAI VIDITEL documentation:

1. The english pound sign (ASCII 35) can be entered by SHIFT-TAB; the graphics symbol corresponding with this sign also can be entered by SHIFT-TAB.
2. The at-sign (ASCII 64) can be entered by SHIFT-0 (zero).
3. The right arrow and left arrow (ASCII 93 and 91) are entered by 1

Postbus 11044, 2301 EA Leiden, tel.071-411769, viditel #234041 1

and [respectively.

4. The release graphics sign (ASCII 95) is entered by CTRL-SHIFT-CHARDEL.

5. Other peculiarities are:

Sign: ASCII: How to enter:

1/2 92 TAB

US 96 CTRL-SPACE

1/4 123 CTRL-;

11 124 CTRL-<

3/4 125 CTRL==

Recommended literature:

J.J.M.Bokland. Viditel: techniek voor de abonneeapparatuur; The Hague, 1981.

P. van der Hijden

Benodigde Viditel-apparatuur

1. Een aangepast TV-toestel met afstandsbediening, bureauterminal of hobby-computer.
2. Een telefoontoestel.
3. Een modem (dat door de PTT wordt verstrekt zonder extra kosten).

Viditel-randapparatuur

De PTT verhuurt Viditel-randapparatuur, waaronder:

- o de Viditel-bureauterminal (VBT-003), bestaande uit een combinatie van kleurenmonitor en alfa-numeriek toetsenpaneel;
- o de Viditel-printer (VPE-403), voor het maken van afdrucken van Viditel-beelden op papier;
- o de Viditel-printertussenschakeleenheid (VPI-403), om in de handel zijnde matrix-printers aan te sluiten op de Viditel-abonnee-modem;
- o de Viditel-automatische nummerzender (VCG-901), voor o.m. het automatisch uitzenden van het toegangsnummer, code en eventuele privé-code;
- o de Viditel-invoerterminal (VIT-003).

Viditel in de toekomst!

Tot nu toe kunt u met Viditel al diverse kanten uit. U kunt informatie opvragen die voor u belangrijk is. Daarnaast is het mogelijk om via antwoordbeelden bestellingen te plaatsen bij bedrijven en berichten te versturen aan andere Viditel-abonnees.

