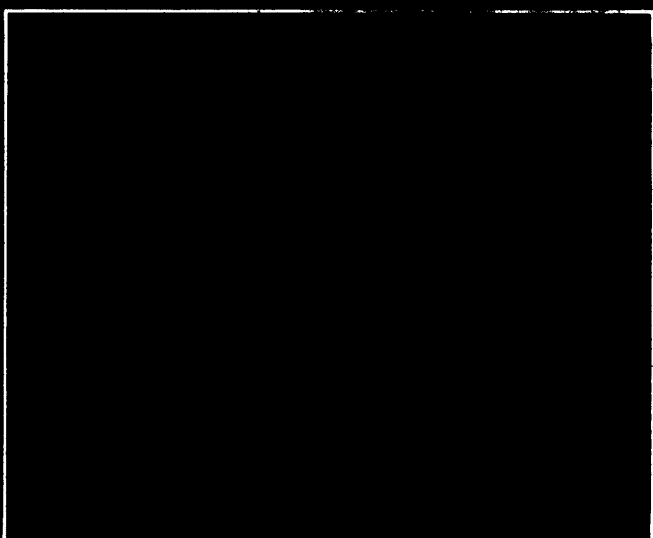
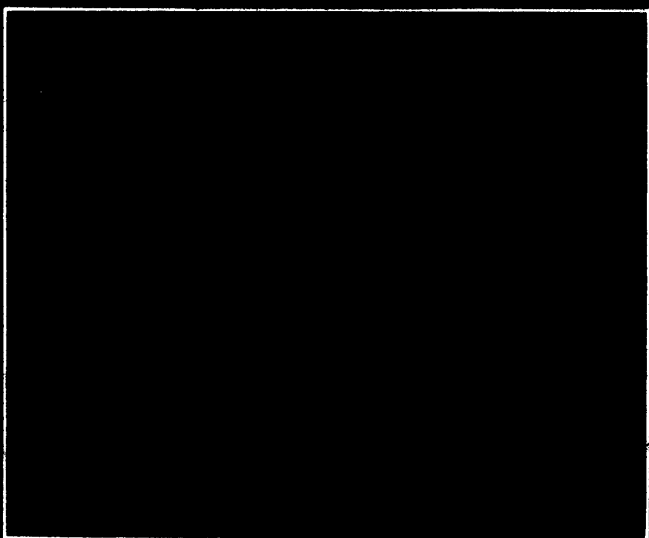
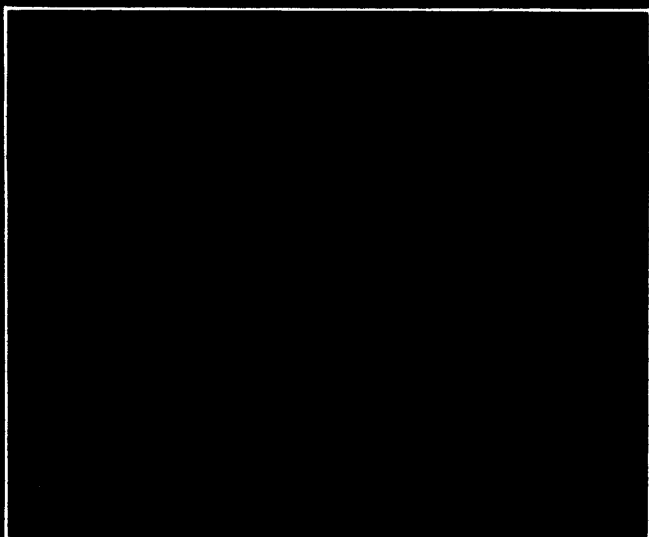


tweemaandelijks tijdschrift

maart - juni 1982



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.

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

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

Lidgeden

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

belangrijke ASCII-waarden in DAIPc

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

ASCII - HEX - ASCII CONVERSION TABLE

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

Beste leden,

Door de vele extra clubactiviteiten van de laatste maanden waren we achter op ons publicatieschema. Met dit dubbelnummer doen we een inhaalmaneuver, zodat we voortaan ons blad weer op tijd kunnen versturen. We hopen dat kwaliteit en volume van nummer 10 het wachten ruimschoots compenseren.

Hierbij geven we een overzicht van de nieuwe publicaties en producties voor de komende maanden:

‡ hardware schema's DA1pc : nu verkrijgbaar
 ‡ tape 80-81 (57 programma's van de vorige jaargangen) : nu verkrijgbaar
 ‡ de 23 programma's uit dit nummer op tape : nu verkrijgbaar
 ‡ "the best of DAInamic" in boekvorm : augustus 82
 ‡ "DAI firmware" door J.Boerrigter : december 82

"softwarepiraat slaat toe via de ether...": dit zou een leuke krantentitel kunnen zijn. Op originele manier hadden we de laatste weken last van piraterij: onze copieerapparaten staan op een 500m van de zendmast van de plaatselijke vrije radio "TELSTAR". Deze jongens zenden er erg enthousiast op los en komen met hun programma's ongevraagd op de aanloopstrook van onze software-bandjes... Gelukkig kan het computersignaal de populaire plaatjes en praatjes volledig onderdrukken.

Onze dank gaat andermaal naar alle medewerkers en auteurs die deze uitgave mogelijk maakten.

We willen iets gaan doen aan het taalprobleem binnen onze vereniging: mensen die artikels en programma's zouden kunnen vertalen naar het engels, duits of frans worden verzocht contact te nemen met onze vertaal-coördinator:

Filip Peeters
 Vlinderstraat 7
 2440 GEEL België tel 014/583018

Uw werk compenseren we graag met gratis software, papier en telefoonkosten worden vergoed.

Onze bijeenkomst op 10 april was andermaal een succes, vooral de lezingen vielen erg in de smaak. Ergens in oktober gaan we een volgende bijeenkomst plannen, we zullen de voordrachten nog beter voorbereiden en aankondigen zodat U deze dag optimaal zal kunnen benutten.

Bij de samenstelling van dit nummer hebben we getracht de beginners niet te vergeten: we vermoeden dat de demo-programma's van Frank het handboek mooi aanvullen. Volgende keer gaan we de EDIT-buffer bestuderen en leggen we de loupe op de BASIC-token structuur.

we hopen dat nummer 10 uw creativiteit stimuleert, uw bijdragen zijn steeds welkom..

 dear members,

Because we were running out of time, we offer you this "double issue", we hope that the many programs and articles can compensate for the waiting.

This is the time-schedule of our new publications and productions:

‡ hardware schematics DA1pc : now available
 ‡ tape 80-81 (57 programs) : now available
 ‡ the 23 programs of this issue: now available
 ‡ "the best of DAInamic" (book) : august 82
 ‡ "DAI firmware" by J.Boerrigter: december 82

We are looking for co-editors/translators : if you can assist in translating articles to english, french or german, please contact our language-coordinator:

Filip Peeters
 Vlinderstraat 7
 2440 GEEL Belgium tel 014/583018

We will compensate your efforts with free software, we will pay your mailing and telephone costs for this job.

Our next meeting will be in october, more information in future issues.

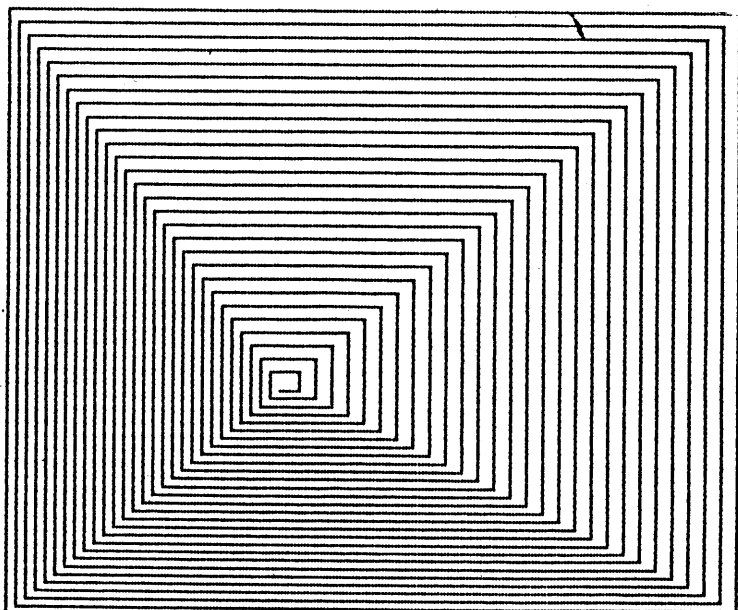
we hope you will enjoy this king size issue, we appreciate all contributions...

Wilfried Hermans

DAI

71	remark	redactiepraatje
72	bladwijzer	
73	bladwijzer	
74	inzenden van programma's	F.Druijff
75	diDAIsoft	B.van Rompay
76	DCE-microline	J.Pekkeriet
77	DAI video-hardware	A.Doornenbal
78	"	
79	"	
80	"	
81	" + drawing with paddles	R.Rens
82	large size characters + beer	F.Druijff
83	alternative printer routine	J.Boerrigter
84	"	
85	spot on text-tumbling lines-patchwork	F.Druijff
86	flug simulator	A.Meystre
87	"	
88	info - info - info	
89	short ml routines	R.Sip
90	4-takt motor	M.Dierckx
91	combination of ml & BASIC (V1.0)	J.Boerrigter
92	"	
93	DAI interface module	F.de Raedt
94	"	
95	"	
96	"	
97	mathpack & rom interface	P.Jongen & F.de Raedt
98	"	
99	"	
100	"	
101	" + tape 1980-1981	
102	the story of Anna List	B.Hunt
103	testbeeld	A.Doornenbal
104	"	
105	ASCII-BINARY-ASCII	P.Jongen
106	"	
107	" +electric energy	Smit
108	Oscilloscope	R.Sip
109	"	
110	" + Mandala	
111	16 color characters in mode 0 + picture stabilisation	
112	primzahlen in graphic	A.Meystre
113	benchmarks	Coughlan
114	DAI op de piste	Vollinga-Esselink
115	birthday song	B.Hunt
116	lopende & staande golven	J.Roelants
117	"	
118	DCR tape control+ phone ringing	
119	letter from Mr Werbeck	
120	getallen INT-FP	F.Druijff
121	Rubic cube + tips	F.Druijff
122	snoopy	C.de Bont
123	change 16 color mode in 4 color mode	W.de Winter

124	more about talk & music	DAI
125	"	
126	"	
127	"	
128	"	
129	" + floating point numbers	B. van Rompay
130	"	
131	"	
132	"	
133	"	
134	"	
135	"	
136	bingo	T.Groeneveld
137	"	
138	grue + chrono	J.Moens
139	240 x 528 resolution	R.Sip
140	PASCAL a description of DTP	R.Theeuws
141	"	
142	"	
143	"	
144	PASCAL prime numbers-towers of hanoi	Nijland-Van Eck
145	" + one text line in mode 5/6	W.Hermans
146	PASCAL decode + dragon curve	M.Dierckx
147	PASCAL library routines	
148	DAI-RS232 , extra features on MX-82	H.Moeys
149	machinetaal in een REM-statement	T.Berckx
150	"	
151	"	
152	"	
153	"	
154	Interactieve videotex	RTT Belgium
155	"	
156	"	
157	"	
158	"	
159	faculteit	M.Dierckx
160	cassette list + DNA modifier	Arts + F.de Raedt
161	weersatelliet foto's op uw TV-scherm	H.Bakker
162	"	
163	"	
164	"	
165	"	
166	"	



* * * * * I N Z E N D E N V A N P R O G R A M M A ' S * * * * *

Veel dainamicers weten niet, is mij gebleken, dat programma's opgestuurd kunnen worden. De programma's worden dan beoordeelt en U krijgt behalve dit commentaar ook nog eens een aantal programma's uit de bibliotheek. Dit laatste geldt alleen als de programma's een ruil rechtvaardigen en de opgezonden programma's -indien zij daarvoor in aanmerking komen- opgenomen mogen worden in de bibliotheek van Dainamic.

De eisen die wij aan programma's stellen om voor ruil in aanmerking te komen zijn echter niet hoog; ook de beginner moet de kans hebben zijn programma's in te kunnen sturen. U kunt trouwens zelf uw voorkeur voor bepaalde collecties opgeven zodat wij U zo goed mogelijk kunnen helpen.

Mensen die vaak insturen zullen begrip hebben dat zij soms niets krijgen en een volgend keer veel meer dan gerechtvaardigd door de inzending van dat moment. Een tweede beloning (blijk van waardering) krijgt U als uw programma opgenomen wordt in een van de collecties - U ontvangt deze collectie dan geheel gratis en automatisch (klaag dus als wij het vergeten).

Nu over het inzenden zelf.

Liefst ontvangen wij uw programma's op DCR-cassette omdat wij dan nooit laadproblemen hebben. Iets minder lief maar nog steeds gewaardeerd op disk of op gewone cassette. De hogere kosten van minicassette/diskette behoeven geen bezwaar te zijn bij insturen daar U terugkrijgt wat U instuurt. Vooral bij gewone cassettes zijn er nog regelmatig laadproblemen, de snelheid waarmee U antwoord krijgt kan hier vaak mee samenhangen.

U kunt deze problemen enigzins verlichten door uw band te beginnen met een twintigtal tests waar wij dan op kunnen afregelen. U doet dit als volgt:

```
NEWreturn
```

```
DIM A(0)return
```

```
FOR I=1 TO 20:SAVEA A "TEST":NEXTreturn
```

en vanzelfsprekend de recorder op opname. Vervolgens kunt U ons erg helpen met een begeleidend schrijven waarin vermeld staat wat wij op de band kunnen verwachten het spaart ons het afspelen van een band waar niets meer op staat. Doe dit begeleidend schrijven inderdaad letterlijk op papier, als wij laadproblemen hebben kunnen wij aanwijzingen op band natuurlijk ook niet lezen. En dan het meest belangrijke : de programma's zelf.

Ik zou bijna elke inzender de raad willen geven om eerst iemand die niets van computers weet eens met het programma te laten werken/spelen.

Ik heb tientallen programma's die na een RUN mij lange tijd lieten wachten en dan bij controle in een eeuwige loop bleken te zitten. Soms in afwachting van een bepaalde code, die mij echter op dat moment nog onbekend was. Soms bleken overigens correcte programma's lange tijd bezig met het inlezen van data. Als daar echter niet voor wordt gewaarschuwd - bij voorkeur met een aankondiging en een aflopend tellertje om aan te geven hoever het staat - irriteert zo'n programma nog voor aanvang al mateloos.

Ook bezit ik vele programma's die na RUN mededelingen geven als OUT OF MEMORY, COLOUR NOT AVAILABLE, NUMBER OUT OF RANGE, OUT OF STRING SPACE, UNDEFINED ARRAY, OUT OF DATA en bijna alle andere mogelijke foutmeldingen waar iets verder in het programma de OFF SCREEN topscoorder is.

Controleert U uw programma aub goed - iemand anders kan dit vaak beter - voordat U het inzendt. Een programma met een geconstateerde fout zal niet opgenomen worden in een collectie en minder gewaardeerd worden.

U kunt vele fouten vermijden door meer structuur in het programma te brengen, te denken aan het feit dat niet iedereen een zelfde beeld heeft dus dat de door U gekozen kleurencombinatie bij een ander vrijwel onleesbaar is. Bij mij is 4/5 of 4/8 of 4/12 onbruikbaar maar 6/7 of 7/8 heel duidelijk. Er zijn vanzelfsprekend nog vele aspecten onbesproken gebleven maar ik wilde dit artikel besluiten met de oproep om toch ook eens in te zenden. Bedenk dat wij allemaal eens als beginneling begonnen zijn en dat niemand ineens perfecte programma's maakte. Vooral beginners hebben vaak nog een originaliteit waar vele oud-gedienden niet aan kunnen tippen.

diDAIsoft

De DAI pc doet zijn intrede in het onderwijs. DAInamic start daarom met een didactische softwaregroep diDAIsoft.

Doelstelling :-didactische software ontwikkelen voor alle vakken uit het secundair onderwijs
-kant en klare handleidingen bij deze programma's samenstellen
-een vaste onderwijsrubriek in de nieuwsbrief verzorgen

Werkwijze : op tweemaandelijksse bijeenkomsten worden concrete afspraken gemaakt in verband met de programma's die gedurende de volgende twee maanden zullen ontwikkeld worden; individuele engagementen voor één van de voorgestelde projecten; op een volgende bijeenkomst worden dan de ontwikkelingen bekeken, besproken en (eventueel) aangepast; op deze wijze komen per periode een aantal zinvolle programma's tot stand, waarvoor de ganse softwaregroep garant staat; publicatie van de software onder het label diDAIsoft.

Wat bieden wij: -toegang tot de reeds beschikbare bibliotheekroutines;
-reiskostenvergoeding en lunch op de dagen van de bijeenkomsten;

Plaats van de bijeenkomsten: Bruno Van Rompaey
Bovenbosstraat 4
3044 HAASRODE
016 - 461085

DATUM VAN DE EERSTE BIJEENKOMST:

dinsdag 6 juli 1982 vanaf 10u tot 16u

Om schikkingen te treffen voor de lunch graag vooraf telefonisch of schriftelijk verwittigen voor 1 juli: contactadres Bruno Van Rompaey.

Geïnteresseerde medewerkers die op deze "stichtingsvergadering" niet aanwezig kunnen zijn, toch maar contact op nemen. We melden hen dan persoonlijk de datum van de volgende werksessie.

Met uw hulp wordt diDAIsoft een garantie voor kwaliteitssoftware.

Bruno

DCE-microline

```

67      *** =====
68      *** *                CONNECTIONS
69      *** *  D.C.E. BUS ----- OKI microline 80
70      *** =====
71      *** Port Bit Pin -----Func.-Pin
72      *** *   GND   4 -----          0V   26
73      *** P0 bit 0  16 -----      data b1   2
74      *** P0      1  14 -----          b2   3
75      *** P0      2  12 -----          b3   4
76      *** P0      3  10 -----          b4   5
77      *** P0      4   9 -----          b5   6
78      *** P0      5  11 -----          b6   7
79      *** P0      6  13 -----          b7   8
80      *** P0      7  15 -----          b8   9
82      *** P2      7  17 ----- <      Busy  11
83      *** P2      1  27 ----- >      Strobe  1
84      *** P2      6  18 ----- <      Acknowledge 10 (not used)
85      *** P2      5  19 ----- <      Paper out/low 12 (not used)
86      *** P2      4  20 ----- <      reserve      (not used)
90      ***
100     *** ::*** M/L ROUTINE FOR PRINTER ON DCE BUS ***::
110     *** -----
111     *** Go in UTILITY, type in next program, save on tape.
120     *** -----
130     *** 029B  20 03          :HEAP
140     *** 029D  00 01          :HSIZE
150     *** 029F  20 04          :TXTBGN
160     *** 02A1  21 04          :TXTUSE
170     *** 02A3  22 04          :STBUSE
180     *** 02DD  C3 F0 02      :DOUTC
190     *** *****
200     *** 02F0  E5              PUSH H
210     *** 02F1  D5              PUSH D
220     *** 02F2  C5              PUSH B
230     *** 02F3  F5              PUSH PSW
240     *** 02F4  57              MOV  D, A
250     *** 02F5  21 03 FE      LXI  H, dble
260     *** 02F8  36 8A          MVI  M, byte
270     *** 02FA  2B              DCX  H
280     *** 02FB  06 20          MVI  B, byte
290     *** 02FD  7E              MOV  A, M
300     *** 02FE  E6 80          ANI  byte
310     *** 0300  C2 FD 02      JNZ  addr
320     *** 0303  7A              MOV  A, D
330     *** 0304  32 00 FE      STA  addr
340     *** 0307  36 FD          MVI  M, byte
350     *** 0309  05              DCR  B
360     *** 030A  C2 09 03      JNZ  addr
370     *** 030D  36 FF          MVI  M, byte
380     *** 030F  F1              POP  PSW
390     *** 0310  C1              POP  B
400     *** 0311  D1              POP  D
410     *** 0312  E1              POP  H
420     *** 0313  C9              RET
430     ***
440     ***
450     REM : USE POKE#131, 3 TO ACTIVATE PRINTER
460     REM : USE POKE#131, 1 TO DEACTIVATE PRINTER
470     REM : Aut. J. A. P.

```


DAI video-hardware

INHOUD OP G A V E

(1)

- =====
- 1 Wijziging DAI computer van interliniering naar niet interliniering.
 - 2 Wijziging DAI pal kleuren kaart t.a.v. video- en kleur-bandbreedte.
 - 3 Afregeling TV lijn frequentie. (15625 Hz)
 - 4 Afregeling- en wijzigingen t.a.v. kleurendraaggolf
 - 5 Afregeling geluid draaggolf
 - 6 programma 'Test beeld'
 - 7 Cassette 'FBT'
 - 8 Ruilen cassette 'Viditel'
 - 9 Adres wijziging ondergetekende
 - 10 Algemeen, slot

- 1 Wijziging DAI computer van interliniering naar niet interliniering.
- =====

Een TV beeld wordt opgebouwd met 625 lijnen. Deze lijnen worden niet in 1 keer op Uw scherm geschreven, maar in 2 keer.

