

hac

OSI Gebruikersgroep

deel 12/13

feb. 1980

MOSI PRINTSERVICE-SPEECHSYSTHEMIZER-OSI MEM
ORY MAP-TEKSTTELEFOON-SOFTWARE PRINTBUFFER-
INITIALISATIE PIA-VERBETERINGEN MONITOR&TOO
LKIT-FULL SCREEN TEKSTEDITOR-LICHTKRANT 64
KARAKTERS-HANDLEIDING MINILOGOPROGRAMMA-LOW
COST GRAPHICS VOOR IEDER OSI-PLOTTERROUTINE
-INKEY\$ NABOOTSER-MERGEN ONDER OS65D 3.3-MA
CINETAAL ANALYSER-CHARACTERSET VOOR PLOTTER
-PLOTROUTINE-FORMUCALC-SCREEN EDITOR VOOR D
ISK ASSEMBLER-SMART TERMINAL HEXDUMP-GEBRUI
K FORTH ASSEMBLER - EXTENDED BASIC VOOR OSI

Voor vragen en/of opmerkingen:
John Hermans
Meerkoetlaan 5
2636 ER Schipluiden
01738 - 8703

Een uitgave van Hobby Computer Club Nederland
Postbus 149 - 2250 AC Voorschoten

Copyright HCC.

De O.S.I.-boekjes.

Nr.	Datum uitgave		
1	januari	1981	
2	mei	1981	
3	oktober	1981	
4	mei	1982	
1/2	september	1982	
5	september	1982	
6	november	1982	
7	februari	1983	
8	juni	1983	
9	oktober	1983	
10	februari	1984	
11	november	1984	
12/13	februari	1985	
HEESTERMAN REIN	OSI UITBREIDINGEN		AA001/198205
EKEREN KO VAN & MARTIEN SCHOT	HET MOSI SYSTEEM.		AA013/198302
HEESTERMAN REIN	EPROMS IN HET GEBRUIK		AA023/198306
EKEREN KO VAN & MARTIEN SCHOT	UITBREIDING MOSI SYSTEEM.		AA032/198310
HEESTERMAN REIN	EPROMS EN HUN POOTJES		AA035/198402
HEESTERMAN REIN	UITBREIDINGSBPRINT HCT001 VAN OSI CIP EN SUPERBOARD.		AA036/198402
RIJKEBOER A.W.	24 KRAM GEHEUGENUITBREIDING		AA047/198402
SCHOT MARTIEN & KO VAN EKEREN	MOSI PRINTSERVICE		AA049/198502 ●
EKEPEN KO VAN	SPEECHSYNTHESIZER		AA050/198502 ●
HERMANS JOHN	OVER MICROCOMPUTERS (VOORAL HETSUPERBOARD II)		AC001/198209
HELVIG TOM	WAIT WATCHER.		AC009/198211
COENEN HANS	PROGRAMMASTRUCTUREN IN BASIC ENFORTH		AC010/198302
			AC013/198306
STUYK C.A.	O.S.I. MEMORY MAP (STANDAARD)		AD001/198502 ●
BURGHOUTS JOS	ARTIKELN VOOR OSI-BEZITTERS INDE HCC NIEUWSBRIEF.		AE001/198202
GAALMAN B.M.A.	OSI PRINT SERVICE		AE003/198205
GRAAFMANS TOM	GOEDKOOP GROEN TV/MONITOR BEELDSCHERM		AE006/198205
VLAAR JAN	OSI-U6 SOFTWARE		AE007/198209
BURGHOUTS JOS	STEKERS VAN HET SUPERBOARD.		BA001/198202
BURGHOUTS JOS	INHOUDSOPSAVE PEEK(65).		AE011/198306
			198402
			198411
HEESTERMAN REIN	SOLDEREN.		AE017/198411
COHEN GEOFF. A.	BREAK-BEVEILIGING DOOR DUBBELETOETSDRUK.		BB001/198110
BURGHOUTS JOS	OMBOUW NAAR 600 EN 1200 BAUD.		BB003/198110
HINDERKS P.	HORIZONTALE BEELDVERSCUIJVING		BB004/198202
DOUWES RIJ	2 MHZ OMBOUW VOOR SUPERBOARD SERIE 2.		BB006/198202
SCHAEER HENK	VERPLAATSBAAAR RAM VOOR HET SUPERBOARD		BB007/198209
LIEKENS F.	ZERD SC ALS EENVOUDIGE I/O POORT.		BB010/198209
LANDMAN J.	AUTOMATISCH OMSCHAKELEN VAN 1 NAAR 2 MHZ MET DYNAMISCHE RAM		BB011/198302
NN	OMBOUW VAN SUPERBOARD 48 KAR. NAAR 48/64 KARAKTERS.		BB013/198306
RIJKEBOER A.W.	EXTRA GEHEUGEN VOOR OSI SUPERBOARD.		BB015/198310
GLASER JOHN	WIJZIGING KEYBOARD VOOR H.W.MONITOR		BB018/198310
LANDMAN J.	GEHEUGENUITBREIDING MET DYNAMISCHE RAMS (4116).		BC001/198110
LANDMAN J.	INPUT/OUTPUT-POORT VOOR OSI-SUPERBOARD.		BC006/198209
LOOYEN H.J. & A.G. MEGENS	SINGLE-STEP VOOR HET OSI-SUPERBOARD		BC008/198209
MOGELAND KEES	2716/2732 EPROM PROGRAMMER.		BO13/198211
LANDMAN J.	LIGHT PEN (MOD) #AARDVARK#.		BC020/198211

BIESEBOS J. VAN DE	SERVO-STURING	BC021/198211
HOOGELAND KEES	GEHEUGENUITBREIDING MET BANKS	BC023/198306
HOOGELAND KEES	REALTIME KLOK/KALENDER VOOR 6502 V1.0	BC029/198411
LUYPEN KEES VAN	WIJZIGING IN 'ZERO' EPROM PROGRAMMER.	BC038/198411
GLASER JOHN	TEKSTTELEFOON VOOR OSI.	BC039/198502
MEESTERMAN REIN	EEN SYSTEEMBUS VOOR OSI	BF001/198205
COENEN H.	CENTRONICS PARALLEL-INTERFACE VOOR OSI.	CB001/198205
ROMIJN C.	TELETYPE OP SUPERBOARD.	CC001/198205
SCHAER HENK	KRIJG OM OSI AAN DE PRAAT	CC005/198209
LUYPEN KEES VAN	UITBREIDING/VERANDERING AAN ELEKTUUR-FLOPPY-DISC-INTERFACE.	CC003/198302
HERMANS JOHN	HET 'DISK' OPERATING SYSTEM VAN OSI	CC012/198306
LUYPEN KEES VAN	EEN OPTISCHE KOPPELING TUSSEN OM SUPERBOARD EN OM PRINTER	CC014/198310
JONG HANS DE	SOFTWARE VOOR PRINTBUFFER	CC015/198502
SCHOT MARTIEN	INITIALISATIE VAN PIA	CC021/198502
GLASER JOHN	FSK MODULATOR	CD001/198302
GLASER JOHN	CONTRA VIDITEL MODEM.	CD002/198302
GLASER JOHN	DUAL D/A CONVERTER MET BUFFER	CF001/198302
TUBB PHIL	BINNENIN BASIC INTERPRETER.	DB001/198209
WEVERS HENK	OSI MICROSOFT BASIC IN ROM.	DB013/198209
MEIJERING TOM	BAS 4.3	DB022/198306
WEVERS HENK	OSI MICROSOFT BASIC IN ROM (VERVOLG).	DB033/198310
BURGHOUTS JOS	REGLPRINTER OSI SUPERBOARD	DB037/198310
BURGHOUTS JOS	CHARGET SUBROUTINE.	DB038/198310
BURGHOUTS JOS	SUBROUTINES UIT DE MONITOR.	DC001/198209
BURGHOUTS JOS	PAGE ZERO ADRESSEN.	DC022/198209
BURGHOUTS JOS	PAGINA 2 ADRESSEN	DC024/198209
SCHAER HENK	VERBETERINGEN AAN DE MONITOR EN TOOLKIT.	DC029/198502
OPPE SIEM	FULL SCREEN TEXTEDITOR.	DE001/198502
NN	BASIC SCHAAK.	EA017/198411
RADEMAKER JAN	SOLITAIRE.	EA001/198110
MATHEMER A.L.	SOLITAIRE.	EA005/198205
HERMANS JOHN	BREEK (EEN SPEL IN FORTH).	EA009/198110
SOMMEVELD E. & C. TOTTE	SPACE REACTION TEST.	EA012/198211
HERMANS JOHN	SATELLIETBAAN.	EA014/198211
DIVERSE AUTEURS	NUTTIGE BASIC SUBROUTINES	EB001/198110 198205 198211
HELWIG TOM	ON ERROR GOTO...	EB010/198402
NN	MORRIS/LOURASH-ROUTINE.	EB011/198402
BROECK JOHN VAN DEN	SUBROUTINE MET TRANSFER VAN X EN Y WAARDEN.	EB012/198402
HELWIG TOM	TREFWOORDEN.	EC001/198110
COENEN HANS	MACHINETAAL IN BASIC DATA-STATEMENTS.	EC004/198110
BROERS PETER	STRING VAN SCHERM.	EC007/198205
HERMANS JOHN	SCHETS EN SCROLL.	EC009/198205
HERMANS JOHN	PUZZLE-KUBUS.	EC011/198211
HERMANS JOHN	VLUCHTSIMULATOR.	EC015/198209
HERMANS JOHN	WEEFPATRONEN.	EC021/198211
COENEN HANS	SIMPLE GAME (FORTH).	EC022/198211
BAETEN M.	MINI-LOGO.	EC023/198310
LUIJSTERBURG LEO	HET PRINTEN VAN EIGEN KARAKTERS.	EC026/198310
GALLAND M.J.P.	FORTH KLOK.	EC027/198310

RIJKEBOER ROE	POEL HIGH RESOLUTION GRAPHICS 4 MB.	EC029/198411
SCHOT MARTIEN	LICHTKRANT VOOR 64 KARAKTER-SCHERM	EC037/198502 ●
BAETEN M.	GEBRUIKSAANWIJZING MINI-LOGO (PROGRAMMA PAG EC023 E.V.)	EC044/198502 ●
SCHAER HENK	PLOTTERROUTINE VOORSUPERBOARD.	EC054/198502 ●
COURBOIS JOS E.A.	HOBBYSCOOP BASICLOADER.	EE001/198205
EDIXHOVEN JAN EN BAS	QUICKSAVE.	EE005/198209
MATHLENER A.L.	BASIC DIT*SEMBLER.	EE019/198209
NN	MORSE TUTOR VERANDERINGEN IN (OSI-CASSETTE 4).	EE022/198209
GOERLITZ M. (J.VAN DER LINDEN)	6502 DISASSEMBLER.	EE023/198110
HERMANS JOHN	FORTH-HEXDUMP OP HET SCHERM.	EE031/198110
JONG HANS DE	OSI MINISPACE.	EE032/198211
MATHLENER A.L.	TEXT-SEARCHER.	EE033/198211
MATHLENER A.L.	HEX-SEARCHER.	EE034/198211
PAGSDALE WILLIAM F. (F16)	FORTH ASSEMBLER VOOR 6502.	EE035/198211
HELWIG TON	PROGRAMMA'S MET DE INSTRUCTIE WAIT.	EE038/198211
FORTH INTEREST GROUP	FORTH EDITOR.	EE039/198211
GLASER JOHN	X-Y RECORDER STURING MET D/A CONVERTERS.	EE044/198302
SCHAER HENK	MINIDATABASE VOOR SUPERBOARD.	EE045/198302
HOBBYSCOOP (NOS)	BASICODE 2.	EE049/198310
HELWIG TON	TITEL LOAD-OFF SAVE-OFF EN AUTOMATISCHE AUTO-RUN OP BAND.	EE055/198310
NN	BASIC SCHAAK.	EA017/198411
RIJKEBOER BOB	ROSI HIGH RESOLUTION GRAPHICS 1MBR.	EC029/198411
BARTEN H.A.	OS-65D ALLERLEI.	EE057/198310
SCHOT MARTIEN	HALTTOETS VOOR ASSEMBLER CASSETTE-VERSIE.	EE058/198310
HERMANS JOHN	'STRIPE SEARCH' DISK-FIND VOOR OS1-65D.	EE059/198310
WAL HENK DE	TEKSTPROCESSOR IN BASIC.	EE065/198402
WEVERS HENK (JOS BURGHOUTS)	HET SMART TERMINAL PROGRAMMA.	EE071/198402
HELWIG TON	6502 TRACER.	EE085/198411
BURGHOUTS JOS	SABR. UIT INTERFACE AGE MAY 1979	EE091/198411
HERMANS JOHN	TOETS-SCHRIJVER ONDER OS-65D.	EE095/198411
HEPMANS JOHN	WINDOW EEN PROGRAMMA VOOR SUPERBOARD ONDER OS-65D.	EE096/198411
WHITTLE MIKE (JOS BURGHOUTS)	LISTER VAN BASIC VARIABELEN	EE097/198411
WILLEMSEN GERARD	HULPROUTINE OM INKEY\$ NA TE BOOTSEN.	EE100/198502 ●
JONG HANS DE	HET MERGEN VAN BASIC PROGRAMMA'S ONDER OS65D 3.3	EE101/198502 ●
BURGHOUTS JOS	MACHINE-TAAL ANALYSER UIT ELEKTUUR	EE102/198502 ●
EKEPEN KO VAN	CHARACTERSET VOOR PLOTTER (BASIC).	EE110/198502 ●
EKEREN KO VAN	PLOT ROUTINE.	EE111/198502 ●
HERMANS JOHN	FORMULCALC	EE113/198502 ●
HEESTERMAN REIN	SCREEN EDITOR VOOR DE OS1 DISK ASSEMBLER.	EE119/198502 ●
BURGHOUTS JOS	EEN HEXDUMP VAN DE SMART-TERMINAL.	EE124/198502 ●
BURGHOUTS JOS	WAT KAN IK MET MACHINETAAL ?.	FA001/198209
HEPMANS JOHN	GEBRUIK VAN DE FORTH ASSEMBLER.VOOR DE 6502-MICROPROCESSOR	FA005/198502 ●
STRUYK COR & KO VAN EKEPEN	EXTENDED BASIC VOOR OS1	FB001/198502 ●
HEPMANS JOHN	FORTH (EEN INLEIDING).	FD001/198110
HEPMANS JOHN	DISK-LOZE FORTH.	FD002/198211
BURGHOUTS JOS & JOHN HERMANS	OPZET VAN EEN FORTH T-APE O-PERATING S-YSTEEM.	FD012/198302
BURGHOUTS JOS	FORTH INTRODUKTIE.	FD015/198306
BURGHOUTS JOS	HET LADEN EN BEWAPEN VAN FORTH-CREENS (TOS).	FD019/198310

OSI FOEL

FEBRUARI 1985

MOSI PRINTSERVICE

Het mosi team heeft inmiddels al heel wat printen beschikbaar daarom hier een lijst met wat op dit moment verkrijgbaar is.

- 1 videokaart 64 characters
- 2 processorinterfacekaart
- 3 geheugenkaart voor statische ram of eprom
- 4 dynamische ramkaart (elektuur) aangepast voor 64K
- 5 floppy interface (elektuur)
- 6 busprint (elektuur)
- 7 pkaart voor 4 pia's
- 8 epromprogrammer
- 9 realtimeclock (kees hoogeland)
- 10 keyboarddecoder
- 11 speechsynthesiser
- 12 stappenmotorsturing

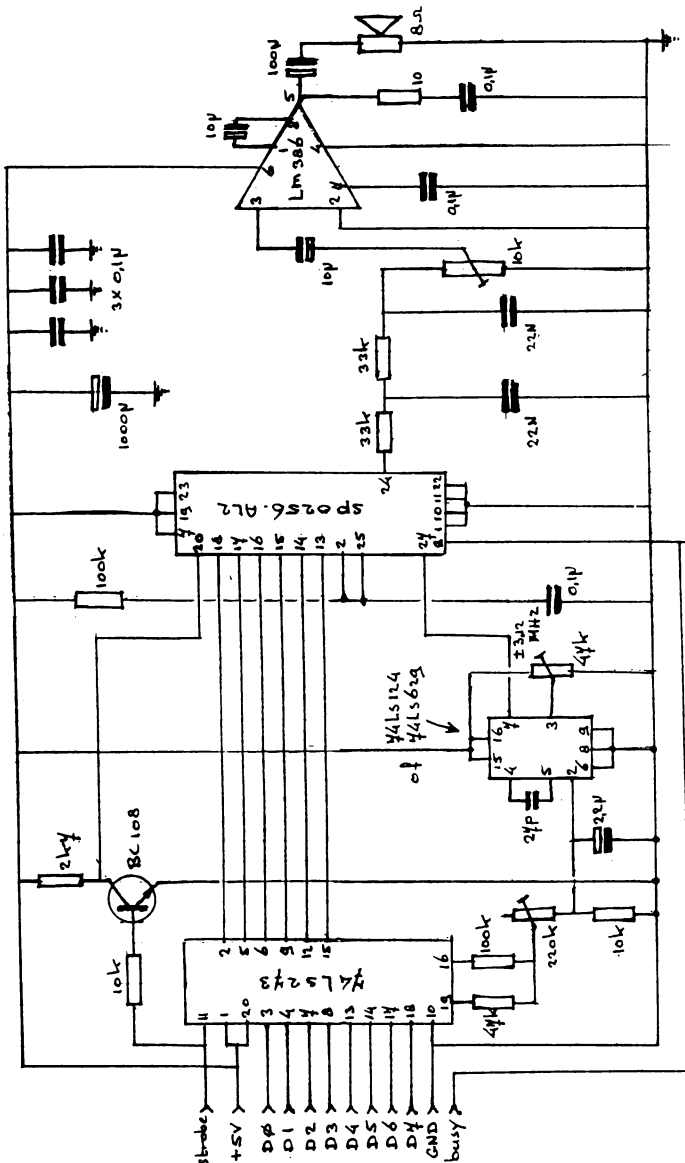
De printen 1 t/m 9 kosten fl.15,- de printen 10 t/m 12 kosten fl.7,50 per stuk en zijn niet geboord.

Uiteraard zijn bij alle printen de nodige schema's en componenten opstellingen.

Wat betreft de elektuur printen zijn de desbetreffende artikelen tegen copy kosten ook verkrijgbaar.

De printen zijn verkrijgbaar bij:

Ko v. Ekeren	Martien Schot
Reijerscop 26	Hofstede 21
3454 HX De Meern	3902 CG Veenendaal
tel. 03406-2767	tel. 08385-13219



Speech synthesiser

SPEECHSYNTHESISER

Deze speechsynthesiser is zeer universeel toepasbaar omdat hij aangesloten kan worden op een centronics printeraansluiting dus door simpel data naar de printeruitgang te sturen is het ding aan de praat te krijgen.

De synthesiser is opgebouwd rondom het General Instruments I.C. SP0256-AL2 deze is o.a. verkrijgbaar bij Tandy, let wel op dat het type AL2 is, er zijn namelijk meer typen in omloop. Omdat het i.c. alleen de datalijnen 0 t/m 5 gebruikt, worden de datalijnen 6 en 7 gebruikt om de stem frequentie softwarematig te beïnvloeden.

Voor een goede spraakweergave is het belangrijk een goed luidsprekertje in een degelijk kastje te gebruiken, een goed luidsprekertje is o.a. de Philips AD3371/Y4.

Er is ook een printje te verkrijgen voor de synthesiser
SUCCES Ko v.Ekeren

bron electronics & computing maart '84

5 REM EXAMPLE OF SPEECH SYNTHESIS WITH SP0256-AL2

6 REM by Ko v. Ekeren

20 RESTORE

25 REM HELLO

30 DATA 27,7,45,15,53,4

35 REM I AM A TALKING

40 DATA 6,3,26,16,3,20,3,13,23,23,2,42,12,44,3

45 REM COMPUTER

50 DATA 42,15,16,9,49,13,51,4,4

55 DATA 0,256

60 READI

65 REM beïnvloeding frequentie

70 I=I+64:REM mag 0,64,128 of 192 zijn

80 IF I>255 THEN I=0

90 POKE 61441,I:WAIT 61440,2

100 GOTO 60

110 END

ready

Code	Phonem		ms	Code	Phonem		ms
00	PA1	Pause	10	32	AW	Out	250
01	PA2	Pause	30	33	DD2	Do	250
02	PA3	Pause	509	34	GG3	Wig	120
03	PA4	Pause	100	35	VV	Vest	130
04	PA5	Pause	200	36	GG1	Guest	80
05	OY	Boy	290	37	SH	Ship	120
06	AY	Sky	170	38	ZH	Azure	130
07	EH	End	50	39	RR2	Brain	80
08	KK3	Comb	80	40	FF	Food	110
09	PP	Pow	150	41	KK2	Sky	140
10	JH	Dodge	100	42	KK1	can't	120
11	NN1	Thin	170	43	ZZ	Zoo	150
12	IH	Sit	50	44	NG	Anchor	200
13	TT2	To	100	45	LL	Lake	80
14	RR1	Rural	130	46	WW	Wool	140
15	AX	Succeed	50	47	XR	Repair	250
16	MM	Milk	180	48	WH	Whig	150
17	TT1	Part	80	49	YY1	Yes	90
18	DH1	They	140	50	CH	Church	150
19	IY	See	170	51	ER1	Fir	110
20	EY	Beige	200	52	ER2	Fir	210
21	DD1	Could	50	53	OW	Beau	170
22	UW1	To	60	54	DH2	They	180
23	AO	Aught	70	55	SS	Vest	60
24	AA	Hot	60	56	NN2	No	140
25	YY2	Yes	130	57	HH2	Hoe	130
26	AE	Hat	80	58	OR	Store	240
27	HH1	He	90	59	AR	Alarm	200
28	BB1	Busy	40	60	YR	Clear	250
29	TH	Thin	130	61	GG2	Got	80
30	UH	Book	70	62	EL	Saddle	140
31	UW2	Food	170	63	BB2	Busy	60

FEBRUARI 1985

OSI FOEL

FEBRUARI 1985

DE O.S.I. MEMORY MAP door C.A. STRUYK

Ondanks de komst van allerlei nieuwe systemen op de toch al druk bezette computermarkt, kunnen we toch wel gerust stellen dat de destijds alom geprezen OSI-SYSTEMEN (SUPERBOARD- en CHALLENGER typen) bepaald NIET dood zijn.....

Een regelmatige bezoeker van de OSI-gebruikersgroep bijeenkomsten in "de BRON" te Utrecht weet wel beter....

Getuigen van de activiteiten rondom de OSI-systemen zijn o.a. het ontstaan van allerlei afgeleide typen en varianten zoals ; MOSI, VLOSI, FROSI, ROSI, CLOSI, BRABOSI en anderen.....

Op het HARDWARE vlak hebben we kennis kunnen maken met diverse nuttige veranderingen en aanpassingen waaronder;

- De modificaties van de VIDEO-PRESENTATIE van 24,48 en 64 karakters per regel
- Het DISK OPERATING SYSTEM
- DIVERSE I/O kaarten en aanpassingen
- GEHEUGENKAARTEN en geheugen uitbreidingen
- De EPROM PROGRAMMER toepassing via de OSI
- HIGH RESOLUTION GRAPHICS (door enkele personen)
- En de onlangs geïntroduceerde REAL-TIME-CLOCK CARD

Op het SOFTWARE-vlak kennen we de modificaties en uitbreidingen van ;

- De MONITOR (typen 4.0 en 5.0)
- De TOOLKIT
- De EXTENDED DOS uitbreiding
- VIDITEL
- EPROM PROGRAMMER
- CEGMON-MONITOR (Onlangs opnieuw geïntroduceerd)
- EXTENDED BASIC (Onlangs geïntroduceerd)

Al met al kunnen we gerust stellen dat een groot aantal (x)-OSI bezitters druk in de weer zijn, (of geweest zijn), met nieuwe ontwikkelingen op HARDWARE- en SOFTWARE-gebied.

Het probleem dat echter dreigt te ontstaan is dat verschillende mensen Hardwareschakelingen en Software ontwikkelen die niet door iedereen gebruikt kunnen worden, doordat de door hen gebruikte geheugenplaatsen al voor andere doeleinden gebruikt zijn.....

Vandaar dan ook dat op de in November j.l. gehouden bijeenkomst van de OSI-gebruikersgroep dit onderwerp ter tafel werd gebracht, met de bedoeling om gezamenlijk te komen tot een GESTANDAARDISEERDE OSI MEMORY-MAP.

Gezien het feit dat al veel mensen de destijds in het OSI-boekje (Blz.BF1) geïntroduceerde ELEKTUUR-BUS als norm hebben aangenomen, en ook veel mensen een BUS-SYSTEEM a la het MOSI-systeem (met elektuur-bus) hebben gebouwd, lijkt het zinvol om een MEMORY-MAP opdeling op te zetten waarin deze zaken niet in conflict zijn.

Wanneer ik een opzet in die richting zou mogen voorstellen, zou deze, uitgaande van 2K-byte gebieden, er als volgt uit gaan zien.

totaal K-bytes	GEBIED (hex)	In gebruik voor :
2	0000 - 07FF	RAM
4	0800 - 0FFF	RAM
6	1000 - 17FF	RAM
8	1800 - 1FFF	RAM
10	2000 - 27FF	RAM
12	2800 - 2FFF	RAM
14	3000 - 37FF	RAM
16	3800 - 3FFF	RAM
18	4000 - 47FF	RAM
20	4800 - 4FFF	RAM
22	5000 - 57FF	RAM
24	5800 - 5FFF	RAM
26	6000 - 67FF	RAM
28	6800 - 6FFF	RAM
30	7000 - 77FF	RAM
32	7800 - 7FFF	RAM
34	8000 - 87FF	RAM
36	8800 - 8FFF	RAM
38	9000 - 97FF	EXTENDED BASIC V.2
40	9800 - 9FFF	EXTENDED BASIC V.2 (Uitbreidingen)
42	A000 - A7FF	BASIC (1)
44	A800 - AFFF	BASIC (2)
46	B000 - B7FF	BASIC (3)
48	B800 - BFFF	BASIC (4)
50	C000 - C7FF	DIVERSEN zie OPMERKINGEN
52	C800 - CFFF	EPROM EXTENDED DOS 3.3
54	D000 - D7FF	VIDEO RAM
56	D800 - DFFF	DIVERSEN zie OPMERKINGEN
58	E000 - E7FF	EPROM TOOLKIT en EXT.MONITOR
60	E800 - EFFF	EPROM TOOLKIT en EXT.MONITOR
62	F000 - F7FF	DIVERSEN zie OPMERKINGEN
64	F800 - FFFF	EPROM MONITOR (V.4 of V.5)

OPMERKINGEN OMTRENT ENKELE GEBIEDEN

- 9000 - 97FF : Bestemd voor de NIEUWE EXTENDED BASIC toevoeging, zoals gedemonstreerd op enkele OSI bijeenkomsten.
Bij geen gebruikmaking hiervan evt. 2K extra RAM, of andere zaken.
- 9800 - 9FFF : Bestemd voor TOEKOMSTIGE uitbreiding van EXTENDED BASIC. oa. Functietoetsen en andere routines.
Bij geen gebruikmaking hiervan evt. 2K extra RAM, of andere zaken.
- C000 - C7FF : C000 DISK CONTROL ADRES
C110 - C11F MOSI I/O kaart (zie OSI boekje blz.AA32)
C400 - C750 NIEUW REAL-TIME-CLOCK waarbij deels ook het gebied C200 - C21F in gebruik is.