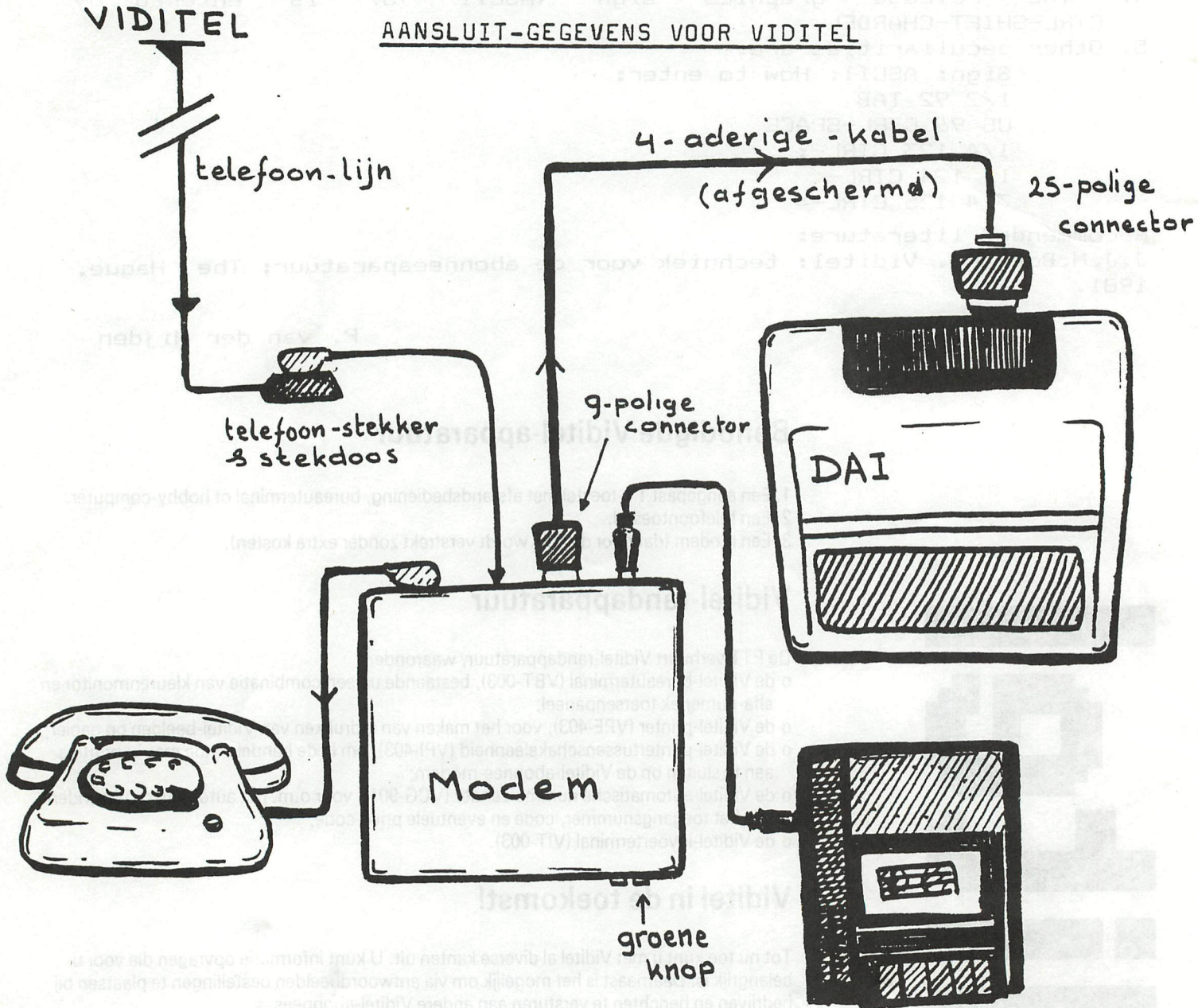
Begin 1982 worden de mogelijkheden met Viditel verder uitgebreid door de zgn. Gateway-functie. Een abonnee kan dan met deze functie via Viditel gekoppeld worden aan (externe) computers van bedrijven en instellingen, waardoor hij over steeds meer en betere informatie- en communicatiemogelijkheden kan beschikken, zoals: het boeken van reizen -met ontvangst van reserveringsbevestiging -, het overboeken van geldbedragen op rekeningen, het laten uitvoeren van berekeningen, het bestellen van producten enz. enz.

Wilt u meer weten over Viditel, vraag dan informatie bij:

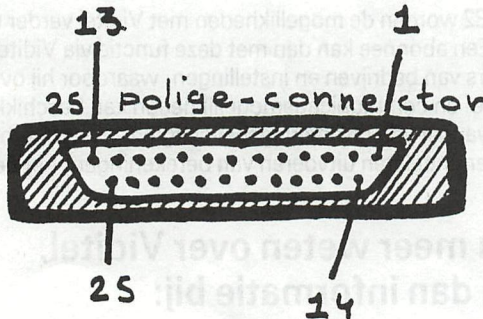
Centrale Directie der PTT
 Directoraat Commerciële Zaken Telecommunicatie
 Bureel Viditel
 Antwoordnummer 6000
 2500 VB 's-Gravenhage
 Een telefoontje kan natuurlijk ook:
 (070) 75 32 69 / 75 33 92 / 75 40 74

Wanneer men Viditel-abbonee wordt krijgt men automatisch het modem te huur à f10,- per maand, de 4-aderige kabel wordt NIET door de PTT geleverd.