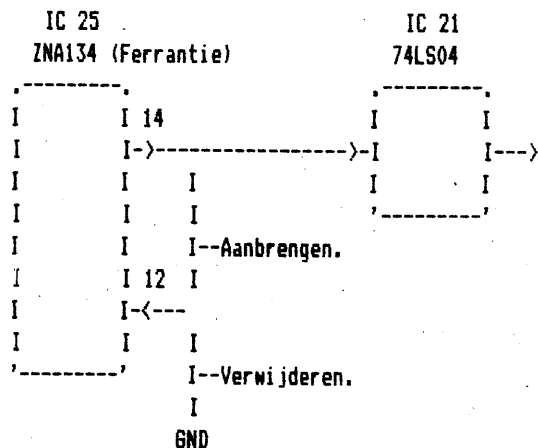
Met andere woorden: eerst wordt op Uw scherm de oneven lijnen geschreven, vervolgens de even lijn nummers. Van de 625 lijnen zijn er ongeveer 550 zichtbaar. Uw DAI computer doet het op dezelfde wijze, echter de informatie van de oneven- en even lijnummers zijn hierbij gelijk.

Stel: de nu volgende schrijfslag zijn de oneven lijn nummers. De de hierna volgende schrijfslag bevat hetzelfde beeld informatie met dien verstande dat het iets lager op Uw TV scherm geschreven wordt. (Afstand tussen beide schrijfslagen is ongeveer van 0.5mm tot 1.0 mm.) Een schrijfslag duurt 20ms. (50Hz)

Samenvattend: de door Uw DAI computer gegenereerde beeld staat iets te dansen in vertikale zin, met een periode tijd van $2 * 20ms = 40ms$

Dit is de oorzaak dat Uw DAI computer een enigszins onrustig beeld geeft.

Wijzigings voorstel interliniering - niet interliniering (3)



Aangezien poot 12 van IC 25 onder deze chip aan de komponent zijde van de print met de 'GND' is verbonden, is het moeilijk om deze verbinding te verbreken. Om deze poot toch vrij te maken kan men het beste vlak boven de print (komponent zijde) deze doorknippen, om vervolgens enigszins omhoog te buigen en met een draadje te kunnen doorverbinden met poot 14.

Voordeel van de hiervoor aangegeven wijziging zal een ieder wel duidelijk zijn. (4)

Deze wijziging heeft ook een nadeel: Indien men gebruik maakt van de 20ms interrupt (vector 7) b.v. een real time clock, zal deze klok iets te snel gaan lopen.

$f(\text{kristal (zna 134)}) = 2,562,500 \text{ Hz}$
 $f(\text{lijn}) = f(\text{kristal}) / 164 = 15625 \text{ Hz}$
 $f(\text{raster}) = f(\text{lijn}) / 312.0 \text{ (was } 312.5) = (50 + 25/312) \text{ Hz}$

De '20ms' interrupt komt nu iedere 19.968 ms. Verder omgerekend komt het op neer dat de real time clock per 24 uur 138 sec te snel zal lopen, voorop gesteld dat de kristal frequentie gelijk is aan de opgegeven waarde.

zie hiervoor verder hoofdstuk 3 van deze brief.

DAI video-hardware

2 Wijziging DAI pal kleuren kaart t.a.v video- en kleur-bandbreedte (5)

Het viel mij op dat het testbeeld, uitgezonden door de nederlandse televisie zenders een hogere resolutie had dan de door mijn DAI computer gegenereerde plaatjes. Oorzaak hiervan is dat de video-bandbreedte van de DAI-pal color-chart maar slechts 3MHZ is. (-6dB) Idem voor de kleur-bandbreedte namelijk 0.75MHZ. Waarom heeft de firma 'DAI' deze bandbreedtes zo klein gekozen? Misschien om interferenties te voorkomen? De bandbreedtes van een televisie toestel zijn resp. 5 MHZ en 1.5 MHZ. Na de op de volgende bladzijden beschreven wijzigingen krijgt de TV interface bandbreedtes van resp 7MHZ en 2.5 MHZ. Dit geeft een enorme verbetering t.a.v.het oplossend vermogen. Na de wijzigingen heb ik in het geheel geen last van interferenties en dergelijke. Bij enkele andere merken TV's echter wel. De TV die ik gebruikt als monitor is een philips met 'KT3' chassis, 43 cm 90 graden beeldbuis. De toepassing van een 90 graden beeldbuis maakt deze TV zeer geschikt als monitor voor uw DAI computer. Omdat het oplossend vermogen van de beeldbuis groter is (focus) dan vanuitgaande van de 5MHZ bandbreedte.

Wijzigings-voorstel voor DAI-pal color chart. (helderheid) (6)

```

74LS164 +5v          74LS125
-----, [J470E 8K2 ,-----, 6 15K
D0---I-ID I---^-----[ ]--I I I-----[ ]--I
--I> I10(Wordt 6) I I I-----[ ]--I
I '-----' I '-----' 3 1K0 I
I I +12v I
I 74LS164 +5v I [J1K0 I-----I
I ,-----, [J470E 4K7 I I 560E I
D1--I-ID I---^-----[ ]--I [ ]--[ ]--I +12v
I-I> I10(Wordt 6) I GND I2N3704 I
I '-----' I 100uH 1K5 I I---I LM1889
I I-----[ ]-I-I---I 12,-----,
I 74LS164 +5v I / I I->I---I I
I ,-----, [J470E 2K2 I ^ 22pF=== I I I
D2--I-ID I---^-----[ ]--I ^ ^ GND 1K0[ ] '-----'
I-I> I10(Wordt 6) I ^ (Verwijderen) -5v
I '-----' I ^
I I (Spoel verwijderen)
I 74LS164 +5v I (en doorverbinden )
I ,-----, [J470E 1K0 I
D3--I-ID I---^-----[ ]--I
CLOCKI-I> I10(wordt 6) ===100pF(Verwijderen)
~20MHZ '-----' GND
    
```

Wijzigings-voorstel DAI-pal color chart. (Kleur) (7)

```

74LS374
-----, 1K8
I I---[ ]--I 470uH(Wordt 100uH)
I I 1K0 I---I-----I-----,
I I---[ ]--I I I I
I I 470E I ===56pF(Wordt 150pF) [J2K2 I
I I---[ ]--I GND GND I LM1889
'-----' I I ,-----,
I +5v ,---[ ]--I---I I
74LS125 I [J270E I 2K2 I I
,-----, 470E I I-----I-----I I
I I---[ ]--' [J150E ===22uF I 2K2 I I
I I---[ ]--, GND GND '---[ ]--I---I I
'-----' 470E I I '-----'
I I
74LS374 I---I-----I-----,
,-----, 1K8 I I I 470uH(Wordt 100uH)
I I---[ ]--I ===56pF(Wordt 150pF) [J2K2
I I 1K0 I GND I
I I---[ ]--I [J500E(Instel)
I I 470E I GND
I I---[ ]--I
'-----'
    
```

3 Afregeling TV lijn frequeentie (15625 Hz) (8)

De lijn frequeentie, welke door mijn DAI computer wordt gegenereerd was ongeveer 5Hz te laag. Dit is echter voor normaal gebruik (Als computer) in het geheel niet van belang, tenzij men er veel waarde aanhecht aan de naukeurigheid van de 20ms interrupt. (V7) Heeft men de wijzigingen doorgevoerd die in hoofdstuk 1 zijn beschreven, is de fout echter veel groter. Daar ik een radio/televisie service-amateur bent hecht ik een grote waarde aan de naukeurigheid van de TV lijn frequeentie t.b.v. afregeling en controle van de kleuren demodulators. Om de lijn frequeentie te kunnen afregelen in Uw DAI computer moet er een kleine wijziging worden uitgevoerd, namelijk: een condensator moet worden vervangen door een instelbare type. (Zie volgende bladzijde.) Nogmaals: een ieder die de DAI computer niet als testbeeld generator gebruikt, heeft het in het geheel geen zin om de in dit hoofdstuk beschreven wijziging en afregeling uit te voeren.

DAI video-hardware

4 Afregeling- en wijzigingen t.a.v. kleuren draaggolf. (10)

=====
 Het gebruik van de DAI computer als testbeeld generator heeft mij snel geleerd dat de frequentie van de kleuren draaggolf te hoog is, zodanig dat enkele TV toestellen het niet kan 'pakken' zodat er niet op kleur weergave wordt overgeschakeld. Dit is echter al eerder op gemerkt, zie hiervoor een DAInamic nieuwsbrief.

Om de genoemde frequentie lager en afgeregeld te krijgen moet er instelbare condensator van 22pF parralel aan de kristal geplaatst te worden. (Kristal, welke op het modulator plaat bevindt)

Deze frequentie moet heel nauwkeurig afgeregeldt worden en omdat de door mijn gebruikte frequentie teller niet nauwkeurig genoeg is het ik een ander methode gebruikt namelijk om de frequentie te vergelijken met de van een TV zender door middel van lisajoux figuren op een scoop. De ene kanaal wordt op Uw DAI computer aangesloten op de ingang van de HF modulator (Kleine blikken doosje, welke op de modulator plaat bevindt, pin 4.) terwijl de ander op de uitgang van de kleuren referentie generator van Uw TV toestel wordt aangesloten.

Typ op Uw DAI computer: '100 COLORG 1 0 0 0:MODE5:GOTO 100' vervolgens: 'RUN'.

Wijziging en afregeling TV lijn frequentie: (9)

```

      ZNA 134
      8.-----10
      .-----I  I----/2-----I---->-1,281,250.0 Hz
      I      I  I  -----^ I5
      C===22pF I  I  ^--/82-----I---->-15,625 Hz
      I      I  I      I
      --- 10scI      I
      []  I  I      I
      ---  I  I      I
      I   9I  I      I
      '-----I  I      I
      '-----'
  
```

Condensator 'C' wordt een instelbare type van 50pF.
 Hiermede regelt men de frequentie op pin 10 van IC25
 (ZNA 134) af op 1,281,250.0 Hz met een frequentie teller.

(11)

Stemt Uw TV toestel af op een TV zender en controleer of het testbeeld nog in kleur wordt weergegeven. Laat nu de computer en de TV een half uur opwarmen. Vervolgens regelt U met de erbij geplaatste instel condasator zodanig af dat het lisajoux figuur op de scoop stilstaat.

5 Afregeling geluid draaggolf (12)

=====
 De frequentie van de geluid draaggolf is eveneens van belang als de DAI computer wordt gebruikt als test beeld generator.

Om deze af te regelen heeft men een scherp geslepen lucifer stokje nodig.

Dit laat zich makkelijk afregelen. Men sluit een frequentie teller aan op pin 1 van de HF modulator. In de deksel van deze modulator bevindt een tweetal gaatjes, men neme de dichtst bijzijnde bij de aansluit zijde en regel de frequentie af op 5,500 KHz.

Hierbij dient 'SOUND OFF' te zijn.

Bij mijn DAI computer was naregeling niet nodig.

DAI video-hardware

6 Programma 'Testbeeld'

(13)

Zoals U al weet ben ik een service amateur, dus had ik een behoefte aan een eigen testbeeld generator. Dit is ook de hoofdreden geweest om mijn DAI computer op diverse punten te wijzigen. (Zie alle voorgaande hoofdstukken.)

Als voorbeeld voor het te genereren plaatje heeft het test beeld plaatje van de nederlandse zenders bestaan. Probleem was echter dat de DAI basic V1.1 in de grafische bewerkingen niet de onzichtbare gedeelten support, hetgeen voor een testbeeld van uiters belang is.

Dat is de reden dat in het programma 'testbeeld' (verder op deze cassette.) erg veel naar het scherm gePOKEt wordt.

De cirkel is wat kleiner uitgevallen omdat ik dit keer wel gebruik wilde maken van de basic support.

Dit programma is niet voorzien van allerlei instelmogelijkheden, gebleken is in de praktijk dat ik toch iedere keer het zelfde plaatje wilde hebben

Een korte beschrijving van het programma:

Regels 100-999 is het hoofd programma welke kort is.

Regels 1000-1060 is de initialisatie van het beeld scherm geheugen, hierbij wordt het scherm enigszins omhoog

geschoven en van onder wat toegevoegd.

Regels 2000-2804 zorgen er voor dat er de witten blokken rondom het plaatje geplaatst worden. (De meesten zijn niet of gedeeltelijk zichtbaar.)

Regels 3000-3070 deelt het beeld op in kleine vierkanten blokjes.

Regels 4000-4050 tekent een eenlijns cirkel welk in het hoofd programma niet gebruikt wordt. Wenst men er wel gebruik van te maken is het aan te roepen met 'GOSUB 4000', in het houdt in dat in regel 110 'GOSUB 5000' moet veranderen in 'GOSUB 4000'.

Regels 5000-5960 zorgt er voor dat er een 'gevulde cirkel' in het midden van het testbeeld wordt opgetekend.

Deze cirkel is als volgt ingedeelt: van boven naar beneden

- Rood/geel balk ter controle van de vertraging lijn.
- Zwart reflectie balk ter controle van reflecties en uitslingeringen van midden frequent gedeelte en dergelijke.
- Grijs gradatie balk ter controle van de lineairtiet van de video eindversterker(s).
- Kleuren balken ter controle van de kleur weergave.
- Het midden van de cirkel is voorzien van een kruis ten behoeve van de convergentie.
- Frequentie balken ter controle van de frequentie

karakteristieken van onder ander MF gedeelte en video eind versterker(s).

(15)

- Grijs gradatie balk.
- Geel/rood balk ter controle van de vertraging lijn.

Tenslotte van dit programma bevindt een eigen variatie van de SGT. (Slow Graf Text)

Dit programma is voor een groot gedeelte overgenomen uit het DAI handboek. Allerelei beveiligingen die moeten voorkomen dat de text string van het beeld af kan lopen zijn verwijderd. Daar dan tegen is dit programma uitgebreid met de optie dat men in staat stelt text onder allerlei verschillende hoeken te doen optekenen, dit in tegenstelling van andere versies van graf text waar dit allen mogelijk is met hoeken met veelvoud van 90 graden.

SGT is aan te roepen met 'GOSUB 40000'.

In het hoofd programma moet men ten behoeve deze subroutine een extra 'CLEAR 1400' plegen.

- Variabele 'A#' bevat de te printen text string.
- Variabelen 'X' en 'Y' bevatten de coördinaten van het startpunt van te printen string.
- Variabele 'DEV' is de hoek waarmee de string wordt afgedrukt, in radialen, tegen de klok in.
- 'C' bevat de kleur code waarmee . het wordt opgetekend
- 'F' bepaalt de vergrotings factor.

7 Cassette 'FGT'

(16)

Ik ben al een enige tijd in het bezit van het programma 'FGT'. het wordt tot volle tevredenheid gebruikt.

Een nadeel van dit programma is dat het een machinetaal programma is. Het is altijd lastig om deze met een basic programma te laden, hetzij apart of deze bevindt als data opslag in het basic programma met alle gevolgen van dien. In het nummer van het blad DAI namic geeft U een redelijke oplossing voor dit probleem, hetgeen ik nog niet uit geprobeert hebt.

Het grote voordeel dat het een machine taal programma is vindt ik ook de snelheid, welke belangrijker is dan de hier voor gegeven bezwaren.

De nu volgende file's ben ik niet in geslaagd om deze in te lezen: - Obj trigisch alfabet,

- Obj morse alfabet.

Wilt U de genoemde files mij alsnog doen toekomen?

DAI video-hardware

8 Ruilen cassette 'Viditel'.

(17)

=====

Op de HCC dag in Utrecht heb ik het programma viditel gekocht. Met dit programma heb ik nog geen ervaring omdat ik nog geen modem tot mijn beschikking heb. In de half-duplex mode heb ik al reeds er mee gespeeld. De belangrijkste functies die naar mening nog missen zijn:

- Het laten knippen van een symbool.
- De mogelijkheid van het opslaan van de pagina's.
- De mogelijkheid van het maken van een hard copy van een pagina.

Bijgevoegd is de cassette die ik heb gekocht.

Deze zou ik graag willen ruilen voor Viditel versie 3.0.

In de bespreking van Viditel in DAINamic jan./feb '82

neem ik aan dat de genoemde functies zijn verweselijkt.

10 Algemeen, slot.

(19)

=====

Veel lof heb ik voor het clubblad DAINamic.

Praktisch elke letter wordt door mij gelezen.

De software krijgt veel aandacht in het blad, b.v. het werk van de heer Boerrigter t.a.v het uitzoeken van de DAI firmware. Ik wacht al ongeduldig op het boek over het genoemde onderwerp.

Hardware matig gebeurt er vrijwel niets.

Graag zou ik willen zien dat er een compleet schema van de DAI computer wordt uitgegeven.

Sinds kort wordt het clubblad DAINamic gebonden, hetgeen veel beter bevalt dan de losbladige, er mee door gaan dus.

In afwachting verblijf ik

Hoogachtend

A. Doornenbal

A. Doornenbal,
Oud Aa 39a
3621 LA Breukelen.
Tel. 03462-3237

```
5 REM +++ DRAWING WITH PADDLES R.RENS +++
10 MODE 6:K%=8
20 COLORG 8 6 4 2:MODE 5
30 XM%=XMAX/2:YM%=YMAX/2:Y3%=YMAX/3:Y23%=YMAX*2/3
40 XN%=XM%:YN%=Y23%:UN%=XM%:VN%=Y3%
1000 XO%=XN%:YO%=YN%:UO%=UN%:VO%=VN%
1010 XD%=(127-PDL(1))*10/256:YD%=(PDL(2)-127)*10/256:UD%=(PDL(5)-127)*10/256:VD
%=PDL(4)-127)*10/256
1100 XN%=XO%+XD%:IF XN%<0 THEN XN%=0
1110 IF XN%>XMAX THEN XN%=XMAX
1120 YN%=YO%+YD%:IF YN%<Y3% THEN YN%=Y3%
1130 IF YN%>YMAX-1 THEN YN%=YMAX-1
1140 UN%=UO%+UD%:IF UN%<0 THEN UN%=0
1150 IF UN%>XMAX-1 THEN UN%=XMAX-1
1160 VN%=VO%+VD%:IF VN%<0 THEN VN%=0
1170 IF VN%>Y23% THEN VN%=Y23%
1200 DRAW XO%,YO% XN%,YN% K%:DRAW UO%,VO% UN%,VN% K1%
1220 EV1%=(PEEK(#FD00) IAND #20) SHR 5:IF EV1%=1 THEN FILL 0,Y3% XMAX,YMAX 8
1230 EV2%=(PEEK(#FD00) IAND #10) SHR 4:IF EV2%=1 THEN FILL 0,0 XMAX,Y23% 8
1300 Z%=PDL(0):ZD%=(Z%-1)/16:K%=ZD%
1310 W%=PDL(3):WD%=(W%-1)/16:K1%=WD%
1400 GOTO 1000
```

```

10 REM LARGE SIZE CHARACTERS / F.H. DRUIJFF 1/82
20 PRINT CHR$(12);
30 PRINT "LARGE SIZE CHARACTERS."
40 PRINT "=====
50 PRINT :PRINT "THIS PROGRAM SHOWS ANOTHER POSSEBILITY OF DAI."
60 PRINT "YOU CAN CREATE LARGE SIZE CHARACTERS BY CHANGING 7 IN THE"
70 PRINT "FIRST HALF OF THE CONTROLBYTE IN 6,5 OR 4."
80 PRINT :PRINT "SINCE THE REAL INFORMATION IS ONLY TWO BITS OF THIS BYTE"
90 PRINT "IT IS USELESS TO TRY OTHER NUMBERS THEN 7,6,5 OR 4."
100 PRINT "IF THE CHARACTERS HAVE A LARGER SIZE IT WILL BE CLEAR THAT"
110 PRINT "LESS CHARACTERS WILL FIT ON ONE LINE. IF YOU USE 7 THERE IS"
120 PRINT "PLACE FOR 66, WITH 6 44,WITH 5 22 AND WITH 4 11 CHARACTERS.":PRINT
130 PRINT "IF YOU WANT THE WHOLE SCREEN TO USE WITH NONSTANDARD CHARAC--"
140 PRINT "TERS, YOU'LL HAVE TO REBUILD THE SCREENAREA BY YOURSELF."
150 PRINT "STARTING EACH LINE WITH TWO(!) CONTROL BYTES; FILLING THE"
160 PRINT "LINE WITH CHARACTER AND COLOR INFORMATION."
170 PRINT "ALSO WILL THE NEW CONTROLBYTE HAVE TO FOLLOW ON THE EXPECTED"
180 PRINT "DISTANCE. (DEPENDING ON THE SIZE YOU USE)"
190 PRINT "NORMALLY YOU ONLY USE LARGER SIZE FOR HEADINGS."
200 PRINT "A SIMPLE WAY OF USING THIS IS: KEEP NORMAL LENGTHS AND WIPE"
210 PRINT "UNDESIRED INFORMATION WITH A 'COLORT x y x x' INSTRUCTION."
220 PRINT :PRINT "START DEMONSTRATION BY TYPING SPACE. ";
230 IF GETC=0 GOTO 230
240 PRINT CHR$(12):PRINT
250 COLORT 8 0 8 8
260 T$="* DAI *"
270 PRINT T$
280 WAIT TIME 100
290 POKE #BFEF-2*#86,#6A
300 WAIT TIME 100
310 POKE #BFEF-2*#86,#5A
320 WAIT TIME 100
330 POKE #BFEF-2*#86,#4A
340 PRINT :PRINT :PRINT :PRINT "NOW ON ONE PAGE."
350 WAIT TIME 100
360 PRINT CHR$(12)
370 PRINT SPC(20);T$
380 PRINT :PRINT SPC(12);T$
390 PRINT :PRINT SPC(4);T$
400 PRINT :PRINT T$
410 POKE #BFEF-3*#86,#6A
420 POKE #BFEF-5*#86,#5A
430 POKE #BFEF-7*#86,#4A
440 WAIT TIME 100
450 PRINT :PRINT :PRINT :PRINT "RESTART DEMONSTRATION WITH SPACEBAR AND STOP W
ITH S."
460 H=GETC
470 IF H=32 GOTO 240
480 IF H<>83 GOTO 460
490 END