D800 - DFFF : D800 - DFFF EPROM VIDITEL (MOSI)
DF00 - DF01 Keyboard (C1P en MOSI) *** (zie opm.)

F000 - F7FF : F000 - F001 ACIA (C1P en MOSI)
F002 - F7FF EPROM PROGRAMMER software (Mosi)
F700 PIA (C2P)

*** Opmerking

Uiteraard dienen de extra toepassingen in de gebieden; C000-C7FF, D800-DFFF en F000-F7FF GOED UITGEDECODEERD te worden. Vooral bij toepassing op de SUPERBOARD-systemen dient men er goed rekening mee te houden dat niet alles even selectief uitgedecodeerd is.

Zie hiervoor ook het artikel van Rein Heestermans (OSI boekje blz. AA1 t/m AA12.)

Vooraf voor toekomstige HARDWARE- en SOFTWARE uitbreidingen is het van belang dat wij (diegenen die de OSI voor uitsterven behoeden) onderling goede afspraken maken omtrent de te gebruiken geheugenplaatsen in de MEMORY-MAP.

Gaarne ben ik bereid om uw eventuele reacties, suggesties op en aanmerkingen in ontvangst te nemen zodat we gezamenlijk tot een eensluidende opzet kunnen komen.

OSI-MEMORY-MAP MET ELKAAR...VOOR ELKAAR

Met vriendelijke OSI groet,

Cor Struyk, Geertruidenberg

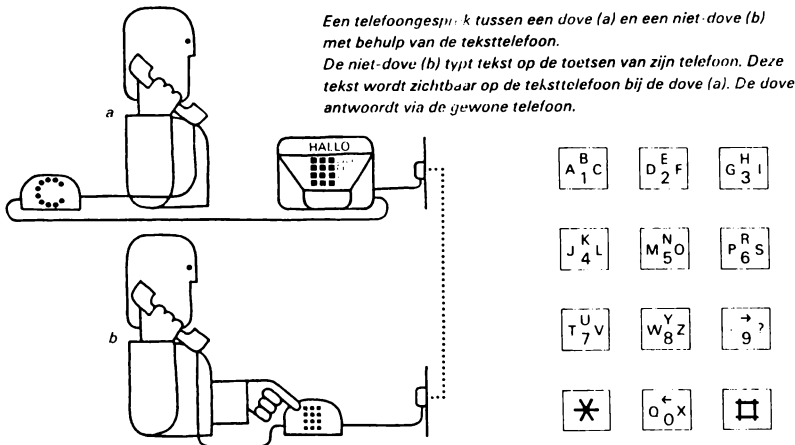
OSI FOEL

FEBRUARI 1985

Wat is en hoe werkt de teksttelefoon?

Het teksttelefoontoestel maakt het een dove mogelijk om via het telefoonnet met anderen (doven en niet-doven) te communiceren. De dove moet dan thuis een gewone telefoontoestel en een teksttelefoon hebben. De teksttelefoon bevat een tekstscherm waarop 8 letter- of cijfertekens zichtbaar gemaakt kunnen worden. De tekst schuift net als bij een lichtkrant van rechts naar links over het scherm.

Elke telefoonabonnee die naar een dove opbelt, kan, mits hij een telefoontoestel met druktoetsen heeft, op de teksttelefoon een tekst doen verschijnen ("schrijven"). Daartoe typt hij de tekst in op het toetsenbordje van zijn druktoetsen-telefoon-toestel. De dove kan dan via de gewone telefoon antwoorden.

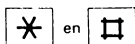


De betekenis van de toetsen volgens deze nieuwe afspraak is hieronder getekend (de teksttelefoon is zo gemaakt dat hij deze afspraak begrijpt).

We zien dat elke toets nu 4 betekenissen kan hebben.

Bijvoorbeeld de toets linksboven kan betekenen: de letter A, de letter B, de letter C of het cijfer 1.

Voor het kiezen van de gewenste betekenis, gebruiken we de twee speciale toetsen



Dit gaat als volgt:

indrukken van gevolgd door indrukken betekent: A

indrukken van gevolgd door indrukken betekent: C

direct indrukken van betekent: B

achter elkaar indrukken van beide speciale toetsen en gevolgd door

betekent: 1

```

10      ; *****
20      ; *** Teksttelefoon voor OSI ***
30      ; ***                               ***
40      ; ***   (c) John Glaser   ***
50      ; ***                               ***
60      ; ***   Tel.033-751472   ***
70      ; *****
80      ;
90      ;
100     *=1000
110     ;
120     ATT   =*-1
130     VIA   =%C400
140     VIADR =VIA+1
150     VIADDR=VIA+3
160     VIAPCR=VIA+12
170     VIAIFR=VIA+13
180     VIDE0 =%F800
190     KEYB  =%FD00
200     EXTMON=%E800
210     ;
220INIT  LDA  #%11110000      A0 T/M A3 INPUT
230     STA  VIADDR
240     LDA  #%00001001      CA1=STROBE (DV)
250     STA  VIAPCR          CA2=ACKN (CL)
260     LDA  #%00000000      CLEAR REGISTER
270     STA  VIAIFR
280     LDA  #1
290     STA  $0212          DISABLE CTRL/C
300     LDA  #26
310     JSR  VIDE0
320     ;
330CLRAAT LDA  #%00000000      BIT0=*
340     STA  ATT            BIT1=#
350BEGIN  CLC
360     LDA  VIAIFR
370     LSR  A
380     LSR  A
390     BCS  GETCHR
400     LDX  #0
410LOOP  INX
420     BNE  LOOP
430     LDA  #254
440     STA  $DF00
450     LDA  $DF00          KEYBOARD
460     CMP  #222          ESC TOETS ?
470     BNE  BEGIN
480     LDA  #16
490     STA  $F0
500     JMP  EXTMON
510     ;
520GETCHR LDX  VIADR
530TEST0  CFX  #$0B
540     BNE  TEST1
550     LDA  ATT
560     ORA  #%00000001      SET BIT0=*
570     STA  ATT
580     JMP  BEGIN
590     ;

```



```

600TEST1   OPX  ##0C           #
610        BNE  ATTO
620        LDA  ATT
630        ORA  #%000000010     SET BIT1=#
640        STA  ATT
650        JMP  BEGIN
660        ;
670ATTO    CLC
680        LDA  ATT
690        BEQ  SETIND
700        LSR  A
710        BCC  LOAD20          BIT0=1 ?
720        LSR  A
730        BCC  LOAD10          BIT1=1 ?
740        TXA
750        ORA  ##30            TABEL3
760        JMP  TRANSF
770        ;
780LOAD20  TXA
790        ORA  ##20            TABEL2
800        JMP  TRANSF
810        ;
820LOAD10  TXA
830        ORA  ##10            TABEL1
840        ;
850TRANSF  TAX
860        ;
870GETIND  LDA  TABEL,X
880        JSR  VIDE0
890        JMP  CLRATT
900        ;
910TABEL   .BYTE $00,'BEHKNRUY', $5F,' '
920        .BYTE $00,$00,$00,$00,$00
930        .BYTE $00,'ADGJMPTW.0'
940        .BYTE $00,$00,$00,$00,$00
950        .BYTE $00,'CFILDSVZ?X'
960        .BYTE $00,$00,$00,$00,$00
970        .BYTE $00,'1234567890'
980        .BYTE $00,$00,$00,$00,$00
990        ;
1000      ; END

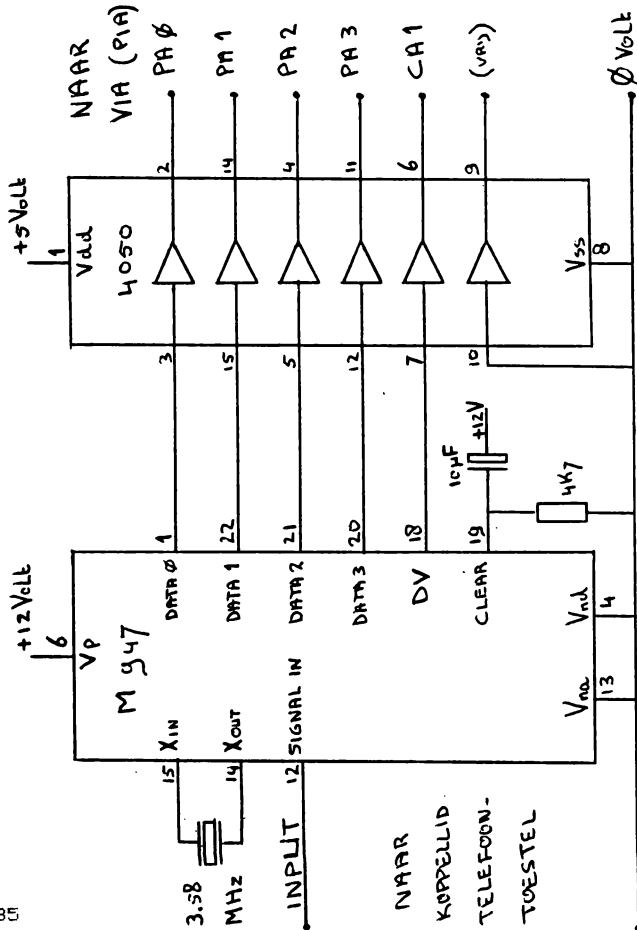
```

TDK Toon-decoder

Detecteert de tonen (cijfers) van een druktoetstelefoon o.a. voor teksttelefoon of afstandsbesturing per telefoon.
 TDK-decoder = M947 van Teltone (Group 2000 Almelo)
 prijs \$53.- per stuk, \$23 per 1000 (tel 05490-21616)
 (deze prijzen worden onder voorbehoud gegeven)
 5 volt driver = 4050
 kristal = 3.579545 MHz (o.a. bij Tandy fl. 6,50 ovb)

VIA initialisatie PA0 t/m 3 = input PA4 t/m 7 = output
 CA1 set interrupt (bit 1) bij negatieve strobe.
 De M947 is een 20 pins IC met 0,4 inch tussen de connecties !

John Glaser 033-751472



```

10          ;*****
20          ;*
30          ;*          PRBUF          *
40          ;*
50          ;* SOFTWARE PRINTBUFFER. TE GEBRUIKEN *
60          ;* BIJ BASIC ONDER OS-650 3.3      *
70          ;*
80          ;* VERSIE 2.2   26-12-'84          *
90          ;*
100         ;* AUTEUR: HANS DE JONG           *
110         ;*
120         ;*****
130         ;
140         ; Functie:
150         ;
160         ; Direct nadat de opdracht is seseven een hoeveel-
170         ; heid tekst of een programmalistings af te drukken
180         ; is de computer weer beschikbaar voor andere ta-
190         ; ken. Er hoeft niet gewacht te worden tot de
200         ; printer klaar is met afdrukken. Dit is een
210         ; alternatief voor een hardware printbuffer.
220         ;
230         ; Wijzigingen in OS-650 3.3:
240         ;
250         ; Worden bij het opstarten van de routine automa-
260         ; tisch aangebracht.
270         ; Deze routine vervangt de driver van output device
280         ; 3 (UART of 430-board). Ook worden patches aange-
290         ; bracht in de keyboard input routine en de routine
300         ; die na uitvoering van een BASIC-programmaresel
310         ; naast of CNTRL-C is ingedrukt.
320         ;
330         ; Werking:
340         ;
350         ; Van de aanwezige 36 k RAM wordt de bovenste 8 k
360         ; gebruikt om de routine in te bewaren en verder
370         ; als bufferegeheugen voor de printer.
380         ;
390         ; Met PRINT#3 of LIST#3 wordt de buffer gevuld.
400         ; Tijdens het draaien van een BASIC-programma of
410         ; het wachten op invoer van het toetsenbord wordt
420         ; telkens nagesaan of de printer een teken wil
430         ; accepteren. Zoja, dan wordt een teken uit de
440         ; buffer verzonden.
450         ; De buffer is cyclisch uitgevoerd. Als de invoer
460         ; of uitvoer pointer aan het einde van de buffer
470         ; is gekomen, wordt weer verdergesaan aan het
480         ; begin. De buffer kent 3 staten:
490         ; - de buffer is leeg: er wordt niets naar de
500         ; printer gestuurd.
510         ; - de buffer is vol: het vullen van de buffer
520         ; moet wachten tot er een teken is verzonden.
530         ; - de buffer is gedeeltelijk gevuld: het vullen
540         ; en lezen van de buffer vindt onafhankelijk
550         ; van elkaar plaats.
560         ;

```

```

570 ; Implementeren:
580 ;
590 ; Het is het beste de source tekst via de assem-
600 ; blen/editor in te tikken en zondias de plaats
610 ; en grootte van de buffer aan te passen aan de
620 ; eigen machine. De source op schijf bewaren.
630 ; Assembleren met A3 en de objectcode op schijf
640 ; zetten met !SA tr,s=start/2 (tr tracknummer,
650 ; s = sectornummer, start =beginadres object-
660 ; code).
670 ; Er is rekening gehouden met een printer die
680 ; via een seriele interface is aangesloten.
690 ; Voor een parallele interface moeten de routines
700 ; PRKLAR, PRZEND en INIT worden aangepast.
710 ;
720 ; Gebruik:
730 ;
740 ; Kan zowel in een programma of interaktief:
750 ;
760 ; DISK!"CA start=tr,s
770 ; DISK!"GO start
780 ;
790 ; Nu is de routine geladen en zijn de wijzigingen
800 ; in OS-650 3.3 aangebracht. Het is geen bezwaar
810 ; als er al een BASIC-programma in het werksehu-
820 ; ren zit, mits er natuurlijk voldoende ruimte over
830 ; is voor de buffer.
840 ;
850 ; Nu kan bijvoorbeeld LIST#3 worden ingetikt.
860 ; Na korte tijd komt er OK op het scherm. Nu kan
870 ; onmiddellijk een ander programma van disk worden
880 ; geladen en dit kan worden gestart of ook naar de
890 ; de printer worden verzonden. Het afdrukken gaat
900 ; gewoon door. Zelfs is het mogelijk de assembler/
910 ; editor te laden zonder het afdrukken te verst-
920 ; ren. Bij terugkeer naar BASIC is dan echter wel
930 ; de patch in de CNTRL C routine verdwenen. Om dit
940 ; te herstellen moet de printbufferroutine opnieuw
950 ; worden geladen (nadat het afdrukken klaar is) en
960 ; opgestart.
970 ;
980 ;*****
990 ;
1000 ; DECLARATIES
1010 ;
1020 2391=          UULPNT=$2391      ;UUL POINTER
1030 238A=          ZNDPNT=$238A      ;ZEND POINTER
1040 2390=          MEMOT=$2390      ;DRIVER OUTPUT DEVICE 5
1050 2389=          MEMIN=$2389      ;DRIVER INPUT DEVICE 5
1060 2363=          A.HLD=$2363      ;TYDELYKE OPSLAG A DOOR OS
1070 FC00=          ACIAC=$FC00      ;CONTROLE-REGISTER ACIA
1080 FC01=          ACIAD=$FC01      ;DATA-REGISTER ACIA
1090 2531=          PAT1AD=$2531      ;KEYBOARD INPUT ROUTINE
1100 0819=          PAT2AD=$0819      ;CNTRL C ROUTINE
1110 3642=          PAT3AD=$3642      ;WACHTLUS KBI ROUTINE
1120 250D=          PAT4AD=$250D      ;OUTPUT DEVICE 3
1130 ;
1140 ;*****
1150 ;
1160 ; ENTRYPOINTS EN ALGEMENE ROUTINES

```

```

1170
1180 7000          *=$7000
1190 7000 406071  START JMP INIT      ;INITIEER ROUTINE
1200 7003 206670  UULST JSR SAUREG   ;REGISTERS BEWAREN
1210 7006 202E70          JSR UUL0      ;UULROUTINE
1220 7009 407070          JMP GETREG    ;REGISTERS OPHALEN EN TERUGKEREN
1230 700C 202570  ZNDST1 JSR ZENDEN  ;ENTRYPPOINT VANUIT AANROEP KBI
1240 700F 40CC32          JMP $32CC    ;NAAR KBI ROUTINE IN DOS EXTEN-
1250                                     ;SIONS
1260 7012 202570  ZNDST2 JSR ZENDEN  ;ENTRYPPOINT VANUIT CNTRL C ROU-
1270                                     ;TINE
1280 7015 AD2523          LDA $2325    ;STOND OP PLAATS PATCH
1290 7018 60          RTS
1300 7019 202570  ZNDST3 JSR ZENDEN  ;ENTRYPPOINT VANUIT WACHTLUS KBI
1310 701C CA          DEX          ;STOND OP PLAATS PATCH
1320 701D D001          BNE ZST3A   ;WACHTEN NOG NIET KLAAR
1330 701F 60          RTS          ;WACHTEN KLAAR
1340 7020 68          ZST3A PLA        ;VERWIJDEREN TERUGKEERADRES LUS
1350 7021 68          PLA        ;VAN STACK
1360 7022 4C3D36          JMP $363D   ;NAAR BEGIN WACHTLUS
1370 7025 206670  ZENDEN JSR SAUREG   ;BEWAREN REGISTERS
1380 7028 204770          JSR ZEND0   ;ZENDROUTINE
1390 702B 4C7070          JMP GETREG  ;REGISTERS OPHALEN EN TERUG
1400          ;
1410          ;*****
1420          ;
1430          ; HOOFDROUTINE
1440          ;
1450 702E 208270  UUL0   JSR BUFUOL  ;BUFFER UUL?
1460 7031 D006          BNE UUL1    ;NEE, DAN VEERDER
1470 7033 204770  UUL0   JSR ZEND0   ;JA, DAN EERST PROBEREN TE ZENDEN
1480 7036 4C2E70  UUL0   JMP UUL0    ;EN OPNIEUW PROBEREN TE VULLEN
1490 7039 AD4971  UUL1   LDA A.TMP   ;TEKEN OPHALEN
1500 703C 209023  UUL1   JSR MEMOT   ;EN IN BUFFER ZETTEN (POINTER
1510                                     ;WORDT IN MEMOT OPGEHOOGD)
1520 703F 20C370          JSR CMPUL0  ;EINDE BUFFER? (UULPOINTER=LIMIT)
1530 7042 D003          BNE ZEND0  ;NEE, DAN NOG EEN TEKEN ZENDEN
1540 7044 20A970  ZEND0 JSR UULBEG  ;JA, DAN UULPOINTER := BEGIN
1550 7047 208F70  ZEND0 JSR BUFLEG  ;BUFFER LEEG?
1560 704A F019          BEQ ZNDEND ;JA, DAN NIKS TE ZENDEN
1570 704C 209870  ZEND0 JSR PRKLAR  ;KAN PRINTER TEKEN ONTVANGEN?
1580 704F 9014          BCC ZNDEND ;NEE, DAN NIKS TE ZENDEN
1590 7051 207A70  ZEND0 JSR SAVAH   ;JA, DAN A.HLD BEWAREN VOOR OS
1600 7054 208923  ZEND0 JSR MEMIN   ;TEKEN UIT BUFFER HALEN (POINTER
1610                                     ;WORDT IN MEMIN OPGEHOOGD)
1620 7057 209E70          JSR PRZEND  ;EN NAAR PRINTER STUREN
1630 705A 207E70  ZEND0 JSR GETAH   ;TERUGHALEN A.HLD VOOR OS
1640 705D 20ED70  ZEND0 JSR CMPZL   ;EINDE BUFFER?(ZNDPOINTER=LIMIT)
1650 7060 D003          BNE ZNDEND ;NEE, DAN TERUG
1660 7062 40B670  ZEND0 JMP ZNDBEG  ;JA, DAN ZNDPOINTER := BEGIN EN
1670                                     ;TERUG
1680 7065 60          ZNDEND RTS
1690          ;
1700          ;*****
1710          ;
1720          ; HULPROUTINES
1730          ;
1740 7066 8D4971  SAUREG STA A.TMP
1750 7069 8E4A71          STX X.TMP
1760 706C 8C4B71          STY Y.TMP

```

```

1770 706F 60          RTS
1780                ;
1790 7070 AD4971     GETREG LDA A.TMP
1800 7073 AE4A71     LDX X.TMP
1810 7076 AC4B71     LDY Y.TMP
1820 7079 60          RTS
1830                ;
1840 707A AC6323     SAVAH LDY A.HLD ;BEWAREN INHOUD ADRES DIE
1850 707D 60          RTS ;DOOR MENIN VERANDERD WORDT
1860                ;
1870 707E 8C6323     GETAH STY A.HLD ;WEER TERUGHALEN
1880 7081 60          RTS
1890                ;
1900 7082 20CC70     BUFVOL JSR CMPUL1 ;UULPNT + 1 = LIMIT?
1910 7085 D004       BNE BFVOL1 ;NEE, DAN POINTERS VERGELIJKEN
1920 7087 20E470     JSR CMPZF ;JA, IS ZNDPNT = FIRST?
1930 708A 60          RTS ;TERUG MET ANTWOORD IN Z-ULAG
1940 708B 20D870     BFVOL1 JSR CMPVZ1 ;UULPNT + 1 = ZNDPNT?
1950 708E 60          RTS ;TERUG MET ANTWOORD IN Z-ULAG
1960                ;
1970 708F 20F670     BUFLEG JSR MOUVA1 ;UULPNT NAAR A1
1980 7092 200871     JSR MOUZA2 ;ZNDPNT NAAR A2
1990 7095 4C3A71     JMP UGLA ;A1 EN A2 VERGELIJKEN EN TERUG
2000                ;
2010 7098 AD00FC     PRKLAR LDA ACIAC ;STATUSREGISTER ACIA BEKIJKEN
2020 709B 4A         LSR A
2030 709C 4A         LSR A ;KLAAR ALS C-ULAG = 1
2040 709D 60          RTS
2050                ;
2060 709E 48          PRZEND PHA ;TEKEN EVEN BEWAREN
2070 709F 209870     JSR PRKLAR ;KAN PRINTER TEKEN ONTVANGEN?
2080 70A2 90FB       BCC PRZEND+1 ;ZONEE, DAN OPNIEUW PROBEREN
2090 70A4 68         PLA ;TEKEN WEER OPHALEN
2100 70A5 8D01FC     STA ACIAD ;ZOJA, DAN TEKEN NAAR PRINTER
2110 70A8 60          RTS
2120                ;
2130 70A9 AD4C71     UULBEG LDA FIRST ;ZET UULPNT WEER AAN BEGIN BUFFER
2140 70AC 8D9123     STA UULPNT
2150 70AF AD4D71     LDA FIRST+1
2160 70B2 8D9223     STA UULPNT+1
2170 70B5 60          RTS
2180                ;
2190 70B6 AD4C71     ZNDBEG LDA FIRST ;ZET ZNDPNT WEER AAN BEGIN BUFFER
2200 70B9 8D8A23     STA ZNDPNT
2210 70BC AD4D71     LDA FIRST+1
2220 70BF 8D8B23     STA ZNDPNT+1
2230 70C2 60          RTS
2240                ;
2250 70C3 20F670     CMPUL0 JSR MOUVA1 ;UULPNT NAAR A1
2260 70C6 201A71     JSR MOULA2 ;LIMIT NAAR A2
2270 70C9 4C3A71     JMP UGLA ;A1 EN A2 VERGELIJKEN EN TERUG
2280                ;
2290 70CC 20F670     CMPUL1 JSR MOUVA1 ;UULPNT NAAR A1
2300 70CF 202371     JSR INCA1 ;A1 MET 1 VERHOGEN
2310 70D2 201A71     JSR MOULA2 ;LIMIT NAAR A2
2320 70D5 4C3A71     JMP UGLA ;A1 EN A2 VERGELIJKEN EN TERUG
2330                ;
2340 70D8 20F670     CMPVZ1 JSR MOUVA1 ;UULPNT NAAR A1
2350 70DB 202371     JSR INCA1 ;A1 MET 1 VERHOGEN
2360 70DE 200871     JSR MOUZA2 ;ZNDPNT NAAR A2

```

```

2370 70E1 4C3A71      JMP UGLA      ;A1 EN A2 VERGELIJKEN EN TERUG
2380                  ;
2390 70E4 20FF70      CMPZF JSR MOUZA1 ;ZNDPNT NAAR A1
2400 70E7 201171      JSR MOUFA2   ;FIRST NAAR A2
2410 70EA 4C3A71      JMP UGLA      ;A1 EN A2 VERGELIJKEN EN TERUG
2420                  ;
2430 70ED 20FF70      CMPZL JSR MOUZA1 ;ZNDPNT NAAR A1
2440 70F0 201A71      JSR MOULA2   ;LIMIT NAAR A2
2450 70F3 4C3A71      JMP UGLA      ;A1 EN A2 VERGELIJKEN EN TERUG
2460                  ;
2470 70F6 AD9123      MOUVA1 LDA UULPNT
2480 70F9 AC9223      LDY UULPNT+1
2490 70FC 4C2C71      JMP STORA1   ;UULPNT IN A1 ZETTEN
2500                  ;
2510 70FF AD8A23      MOUZA1 LDA ZNDPNT
2520 7102 AC8B23      LDY ZNDPNT+1
2530 7105 4C2C71      JMP STORA1
2540                  ;
2550 7108 AD8A23      MOUZA2 LDA ZNDPNT
2560 710B AC8B23      LDY ZNDPNT+1
2570 710E 4C3371      JMP STORA2
2580                  ;
2590 7111 AD4C71      MOUFA2 LDA FIRST
2600 7114 AC4D71      LDY FIRST+1
2610 7117 4C3371      JMP STORA2
2620                  ;
2630 711A AD4E71      MOULA2 LDA LIMIT
2640 711D AC4F71      LDY LIMIT+1
2650 7120 4C3371      JMP STORA2
2660                  ;
2670 7123 EE5071      INCA1 INC A1      ;VERHOOG LB A1
2680 7126 D003        BNE INCA1E    ;LB NIET 0 DAN TERUG
2690 7128 EE5171      INC A1+1     ;WEL 0, DAN HB OOK VERHOGEN
2700 712B 60          INCA1E RTS
2710                  ;
2720 712C 8D5071      STORA1 STA A1
2730 712F 8C5171      STY A1+1
2740 7132 60          RTS
2750                  ;
2760 7133 8D5271      STORA2 STA A2
2770 7136 8C5371      STY A2+1
2780 7139 60          RTS
2790                  ;
2800 713A AD5071      UGLA  LDA A1      ;LB VERGELIJKEN
2810 713D CD5271      CMP A2      ;GELIJK?
2820 7140 D006        BNE UGLAE    ;NEE, DAN TERUG
2830 7142 AD5171      LDA A1+1    ;HB VERGELIJKEN
2840 7145 CD5371      CMP A2+1    ;GELIJK?
2850 7148 60          UGLAE  RTS      ;TERUG MET ANTWOORD IN Z-ULAG
2860                  ;
2870                  ;*****
2880                  ;
2890                  ; VARIABELEN EN CONSTANTEN
2900                  ;
2910 7149 00          A.TMP .BYTE 0
2920 714A 00          X.TMP .BYTE 0
2930 714B 00          Y.TMP .BYTE 0
2940 714C 5471      FIRST .WORD PAT1 ;BUFFER SCHRIJFT OVER
2950                  ;PATCHES EN DE INITIEELE
2960                  ;ROUTINES HEEN

```

```

2970 714E 0090   LIMIT   .WORD $9000
2980 7150 0000   A1      .WORD 0
2990 7152 0000   A2      .WORD 0
3000           ;
3010           ;*****
3020           ;
3030           ; PATCHES
3040           ;
3050 7154 200C70  PAT1    JSR ZNDST1
3060 7157 201270  PAT2    JSR ZNDST2
3070 715A 201970  PAT3    JSR ZNDST3
3080 715D 4C0370  PAT4    JMP UULST
3090           ;
3100           ;*****
3110           ;
3120           ; AANBRENGEN PATCHES
3130           ;
3140 7160 A202    INIT    LDX #2
3150 7162 BD5471          LDA PAT1.X
3160 7165 9D3125          STA PAT1AD.X
3170 7168 BD5771          LDA PAT2.X
3180 716B 9D1908          STA PAT2AD.X
3190 716E BD5A71          LDA PAT3.X
3200 7171 9D4236          STA PAT3AD.X
3210 7174 BD5D71          LDA PAT4.X
3220 7177 9D0D25          STA PAT4AD.X
3230 717A CA            DEX
3240 717B 10E5          BPL INIT+2
3250           ;
3260           ;*****
3270           ;
3280           ; INITIEREN POINTERS EN INTERFACES
3290           ;
3300 717D A96F          LDA #START/256-1
3310 717F 8DFE2F          STA $2FFE ; GESWAPTE #85
3320 7182 8DFC2F          STA $2FFC ; GESWAPTE #83
3330 7185 8DFA2F          STA $2FFA ; GESWAPTE #81
3340 7188 8D0023          STA $2300 ; HIMEM DOS
3350 718B 20A970          JSR UULBEG ;UULPNT AAN BEGIN BUFFER
3360 718E 20B670          JSR ZNDBEG ;ZNDPNT AAN BEGIN BUFFER
3370 7191 A903          LDA #3
3380 7193 8D00FC          STA ACIAC
3390 7196 A910          LDA #10 ;ACIA OP 4800 BAUD
3400 7198 8D00FC          STA ACIAC
3410 719B A960          LDA #560 ;BEVEILIGING TEGEN NOG EENS
3420 719D 8D0070          STA START ;INITIEREN
3430 71A0 60            RTS ;INITIEREN KLAAR
3440           ;
3450           ;*****

```


INITIALISATIE VAN PIA

De MOSI PIA kaart bevat 4 Pias , iedere Pia bestaat weer uit 2 helften namelijk de A zijde en de B zijde . Iedere zijde bestaat uit 1 IO Poort (dit zijn 8 input/output aansluitingen) en 2 controle aansluitingen welke we voor het gemak maar even vergeten (ze zijn van belang bij bijvoorbeeld het besturen van de IRQ of NMI lijn). Zijde A bestaat intern uit 3 registers n.l het controle register , het IO register en het datadirection register . Maar er zijn maar 2 adressen beschikbaar (bij Mosi kaart Pia no 3 is C110 het io register en C111 het controle register) dus is onmogelijk om het datadirection register direkt te bereken , daarom moet je eerst bit 2 van het controle register 0 maken dan bevindt zich op het adres van het IO register het datadirection register (direction=richting) . Wil je weer het IO register bereiken dan moet je bit 2 van het controle register weer 1 maken .