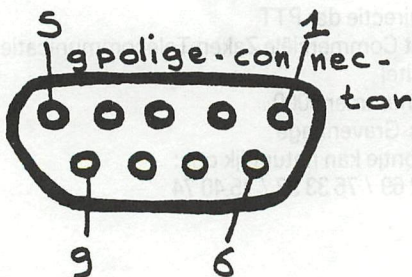
AANSLUIT-GEGEVENS VOOR VIDITEL



DAI:



MODEM:



cassette - recorder

AANSLUIT-GEGEVENS		
	MODEM	DAI
aarde	5	1
serie in	4	3
serie out	3	2
DTR	7	8

(DTR=Data Terminal Ready)

Amphenol:
17-90250-16 &
17-1372

Amphenol:
17-90090-16 &
17-1370

leverancier RODELCO

===== D A I - V I D I T E L =====

DAI-VIDITEL kent alle standaard VIDITEL-attributen.

Zoals bv.

: 8 kleuren.

: Grafische symbolen & 'gescheiden' grafische symbolen.

: Dubbele hoogte karakters.

: Flashing.

enz.

Er is geen enkele hardware-uitbreiding of aanpassing in de DAI nodig.

De beeld-opbouw geschied met behulp van een 16 kleuren grafische mode. (gelijk aan mode 5)

Het programma is geheel in machine-code geschreven en beslaat ongeveer 3½K geheugenruimte.

Het toetsen-bord is zodanig aangepast dat het mogelijk is de DAI te gebruiken voor het editen van viditel-pagina's.

Uiteraard kan men het toetsen-bord ook gebruiken voor het verzenden van 'electronische brieven' via de Viditel-VIDIBUS service, of het verzenden van 'Antwoord-pagina's aan Viditel informatie-leveranciers.

=====
Edit-mogelijkheden:

Break: full duplex, Break Break: half duplex enz.

Ctrl Q : cursor on
Ctrl T : cursor off } volledige cursorbesturing

Ctrl L : clear

Ctrl ↑ : home

Ctrl [: ESCAPE

ESCAPE-functies

ESC A : alfa-numeriek rood	;ESC Q : grafisch rood
ESC B : alfa-numeriek groen	;ESC R : grafisch groen
ESC C : alfa-numeriek geel	;ESC S : grafisch geel
ESC D : alfa-numeriek blauw	;ESC T : grafisch blauw
ESC E : alfa-numeriek magenta	;ESC U : grafisch magenta
ESC F : alfa-numeriek cyan	;ESC V : grafisch cyan
ESC G : alfa-numeriek wit	;ESC W : grafisch wit
ESC H : flash	;ESC Y : continue grafisch
ESC I : steady	;ESC Z : gescheiden grafisch
ESC J : End	;ESC L : normale hoogte
ESC K : Start	;ESC M : dubbele hoogte
ESC] : nieuwe achtergrond	;ESC ↑ : grafisch 'houdend'
ESC TAB: zwarte achtergrond	;ESC shift TAB: release

Voor off-line editen (het maken van beelden zonder dat viditel is aangesloten):

- 1 toets de 'BREAK'toets in.
- 2 druk op de 'CTRL'toets en toets tegelijk de 'Q' in. (cursor on).

Enkele belangrijke geheugen-adressen :

301, : Geheugen-grootte : #C0 bij 48K en #80 bij 32K
303, : ASCII-pagina : #54 bij 48K en #14 bij 32K
304 t/m # 30B, : Gebruikte kleuren voor achtereenvolgens: zwart, rood, groen, geel, blauw, magenta, cyaan en wit

=====

HANDLEIDING VOOR HET HELE GEZIN: HOE KRIJG IK VERBINDING MET VIDITEL?