```

large size characters

```

10 GOTO 200:REM BEER / F.H. DRUIJFF 2/82
20 H=(J+4)/10:DRAW X-H,J X29+H,J C:RETURN
30 FOR X=11 TO 291 STEP 56:X10=X+10:X18=X+18:X29=X+29
40 FOR Y=147 TO 2 STEP -3:FILL X10,Y X18,Y+2 22:NEXT
50 K=0:C=23:FOR J=2 TO 7:WAIT TIME 2:GOSUB 20:NEXT
60 FOR I=8 TO 63:C=22:J=I-6:GOSUB 20:WAIT TIME 2
70 C=23:J=I+K:GOSUB 20:IF I MOD 4=0 THEN K=K+1:J=I+K:GOSUB 20
80 NEXT:FOR Y=147 TO 80 STEP -3:IF Y=24*24 THEN J=J+1:GOSUB 20
90 FILL X10,Y X18,Y+2 20:NEXT:J=J+1:GOSUB 20:X1=X+16:X2=X+18:X3=X+17
100 FOR Y=149 TO 147 STEP -1:DOT X3,Y 22:WAIT TIME 2
110 DRAW X1,Y X2,Y 22:WAIT TIME 2:NEXT:DOT X3,146 22
120 FOR Y=145 TO 83 STEP -2:FILL X1,Y X2,Y+1 22
130 DOT X3,Y-1 22:FILL X1,Y+3 X2,Y+4 20:NEXT
140 DRAW X+10,82 X+24,82 23:FILL X1,84 X2,85 20:T=1-T
150 IF T=1 GOTO 100:DRAW X+14,83 X+22,83 23:NEXT:END
200 MODE 6A:COLORT 5 5 5 5:COLORG 6 0 14 15:FOR X=10 TO 290 STEP 56
210 DRAW X,0 X-8,81 21:FILL X,0 X+30,1 21:DRAW X+31,0 X+39,81 21
220 DRAW X-1,0 X-9,81 21:DRAW X+32,0 X+40,81 21:FILL X+10,154 X+20,211 21
230 DRAW X+13,154 X+13,211 22:FILL X+8,150 X+22,153 21:NEXT:GOTO 30

```

beer



alternative printer routine

The RS232 interface on your DAI is designed for operation of a printer via a handshake protocol. The serial output interface chip 5501 sends only data to the printer if a 'ready' signal is received. The software routine for serial output can be found on the addresses #DD94-#DDB3.

Several times I have been called by people who wanted to use a printer without the handshake facilities as expected by the DAI. Therefore I designed the annexed alternative printer routine.

This specific routine is written for a printer which had timing problems with the line feed signal. But we will see that the routine can easily be adapted to cope with other problems.

The principle of the routine is based on the DAI feature that input and output routines, available in the ROM's, can be replaced by any other, user specified, routine. As can be found in the memory map, the output direction is determined by the output direction pointer OTSW on address #131. For normal RS232 operation, it has the value #00. But if the value is #03, it points to the instruction on address #02DD (DOUTC). Normally, you will find a return instruction on this address, but this can be replaced by a jump instruction to a user specified routine in RAM.

The alternative routine starts with a initialisation routine. It changes the startaddress of the Heap to after this ML program. Then on address #02DD a jump to the alternative printer routine is loaded. If you load the routine directly after switching on the DAI, it must be followed immediately by UT > 6300 before loading a Basic program.
(Note: For machines with Basic V1.1 the NEW routine is on address #DEB8!).

The alternative printer routine starts on #0317. It is the same routine as on #DD94, except the additional delay on #0337. When required, another delay routine can be used for more or for less delay time.
The comments annexed to the ML routine indicate how to use this routine.

If your printer doesnot have any handshake facilities at all, the instructions on #0322-#0327 (waiting for a ready signal from the printer) can be cancelled. If required, a delay time can be inserted in stead of this 'wait for printer ready'.

It is possible of course to integrate this alternative printer routine in a Basic program. See therefore the article 'Combination Basic and ML programs/rules for standardization'
But then the lines #301-#309 must be skipped!

Jan Boerrigter - Febr.'82

alternative printer routine

```

002          *
003          ORG      :300
004          *
005 0300 E5      INIT  PUSH  H
006 0301 213C03  LXI   H, LAST
007 0304 229B02  SHLD  :029B      HEAP POINTER AFTER ROUTINE
008 0307 CDB5DE  CALL  :DEB5      RUN NEW (SHIFT TEXTBUF)
009 030A 21DD02  LXI   H, :02DD  )
010 030D 36C3    MVI   M, :C3    ) SET DOUTC FOR
011 030F 211703  LXI   H, RS232 ) REPLACEMENT
012 0312 22DE02  SHLD  :02DE  ) ROUTINE
013 0315 E1     POP   H
014 0316 C9     RET
015          *
016          *
017          *
018 0317 F5      RS232 PUSH  PSW
019 0318 EF      RST   5      OUTPUT TO SCREEN
020 0319 03      DATA  :03
021 031A 3A00FD  OUTRDY LDA  :FD00
022 031D E608    ANI   :08
023 031F CA1A03  JZ    OUTRDY    WAIT OUTPUT READY
024 0322 3AF3FF  BUFEMP LDA  :FFF3
025 0325 E610    ANI   :10
026 0327 CA2203  JZ    BUFEMP    WAIT BUFFER EMPTY
027 032A F1     POP   PSW
028 032B 32F6FF  STA  :FFF6      OUTPUT CHAR
029 032E FE0D    CPI   :0D      CAR.RET?
030 0330 C0      RNZ
031 0331 F5      PUSH  PSW      READY WHEN NOT
032 0332 3E0A    MVI   A, :0A
033 0334 CD1703  CALL  RS232    PRINT LINE FEED
034 0337 CD41DE  CALL  :DE41    DELAY
035 033A F1     POP   PSW
036 033B C9     RET
037          *
038 033C 00      LAST  NOP
039          *
040          *
041          * After loading this routine: UT >B 0300,
042          * to run initialisation.
043          *
044          * In Basic program: POKE #131,3 to activate
045          * printer; POKE #131,1 to switch off.
046          *
047          * For use with machine language programs, the ORG
048          * must refer to an address outside the ML program!
049          *
050 033D          END

```

```

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

```

```

BUFEMP 0322  INIT  0300  LAST  033C  OUTRDY 031A
RS232  0317

```


spot on text

```
10 REM SPOT ON TEXT / F.H. DRUIJFF 1/82
20 PRINT CHR$(12);:COLORT 8 0 8 14
30 PRINT "SPOT ON TEXT"
40 PRINT "=====":PRINT
50 PRINT "IN THIS PROGAM WILL BE SHOWN THAT YOU CAN CHANGE TEXTCOLOR"
60 PRINT "AND BACKGROUND COLOR OF ANY PART OF THE TEXT."
70 PRINT :PRINT "YOU PERFORM THIS BY FILLING THE COLORBYTES OF THE CHARACTERS"
80 PRINT "YOU WANT TO CHANGE WITH #FF (DECIMAL 255)."
```

90 PRINT "IF YOU CHOOSE ANYTHING ELSE THEN #FF THEN ONLY THE COLUMNS OF"
100 PRINT "THE CHARACTERMATRIX WILL CHANGE OF WHICH THE CORRESPONDING"
110 PRINT "BIT IN THE COLORBYTE IS 1."
120 PRINT "IF THIS IS TOO COMPLICATED AT THE MOMENT JUST TRY SOMETHING"
130 PRINT "AND REMEMBER TO CONVERT THE NUMBER YOU INTEND TO POKE IN BI-"
140 PRINT "NARY NOTATION."
150 PRINT "THE BYTE TO POKE IS FOUND IN THE FOLLOWING WAY:"
160 PRINT "?#BFEF-Y*#86-X*2-3 WITH X=POSITION IN LINE AND Y=LINE-1."
170 PRINT "REMEMBER THAT FIRST 3 CHARACTERS ARE NOT PRINTED BY PRINT."
180 PRINT :PRINT "THE COLORS USED ARE GIVEN BY A 'COLORT' INSTRUCTION."
190 PRINT "THE POKED COLORBYTES WILL FORCE THE SECOND SET."
200 FOR I=#BFEF-6*#86-36*2-3 TO I-18 STEP -2
210 POKE I,#FF:NEXT
220 FOR I=#BFEF-9*#86-8*2-3 TO I-28 STEP -2
230 POKE I,#FF:NEXT
240 FOR I=#BFEF-11*#86-19*2-3 TO I-20 STEP -2
250 FOR J=0 TO 8:POKE I,2^J:NEXT:NEXT
260 PRINT :PRINT "CONTINUE BY SPACE STOP WITH S."
270 C3=RND(16):C4=RND(16):IF C3=C4 GOTO 270
280 COLORT 8 0 C3 C4
290 CURSOR 0,1:PRINT "YOU ARE USING NOW COLORT 8 0";C3;C4;" ";
300 G=GETC:IF G=32 GOTO 270:IF G<>83 GOTO 300
310 COLORT 8 0 0 0
320 PRINT CHR\$(12):END

tumbling lines

```
5 COLORG 0 15 15 15
10 REM TUMBLING LINES F.H. DRUIJFF 2/82
20 MODE 4:CLEAR 2000:DIM X(18),Y(18):GOSUB 90:GOTO 50
30 FOR I=17 TO 0 STEP -1:P=X-X(I):DRAW X,0 P,Y(I) 22:DRAW X,0 P,Y(I) 0:NEXT:R
RETURN
40 FOR I=0 TO 18:P=X+X(I):DRAW X,0 P,Y(I) 22:DRAW X,0 P,Y(I) 0:NEXT:RETURN
50 FILL 0,0 24,26 22:FOR A=24 TO 0 STEP -1
60 X=A:GOSUB 40:E=XMAX-28:IF A=0 THEN E=E-26
70 FOR X=A+26 TO E STEP 26:GOSUB 30:GOSUB 40:NEXT
80 GOSUB 30:DRAW X,0 P,Y(0) 22:NEXT:END
90 FOR I=0 TO 18:READ X(I):Y(18-I)=X(I):NEXT:RETURN
100 DATA 0,2,4,6,8,10,12,14,16,18,20,21,22,23,24,25,25,26,26
```

patchwork

```
10 REM PATCHWORK / F.H. DRUIJFF 2/82
20 MODE 1:DIM C(3):REM PROGRAM RUNS ALSO IN MODE 3 OR 5
30 FOR I=XMAX-7 TO 0 STEP -8:FOR J=0 TO YMAX-3 STEP 4
40 FOR K=0 TO 3:C(K)=RND(16):NEXT:FILL I,J I+7,J+3 RND(16)
50 FOR Y=J TO J+3:M=2-M:FOR X=I+7 TO I STEP -1:N=1-N
60 K=M+N:DOT X,Y C(K):NEXT:NEXT:NEXT:NEXT:GOTO 30
```

10 REM FLUG-SIMULATOR A.MEYSTRE ANDLAUERSTR. 10
 15 REM FEB. 82 CH-4132 MUTTENZ SCHWEIZ

20 REM -----

25 REM

30 REM DIESER EINFACHE FLUG-SIMULATOR ZEIGT EINIGE
 35 REM VIDEO-EIGENSCHAFTEN DES DAI-COMPUTERS AUF.

40 REM

50 REM WERT MODUS PUNKTE/ZEILE BYTES/ZEILE

52 REM 0 UNIT-COLOR-MODE 528 4

55 REM 2 LOW 88 24

58 REM 4 MEDIUM 176 46

60 REM 6 HIGH 352 90

62 REM 8 ULTRA 528 134

65 REM

68 REM FALSCH EINGABE ERGIBT WERT 0

70 REM

72 REM TASTEN-FUNKTIONEN IM FLUG (mit REPT-Taste)

75 REM

78 REM CURSOR AUF/AB zum STEIGEN und FALLEN

80 REM CURSOR AUF/AB + SHIFT langsamer / schneller

85 REM CURSOR AUF+AB Ruettelflug

90 REM M-Taste Dieses Menue

95 REM

110 REM Bis zum fertigen Flugsimulator mit Piste und Berge

120 REM ist noch ein weiter Weg. Diese einfache Version

130 REM soll nur die Video-Moeglichkeiten durch

140 REM Manipulation der Zeilenkontrollbytes aufzeigen.

150 REM Die Wirkung ist augenblicklich !

160 REM Die Darstellung braucht in allen MODES 31 Zeilen.

170 REM Weitere Zeilen verschwinden am unteren Bildrand.

180 REM

190 REM Je nach MODE sind LI Anzahl Bytes/Zeile

200 REM . MCO Colorbyte-Maske

210 REM . MASK Modebyte-Maske

220 REM zu setzen.

1000 CLEAR 1000

1010 PRINT CHR\$(12):LIST -100:MCO=#CO:MIX=0

1020 INPUT "99 REM GEWUENSCHTER MODUS EINGEBEN ";GMO:PRINT

1030 IF GMO=2 THEN MODE 2:LI=24:MASK=#0:GOTO 1090

1040 IF GMO=4 THEN MODE 4:LI=46:MASK=#10:GOTO 1090

1050 IF GMO=6 THEN MODE 6:LI=90:MASK=#20:GOTO 1090

1051 REM

1052 REM Fuer die Darstellungen 0 und 8, die im BASIC nicht

1053 REM unterstuetzt werden, sind stoerende Kontrollbytes

1054 REM von MODE 6 auf 00 zu setzen.

1055 REM LCB zeigt jeweils auf das Kontrollbyte.

1056 REM

1060 MODE 6:LCB=#BFEF:FOR I=0 TO 60:POKE LCB,0:POKE LCB-1,0:LCB=LCB-90:NEXT

1070 IF GMO=8 THEN LI=134:MASK=#30:GOTO 1090

1080 LI=4:MCO=#80:MASK=#30

1090 COLORG 1 12 12 1:REM Grundfarben (Himmel)

1099 REM Tabellen fuer 31 Zeilen (von oben nach unten)

1100 DIM BL(30.0),BC(30.0),MIX(7.0)

1199 REM Zeilenbreite

1200 DATA 15,10, 1, 2, 3, 4, 4, 3, 2, 1,10,15,15,15, 1, 0, 1, 2, 3, 4, 5, 6, 7,
 8, 9,10,11,12,13,14,15

1299 REM Hintergrundfarbe der Zeile

1300 DATA 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,12,12,12,12,15,15,14,14,13,13,13, 6, 6,
 5, 5, 5, 5, 7, 7, 5, 7

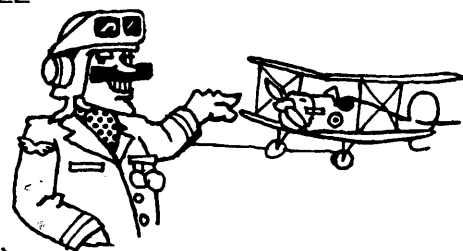
1310 REM Muster zur Farbmischung Dunkelblau/Hellblau

1320 DATA #20,#84,#51,#AA,#55,#BA,#6F,#FD

1400 RESTORE:REM Einlesen von Zeilenbreite und Farbe

1500 FOR I=0 TO 30:READ BL(I):NEXT:FOR I=0 TO 30:READ BC(I):NEXT:FOR I=0 TO 7:R
 EAD MIX(I):NEXT

flug simulator



```

1505 REM
1510 REM In der Bildmitte ist der duenne, weisse Horizont.
1520 REM Im oberen Bildteil wird der Himmel durch
1530 REM verschiedene Blautoene dargestellt. Die untere
1540 REM Bildhaelfte zeigt dahingleitende Felder in den
1550 REM Farben gruen, gelb und braun. Das Grundbild
1560 REM wird nun aufgebaut durch Veraenderung der
1570 REM Kontrollbytes-Informationen. LCB ist jeweils
1580 REM die Kontrollbyte-Adresse und LI ist die
1590 REM Differenz bis zum naechsten Kontrollbyte.
1595 REM
1600 LCB=#BFEE:FOR I=0 TO 30
1700 REM In den Zeilen 2 bis 9 wird ein Mischmuster eingesetzt
1800 IF I>1 AND I<10 THEN MLCB=LCB-2:FOR J=1 TO LI-2 STEP 2:POKE MLCB-J,MIX(MIX
):NEXT:MIX=MIX+1
1900 POKE LCB,BL(I) IOR MASK:POKE LCB-1,BC(I) IOR MCO:LCB=LCB-LI:NEXT
1950 REM
1960 REM U ist die Anfangsfluggeschwindigkeit
1970 REM H, H4, H8 und HC sind die Adressen der Bildkopfkontrollbytes
1980 REM Durch Veraenderung der Zeilenbreite in diesen 4 Kopfzeilen
1990 REM wird die Horizonthoehe verschoben.
1992 REM M ist eine Maske zum einsetzen der Zeilenbreite.
1995 REM
2000 U=6:H=#BFFF:H4=H-4:H8=H-8:HC=H-12:M=#30
2010 REM
2020 REM Ab hier entsteht die ganze Bewegung der unteren Bildhaelfte.
2030 REM Die Farbe der Zeile wird jeweils zum naechsttieferen Kontrollbyte
2040 REM versetzt. Dadurch entsteht die Illusion einer Vorwaertsbewegung.
2050 REM
2100 LCB=#BFEE-LI*16:F1=PEEK(LCB)
2110 FOR I=2 TO 15:LCB=LCB-LI:F2=PEEK(LCB):POKE LCB,F1:F1=F2:NEXT
2115 REM Am Horizont erscheint ein neues Feld mit Zufallsfarbe
2120 POKE #BFEE-LI*16,BC(15.0+RND(15.0)) IOR MCO
2122 REM
2125 K=GETC:REM einlesen der gewuenschten Bewegung
2127 REM Das Flugzeug steigt, die Geschwindigkeit nimmt dabei ab.
2128 REM Die Breite der obersten Zeilen nimmt zu, so dass der Horizont
2129 REM nach unten gleitet und Sie somit hoeher fliegen.
2130 REM
2132 IF K=16 AND U<15 THEN U=U+1:U3=U IOR M:POKE H,U3:POKE H4,U3:POKE H8,U3:POK
E HC,U3
2135 REM
2136 REM Das Flugzeug sinkt, die Geschwindigkeit nimmt zu. Die Breite der
2137 REM obersten Zeilen nimmt ab, dadurch wandert der Horizont
2138 REM gegen den oberen Bildrand. Die Erde kommt naeher und naeher.
2139 REM
2140 IF K=17 AND U>0 THEN U=U-1:U3=U IOR M:POKE H,U3:POKE H4,U3:POKE H8,U3:POKE
HC,U3
2145 REM
2146 REM Die Geschwindigkeit nimmt ab ( die Wartezeit S wird groesser ).
2147 REM
2150 IF K=20 OR K=16 THEN S=S+1
2155 REM
2156 REM Die Geschwindigkeit nimmt zu ( die Wartezeit S wird kleiner, aber nich
t negativ ).
2157 REM
2160 IF K=21 OR K=17 THEN S=S-1:IF S<0 THEN S=0
2165 REM
2166 REM wurde M eingegeben, so erscheint das Menuue.
2167 REM
2170 IF K=ASC("M") THEN MODE 0:GOTO 1010
2175 REM
2176 REM Vor der naechsten Bewegung wird umgekehrtproportional zur Geschwindigk
eit gewartet.
2177 REM
2180 WAIT TIME S:GOTO 2100
2200 REM
2210 REM Viel Spass ! Versuchen Sie mal dieses Programm zu erweitern,
2220 REM z.B. mit Motorgeraesch, Piste, Berge, Wolken, Bordinstrumente ...
2230 REM Bitte berichten Sie mir falls Sie etwas erreichen, viel Erfolg !

```

flug simulator


SPACE INVADER

On some machines, and for an unknown reason, the program doesn't initiate the values of POINTS, LASER and HIGHSCORE to the right contents.