Wanneer je het datadirection register met binair 1111111 vult zullen alle IO lijnen als uitgang fungeren . Allemaal nullen doet alle IO lijnen als ingang fungeren . Iedere lijn is dus te Programmeren als in of als uitgang door in het datadirection register een ander waarde te zetten .

Nu een voorbeeld in basic :

```
10 IO=49424 :REM HEX C110
20 CR=49425 :REM HEX C111
30 POKE CR,0 :REM OP ADRES VAN IO REGISTER IS NU DATADIRECTION
REGISTER
40 POKE IO,15 :REM 15=BINAIR 00001111 DUS 4 RECHTSE WORDEN
UITGANG EN 4 LINKSE INGANG
50 POKE CR,4 :4=BIT 2 MOET 1 WORDEN DAN KOMT HET IO REGISTER
WEER OP ZIJN PLAATS
NU KUN JE PEEKEN EN POKEN OM TE LEZEN EN TE SCHRIJVEN UIT OF IN
JE POORTEN .
```

Martien Schot je weet wel van MOSI

OSI POEL

·
FEBRUARI 1985

Verbeteringen aan de monitor en toolkit.

Als uitbreiding op de aanpassingen van Jos Burghouts van september 1982 volgen hier nog enkele verbeteringen aan de H.W.Monitor en de Brabosi toolkit. Mensen die binnenkort toch hun EPROMs moeten programmeren kunnen die aanpassingen meteen aanbrengen.

DE TOOLKIT:

1. De H.W.Monitor doet naast datgene wat 'ie moet doen ook nog enkele andere zaken zoals Editen, piepjes maken en prompts op het scherm schrijven. Deze extra zaken hebben allemaal te maken met het Basic; als de monitor vanuit een ander programma wordt aangeroepen mag hij die extra's niet uitvoeren. Om nu onderscheid te kunnen maken tussen de verschillende programma's kijkt hij naar de Basic-flag op adres \$0228. Als deze ongelijk 0 is heeft hij te maken met Basic en anders met een ander programma DAT ER ZELF VOOR MOET ZORGEN DAT DE BASIC-FLAG 0 WORDT. De toolkit vergeet dat soms echter en moet op die plekken dus veranderd worden.
2. Bij de hele-scherm-editor (#E) wordt de ondergrens van het beeld verplaatst naar de regel waar men bezig is. Na afloop wordt die ondergrens echter teruggezet op \$1C, een vaste waarde dus. Die waarde is wel goed bij 24 en 48 kar. versies maar daarentegen fout bij 64 karakters zoals bij de MOSI.
3. De plaats op het scherm van de NMI-klok kan men van de onderkant van het beeld naar de bovenkant verplaatsen om te voorkomen dat de klok mee gaat bij het scrollen. Dit probleem doet zich alleen voor bij de 64 kar. versies omdat die het hele beeld gebruiken.

DE MONITOR:

4. Bij de 64 kar. versies werkt de insert niet goed. Hij werkt dan maar op een regel en gooit daar alle karakters vanaf. Dit komt omdat de linker beeldschermgrens bij 64 kar. op 0 staat (\$0222 ROWS). Deze fout kan men trouwens wegprogrammeren zonder de EPROM te hoeven wissen; er hoeven namelijk maar een paar bitjes geRESET te worden.
5. De hulpopbergplaats \$02FF moet waarschijnlijk \$022F zijn. Deze fout is er mogelijk in geslopen bij het in-tikken van de code voor de monitor.

Dit waren een paar foutjes. Er zijn best nog wel meer aanpassingen te verzinnen, maar die laat ik hier weg om van dit verhaal geen bomen-bos te maken. Wie er belangstelling voor heeft die vraagt er maar eens naar.

We gaan uit van een toolkit op \$7000:

1. 7435 4CEB7F JMP \$7FEB afzet, doe warme exit
- 771B 4CEB7F JSx\$7FEB anders uit via warme exit
- 77C2 4CEB7F JMP \$7FEB en doe warme exit

Als u ergens nog een plekje kunt vinden voor 14 bytes, dan kunt u de volgende verandering ook doorvoeren:

```
7B07 A957 LDA ##57 Herstel CR+LF+
7B09 8D2802 STA $0228 Basic-flag Control-C
7B0C 4Cbbaa JMP $aabb en doe uitstapje
```

```
aabb 209BFF JSR $FF9B Control-C
+3 A900 LDA ##00 Clear
+5 8D2802 STA $0228 de Basic-flag
+B 206CAB JSR $AB6C doe CR+LF
+D A9FF LDA ##FF Vul accu
+D 60 RTS en ga terug
```

2. 7360 A2cc LDX ##cc dd: 1C bij 48 kar.
1F bij 64 kar.
(is orig. 1C)

```
75EB 9D22D0 STA $D033,X
75F1 9D33D0 STA $D033,X
75FA 8D33D0 STA $D033
```

4. FA57 1009 BPL \$FA62 zoja, verder springen

5. FA92 8E2F02 STX \$022F
- FA95 AD2F02 LDA \$022F
- FA98 8E2F02 STX \$022F

Henk Schaer,
Graswinckelstraat 2F,
2613 PW Delft.

FEBRUARI 1985

OSI POEL

```

*****
*                               *
* FULL SCREEN *
* TEKSTEDITOR *
* SIEM OPPE  *
* JANUARI '85 *
*                               *
*****

```

Met dit Programma kan tekst worden ingevoerd, verbeterd, gePrint en op-geborgen.

Het Programma werkt met Pagina's. De Pagina's zijn een scherm groot. De grootte kan dus van machine tot machine verschillen. Een scherm kan worden getypt, verbeterd en dan in het geheugen worden opgeborgen. Bij het typen van een scherm komen de "full-screen-editor" mogelijkheden van onze OSI-monitor volledig tot hun recht. CTRL-A,U,J,R,L,I,D (om de belangrijkste maar te noemen) kunnen alle worden gebruikt bij de opmaak van het scherm. Het scherm kan vervolgens met een Pagina-nummer naar het geheugen worden geschreven en later worden gePrint.

Het Programma komt op met de vraag om de Printbreedte aan te geven. Het (decimaal) ingevoerde getal geeft de lengte aan van de regels die door de Printer worden afgedrukt. De lengte is onafhankelijk van de lengte van de regels op het scherm. Wel geeft een Printindicator op het scherm aan tot hoever een regel mag doorlopen. De tekst na deze indicator wordt onderdrukt. Na het invoeren van de Printbreedte (een uit twee cijfers bestaand getal gevolgd door CR) wordt Pagina 00 op het scherm afgebeeld met informatie over de diverse mogelijkheden. Deze informatie kan worden overschreven als men de bijbehorende geheugenruimte wil benutten. Bovenaan het scherm staan de mogelijke commands afgedrukt waaruit men vervolgens moet kiezen. Typt men T01 en CR in, dan verschijnt een schoon scherm.

De Type-optie.

Vanuit de Command-mode kan men in de Type-mode komen, door Txx te typen. T00 geeft de informatie over de diverse opties terug. Binnen de Type-mode kan men behalve typen en verbeteren ook een regel tussen voegen of weghalen na de regel waarop de cursor staat. Dit gebeurt resp. met ESC + (ESC ; mag ook) en ESC - . Het scherm wordt dan eerst weggeschreven naar het geheugen, vervolgens wordt de betreffende tekst opgeschoven en teruggespronken naar de Type-mode. De cursor wordt teruggezet waardoor herhaald toepassen mogelijk is. Men kan deze opties ook gebruiken om ingetypte tekst tussentijds op te bergen. Met ESC,ESC springt men uit de Type-mode terug naar de Command-mode. Men kan dan een Pagina opbergen in het geheugen mbv het Sxx commando, Printen enz. Let erop dat bij ESC de shiftlock is ingedrukt!

De Save-, Print- en Move-optie spreken voor zich. Men kan met Sxx het scherm ook naar een andere Pagina wegschrijven dan waarop men heeft getypt. De Move-optie is beveiligd tegen het overschrijven van het RAM-geheugen dat voor Pagina 00 ligt. Met deze optie kan ruimte worden gemaakt voor een blok tekst. Met de Save-optie kan men tekst hiernaar verplaatsen.

De Extended-Monitor-optie.

Een optie om naar cassette weg te schrijven is er niet. Men kan echter wel de Extended Monitor oproepen (op adres E2E4) en met behulp van bv. GE700 tekst naar cassette schrijven en met E600 weer binnen halen. Het

OSI POEL

returnadres is dan 0000 (bv. 0800/2000/0000). Ook kan men met deze optie een disassembly vragen, van commentaar voorzien en vervolgens in het geheugen opslaan en Printen. Met U kan worden teruggeprongen naar de Command-optie.

Er is zowel een RAM-versie als een EPROM versie vanaf C000.

Voor je gaat laden of vanuit een EPROM opstart, eerst met BREAK en C de parameters goed zetten, anders geeft de Extended Monitor soms problemen. Soms zijn deze op te vangen met een warme start, of een koude start gevolgd door bv. S20.

Tenslotte volgt er nog een assembler listing.

Veel succes!

S.O.P.P.e

Nassaulaan 7B
2382GP Zoeterwoude
tel 070-410460

COMMANDS:

Txx: TYPE PG xx

ESC,ESC: CMD-MODE

ESC, + : INSERT LINE NA CURSOR

ESC, - : DELETE LINE NA CURSOR

Sxx: SAVE PG xx

Pxx,yy,zz: PRINT PG xx T/M yy, LINE zz

Mxx,yy,+zz: SCHUIF PG xx LINE yy, zz LINES OP

Mxx,yy,-zz: SCHUIF PG xx LINE yy, zz LINES TERUG

E: EXT.MON.

TERUG: U

ENIGE BELANGRIJKE ADRESSEN:

KOUDE START : 0300

WARME START : 0303

PAGE ZERO ADRESSEN:

00E0/00	OORSPRONG COPY/MOVE	<L>
00E1/00		<H>
00E2/00	BESTEMMING COPY	<L>
00E3/00		<H>

```

00E4/76   BESTEMMING MOVE                (L)
00E5/04   (H)
00E6/04   *****
00E7/1A   AANTAL REGELS OP SCHERM
00E8/47   HOMEADRES SCHERM                (L)
00E9/D1   (H)
00EA/C7   HOMEADRES COMMANDSTRING        (L)
00EB/D0   (H)
00EC/02   LAATSTE PAGINA (PRINT)
00ED/C7   AANTAL REGELS LAATSTE PG (PRINT)
00EE/01   AANTAL REGELS SCHUIVEN (MOVE)
00EF/96   REGELLENGTE (PRINT)
00F0/09   *****
00F1/90   *****
00F2/F0   BEGINADRES PAGINA nn (BV Tnn)    (L)
00F3/20   (H)
00F4/04   LAATSTE PAGINA
00F5/36   REGELLENGTE SCHERM
00F6/7C   PAGINA GROOTTE                    (L)
00F7/05   (H)
00F8/00   BEGINADRES TEKST                  (L)
00F9/0B   (H)
00FA/01   NUMMER VAN DE PAGINA IN GEBRUIK

```

PAGE TWO ADRESSEN:

```

022F/00   STACK-INDICATOR (TYPE)
0290 T/M  0293 HULPADRESSEN (MOVE)
029B T/M  02CF COMMANDSTRING
02DB T/M  02DD HULPADRESSEN (COPY)
02E0 T/M  02E9 INVOER COMMANDSTRING

```

OSI POEL

10	0300		*=0300
20	0300	4C0C03	JMP K0UD ; KOUDE START
30	0303	EE2402	INC #0224
40	0306	EE2402	INC #0224
50	0309	4C3204	JMP DCD ; WARME START
60	030C	A94C	LDA ##4C
70	030E	8500	STA #0000
80	0310	AD0A03	LDA #030A
90	0313	8501	STA #0001
100	0315	AD0B03	LDA #030B
110	0318	8502	STA #0002
120	031A	A900	LDA ##0
130	031C	8D2F02	STA #022F
140	031F	8D9002	STA #0290
150	0322	A900	LDA ##00 ; BEGIN GEHEUGEN
160	0324	85F8	STA #F8
170	0326	A90B	LDA ##0B
180	0328	85F9	STA #F9
190	032A	A232	LDX ##32
200	032C	8D6909	LDA TAB-1,X ; CMD-STRING
210	032F	9D9B02	STA TAB2-1,X
220	0332	CA	DEX
230	0333	D0F7	BNE CMD00
240	0335	2052FF	JSR #FF52 ; SCHERM SCHOON/HOME
250	0338	AD2C02	LDA #022C
260	033B	85EA	STA #EA ; SCHERMADRES CMD#
270	033D	AD2D02	LDA #022D
280	0340	85EB	STA #EB
290	0342	38	SEC
300	0343	AD2302	LDA #0223
310	0346	ED2202	SBC #0222
320	0349	85F5	STA #F5
330	034B	E6F5	INC #F5 ; REGELLENGTE SCHERM
340	034D	EE2402	INC #0224
350	0350	EE2402	INC #0224
360	0353	2093F9	JSR #F993
370	0356	AD2C02	LDA #022C
380	0359	85E8	STA #E8 ; HOME ADRES
390	035B	AD2D02	LDA #022D
400	035E	85E9	STA #E9
410	0360	18	CLC ; PAG. GROOTTE
420	0361	AD2502	LDA #0225
430	0364	ED2402	SBC #0224
440	0367	85F6	STA #F6
450	0369	E6F6	INC #F6
460	036B	A5F6	LDA #F6
470	036D	85E7	STA #E7 ; AANTAL REGELS
480	036F	A900	LDA ##0
490	0371	85F4	STA #F4 ; LAST PAGE
500	0373	85FA	STA #FA ; PAGE NR.
510	0375	85F7	STA #F7
520	0377	A209	LDX ##9
530	0379	18	CLC
540	037A	66F7	ROR #F7
550	037C	66F6	ROR #F6

560	037E	9007		BCC	DCR
570	0380	18		CLC	
580	0381	A5F5		LDA	#F5
590	0383	65F7		ADC	#F7
600	0385	85F7		STA	#F7
610	0387	0A	DCR	DEX	
620	0388	D0F0		BNE	MUL ; PAG.GROOTTE IN F6,F7
630	038A	18		CLC	
640	038B	AD2202		LDA	#0222
650	038E	690E		ADC	#9E
660	0390	8D2902		STA	#0229
670	0393	AD2402		LDA	#0224
680	0396	8D2A02		STA	#022A
690	0399	208BF8		JSR	#F808
700	039C	AD2C02		LDA	#022C
710	039F	85E0		STA	#E0
720	03A1	AD2D02		LDA	#022D
730	03A4	85E1		STA	#E1
740	03A6	A000		LDY	#00 ; PRINTBREEDTE
750	03A8	B99C09	PBR	LDA	TAB1,Y
760	03AB	91E8		STA	(#E8),Y
770	03AD	08		INY	
780	03AE	C00D		CPY	#0D
790	03B0	D0F6		BNE	PBR
800	03B2	20E9F8		JSR	#FAE9
810	03B5	A907		LDA	#907
820	03B7	2000F8		JSR	#FA00
830	03BA	20EDFE	LL	JSR	#FEED
840	03BD	2000F8		JSR	#FA00
850	03C0	C90D		CMP	#0D
860	03C2	D0F6		BNE	LL
870	03C4	A000		LDY	#00
880	03C6	B1E0	TR	LDA	(#E0),Y
890	03C8	99CE02		STA	TAB2+#32,Y
900	03CB	99E400		STA	#E4,Y
910	03CE	08		INY	
920	03CF	C002		CPY	#02
930	03D1	D0F3		BNE	TR
940	03D3	38		SEC	
950	03D4	A5E0		LDA	#E0
960	03D6	E91D		SBC	#91D
970	03D8	85E0		STA	#E0
980	03DA	A5E1		LDA	#E1
990	03DC	E900		SBC	#90
1000	03DE	85E1		STA	#E1
1010	03E0	20DA05		JSR	CLEAN
1020	03E3	200605		JSR	CYF
1030	03E6	A5E5		LDA	#E5
1040	03E8	85EF		STA	#EF
1050	03EA	85E4	CCD	STA	#E4
1060	03EC	A5F5		LDA	#F5
1070	03EE	C5E4		CMP	#E4
1080	03F0	B008		BCS	CCD
1090	03F2	38		SEC	
1100	03F3	A5EF		LDA	#EF

1110	03F5	E5F5		SBC #F5
1120	03F7	4CEA03		JMP CDD
1130	03FA	18	CCC	CLC
1140	03FB	A5E4		LDA #E4
1150	03FD	6940		ADC ##40
1160	03FF	8DDE02		STA #02DE ; PRINT INDICATOR
1170	0402	2052FF		JSR #FF52
1180	0405	A200		LDX ##0 ; WITLEG NAAR PG 00
1190	0407	BDA909	ARX:	LDA TAB3,X
1200	040A	2000F8		JSR #F800
1210	040D	EA		INX
1220	040E	E0FB		CPX ##FB
1230	0410	D0F5		BNE ARX:
1240	0412	A200		LDX ##00
1250	0414	BDA40A	ARY:	LDA TAB4,X
1260	0417	2000F8		JSR #F800
1270	041A	EA		INX
1280	041B	E033		CPX ##33
1290	041D	D0F5		BNE ARY:
1300	041F	A930		LDA ##30
1310	0421	8DE102		STA #02E1
1320	0424	8DE202		STA #02E2
1330	0427	A5F8		LDA #F8
1340	0429	85F2		STA #F2
1350	042B	A5F9		LDA #F9
1360	042D	85F3		STA #F3
1370	042F	20E405		JSR SAVE
1380	0432	A907	CDD	LDA ##07
1390	0434	2000F8		JSR #F800
1400	0437	A5EA		LDA #EA
1410	0439	85E0		STA #E0
1420	043B	A5EB		LDA #EB
1430	043D	85E1		STA #E1
1440	043F	20DA05		JSR CLEAN
1450	0442	A010		LDY ##10
1460	0444	899802	PNT	LDA TAB2-1,Y ; PRINT CMD-STRING
1470	0447	91EA		STA (#EA),Y
1480	0449	88		DEY
1490	044A	D0FB		BNE PNT
1500	044C	18		CLC ; PRINT STREEP
1510	044D	A5F5		LDA #F5
1520	044F	6940		ADC ##40
1530	0451	85E4		STA #E4
1540	0453	A4E4		LDY #E4
1550	0455	A92D		LDA ##2D
1560	0457	91EA	PRT	STA (#EA),Y
1570	0459	88		DEY
1580	045A	C040		CPY ##40
1590	045C	B0F9		BCS PRT
1600	045E	ACDE02		LDY #02DE ; PRINT INDICATOR
1610	0461	A9EE		LDA ##EE
1620	0463	91EA		STA (#EA),Y
1630	0465	18		CLC ; HAAL CMD-STRING BINNEN
1640	0466	AD2202		LDA #0222
1650	0469	6911		ADC ##11

1660	046B	8D2902		STA	#0229
1670	046E	AE2402		LDX	#0224
1680	0471	CA		DEX	
1690	0472	CA		DEX	
1700	0473	8E2A02		STX	#022A
1710	0476	20EDFE	LS	JSR	#FEED
1720	0479	C90D		CMP	#00D
1730	047B	F005		BEQ	DECOD
1740	047D	2000F8		JSR	#F000
1750	0480	D0F4		BNE	LS
1760	0482	A011	DECOD	LDY	#011 ; DECODEREN CMD*
1770	0484	B1EA	STO	LDA	(#EA),Y
1780	0486	99CF02		STA	#02CF,Y ; STORE VANAF #02E0
1790	0489	C8		INY	
1800	048A	C01B		CPY	#01B
1810	048C	D0F6		BNE	STO
1820	048E	ADE002		LDA	#02E0
1830	0491	C953		CMP	#053
1840	0493	D006		BNE	A1 :
1850	0495	20E405		JSR	SAVE
1860	0498	4C3204		JMP	DCD
1870	049B	C973	A1 :	CMP	#073
1880	049D	D006		BNE	A :
1890	049F	20E405		JSR	SAVE
1900	04A2	4C3204		JMP	DCD
1910	04A5	C954	A :	CMP	#054
1920	04A7	D003		BNE	B1 :
1930	04A9	4C2306		JMP	TYPE
1940	04AC	C974	B1 :	CMP	#074
1950	04AE	D003		BNE	B :
1960	04B0	4C2306		JMP	TYPE
1970	04B3	C950	B :	CMP	#050
1980	04B5	D003		BNE	C1 :
1990	04B7	4C4D07		JMP	PRINT
2000	04BA	C970	C1 :	CMP	#070
2010	04BC	D003		BNE	C :
2020	04BE	4C4D07		JMP	PRINT
2030	04C1	C945	C :	CMP	#045
2040	04C3	D003		BNE	D :
2050	04C5	4C2D05		JMP	EMTOOL
2060	04C8	C94D	D :	CMP	#04D
2070	04CA	D00B		BNE	E1 :
2080	04CC	20B007		JSR	MOVE
2090	04CF	A907		LDA	#007
2100	04D1	2000F8		JSR	#F800
2110	04D4	4C2306		JMP	TYPE
2120	04D7	C96D	E1 :	CMP	#06D
2130	04D9	D00B		BNE	E :
2140	04DB	20B007		JSR	MOVE
2150	04DE	A907		LDA	#007
2160	04E0	2000F8		JSR	#F800
2170	04E3	4C2306		JMP	TYPE
2180	04E6	4C3204	E :	JMP	DCD ; EINDE DECODEREN
2190	04E9	A5F8	BMPX	LDA	#F8 ; BEREKENEN PG GROOTTE
2200	04EB	85F2		STA	#F2

```

2210 04ED A5F9          LDA #F9
2220 04EF 85F3          STA #F3
2230 04F1 A6FA          LDX #FA
2240 04F3 F010          BEQ END
2250 04F5 18
2260 04F6 A5F2          LUS   CLC
2270 04F8 65F6          LDA #F2
2280 04FA 85F2          ADC #F6
2290 04FC 85F3          STA #F2
2300 04FE 65F7          LDA #F3
2310 0500 85F3          ADC #F7
2320 0502 CA           STA #F3
2330 0503 D0F0          DEX
2340 0505 60           BNE LUS
2350 0506 38           END   RTS
2360 0507 A5E4          CYF   SEC ; DEC. GETAL(2) NAAR HEX
2370 0509 E930          LDA #E4
2380 050B 85E4          SBC ##30
2390 050D A5E5          STA #E4
2400 050F E930          LDA #E5
2410 0511 85E5          SBC ##30
2420 0513 A6E4          STA #E5
2430 0515 F00A          LDX #E4
2440 0517 18           BEQ EIND
2450 0518 A5E5          CLC
2460 051A 690A          KEER LDA #E5
2470 051C 85E5          ADC ##A
2480 051E CA           STA #E5
2490 051F D0F7          DEX
2500 0521 60           BNE KEER
2510 0522 A034          EIND RTS
2520 0524 B99B02        STRING LDY ##34 ; PRINT CMD#
2530 0527 91EA          PRNT  LDA TAB2-1,Y
2540 0529 88           STA (#EA),Y
2550 052A D0F8          DEY
2560 052C 60           BNE PRNT
2570 052D 2093F9        EMTOOL JSR #F993
2580 0530 2054F9        JSR #F954
2590 0533 4CE8E2        JMP #E2E8 ; NAAR TOOLKIT
2600 0536 ACE802        COPY  LDX #02DB ; AANTAL REGELS
2610 0539 A900          LDA ##0
2620 053B 85E4          STA #E4
2630 053D A000          STAP1 LDY ##0
2640 053F AD0502        LDA #0205
2650 0542 F025          BEQ STAP2
2660 0544 B1E0          STAP7 LDA (#E0),Y
2670 0546 2069FF        JSR #FF69
2680 0549 E6E4          INC #E4
2690 054B A5E4          LDA #E4
2700 054D C5EF        CMP #EF
2710 054F D011          BNE AAA
2720 0551 A90D          LDA ##0D
2730 0553 2069FF        JSR #FF69
2740 0556 A90A          LDA ##0A
2750 0558 2069FF        JSR #FF69