- stap 1. Is de DAI met de telefoon verbonden? (via de modem)
Zo nee: Maak de verbinding door de 25 polige 'stekker' vanuit de modem in de 'RS232 bus' van de DAI te steken, deze bevindt zich aan de achterzijde.
- stap 2. Zoek de Viditel-cassette en stop deze in de cassette-recorder.
- stap 3. Zet de TV aan.
- stap 4. Zet de DAI aan, de schakelaar bevindt zich aan de rechter-achterzijde.
- stap 5. U krijgt nu een groen (grijs) beeld met in grote letters: DAI PERSONAL COMPUTER. : druk een willekeurige toets in.
- stap 6. Er is 'BASIC' op het scherm verschenen.
U geeft nu de computer z'n eerste instructie:
Typ UT in en druk op de 'RETURN'toets.
- stap 7. Er is nu 'UTILITY' op het scherm verschenen;
geef de tweede instructie:
Typ Z3 in en druk op de 'RETURN'toets.
- stap 8. Het moment is nu gekomen dat U het Viditel-programma vanaf de cassette in de DAI kunt gaan lezen:
1. Typ R in en druk op de 'RETURN'toets.
2. Start de cassette-recorder.
- stap 9. Als het programma ingelezen is verschijnt er weer een '>'.
Het DAI-VIDITEL programma kan nu in werking worden gesteld: Typ G400 en druk de 'RETURN'toets in.
- stap 10. Zo, het programma loopt, het beeld wordt zwart.
Bel nu naar VIDITEL:
020-318318 voor AMSTERDAM of
070-151515 voor DEN HAAG.
- stap 11. Als u de hoge piep-toon hoort, druk dan op de groene knop aan de voorzijde van het modem, en leg de hoorn weer op de haak.
- stap 12. VIDITEL presenteert zich.
Typ uw toegangsnr. in :
Typ uw code-nummer in :
En typ eventueel uw prive-code-nummer in :

U bent nu met VIDITEL verbonden. Veel genoegen!!

***VIDITEL is standard in Belgium, Holland, UK & Germany.

***latest version has extra features for printer and cassette/DCR

Krantel Studio 649084j Oc

HIS MASTERS VOICE
NU OOK IN BEELD.
THORN-EMI BRENGT FILMS OP VIDEO.
vervolg

JWT

9 voor keuzepagina

Krantel Studio 649084h Oc

HET HABELT
HET HABELT
GROTE KORRELS VENZ

vervolg
0 index jwt

JWT

Krantel Studio 649084c Oc

ONZE NAAM FLEEFT OOK
AAN 'T NIETIGSTE DRUKWERK

JWT

Krantel Studio 649084e Oc

KOPEN ZONDER KATER

VERVAARDIGD VOOR POSTBUS 51
CONSUMENTENINFORMATIE

JWT

Viditel 180139a Oc

en 1982 zal ook wel lullen...

DE PESTE WENSEN

JWT

Krantel Studio 64908a Oc

J. WALTER THOMPSON COMPANY

voor kennismaking 1 index jwt

JWT

Pag 000 SYSTEM 005 min

***** HOOFDINDEX *****

1. WAT IS ABC-DATAVISION??
2. ALGEMENE INDEX.
3. INFORMATIE ABC-Data B.V. (NIEUW)
4. PRODUCT INFORMATIE.
5. GEBRUIKSAANWIJZING.
6. LIJST VAN ONDERWERPEN.
7. GRAFISCHE VOORBEELDEN.
8. HOBBY AANBIEDINGEN.
9. EDITING, Alleen voor I.P.'s

Toets het nummer van Uw keuze.

VIEWDATA
3000

Thank you for calling.

Please call again.

Viditel 180130a Oc

CEES P. RUIGROK'S UILEN WENSEN U
NE 800

EEN WIJS
EN VREEDZAAM
KERSTFEEST

EN EEN
KRITIEKLOOS
1982

JWT

Krantel Studio 649084b Oc

UW
GENERATIE
WERKT
BIJ RANDSTAD

vervolg
0 index jwt

JWT

Krantel Studio 649084m Oc

FORD. MEER WAARD OP DE WEG.

vervolg
0 index jwt

JWT

Philips'
Natuurkundig
Laboratorium

Viewdata

Alleen voor besloten gebruikersgroepen
Voor inlichtingen: 040-745072

Mag ik uw gebruikersnummer?

Vries Data 301065a Oc

DAI-NAMIC

DAI PERSONAL COMPUTER CLUB

CONTACT-ADRES: W.F. HERMANS
HEIDE 90
WESTMEERBEEK
BELOIE

toets....