This problem can be fixed with the following patch: The area from D4-DE must be cleared before starting the game:

```

49A    JMP      :3E0      jump to patch
3E0    LXI H   :0        old instr in 49A
3E3    SHLD   :D4        clear D4-DE
3E6    SHLD   :D6
3E9    SHLD   :D8
3EC    SHLD   :DA
3EF    SHLD   :DC
3F2    SHLD   :DD
3F5    JMP      :49D      cont program
    
```



If you have problems with SPACE INVADER on your machine, and you cannot make the patch yourself, please return the tape, we will send a new one, at no costs. (This applies to the whole library)

Ondergetekende zou gaarne in contact treden met DAI-gebruikers die bezig zijn met, c.q. reeds ontwikkeld hebben, een programma voor het samenstellen van roosters en/of clusters voor het gebruik op scholen. Liefst z.s.m.

R.A.B.FABER tel : 01856:2865
 Keizersdijk 36
 3291 CE STRIJEN NEDERLAND

OUTPUT TO RS232 ONLY

POKE H2DD,H3C : POKE H2DE,H94 : POKE H2DF,HDD : POKE H131,3
 switch back with : POKE H131,0

revision 7 of DAIPC has been released : DCR_users cannot use the first version of Eprom-print on this machine, Memocom is already supplying adapted prints.

short ml routines

INPUT FROM KEYBOARD

INPUT KEYBOARD
OUTPUT CHR IN A

```
IN  PUSH B
    PUSH D
    PUSH H
    CALL :D6BB
    POP  H
    POP  D
    POP  B
    RET
```

OUTPUT TO VIDEO

INPUT CHR IN A
OUTPUT VIDEO

```
OUT PUSH B
    PUSH D
    PUSH H
    CALL :D695
    POP  H
    POP  D
    POP  B
    RET
```

INPUT FROM RS232

INPUT RS232
OUTPUT CHR IN A

```
INM  PUSH B
     PUSH D
     PUSH H
     LDA  :FF03
     ANI  :04
     JZ   WAIC
     MVI  A,0
     JMP  RETM
WAIC  LDA  :FF03
     ANI  :08
     JZ   WAIC
     LDA  :FF00
RETM  POP  H
     POP  D
     POP  B
     RET
```

OUTPUT TO RS232

INPUT CHR IN A
OUTPUT RS232

```
OUTM PUSH B
     PUSH D
     PUSH H
     PUSH PSW
RED1  LDA  :FD00
     ANI  :08
     JZ   RED1
RED2  LDA  :FFF3
     ANI  :10
     JZ   RED2
     POP  PSW
     STA  :FFF6
     POP  H
     POP  D
     POP  B
     RET
```

8-BIT MULTIPLY

INPUT D=MULTIPLICAND
C=MULTIPLIER
OUTPUT BC=RESULT

```
MULT  MVI  B, :00
     MVI  E, :09
MULTO  MOV  A,C
     RAR
     MOV  C,A
     DCR  E
     JZ   DONE
     MOV  A,B
     JNC MULT1
     ADD  D
MULT1  RAR
     MOV  B,A
     JMP MULTO
DONE  RET
```

8-BIT DIVIDE

INPUT B=DIVIDEND
C=QUOTIENT
OUTPUT D=DIVISOR
E=REMAINDER

```
DIV  MOV  A,D
     CMA
     MOV  D,A
     MOV  A,E
     CMA
     MOV  E,A
     INX  D
     LXI  H, :00
     MVI  A, :17
DVO  PUSH H
     DAD  D
     JNC DV1
     XTHL
DIV1  POP  H
     PUSH PSW
     MOV  A,C
     RAL
     MOV  C,A
```

```
MOV  A,B
RAL
MOV  B,A
MOV  A,L
RAL
MOV  L,A
MOV  A,H
RAL
MOV  H,A
POP  PSW
DCR  A
JNZ  DVO
ORA  A
MOV  A,H
RAR
MOV  D,A
MOV  A,L
RAR
MOV  E,A
RET
```

```

10  REM DIT PROGRAMMA SIMULEERT EEN 4 TAKT MOTOR
20  POKE #75,32
110 REM VAST GEDEELTE
120 PRINT CHR$(12)
133 PRINT
135 COLORT 0 8 10 15:COLORG 0 8 15 6
136 MODE 4A
137 DRAW 37,10 43,10 21
138 DRAW 49,10 59,10 21
139 FILL 40,15 85,11 23
140 DRAW 39,10 39,56 21:DRAW 86,10 86,56 21
150 DRAW 37,7 52,7 21
160 DRAW 72,7 88,7 21
170 DRAW 65,10 75,10 21
180 DRAW 81,10 88,10 21
190 DRAW 52,7 52,9 21:DRAW 59,7 59,9 21
195 DRAW 65,7 65,9 21:DRAW 73,7 72,9 21
200 FILL 60,5 64,8 21
250 DOT 62,10 21:DOT 62,9 21
260 DRAW 64,9 64,12 21
270 DOT 62,12 21:DOT 63,12 21
280 REM KLEPPEN BEGINSTAND
290 DRAW 46,4 46,11 22:DRAW 78,4 78,11 22
300 DRAW 43,11 49,11 22:DRAW 75,11 81,11 22
320 REM ZUIGER BEGINSTAND
330 DRAW 40,15 85,15 22:DRAW 40,14 85,14 23
340 FILL 41,15 40,22 22:FILL 85,15 84,22 22
345 DOT 40,17 20:DOT 40,20 20:DOT 85,17 20:DOT 85,20 20
350 REM DRIJFSTANG
360 FILL 62,15 63,60 22
400 REM #### WERKING VAN DE MOTOR ####
500 REM INLAATSLAG
510 PRINT CHR$(12);"INLAATSLAG"
520 DRAW 43,12 49,12 22:DRAW 43,11 49,11 23
530 DOT 46,11 22:DOT 46,4 20
540 GOSUB 1000
550 REM COMPRESSIESLAG
560 DRAW 43,11 49,11 22:DRAW 43,12 49,12 23:DOT 46,4 22
570 PRINT CHR$(12);"COMPRESSIESLAG"
580 GOSUB 2000
620 REM ARBEIDSLAG
640 PRINT CHR$(12);"ARBEIDSLAG"
645 COLORG 0 8 15 10
650 GOSUB 1000
655 COLORG 0 8 15 6
670 DRAW 75,12 81,12 22:DRAW 75,11 81,11 23
680 DOT 78,11 22:DOT 78,4 20
710 PRINT CHR$(12);"UITLAATSLAG"
720 GOSUB 2000
730 DRAW 75,11 81,11 22:DRAW 75,12 81,12 23
740 DOT 78,4 22
750 GOTO 500
1000 POKE #305,#5B:POKE #306,#B6
1010 POKE #308,#2D:POKE #309,#B6
1020 POKE #31C,#D2:POKE #31D,#FF
1030 FOR J%=0 TO 28:CALLM #300:NEXT:RETURN
2000 POKE #31C,46:POKE #31D,0
2010 POKE #305,#88:POKE #306,#AF
2020 POKE #308,#B6:POKE #309,#AF
2030 FOR J%=0 TO 28:CALLM #300:NEXT:RETURN

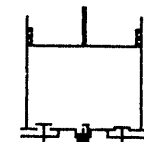
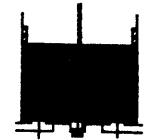
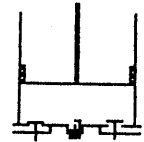
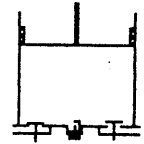
```

```

0300 E5 D5 C5 F5 21 88 AF 11 B6 AF D5 06 26 0E 0E 1A
0310 77 2B 1B 0D 79 B7 C2 0F 03 E1 E5 11 2E 00 19 5D
0320 54 E3 05 78 B7 C2 0D 03 F1 F1 C1 D1 E1 C9 20 20

```

4-takt motor



```

*****
*
* COMBINATION OF MACHINE LANGUAGE AND BASIC *
* ===== *
* - rules for standardization - *
* *
*****

```

Several programs, written in Basic, require that some parts are written in machine language to obtain a higher execution speed. There are several ways to combine these two parts. The most common methods are:

- Keep them separate. Before program execution, both routines must be loaded into the machine.
- Insert the ML program into DATA statements. Via a READ command, the instructions will be moved into an array during program execution.

The first method is not very useful when you just learned your kids how to load a simple Basic program; the second one needs a lot of typing with a high failure rate (although this can be overcome by using a data statement generator; but that requires more memory).

In order to get a normalisation in this matter, the following procedure is proposed as a standard for combining Basic and ML into one program.

Some standards are required:

- The startaddress of the Heap will always be #02EC (its default value).
- The ML routine starts always at #0300.

The length of the ML program is called MLPLEN (use hex-value!).

The method of combining Basic and ML routines is based on the principle that the ML program is moved into the text buffer. When you save a Basic program, both textbuffer and symboltable contents are written on tape, so now the ML routine is saved too. After loading, the ML part must be moved back to its original location.

The Basic program must contain the following lines at the very beginning of the program:

```

1 POKE #29B,#EC: POKE #29C,#02      Default start Heap
2 CLEAR xxxx                       xxxx > MLPLEN
3 MLPLEN=yy                         yy = length ML program
4 QQ=INT(SQR(MLPLEN/4+4))
5 DIM MLP(QQ,QQ)
6 STMLP=PEEK(#2A1)+PEEK(#2A2)*256-MLPLEN-1
7 FOR I=0 TO MLPLEN-1
8 POKE #300+I,PEEK(STMLP+I)
9 NEXT

```

These lines reserve by means of a dummy array memory space for the ML program. A 2-dimensional array is used to cope with ML programs of more than 1K. From line 7 onwards, the ML bytes are moved from the textbuffer into this area. This may last some time (FGT 8 secs). The ML routine can be called by Basic with CALLM #300.

The procedure (to be used during program development) for moving the ML program into the textbuffer is as follows:

```

Hard RESET
* CLEAR xxxx                       xxxx > MLPLEN
* UT
> R MLprogram
> B

```

combination of ml & BASIC

```
* LOAD "Basic program"
* UT
> D2A1 2A4 (2A2/1)=STBBGN
(2A4/3)=STBUSE
> M(2A2/1) (2A4/3) (2A2/1)+MLPLEN+1 Shift symboltable
> S(2A2/1) into (2A2/1)+MLPLEN+1 Update STBBGN
> S(2A4/3) into (2A4/3)+MLPLEN+1 Update STBUSE
> M300 300+MLPLEN (2A2/1)
> B
* SAVE "Basic program incl. MLP"
```

The CLEAR reserves space for the ML program. After loading both parts, the symboltable is moved to make place for the MLP. Then the pointers for the symboltable are updated. Now the MLP is moved into the now free area after the textbuffer. After saving, the program now on tape contains both Basic and ML programs. It can be loaded, executed, updated, and listed (listing the MLP part produces rubbish of course). Editing is only allowed until the last Basic linenumber!! Editing the ML part destroys your program!!!!

As an example, the transfer of FGT will be demonstrated. FGT is written on the memory locations #300-#900, but saved on tape with the Heap pointers: #29B-#900.

```
> R FGT(29B-900)
> W300 900 FGT(300-900) Save FGT only
Hard RESET
* CLEAR #700 MLPLEN=#600
* UT
> R FGT(300-900)
> B
* LOAD "Basic program"
* UT Suppose you find:
> D2A1 2A4 STBBGN = #126A
STBUSE = #1325
> M126A 1325 186B #186B=#126A+#600+1
> S2A1 6A-6B 12-18 25-26 13-19 #1926=#1325+#600+1
> M300 900 126A
> B
* SAVE "Complete program"
```

If the ML routine needs RAM-space for pointers, the area #2EE-#2FF can be used. If you need more space, place them above the ML routine. This area must be also counted for MLPLEN!

A final remark: This method is a proposal for standardization. You don't have to follow it. As long as you only write programs to be used by yourself, nobody cares about any standard. But if you intend to send your programs to DAInamic for the software library, to be used by other people, then certain rules are necessary.

I don't know your opinion. But I hate programs that require the loading of separate routines. I hate to find out after loading the routines that I should have moved the Heap before loading. I also hate all those different startaddresses of ML programs. I do like those programs which require only a simple LOAD command after the machine is switched on. And I am sure you like that too!

Jan Boerrigter


```

174      * IF FIRST CHAR IS INPUTTED ON RS232 DEVICE THEN
175      * INSW IS SET TO 1.
176      * NOTE THAT INSW IS ALSO USED FOR OTHER PURPOSES
177      * IN DAI DOS AND MEMOCOM DCR SOFTWARE.
178      *
179      GETFRC EQU :D6BB      FORCE 1 SCAN TO GET CHAR
180      * ENTRY : INPUT SWITCH INSW SET TO 0 OR 1
181      * FUNC  : IDEM GETC, KNSCAN = 0 NOT NEEDED, BUT
182      *        GETFRC HAS POOR DEBOUNCING.
183      * EXIT  : AS GETC
184      *
185      OUTC  EQU  :DD60      OUTPUT A CHAR
186      * ENTRY : ASCI VALUE IN A
187      *        OTSW SET TO 0, 1, 2 OR 3
188      * FUNC  : OUTPUT CHAR TO DEVICE OR BUFFER AS
189      *        SPECIFIED IN OTSW
190      * EXIT  : A, F CORRUPTED
191      *        OTSW = 0 : CHAR ON SCREEN (AT CURSOR)
192      *                  AND TO RS232
193      *        OTSW = 1 : CHAR ON SCREEN ONLY
194      *        OTSW = 2 : CHAR APPENDED TO TEXT IN EDIT-
195      *                  BUFFER (DON'T USE, EDITBUFFER
196      *                  POINTERS MUST BE SET UP FIRST)
197      *        OTSW = 3 : CHAR SEND TO A ROUTINE VIA
198      *                  A VECTOR ON DOUTC
199      *                  (NORMALLY DO NOTHING, JUST RET
200      *                  USED FOR DISK OPERATING SYSTEM)
201      OTSW   EQU  :131      OUTPUT SWITCHING
202      *
203      OUTCPR EQU :D695      OUTPUT CHAR, PRESERVE PSW
204      * ENTRY : IDEM OUTC
205      * FUNC  : IDEM OUTC
206      * EXIT  : IDEM OUTC, BUT A, F NOT CORRUPTED
207      *
208      CRLF  EQU  :DD5E      OUTPUT A CARRIAGE RETURN
209      * ENTRY : IDEM OUTC, BUT CHAR IS FORCED TO
210      *        ASCI VALUE :D
211      * FUNC  : FOR CRLF IDEM OUTC
212      * EXIT  : IDEM OUTC
213      *
214      *
215      COLO  EQU  :DD55      OUTPUT CR IF CURX <> 0
216      * ENTRY : OTSW SET TO 0, 1, 2 OR 3
217      * FUNC  : IF CURX <> 0 THEN OUTPUT CAR RET
218      * EXIT  : IDEM OUTC
219      *
220      *
221      * PART 3 : MESSAGE OUTPUT ROUTINES
222      * -----
223      *
224      PMSG  EQU  :DAD4      PRINT A MESSAGE
225      * ENTRY : POINTER TO MESSAGE IN HL
226      * FUNC  : PRINT A MESSAGE, WHICH MAY INCLUDE
227      *        INTERNAL REFERENCES TO OTHER SUBMESS.
228      * EXIT  : POINTER AFTER STRING IN HL
229      * MESSAGE FORMAT :
230      *   A SERIES OF BYTES :
231      *   0      -> END OF STRING
232      *   1-:7F -> ASCI VALUE CHAR TO PRINT
233      *   >:80 -> BITS 0-13 ARE OFFSET FROM :C000 TO ..
234      *               IF BIT 14 = 1 THEN
235      *               .. TO A MESSAGE (CHARS TERMINATED
236      *               BY A 0)
237      *               IF BIT 14 = 0 THEN

```

```

238 * .. TO A STRING (1 BYTE LENGTH
239 * + CORRECT NUMBER OF CHAR)
240 * REF : MESSAGES IN DAI ROM STARTING AT ADDR :DB6F
241 * AND POINTERS TO ERROR MESSAGES AT :DA94
242 *
243 PMSGR EQU :DAFF PRINT A MESSAGE
244 * ENTRY : POINTER TO MESSAGE IN THE 2 MEMORY LOCA-
245 * TIONS AFTER THE CALL TO PMSGR (LOW-HIGH)
246 * FUNC : IDEM PMSG
247 * EXIT : IDEM PMSG
248 *
249 PSTR EQU :DB32 PRINT A STRING
250 * ENTRY : POINTER TO STRING IN HL
251 * FUNC : PRINT A STRING
252 * EXIT : HL POINTS AFTER STRING
253 * STRING FORMAT :
254 * 1 BYTE LENGTH ( 0 = NO DATA )
255 * N BYTES DATA (ASCII VALUES)
256 *
257 PSTRM EQU :DB44 PRINT A STRING
258 * ENTRY : POINTER TO STRING IN HL
259 * LENGTH OF STRING IN A
260 * FUNC : PRINT A STRING
261 * EXIT : HL POINTS AFTER STRING
262 * FORMAT :
263 * N BYTES DATA (ASCII VALUES)
264 *
265 *
266 * PART 4 : GETTING INPUT AND ENCODE IT
267 * -----
268 *
269 * NOTE : DAI BASIC USE A SPECIAL WAY TO HANDLE
270 * INPUT OF COMMANDS, PROGRAM LINES AND
271 * 'INPUT' VALUES.
272 * FIRST ALL INPUTTED KEYS ARE SEND TO THE
273 * SCREEN (=TO THE CURRENT LINE).
274 * THIS CONTINUES UNTIL A CARRIAGE RETURN IS
275 * PRESSED.
276 * NOW THE PROPER ROUTINE FETCHES THE
277 * CHARACTERS BACK FROM THE SCREEN (=FROM THE
278 * CURRENT LINE) TO COMPILE OR TRANSLATE THEM.
279 * ADVANTAGE : NO NEED FOR A EXTRA INPUTBUFFER
280 * TO STORE CHARACTERS ; CHAR DEL IS HANDLED
281 * BY THE SCREEN PACK. A CURRENT LINE MAY
282 * HAVE 3 CONTINUATION LINES.
283 * ABOVE METHOD IS NOT USED IN PC UTILITY.
284 *
285 INLINE EQU :DD1A GET CHARACTERS UNTIL CR
286 * ENTRY : PROMPTCHARACTER IN A
287 * FUNC : IF CURX<>0 THEN PRINT CR (DO COLO)
288 * PRINT CHARACTER IN A
289 * ENABLE FULL KEYBOARDSCAN (KNSCAN->0)
290 * INPUT CHARACTERS FROM KEYBOARD OR RS232
291 * ( REF TO GETC FUNCTION )
292 * IF CHAR = CAR RET THEN TERMINATE
293 * IF BREAK PRESSED THEN TERMINATE
294 * IF PRINTABLE OR CHARDEL THEN SEND
295 * CHARACTER TO SCREEN
296 * INPUT NEXT CHARACTER
297 * EXIT : CY = 0 : VALID TERMINATE WITH CAR RET
298 * REG C IS SET TO 1 ( POSITION
299 * FIRST INPUTTED CHARACTER )
300 * NOTE : NO CR IS OUTPUTTED TO
301 * THE SCREEN -> CURSOR AFTER

```

```

302      *                               END OF INPUTTED CHARACTERS
303      *                               CY = 1 : BREAK PRESSED
304      *                               LINE IS TERMINATED WITH A '?'
305      *                               CHARACTER AND A CAR RET
306      *                               KNSCAN IS ALWAYS SET FOR SCAN BREAK ONLY
307      * NOTE : A CHAR IS PRINTABLE IF ASCI VAL >= :20
308      *
309      INLINX EQU :DD1F      GET CHARACTERS, NO COLO
310      * ENTRY : IDEM INLINE
311      * FUNC  : IDEM INLINE, BUT NO COLO PERFORMED AT
312      *        BEGIN ( ALLOWS EG. XYZ> PROMPTING )
313      * EXIT  : IDEM INLINE, BUT POSITION C=1 NOT VALID
314      * EXAMPLE OF USAGE :
315      *        LXI H,MSG#    MESSAGE BEFORE INPUT
316      *        CALL PMSG
317      *        RST 5        GET CURSOR POSITION
318      *        DATA :0C    X POS IN REG L
319      *        INR L        INCREMENT FOR PROMPT
320      *        PUSH H      SAVE CURX
321      *        MVI A,' '   BLANK PROMPT
322      *        CALL INLINX
323      *        POP B       GET CURX IN REG C
324      *        JC BREAK    EXIT IF BREAK PRESSED
325      *        REM : READY TO USE ENCINP OR LOOKC
326      *
327      ENCINP EQU :DDE0     GET CHARACTER TO ENCODE
328      * ENTRY : C : POSITION ON CURRENT LINE OF FIRST
329      *        CHAR TO INPUT (MAX 219)
330      *        EFSW : SOURCE FOR CHARACTERS
331      *        SOURCE : CURRENT SCREENLINE OR
332      *        STRING OR EDITBUFFER
333      * FUNC  : CHECK EFSW FOR SOURCE
334      *        = 0 : FETCH CHARACTER FROM CURRENT LINE
335      *        (=CURSOR LINE) OF SCREEN, A CAR RET
336      *        CHAR IS RETURNED IF AT OR PAST
337      *        CURSOR
338      *        = 1 : FETCH CHARACTER FROM STRING
339      *        = 2 : FETCH CHARACTER FROM EDITBUFFER
340      *        NOTE : FOR 1 AND 2 EFEPT (:132) AND EFECT
341      *        (:134) MUST BE SET UP CORRECTLY,
342      *        THIS WILL BE DESCRIBED ANOTHER TIME
343      * EXIT  : CHARACTER IS RETURNED IN REG A
344      *        A CAR RET CHAR FLAGS END OF SOURCELINE,
345      *        POSITION IN REG C IS NOT CHANGED, THIS
346      *        MUST BE DONE BY THE USER (REASON : SAME
347      *        POSITION MAY BE FETCHED MORE THAN ONCE)
348      EFSW EQU :135      ENCODING INPUT SWITCH
349      *
350      IGNB EQU :DDD2     GET CHAR, IGNORE BLANKS
351      * ENTRY : IDEM ENCINP
352      * FUNC  : IDEM ENCINP, BUT SPACE AND TAB (ASCI :9)
353      *        ARE IGNORED (NOT RETURNED)
354      * EXIT  : IDEM ENCINP, BUT POSITION IN REG C MAY
355      *        BE INCREMENTED IF SPACE(S) OR TAB(S)
356      *        FOUND
357      *
358      LOOKC EQU :CA34    LOOKUP INPUT IN A TABLE
359      * ENTRY : C : POSITION ON CURRENT LINE
360      *        HL : STARTADDRESS TABLE
361      *        E : (NUMBER OF CODEBYTES IN TABLE) - 1
362      * FUNC  : FETCH CHARACTERS VIA EFSW, STARTING AT
363      *        POSITION IN C
364      *        LEADING BLANKS OR TABS ARE IGNORED
365      *        MATCH A SEQUENCE OF SOURCE CHARACTERS