```

2760	055B	A900		LDA	##0
2770	055D	05E4		STA	##E4
2780	055F	4C7205		JMP	STAP9
2790	0562	C8	AAA	INY	
2800	0563	C4F5		CPY	##F5
2810	0565	D0DD		BNE	STAP7
2820	0567	F009		BEQ	STAP9
2830	0569	B1E0	STAP2	LDA	(##E0),Y
2840	056B	91E2		STA	(##E2),Y
2850	056D	C8		INY	
2860	056E	C4F5		CPY	##F5
2870	0570	D0F7		BNE	STAP2
2880	0572	18	STAP9	CLC	
2890	0573	A5E0		LDA	##E0
2900	0575	6DD002		ADC	##02DD
2910	0578	05E0		STA	##E0
2920	057A	A5E1		LDA	##E1
2930	057C	6900		ADC	##00
2940	057E	05E1		STA	##E1
2950	0580	18		CLC	
2960	0581	A5E2		LDA	##E2
2970	0583	6DDC02		ADC	##02DC
2980	0586	05E2		STA	##E2
2990	0588	A5E3		LDA	##E3
3000	058A	6900		ADC	##00
3010	058C	05E3		STA	##E3
3020	058E	CA		DEX	
3030	058F	D0AC		BNE	STAP1
3040	0591	60		RTS	; EIND COPY
3050	0592	ADE002	LINESM	LDA	##02E0
3060	0595	05E4		STA	##E4
3070	0597	ADE902		LDA	##02E9
3080	059A	05E5		STA	##E5
3090	059C	200605		JSR	CYF
3100	059F	A5E5		LDA	##E5
3110	05A1	05EE		STA	##EE
3120	05A3	60		RTS	
3130	05A4	ADE702	LINESP	LDA	##02E7
3140	05A7	05E4		STA	##E4
3150	05A9	ADE802		LDA	##02E8
3160	05AC	05E5		STA	##E5
3170	05AE	200605		JSR	CYF
3180	05B1	A5E5		LDA	##E5
3190	05B3	05ED		STA	##ED
3200	05B5	60		RTS	
3210	05B6	ADE102	BEGPG	LDA	##02E1
3220	05B9	05E4		STA	##E4
3230	05BB	ADE202		LDA	##02E2
3240	05BE	05E5		STA	##E5
3250	05C0	200605		JSR	CYF
3260	05C3	A5E5		LDA	##E5
3270	05C5	05FA		STA	##FA
3280	05C7	60		RTS	
3290	05C8	ADE402	LASTPG	LDA	##02E4
3300	05CB	05E4		STA	##E4

3310	050D	ADE502		LDA	#02E5
3320	050E	85E5		STA	#E5
3330	0502	200605		JSR	CYF
3340	0505	A5E5		LDA	#E5
3350	0507	85EC		STA	#EC
3360	0509	60		RTS	
3370	050A	A4F5	CLEAN	LDY	#F5
3380	050C	A920		LDA	##20
3390	050E	91E0	LOP1	STA	(#E0).Y
3400	05E0	88		DEY	
3410	05E1	D0FB		BNE	LOP1
3420	05E3	60		RTS	
3430	05E4	20B605	SAVE	JSR	BEGPG
3440	05E7	A5F4		LDA	#F4
3450	05E9	C5FA		CMP	#FA
3460	05EB	B010		BCS	STAP
3470	05ED	ADE102		LDA	#02E1
3480	05F0	8DC602		STA	TAB2+#2A
3490	05F3	ADE202		LDA	#02E2
3500	05F6	8DC702		STA	TAB2+#2B
3510	05F9	A5FA		LDA	#FA
3520	05FB	85F4		STA	#F4
3530	05FD	20E904	STAP	JSR	BMFX
3540	0600	A5F2		LDA	#F2
3550	0602	85E2		STA	#E2
3560	0604	A5F3		LDA	#F3
3570	0606	85E3		STA	#E3
3580	0608	A5E8		LDA	#E8
3590	060A	85E0		STA	#E0
3600	060C	A5E9		LDA	#E9
3610	060E	85E1		STA	#E1
3620	0610	A5E7		LDA	#E7
3630	0612	8DD802		STA	#02DB
3640	0615	A5F5		LDA	#F5
3650	0617	8DDC02		STA	#02DC
3660	061A	A940		LDA	##40
3670	061C	8DDD02		STA	#02DD
3680	061F	203605		JSR	COPY
3690	0622	60		RTS); EINDE SAVE
3700	0623	20B605	TYPE	JSR	BEGPG
3710	0626	ADE102		LDA	#02E1
3720	0629	8DB702		STA	TAB2+#1B
3730	062C	ADE202		LDA	#02E2
3740	062F	8DB802		STA	TAB2+#1C
3750	0632	AD2202		LDA	#0222
3760	0635	8D2902		STA	#0229
3770	0638	AD2502		LDA	#0225
3780	063B	8D2A02		STA	#022A
3790	063E	2054F9		JSR	#F954
3800	0641	202205		JSR	STRING
3810	0644	A5F4		LDA	#F4
3820	0646	C5FA		CMP	#FA
3830	0648	B009		BCS	STAP3
3840	064A	2093F9		JSR	#F953
3850	064D	2054F9		JSR	#F954

3860	0650	4C7806		JMP	STAP4
3870	0653	20E904	STAP3	JSR	BMPX
3880	0656	A5F2		LDA	#F2
3890	0658	85E0		STA	#E0
3900	065A	A5F3		LDA	#F3
3910	065C	85E1		STA	#E1
3920	065E	A5E8		LDA	#E8
3930	0660	85E2		STA	#E2
3940	0662	A5E9		LDA	#E9
3950	0664	85E3		STA	#E3
3960	0666	A5E7		LDA	#E7
3970	0668	8DD802		STA	#02D8
3980	066B	A940		LDA	#940
3990	066D	8DDC02		STA	#02DC
4000	0670	A5F5		LDA	#F5
4010	0672	8DDD02		STA	#02DD
4020	0675	203605		JSR	COPY
4030	0678	2093F9	STAP4	JSR	#F993
4040	067B	AD2F02		LDA	#022F
4050	067E	C900		CMP	#000
4060	0680	F011		BEQ	INVOER
4070	0682	68		PLA	
4080	0683	8D2A02		STA	#022A
4090	0686	68		PLA	
4100	0687	8D2902		STA	#0229
4110	068A	20BFF8		JSR	#F8BF
4120	068D	20E9F0		JSR	#F8E9
4130	0690	CE2F02		DEC	#022F
4140	0693	20EDFE	INVOER	JSR	#FEED
4150	0696	2000FB		JSR	#F800
4160	0699	C91B		CMP	#01B ; ESCAPE
4170	069B	D07F		BNE	CC
4180	069D	20EDFE		JSR	#FEED
4190	06A0	C93B		CMP	#03B ; MOVE EEN REGEL +
4200	06A2	D008		BNE	AK
4210	06A4	A92B		LDA	#02B
4220	06A6	8DE702		STA	#02E7
4230	06A9	4CBD06		JMP	BB
4240	06AC	C92B	AK	CMP	#02B
4250	06AE	D006		BNE	AA
4260	06B0	8DE702		STA	#02E7
4270	06B3	4CBD06		JMP	BB
4280	06B6	C92D	AA	CMP	#02D ; MOVE EEN REGEL -
4290	06B8	D07E		BNE	STAP5
4300	06BA	8DE702		STA	#02E7
4310	06BD	AD2202	BB	LDA	#0222
4320	06C0	48		PHA	
4330	06C1	AD2A02		LDA	#022A
4340	06C4	48		PHA	
4350	06C5	A901		LDA	#001
4360	06C7	8D2F02		STA	#022F
4370	06CA	A930		LDA	#030
4380	06CC	8DE802		STA	#02E8
4390	06CF	A931		LDA	#031
4400	06D1	8DE902		STA	#02E9

```

4410 06D4 38          SEC          ; CURSORPLAATS
4420 06D5 A200       LDX  ##0
4430 06D7 AC2A02     LDY  #022A
4440 06DA C8        INY
4450 06DB 98        TYA
4460 06DC ED2402     SBC  #0224
4470 06DF E90A       TEL  SBC  #10
4480 06E1 3004       BMI  GETAL
4490 06E3 E8        INX
4500 06E4 4CDF06     JMP  TEL
4510 06E7 693B       GETAL ADC  ##3B
4520 06E9 8DE502     STA  #02E5
4530 06EC 8A        TXA
4540 06ED 692F       ADC  ##2F
4550 06EF 8DE402     STA  #02E4
4560 06F2 ADE702     LDA  #02E7
4570 06F5 C92D       CMP  ##2D
4580 06F7 D003       BNE  XX
4590 06F9 EEE502     INC  #02E5
4600 06FC AD2202     XX  LDA  #0222
4610 06FF 8D2902     STA  #0229
4620 0702 AD2502     LDA  #0225
4630 0705 8D2A02     STA  #022A
4640 0708 20BFF8     JSR  #F8BF
4650 070B 20E9F8     JSR  #F8E9
4660 070E 20E405     JSR  SAVE
4670 0711 20B007     JSR  MOVE
4680 0714 A907        LDA  ##07
4690 0716 2000F8     JSR  #F800
4700 0719 4C2306     JMP  TYPE ; EINDE REGEL SCHUIVEN
4710 071C C90D        CC  CMP  ##D
4720 071E D005       BNE  STAP6
4730 0720 A90A        LDA  #50A
4740 0722 2000F8     JSR  #F800
4750 0725 AD2A02     STAP6 LDA  #022A
4760 0728 CD2502     CMP  #0225
4770 072B D008       BNE  STAP6A
4780 072D A907        LDA  ##07
4790 072F 2000F8     JSR  #F800
4800 0732 CE2A02     DEC  #022A
4810 0735 4C9306     STAP6A JMP  INVOER
4820 0738 AD2202     STAP5 LDA  #0222
4830 073B 8D2902     STA  #0229
4840 073E AD2502     LDA  #0225
4850 0741 8D2A02     STA  #022A
4860 0744 20BFF8     JSR  #F8BF
4870 0747 20E9F8     JSR  #F8E9
4880 074A 4C3204     PRINT JMP  DCD ; EINDE TYPE
4890 074D A901        LDA  ##1
4900 074F 8D0502     STA  #0205
4910 0752 85EE       STA  #EE
4920 0754 20B605     JSR  BEGPG
4930 0757 20C805     JSR  LASTPG
4940 075A 20A405     JSR  LINESP
4950 075D 2093F9     JSR  #F993

```


4960	0760	2054F9		JSR #F954
4970	0763	A5E7		LDA #E7
4980	0765	80E702		STA #02E7
4990	0768	20E904	PRNTPG	JSR BMPX
5000	076B	A5F2		LDA #F2
5010	076D	85E0		STA #E0
5020	076F	A5F3		LDA #F3
5030	0771	85E1		STA #E1
5040	0773	A5E8		LDA #E8
5050	0775	85E2		STA #E2
5060	0777	A5E9		LDA #E9
5070	0779	85E3		STA #E3
5080	077B	A5E7		LDA #E7
5090	077D	80DB02		STA #02DB
5100	0780	A940		LDA ##40
5110	0782	80DC02		STA #02DC
5120	0785	A5F5		LDA #F5
5130	0787	80DD02		STA #02DD
5140	078A	A5FA		LDA #FA
5150	078C	C5EC		CMP #EC
5160	078E	900F		BCC STAPB
5170	0790	A5ED		LDA #ED
5180	0792	C5E7		CMP #E7
5190	0794	D002		BNE SSS
5200	0796	A5E7		LDA #E7
5210	0798	80DB02	SSS	STA #02DB
5220	079B	A900		LDA #00
5230	079D	85EE		STA #EE
5240	079F	203605	STAPB	JSR COPY
5250	07A2	E6FA		INC #FA
5260	07A4	A5EE		LDA #EE
5270	07A6	D0C0		BNE PRNTPG
5280	07A8	A900		LDA #00
5290	07AA	800502		STA #0205
5300	07AD	4C3204		JMP DCD ; EINDE PRINT
5310	07B0	20B605	MOVE	JSR BEGPG
5320	07B3	20C805		JSR LASTPG
5330	07B6	209205		JSR LINESM
5340	07B9	20E904		JSR BMPX
5350	07BC	A6EC		LDX #EC
5360	07BE	CA		DEX
5370	07BF	F010		BEQ SSA
5380	07C1	18		CLC
5390	07C2	A5F2	SSB	LDA #F2
5400	07C4	65F5		ADC #F5
5410	07C6	85F2		STA #F2
5420	07C8	A5F3		LDA #F3
5430	07CA	6900		ADC #00
5440	07CC	85F3		STA #F3
5450	07CE	CA		DEX
5460	07CF	D0F1		BNE SSB
5470	07D1	A5F2	SSA	LDA #F2
5480	07D3	85E0		STA #E0 ; OORSPRONG ADRES FL
5490	07D5	A5F3		LDA #F3
5500	07D7	85E1		STA #E1

5510	07D9	A5FA		LDA #FA
5520	07DB	8DDB02		STA #02DB
5530	07DE	A5F4		LDA #F4
5540	07E0	85FA		STA #FA
5550	07E2	E6FA		INC #FA
5560	07E4	20E904		JSR BMPX
5570	07E7	38		SEC
5580	07E8	AE9002		LDX #0290
5590	07EB	A5F2	STEPC	LDA #F2
5600	07ED	E5F5		SBC #F5
5610	07EF	85F2		STA #F2 ; OORSPRONG ADRES LL
5620	07F1	A5F3		LDA #F3
5630	07F3	E900		SBC #00
5640	07F5	85F3		STA #F3
5650	07F7	CA		DEX
5660	07F8	10F1		BPL STEPC
5670	07FA	A5F2		LDA #F2
5680	07FC	8D9102		STA #0291
5690	07FF	A5F3		LDA #F3
5700	0801	8D9202		STA #0292
5710	0804	ADE702		LDA #02E7
5720	0807	C92B		CMP #02B
5730	0809	F00A		BEQ STEP1
5740	080B	C92D		CMP #02D
5750	080D	D003		BNE FOUT
5760	080F	4CE108		JMP STEP2
5770	0812	4C3204	FOUT	JMP DCD
5780	0815	38	STEP1	SEC
5790	0816	AD9002		LDA #0290
5800	0819	E5EE		SBC #EE
5810	081B	8D9002		STA #0290
5820	081E	B024	STEPA	BCS STE1
5830	0820	E6F4	STEPB	INC #F4
5840	0822	18		CLC
5850	0823	65E7		ADC #E7
5860	0825	8D9002		STA #0290
5870	0828	A5F4		LDA #F4
5880	082A	85FA		STA #FA
5890	082C	A5E7		LDA #E7
5900	082E	8D9302		STA #0293
5910	0831	20E904		JSR BMPX
5920	0834	A5F2		LDA #F2
5930	0836	85E4		STA #E4
5940	0838	A5F3		LDA #F3
5950	083A	85E5		STA #E5
5960	083C	20C208		JSR BBH
5970	083F	AD9002		LDA #0290
5980	0842	30DC		BMI STEPB
5990	0844	AD9102	STE1	LDA #0291
6000	0847	85E4		STA #E4
6010	0849	AD9202		LDA #0292
6020	084C	85E5		STA #E5
6030	084E	A5E7		LDA #E7
6040	0850	CDE002		CMP #02E0
6050	0853	B00C		BCS STE

6060	0055	38		SEC
6070	0056	ADE002		LDA #02E0
6080	0059	E5E7		SBC #E7
6090	005B	0DE002		STA #02E0
6100	005E	4C4408		JMP STE1
6110	0061	A5F2	STE	LDA #F2
6120	0063	05E4		STA #E4
6130	0065	A5F3		LDA #F3
6140	0067	05E5		STA #E5
6150	0069	A6EE		LDX #EE
6160	006B	18		CLC
6170	006C	A5E4	BBC	LDA #E4 ; BESTEMMING LL
6180	006E	65F5		ADC #F5
6190	0070	05E4		STA #E4
6200	0072	A5E5		LDA #E5
6210	0074	6900		ADC #00
6220	0076	05E5		STA #E5
6230	0078	CA		DEX
6240	0079	D0F1		BNE BBC
6250	007B	A000	BBD	LDY #00
6260	007D	B1F2	BBE	LDA (#F2),Y
6270	007F	91E4		STA (#E4),Y
6280	0081	C8		INY
6290	0082	C4F5		CPY #F5
6300	0084	D0F7		BNE BBE
6310	0086	A5F2		LDA #F2
6320	0088	C5E0		CMP #E0
6330	008A	D008		BNE BBF
6340	008C	A5F3		LDA #F3
6350	008E	C5E1		CMP #E1
6360	0090	D002		BNE BBF
6370	0092	F01D		BEQ BBG
6380	0094	38	BBF	SEC
6390	0095	A5F2		LDA #F2
6400	0097	E5F5		SBC #F5
6410	0099	05F2		STA #F2
6420	009B	A5F3		LDA #F3
6430	009D	E900		SBC #00
6440	009F	05F3		STA #F3
6450	00A1	38		SEC
6460	00A2	A5E4		LDA #E4
6470	00A4	E5F5		SBC #F5
6480	00A6	05E4		STA #E4
6490	00A8	A5E5		LDA #E5
6500	00AA	E900		SBC #00
6510	00AC	05E5		STA #E5
6520	00AE	4C7B08		JMP BBD
6530	00B1	A5EE	BBG	LDA #EE
6540	00B3	0D9302		STA #0293
6550	00B6	A5E0		LDA #E0
6560	00B8	05E4		STA #E4
6570	00BA	A5E1		LDA #E1
6580	00BC	05E5		STA #E5
6590	00BE	20C208		JSR BBH
6600	00C1	60		RTS

6610	08C2	AE9302	BBH	LDX	#0293	;REGELS WISSEN
6620	08C5	A920	ABC	LDA	#020	
6630	08C7	A000		LDY	#00	
6640	08C9	91E4	BBI	STA	(&#E4),Y	
6650	08CB	C8		INY		
6660	08CC	C4F5		CPY	#F5	
6670	08CE	D0F9		BNE	BBI	
6680	08D0	18		CLC		
6690	08D1	A5E4		LDA	#E4	
6700	08D3	65F5		ADC	#F5	
6710	08D5	85E4		STA	#E4	
6720	08D7	A5E5		LDA	#E5	
6730	08D9	6900		ADC	#00	
6740	08DB	85E5		STA	#E5	
6750	08DD	CA		DEX		
6760	08DE	D0E5		BNE	ABC	
6770	08E0	60		RTS		
6780	08E1	18	STEP2	CLC		
6790	08E2	AD9002		LDA	#0290	
6800	08E5	65EE		ADC	#EE	
6810	08E7	8D9002		STA	#0290	
6820	08EA	38		SEC		
6830	08EB	E5EF	STEPD	SBC	#EF	
6840	08ED	3008		BMI	STEPE	
6850	08EF	C6F4		DEC	#F4	
6860	08F1	8D9002		STA	#0290	
6870	08F4	4CEB08		JMP	STEPD	
6880	08F7	A5E0	STEPE	LDA	#E0	
6890	08F9	85E4		STA	#E4	
6900	08FB	A5E1		LDA	#E1	
6910	08FD	85E5		STA	#E5	
6920	08FF	A6EE		LDX	#EE	
6930	0901	38		SEC		
6940	0902	A5E4	ARC	LDA	#E4	; BESTEMMING FL
6950	0904	E5F5		SBC	#F5	
6960	0906	85E4		STA	#E4	
6970	0908	A5E5		LDA	#E5	
6980	090A	E900		SBC	#00	
6990	090C	85E5		STA	#E5	
7000	090E	CA		DEX		
7010	090F	D0F1		BNE	ARC	
7020	0911	A5F9		LDA	#F9	
7030	0913	C5E5		CMP	#E5	
7040	0915	900B		BCC	ARD	
7050	0917	D006		BNE	ARX	
7060	0919	A5E4		LDA	#E4	
7070	091B	C5F8		CMP	#F8	
7080	091D	B003		BCS	ARD	
7090	091F	4C3204	ARX	JMP	DCD	
7100	0922	A000	ARD	LDY	#00	
7110	0924	B1E0	ARE	LDA	(&#E0),Y	
7120	0926	91E4		STA	(&#E4),Y	
7130	0928	C8		INY		
7140	0929	C4F5		CPY	#F5	
7150	092B	D0F7		BNE	ARE	

7160	092D	A5E0		LDA #E0
7170	092F	C5F2		CMP #F2
7180	0931	D008		BNE AAF
7190	0933	A5E1		LDA #E1
7200	0935	C5F3		CMP #F3
7210	0937	D002		BNE AAF
7220	0939	F01D		BEQ AAG
7230	093B	18	AAF	CLC
7240	093C	A5E0		LDA #E0
7250	093E	65F5		ADC #F5
7260	0940	85E0		STA #E0
7270	0942	A5E1		LDA #E1
7280	0944	6900		ADC #00
7290	0946	85E1		STA #E1
7300	0948	18		CLC
7310	0949	A5E4		LDA #E4
7320	094B	65F5		ADC #F5
7330	094D	85E4		STA #E4
7340	094F	A5E5		LDA #E5
7350	0951	6900		ADC #00
7360	0953	85E5		STA #E5
7370	0955	4C2209		JMP AAD
7380	0958	A6EE	AAG	LDX #EE
7390	095A	18		CLC
7400	095B	A5E4		LDA #E4
7410	095D	65F5		ADC #F5
7420	095F	85E4		STA #E4
7430	0961	A5E5		LDA #E5
7440	0963	6900		ADC #00
7450	0965	85E5		STA #E5
7460	0967	4CC208		JMP BBH ; EINDE MOVE
7470	096A	43	TAB	.BYTE 'CMD(S/T/P/E/M):? ',#95

TABELLEN:

7470	TAB	.BYTE 'CMD(S/T/P/E/M)? ',#95
7480		.BYTE ' PAGE NR:00 ',#95
7490		.BYTE ' LAST PAGE:00 ',#95
7500		.BYTE ' LL: '
7510	TAB1	.BYTE 'PRINTBREEDTE?'
7520	TAB2	= #029C
7530	TAB3	.BYTE 'COMMANDS:',#0D,#0A,#0D,#0A
7540		.BYTE 'Txx: TYPE PG xx',#0D,#0A
7550		.BYTE ' ESC,ESC: CMD-MODE'
7560		.BYTE #0D,#0A,' ESC, + : '
7570		.BYTE 'INSERT LINE NA CURSOR'
7580		.BYTE #0D,#0A,' ESC, - : '
7590		.BYTE 'DELETE LINE NA CURSOR'
7600		.BYTE #0D,#0A,#0D,#0A
7610		.BYTE 'Sxx: SAVE PG xx',#0D,#0A
7620		.BYTE #0D,#0A
7630		.BYTE 'Pxx,yy,zz: PRINT PG '
7640		.BYTE 'xx T/M yy, '
7650		.BYTE 'LINE zz',#0D,#0A,#0D,#0A
7660		.BYTE 'Mxx,yy,+zz: '
7670		.BYTE 'SCHUIF PG xx '
7680		.BYTE 'LINE yy, zz LINES '
7690		.BYTE 'OP',#0D,#0A
7700		.BYTE 'Mxx,yy,-zz: '
7710		.BYTE 'SCHUIF PG xx '
7720	TAB4	.BYTE 'LINE yy, zz LINES '
7730		.BYTE 'TERUG',#0D,#0A,#0D,#0A
7740		.BYTE 'E: EXT. MON.',#0D,#0A
7750		.BYTE ' TERUG: U',#0D,#0A

L I C H T K R A N T

Dit Programma laat een tekst over het beeldscherm schuiven van rechts naar links en als de tekst links verdwijnt, verschiint deze weer een regel hoger.

De karakters zijn opgebouwd met de Grafische karakterset van het OSI systeem. Ieder karakter is opgebouwd uit de Grafische karakters in een formaat van 3*4.

Enige toelichting bij het Programma :

Regel 1090-1120 Inlezen van machine taal deel
 Regel 1100-1180 inlezen van arrays voor karakters
 Regel 1190 USR Pointer naar machinetaal deel laten wijzen
 Regel 1200-1310 inPut deel (niet meer dan 10 schermen)
 Regel 1320-1390 colom van vier Grafische karakters wordt rechts onderin het scherm gepoked.
 Regel 1400 machinetaal routine welke scherm opschuift.
 Regel 1420-1470 voegt lege colom tussen 2 karakters.
 Regel 1640-1650 routine welke loze regel tussen 2 schermen voegt.
 Regel 1660-1740 data machinetaal deel
 Regel 1880-einde data voor karakteropbouw

Let op !!!!!!!!!!! ,wanneer je dit Programma in wilt typen, dan moet je niet vergeten eerst ruimte te maken voor je machinetaal deel. Je moet gewoon even de Pointer veranderen welke het begin van je basicfile aangeeft. Dus voor je begint doe je POKE 121,59.

Martien Schot

Hofstede 21

3902CG Veenendaal

Tel 08385-13219

```

10      ;MACHINETAAL ROUTINE VOOR LICHTKRANT
20      ;DOORSPRONKELIJK VOOR 48*24 SCHERM
30      ;GEMAAKT DOOR J M A HERMANS
40      ;
50      ;AANGEPAST VOOR 64*32 SCHERM DOOR MARTIEN
60      ;SCHOT
70      ;
80      ;MET LIJN WORDT EEN REGEL VAN 64 KARAKTERS
90      ;BEDOELD
100     ;MET PAGE WORDEN 4 REGELS BEDOELD
110     ;MET KARAKTER WORDT DATA OP EEN VIDEO ADRES
120     ;BEDOELD

```

```

130 3A80          *=$3A80 ;BEGIN VAN ROUTINE
140              )
150 3A80 A900     BEGIN LDA ##00
160 3A82 8510     STA #10      ;PLAATS BASIS ADRES VOOR Y GE-
170 3A84 A9D0     LDA ##D0      ;INDEXEERDE ADRESERING OP PAGE
180 3A86 8511     STA #11      ;ZERO VAN HET BEELDSCHERM.
190 3A88 A908     LDA ##08      ;MARK TELLERS IN PAGE ZERO OM
200 3A8A 8528     STA #28      ;8 PAGES TE TELLEN VAN 4 LIJN-
210 3A8C A904     NEXTPA LDA ##04 ;EN HOOG.
220 3A8E 8529     STA #29
230 3A90 A001     LDY #1        ;BOVEN BIJ 2de KARAKTER BEGINNEN
240 3A92 A23F     NEXTLY LDX #63 ;EN DAN 63 KARAKTERS
250 3A94 B110     NEXTKA LDA ($10),Y ;LAAD KARAKTER EN SCHUIF DIT
260 3A96 88       DEY          ;NAAR LINKS.
270 3A97 9110     STA ($10),Y
280 3A99 C8       INY          ;NU WEER 2 NAAR RECHTS KLAAR OM
290 3A9A C8       INY          ;VOLGEND KARAKTER OP TE HALEN.
300 3A9B CA       DEX          ;NOG GEEN 63 KARAKTERS GEHAD?
310 3A9C D0F6     BNE NEXTKA   ;DAN VOLGENDE.
320 3A9E 98       TYA
330 3A9F 18       CLC
340 3AA0 6901     ADC #1        ;ZORG DAT JE WEER BIJ 2de KAR-
350 3AA2 A8       TAY          ;KAKTER BEGINT OP NEXTLIJN.
360 3AA3 C629     DEC #29      ;VERMINDER LIJNTELLER
370 3AA5 D0E8     BNE NEXTLY   ;4 LIJNEN GEHAD ?
380 3AA7 C628     DEC #28      ;VERMINDER PAGE TELLER
390 3AA9 F029     BEQ KLAAR    ;ALLE PAGES GEHAD ANDERS DOORGAAN
400 3AAB E611     INC #11      ;VERHOOG BASIS ADRES
410 3AAD A000     LDY #0
420 3AAF A204     LDX #4        ;PAK 4 LINKSE
430 3AB1 B110     LINKS LDA ($10),Y ;KARAKTERS BOVEN ELKAAR
440 3AB3 9530     STA #30,X   ;EN BERG DIE EVEN
450 3AB5 98       TYA          ;OP OP PAGE ZERO
460 3AB6 18       CLC
470 3AB7 6940     ADC #64
480 3AB9 A8       TAY
490 3ABA CA       DEX          ;ALLE 4 GEHAD
500 3ABB D0F4     BNE LINKS   ;NEE-VOLGENDE ANDERS DOORGAAN
510 3ABD C611     DEC #11      ;VERLAAG BASIS ADRES WEER
520 3ABF A03F     LDY #63
530 3AC1 A204     LDX #4        ;PAK DE 4 KARAKTERS
540 3AC3 B530     RECHTS LDA #30,X   ;UIT PAGE ZERO
550 3AC5 9110     STA ($10),Y ;EN ZET DIE RECHTS
560 3AC7 98       TYA          ;BOVEN ELKAAR 1 PAGE
570 3AC8 18       CLC          ;HOGER
580 3AC9 6940     ADC #64
590 3ACB A8       TAY
600 3ACC CA       DEX
610 3ACD D0F4     BNE RECHTS  ;4 GEHAD? NEE?->VOLGENDE
620 3ACF E611     INC #11      ;VERHOOG BASISADRES WEER
630 3AD1 38       SEC
640 3AD2 B0B8     BCS NEXTPA   ;SPRING ALTIJD
650 3AD4 60       KLAAR RTS    ;TERUG NAAR BASIC.