Vries Data 301065b Oc

DAI-NAMIC NIET
OFFICELEEL
TOETS

1. DAINAMIC COLOFON
2. UIT DE DAINAMIC PERIODE
3. VOOR DAINAMIC PROGRAMMA'S
4. AGENDA 2. ROUTEBESCHRIJVING

Vries Data 30106a 27c

viditel
hcc telecommunicatie

VIDITEL OP DE DAI PC

Machine code programma start op 4400

Volledige Handleiding komt in NOVVA
en in DAI-NAMIC

Vries Data 3010651a Oc

DAI-NAMIC
COLOFON

1. Dainamic - periode
2. Lid worden van Dainamic
3. Tel. contact 3. Correspondentie

Viditel 180105a Oc

BARIC

Een...
...erf...
...kerst!

En natuurlijk...

Vries Data 301060a Oc

***** HOOFD-MENU *****

1. TELE-SOFTWARE (nog te ontwikkelen)
2. AGENDA
3. ACTUALITEITEN
4. GRAFISCHE POSTINGEN
5. DAINAMIC
 - RESERVE
 - RESERVE
6. NIEUWJES ALLERLEI
9. INFO VOOR DAI-GEbruikers

VIDITEL NU OOK OP DAI

met dank aan JÜRIG DE VRIES.
originele screen copies op
RX-80, gereduceerd formaat.
Laat uw abonneenummer kennen
als je er ook bij bent ...

Vries Data 3010691a Oc

DAI USER'S OP VIEWDATA/3000 Pag.1

5506 JÜRIG DE VRIES (IL):30106
BUNCHESTR. 26
2286 GL RIJSWIJK (Z-H)
TEL. 070-923607
VID. 400003419

5525 R. DEN BREEMS
HOONSTADDEEF 27
2067 RT ROTTERDAM
TEL. 010-551234
VID. 400004252

5539 SJEFF VAN BERLO
G. BEZELLEMAN 5
2624 KT DELFT
TEL. 015-565440
VID. 222222222

Zend Antwoordpagina: toets 9

Viditel 180107a Oc

DEN-SPORT WENST U EEN 'SPEELS'

BUON NATAL
BUON FEST
PRETTIGE KE
ERST'DAGEN F
ROHE WEHN
NACHTEN G
T NYTT A
OD JUL
KJIG N
JAAR
OHRI
ONNE LE
BUON NATALE
GELUKKIG NI
ELUJAAR BUO

Pag 090 SYSTEM 006 min

HARTELIJK DANK VOOR UW INTERESSE
EN GRAAG TOT EEN VOLGENDE KEER

vervolg
0 index jwt

JWT

Krantel Studio 649084d Oc

FEMINA.
'N ZACHTHEID
WAAROP JE ALTIJD
MAG VERTROUWEN.

vervolg
0 index jwt

JWT

Vries Data 3010691b Oc

DAI USER'S OP VIEWDATA/3000 Pag.2

5539 THEO VERBERKT (IL):3015539
v. BUERENSTRAAT 13
GUD HEUSDEN
TEL. 04162-2667
VID. 222222222

5548 PIETER v/d HEIDEN
VOORDORP 53
2352 RT LEIDERDORP
TEL. 071-411769
VID. 400004581

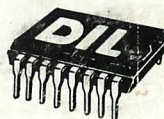
5550

Zend Antwoordpagina: toets 9

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

Meensesteenweg 49
8800 ROESELARE
Tel. 051/21 30 89

Beenhouwersstraat 87
8000 BRUGGE
050/33 08 01



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

Guibernau Electronica, s.a.

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

IDS 2000

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

LEGOTRONICS

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

MEMOCOM

Mini-digitale cassetterecorder

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

MICRO SELECT

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

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



Bennenbergweg 1
3221 NIEUWRODE (bij AARSCHOT)
Tel. 016/56 87 70
DAI - Epson Printers - Memocom digitale cassette
recorder - Barco kleurenmonitor - Software -
Microlectuur - Service

Publishing House J. VAN IN

att. : L. CAMPS

Educational Software
primary - secondary schools

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

TEVETRONIC

Avenue Milcamps 57
1040 BRUSSEL
Tel. 02/738 61 24