```

```

366      *           WITH KEYWORDS IN THE TABLE
367      *           THE WHOLE TABLE IS SCANNED, RESULTBYTES
368      *           AFTER THE KEYWORDS ARE SKIPPED
369      * EXIT   : CY = 1 : MATCHING KEYWORD FOUND
370      *           HL : POINTS TO BYTE AFTER KEYWORD
371      *           = FIRST RESULTBYTE
372      *           C  : POSITION AFTER LAST CHAR OF
373      *           MATCHED SEQUENCE
374      *           A  : LAST CHAR FETCHED FROM SCREEN
375      *           CY = 0 : NO MATCHING KEYWORD FOUND
376      *           C  : MAY BE CHANGED IF SPACE OR TABS
377      *           SKIPPED, IN EACH CASE POSITION
378      *           OF FIRST NON BLANK/TAB CHAR
379      *           HL : POINTS AFTER END OF TABLE
380      *           A  : ZERO
381      *
382      * TABLE FORMAT :
383      * A NUMBER OF RECORDS, EACH IN FORMAT :
384      *   1 BYTE   = LENGTH OF KEYWORD
385      *   N BYTES  = KEYWORD IN ASCI
386      *   M BYTES  = RESULTBYTES, VECTOR TO ROUTINE
387      *               OR A CODEVALUE
388      *   NOTE : M MUST BE SAME FOR ALL RECORDS (M-1 IS
389      *           A INPUT (REG C) TO THE LOOKC FUNCTION)
390      * END OF TABLE : DUMMY RECORD WITH LENGTH OF
391      *                   KEYWORD = 0
392      *
393      * EXAMPLE : (SET REG E TO 2)
394      *   DATA  3
395      *   ASC   'DNA'
396      *   DATA  :B0          CONTROLVALUE
397      *   DBL   :1100        POINTER TO ROUTINE
398      *   DATA  7
399      *   ASC   'NOTHING'
400      *   DATA  :FF
401      *   DBL   :FFFF
402      *   DATA  0          FLAGS END OF TABLE
403      *
404      * REF : BASIC COMMAND TABLE STARTING AT ROM ADDRESS
405      *       :CBBF, RESULTBYTES : 1 BYTE COMMANDCODE,
406      *       2 BYTES VECTOR TO COMPILE ROUTINE
407      *
408 0000      END

```

```

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

```

ADDA	DE30	ADDMI	DE39	ALFA	DE02	ALFNUM	DE09
COLD	DD55	COMPDE	DE14	CRLF	DD5E	DELAY	DE41
EFSW	0135	ENCINP	DDE0	FILL	DE7C	GETC	D6BE
GETFRC	D6BB	IGNB	DDD2	INLINE	DD1A	INLINX	DD1F
INSW	0296	KNSCAN	O2B9	LOOKC	CA34	MOVE	DE4F
MULA	DEBF	NEGHL	DE26	NUMER	DE0D	OTSW	0131
OUTC	DD60	OUTCPR	D695	PMSG	DAD4	PMSGR	DAFF
POPRET	C14D	PSTR	DB32	PSTRM	DB44	RSTART	C80C
SUBDE	DE1A	SUBDED	D790				

math-pack & rom-interface

```

002 *****
003 *
004 * THIS DEMO SHOWS USE OF INFO DESCRIBED IN : *
005 * DAI MATH-PACK (REF TO DAINAMIC 82/1,PAGE 7) *
006 * DAI ROM INTERFACE (82/1,PAGE 14 AND 82/2) *
007 *
008 * WITH COMMANDS ENTERED AFTER A SPECIAL PROMPT *
009 * IT IS POSSIBLE TO PERFORM SOME FUNCTIONS ON A *
010 * 4 BYTE FLOATING POINT VALUE. *
011 * AFTER EACH FUNCTION THE VALUE IS PRINTED IN *
012 * DECIMAL (FLOATING POINT) AND HEXADECIMAL FORMAT.*
013 * THE VALUE IS STORED IN A RAM AREA OF 4 BYTES *
014 * STARTING AT LABEL 'NUMBER'. *
015 *
016 *****
017 *
018 * ORG : P. JONGEN AND F. DE RAEDT
019 *
020 * INTERFACE WITH DAI ROM (REF DAI INTERFACE) :
021 POPRET EQU :C14D POP REGS, RET
022 LOOKC EQU :CA34 LOOKUP CMD IN TABLE
023 PMSG EQU :DAD4 PRINT A MESSAGE
024 PSTR EQU :DB32 PRINT A STRING
025 INLINE EQU :DD1A GET CHARACTERS UNTIL CR
026 COLO EQU :DD55 OUTPUT CR IF CURX<>0
027 CRLF EQU :DD5E OUTPUT CR
028 IGNB EQU :DDD2 GET CHAR, IGN BLANKS
029 NUMER EQU :DE0D CHECK 0..9
030 *
031 * FOLLOWING ENTRYPPOINTS FOR NUMBER INPUT/OUTPUT
032 * ARE TAKEN FROM THE DAI MANUAL PAGE 126 :
033 XFBC EQU :C01E INPUT FPT
034 XFBC EQU :C021 CONVERT FPT
035 XHBC EQU :C02D CONV HEX
036 XPRTY EQU :C030 PRETTY UP
037 *
038 * DECLARE LITERALS :
039 FF EQU OC FORMFEED
040 CR EQU :OD CAR RET
041 PRCH EQU :40 PROMPTCHARACTER
042 *
043 *
044 * SET BEGIN OF PROGRAMCODE :
045 ORG :300 THIS ALSO ENTRY OF PROGRAM
046 UT>Z3>G300 OR *CALLM #300
047 *
048 * PART 1 : COMMAND HANDLER
049 *
050 0300 F5 ENTRY PUSH PSW SAVE ALL REGISTERS
051 0301 C5 PUSH B ON STACK
052 0302 D5 PUSH D
053 0303 E5 PUSH H
054 0304 214203 LXI H,TITL$ PRINT TITLE
055 0307 CDD4DA CALL PMSG
056 030A 212D04 LXI H,HELP$ PRINT MENU
057 030D CDD4DA CALL PMSG
058 0310 C39603 JMP ZERO1 INIT NUMBER TO 0
059 * CODE ENTERED WITH PREVIOUS JUMP TRANSFERS CONTROL
060 * TO THE FOLLOWING LINE
061 0313 3E40 NEXT MVI A,PRCH
062 0315 CD1ADD CALL INLINE GET CHAR UNTIL CAR RET
063 0318 DA1303 JC NEXT CANCEL LINE IF BREAK
064 * INLINE RETURNS WITH C = 1 : POINTS TO FIRST
065 * INPUTTED CHARACTER ON CURRENT LINE

```

math-pack & rom-interface

```

066 031B CDD2DD          CALL  IGNB      IF LINE HAS ONLY BLANKS
067 031E FE0D          CPI    CR        THEN NO VALID INPUT
068 0320 CA1303        JEQ    NEXT      AND START AGAIN
069                    * NOW LOOK FOR A VALID COMMAND
070 0323 216603        LXI    H,CMDTBL  BEGIN OF COMMANDTABLE
071 0326 1E01          MVI    E,CMDLNG-1 LENGTH RESULT BYTES TABLE
072 0328 CD34CA        CALL  LOOKC     LOOKUP CMD IN TABLE
073 032B D23603        JNC    SYNERR   NO VALID COMMAND FOUND
074 032E 5E           MOV    E,M      GET ROUTINE ADDR
075 032F 23           INX    H        FROM TABLE
076 0330 56           MOV    D,M
077 0331 EB           XCHG          AND TRANSFER CONTROL
078 0332 E9           PCHL         TO THE ROUTINE
079                    * THIS IS A JUMP, NOT A CALL ; IF NO ERRORS OCCUR
080                    * THE ROUTINE SHOULD JUMP BACK TO 'NEXT'
081                    *
082                    EXIT          CMD : EXIT TO BASIC OR UTIL
083 0333 C34DC1        JMP    POPRET   RESTORE ALL REGISTERS
084                    *
085                    SYNERR       ROUTINE HAS FOUND INVALID
086                    SYNTAX IN COMMAND LINE
087 0336 CD55DD        CALL  COLO
088 0339 2123DC        LXI    H,:DC23  MSG 'SYNTAX ERROR'
089 033C CDD4DA        CALL  PMSG
090 033F C31303        JMP    NEXT
091                    *
092 0342 0C           TITL$  DATA  FF
093 0343 44454D        ASC   'DEMO ROM INTERFACE AND '
094 035A 4D4154        ASC   'MATH PACK'
095 0363 0D0D00        DATA CR,CR,0
096                    *
097                    * TABLE FOR COMMAND LOOKUP :
098                    CMDLNG EQU 2      LENGTH OF RESULT FIELD
099 0366 04           CMDTBL DATA 4      LENGTH OF COMMAND$
100 0367 5A45524F      ASC   'ZERO'      COMMAND$
101 036B 8E03          DBL   ZERO        RESULT FIELD
102 036D 03           DATA 3          NEXT COMMAND ...
103 036E 414444        ASC   'ADD'
104 0371 9F03          DBL   ADD
105 0373 04           DATA 4
106 0374 45584954      ASC   'EXIT'
107 0378 3303          DBL   EXIT
108 037A 03           DATA 3
109 037B 4E4547        ASC   'NEG'
110 037E BB03          DBL   NEG
111 0380 03           DATA 3
112 0381 535152        ASC   'SQR'
113 0384 CD03          DBL   SQR
114 0386 04           DATA 4
115 0387 48454C50      ASC   'HELP'
116 038B 2404          DBL   HELP
117 038D 00           DATA 0          FLAG END OF TABLE
118                    *
119                    ZERO          CMD : ZERO OUT NUMBER
120 038E CDD2DD        CALL  IGNB      TEST OTHER CHARACTERS ON
121 0391 FE0D          CPI    CR        SAME LINE
122 0393 C23603        JNE    SYNERR   ERROR IF TRUE
123 0396 AF           ZAR01 ZAR       MAKE A 0 FPT VALUE
124 0397 47           MOV    B,A      IN REGS A,B,C,D
125 0398 4F           MOV    C,A
126 0399 57           MOV    D,A
127 039A E7           RST    4        SEND IT TO FPT ACC
128 039B 12           DATA :12
129 039C C3DF03        JMP    DONE     DUMP RESULT

```

```

130
131          *
          ADD          CMD : ADD INPUTTED VALUE
132 039F CDD2DD      CALL  IGNB          IF CHARACTER AFTER CMD
133 03A2 CD0DDE      CALL  NUMER        NOT NUMBER THEN
134 03A5 D23603      JNC   SYNERR        SYNTAX ERROR
135 03A8 CD1EC0      CALL  XFBC          INPUT FPT NUMBER TO ACC
136 03AB CDD2DD      CALL  IGNB          CHECK FOR ILLEGAL CHARS
137 03AE FE0D        CPI   CR           AFTER NUMBER
138 03B0 C23603      JNE   SYNERR
139 03B3 218E05      LXI   H,NUMBER    POINTS TO NUMBER
140 03B6 E7          RST   4           ADD NUMBER TO
141 03B7 00          DATA :00        FPT ACC
142 03BB C3DF03      JMP   DONE          DUMP RESULT
143
144          *
          NEG          CMD : NEGATE NUMBER
145 03BB CDD2DD      CALL  IGNB          ERROR IF CHARS AFTER
146 03BE FE0D        CPI   CR           COMMAND
147 03C0 C23603      JNE   SYNERR
148 03C3 218E05      LXI   H,NUMBER    POINTS TO NUMBER
149 03C6 E7          RST   4
150 03C7 0C          DATA :0C        FETCH NUMBER IN FPT ACC
151 03C8 E7          RST   4
152 03C9 1B          DATA :1B        CHANGE SIGN OF FPT ACC
153 03CA C3DF03      JMP   DONE          DUMP RESULT
154
155          *
          SQR          CMD : SQR OF NUMBER
156 03CD CDD2DD      CALL  IGNB          ERROR IF CHARS AFTER
157 03D0 FE0D        CPI   CR           COMMAND
158 03D2 C23603      JNE   SYNERR
159 03D5 218E05      LXI   H,NUMBER    POINTS TO NUMBER
160 03D8 E7          RST   4
161 03D9 0C          DATA :0C        FETCH NUMBER IN FPT ACC
162 03DA E7          RST   4
163 03DB 33          DATA :33        TAKE SQR OF FPT ACC
164 03DC C3DF03      JMP   DONE          DUMP RESULT
165
166          *
          * DUMP RESULT :
167          *   PUT RESULT IN NUMBER
168          *   DISPLAY RESULT IN  FPT AND HEX FORMAT
169          *   GOTO NEXT FOR NEW COMMAND
170          *
171 03DF 218E05      DONE LXI   H,NUMBER    POINTS TO NUMBER
172 03E2 E7          RST   4
173 03E3 0F          DATA :0F        STORE FPT ACC IN NUMBER
174 03E4 CD5EDD      CALL  CRLF        NEW LINE
175 03E7 210D04      LXI   H,VAL$     PRINT 'DEC : '
176 03EA CDD4DA      CALL  PMSG
177 03ED CD21C0      CALL  XFBC        CONVERT FPT FOR OUTPUT
178 03F0 3E06        MVI   A,6         PRECISION
179 03F2 CD30C0      CALL  XPRTY       PRETTY UP
180 03F5 21E300      LXI   H,:E3       POINTS TO CONVERTED VALUE
181 03FB CD32DB      CALL  PSTR        AND PRINT IT
182                  (FORMAT : LENGTH+CHARS)
183 03FB 211904      LXI   H,HEX$     PRINT : 'HEX : #'
184 03FE CDD4DA      CALL  PMSG
185 0401 CD2DC0      CALL  XHBC        CONVERT TO HEX
186 0404 21E300      LXI   H,:E3       AND PRINT
187 0407 CD32DB      CALL  PSTR
188 040A C31303      JMP   NEXT       TIME FOR NEXT COMMAND
189
190          *
          VAL$        ASC  'VALUE DEC : '
191 0418 00          DATA 0
192 0419 202020      HEX$  ASC  '   HEX : #'
193 0423 00          DATA 0

```

math-pack & rom-interface

```

194
195
196 0424 212D04
197 0427 CDD4DA
198 042A C31303
199
200 042D 0D
201 042E 544849
202 044E 204655
203 0469 0D
204 046A 494E20
205 0481 0D
206 0482 414654
207 04A0 495320
208 04BC 0D
209 04BD 464F52
210 04C4 0D
211 04C5 56414C
212 04D9 0D
213 04DA 20205A
214 04F3 0D
215 04F4 202041
216 050F 202850
217 052E 0D
218 052F 20204E
219 0543 0D
220 0544 202053
221 0555 0D
222 0556 202048
223 056C 0D
224 056D 202045
225 058B 0D0D00
226
227
228 058E
229
230 0592

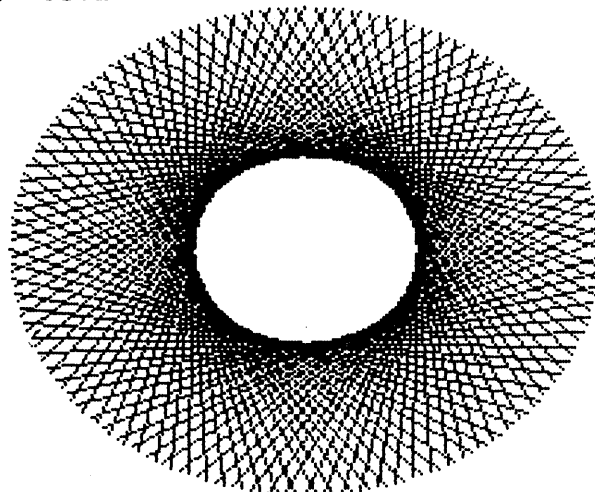
*
HELP
LXI H,HELP$
CALL PMSG
JMP NEXT

*
HELP$ DATA CR
ASC 'THIS DEMO ALLOWS SOME MATHEMATIC'
ASC 'FUNCTIONS ON A VALUE SAVED'
DATA CR
ASC 'IN THE LOCATION NUMBER.'
DATA CR
ASC 'AFTER EACH FUNCTION THE VALUE '
ASC 'IS DISPLAYED IN FPT AND HEX '
DATA CR
ASC 'FORMAT.'
DATA CR
ASC 'VALID COMMANDS ARE : '
DATA CR
ASC ' ZERO : RESET VALUE TO 0'
DATA CR
ASC ' ADD XXX.XX : ADD TO VALUE'
ASC ' (POSITIVE AND START WITH 0..9)'
DATA CR
ASC ' NEG : CHANGE SIGN'
DATA CR
ASC ' SQR : TAKE SQR'
DATA CR
ASC ' HELP : SHOW COMMANDS'
DATA CR
ASC ' EXIT : RETURN TO UT OR BASIC'
DATA CR,CR,0

*
*
NUMBER RES 4,0 AREA FOR FPT NUMBER
*
END
    
```