```



```
1000 REM LICHTKRANT VOOR DOS BASIC
1005 :
1010 REM OORSPRONKELIJK VOOR 48 KARAKTERS
1020 REM J.M.A.HERMANS SCHIPLUIDEN JUNI/JULI 1981
1030 :
1040 REM Aangepast voor 64*32 scherm door Martien Schot
1050 REM VEENENDAAL -JE WEET WEL VAN MOSI-
1055 :
1060 PRINT!(28)
1070 PRINT:PRINT:PRINT:PRINTTAB(12);"LICHTKRANT":PRINT:PRINT
1080 PRINTTAB(10);"EVEN GEDULD !!!"
1090 P=14976
1100 DIMZ(90,2,3),AA(3,3),A$(7),AB$(9)
1110 READR:IFR<0THEN1130
1120 POKEP,R:P=P+1:GOTO1110
1130 FORG=0TO90
1140 FORJ=0TO2
1150 FORK=0TO3
1160 READZ(G,J,K)
1170 NEXTK,J
1180 NEXTG
1190 POKES74,128:POKE575,58
1200 PRINT!(28)
1210 FORI=0TO9:AB$(I)="":NEXTI
1220 PRINT"ANTAL PAGINA'S TEKST":PRINT
1230 INPUT"(ELK 7 REGELS VAN 16 POS.):":N:PRINT
1240 FORI=1TON
1250 FORJ=1TO7
1260 PRINT"
....."
1270 INPUT"TEKST: ";A$(J):PRINT
1280 IFLEN(A$(J))<=15THENFORKL=LEN(A$(J))TO15:A$(J)=A$(J)+" ":NEXTKL
1290 AB$(I)=AB$(I)+A$(J):NEXTJ
1300 AB$(I)=AB$(I)+" "
1310 PRINT!(28):NEXTI
1320 FORL=1TON
1330 FORIJ=1TO112
1340 I=ASC(MID$(AB$(L),IJ,1))-32
1350 FORJ=0TO2
1360 FORK=0TO3
1370 A(J,K)=Z(I,J,K)
1380 POKES4847+K*64,A(J,K)
1390 NEXTK
1400 B=USR(B)
1410 NEXTJ
1420 REM IFI=0THEN1340
1430 REM FORTR=1TO999:NEXTTR
1440 FORK=0TO3:POKES4847+K*64,32:NEXTK
1450 IFIJ=112THEN1470
1460 B=USR(B)
1470 NEXTIJ
1480 IFL<NTHENFORTT=1TO8000:NEXTTT:GOSUB1640
1490 NEXTL
1500 GOSUB1590
```

FEBRUARI 1985

```
1510 FORTT=1T0600
1520 POKE2073,96:POKE57088,239
1530 IFPEEK(57088)=239THENPOKE2073,173:ZZ=1
1550 POKE2073,173:NEXTTT
1560 IFZZ=1THENFORI=1T0200:NEXTI
1570 IFZZ=1THENZZ=0:GOTO1200
1580 PRINT!(28):GOTO1320
1590 PP=55180
1600 B#="ANDERE TEXT ? DAN DRUK OP T"
1610 BL=LEN(B#)
1620 FORI=1TOBL:POKEPP+I,ASC(MID$(B#,I,1)):NEXTI
1630 RETURN
1640 FORF=1T065:B=USR(B)
1650 FORTT=1T070:NEXTTT:NEXTF:RETURN
1660 DATA169,0,133,16,169,208,133,17,169,8
1670 DATA133,40,169,4,133,41,168,1,162,63
1680 DATA177,16,136,145,16,200,200,202,208,246
1690 DATA152,24,105,1,168,198,41,208,235,196,40
1700 DATA240,40,230,17,160,0,162,4,177,16
1710 DATA149,48,152,24,105,64,168,202,208,244
1720 DATA198,17,160,63,162,4,181,48,145,16
1730 DATA152,24,105,64,168,202,208,244,230,17
1740 DATA56,176,184,96
1870 DATA=1
1880 REM SPATIE
1890 DATA32,32,32,32,32,32,32,32,32,32,32
1900 REM!
1910 DATA32,32,32,32,157,157,166,166,32,32,32,32
1920 REM"
1930 DATA32,32,32,32,156,32,32,32,156,32,32,32
1940 REM#
1950 DATA32,166,166,32,156,170,170,168,156,170,170,168
1960 REM$
1970 DATA32,166,165,32,170,170,170,166,154,167,170,32
1980 REM%
1990 DATA157,32,165,32,156,165,168,32,165,168,154,155
2000 REM&
2010 DATA165,166,157,32,169,170,166,155,32,32,170,166
2020 REM'
2030 DATA32,32,32,32,157,32,32,32,32,32,32,32
2040 REM(
2050 DATA32,157,166,32,170,32,167,166,32,32,32,32
2060 REM)
2070 DATA32,32,32,32,166,32,32,166,167,157,170,32
2080 REM*
2090 DATA165,32,165,32,157,169,169,166,165,168,169,32
2100 REM+
2110 DATA32,165,32,32,165,170,157,32,32,154,32,32
2120 REM,
2130 DATA32,32,32,32,32,32,32,166,32,32,156,32
2140 REM-
2150 DATA32,165,32,32,32,154,32,32,32,154,32,32
2160 REM.
2170 DATA32,32,32,32,32,32,32,166,32,32,32,32
```

2180 REM/
2190 DATA32,32,165,32,32,165,168,32,165,168,32,32
2200 REM0
2210 DATA165,157,157,32,155,165,168,155,169,175,157,168
2220 REM1
2230 DATA32,32,32,32,176,157,157,155,32,32,32,168
2240 REM2
2250 DATA165,32,165,166,155,165,168,155,169,170,32,155
2260 REM3
2270 DATA166,32,165,32,155,165,32,155,177,156,157,168
2280 REM4
2290 DATA32,165,166,32,165,168,155,32,156,156,170,168
2300 REM5
2310 DATA157,166,165,32,155,155,32,155,155,169,157,168
2320 REM6
2330 DATA32,157,157,32,170,154,32,155,155,167,157,168
2340 REM7
2350 DATA166,32,32,32,155,165,156,168,177,168,32,32
2360 REM8
2370 DATA165,166,157,32,155,154,32,155,169,170,157,168
2380 REM9
2390 DATA165,166,32,166,155,154,32,155,169,157,170,32
2400 REM:
2410 DATA32,32,32,32,32,166,166,32,32,32,32,32
2420 REM;
2430 DATA32,32,32,32,32,166,157,168,32,32,32,32
2440 REM<
2450 DATA32,165,32,32,165,168,169,32,168,32,32,168
2460 REM=
2470 DATA32,166,166,32,32,155,155,32,32,155,155,32
2480 REM>
2490 DATA32,32,32,32,169,32,165,168,32,169,168,32
2500 REM?
2510 DATA165,32,32,32,155,165,166,166,169,168,32,32
2520 REMAT-SIGN
2530 DATA165,157,157,32,155,157,166,155,169,176,168,155
2540 REMA
2550 DATA32,157,157,166,170,32,155,32,167,157,169,166
2560 REMB
2570 DATA157,157,157,166,155,154,32,155,169,170,157,168
2580 REMC
2590 DATA165,157,157,32,155,32,32,155,169,32,165,166
2600 REMD
2610 DATA157,157,157,166,155,32,32,155,169,157,157,168
2620 REME
2630 DATA157,157,157,166,155,154,32,155,155,167,32,155
2640 REMF
2650 DATA157,157,157,166,155,154,32,32,155,167,32,32
2660 REMG
2670 DATA165,157,157,32,155,32,32,155,155,32,177,155
2680 REMH
2690 DATA157,157,157,166,32,154,32,32,157,157,157,166
2700 REMI
2710 DATA32,32,32,32,177,157,157,155,168,32,32,168

2720 REMJ
2730 DATA32,32,165,32,32,32,32,155,157,157,157,166
2740 REMK
2750 DATA157,157,157,166,32,170,166,32,170,32,167,166
2760 REML
2770 DATA157,157,157,166,32,32,32,155,32,32,32,155
2780 REMM
2790 DATA157,157,157,166,167,157,32,32,170,157,157,166
2800 REMN
2810 DATA157,157,157,166,32,169,32,32,157,157,169,166
2820 REMO
2830 DATA165,157,157,32,155,32,32,155,169,157,157,166
2840 REMP
2850 DATA157,157,157,166,155,154,32,32,169,170,32,32
2860 REMQ
2870 DATA165,157,157,32,155,32,166,155,169,157,170,166
2880 REMR
2890 DATA157,157,157,166,155,154,166,32,169,170,167,166
2900 REMS
2910 DATA165,166,165,32,155,154,32,155,169,167,157,166
2920 REMT
2930 DATA166,32,32,32,155,157,157,166,155,32,32,32
2940 REMU
2950 DATA157,157,157,32,32,32,32,155,157,157,157,166
2960 REMV
2970 DATA157,157,166,32,32,32,167,166,157,157,170,32
2980 REMW
2990 DATA157,157,157,166,32,165,170,32,157,157,154,166
3000 REMX
3010 DATA157,32,165,166,32,169,168,32,157,168,169,166
3020 REMY
3030 DATA157,32,32,32,32,169,157,166,157,168,32,32
3040 REMZ
3050 DATA166,32,165,166,155,165,168,155,177,168,32,155
3060 REM
3070 DATA157,157,157,166,155,156,156,155,155,2,32,155
3080 REM\
3090 DATA165,32,32,32,32,169,32,32,32,32,169,32
3100 REMJ
3110 DATA166,32,32,166,155,32,32,156,161,161,161,155
3120 REM^
3130 DATA32,32,166,32,32,170,32,32,32,167,166,32
3140 REM UNDERSCORE
3150 DATA32,32,32,166,32,32,32,155,32,32,32,155
3155 REM SPATIE
3160 DATA32,32,32,32,32,32,32,32,32,32,32,32
3170 REMa
3180 DATA32,165,157,32,32,155,32,155,32,157,157,166
3190 REMb
3200 DATA157,157,157,166,32,170,167,166,32,169,157,166
3210 REMc
3220 DATA32,165,157,32,32,155,32,155,32,155,32,155

FEBRUARI 1985

3230 REMd
3240 DATA32,165,157,32,32,155,32,155,157,170,170,166
3250 REMe
3260 DATA32,165,157,32,32,155,155,155,32,169,155,166
3270 REMf
3280 DATA32,32,32,32,165,170,157,166,168,167,32,32
3290 REMg
3300 DATA32,157,166,32,154,32,154,154,165,169,169,170
3310 REMh
3320 DATA157,157,157,166,32,155,32,32,32,169,157,166
3330 REMi
3340 DATA32,32,32,32,165,154,157,155,32,32,32,168
3350 REMj
3360 DATA32,32,32,32,32,32,32,154,167,167,156,168
3370 REMk
3380 DATA157,157,157,166,32,165,155,32,165,168,167,166
3390 REMl
3400 DATA32,32,32,32,157,157,157,155,32,32,32,168
3410 REMm
3420 DATA32,157,157,166,32,169,157,166,32,169,157,166
3430 REMn
3440 DATA32,157,157,166,32,155,32,32,32,169,157,166
3450 REMo
3460 DATA32,165,157,32,32,155,32,155,32,169,157,168
3470 REMp
3480 DATA165,157,157,157,165,168,169,32,167,157,170,32
3490 REMq
3500 DATA32,157,166,32,154,32,154,32,165,157,169,157
3510 REMr
3520 DATA32,157,157,166,32,170,32,32,32,168,32,32
3530 REMs
3540 DATA32,165,32,166,32,155,155,155,32,155,169,166
3550 REMt
3560 DATA32,166,32,32,165,169,157,32,32,155,32,168
3570 REMu
3580 DATA32,157,157,32,32,32,32,155,32,157,157,155
3590 REMv
3600 DATA32,157,32,32,32,32,156,166,32,157,156,32
3610 REMw
3620 DATA32,157,157,32,32,32,165,168,32,157,157,168
3630 REMx
3640 DATA32,166,32,166,32,167,170,32,32,170,167,166
3650 REMy
3660 DATA32,32,32,32,32,156,169,154,32,157,170,170
3670 REMz
3680 DATA32,166,32,166,32,155,170,155,32,170,32,155

FEBRUARI 1985

::: Gebruiksaanwijzing MINI-LOGO :::

Dit is een programma om de computer te laten tekenen. Je moet hem dat dan wel heel duidelijk zeggen, dat wil zeggen, je moet hem opdrachten geven in een bepaalde volgorde !

De volgorde wordt aangegeven met nummers. Je begint dus een nummer (tussen 1 en 999) in te typen, bijvoorbeeld 10.

De opdrachten zijn:

KLEUR

(KLEUR=0 betekent: niet tekenen,
elke andere waarde: wel tekenen !)

STAP

(Kleur een hokje in de richting, waarin je gaat.
Dus STAP 10 : kleur 10 hokjes in die richting.)

DRAAI

(verander van richting; je moet dan wel opgeven hoe:
DRAAI 1 : draai 45 graden rechtsom,
DRAAI 2 : draai 90 graden, etc.)

HERHAAL N MAAL

(herhaal de programma regels, die nu volgen tot
"TOT HIER" N maal.)

Iedere programmeerregel wordt pas ingevoerd, als je de
<RETURN> toets indrukt.

Hetzelfde geldt voor de volgende opdrachten (die zonder
regelnummer moeten worden ingevoerd):

DOE

(geeft aan de computer de opdracht het programma
uit te voeren; dit kan overigens wel even duren !)

LIJST

(de computer geeft een lijst van het tot dan toe
ingevoerde programma.)

PRINT

(de computer geeft de lijst op de printer.)

NIEUW

(de computer wist het oude MINI-LOGO programma,
zodat een nieuw programma kan worden gemaakt door
dezelfde persoon.)

STOP

(de computer start het totale programma opnieuw,
zodat een ander ermee kan gaan werken.)

Je kunt programmaregels veranderen door ze opnieuw in te
typen. Bovendien kun je later regels tussenvoegen (daar-
om kun je de regelnummers het best met 10 laten springen.)

Het volgende programma tekent een vierkantje op je scherm:

```
10 KLEUR = 1
20 STAP 8
30 HERHAAL 3 MAAL
40 DRAAI 2
50 STAP 8
60 TOT HIER
```

Je mag ook meerdere opdrachten op een regel schrijven.
Deze moeten dan van elkaar gescheiden worden d.m.v. "/".
Zo zou het vierkant ook getekend kunnen worden met:

```
10 KLEUR = 1 / STAP 8
20 HERHAAL 3 MAAL / DRAAI 2
30 STAP 8 / TOT HIER
```

Bovendien mogen alle opdrachten, met uitzondering van
STOP en NIEUW, worden afgekort tot de eerste letter. B.v.:

```
10 K1/S8/H3/D2/S8/T
```

Je programma wordt er zo echter niet duidelijker op !

Wijzigingen in de listing (pag. EC 23 e.v.):

```
6010 NC=0;CC=0;TE(NI)=0;FORCL=1TO etc.
6170 NI=NI-1
```

M. BAETEN.

LOW-COST GRAPHICS voor iedere OSI
met viditel character generator
Kees Hoogeland

Januari 1985

M.b.v. de Viditel charactergenerator (Ko van Ekeren) en de hierbij gepresenteerde software is het mogelijk om zéér snel punten, lijnen en series lijnstukken te plotten en ook te UN-plotten.

De hier gegeven software plot alléén naar het scherm. Voor de figuren heb ik een aanvullende plot-routine gebruikt die met dubbele resolutie voor X en voor Y naar een buffer plot, om zodoende de mogelijkheden van de OKI-printer volledig te benutten.

In de BASIC-versie is de syntax en het gebruik aangegeven. Ik heb hier gebruik gemaakt van enige Cegmon/Exbasic uitbreidingen die echter te vervangen zijn door Poke11,..:Poke12,..etc. Echter vooral in combinatie met de grafische mogelijkheden van Exbasic zijn we nu in staat om de mooiste figuren snél! te realiseren.

Het is interessant te zien dat de USR-functie hier vergezeld gaat van twee, vier of zelfs nog meer variabelen,zoals in Exbasic.

Het is ook mogelijk om de plot-syntax in Exbasic op te nemen ,evt. met SQUARE.. en CIRCLE..

Fig.1 en fig.3 zijn gemaakt met het (bijna niet gewijzigde) Apple H-plot programma "Landscapes" uit Byte. Fig.2 is gemaakt met het gewijzigde programma "Hypocycloids" van Ed Morris en fig.4 ontstond na wat experimenteren met vierkanten en cirkels.

Literatuur:

OSI/UK newsletter Bit-mapped display S.Graham
idem MX-80 graphics M.Taploy
First book of Osi by J.Willams and G.Dorner
Micro the 6502-journal 6/80 Ed.Morris
Byte Sept.1984 "Landscapes" D.Cooper



Fig.1

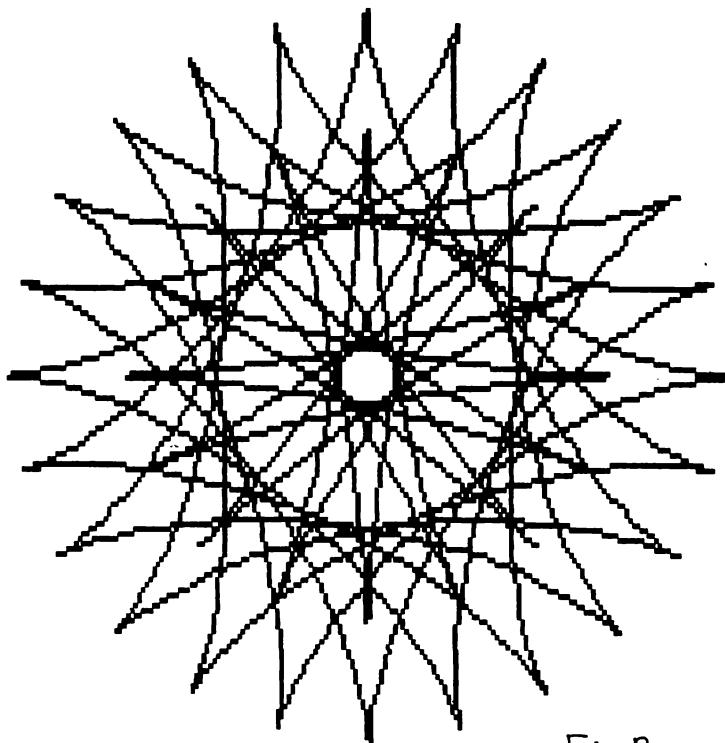


Fig.2.

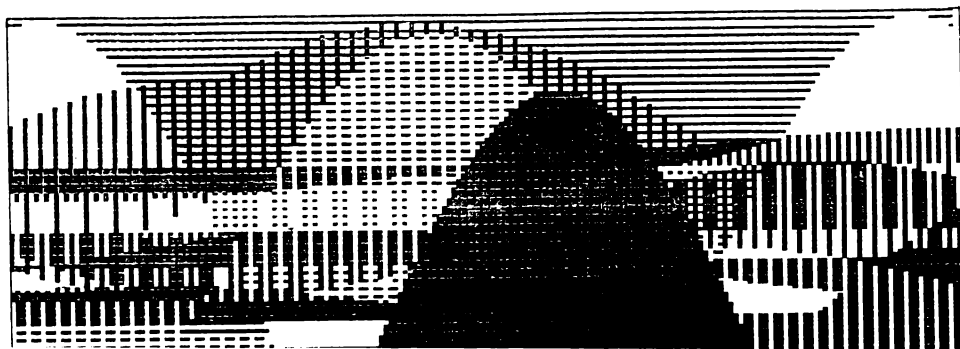


Fig3 ■

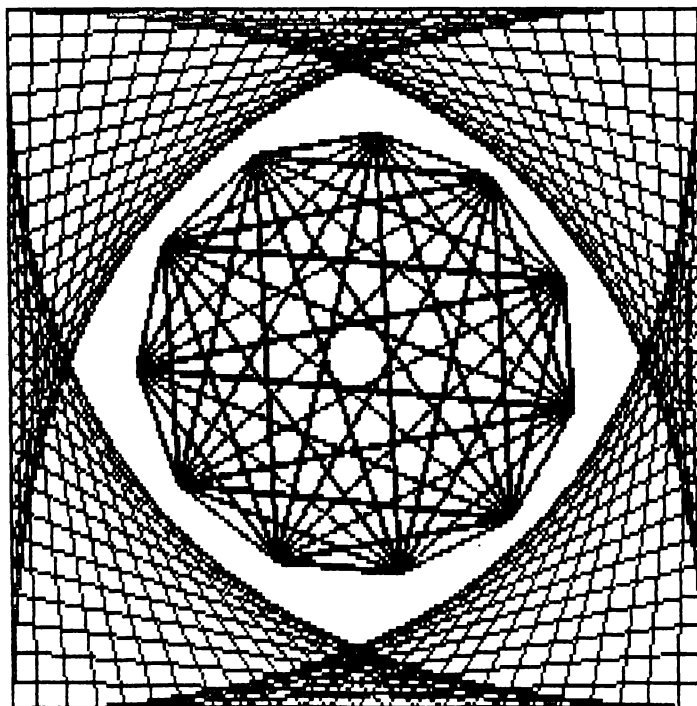


Fig.4.

```

10      ;
20      ;
30      ;SYNTAX:
40      ;Q = USR(1) P,Q TO R,S TO T,U
50      ;
60      ; Steve Graham 1982
70      ;Modified for viditel/DKI-graphics
80      ;and screen/printer compatibility
90      ;by Kees Hooseland
100     ;
110    02E0= X1 = $02E0
120    02E1= Y1 = X1+1
130    02E2= X2 = X1+2
140    02E3= Y2 = X1+3
150    02E4= MX = X1+4           MX = ABS(X2-X1)
160    02E5= MY = X1+5           MY = ABS(Y2-Y1)
170    02E6= SX = X1+6           SX = SGN(X2-X1)
180    02E7= SY = X1+7           SY = SGN(Y2-Y2)
190    02E8= DX = X1+8           current deviation from line
200    02E9= DY = X1+9           " " " "
210     ;
220    02EA= PLOTPM = X1+10      plotting parameter (white/black)
230    02EB= TPL = X1+11        temporary variable
240    02EC= TMPH = X1+12       " "
250     ;
260    00EE= PLOTLO = $EE      must be page zero
270    00EF= PLOTTH = $EF
280     ;
290    00C0= DRIGL = $C0        bottom left
300    00D7= DRIGH = $D7        of screen
310     ;
320    00BC= CHRGET= $00BC      get character
330    AE88= FCERR = $AE88      FC ERROR caller
340    AC01= COMMA = $AC01      SN ERROR if next char
350     ;                          ; is not a comma
360    009D= TD = $9D           BASIC token
370     ;
380    5000  * = $5000
390     ;
400    5000 AEAC  USR  LDX $AC      get USR argument
410    5002 8EEA02 STX PLOTPM
420    5005 20C200 JSR CHRGET+6    haal laatste bewerkte byte
430    5008 C99D  CMP #TO          IF no first pair THEN
440    500A F019  BEQ PAIR2        get second pair
450     ;
460    500C 204250 PAIR1 JSR EVAL7   7-bit evaluate expression
470    500F 8EE002 STX X1
480    5012 2001AC JSR COMMA        SN err if no comma
490    5015 204250 JSR EVAL7
500    5018 8DE102 STA Y1
510     ;
520    501B 20C200 JSR CHRGET+6
530    501E C99D  CMP #TO
540    5020 F003  BEQ PAIR2        plot 1 point if
550    5022 4C0451 JMP PLOT        no 'TO'
560     ;
570    5025 20BC00 PAIR2 JSR CHRGET
580    5028 204250 JSR EVAL7
590    502B 8DE102 STA X2

```

```

600 502E 2001AC      JSR  CDMMA
610 5031 204250      JSR  EVAL7
620 5034 8DE302      STA  Y2
630
640 5037 20AC50      JSR  LINE           plot line X1,Y1 to X2,Y2
650 503A 20C200      JSR  CHRGET+6
660 503D C95D        CMP  #TD           test for 'chaining'
670 503F F0E4        BEQ  PAIR2
680 5041 60          RTS
690
700 5042 20AEB3 EVAL7 JSR  #BSAE         8-bit argument -> X
710 5045 8A          TXA
720 5046 1003        BPL  OK
730 5048 4CB8AE      JMP  FCERR         too large
740 504B 60          OK
750
760
770
780
790
800 504C ADE002 LINE  LDA  X1           prevent error when
810 504F CDE202      CMP  X2           X1=X2 and Y1=Y2
820 5052 D00C        BNE  UNEQ
830 5054 ADE102      LDA  Y1
840 5057 CDE302      CMP  Y2
850 505A D004        BNE  UNEQ
860 505C 200451      JSR  PLOT
870 505F 60          RTS
880 5060 A201 UNEQ   LDX  #1
890 5062 38          SEC
900 5063 ADE202      LDA  X2
910 5066 EDE002      SBC  X1
920 5069 0006        BCS  POSMX         X2-X1 positive
930 506B 49FF        EOR  #FF
940 506D 6901        ADC  #1
950 506F A2FF        LDX  #FF
960 5071 8DE402 POSMX STA  MX
970 5074 8DE902      STA  DY
980 5077 8EE602      STX  SX
990
1000 507A A201      LDX  #1
1010 507C 38          SEC
1020 507D ADE302      LDA  Y2
1030 5080 EDE102      SBC  Y1
1040 5083 0006        BCS  POSMY
1050 5085 49FF        EOR  #FF
1060 5087 6901        ADC  #1
1070 5089 A2FF        LDX  #FF
1080 508B 8DE502 POSMY STA  MY
1090 508E 8DE802      STA  DX
1100 5091 8EE702      STX  SY
1110
1120
1130 5094 200451 REPEAT JSR  PLOT         top of main loop
1140
1150 5097 ADE902      LDA  DY
1160 509A CDE802      CMP  DX
1170 509D 901C        BCC  DXGTDY       DX > DY
1180
1190 509F 18          DXLEDY CLC           IF DX <= DY THEN
1200 50A0 ADE802      LDA  DX
1210 50A3 6DE502      ADC  MY           DX = DX + MY
1220 50A6 8DE802      STA  DX