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

ADD	039F	CMDLNG	0002	CMDTBL	0366	COLO	DD55
CR	000D	CRLF	DD5E	DONE	03DF	ENTRY	0300
EXIT	0333	FF	000C	HELP	0424	HELP\$	042D
HEX\$	0419	IGNB	DDD2	INLINE	DD1A	LOOKC	CA34
NEG	03BB	NEXT	0313	NUMBER	058E	NUMER	DE0D
PMSG	DAD4	POPRET	C14D	PRCH	0040	PSTR	DB32
SQR	03CD	SYNERR	0336	TITL\$	0342	VAL\$	040D
XFCB	C021	XFCB	C01E	XHBC	C02D	XPRTY	C030
ZERO	038E	ZERO1	0396				



math-pack & rom-interface

TAPE 1980-1981

J#P.

```

0300 F5 C5 D5 E5 21 42 03 CD D4 DA 21 2D 04 CD D4 DA
0310 C3 96 03 3E 40 CD 1A DD DA 13 03 CD D2 DD FE OD
0320 CA 13 03 21 66 03 1E 01 CD 34 CA D2 36 03 5E 23
0330 56 EB E9 C3 4D C1 CD 55 DD 21 23 DC CD D4 DA C3
0340 13 03 0C 44 45 4D 4F 20 52 4F 4D 20 49 4E 54 45
0350 52 46 41 43 45 20 41 4E 44 20 4D 41 54 48 20 50
0360 41 43 4B OD OD 00 04 5A 45 52 4F 8E 03 03 41 44
0370 44 9F 03 04 45 58 49 54 33 03 03 4E 45 47 BB 03

0380 03 53 51 52 CD 03 04 48 45 4C 50 24 04 00 CD D2
0390 DD FE OD C2 36 03 AF 47 4F 57 E7 12 C3 DF 03 CD
03A0 D2 DD CD OD DE D2 36 03 CD 1E C0 CD D2 DD FE OD
03B0 C2 36 03 21 8E 05 E7 00 C3 DF 03 CD D2 DD FE OD
03C0 C2 36 03 21 8E 05 E7 0C E7 1B C3 DF 03 CD D2 DD
03D0 FE OD C2 36 03 21 8E 05 E7 0C E7 33 C3 DF 03 21
03E0 8E 05 E7 0F CD 5E DD 21 OD 04 CD D4 DA CD 21 C0
03F0 3E 06 CD 30 C0 21 E3 00 CD 32 DB 21 19 04

03FE CD D4
0400 DA CD 2D C0 21 E3 00 CD 32 DB C3 13 03 56 41 4C
0410 55 45 20 44 45 43 20 3A 00 20 20 20 48 45 58 20
0420 3A 20 23 00 21 2D 04 CD D4 DA C3 13 03 OD 54 48
0430 49 53 20 44 45 4D 4F 20 41 4C 4C 4F 57 53 20 53
0440 4F 4D 45 20 4D 41 54 48 45 4D 41 54 49 43 20 46
0450 55 4E 43 54 49 4F 4E 53 20 4F 4E 20 41 20 56 41
0460 4C 55 45 20 53 41 56 45 44 OD

046A 49 4E 20 54 48 45
0470 20 4C 4F 43 41 54 49 4F 4E 20 4E 55 4D 42 45 52
0480 2E OD 41 46 54 45 52 20 45 41 43 48 20 46 55 4E
0490 43 54 49 4F 4E 20 54 48 45 20 56 41 4C 55 45 20
04A0 49 53 20 44 49 53 50 4C 41 59 45 44 20 49 4E 20
04B0 46 50 54 20 41 4E 44 20 48 45 58 20 OD 46 4F 52
04C0 4D 41 54 2E OD 56 41 4C 49 44 20 43 4F 4D 4D 41
04D0 4E 44 53 20 41 52 45 20 3A OD

04DA 20 20 5A 45 52 4F
04E0 20 3A 20 52 45 53 45 54 20 56 41 4C 55 45 20 54
04F0 4F 20 30 OD 20 20 41 44 44 20 58 58 58 2E 58 58
0500 20 3A 20 41 44 44 20 54 4F 20 56 41 4C 55 45 20
0510 28 50 4F 53 49 54 49 56 45 20 41 4E 44 20 53 54
0520 41 52 54 20 57 49 54 48 20 30 2E 2E 39 29 OD 20
0530 20 4E 45 47 20 20 3A 20 43 48 41 4E 47 45 20 53
0540 49 47 4E OD 20 20 53 51 52 20 20 3A 20 54 41 4B
0550 45 20 53 51 52 OD

0556 20 20 48 45 4C 50 20 3A 20 53
0560 48 4F 57 20 43 4F 4D 4D 41 4E 44 53 OD 20 20 45
0570 58 49 54 20 3A 20 52 45 54 55 52 4E 20 54 4F 20
0580 55 54 20 4F 52 20 42 41 53 49 43 OD OD 00

058E 00 00
0590 00 00

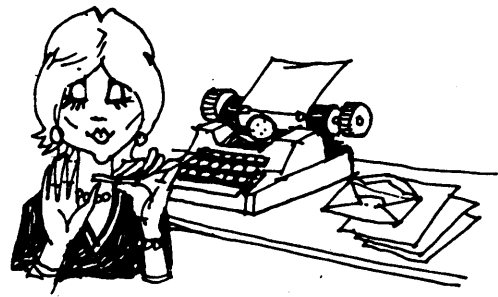
```

```

0 GETALCONVERSIES DAInamic 0-1 1980
0 4 OP EEN RIJ DAInamic 0-1 1980
0 RAKETSPEL DAInamic 0-1 1980
0 PROPELLER DAInamic 0-1 1980
0 4 COLOR DEMO DAInamic 0-1 1980
0 CITROEN B14 DAInamic 0-1 1980
0 BLOXY COLOR DEMO DAInamic 01 1980
0 MUSIC DAInamic 2 1980
0 REAL-TIME KLOK DAInamic 2 1980
0 DIGITAAL-ANALOOG KLOK DAInamic 2 1980
0 FORMAAT LISTING DAInamic 2 1980
0 LIST-PRINT DAInamic 2 1980
0 ZAKDOEK LEGGEN DAInamic 2 1980
0 BLUE MOON DAInamic 3 1981
0 TUBELAR BELLS DAInamic 3 1981
0 RADAR-SIMULATIE DAInamic 3 1981
0 VIDEORAM IN MODE 0 DAInamic 3 1981
0 VIDED TEXT DAInamic 3 1981
0 VARIABELEN ATLAS DAInamic 1981
0 BARRICADE DAInamic 3 1981
0 BEKLEURDE ACHTERGRONDEN IN MODE 0 DAInamic 3
0 KANTEN KLEEDJE DAInamic 4 1981
0 ALLE TEKENS DAInamic 4 1981
0 PRIEMGETALLEN 110 S DAInamic 4 1981
0 ROTERENDE ELLIPSEN DAInamic 4 1981
0 FILM TITLE SIMULATION DAInamic 4 1981
0 PADDELEN MET FGT DAInamic 5 1981
0 TIME OUT DAInamic 5 1981
0 RESPONTIE TIJD DAInamic 5 1981
0 REAL TIME CLOCK DAInamic 5 1981
0 RANDOM GENERATOR TEST DAInamic 5 1981
0 FASING KEYBORD MET BOOTSTRAP LOADER DAInamic 5
1FASING KEYBORD
0FASING KEYBORD
0 WILHELMUS DAInamic 6 1981
0 TORENS VAN HANDEI DAInamic 6 1981
0 VIJFDE GRAADS POLYNOMEN DAInamic 6 1981
0 8080 SIMULATOR BOOTSTRAP LOADER DAInamic 7 1981
1ML 8080 SIMULATOR
0BASIC 8080 SIMULATOR
0 BITRIJGENERAATOR DAInamic 7 1981
0 16 KLEUREN IN 4 KLEUREN MODE DAInamic 7 1981
0 TALK EDITOR DAInamic 7 1981
0 CIJFERTABEL DAInamic 7 1981
0 CARPENTERS MYSTERY DAInamic 8 1981
0 BASICODE MET BOOTSTRAPLOADER DAInamic 8 1981
1UTILITY BASICODE 2EE 527
0DOCUMENTATIE BASICODE
0 DIGITALE CLOCK DAInamic 8 1981
0 MEMORY-MAP MODE 4 DAInamic 8 1981
0 THE HAT DAInamic 8 1981
0 POKE ACTION DAInamic 8 1981
0 LETTER DAInamic 8 1981
0 SIENTJE DAInamic 8 1981
0 DEMO DAInamic 8 1981
0 TELEX IN BASIC DAInamic 8 1981
0 KRAAN DAInamic 8 1981

```

the story of anna list



Little Anna List was a real character. Dynamic but a bit fixed in her ideas, she executed her daily routine without ever understanding its function or ever passing a remark. She lived in a software house, a multi-level structure comfortable but basic. At the back was a data field where she kept a pet ram tied up with a bit string.

One day, an operator in Fort Ranfow noticed that Anna had not returned any data. He had nothing in his program for that day, so he decided to give her a call. He took the DCE bus to Anna's address and was surprised to find the house surrounded by an array of bins fixed to the wall. The only entry point was an access port on the left. He entered and found Anna sitting at a symbol table just about to take a byte out of an apple.

He addresses Anna, "What are all those bins?"

"I like apples", she declared, "and those are the core dumps."

"I'm from I.B.M. (Internal Buffer Maintenance)" the operator commented, and continued "Why haven't you computed your data?"

Anna displayed anguish. "I can't read the code", she sobbed,

"I can only read ASCII and the formats are all in EBCDIC."

The operator's response was unprintable, he cursed and cursed (he was a real cursor). "If data returned then okay else I shall compile a report" he ended.

After this exit, Anna decided to call her friend Pascal Bull. She called his linenummer and he told her to go to his address. Anna repeated what the I.B.M. man had said.

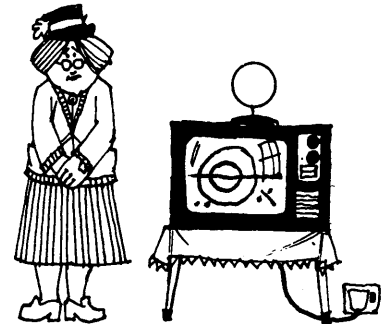
"Don't worry," Pascal said "no need to poke around; I'll get my brother Franco to take a peek at the problem."

The next day, Pascal gave Anna a conversion routine written by Franco Bull for translating EBCDIC into ASCII. Anna has compiled her data returns happily ever after.

Barry Hunt

testbeeld

```
100 REM TESTBEELD V1.0
110 REM ONTWERP A DOORNENBAL.
120 REM TEKENEN DUURT 2 MIN.
200 CLEAR 300+1400
210 GOSUB 1000:GOSUB 2000:GOSUB 3000:GOSUB 5000
220 REM GOSUB 3000 WORDT GOSUB 4000 VOOR LEGE CIRKEL
300 A$="TESTBEELD":X!=130.0:Y!=190.0:DEV!=0.0:C!=0.0
310 F!=1.0:GOSUB 40000
400 ENVELOPE 0 15,10;0,10;;SOUND 0 0 15 0 2000
999 GOTO 999
1000 REM BEELD OPMAAK
1005 COLORG 8 15 15 15
1010 MODE 5:POKE #BFFF,#B3:POKE #BFFB,#B2:POKE #BFF7,#B3:POKE #BFF3,#B3
1015 POKE #2A5,#AF:POKE #2A6,#5A
1020 FOR X=#BFEF-256*90 TO #BFEF-288*90 STEP -2
1030 POKE X,#FF:POKE X-1,#80:NEXT X
1040 FOR X=#BFEF-256*90 TO #BFEF-288*90 STEP -90
1050 POKE X,#A0:POKE X-1,#40:NEXT X
1060 RETURN
1799 DATA BLOK
2000 REM WITTE BLOKKEN OM BEELD
2010 DIM BLOK(43.0):REM CLEAR 300
2015 RESTORE
2016 READ A$:IF A$<>"BLOK" THEN 2016
2020 FOR X=0 TO 43:READ BLOK(X):NEXT X
2110 FOR LIJN=0 TO 7:GOSUB 2500:NEXT LIJN
2120 FOR LIJN=282 TO 288:GOSUB 2500:NEXT LIJN
2130 FOR LIJN=8 TO 280:A=#6F:B=#8F:C=#C0:D=#80
2140 IF ((LIJN-8)/21) MOD 2=1 THEN A=#90:B=#F8:C=#C0:D=#8F
2150 GOSUB 2600:NEXT LIJN:GOSUB 2600:RETURN
2500 Y=#BFEF-LIJN*90-2:FOR X=0 TO 43
2510 POKE Y-2*X,BLOK(X) IXOR #FF:POKE Y-2*X-1,#F:NEXT X
2520 RETURN
2600 Y=#BFEF-LIJN*90-2:POKE Y,A:POKE Y-1,B
2610 POKE Y-86,C:POKE Y-87,D:RETURN
2800 DATA #80,0,1,#FF,#FF,#F0,0,1,#FF,#FF,#F0,0
2801 DATA 1,#FF,#FF,#F0,0,1,#FF,#FF,#F0,0
2802 DATA 1,#FF,#FF,#F0,0,1,#FF,#FF,#F0,0
2803 DATA 1,#FF,#FF,#F0,0,1,#FF,#FF,#F0,0
2804 DATA 0,#3F
3000 REM RUITEN TESTBEELD
3010 FOR LIJN=8 TO 281:Y=#BFEF-LIJN*90-6
3020 FOR A=0 TO 7:POKE Y-A*10,#FE:POKE Y-A*10-1,#8F
3030 POKE Y-A*10-6,#EF:POKE Y-A*10-7,#8F:NEXT A
3040 POKE Y-A*10,#FE:POKE Y-A*10-1,#8F:NEXT LIJN
3050 FOR LIJN=8 TO 281 STEP 21:Y=#BFEF-90*LIJN-2
3060 FOR A=0 TO 43:POKE Y-A*2,#FF:POKE Y-A*2-1,#F8:NEXT A:NEXT LIJN
3070 RETURN
4000 REM CIRKEL
4010 X!=XMAX/2.0-2.0:D!=0.0:C!=0.0
4020 FOR A=0 TO 221:B!=20.0*SQR(221.0*A-A*A)/21.0
4030 IF ABS(B!-C!)<1.0 THEN NEXT A
4040 DRAW X!-C!,D! X!-B!,A 15:DRAW X!+C!,D! X!+B!,A 15
4050 C!=B!:D!=A:NEXT A:RETURN
5000 REM GEVULDE CIRKEL
5100 REM GEEL/ROOD BALK
5110 A=0:B=16:C=14:GOSUB 5900:FILL 155,0 175,16 3
5200 REM WIT REFLECTIE BALK
5210 A=17:B=37:C=15:GOSUB 5900:FILL 115,17 215,37 0
5220 DRAW 135,17 135,37 15
5300 REM GRIJS GRADATIES
5310 AB=38:AC=58:GOSUB 5320:AB=164:AC=184:GOSUB 5320:GOTO 5400
5320 C=0:FOR X=AB TO AC:A=X:B=X:GOSUB 5900
5330 DRAW 185,X X!+B!,X 15:NEXT X
5340 A=105:B=145:GOSUB 5950:FILL 145,AB 185,AC 8
5350 A=185:B=223.0:GOSUB 5950:RETURN
```



testbeeld

```
5400 REM FREQUENTIE BALKEN
5410 A=59:B=100:C=0:GOSUB 5900
5420 FOR X=85 TO 105 STEP 10:FILL X,59 X+4,100 15:NEXT X
5430 FOR X=117 TO 141 STEP 6:FILL X,59 X+2,100 15:NEXT X
5440 FOR X=149 TO 177 STEP 4:FILL X,59 X+1,100 15:NEXT X
5450 FOR X=181 TO 208 STEP 3:FILL X,59 X+1,100 15:NEXT X
5460 FOR X=213 TO 245 STEP 2:DRAW X,59 X,100 15:NEXT X
5500 REM KLEUREN BALKEN
5510 C=14:FOR X=122 TO 163:A=X:B=X:GOSUB 5900
5520 DRAW 185,X X!+B!,X 1:NEXT X
5530 FILL 95,122 130,163 13:FILL 130,122 165,163 12
5540 FILL 165,122 200,163 11:FILL 200,122 235,163 3
5600 REM HET MIDDEN VAN DE CIRKEL
5610 FILL 155,79 175,142 0:A=101:B=111:C=0:GOSUB 5900
5620 DRAW X!-B!,111 X!+B!,111 15:B=121:GOSUB 5900:DRAW 165,79 165,142 15
5630 FOR X=75 TO 275 STEP 20:DRAW X,101 X,121 15:NEXT X
5700 REM ZWART REFLECTIE BALK
5710 A=185:B=205:C=0:GOSUB 5900:FILL 115,185 215,205 15
5720 DRAW 135,185 135,205 0
5800 REM BOVENKANT CIRKEL & ROOD/GEEL BALK
5810 A=206:B=221:C=3:GOSUB 5900:FILL 155,206 175,221 14
5890 RETURN
5900 X!=XMAX/2.0-2.0:FOR A=A TO B:B!=20.0*SQR(221.0*A-A*A)/21.0
5910 DRAW X!-B!,A X!+B!,A C:NEXT A:RETURN
5950 FOR Y=AB TO AC:FOR X=A+Y MOD 2 TO B+Y MOD 2 STEP 2
5960 DOT X,Y B:NEXT X:NEXT Y:RETURN
40000 REM SLOW GRAF TEXT SGT
40001 REM ONTWERP A DOORNENBAL
40002 REM CLEAR 1400
40005 IF INIT!=1.0 THEN 40040:DIM CAR$(90.0)
40010 RESTORE
40012 READ B$:IF B$<>"SGT" THEN 40012
40015 FOR Z!=32.0 TO 91.0:READ B$
40025 IF B$="STOP" THEN INIT!=1.0:GOTO 40040
40030 READ CAR$(Z!):NEXT Z:INIT!=1.0
40040 CO!=COS(DEV!) *F!:SI!=SIN(DEV!) *F!:X!=X!+0.5:Y!=Y!+0.5
40042 FOR M!=0.0 TO LEN(A$)-1.0:GR$=CAR$(ASC(MID$(A$,M!),1))
40060 FOR N!=0.0 TO LEN(GR$)-1.0 STEP 4.0
40065 IF MID$(GR$,N!,1)="/" THEN X!=X!+CO!*8.0:Y!=Y!+SI!*8.0:NEXT M!:X!=X!-0.5:Y
!=Y!-0.5:RETURN
40080 JC1!=VAL(MID$(GR$,N!,1)):JC2!=VAL(MID$(GR$,N!+1,1))
40082 JC3!=VAL(MID$(GR$,N!+2,1)):JC4!=VAL(MID$(GR$,N!+3,1))
40090 DRAW X!+JC1!*CO!-JC2!*SI!,Y!+JC2!*CO!+JC1!*SI! X!+JC3!*CO!-JC4!*SI!,Y!+JC4
!*CO!+JC3!*SI! C!
40095 NEXT N!
40999 DATA SGT
41000 DATA BLANCO,/,UITROEP!,31313337/,QUOTES,25274547/,#
41001 DATA 1353155521274147/,$,124242532444152626563137/
41002 DATA %,17271626125641514252/,&,121321315331155116273536/, '
41003 DATA 3537/,(,131513311537/
41004 DATA ),315353555537/,*,125616523137/,+,32361454/,COMMA,21323233/
41005 DATA -,1454/,,31423241/,/,1256/,0,12162141525627471256/
41006 DATA 1,214131372637/,2,115112334444555647271627/,3
41007 DATA 122121415253345617574453/,4,414713531447/
41008 DATA 5,122121415254154515171757/,6,214112151444525315373757/,7
41009 DATA 21223561757/,8,2141244427471213151652535556/
41010 DATA 9,113131535356245415162747/,,33333535/,;,213232333535/,<
41011 DATA 14471441/
41012 DATA =,13531555/,>,21545427/,?,16272747343331313456/,APE, /
41013 DATA A,11155155135315373755/,B,111717471444114152535556/,C
41014 DATA 12162747445621414152/,D,1117114152561747/
41015 DATA E,1117115114441757/,F,111714441757/,G,12162757215151535343/,H
41016 DATA 111714545157/
41017 DATA I,214131372747/,J,122121415257/,K,111713572451/,L,11171151/
41018 DATA M,11171735353435575751/,N,111751571652/,O,1216274756522141/,P
41019 DATA 1117144417475556/
41020 DATA Q,12162747565321313351/,R,11171747565514442451/,S
41021 DATA 1221214152532444151627474756/,T,17573137/
41022 DATA U,111721415157/,V,1317535713313153/,W,11175157113333513334/,X
41023 DATA 111217165152575612561652/
41024 DATA Y,16175657163434563134/,Z,175712561151/
41025 DATA STOP
```

Naar aanleiding van enkele reacties op mijn bijdrage in DAI-namic jan/feb is gebleken dat enige uitleg over de getal-conversie routines welkom zou zijn. Op deze tape is zowel uitleg over het gebruik van de ROM routines bij het omzetten van ASCII input naar de Math-accumulator als de omgekeerde bewerking opgenomen. Mogelijk vindt u een en ander geschikt voor publikatie in een volgende DAI-namic. Met behulp van deze routines hoeft de assembler programmeur geen uitstapjes meer te maken naar BASIC voor zijn output.

Met vriendelijke groet,
 Peter Jongen
 Zeemanhof 25
 NL 2871JW SCHOONHOVEN
 tel. 01823-5170

ASCII-BINARY-ASCII

=====

If you want to use ASCII input for your M.L.-programs the DAI-encoding routines can be very usefull to you. Following an explanation how to convert ASCII input to internal binary format. There are 3 calls:

CALL :C01E Floating point input to MATH. ACCUMULATOR
 :C024 Integer input to MATH ACC.
 :C02A Hex input to MATH ACC.

If you call one of these routines they will need the ASCII input characters one at a time. Therefore a CALL will be made to a routine whose address is in RAM loc :D2/:D3. This is the pointer to the DAI-input-routine for expression encoding. You should supply a subroutine in your program that can put one (1) ASCII character in the A-reg on call. Store the address of your routine in :D2 before calling the conversion routine.

Conversion will stop if any non numeric (0-9) character is entered. (except for first: . E and - in case of FPT) The conversion-routine will add 1 for every character to the C-register. Register C is available on the call to your input routine.

EXAMPLE:

```
* INITIATE INPUT POINTER FOR EXPRESION ENCODING.
  LXI    H,INPUT      ADDR OF INPUT SUBR
  SHLD   :D2          STORE IT IN POINTER
  .
```

```
* CONVERT ASCII INPUT TO MATH.ACC
  MVI    C,0          CLEAR CHAR COUNTER
  CALL   :C01E        INPUT FPT NUMBER TO ACC
  .
```

On return the converted integer number is in the MATH-ACC.

```
* INPUT SUBROUTINE.
INPUT  MOV    A,C      CHAR COUNT
        CPI    10      ALLOW ONLY 10 CHAR.
        JZ     INRET   RETURN WITH ILLEGAL CHAR TO STOP
GETC   CALL   :6DBE    GETC A CHARACTER
        JZ     GETC    WAIT FOR IT
INRET  RET
```

If you already have the ASCII-number in a string in memory the input routine could look like:

ASCII-BINARY-ASCII

```

INPUT   PUSH   H
        PUSH   B
        LXI   H,STRING   GET FIRST ADDRESS OF STRING
        MVI   B,0        ADD C (OFFSET) TO ..
        DAD   B          HL TO GET NEXT ASCII DIGIT
        MOV   A,M        GET THE DIGIT IN A
        POP   B
        POP   H
        RET
STRING  ASC     '12.345000' SPACE TO STORE INPUT
NEXT    ASC     'S'       ANY NON-NUM. TO STOP CONV.

```

In above program conversion will stop if the 'S' is passed to the conversion routine.

All register except C preserved.

Note:

-The conversion routines will not recognize a leading '-' or '+' sign. For negative numbers use 'RST 4 DATA :60 or :1B
For FPT input 'E', '.' and '-' for (neg-)exponent are allowed. (e.g. 1.234E-5)

-If you want to return to BASIC after M.L. programs using above, be sure to save and restore :D2/D3

DAI-BINARY TO ASCII

=====

Following a description how to use the DAI-ROM-routines to convert the result in the MATH-ACC to ASCII for output.

On return from these CALL's you will find a string of characters starting in memory location :E4.

For both INT and FPT the string will have following format:

+.XXXXXXX or -.XXXXXXX X=ASCII numeric

As you can see the number is always <1 AND >-1.

The power of 10 to apply is stored at loc :F1 in binary

This string is not nice for output as it is. A second CALL is provided to pretty it up for printing.

```
CALL :C021 (XFBC)
```

=====

Convert FPT-number in MATH-ACC to ASCII-string.

On return:

```
String in :E4 - - - - - :EC
Format   + . X X X X X X X
```

X=ASCII-numeric.

String is always <1 and >-1 and 7 digits precision.

The power of 10 to apply in :F1.

EXAMPLE:

```
FPT 100.5 in ACC after conversion will look like:
+. 1 0 0 5 0 0 0 :F1=03
```

```
CALL :C027 (XIBC)
```

=====

Convert INT-number in MATH=ACC to ASCII-string.

On return:

```
String in :E4 and on, same format as above except
Number of digits is min 3 and max 10.
```

EXAMPLE:

```
INT 100 in ACC after conversion will look like:
+. 1 0 0 0 0 U U U :F1=03
U=unchanged
```

CALL :C030 (XPRTY)

ASCII-BINARY-ASCII

=====

Pretty-up FPT or INT string.

Before CALLing load the A-reg with the number of digits for precision. (max 7)

If you use this CALL for an INT number, load A-reg with content of :F1

ON RETURN:

A nice printable string is in memory lokation :E4 and on. The number of valid digits is in :E3.

CALL :C02D (XHBC)

=====

Convert MATH-ACC to ASCII-Hex for output.

ON RETURN:

String in :E4 - :EB (max 8 digits)

The number of vallid digits in :E3

EXAMPLE:

	CALL	:C027	CONVERT INTEGER FOR OUTPUT
	LDA	:F1	PRECISION
	CALL	:C030	PRETTY IT UP
	LXI	H,:C4	START OF STRING
	LDA	:C3	NUMBER OF DIGITS
	MOV	B,A	
PRLOOP	MOV	A,M	GET CHARACTER
	CALL	:D695	PRINT IT
	INX	H	
	DCR	B	MORE CHARACTERS
	JNZ	PRLOOP	LOOP FOR PRINTING

for FPT numbers in ACC first two lines will be:

CALL	:C021	CONVERT FPT
MVI	A,:7	SET PRECISION

GOOD LUCK

Peter Jongen. Zeemanhof 25. 2871JW SCHOONHOVEN. 01823-5170

electric energy

```
100 REM ONTLADING ELECTRIC ENERGY +++ SMIT +++
150 MODE 6
160 COLORG 0 0 5 0
170 DRAW 50,50 100,100 21:DRAW 100,100 100,80 21:DRAW 100,80 150,130 21
180 FILL 52,52 48,48 22:FILL 152,132 148,128 22
200 FOR A=1.0 TO RND(4.0)
205 COLORG 0 3 5 0
210 SOUND 1 0 15 0 FREQ(1200.0)
220 SOUND OFF
225 COLORG 0 0 5 0
230 WAIT TIME RND(6.0)
240 NEXT
250 GOTO 200
```

oscilloscope

```

001
002
003
004
005
006
007
008 0300 21D803
009 0303 015001
010 0306 3600
011 0308 23
012 0309 0B
013 030A 7B
014 030B B1
015 030C C20603
016
017 030F 21D803
018 0312 015001
019 0315 110000
020 0318 7B
021 0319 32CF03
022 031C 7A
023 031D 32D003
024 0320 7E
025 0321 32D103
026 0324 3E00
027 0326 32D703
028 0329 CD5103
029 032C C5
030 032D D5
031 032E E5
032 032F 3E00
033 0331 CDC6EB
034 0334 3ADB00
035 0337 E1
036 0338 D1
037 0339 C1
038 033A 32D103
039 033D 77
040 033E 3E01
041 0340 32D703
042 0343 CD5103
043 0346 23
044 0347 13
045 0348 0B
046 0349 7B
047 034A B1
048 034B C21803
049 034E C30F03
050
051
052
053

*
* OSCILLOSCOPE
* LE SIGNAL EST ENTRE PAR LE PDL (0)
*
START ORG :300
* START FROM BASIC WITH : MODE 6 :CALLM#300
* CLEAR TABLE
      LXI H, TABLE
      LXI B, 336
CLEAR MVI M, 0
      INX H
      DCX B
      MOV A, B
      ORA C
      JNZ CLEAR

*
RESTAR LXI H, TABLE
      LXI B, 336
      LXI D, 0
OSCIL MOV A, E
      STA X
      MOV A, D
      STA X+1
      MOV A, M
      STA Y
      MVI A, 0
      STA COL
      CALL DOT
      PUSH B
      PUSH D
      PUSH H
      MVI A, 0
      CALL :EBC6
      LDA :DB
      POP H
      POP D
      POP B
      STA Y
      MOV M, A
      MVI A, 1
      STA COL
      CALL DOT
      INX H
      INX D
      DCX B
      MOV A, B
      ORA C
      JNZ OSCIL
      JMP RESTAR

* NOMBRE DE POINTS
* COMPTEUR X

* ANCIEN X

* EFFACE L'ANCIEN POINT

* PDL (0)
* VALEUR DU PDL (0)

* NOUVELLE VALEUR DE Y

* ALLUME LE NOUVEAU POINT
* INCREMENTE LA TABLE
* INCREMENTE X
* DECREMENTE LE COMPTEUR X

* POINT SUIVANT

*****
* SOUS-ROUTINE DE DESSIN D'UN POINT
*****
* PLACE MEMOIRE = #6644+(Y**5A)-(X/4)IAND#7E (3E

```


oscilloscope

054	0351	C5	DOT	PUSH	B	
055	0352	D5		PUSH	D	
056	0353	E5		PUSH	H	
057	0354	3ACF03		LDA	X	
058	0357	32D303		STA	XAT	
059	035A	3AD003		LDA	X+1	
060	035D	32D403		STA	XAT+1	
061	0360	3AD103		LDA	Y	
062	0363	115A00		LXI	D, :5A	
063	0366	210000		LXI	H, :0000	
064			*			
065	0369	37	BCL2	STC		* MULTIPLIE A*D>HL
066	036A	3F		CMC		
067	036B	1F		RAR		
068	036C	D27003		JNC	BCL1	
069	036F	19		DAD	D	
070	0370	B7	BCL1	ORA	A	
071	0371	CA7A03		JZ	CONT1	
072	0374	EB		XCHG		
073	0375	29		DAD	H	
074	0376	EB		XCHG		
075	0377	C36903		JMP	BCL2	
076			*			
077	037A	114466	CONT1	LXI	D, :6644	* PREMIERE PLACE MEMOIRE 6
078	037D	19		DAD	D	* HL=(YAT*#5A)+#6644
079	037E	C5		PUSH	B	
080	037F	0602		MVI	B,2	
081	0381	37	LOPI	STC		
082	0382	3F		CMC		
083	0383	3AD403		LDA	XAT+1	
084	0386	1F		RAR		
085	0387	32D403		STA	XAT+1	
086	038A	3AD303		LDA	XAT	
087	038D	1F		RAR		
088	038E	32D303		STA	XAT	
089	0391	05		DCR	B	
090	0392	C28103		JNZ	LOPI	
091	0395	C1		POP	B	
092	0396	E67E		ANI	:7E	* X=X SHR 2 IAND #7E
093	0398	47		MOV	B,A	
094	0399	7D		MOV	A,L	
095	039A	90		SUB	B	
096	039B	6F		MOV	L,A	
097	039C	7C		MOV	A,H	
098	039D	DE00		SBI	:00	
099	039F	67		MOV	H,A	* DANS HL: POSITION MEMOIRE
100	03A0	3ACF03		LDA	X	
101	03A3	E607		ANI	:07	* POSITION DU POINT
102	03A5	47		MOV	B,A	
103	03A6	3E00		MVI	A, :00	
104	03A8	04		INR	B	
105	03A9	37		STC		* CARRY=1
106	03AA	1F	FINPT	RAR		
107	03AB	05		DCR	B	
108	03AC	C2AA03		JNZ	FINPT	
109	03AF	47		MOV	B,A	* DANS B MASQUE 00100000
110	03B0	EEFF		XRI	:FF	
111	03B2	4F		MOV	C,A	* DANS C MASQUE 11011111
112	03B3	1602		MVI	D, :2	
113	03B5	3AD703		LDA	COL	
114	03B8	5F		MOV	E,A	
115	03B9	1F	ROT	RAR		
116	03BA	5F		MOV	E,A	

oscilloscope

```

117 03BB 7E          MOV    A,M
118 03BC DAC303      JC     SET
119 03BF A1          ANA   C          * EFFACE UN POINT
120 03C0 C3C403      JMP   CONTI
121 03C3 B0          SET   ORA   B          * DESSINE UN POINT
122 03C4 77          CONTI MOV   M,A          * DESSINE LE POINT EN MEM
123 03C5 7B          MOV   A,E
124 03C6 23          INX  H
125 03C7 15          DCR  D
126 03C8 C2B903      JNZ  ROT
127 03CB E1          POP  H
128 03CC D1          POP  D
129 03CD C1          POP  B
130 03CE C9          RET
131 03CF 0000        X     DBL   :0000
132 03D1 0000        Y     DBL   :0000
133 03D3 0000        XAT  DBL   :0000          * X POUR ROUTINE DOT
134 03D5 0000        YAT  DBL   :0000          * Y POUR ROUTINE DOT
135 03D7 00          COL  DATA :00
136 03D8 0000        TABLE DBL   :0000
137                   FIN   ORG   TABLE+337
138 0529             END

```

```

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

```

```

BCL1  0370  BCL2  0369  CLEAR  0306  COL    03D7
CONT1  037A  CONTI  03C4  DOT    0351  FIN    03DA
FINPT  03AA  LOPI  03B1  OSCIL  0318  RESTAR 030F
ROT    03B9  SET   03C3  START  0000  TABLE 03D8
X      03CF  XAT  03D3  Y      03D1  YAT    03D5

```

```

5      REM +++ MANDALA ....who is the author ? +++
10     T=9.0:MODE 6:COLORG 3 5 9 14
30     GOSUB 400:MODE 4:GOSUB 400:MODE 2:GOSUB 400:GOTO 10
35     END:REM -----
40     FOR S%=0 TO 20
41     L=RND(16.0):M=RND(16.0):N=RND(16.0):O=RND(16.0)
42     COLORG L M N O
44     FOR A=0.0 TO 700.0:K=GETC:IF K=0.0 THEN NEXT A
50     IF K=32.0 THEN GOSUB 202
55     IF K>0.0 AND K<>32.0 THEN GOSUB 200
60     NEXT S%:RETURN
90     REM -----
200    L=9.0:M=L+1.0:N=L+3.0:O=L+2.0
202    FOR X=0.0 TO 20.0
205    COLORG L M N O:WAIT TIME T
210    COLORG O L M N:WAIT TIME T
220    COLORG N O L M:WAIT TIME T
225    COLORG M N O L:WAIT TIME T:NEXT
230    RETURN
390    REM -----
400    D=YMAX:E=D/2.0:F=E+0.5:G=F/4.0:H=(YMAX+1.0)/8.0
470    M=INT(H*4.0-1.0):FOR X=0.0 TO M:MD=INT(XMAX/2.0)
500    DRAW MD+X,0 MD-X,M*2 20+X MOD 4
505    DRAW MD-X,0 MD+X,M*2 20+X MOD 4
510    DRAW (MD-M),M+X MD+M,M-X 20+X MOD 4
515    DRAW (MD-M),M-X MD+M,M+X 20+X MOD 4
520    NEXT X:GOSUB 40:RETURN

```

mandala

For these modifications you need:

- A good soldering iron, preferably temperature controlled, around 40W.
A soldering pistol won't do because you damage both print and components.
- A sharp knife.
- Some thin, isolated wire.
- Some solder, pliers and cutters.
- A resistor of 33K (orange/orange/orange).
- A capacitor of 1n5 polyester.
- A silicium diode 1N4148 or 1N914 (the yellow bar is the cathode).

These modifications are to be made on the main board rev.3, rev.4 or rev.5 only. Some soldering experience is a necessity. We are not responsible for damage on your machine because of bad soldering or improperly executing these modifications.

1. 16 color characters in mode 0

There is one cut to be made on the solder side of the board. It is on the track coming from pin 1 of the 74LS257 next to the ZNA134 to a plated through hole below this chip. Cut it next to the hole.

Then solder 3 wires under the chip on the left of the former one.

- From pin 3 to pin 10.
- From pin 7 to pin 11.
- From pin 9 to pin 1 of the first chip (where you cutted the track).

2. Picture stabilisation

Again there is a cut to be made on the solder side of the board. It is on the big track underneath the ZNA134. Cut it between pin 15 and the plated through hole in the middle of the chip.

The components are soldered on the parts side of the board.

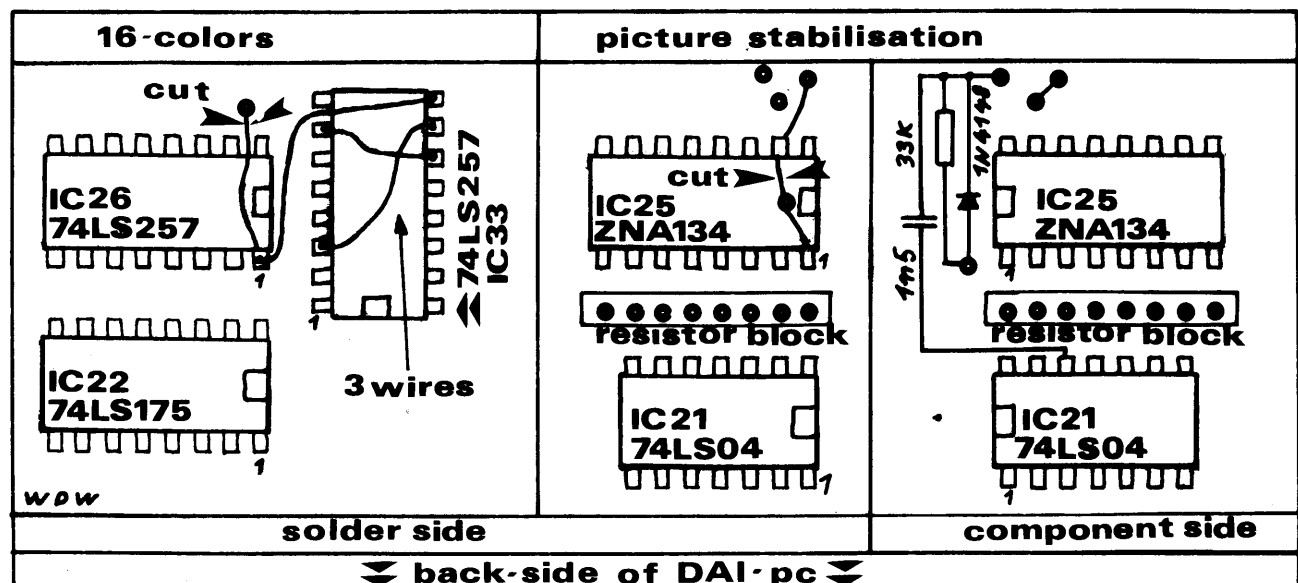
The capacitor is soldered from pin 12 of the 74LS04 above the ZNA134 to the plated through hole below the ZNA134 (the big hole on the right of the link).

Both the resistor and the diode are soldered from pin 1 of the ZNA134 to the hole mentioned before. The cathode of the diode must be on the bottom side of the board.

Now you can turn on your machine again. The jumping of your picture must be gone now. A 16 color characters demo program is available.

For further information please contact the club, they will give you my address or phone number.

Good luck.

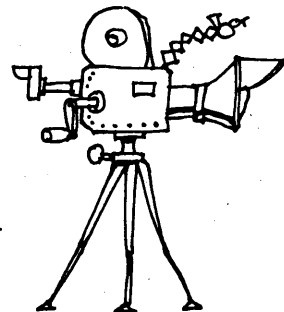


primzahlen

```
T
10 REM PRIMZAHLEN VON 1 BIS 1024 IN 10 SEKUNDEN
12 REM A. MEYSTRE , CH-4132 MUTTENZ / APRIL 81
15 CLEAR (2000):DIM S(255.0):S=VARPTR(S(0.0)):GOTO 100
20 M=1023.0:E=S+M:T=0:U=1:Z=2
25 A=#1010101:B=#1010001:C=#10101:D=#101:P=0
30 FOR I=1 TO 17:S(P)=C:S(P+1)=B:S(P+2)=B:S(P+3)=C:S(P+4)=D:S(P+5)=B:S(P+6)=A
34 S(P+7)=D:S(P+8)=A:S(P+9)=C:S(P+10)=D:S(P+11)=B:S(P+12)=C:S(P+13)=C:S(P+14)
=B
35 P=P+15:NEXT:S(255)=C
40 PRINT CHR$(12),2,3,5,:A=S+4
50 FOR N=7 TO M STEP Z:A=A+Z:IF PEEK(A)<>T THEN NEXT:SOUND 1 0 15 0 FREQ(750)
:GOTO 100
70 PRINT N,:IF N<32 THEN FOR I=A TO E STEP N+N:POKE I,U:NEXT
90 NEXT
100 WAIT TIME 20:SOUND OFF :WAIT TIME 200:GOTO 2200
1000 REM GRAPHISCHE DARSTELLUNG DER PRIMZAHLENGENERIERUNG
1010 REM A. MEYSTRE, CH-4132 MUTTENZ / FEB. 82
1015 F1=0:F2=5:F3=8:F4=15
1020 COLORG F1 F2 F3 F4:MODE 2A:MODE 2A:POKE #75,#20
1024 PRINT CHR$(12);TAB(4);"GRAPHISCHE DARSTELLUNG DER PRIMZAHLENGENERIERUNG"
1025 PRINT TAB(4);"NACH DER METHODE VON ERATOSTHENES (275-194 v. Chr.)"
1026 PRINT TAB(24);CHR$(#5F);CHR$(#5F);CHR$(#5F);CHR$(#5F);CHR$(#5F)
1028 PRINT TAB(4);"SETZEN VON 2 BIS ";CHR$(142);CHR$(143);" 1024";
1029 PRINT TAB(38);"PRIMZAHL :";
1030 FILL 16,8 55,47 F3:FILL 20,12 51,43 F2:DOT 20,12 F1:DOT 21,12 F1
1040 FOR I=2.0 TO 32:J=I:GOSUB 2000:IF SCRNX(X,Y)<>F2 THEN 1100
1050 CURSOR 48,0:PRINT I;
1060 FOR L=I+1 TO 1023 STEP I:J=L:GOSUB 2000
1065 IF SCRNX(X,Y)=F2 AND I>3 THEN SOUND 1 0 15 0 FREQ(750.0):DOT X,Y F4:WAIT TI
ME 10
1070 DOT X,Y F4:WAIT TIME SQR(I)+1.0:DOT X,Y F1:SOUND OFF :NEXT
1100 NEXT:GOTO 2200
1110 K=GETC:WAIT TIME 5:IF K=0 THEN 1110
1120 IF K=ASC("G") THEN 1000
1125 IF K=ASC("Z") THEN 20
1130 IF K=ASC("P") AND F4>0 THEN GOSUB 2100:GOTO 2200
1135 IF K=ASC("P") AND F4=0 THEN PRINT :PRINT "ZUERST GRAPHIK ERSTELLEN !!!":GO
TO 1110
1140 IF K=ASC("E") THEN POKE #75,#5F:END
1150 GOTO 1110
2000 X=J MOD 32:Y=J/32:X=X+20:Y=Y+12:RETURN
2100 PRINT CHR$(12);TAB(23);"DIE PRIMZAHLEN"
2130 FOR J=2 TO 1023:GOSUB 2000:IF SCRNX(X,Y)=F1 THEN NEXT:RETURN
2140 CURSOR 28,1:PRINT J:DOT X,Y F4:INPUT Z$:IF Z$<>" " THEN 2200:DOT X,Y F2:NEX
T:RETURN
2200 PRINT CHR$(12);"TASTEN-FUNKTION: G = GRAPHISCHE DARSTELLUNG"
2210 PRINT " P = PRIMZAHLEN AUSGEBEN (VIA RETURN-TASTE)"
2220 PRINT " Z = ALLE PRIMZAHLEN BIS 1021 IN 10s"
2230 PRINT " E = ENDE";:CURSOR 1,1
2290 GOTO 1110
```

kaleidoskop

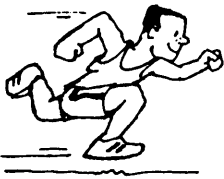
```
100 REM KALEIDOSKOP A. MEYSTRE / DEZ. 81
105 REM DERIVED FROM THE POPULAR APPLE II DEMO
110 MODE 5:COLORG 0 0 0 0:Y=YMAX/8
120 FOR W=3 TO 50:FOR I=1 TO Y:FOR J=0 TO Y
130 K=I+J
140 C=(J#3/(I+3)+I#W/12) MOD 24
150 XI=I#4+40:XK=K#4+40:XMI=YMAX-I#4+41:XMK=YMAX-K#4+41
160 YI=I#4:YK=K#4:YMI=YMAX-I#4+1:YMK=YMAX-K#4+1
170 XI3=I#4+43:XK3=K#4+43:XMI3=YMAX-I#4+44:XMK3=YMAX-K#4+44
180 YI3=I#4+3:YK3=K#4+3:YMI3=YMAX-I#4+4:YMK3=YMAX-K#4+4
190 FILL XI,YK XI3,YK3 C:FILL XK,YI XK3,YI3 C
200 FILL XMI,YMK XMI3,YMK3 C:FILL XMK,YMI XMK3,YMI3 C
210 FILL XK,YMI XK3,YMI3 C:FILL XMI,YK XMI3,YK3 C
220 FILL XI,YMK XI3,YMK3 C:FILL XMK,YI XMK3,YI3 C
230 NEXT:NEXT:NEXT
240 GOTO 120
```