```

```

1230 50A9 18          CLC
1240 50AA ADE002     LDA X1          X1 = X1 +SX
1250 50AD 6DE602     ADC SX
1260 50B0 8DE002     STA X1
1270 50B3 ADE802     LDA DX
1280 50B6 CDE902     CMP DY
1290 50B9 9029       BCC DYGRIT      IF DX ( DY (still) THEN
1300                  ;                               GOTO 'DYGRIT'
1310                  ;
1320                  ;
1330 50BB 18          DXGTDY CLC          IF DX > DY THEN -
1340 50BC ADE902     LDA DY
1350 50BF 6DE402     ADC MX          DY + DY + MX
1360 50C2 8DE902     STA DY
1370 50C5 18          CLC
1380 50C6 ADE102     LDA Y1
1390 50C9 6DE702     ADC SY          Y1 = Y1 + SY
1400 50CC 8DE102     STA Y1
1410                  ;
1420 50CF ADE802     LDA DX
1430 50D2 CDE902     CMP DY
1440 50D5 900D       BCC DYGRIT
1450                  ;
1460 50D7 EDE902     DXGRT  SBC DY          IF DX >= DY THEN -
1470 50DA 8DE802     STA DX          DX = DX - DY
1480 50DD A900       LDA #0          DY = 0
1490 50DF 8DE902     STA DY
1500 50E2 F00F       BEQ RPTST
1510                  ;
1520 50E4 38          DYGRIT SEC          ELSE -
1530 50E5 ADE902     LDA DY
1540 50E8 EDE802     SBC DX          DY = DY - DX
1550 50EB 8DE902     STA DY
1560 50EE A900       LDA #0          DX = 0
1570 50F0 8DE802     STA DX
1580                  ;
1590 50F3 ADE002     RPTST  LDA X1          REPEAT UNTIL X1=X2 & Y1=Y2
1600 50F6 CDE202     CMP X2
1610 50F9 D099       BNE REPEAT
1620 50FB ADE102     LDA Y1
1630 50FE CDE302     CMP Y2
1640 5101 D091       BNE REPEAT
1650                  ;
1660 5103 60          RTS
1670                  ;
1680                  ;
1690                  ; Plot a point at (x1,y1) -
1700                  ; on if PLOTM (<)0 , off if PLOTM = 0
1710                  ;
1720                  ;
1730                  ; Calculate character address
1740                  ; in graphic display
1750                  ;
1760 5104 AEE002     PLOT  LDX X1
1770 5107 8A          TXA
1780 5108 4A          LSR A          X/2-->ACC
1790 5109 18          CLC          P = ORIGIN + X/2 - 64*Y/3
1800 510A 69C0       ADC #ORIGL
1810 510C 85EE       STA PLOTLD    X/2+#C0-->PLOTLO
1820 510E A900       LDA #0          #D7-->PLOTHI
1830 5110 69D7       ADC #ORIGH
1840 5112 85EF       STA PLOTHI
1850

```

1860	5114	38		SEC	
1870	5115	A900		LDA #0	
1880	5117	8DEC02		STA TMPH	Y/3 in TMPH maakt 256*
1890	511A	8DEB02		STA Tmpl	daarom 2* ROR naar Tmpl
1900	511D	ADE102		LDA Y1	
1910	5120	E903	DIV3	SBC #3	
1920	5122	3005		BMI REST	rest = 0, 1 of 2
1930	5124	EEEC02		INC TMPH	quotient van Y/3
1940	5127	D0F7		BNE DIV3	branch always if here
1950	5129	6903	REST	ADC #3	
1960	512B	AB		TAY	rest bewaren voor bitvalue
1970	512C	4EEC02		LSR TMPH	/4 maakt *64
1980	512F	6EEB02		ROR Tmpl	
1990	5132	4EEC02		LSR TMPH	
2000	5135	6EEB02		ROR Tmpl	Tmpl/TMPH=Y/3*64
2010			;		
2020	5138	38		SEC	
2030	5139	A5EE		LDA PLOTLO	
2040	513B	EDEB02		SBC Tmpl	
2050	513E	85EE		STA PLOTLO	
2060	5140	A5EF		LDA PLOTHI	
2070	5142	EDEC02		SBC TMPH	
2080	5145	85EF		STA PLOTHI	PLDTLO/HI=BASE+X/2-Y/3*64
2090			;		
2100			;		
2110			;		Calculate bit value
2120			;		in character
2130			;		
2140	5147	98		TYA	rest van Y/3
2150	5148	F009		BEQ TIEN	
2160	514A	4A		LSR A	
2170	514B	B003		BCS VIER	
2180	514D	A901		LDA #1	
2190	514F	2C		.BYTE \$2C	
2200	5150	A904	VIER	LDA #4	
2210	5152	2C		.BYTE \$2C	
2220	5153	A910	TIEN	LDA #\$10	
2230	5155	8DEB02		STA Tmpl	Tmpl bevat pixel Y
2240	5158	8A		TXA	
2250	5159	A000		LDY #0	ivm. (PLOTLO),Y in PLT ev.
2260	515B	4A		LSR A	
2270	515C	ADEB02		LDA Tmpl	
2280	515F	9001		BCC PLT	carry bevat pixel X
2290	5161	0A		ASL A	voor pixel Y * 2
2300			;		
2310	5162	AEEA02	PLT	LDX PLOTPLM	set or unset pixel
2320	5165	D005		BNE NONEG	as required
2330	5167	49FF		EOR #\$FF	
2340	5169	31EE		AND (PLOTLO),Y	
2350	516B	2C		.BYTE \$2C	
2360			;		
2370	516C	11EE	NONEG	ORA (PLOTLO),Y	
2380	516E	31EE	STORE	STA (PLOTLO),Y	
2390	5170	60		RTS	
2400			;		

```

35000 :::::::::::::::::::::RUN35020
35001 REM *****
35002 REM ***** LOW-COST GRAPHICS for OSI *****
35003 REM ***** Kees Hoogeland Januari 1985 *****
35004 REM *****
35005 REM .
35020 REM * GRAPHICS(50P) 128X96 ALLEEN SCHERM *
35021 DATA20480,20848:REM: *5000/5170
35022 DATA166,172,142,234,2,32,194,0,201,157,240,25,32,66,80
35023 DATA142,224,2,32,1,172,32,66,80,141,225,2,32,194,0,201
35024 DATA157,240,3,76,4,81,32,188,0,32,66,80,141,226,2,32,1
35025 DATA172,32,66,80,141,227,2,32,76,80,32,194,0,201,157,240
35026 DATA228,96,32,174,179,138,16,3,76,136,174,96,173,224,2
35027 DATA205,226,2,208,12,173,225,2,205,227,2,208,4,32,4,81
35028 DATA96,162,1,56,173,226,2,237,224,2,176,6,73,255,105,1
35029 DATA162,255,141,228,2,141,233,2,142,230,2,162,1,56,173
35030 DATA227,2,237,225,2,176,6,73,255,105,1,162,255,141,229
35031 DATA2,141,232,2,142,231,2,32,4,81,173,233,2,205,232,2,144
35032 DATA28,24,173,232,2,109,229,2,141,232,2,24,173,224,2,109
35033 DATA230,2,141,224,2,173,232,2,205,233,2,144,41,24,173,233
35034 DATA2,109,228,2,141,233,2,24,173,225,2,109,231,2,141,225
35035 DATA2,173,232,2,205,233,2,144,13,237,233,2,141,232,2,169
35036 DATA0,141,233,2,240,15,56,173,233,2,237,232,2,141,233,2
35037 DATA169,0,141,232,2,173,224,2,205,226,2,208,153,173,225
35038 DATA2,205,227,2,208,145,96,174,224,2,138,74,24,105,192
35039 DATA133,238,169,0,105,215,133,239,56,169,0,141,236,2,141
35040 DATA235,2,173,225,2,233,3,48,5,238,236,2,208,247,105,3
35041 DATA168,78,236,2,110,235,2,78,236,2,110,235,2,56,165,238
35042 DATA237,235,2,133,238,165,239,237,236,2,133,239,152,240
35043 DATA9,74,176,3,169,1,44,169,4,44,169,16,141,235,2,138,160
35044 DATA0,74,173,235,2,144,1,10,174,234,2,208,5,73,255,49,238
35045 DATA44,17,238,145,238,96
35046 REM ** EINDE DATA GRAPHICS(50P) 128X96 ALLEEN SCHERM **
35047 READ AD,EN:FORX=ADTOEN:READY:POKEX,Y:NEXT
35100 FILLS128*PUSR5+4096+HOME
35110 PRINT"GRAPHICS BELADEN"
35120 PRINT"SYNTAX: Q=USR(1) P,Q (TD R,S enz.) voor Plotten"
35130 PRINT" Q=USR(0) P,Q (TD R,S enz.) voor unplotten"
35140 PRINT" VOORAF FILLS128 GEVEN i.p.v.CLS"
35150 REM VOORBEELD
35160 Q=USR(9) 0,0 TO 63,63 TO 127,0
35170 PAUSE 200
35180 Q=USR(0) 0,0 TO 63,63 TO 127,0
35190 PAUSE 200
35200 GOTO 35160
Ready

```

Plotteroutine voor Superboard

Deze routine is bedoeld voor diegenen die een Viditel karakter-set in hun OSI hebben ingebouwd. Het is eigenlijk een uitbreiding van de Basic-instructieset met drie nieuwe instructies die op een bepaald coördinaat van het scherm een blokje tekenen. De routine vertaalt de coördinaten in viditelkarakters, waardoor het lijkt alsof men elk blokje apart kan aansturen. Omdat een viditelkarakter bestaat uit 6 blokjes (2*3) bedraagt de totale resolutie 2 maal het aantal karakters per regel (van uw systeem) bij 3 maal het aantal regels. (0,0) staat linksboven. De nieuwe opdrachten zijn:

ZON x,y zet een blokje op coördinaat (x,y)
ZEND x,y wis een blokje uit op (x,y)
ZIF x,y inverteer het blokje op (x,y)

De nieuwe commando's moeten in een programma staan; in de direct mode werken ze niet. Verder moeten ze worden voorafgegaan door een dubbele punt.

De routine wordt opgestart door de Control-C vector om te leggen op \$0269 dat gebeurt door:

POKE 541,2 zet plotteroutine aan
POKE 541,252 dat is de trace uit de H.W.Monitor
POKE 541,255 uit

Na een warme start moet er opnieuw worden opgestart.

Een schuine lijn over het beeld kan men dus maken met:

```
100 FOR A=0 TO 50 : ZON A,A : NEXT A
```

-

Het geheel lijkt erg veel op de SET en RESET opdrachten van de Tandy TRS-80. Met deze routine is het nu mogelijk om grafische programma's voor deze computer om te schrijven voor OSI.

Henk Schaer,
Graswinckelstraat 2F,
2613 FW DELFT.

Plotter routine voor Superboard.

CONTROL-C

0230	A588	LDA	88	Welke mode?
0232	C9FF	CMP	i FF	is het immediate?
0234	D003	BNE	0239	Nee, dan routine in
0236	4C9BFF	JMP	FF9B	anders weg naar cntrl-c
0239	A5C3	LDA	C3	Save de huidige
023B	85EE	STA	EE	basicpointers
023D	A5C4	LDA	C4	op de adressen
023F	85EF	STA	EF	EE en EF
0241	20C200	JSR	00C2	Haal teken op
0244	C93A	CMP	i 3A	is het een dubbele punt?
0245	D01E	BNE	0255	nee, dan ook weg
0248	20BC00	JSR	00BC	haal volgend teken
024B	C95A	CMP	i 5A	is het een Z?
024D	D00F	BNE	025E	nee, dan ook weg
024F	20BC00	JSR	00BC	haal volgende (token)
0252	C98A	CMP	i 8A	is het IF?
0254	F013	BEQ	0269	ja, verder
0256	C990	CMP	i 90	is het ON?
0258	F00F	BEQ	0269	ja, verder
025A	C980	CMP	i 80	is het END?
025C	F00B	BEQ	0269	ja, verder
025E	A5EE	LDA	EE	herstel
0260	85C3	STA	C3	weer
0262	A5EF	LDA	EF	de oude
0264	85C4	STA	C4	basicpointer
0266	4C9BFF	JMP	FF9B	en naar cntrl-c

PLOTTER

0269	85E7	STA	E7	zet token weg
026B	A900	LDA	i 00	reset
026D	85E3	STA	E3	overflow hi
026F	208802	JSR	0288	doe plotterroutine
0272	4C3902	JMP	0239	en naar basic terug

DEEL DCOR DRIE

0275	26E2	ROL	E2	rol 8 bits
0277	CA	DEX		naar E3
0278	300D	BMI	0287	
027A	26E3	ROL	E3	
027C	38	SEC		delen
027D	A5E3	LDA	E3	door herhaald
027F	E5E4	SBC	E4	aftrekken
0281	90F2	BCC	0275	
0283	85E3	STA	E3	
0285	BOEE	BCS	0275	
0287	60	RTS		en terug

 TEKENROUTINE

0288	20BC00	JSR	00BC	zet pointer verder
028B	20FCB3	JSR	B3FC	haal coördinaten X,Y
029E	36E2	STX	E2	Y richting in E2
0290	A511	LDA	11	X richting
0292	85E0	STA	E0	in E0
0294	A901	LDA	i 01	zet masker
0296	35E1	STA	E1	klaar voor schuiven
0298	46E0	LSR	E0	deel x door twee
029A	9003	BCC	029F	even-oneven?
029C	18	CLC		oneven,
029D	06E1	ASL	E1	dus masker schuiven
029F	A903	LDA	i 03	delen door 3
02A1	85E4	STA	E4	
02A3	A208	LDX	i 08	8 bits opschuiven
02A5	207502	JSR	0275	en gaan delen
02A8	A6E3	LDX	E3	rest van deling
02AA	F007	BEQ	02B3	rest 0, niet schuiven
02AC	06E1	ASL	E1	rest 1, dus 2 bits
02AE	06E1	ASL	E1	opschuiven
02B0	CA	DEX		rest 2?
02B1	DOF9	BNE	02AC	dan nog een keer
02B3	18	CLC		
02B4	A5E0	LDA	E0	haal x-richting
02B6	6D2202	ADC	0222	tel kantlijn op
02B9	8D2902	STA	0229	en verzet cursor
02BC	A5E2	LDA	E2	haal y-richting
02BE	6D2402	ADC	0224	tel bovenkant op
02C1	8D2A02	STA	022A	en verzet cursor
02C4	A9AD	LDA	i AD	zet videohulp
02C6	8D2B02	STA	022B	op LDA
02C9	208BF8	JSR	F88B	bereken cursor
02CC	202B02	JSR	022B	en haal teken van scherm
02CF	AA	TAX		save accu
02D0	3002	BMI	02D4	viditelteken
02D2	A280	LDX	i 80	pak een vidispatie
02D4	A98D	LDA	i 8D	zet videohulp
02D6	8D2B02	STA	022B	op STA
02D9	A5E7	LDA	E7	haal token weer op
02DB	C98A	CMP	i 8A	is het ZIF?
02DD	F011	BEQ	02F0	ja, verwerken
02DF	3005	BMI	02E6	het is ZEND
02E1	8A	TXA		teken weer in accu
02E2	05E1	ORA	E1	OR met masker
02E4	300D	BMI	02F3	spring altijd, verwerk
02E6	86E5	STX	E5	teken in E5
02E8	A9FF	LDA	i FF	inverteren
02EA	45E1	EOR	E1	EXOR met masker
02EC	25E5	AND	E5	clear bep. bit
02EE	3003	BMI	02F3	spring altijd, verwerk
02FO	8A	TXA		teken weer in accu
02F1	45E1	EOR	E1	flip bit
02F3	202B02	JSR	022B	zet op scherm
02F6	A9FF	LDA	i FF	zet cursor weg
02F8	8D2D02	STA	022D	van zichtbaar scherm
02FB	60	RTS		na gedane arbeid....

```

10          ;***** G.WILLEMSSEN *****
20          ;HULP ROUTINE OM INKEY$ NA TE BOOTSEN
30          ;
40          ;
50          ;om de instructie te initialiseren
60          ; DISK!"60 7800" intypen
70          ;
80          ;
90          ;GEBRUIK: A$=CHR$
100         ;
110        ;WERKING: er wordt gekeken of het token van
115        ; CHR$ gebruikt wordt. Is dit het
120        ; geval, dan wordt gecontroleerd of
125        ; een "(" volgt. Is dit zo, dan
130        ; wordt de normale CHR$(expressie)
135        ; gebruikt. In het geval van
140        ; A$=CHR$ wordt een karakter van het
145        ; keyboard gehaald en teruggegeven
150        ; aan A$ (of ieder andere willekeur-
160        ; rige string variabele)
170        ;
180        ;
190        ;
200 7800    *$=7800 ; OF IEDER ANDER WILLEKEURIG ADRES
210        ;
220        ;
230 7800 08  PHP      ; INITIALISATIE ROUTINE
240        ; KAN NIET ! VIA POKE INSTRUCTIE
250 7801 D8   CLD
260 7802 A90E  LDA #START
270 7804 8D4E0E STA $0E4E
280 7807 A978  LDA #START/256
290 7809 8D4F0E STA $0E4F
300 780C 28   PLP
310 780D 60   RTS
320 780E 20C000 START JSR $00C0
330 7811 E080  CPX #$80 ; INDIEN CHR$ TOKEN, KIJK
340        ; DAN OF ER EEN "(" VOLGT.
350 7813 F001  BEQ *+3
360 7815 60   RTS
370 7816 C928  CMP #' ('
380 7818 F0FB  BEQ *-3
390 781A 68   PLA
400 781B 68   PLA
410 781C 68   PLA ; FULL RETURN ADRES EN TOKEN
420        ; VAN DE STACK
430 781D AC0423 LDY $2304
440 7820 AE0323 LDX $2303
450 7823 EB   INX
460 7824 D001  BNE *+3
470 7826 C8   INY
480 7827 86A2  STX $A2
490 7829 84A3  STY $A3
500 782B 20A100 JSR $00A1 ; LEES EEN KARAKTER
505        ; VAN HET KEYBOARD
510 782E AA   TAX
520 782F 206A15 JSR $156A ; EN GEEF DE ASCII WAARDE TERUG
530        ; VIA DE NORMALE CHR$ ROUTINE

```

Het menen van BASIC programma's onder OS 65D 3.3

In het 65D manual staat aangedreven dat dit zou kunnen door gebruik te maken van indirect files. Wie dit wel eens geprobeerd heeft, zal gemerkt hebben dat het enige resultaat is dat het systeem crasht. Toch is het heel eenvoudig BASIC programma's te menen. Het is mogelijk met behulp van het zosenaamde MEMORY I/O DEVICE de tekst van (een deel van) een programma tijdelijk in vrij RAM op te slaan en deze tekst op een later tijdstip terug te lezen alsof hij op het toetsenbord wordt ingetikt.

Voorbeeld

Stel dat de computer over 36 K beschikt (\$0000-\$8FFF). We willen twee programma's samenvoeden die beide op disk staan.

- Programma AA is 6 K groot
- Programma BB is 3 K groot

Ga nu als volgt te werk:

Creëer eerst een disk file CC van 5 tracks (10 K).

Tik de volgende commando's in:

```
DISK!"LO BB           (laad kleinste programma)
DISK!"MEM 8000,8000   (zet memory-pointers op besin vrij RAM)
LIST#5               (zet de tekst van programma BB in het vrije
                    RAM)
PRINT#5,"einde"      (of een andere tekst die als commando een
                    SN ERROR zou opleveren)
DISK!"LO AA          (laad het grootste programma)
DISK!"IO 10          (input uit vrij RAM; nu verschijnen de
                    programmaregels van BB stuk voor stuk op
                    het beeldscherm en worden ingevoerd
                    tussen de regels van programma AA, het
                    woord "EINDE" geeft een foutmelding en
                    zorgt ervoor dat de input weer gewoon van
                    het toetsenbord komt)
DISK!"PU CC          (zet het samengevoegde programma weg op
                    disk)
```

Deze methode kan ook gebruikt worden om een aantal op disk opgeslagen subroutines (bijvoorbeeld BASICODE-subroutines) aan een programma toe te voegen. In plaats van LIST#5 type je dan in: LIST#5,100-200 LIST#5,400-600.

Op dezelfde wijze kun je een hele serie regels uit een programma verwijderen: LIST alleen de regels die je over wil houden naar #5, tik NEW in en haal de regels terug met DISK!"IO 10.

Een laatste toepassing is het veranderen van het aantal DATA-BUFFERS van een bestaand programma. Start na LIST#5 het programma BEXEC* op en kies via optie 7 het gewenste aantal buffers. Vervolgens weer NEW en DISK!"IO 10.

hans de jons
 prinses beatrixlaan 543
 2284 av rijswijk
 telefoon 070-937296

november 1984

```
10 ; MACHINE-TAAL ANALYSER
20 ; UIT ELEKTUUR
30 ;
40 ; ENIGSZINS VOOR SUPERBOARD BEWERKT
50 ; DOOR Jos Burghouts
60 ;
70 ; Met dit programma kan een ML-programma
80 ; stap voor stap worden geanalyseerd. Het is
90 ; alleen niet nodig hardware-veranderingen
100 ; aan te brengen ! Het programma vraagt na
110 ; het starten (in deze vorm op $300) vanaf
120 ; welk adres de analyse moet starten. Als
130 ; het adres niet juist wordt ingetoetst,
140 ; moet na de vraag Ready PRINT een andere
150 ; letter dan de 'Y' worden gegeven, alleen
160 ; na Y gaat het programma verder.
170 ; Er wordt telkens een scherm vol weergege-
180 ; ven met een toelichtende kop erboven.
190 ; Om naar een volgend scherm te gaan kan
200 ; iedere willekeurige letter worden gegeven,
210 ; BEHALVE de P, de S en de N.
220 ; P: de save-mode wordt aangezet, zodat alle
230 ; informatie naar een printer kan worden
240 ; gestuurd.
250 ; N: de save-mode wordt weer uitgezet, het P
260 ; commando wordt weer ongedaan gemaakt.
270 ; S: het programma springt terug naar de
280 ; monitor, zodat de uitvoering stopt.
290 ;
300 ; De werking van dit programma is als volgt:
310 ; Er wordt een dubbele boekhouding bijgehou-
320 ; den, zowel namelijk van het analyserende
330 ; programma als van het te analyseren pro-
340 ; gramma. Alle relevante informatie over
350 ; registers, stack, etc. wordt na iedere op-
360 ; dracht bijgewerkt en op het scherm gezet.
370 ; Om zekere te zijn, dat maar een opdracht
380 ; per keer wordt afgewerkt, wordt na de
390 ; opcode en de bijbehorende gegevens een 0
400 ; gezet, die de computer in een interrupt-
410 ; positie brengt. De computer krijgt als het
420 ; ware een software break. Standaard wordt
430 ; dan naar de NMI-plek gesprongen op $01CO,
440 ; waar de start van het analyse programma
450 ; staat. Let dus op: het te analyseren pro-
460 ; gramma wordt in feite ook opdracht voor
470 ; opdracht uitgevoerd !!! Er kunnen konflik-
480 ; ten ontstaan bij het zeropage en page 2
490 ; geheugengebruik.
500 ; Ook zullen grote fouten in het geanaly-
510 ; seerde programma soms de computer op TILT
520 ; kunnen zetten, maar dat is meestal ook de
530 ; bedoeling. Wel moet erop worden gelet, dat
540 ; geen data-velden met betekenisloze infor-
```

```

550 ; matie zoals tekst etc. worden doorgelicht,
560 ; omdat ook dan de uitkomst vaak een op
570 ; stop gezette computer is !!
580 ;
590 ; De basisopzet van dit programma komt uit
600 ; ELEKTUUR. Er is alleen een aanpassing ge-
610 ; maakt voor de OSI em het gebruiksgemak is
620 ; flink verhoogd.
630 ; Voor de mensen met een wat beperkt geheu-
640 ; gen is ook nog een HEX-dump gegeven, dan
650 ; hoeven ze niet via de assembler en een
660 ; reallocatie alle tekst in te voeren.
670 ; Het is trouwens ook nog de bedoeling om
680 ; de komplette tekst op een OSI bandje/
690 ; schijfje uit te brengen, samen met een
700 ; aantal andere machinetaalprogramma's,
710 ; allemaal in assembler editor Print formaat
720 ;
730 1D00= START1=$1D00
740 1D00 * =START1
750 01C0= NMIVEC=$01C0
760 0048= PSTACK=$48
770 0049= NSP =PSTACK+1 Nieuwe stackpointer
780 004B= PSTELL=PSTACK+3 Pseudo stackpointer-teller
790 0050= PSACC =PSTACK+8 Pseudo accumulator
800 0051= PSY =PSTACK+9 Pseudo Y-register
810 0052= PSX =PSTACK+10 Pseude X-register
820 0053= LENINS=PSTACK+11 Instruktielengte
830 0055= PSSTR =PSTACK+13 Pseudo stackpointer
840 0078= REGELT=$78 Aantal schermregels
850 0071= PROGST=$71 Startadres programma
860 FFEE= OUTPUT=$FFEE
870 FD00= KEYB =$FD00
880 FE93= HEX =$FE93
890 FE00= MONIT =$FE00
900 FF96= SAVE =$FF96
910 0205= SAVEFL=$0205
920 ;
930 1D00 4C031D START1 JMP BEGIN
940 1D03 58 BEGIN CLI ; ZET ALLES KLAAR
950 1D04 A900 LDA ##00 ; 16 MAAL 00 IN DE
960 1D06 A00F LDY ##0F ; PSEUDOSTACK, WWAR HET
970 1D08 994800 INIT1 STA PSTACK,Y ; TE ANALYSEREN PROGR. MET
980 1D08 88 DEY ; REGISTERS KOMT
990 1D0C DOFA BNE INIT1
1000 1D0E B478 STY REGELT ; ZET REGELTELLER OP 0
1010 1D10 202A1D JSR VRAAG ; HAAL STARTADRES OP
1020 1D13 20C71D JSR KOPREG ; GEEF DE KOPREGEL
1030 1D16 A94C LDA ##4C ; ZET $4C=JMP
1040 1D18 8DC001 STA NMIVEC ; OP DE NMI-VECTOR
1050 1D1B AD4B1F LDA LABEL ; HAAL HET START2 ADRES
1060 1D1E 8DC101 STA NMIVEC+1
1070 1D21 AD4C1F LDA LABEL+1
1080 1D24 8DC201 STA NMIVEC+2
1090 1D27 4C461E JMP START3 ; GA HET HOOFDPROGR, UITVOERE
1100 ;
1110 1D2A 20381D VRAAG JSR MESSAG ; VRAAG HET BEGINADRES
1120 1D2D 204A1D JSR GET2 ; TWEE GELDIGE HEXBYTES