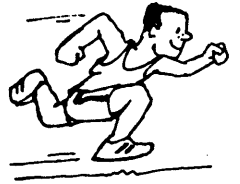
benchmarks

All Timings in seconds. All variables FPT

COMPUTER	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8
DAI with 9511	0.93	4.78	10.05	9.78	11.20	18.12	30.11	2.14
APPLE II	1.3	8.5	16.0	17.8	19.1	28.6	44.8	10.7
TRS 80	2.5	18.0	34.5	39.0	45.0	67.0	109.0	
COMPUCOLOR II	2.0	10.9	22.4	23.9	25.7	38.7	55.2	10.8
SORCEROR (EXIDY)	1.8	10.0	20.7	22.2	24.3	37.6	53.7	9.6
PET	1.7	9.9	18.4	20.4	21.0	32.5	50.9	12.3
SUPER PET	1.7	10.0	18.4	20.3	21.9	32.4	51.0	11.9
CHALLENGER C2-4P	1.4	7.8	15.0	16.5	17.8	27.0	39.5	7.9
ATARI 800	2.35	7.41	19.89	23.16	26.78	40.75	61.51	43.08
SUPERBRAIN	1.56	5.25	14.01	13.91	14.77	26.26	43.24	5.56
SHARP MZ-80K	1.4	9.4	16.3	22.5	25.4	36.8	51.1	10.2
SINCLAIR ZX81 (FAST MODE)	4.5	6.9	16.4	15.8	18.6	49.7	68.5	22.9
SINCLAIR ZX81 (SLOW MODE)	17.7	27.2	65.3	63.0	74.2	199.3	275.6	91.6
CROMEMCO Z-2D	2.0	5.8	16.4	19.1	20.1	31.5	42.5	23.0
COMMODORE VIC	1.4	8.3	15.5	17.1	18.3	27.2	42.7	9.9
TRS 80 COLOR COMPUTER	2.0	11.5	22.2	23.9	27.0	41.5	61.1	13.0
NEC PC-8001	1.7	8.3	18.1	17.8	18.6	29.5	49.2	7.0
TRS 80 MODEL 11 (S/PRECISION) Z80	1.0	5.0	13.0	13.0	14.0	23.0	35.0	6.0



OM I OP DE PISTE



Op 22 Mei j.l. werd in Eindhoven een scholen-sportdag gehouden, waarbij met groot succes een tweetal DAI-PC's werd ingezet. Er werd deelgenomen door twaalf zesde klassen van lagere scholen, waarbij jongens en meisjes elk een zestal individuele sportprestaties moesten leveren zoals 60 mtr hardlopen, kogelstoten, verspringen. Dus 24 groepen van gemiddeld vijftien kinderen. Vanuit het gemiddelde per groep werden punten voor die groep berekend met een omrekeningsformule per sport. Daarnaast werd in twee pools voetbal en kastie gespeeld. Ook van deze uitslagen werden punten toegekend, doelsaldi berekend etc.

Een DAI 48K zorgde voor het berekenen van de punten, punten totalen, overzicht van verwerkte uitslagen, overzichten van de poolstanden, en natuurlijk meteen aan het einde van de dag voor elke school een totaal-lijst.

Bovendien werd op geschikte momenten tijdens de dag een array gevuld met de schoolnamen en de bijbehorende tussenstanden, waarna een SAVEA. De tweede DAI stond met LOADA te wachten, zijn recorderingang was via een tussenversterkertje op de recorderuitgang van de eerste gekoppeld. Zodra de LOADA was ontvangen gaf deze PC de tussenstanden in kleur weer op een grootbeeld TV op het veld. En met het knipperlicht op de momenteel hoogste tussenstand. Dat bleef zo actief tot de volgende LOADA.

Waar vorige jaren vier mensen bezig waren met rekenen en lijsten kon nu met twee mensen, een om de binnenkomende uitslagenlijsten op te lezen en een tikker, veel meer, veel sneller en met overzicht tussen de inputruns gewerkt worden.

Bij het ontwikkelen van het programma bleek het rekenende gedeelte nauwelijks problemen te geven. Foutmeldingen tijdens de run, zowel vanuit het programma als door foutieve input, waren natuurlijk absoluut ontoelaatbaar. Het zal maar gebeuren dat in de loop van de middag, met 80% van de gegevens in het geheugen, een ERROR optreedt !! Aan beveiliging tegen inputfouten en een gedegen controle van het programma in alle mogelijke takken moest dus erg veel tijd besteed worden, en een regelmatige tussentijdse SAVEA van diverse arrays moest worden veilig gesteld.

Een paar weken vooraf werden per school, per groep (jongens/meisjes), per sport ook de alfabetische naamlijsten (144) geprint. Er namen in totaal 174 jongens en 195 meisjes deel, zodat het nu ook de moeite waard is statistisch nog wat gegevens te berekenen. Gemiddelde, spreiding, hoogste prestatie individueel, en vooral of de toegepaste puntenberekening nog op de werkelijke prestaties klopt.

Mocht er belangstelling bestaan voor dit soort programma's dan zullen we ze graag aanbieden.

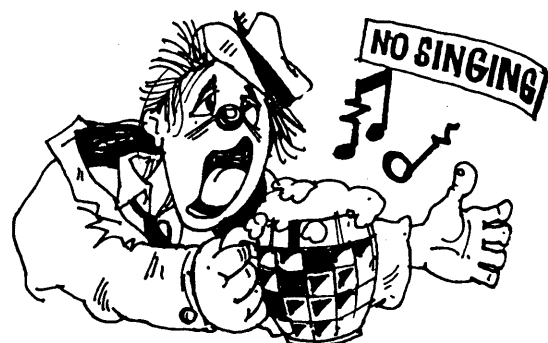
Met vriendelijkê groeten,

Hans Vollinga
Chopinlaan 44
5653 EW Eindhoven

Ger Esselink
Nederhoven 11
5655 BR Eindhoven

birthday song

```
5 REM +++ BIRTHDAY SONG - Barry Hunt - 4/3/82 +++
6 REM +++ The name of the person whose birthday +++
7 REM +++ is being celebrated should be substituted +++
8 REM +++ for "CLAUDIE" (the name of my wife) +++
9 REM +++ in the third line of the song +++
10 REM +++ and the third DATA statement should be +++
11 REM +++ changed accordingly +++
15 PRINT CHR$(12):POKE #75,32
20 CURSOR 20,20:PRINT "BIRTHDAY SONG"
30 CURSOR 20,19:FOR I%=1 TO 14:PRINT CHR$(11);:NEXT I%
40 CURSOR 16,16:PRINT "Press any key to start."
50 IF GETC=0.0 THEN 50
60 CURSOR 16,16:PRINT " "
70 FOR I%=3 TO 1 STEP -1
80 CURSOR 25,16:PRINT I%:WAIT TIME 30
90 NEXT I%
110 MODE 0:COLORT 8 0 8 12:PRINT CHR$(12)
120 ENVELOPE 0 2,4;10,6;15,20;12,30;8,40;4,50;0
125 REM ++ change text letter size
130 FOR ADDR%=#BFEF TO #BCCB STEP -#86
140 POKE ADDR%,#6A
150 NEXT ADDR%
160 PRINT
170 PRINT " HAPPY BIRTHDAY TO YOU"
180 PRINT " HAPPY BIRTHDAY TO YOU"
190 PRINT " HAPPY BIRTHDAY DEAR CLAUDIE"
200 PRINT " HAPPY BIRTHDAY TO YOU"
205 REM ++ set start of line and current position
210 START%=#BED2:POSN%=START%
220 FOR I%=1 TO 28
225 REM ++ read note frequency, note length and (length of syllable - 1)
230 READ NOTE%,NL%,CL%
235 REM ++ Note of zero means new line
240 IF NOTE%=0 THEN 380
250 SOUND 0 0 15 0 FREQ(NOTE%)
260 SOUND 0 0 8 0 FREQ(NOTE%*2.0)
270 SOUND 2 0 4 0 FREQ(NOTE%*4.0)
275 REM ++ change colour of syllable
280 FOR ADDR%=POSN% TO (POSN%-2*CL%) STEP -2
290 POKE ADDR%,#FF
300 NEXT ADDR%
310 WAIT TIME 8*NL%
320 SOUND OFF
325 REM ++ restore colour of syllable
330 FOR ADDR%=POSN% TO (POSN%-2*CL%) STEP -2
340 POKE ADDR%,#0
350 NEXT ADDR%
355 REM ++ update current position
360 POSN%=POSN%-2*CL%-2
370 GOTO 400
375 REM ++ new line
380 WAIT TIME NL%*8
390 START%=START%+#86:POSN%=START%
400 NEXT I%
410 DATA 264,1,2,264,1,3,297,2,4,264,2,3,352,2,3,330,4,3,0,2,0
420 DATA 264,1,2,264,1,3,297,2,4,264,2,3,396,2,3,352,4,3,0,2,0
430 DATA 264,1,2,264,1,3,528,2,4,440,2,3,352,2,5,330,2,4,297,2,2,0,0,0
440 DATA 467,1,2,467,1,3,440,2,4,352,2,3,396,2,3,352,4,3
450 POKE #75,95
500 RESTORE:GOTO 110
```

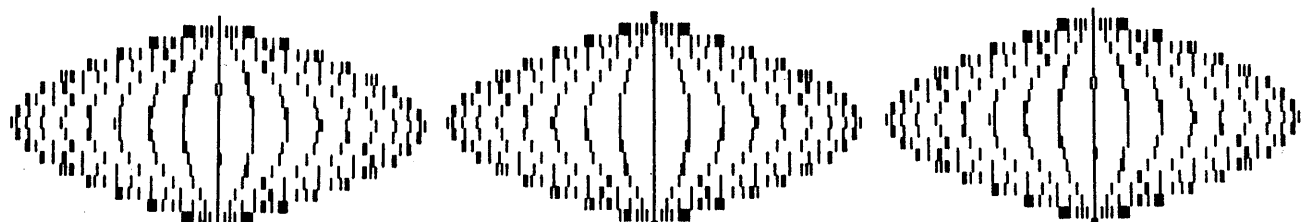
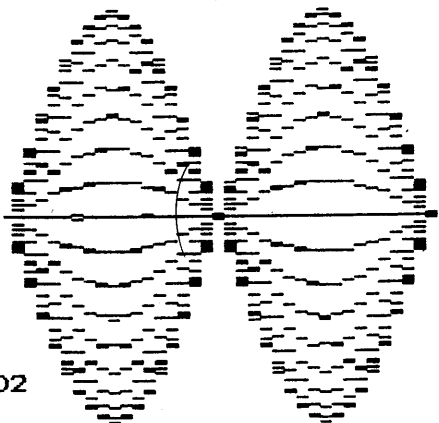


lopende & staande golven

```

20 CLEAR 200
30 COLORT 14 0 0 0
40 PRINT CHR$(12):PRINT
50 PRINT " Staande GOLVEN II - Natuurkunde ":PRINT
60 PRINT " Roelants J.":PRINT :PRINT
75 PRINT " Druk op de SPACE BAR om het bewegend beeld te bevroezen."
76 PRINT " Druk op de B-toets om een andere tijdvertraging "
77 PRINT
80 EE%=0
100 DATA #E5,#C5,#D5,#F5,#11,#00,#10,#0E,#24
140 DATA #06,#25,#D5,#CD,#00,#35,#36,#FF
180 DATA #CD,#00,#35,#23,#36,#FF,#CD,#00,#35
220 DATA #36,#FF,#23,#36,#FF,#05,#C2,#0C,#30
270 REM TIJDVERTRAGING
271 DATA #26,#FF,#2E,#80,#2D,#C2,#27,#30
272 DATA #CD,#BB,#D6,#B7,#FE,#42,#CA,#5B,#30,#FE,#20,#CA,#07,#35
273 DATA #25,#C2,#25,#30
300 REM
310 DATA #D1,#06,#25,#CD,#00,#35,#36,#00
340 DATA #CD,#00,#35,#23,#36,#00
360 DATA #CD,#00,#35,#36,#00,#05,#C2,#40,#30,#0D
410 DATA #C2,#09,#30,#C3,#04,#30
415 REM BASIC
420 DATA #D1,#F1,#D1,#C1,#E1,#C9,-1
490 REM SUB TEKEN
500 DATA #1A,#6F,#13,#1A,#67,#13,#C9
560 REM GETC
570 DATA #CD,#BB,#D6,#B7,#CA,#07,#35,#C3,#3D,#30,-1
800 X%=#3000
802 READ T%:IF T%<>(-1) THEN POKE X%,T%:X%=X%+1:GOTO 802
805 X%=#3500
806 READ T%:IF T%<>(-1) THEN POKE X%,T%:X%=X%+1:GOTO 806
810 INPUT " --> Tijdvertraging 1-255 ";V%:PRINT
812 IF V%>255 OR V%<1 THEN GOTO 810
815 POKE #3024,V%:POKE #3026,V%
900 INPUT " --> Berekening van de golven J/N";A%:PRINT
905 IF A%="J" GOTO 1000
910 IF A%<>"J" THEN 1500
920 PRINT " --> Eerste maal steeds berekenen ":GOTO 900
1000 REM CREEER DATA
1010 INPUT " --> Aantal golflengten per beeld 1...3 ";U:PRINT
1020 A=2.0*PI/36.0:F%=35:EE%=1
1030 PRINT CHR$(12):PRINT "--> Even wachten om de golf te berekenen .":PRINT
1040 J%=#1000
1060 FOR T%=0 TO 35:FOR X%=0 TO 36
1080 Y1%=42*SIN((U*X%-T%)*A):Y%=Y1%+42:GOSUB 2000
1100 Y2%=42*SIN((U*X%+T%)*A):Y%=Y2%+42:GOSUB 2000
1120 Y%=170+Y1%+Y2%:GOSUB 2000
1140 NEXT X%:CURSOR 30,2:PRINT " --> countdown ";F%-T%;" #":NEXT T%
1500 MODE 6:MODE 6
1510 COLORG 0 5 10 15:REM TRY OTHER VALUES..
1520 CALLM #3000
1540 GOTO 810
1990 END
2000 AD%=26098+2*(40-X%)+Y%*90
2020 LO%=AD% IAND 255:HI%=AD% SHR 8
2040 POKE J%,LO%:POKE J%+1,HI%:J%=J%+2
2060 RETURN

```



lopende & staande golven

LOPENDE EN STAANDE GOLVEN

Bespreking van het programma

Het programma demonsteert de mogelijkheid om zeer snel een aantal opeenvolgende beelden op het scherm te plaatsen om zo een bewegend effect te bekomen .

Daartoe worden vanuit een M.T. programma rechtstreeks een aantal bytes in de video RAM (MODE6 - start 26098) geplaatst .

Het BASIC programma berekent eerst de opeenvolgende beelden, en pookt de gevonden waarden weg in een tabel die start op adres :3000 en ongeveer:2000 bytes in beslag neemt . (

Het M.T. programma is geplaatst vanaf :5000 tot :5071 en kan via het S commando in UT. worden ingevoerd (SAVE via W5000 5071).

Om te beweging te vertragen is er een tijdlus ingebouwd . De SPACE BAR werkt als aan-uit schakelaar om het beeld te bevriezen of verder te laten lopen . Door de B-toets in te drukken keren we terug van de machinitaal subroutine naar het BASIC programma van waaruit we andere beelden kunnen berekenen of een andere tijdsvertraging kiezen .

De berekening van de lopende golven gebeurt op de regelnummers 1080 en 1100, de staande golf op 1120 .

Subroutine 2000 zorgt er telkens voor dat de coördinaten X% en Y% worden omgerekend naar hun overeenstemmend adres AD% in de VIDEO RAM. AD% wordt nu opgesplitst in LO% en HI% en weggepookt op de adressen J% en J%+1 in de tabel .

In totaal beschikken we hier over 36 beelden waarin drie figuren bestaande uit 37 bytes zijn opgeslagen .

Voeren we andere waarden in voor X% en Y% dan zal er een andere reeks van beelden ontstaan .

J. Roelants

DCR tape control

```
*LIST
1000 REM dcr tapecontrol
1001 REM ref. ward adriaens wilfried hermans inno broekman
1020 CLEAR 5000:CO:ORT 8 0 8 0
1040 PRINT CHR$(12):CURSOR 10,16:PRINT "D C R T A P E C O N T R O L"
1060 WAIT TIME 40:PRINT CHR$(12)
1100 PRINT CHR$(12):RESTORE
1120 READ N%:READ P%:IF P%="###" GOTO 1220
1140 IF N%<10.0 THEN PRINT " ";N%:">> " ;P%
1160 IF N%>9.0 THEN PRINT N%:">> " ;P%
1180 GOTO 1120
1220 REM wandelende cursor
1240 POKE #74,3:POKE #75,#20
1260 REM start keuze
1280 FOR X%=23 TO 0 STEP -1
1300 CURSOR 5,X%:PRINT CHR$(1)
1340 FOR Z%=1 TO 200
1360 IF GETC<>0 GOTO 1480
1380 NEXT Z%
1400 CURSOR 5,X%:PRINT CHR$(32)
1440 NEXT X%
1460 GOTO 1100
1480 CN%=24.0-X%
1500 RESTORE
1520 FOR I=1.0 TO CN%:READ N%:READ P%
1540 IF P%="###" THEN PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM kent
dit programma niet ! ";CHR$(1):WAIT TIME 200:GOTO 1100
1560 IF CN%=X% THEN GOTO 1100
1580 IF P%="geen programma" GOTO 1520
1590 IF CN%>22 THEN GOTO 1670
1600 NEXT I
1660 PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM is nu aan het zoeken naa
r : ";P%:" ";CHR$(1):GOTO 2000
1670 PRINT CHR$(12):CURSOR 0,2:PRINT CHR$(1);" MEMOCOM draait nu naar het einde
van de band. ";CHR$(1):GOTO 2070
2000 IF N%=1.0 THEN GOTO 2050
2005 IF CN%>22 THEN GOTO 2060
2010 FOR LOOP%=0 TO N%-2
2020 CALLM #F000:REM DCR0:SKIP1
2040 NEXT LOOP%
2050 LOAD
2070 CALLM #F000:REM DCR0:SKIP9:SKIP9:SKIP9
2080 END
3000 DATA 1,"DAMMEN"
3010 DATA 2,"SCHAKEN"
3020 DATA 3,"ROTERENDE PYRAMIDE"
3030 DATA 4,"CURSORTEKENEN"
3040 DATA 5,"BARRICADE"
3050 DATA 6,"VIER OP EEN RIJ"
3060 DATA 7,"RAKETSPEL"
3070 DATA 8,"TORENS VAN HANDI"
3080 DATA 9,"ACIGOL"
3090 DATA 10,"OMKEREN"
3100 DATA 11,"DOOLHOF"
3110 DATA 12,"SKIPISTE"
3120 DATA 13,"BLACKJACK"
3130 DATA 14,"ANDROIDENSPEL"
3140 DATA 15,"LANDMIJNEN"
3150 DATA 16,"geen programma"
3160 DATA 17,"geen programma"
3170 DATA 18,"geen programma"
3180 DATA 19,"geen programma"
3190 DATA 20,"geen programma"
3200 DATA 21,"geen programma"
3210 DATA 22,"geen programma"
3220 DATA 23,"naar bandeinde"
3230 DATA 24,"###"

100 REM +++ PHONE RINGING +++
120 ENVELOPE 0 10,2;15,2;14,4;12,5;8,10;0
130 ENVELOPE 0 15,10;0,10;15,10;0,10;0
150 SOUND 0 1 15 0 FREQ(1000.0)
160 SOUND 0 0 13 1 FREQ(2000.0)
170 SOUND 0 0 15 2 FREQ(5000.0)
1000 GOTO 120
```