```

```

1130 1D30 206C1D      JSR READY
1140 1D33 C959        CMP #'Y           ; ZOLANG GEEN Y INGETOETST
1150 1D35 D0F3        BNE VRAAG        ; OPNIEUW VRAGEN
1160 1D37 60          RTS
1170                  ;
1180 1D38 A0FF MESSAG LDY ##FF           ; DRUK MEDEDELING
1190 1D3A C8          LM.1  INY
1200 1D3B B94D1F      LDA TEKST,Y       ; DRUK TEKSTTABEL AF
1210 1D3E 3005        BMI LM.2          ; LAATSTE LETTER
1220 1D40 20EEFF      JSR OUTPUT
1225 1D43 D0F5        BNE LM.1          ; SFRING ALTIJD
1230 1D45 297F        LM.2  AND ##7F      ; HAAL HOOGSTE BIT WEG
1240 1D47 4CEEFF      JMP OUTPUT        ; EN KLAAR
1245                  ;
1250 1D4A A201 GET2   LDX ##01          ; ALS HEX5 UIT TOOLKIT
1255 1D4C 20601D GETP JSR GETNIB        ; HAAL EEN NIBBLE OP
1260 1D4F 0A          ASL A
1270 1D50 0A          ASL A
1280 1D51 0A          ASL A
1290 1D52 0A          ASL A
1300 1D53 9571        STA PROGST,X
1310 1D55 20601D      JSR GETNIB
1320 1D58 1571        ORA PROGST,X
1330 1D5A 9571        STA PROGST,X
1340 1D5C CA          DEX
1350 1D5D 10ED        BPL GETP
1360 1D5F 60          RTS
1370                  ;
1380 1D60 2000FD GETNIB JSR KEYB          ; HAAL EEN TOETS OP
1390 1D63 20EEFF      JSR OUTPUT        ; DRUK DIE AF
1400 1D66 2093FE      JSR HEX           ; TESTEN
1410 1D69 30F5        BMI GETNIB        ; TERUG INDIEN GEEN HEX
1420 1D6B 60          RTS
1430                  ;
1440 1D6C A013 READY  LDY ##13
1450 1D6E 203A1D      JSR LM.1
1460 1D71 2000FD      JSR KEYB
1470 1D74 60          RTS
1480                  ;
1490 1D75 0930 ASCII  ORA ##30          ; MAAK ASCII VAN RECHTER NIBB
1500 1D77 C93A        CMP ##3A
1510 1D79 9002        BCC AF
1520 1D7B 6906        ADC ##06
1530 1D7D 4CEEFF AF    JMP OUTPUT
1540                  ;
1550 1D80 4B          HEXUIT PHA           ; ZET HEXGETAL OP SCHERM
1560 1D81 4A          LSR A
1570 1D82 4A          LSR A
1580 1D83 4A          LSR A
1590 1D84 4A          LSR A           ; LINKER NIBBLE=RECHTS
1600 1D85 20751D      JSR ASCII
1610 1D88 68          PLA
1620 1D89 290F        AND ##0F          ; ALLEEN RECHTER NIBBLE
1630 1D8B 4C751D      JMP ASCII
1640                  ;
1650 1D8E A920 SPATIE LDA #'           ; DRUK EEN SPATIE AF
1660 1D90 4CEEFF      JMP OUTPUT
1670                  ;

```

```

1680 1D93 A90D REGEL LDA ##0D ; GEEF CR/LF EN TEL ZE
1690 1D95 20EEFF JSR OUTPUT ; TOT AAN 21
1700 1D98 A90A LINEFD LDA ##0A
1710 1D9A 20EEFF JSR OUTPUT
1720 1D9D A57B LDA REGELT ; KIJK HOEVEEL REGELS
1730 1D9F C915 CMP #21 ; ZOLANG < 21 KLAAR
1740 1DA1 D021 BNE REG.1
1750 1DA3 2000FD JSR KEYB ; WACHT OP TOETS
1760 1DA6 C953 CMP #'S ; EEN S INGETOETST PRINT
1770 1DAB D003 BNE CH1
1780 1DAA 4C00FE JMP MONIT ; JA, NAAR MONITOR !
1790 1DAD C950 CH1 CMP #'P ; EEN P (PRINTER)
1800 1DAF D003 BNE CH2
1810 1DB1 2096FF JSR SAVE ; JA SAVEFLAG AANZETTEN
1820 1DB4 C94E CH2 CMP #'N ; EEN N INGETOETST PRINT
1830 1DB6 D005 BNE CHOP
1840 1DB8 CE0502 CH3 DEC SAVEFL ; VERMINDER MET 1 TOTDAT
1850 1DBB D0FB BNE CH3
1860 1DBD 20C71D CHOP JSR KOPREG ; GEEF EEN KOPJE
1870 1DC0 A900 LDA ##00 ; EN REGELTELLER OP 00
1880 1DC2 857B STA REGELT
1890 1DC4 E67B REG.1 INC REGELT ; VOLGENDE SCHERMREGEL
1900 1DC6 60 RTS
1910 ;
1920 1DC7 A000 KOPREG LDY ##00 ; GEEF KOPJE WEER
1930 1DC9 B96C1F KOP.1 LDA KOPJE,Y
1940 1DCC 20EEFF JSR OUTPUT
1950 1DCF CB INY
1960 1DD0 C059 CPY ##59
1970 1DD2 D0F5 BNE KOP.1
1980 1DD4 A000 LDY ##00 ; Y MOET 0 ZIJN !!!!!
1990 1DD6 60 RTS
2000 ;
2010 1DD7 8550 START2 STA PSACC ; BEGIN INTERRUPT PROGR.
2020 1DD9 68 PLA
2030 1DDA 8555 STA PSSTR ; PSEUDO STACKPOINTER
2040 1DDC 68 PLA ; EN OORSPRONKELIJK ADRES
2050 1DDD 68 PLA ; V/D/ PROGRAMMA-TELLER
2060 1DDE 8451 STY PSY
2070 1DE0 8652 STX PSX
2080 1DE2 BA TSX
2090 1DE3 8649 STX NSP ; NIEUWE STACKPOINTER
2100 1DE5 D8 CLD
2110 1DE6 58 CLI
2120 1DE7 A003 LDY ##03
2130 1DE9 B94A00 ST2.1 LDA PSTACK+2,Y ; ZET PSEUDOSTACK
2140 1DEC 20801D JSR HEXUIT
2150 1DEF 208E1D ST2.2 JSR SPATIE
2160 1DF2 CB INY
2170 1DF3 C006 CPY ##06
2180 1DF5 B00F BCS ST2.4
2190 1DF7 A54B LDA PSTELL
2200 1DF9 D009 BNE ST2.3
2210 1DFB 208E1D JSR SPATIE
2220 1DFE 208E1D JSR SPATIE
2230 1E01 4CEF1D JMP ST2.2
2240 1E04 C64B ST2.3 DEC PSTELL
2250 1E06 C009 ST2.4 CPY ##09

```



```

2260 1E08 D0DF          BNE ST2.1
2270 1E0A A555          LDA PSSTR           ; PSEUDOSTACK-POINTER
2280 1E0C 29CF          AND #0CF
2290 1E0E 8548          STA PSTACK
2300 1E10 A208          LDX #08           ; GA HET STATUSREGISTER
2310 1E12 0648          ST2.5             ASL PSTACK         ; UITLEZEN IN CARRY-BIT
2320 1E14 9004          BCC ST2.6
2330 1E16 A931          LDA #1           ; ER STOND EEN 1
2340 1E18 D002          BNE ST2.7
2350 1E1A A92E          ST2.6             LDA #1
2360 1E1C 20EEFF        ST2.7             JSR OUTPUT
2370 1E1F CA            DEX
2380 1E20 D0F0          BNE ST2.5
2390 1E22 208E1D        JSR SPATIE
2400 1E25 A549          LDA NSP           ; DRUK NIEUWE STACKPOINTER
2410 1E27 20801D        JSR HEXUIT
2420 1E2A A93E          LDA #'>
2430 1E2C 20EEFF        JSR OUTPUT
2440 1E2F BA            TSX               ; HAAL STACKPOINTER OP
2450 1E30 C9FF          CMP #0FF         ; BIJ STACK LEEG, KLAAR
2460 1E32 B012          BCS START3
2470 1E34 68            PLA               ; DRUK HEM OOK AF
2480 1E35 854B          STA PSTELL        ; EVEN BEWAREN VOOR DADELIJK
2490 1E37 20801D        JSR HEXUIT
2500 1E3A E0FE          CPX #0FE
2510 1E3C B005          BCS ST2.8
2520 1E3E 68            PLA
2530 1E3F 4B            PHA               ; HERSTEL STACK WEER
2540 1E40 20801D        JSR HEXUIT
2550 1E43 A54B          ST2.8             LDA PSTELL
2560 1E45 4B            PHA
2570 1E46 A000          START3           LDY #00           ; HET EIGENLIJKE HOOFDPROGR.
2580 1E48 20931D        JSR REGEL         ; VOLGENDE REGEL
2590 1E4B A572          LDA PROGST+1
2600 1E4D 20801D        JSR HEXUIT
2610 1E50 A571          LDA PROGST
2620 1E52 20801D        JSR HEXUIT
2630 1E55 208E1D        JSR SPATIE
2640 1E58 B171          LDA (PROGST),Y ; HAAL VOLGENDE OPCODE
2650 1E5A 8CB01E        STY INPUT+1
2660 1E5D 8CB11E        STY INPUT+2
2670 1E60 844F          STY PSTACK+7
2680 1E62 844E          STY PSTACK+6
2690 1E64 202B1F        JSR LENACC        ; HAAL AANTAL BYTES BIJ
2700 1E67 8453          STY LENINS        ; DEZE OPCODE, BEWAREN
2710 1E69 98            TYA
2720 1E6A 854B          STA PSTELL
2730 1E6C CA            DEX PSTELL
2740 1E6D 88            ST3.1            DEY
2750 1E6E B171          LDA (PROGST),Y ; HAAL VERVOLGBYTES
2760 1E70 99AF1E        STA INPUT,Y
2770 1E73 994D00        STA PSTACK+5,Y
2780 1E76 98            TYA
2790 1E77 D0F4          BNE ST3.1
2800 1E79 E671          ST3.2            INC PROGST
2810 1E7B D002          BNE ST3.3
2820 1E7D E672          INC PROGST+1
2830 1E7F C653          ST3.3            DEC LENINS

```

```

2840 1E81 D0F6      BNE ST3.2      ; TOTDAT ALLES OP IS
2850 1E83 A54D      LDA PSTACK+5
2860 1E85 290F      AND ##0F      ; DECODEER SPECIALE
2870 1E87 D012      BNE ST3.4      ; OPCODES
2880 1E89 A54D      LDA PSTACK+5
2890 1E8B C920      CMP ##20
2900 1E8D F024      BEQ ST3.6      ; VOER JSR-OPCODE UIT
2910 1E8F C940      CMP ##40
2920 1E91 F029      BEQ ST3.7      ; DE RTI-OPCODE
2930 1E93 C960      CMP ##60
2940 1E95 F028      BEQ ST3.8      ; DE RTS-OPCODE
2950 1E97 2910      AND ##10      ; DECODEER RELATIEVE SPRONG
2960 1E99 D059      BNE ST3.12
2970 1E9B A54D      ST3.4 LDA PSTACK+5
2980 1E9D C94C      CMP ##4C
2990 1E9F F027      BEQ ST3.9      ; JMP-OPCODE
3000 1EA1 C96C      CMP ##6C
3010 1EA3 F035      BEQ ST3.11     ; JMP-INDIRECT-OPCODE
3020 1EA5 A652      ST3.5 LDX PSX
3030 1EA7 A451      LDY PSY
3040 1EA9 A555      LDA PSSTR
3050 1EAB 4B        PHA
3060 1EAC A550      LDA PSACC
3070 1EAE 2B        PLP
3080 1EAF 00        INPUT  .BYTE 0,0,0,0
3080 1EB0 00
3080 1EB1 00
3080 1EB2 00
3090
3095 ; IN INPUT WORDT STEEDS DE
3100 ; VOLGENDE INSTRUKTIE MET
3110 ; BIJBEHORENDE DATAVELDEN
3120 ; VAN HET TE ONDERZOEKEN
3130 ; PROGRAMMA GEZET EN UITGE-
3140 ; VOERD.
3150 ; HIERDOOR IS DIT PROGRAMMA
3160 ; NIET IN EEN EPROM TE
3165 ; ZETTEN !!
3170 1EB3 A571      ST3.6 LDA PROGST
3180 1EB5 4B        PHA
3190 1EB6 A572      LDA PROGST+1
3200 1EB8 4B        PHA
3210 1EB9 4CC81E    JMP ST3.9
3220 1EBC 68        ST3.7 PLA
3230 1EBD 8555      STA PSSTR
3240 1EBF 68        ST3.8 PLA
3250 1EC0 8572      STA PROGST+1
3260 1EC2 68        PLA
3270 1EC3 8571      STA PROGST
3280 1EC5 4CD21E    JMP ST3.10
3290 1EC8 ADB01E    ST3.9 LDA INPUT+1
3300 1ECB 8571      STA PROGST
3310 1ECD ADB11E    LDA INPUT+2
3320 1ED0 8572      STA PROGST+1
3330 1ED2 A900      ST3.10 LDA ##00
3340 1ED4 8DAF1E    STA INPUT
3360 1ED7 4CA51E    JMP ST3.5      ; WERK REST VAN DE REGEL AF
3370 1EDA ADB01E    ST3.11 LDA INPUT+1 ; SIMULEER INDIREKTE JMP

```

```

3380 1EDD 8571          STA PROGST
3390 1EDF ADB11E       LDA INPUT+2
3400 1EE2 8572          STA PROGST+1
3410 1EE4 A000         LDY ##00
3420 1EE6 B171         LDA (PROGST),Y
3430 1EE8 AA           TAX
3440 1EE9 C8           INY
3450 1EEA B171         LDA (PROGST),Y
3460 1EEC 8572          STA PROGST+1
3470 1EEE 8A           TXA
3480 1EEF 8571          STA PROGST
3490 1EF1 4CD21E       JMP ST3.10
3500 1EF4 A555         ST3.12 LDA PSSTR      ; VERWERK BRANCHES
3510 1EF6 48           PHA
3520 1EF7 A54D         LDA PSTACK+5
3530 1EF9 8DFD1E       STA ST3..
3540 1EFC 28           PLP
3550 1EFD D003         ST3.. BNE ST3.13   ; VOER DE BRANCH UIT
3560                   ;
3570                   ; NOG EEN REDEN WAAROM
3575                   ; DEZE CODE NIET IN
3580                   ; EPROM TE ZETTEN IS
3590                   ;
3600 1EFF 4C121F       JMP ST3.14
3610 1F02 58         ST3.13 CLI
3620 1F03 D8         CLD
3630 1F04 ADB01E       LDA INPUT+1
3640 1F07 3011         BMI ST3.15
3650 1F09 18         CLC
3660 1F0A 6571         ADC PROGST
3670 1F0C 8571         STA PROGST
3680 1F0E 9002         BCC ST3.14
3690 1F10 E672         INC PROGST+1
3700 1F12 A900         ST3.14 LDA ##00
3710 1F14 8DB01E       STA INPUT+1
3720 1F17 4C9B1E       JMP ST3.4
3730 1F1A 18         ST3.15 CLC
3740 1F1B 6571         ADC PROGST
3750 1F1D 8571         STA PROGST
3760 1F1F B0F1         BCS ST3.14
3770 1F21 C672         DEC PROGST+1
3780 1F23 90ED         BCC ST3.14
3790 1F25 4C931D       JMP REGEL
3800 1F28 A001         LENACC LDY ##01      ; BEREKEN HET AANTAL DATA
3810 1F2A C900         CMP ##00      ; VELDEN BIJ DEZE OPCODE
3820 1F2C F01A         BEQ LEN.1
3830 1F2E C940         CMP ##40
3840 1F30 F016         BEQ LEN.1
3850 1F32 C960         CMP ##60
3860 1F34 F012         BEQ LEN.1
3870 1F36 A003         LDY ##03
3880 1F38 C920         CMP ##20
3890 1F3A F00C         BEQ LEN.1
3900 1F3C 291F         AND ##1F
3910 1F3E C919         CMP ##19
3920 1F40 F006         BEQ LEN.1

```

```

3930 1F42 290F      AND #*0F
3940 1F44 AA        TAX
3950 1F45 BCC61F    LDY LENTAB,X
3960 1F48 8456     LEN.1  STY PSSTR+1
3970 1F4A 60       RTS
3980                ;
3990 1F4B D71D     TABEL  .WORD START2
4000 1F4D 53       TEKST  .BYTE 'Startadres Print',#87
4010 1F5E 0D                .BYTE $0D,$0A,'Ready PRINT',#87
4030 1F6C 1A       KOPJE  .BYTE $1A,'6502 - Tracer (S=Stop/'
4040 1FB5 50                .BYTE 'P=Printer/N=No-printer)',#0D,$0A
4050 1F9E 41                .BYTE 'Adr. -Instr.- :A :Y :X '
4060 1FB5 4E                .BYTE 'NV11DIZV Stack ',#0D,$0A
4080 1FC6 02       LENTAB  .BYTE 2,2,2,1,2,2,2,1,1,2,1,1
4090 1FD2 03                .BYTE 3,3,3,3,0
    
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1D00	4C	03	1D	58	A9	00	A0	0F	99	48	00	88	D0	FA	84	78
1D10	20	2A	1D	20	C7	1D	A9	4C	8D	C0	01	AD	4B	1F	8D	C1
1D20	01	AD	4C	1F	8D	C2	01	4C	46	1E	20	38	1D	20	4A	1D
1D30	20	6C	1D	C9	59	D0	F3	60	A0	FF	C8	B9	4D	1F	30	05
1D40	20	EE	FF	D0	F5	29	7F	4C	EE	FF	A2	01	20	60	1D	0A
1D50	0A	0A	0A	95	71	20	60	1D	15	71	95	71	CA	10	ED	60
1D60	20	00	FD	20	EE	FF	20	93	FE	30	F5	60	A0	11	20	3A
1D70	1D	20	00	FD	60	09	30	C9	3A	90	02	69	06	4C	EE	FF
1D80	48	4A	4A	4A	4A	20	75	1D	68	29	0F	4C	75	1D	A9	20
1D90	4C	EE	FF	A9	0D	20	EE	FF	A9	0A	20	EE	FF	A5	78	C9
1DA0	15	D0	21	20	00	FD	C9	53	D0	03	4C	00	FE	C9	50	D0
1DB0	03	20	96	FF	C9	4E	D0	05	CE	05	02	D0	FB	20	C7	1D
1DC0	A9	00	85	78	E6	78	60	A0	00	B9	6C	1F	20	EE	FF	C8
1DD0	C0	59	D0	F5	A0	00	60	85	50	68	85	55	68	68	84	51
1DE0	86	52	BA	86	49	D8	58	A0	03	B9	4A	00	20	80	1D	20
1DF0	8E	1D	CB	00	06	B0	0F	A5	4B	D0	09	20	8E	1D	20	8E
1E00	1D	4C	EF	1D	C6	4B	C0	09	D0	DF	A5	55	29	CF	85	48
1E10	A2	08	06	48	90	04	A9	31	D0	02	A9	2E	20	EE	FF	CA
1E20	D0	F0	20	8E	1D	A5	49	20	80	1D	A9	3E	20	EE	FF	BA
1E30	C9	FF	B0	12	68	85	4B	20	80	1D	E0	FE	B0	05	68	48
1E40	20	80	1D	A5	4B	48	A0	00	20	93	1D	A5	72	20	80	1D
1E50	A5	71	20	80	1D	20	8E	1D	B1	71	8C	B0	1E	BC	B1	1E
1E60	84	4F	84	4E	20	28	1F	84	53	98	85	48	CA	88	B1	71
1E70	99	AF	1E	99	4D	00	98	D0	F4	E6	71	D0	02	E6	72	C6
1E80	53	D0	F6	A5	4D	29	0F	D0	12	A5	4D	C9	20	F0	24	C9
1E90	40	F0	29	C9	60	F0	28	29	10	D0	59	A5	4D	C9	4C	F0
1EA0	27	C9	6C	F0	35	A6	52	A4	51	A5	55	48	A5	50	28	00
1EB0	00	00	00	A5	71	48	A5	72	48	4C	C8	1E	68	85	55	68
1EC0	85	72	68	85	71	4C	D2	1E	AD	B0	1E	85	71	AD	B1	1E
1ED0	85	72	A9	00	8D	AF	1E	4C	A5	1E	AD	B0	1E	85	71	AD
1EE0	B1	1E	85	72	A0	00	B1	71	AA	C8	B1	71	85	72	8A	85
1EF0	71	4C	D2	1E	A5	55	48	A5	4D	8D	FD	1E	28	D0	03	4C
1F00	12	1F	58	D8	AD	B0	1E	30	11	18	65	71	85	71	90	02
1F10	E6	72	A9	00	8D	B0	1E	4C	9B	1E	18	65	71	85	71	B0
1F20	F1	C6	72	90	ED	4C	93	1D	A0	01	C9	00	F0	1A	C9	40
1F30	F0	16	C9	60	F0	12	A0	03	C9	20	F0	0C	29	1F	C9	19
1F40	F0	06	29	0F	AA	BC	C6	1F	84	56	60	D7	1D	53	74	61

```

5 REM CHARACTERSET VOOR PLOTTER
10 RESTORE
12 PRINT:PRINT
15 INPUT "Geef characterhoogte in mm.":H
20 Z=49424
30 READ X,R
50 IF R<16 THEN I=0
60 IF R>16 THEN I=16
65 POKEZ,I
67 K=H*.33:X=X*K
68 FOR F=@T025:NEXT
70 FORA=1TOX:POKEZ,R:POKEZ,I:NEXT
97 GOTO 30
100 DATA25,20,5,21,10,17,5,25,25,24,15,6,5,2,20,17,15,8,10,1
110 DATA30,20,15,17,5,25,5,24,5,26,15,18,15,1,5,25,5,24,5,26,15,18
115 DATA30,1
120 DATA20,1,5,4,5,26,10,18,5,22,20,20,5,21,10,17,5,25,25,8,10,1
130 DATA30,20,15,17,5,25,20,24,5,26,15,18,30,1
140 DATA30,20,20,17,15,8,5,2,15,18,15,8,20,17,10,1
150 DATA30,20,20,17,15,8,5,2,15,18,15,8,30,1
160 DATA15,5,5,17,10,24,5,26,10,18,5,22,20,20,5,21,10,17,5,25,25,8
165 DATA10,1
170 DATA30,20,15,8,20,17,15,4,30,24,10,1
180 DATA10,17,5,18,30,20,5,18,10,17,30,8,10,1
190 DATA5,4,5,25,10,17,5,21,25,20,30,8,10,1
200 DATA30,20,20,1,20,26,5,21,15,25,10,1
210 DATA30,4,30,24,20,17,10,1
220 DATA30,20,15,25,15,21,30,24,10,1
230 DATA30,20,20,25,20,4,30,24,10,1
240 DATA5,1,5,22,20,20,5,21,10,17,5,25,20,24,5,26,10,18,25,1
250 DATA30,20,15,17,5,25,5,24,5,26,15,18,30,1,15,8
260 DATA5,1,5,22,20,20,5,21,10,17,5,25,20,24,5,26,10,18,7,5,5,7,5,25
265 DATA10,1
270 DATA30,20,15,17,5,25,5,24,5,26,15,18,5,17,15,25,10,1
280 DATA5,4,5,25,10,17,5,21,5,20,5,22,10,18,5,22,5,20,5,21,10,17
285 DATA5,25,25,8,10,1
290 DATA10,1,30,20,10,18,20,17,30,8,10,1
300 DATA30,4,25,24,5,25,10,17,5,21,25,20,30,8,10,1
310 DATA30,4,20,24,10,25,10,21,20,20,30,8,10,1
320 DATA30,4,30,24,15,21,15,25,30,20,30,8,10,1
330 DATA10,21,10,20,10,22,20,1,10,26,10,24,10,25,10,1
340 DATA30,4,10,25,10,21,10,26,20,24,20,1
350 DATA30,4,20,17,5,24,20,26,5,24,20,17,10,1
355 DATA5,1,5,22,20,20,5,21,10,17,5,25,20,24,5,26,10,18,7,2,3,4,24,21
357 DATA27,8,7,1
360 DATA25,4,5,21,30,24,5,18,10,17,10,1
370 DATA25,4,5,21,10,17,5,25,5,24,5,26,10,18,5,26,10,24,20,17,10,1
380 DATA25,4,5,21,10,17,5,25,5,24,5,26,10,18,10,17,5,25,5,24,5,26
385 DATA10,18,5,22,5,9,25,1
390 DATA20,1,10,4,20,18,5,20,15,21,30,24,15,1
400 DATA15,17,5,21,5,20,5,22,15,18,15,20,20,17,30,8,10,1
410 DATA15,4,15,17,5,25,5,24,5,26,10,18,5,22,20,20,5,21,10,17,15,1
415 DATA30,8
420 DATA30,4,20,17,5,24,10,26,15,24,20,1
430 DATA5,1,5,22,5,20,5,21,5,22,5,20,5,21,10,17,5,25,5,24,5,26,10,18
435 DATA10,17,5,25,5,24,5,26,10,18,25,1
440 DATA5,4,5,25,10,17,5,21,20,20,5,22,10,18,5,26,5,24,5,25,15,17
445 DATA10,1,15,8

```

```
5 :::::::::::::::::::::::::::::::::::::::::::::RUN27
10 REM PLOT ROUTINE
20 REM COMPUTER BIJ VOORKEUR LATEN WERKEN OP 2 MHZ
27 DIMA(1000)
30 PRINTCHR$(26)
31 Z1=61441:GOTO35
32 Z1=49424
35 PRINT
40 INPUT"Geef de waarden voor X en Y ":X,Y
42 INPUT"Pen op of neer ( 0/N ) ":J$
45 GOTO 700
50 GOSUB 340
60 PRINT
65 IF J$="0"THEN R1=R1-16
66 IF J$="0"THEN R2=R2-16
70 PRINTR1;R2
80 PRINT
90 PRINT"Ogenblikje geduld s.v.p."
95 PRINT
100 X=ABS(X):Y=ABS(Y)
120 FOR T1=1TOX:Z=Y/X*T1:GOSUB210:NEXT
130 FOR T1=1TOX:T2=T1-1:A(T2)=A(T1)-A(T2):NEXT
140 FOR T1=0TOX-1
150 IFA(T1)=0THEN190
160 A(T1)=R1
170 NEXT
180 GOTO251
190 A(T1)=R2
200 GOTO170
210 D=INT(Z):IFZ-D>=.5THEN220:X1=D:GOTO230
220 X1=D+1
230 A(T1)=X1
240 RETURN
251 IF J$="N"THEN P=16:GOTO300
252 IF J$="0"THEN P=0:GOTO300
253 GOTO 250
300 FORT5=0TO9
310 FORT1=0TOX-1:POKEZ1,P:POKEZ1,A(T1):NEXT:NEXTT5
330 POKEZ1,0:GOTO30
340 IF X/Y=> 1 THEN GOTO 470
350 IF X/Y > 0 THEN GOTO 500
360 IF X/Y<-1 THEN GOTO 560
370 IF X/Y < 0 THEN GOTO 600
380 IF R=1 THEN R1=21 : R2=20
390 IF R=2 THEN R1=21 : R2=17
400 IF R=3 THEN R1=25 : R2=17
410 IF R=4 THEN R1=25 : R2=24
420 IF R=5 THEN R1=26 : R2=24
430 IF R=6 THEN R1=26 : R2=18
440 IF R=7 THEN R1=22 : R2=18
450 IF R=8 THEN R1=22 : R2=20
460 RETURN
```

```
470 IF X<0 THEN 490
480 R=2:GOTO380
490 R=6:GOTO380
500 IF X<0 THEN 530
510 R=1:GOTO540
530 R=5
540 GOSUB 650
550 GOTO380
560 IF X<0 THEN 580
570 R=3:GOTO380
580 R=7:GOTO380
590 GOTO380
600 IF X<0 THEN 630
610 R=4:GOSUB650
620 GOTO380
630 R=8:GOSUB650
640 GOTO380
650 X5=Y:Y=X:X=X5:RETURN
700 IF X=0 AND Y=0 THEN 1000
702 IF ABS(X)=ABS(Y) THEN 1200
705 IF X<>0 AND Y<>0 THEN 50
710 IF Y=0 THEN 900
720 IF X=0 THEN GOSUB 650
800 IF X>0 THEN R1=20
810 IF X<0 THEN R1=24
820 GOTO 1500
900 IF X>0 THEN R1=17
910 IF X<0 THEN R1=18
920 GOTO 1500
1000 PRINT:PRINT:PRINT"FOUTIEVE INVOER"
1010 FORT=0TO1000:NEXT
1020 GOTO30
1200 IF X/Y > 0 THEN 1300
1210 IF X/Y < 0 THEN 1400
1300 IF X > 0 THEN R1=21
1310 IF X < 0 THEN R1=26
1315 PRINTR1
1320 GOTO1500
1400 IF X > 0 THEN R1=25
1410 IF X < 0 THEN R1=22
1415 PRINTR1
1420 GOTO1500
1500 X=ABS(X)
1510 FOR T1=0 TO X-1:A(T1)=R1:NEXT
1520 GOTO 251
```

OK

F O R M U C A L C

BETEKENIS/FORMULE	WAARDE
X0=STRAAL	X0= 1
X1=PI	X1= 3.141592
X2= XA=OMTREK CIRKEL	X2= 0
X3= XB=OPP.CIRKEL	X3= 0
X4= XC=OPP.BOL	X4= 0
X5= XD=INHOUD BOL	X5= 0
X6=	X6= 0
X7=	X7= 0
X8=	X8= 0
X9=	X9= 0
XA=X0*X1*2	XA= 6.283184
XB=XA*X0/2	XB= 3.141592
XC=XB*4	XC= 12.566368
XD=XC*X0/3	XD= 4.18878933
XE=	XE= 0
XF=	XF= 0
XG=	XG= 0
XH=	XH= 0
XI=	XI= 0
XJ=	XJ= 0

OPDRACHT: F/C/I/P/S P

MET DIT PROGRAMMA KUNT U FORMULES INTIKKEN. U KUNT DEZE FORMULES LATEN UITREKENEN IN DE FORMULES ZIJN X0 T/M X9 MOGELIJKE CONSTANTEN, OMDAT U AAN DEZE EEN WAARDE KUNT TOEKENNEN. XA T/M XJ ZIJN VARIABELEN. IN DE REGELS, DIE MET XA T/M XJ BEGINNEN, WORDEN FORMULES INGETIKT. DE REGELS, DIE MET X0 T/M X9 BEGINNEN KUNT U GEBRUIKEN VOOR INFORMATIE BETREFFENDE DE FORMULES.

- FORMULEREN (F): IN DEZE TOESTAND ACCEPTEERT HET PROGRAMMA FORMULES. EEN EERDER INGETIKTE FORMULE WORDT GECOPIEERD INDIEN U OP DE BETREFFENDE REGEL HET TEKEN > INTIKT, GEVOLGD DOOR <RETURN>.
- CALCULEREN (C): IN DEZE TOESTAND VERWACHT HET PROGRAMMA WAARDEN VOOR DE CONSTANTEN. NA INVOER HIERVAN WORDEN DE WAARDEN VAN XA T/M XJ BEREKEND EN OP HET SCHERM WEERGEGEVEN. OOK HIER KAN > HELPEN, NU OM CONSTANTEN TE COPIEREN.
- INCREMENTEREN (I): NA HET INTIKKEN VAN I WORDT X0 MET 1 VERHOOGD EN DE BEREKENING HERHAALD.
- PRINTEN (P): NA HET INTOETSEN VAN <P> WORDT HET SCHERMBEELD OP DE PRINTER UITGEPRINT.
- STOPPEN (S): NA HET INTOETSEN VAN <S> WORDT HET PROGRAMMA VERLATEN.


```
1 GOTO13
2 ::::::::::::::::::::::::::::::::::::::::::::
3 ::::::::::::::::::::::::::::::::::::::::::::
4 ::::::::::::::::::::::::::::::::::::::::::::
5 ::::::::::::::::::::::::::::::::::::::::::::
6 ::::::::::::::::::::::::::::::::::::::::::::
7 ::::::::::::::::::::::::::::::::::::::::::::
8 ::::::::::::::::::::::::::::::::::::::::::::
9 ::::::::::::::::::::::::::::::::::::::::::::
10 ::::::::::::::::::::::::::::::::::::::::::::
11 ::::::::::::::::::::::::::::::::::::::::::::
12 RETURN
13 REM REGELS 1 T/M 13 NIET WISSEN !!!
20 POKE13050,74:POKE2797,32
30 POKE2888,0:POKE8722,0
40 REM FORMUCALC
50 REM EEN PROGRAMMA
60 REM VAN JOHN HERMANS
70 REM SCHIPLUIDEN
80 REM AUGUSTUS 1983
100 PRINT!(28)
110 PRINT!(17,8,5);"*** F O R M U C A L C ***"
112 PRINT!(17,13,8);"* FORMULEREN"
113 PRINT!(17,13,10);"* CALCULEREN"
114 PRINT!(17,13,12);"* INCREMENTEREN"
115 PRINT!(17,13,14);"* PRINTEN"
116 PRINT!(17,13,17);"EEN PROGRAMMA VAN"
117 PRINT!(17,13,19);"JOHN HERMANS"
118 PRINT!(17,13,22);"AUGUSTUS 1983"
120 FORI=0T09
130 FORJ=14988T015025
140 POKEJ+I*45,58
150 NEXTJ,I
160 TRAP33000
165 TRAPO
170 DIME$(9),F$(9),X$(9),Y$(9),Y(9),KK(9)
180 FORI=0T09:READX$(I):NEXTI
190 DATAA=,XB=,XC=,XD=,XE=,XF=,XG=,XH=,XI=,XJ=
200 AB$="FCIPS"
205 GOSUB34900:PRINT!(28)
208 PRINT!(17,8,5);"*** F O R M U C A L C ***"
210 PRINT!(17,8,9);"1. INFORMATIE GEWENST"
220 PRINT!(17,8,11);"2. GEEN INFORMATIE NODIG"
230 PRINT!(17,14,14);"UW KEUZE 1/2 ";
240 POKE2073,96:POKE57088,127
250 IFPEEK(57088)=127THENGOSUB34000:POKE2073,173:GOTO280
260 IFPEEK(57088)=191THENPOKE2073,173:GOTO280
270 POKE2073,173:GOTO240
280 GOSUB32000
290 PRINT!(17,0,26);"OPDRACHT: F/C/I/P/S";
300 INPUTAC$
305 A=0
310 FORI=1T05
320 IFAC$=MID$(AB$,I,1)THENA=I
330 NEXTI
```

```

340 DNAGOSUB20000,24000,25000,26000,35000
350 GOTO280
500 PRINT!(17,5,25);"VOOR VERVOLG DRUK OP SPATIEBALK"
510 POKE2073,96;POKE57088,253
520 IFPEEK(57088)=239THENPOKE2073,173:RETURN
530 POKE2073,173;GOTO510
20000 FO=20000:REM INFORMATIE
20010 FORI=0TO9:C$=""
20020 PRINT!(17,3,I+5);:INPUTC$
20030 IFC$<>">"THENC$(I)=C$
20040 PRINT!(17,3,I+5);SPC(20)
20050 PRINT!(17,3,I+5);E$(I)
20060 NEXTI
20100 FF=20100:REM FORMULES
20110 FORI=0TO9:KK(I)=0:C$=""
20120 PRINT!(17,3,16+I);:INPUTC$
20130 IFC$=">"THENKK(I)=1:GOTO20140
20135 F$(I)=C$
20140 PRINT!(17,3,16+I);SPC(20)
20150 PRINT!(17,3,16+I);F$(I)
20160 NEXTI:GOSUB20200
20170 PRINT!(17,0,26);:INPUT"IS DIT O.K. ? (J/N) ";J$
20180 PRINT!(17,0,26);SPC(25):IFLEFT$(J$,1)="J"THENRETURN
20190 GOTO20000
20200 FORI=0TO9
20210 IFKK(I)=1THEN20260
20220 FORJ=14988TO15025:REM WIS SUBROUTINE-REGEL
20230 POKEJ+I*45,58
20240 NEXTJ
20250 IFLEN(F$(I))>0THENL$=X$(I)+F$(I):GOSUB22000
20260 NEXTI
20270 U=1
20280 RETURN
22000 REM
22010 FO=2
22020 FINI=14988+I*45
22030 L=LEN(L$)
22040 FORQ=1TOL
22045 LL=0
22050 A=ASC(MID$(L$,Q,1))
22053 IFQ+1<LTHENLL=1:B=ASC(MID$(L$,Q+1,1))
22056 IFQ+2<LTHENLL=2:C=ASC(MID$(L$,Q+2,1))
22060 IFA=32THEN22340:REM SPATIE
22070 IFA=42THENA=165:REM *
22080 IFA=43THENA=163:REM +
22090 IFA=45THENA=164:REM -
22100 IFA=47THENA=166:REM /
22110 IFA=60THENA=172:REM <
22120 IFA=61THENA=171:REM =
22130 IFA=62THENA=170:REM >
22140 IFA=94THENA=167:REM ^
22145 IFLL=0THEN22320
22150 IFA=79ANDB=82THENA=169:GOTO22300:REM OR
22155 IFLL=1THEN22320
22160 IFA=83ANDB=71ANDC=78THENA=173:GOTO22310:REM SGN
22170 IFA=65ANDB=78ANDC=68THENA=168:GOTO22310:REM AND
22180 IFA=73ANDB=78ANDC=84THENA=174:GOTO22310:REM INT
22190 IFA=78ANDB=79ANDC=84THENA=161:GOTO22310:REM NOT

```

```

22200 IFA=65ANDB=66ANDC=83THENA=175:GOTO22310:REM ABS
22210 IFA=82ANDB=78ANDC=68THENA=180:GOTO22310:REM RND
22220 IFA=83ANDB=81ANDC=82THENA=179:GOTO22310:REM SGR
22230 IFA=76ANDB=79ANDC=71THENA=181:GOTO22310:REM LOG
22240 IFA=69ANDB=88ANDC=80THENA=182:GOTO22310:REM EXP
22250 IFA=67ANDB=79ANDC=83THENA=183:GOTO22310:REM COS
22260 IFA=83ANDB=73ANDC=78THENA=184:GOTO22310:REM SIN
22270 IFA=84ANDB=65ANDC=78THENA=185:GOTO22310:REM TAN
22290 GOTO22320
22300 Q=Q+1:GOTO22320
22310 Q=Q+2
22320 FOKEFINI,A
22330 FINI=FINI+1
22340 NEXTQ
22350 RETURN
23000 REM SCHRIJF DE OUDE CONSTATE
23010 Y$(I)=H$:PRINT!(17,31,5+I);"X";CHR$(I+48);"=";Y$(I)
23020 RETURN
24000 FO=24000:REM CALCULEREN
24005 FORI=0TO9:C$=""
24010 PRINT!(17,36,5+I);:INPUTC$
24015 IFC$<>">"THENY$(I)=C$
24025 PRINT!(17,36,5+I);SPC(12)
24030 PRINT!(17,36,5+I);Y$(I):Y(I)=VAL(Y$(I))
24035 IFI=9THENGOSUB24280:GOTO24070
24060 ONI+1GOSUB24190,24200,24210,24220,24230,24240,24250,
24260,24270
24070 NEXTI
24075 XA=0:XB=0:XC=0:XD=0:XE=0:XF=0:XG=0:XH=0:XI=0:XJ=0
24080 GOSUB2
24085 RETURN
24090 IFXA=0THEN24095
24092 PRINT!(17,33,16);X$(0);XA
24095 IFXB=0THEN24105
24100 PRINT!(17,33,17);X$(1);XB
24105 IFXC=0THEN24115
24110 PRINT!(17,33,18);X$(2);XC
24115 IFXD=0THEN24125
24120 PRINT!(17,33,19);X$(3);XD
24125 IFXE=0THEN24135
24130 PRINT!(17,33,20);X$(4);XE
24135 IFXF=0THEN24145
24140 PRINT!(17,33,21);X$(5);XF
24145 IFXG=0THEN24155
24150 PRINT!(17,33,22);X$(6);XG
24155 IFXH=0THEN24165
24160 PRINT!(17,33,23);X$(7);XH
24165 IFXI=0THEN24175
24170 PRINT!(17,33,24);X$(8);XI
24175 IFXJ=0THEN24185
24180 PRINT!(17,33,25);X$(9);XJ
24185 RETURN
24190 X0=VAL(Y$(I)):RETURN
24200 X1=VAL(Y$(I)):RETURN
24210 X2=VAL(Y$(I)):RETURN
24220 X3=VAL(Y$(I)):RETURN
24230 X4=VAL(Y$(I)):RETURN
24240 X5=VAL(Y$(I)):RETURN
24250 X6=VAL(Y$(I)):RETURN

```

```
24260 X7=VAL(Y$(I)):RETURN
24270 X8=VAL(Y$(I)):RETURN
24280 X9=VAL(Y$(I)):RETURN
25000 REM INCREMENTEREN
25010 Y(O)=Y(O)+1:Y$(O)=STR$(Y(O)):X0=X0+1
25020 GOSUB2
25030 RETURN
26000 REM NAAR PRINTER
26005 POKE8994,1
26010 PP=53248
26020 FORI=0TO27
26030 FORJ=0TO63
26040 PRINTCHR$(PEEK(PP+I*64+J));
26050 NEXTJ:PRINT
26060 NEXTI
26065 PRINT:PRINT:PRINT:PRINT
26070 POKE8994,2
26080 RETURN
31000 REMSCHERMOPBOUW
31010 PRINT:(28):PRINT"F O R M U C A L C":PRINT
31011 FO=5
31020 PRINT"   BETEKENIS/FORMULE           WAARDE":PRINT
31030 A$="X"
31040 FORI=0TO19
31050 J=I:IFI>9THENJ=J+7
31060 IFI=10THENPRINT
31070 B$=A$+CHR$(J+48)+CHR$(61):PRINTB$
31080 NEXTI
31140 GOSUB24090
31150 RETURN
32000 REM VOLLEDIG SCHERM OPZETTEN
32010 GOSUB31000
32015 FO=31000
32020 FORI=0TO9
32030 PRINT!(17,3,5+I);E$(I)
32035 PRINT!(17,33,5+I);"X";CHR$(I+48);"=";Y(I)
32040 NEXTI
32050 FORI=0TO9
32060 PRINT!(17,3,16+I);F$(I)
32070 NEXTI
32080 PRINT!(17,33,16);"XA=";XA
32090 PRINT!(17,33,17);"XB=";XB
32100 PRINT!(17,33,18);"XC=";XC
32110 PRINT!(17,33,19);"XD=";XD
32120 PRINT!(17,33,20);"XE=";XE
32130 PRINT!(17,33,21);"XF=";XF
32140 PRINT!(17,33,22);"XG=";XG
32150 PRINT!(17,33,23);"XH=";XH
32160 PRINT!(17,33,24);"XI=";XI
32170 PRINT!(17,33,25);"XJ=";XJ
32180 IFU=1THENU=0:GOTO32200
32190 GOSUB2
32200 RETURN
33000 PRINT:(28)
33010 PRINT"U MAAKTE EEN FOUT !!!  NR.":FO
33020 GOSUB500
33030 GOSUB32010
```

```
34000 PRINT!(28):PRINT:PRINT
34010 PRINT!(17,8,5);"*** F O R M U C A L C ***"
34020 PRINT:PRINT"MET DIT PROGRAMMA KUNT U FORMULES INTIKKEN."
34030 PRINT"U KUNT DEZE FORMULES LATEN UITREKENEN"
34040 PRINT"IN DE FORMULES ZIJN XO T/M X9 MOGELIJKE"
34050 PRINT"CONSTANTEN, OMDAT U AAN DEZE EEN WAARDE KUNT"
34055 PRINT"TOEKENNEN."
34058 PRINT"XA T/M XJ ZIJN VARIABELEN. IN DE REGELS, DIE"
34060 PRINT"MET XA T/M XJ BEGINNEN, WORDEN FORMULES"
34062 PRINT"INGETIKT."
34065 PRINT"DE REGELS, DIE MET XO T/M X9 BEGINNEN KUNT U"
34068 PRINT"GEBRUIKEN VOOR INFORMATIE BETREFFENDE"
34069 PRINT"DE FORMULES.":PRINT
34070 PRINT"-- FORMULEREN (F): IN DEZE TOESTAND ACCEPTEERT"
34080 PRINT" HET PROGRAMMA FORMULES."
34082 PRINT" EEN EERDER INGETIKTE FORMULE WORDT GECOPIEERD"
34084 PRINT" INDIEN U OP DE BETREFFENDE REGEL"
34086 PRINT" HET TEKEN > INTIKT, GEVOLGD DOOR <RETURN>." :PRINT
34087 GOSUB34900:PRINT!(28)
34088 PRINT!(17,8,5);"*** F O R M U C A L C ***"
34090 PRINT"-- CALCULEREN (C): IN DEZE TOESTAND VERWACHT"
34100 PRINT" HET PROGRAMMA WAARDEN VOOR DE CONSTANTEN."
34110 PRINT" NA INVOER HIERVAN WORDEN DE WAARDEN VAN"
34120 PRINT" XA T/M XJ BEREKEND EN OP HET SCHERM"
34130 PRINT" WEERGEGEVEN. OOK HIER KAN > HELPEN,"
34132 PRINT" NU OM CONSTANTEN TE COPIEREN.":PRINT
34140 PRINT"-- INCREMENTEREN (I): NA HET INTIKKEN VAN I"
34150 PRINT" WORDT XO MET 1 VERHOOGD EN DE BEREKENING"
34160 PRINT" HERHAALD.":PRINT
34190 PRINT"-- PRINTEN (P): NA HET INTOETSEN VAN <P>"
34200 PRINT" WORDT HET SCHERMBEELD OP DE PRINTER"
34210 PRINT" UITGEPRINT.":PRINT
34220 PRINT"-- STOPPEN (S): NA HET INTOETSEN VAN <S>"
34230 PRINT" WORDT HET PROGRAMMA VERLATEN."
34300 GOSUB34900
34890 RETURN
34900 PRINT!(17,4,27);"VOOR VERVOLG DRUK OP SPATIEBALK"
34910 POKE2073,96:POKE57088,253
34920 IFPEEK(57088)=239THENPOKE2073,173:RETURN
34930 POKE2073,173:GOTO34910
35000 PRINT!(28)
35010 PRINT!(17,10,10);"*** EINDE ***"
35020 POKE2888,27:POKE8722,27
35030 PRINT!(17,0,20);" ":END
```

SCREEN EDITOR VOOR DE OSI DISK ASSEMBLER

In MICRO van October 1982 staat een programma van de hand van Les Cain uit PHOENIX ARIZ. dat voor een 2P goed werkt maar niet voor een op 48 char. omgebouwd Superboard.

Het programma is nu zo omgewerkt dat de schermparameters ingevoerd kunnen worden in de 4 variabelen: LINEST - CHRLIN - FSTLIN en LASLIN. De beide laatsten zijn \$D1 en \$D7 en zullen wel voor iedereen hetzelfde zijn. LINEST echter geeft de uiterste linker kantlijn weer en ligt meestal tussen \$0A en \$0D. CHRLIN geeft aan het aantal zichtbare characters min één - dus, 45 char. te zien, dan staat daar \$2C (44 dec).

Het Editor programma bevindt zich tussen de Assembler - einde op \$1573 - en het begin van de Extended Monitor - \$1700.

Er moet natuurlijk uit de bestaande Assembler worden weggesprongen naar dit programma. Dit gebeurt op \$020C, daar staat \$20 88 05 en wij veranderen dit naar \$4C 80 15.

In de Editor vindt U de volgende mogelijkheden:

Control 0 = schakel het programma in

- | | | |
|---|------------------------------------|---------------------------------|
| " | H = editcursor staat bovenin links | |
| " | U = cursor omhoog | } alleen aan de linker kantlijn |
| " | D = cursor omlaag | |
| " | R = cursor naar rechts | |
| " | L = cursor naar links | |

Alle tekst links van de cursor kan met de Rubout toets (zonder control) gewist worden, waarna weer nieuwe tekst ingetypt of een andere verandering aangebracht kan worden.

Door een Return schrijft nu het programma de verandering in de Assemblerbuffer en verdwijnt de editingcursor, de laatste plaats van de editingcursor maakt daarbij niets uit.

De Editor werkt alleen met Source die door een PRINT cmd. b.v. P400-600 o.i.d. op het scherm komt. Het toevoegen van commentaar aan de sourcelisting is een heel fijne toepassing van de Editor. Hiertoe typt U na de Operand één spatie en een punt-komma.

Rein Heesterman

FEBRUARI 1985

```

10      ; *****
20      ; *
30      ; * SCREEN EDITOR FOR DSI ASSEMBLER *
40      ; *
50      ; * FROM MICRO OCTOBER 1982
60      ; *
70      ; * BY LES CAIN PHOENIX ARIZONA *
80      ; *
90      ; *****
100     ;
110     ; ASSEMBLE AT $1580 AND
120     ; SAVE (!SA 09,1=1200/8)
130     ; CHANGE $020C TO JMP$1580 AND
140     ; SAVE (!SA 07,1=0200/8)
150     ; WITH TRACK ZERO UTILITY AND
160     ; EXT MON. CHANGE $2269 TO $60
170     ;
180 157C      *=$157C
190 020C=     GETCHR=$020C
200 020F=     BAKAS =$020F
210 0588=     INPUT =$0588
220 252B=     KEYIN =$252B
230 006C=     SCRLO =$006C
240 006D=     SCRHI =$006D
250 008D=     YHOLD =$008D
260 008E=     TEMP  =$008E
270 0002=     BUFST =$0002
280     ;
290 157C 0A      LINEST .BYTE $0A
300 157D 2C      CHRLIN .BYTE $2C
310 157E D1      FSTLIN .BYTE $D1
320 157F D7      LASLIN .BYTE $D7
330     ;
340 1580 208805   JSR INPUT
350 1583 C90F     CMP #$0F
360 1585 F003     BEQ EDIT
370 1587 4C0F02   JMP BAKAS
380     ;
390     ; MAIN PART OF PROGRAM
400     ;
410 158A A000     EDIT   LDY #$0
420 158C 848D     STY YHOLD
430 158E AC7C15   LDY LINEST
440 1591 846C     STY SCRLO
450 1593 AD7F15   LDA LASLIN
460 1596 856D     STA SCRHI
470 1598 A48D     EDITA  LDY YHOLD
480 159A B16C     LDA (SCRLO),Y
490 159C 858E     STA TEMP
500 159E A9A1     LDA #$A1
510 15A0 916C     STA (SCRLO),Y
520     ;
530     ; GET KEY ROUTINE
540     ;

```

550	15A2	202B25		JSR	KEYIN
560	15A5	48		PHA	
570	15A6	A48D		LDY	YHOLD
580	15A8	A58E		LDA	TEMP
590	15AA	916C		STA	(SCRLO),Y
600	15AC	68		FLA	
610	15AD	A48D		LDY	YHOLD
620	15AF	D012		BNE	SIDE
630	15B1	C904		CMF	##4
640	15B3	D005		BNE	STCMP
650	15B5	48		PHA	
660	15B6	208616		JSR	CTRLD
670	15B9	68		FLA	
680	15BA	C915	STCMP	CMF	##15
690	15BC	D005		BNE	SIDE
700	15BE	48		PHA	
710	15BF	206A16		JSR	CTRLU
720	15C2	68		FLA	
730	15C3	C90C	SIDE	CMF	##0C
740	15C5	D005		BNE	CTR
750	15C7	48		PHA	
760	15C8	201E16		JSR	CTRLL
770	15CB	68		FLA	
780	15CC	C912	CTR	CMF	##12
790	15CE	D005		BNE	CRD
800	15D0	48		PHA	
810	15D1	201216		JSR	CTRLR
820	15D4	68		FLA	
830	15D5	C90D	CRD	CMF	##0D
840	15D7	F076		BEQ	CR
850	15D9	C95F		CMF	##5F
860	15DB	F049		BEQ	RUBOUT
870	15DD	C908		CMF	##08
880	15DF	F05E		BEQ	HOME
890	15E1	C920		CMF	##20
900	15E3	90B3		BCC	EDITA
910	15E5	C97D		CMF	##7D
920	15E7	B0AF		BCS	EDITA
930				:	
940				:	STORE CHAR ON SCREEN
950				:	
960	15E9	A48D		LDY	YHOLD
970	15EB	CC7D15		CFY	CHRLIN
980	15EE	F019		BEQ	ST2
990	15F0	E68D		INC	YHOLD
1000	15F2	48	ST1	PHA	
1010	15F3	B16C		LDA	(SCRLO),Y
1020	15F5	AA		TAX	
1030	15F6	68		FLA	
1040	15F7	916C		STA	(SCRLO),Y
1050	15F9	C960		CMF	##60
1060	15FB	F00C		BEQ	ST2
1070	15FD	8A		TXA	
1080	15FE	C8		INY	
1090	15FF	CC7D15		CFY	CHRLIN
1100	1602	D0EE		BNE	ST1
1110	1604	88		DEY	
1120	1605	A960		LDA	##60


```

1130 1607 9160          STA (SCRLO),Y
1140 160F A48D          STZ   LDY YHOLD
1150 160B A58E          LDA   TEMP
1160 160D 9160          STA (SCRLO),Y
1170 160F 18           STZ   CLC
1180 1610 9086          BCC  EDITA
1190                   ;
1200                   ;CURSOR RIGHT CTRL R
1210                   ;
1220 1612 A58E          CTRLR LDA   TEMP
1230 1614 C960          CMP  ##60
1240 1616 F005          BEQ  CTR2
1250 1618 A48D          LDY  YHOLD
1260 161A 08           INY
1270 161B 948D          CTR1  STY  YHOLD
1280 161D 60           CTR2  RTS
1290                   ;
1300                   ;CURSOR LEFT CTRL L
1310                   ;
1320 161E A48D          CTRLL LDY  YHOLD
1330 1620 F0FF          BEQ  CTR1
1340 1622 88           DEY
1350 1623 18           CLC
1360 1624 90F5          BCC  CTR1
1370                   ;
1380                   ;RUBOUT
1390                   ;
1400 1626 A48D          RUBOUT LDY  YHOLD
1410 1628 F012          BEQ  RUB3
1420 162A B160          RUB1  LDA (SCRLO),Y
1430 162C 88           DEY
1440 162D 9160          STA (SCRLO),Y
1450 162F C960          CMP  ##60
1460 1631 F007          BEQ  RUB2
1470 1633 08           INY
1480 1634 08           INY
1490 1635 0C7D15        CPY  CHRLIN
1500 1638 30F0          BMI  RUB1
1510 163A C68D          RUB2  DEC  YHOLD
1520 163C 18           RUB3  CLC
1530 163D 90D0          BCC  ST3
1540                   ;
1550                   ;CURSOR HOME CTRL H
1560                   ;
1570 163F AD7D15        HOME  LDA  LINEST
1580 1642 8560          STA  SCRLO
1590 1644 A900          LDA  ##00
1600 1646 858D          STA  YHOLD
1610 1648 AD7E15        LDA  FSTLIN
1620 164B 856D          STA  SCRHI
1630 164D D0C0          BNE  ST3
1640                   ;
1650                   ;EDIT END CR
1660                   ;
1670 164F A000          CR   LDY  ##0
1680 1651 B160          CR1  LDA (SCRLO),Y
1690 1653 C960          CMP  ##60
1700 1655 F008          BEQ  CR2

```

1710	1657	9102		STA (BUFST),Y
1720	1659	C8		INY
1730	165A	CC7D15		CFY CHRLIN
1740	165D	D0F2		BNE CR1
1750	165F	9B	CR2	TYA
1760	1660	D003		BNE RETURN
1770	1662	4C0C02		JMP GETCHR
1780				:
1790				:BACK TO ASM
1800				:
1810	1665	A90D	RETURN	LDA ##0D
1820	1667	4C0F02		JMP BAKAS
1830				:
1840				:CURSOR UP CTRL U
1850				:
1860	166A	A56C	CTRLU	LDA SCRLO
1870	166C	3B		SEC
1880	166D	E940		SBC ##40
1890	166F	AA		TAX
1900	1670	A56D		LDA SCRHI
1910	1672	B008		BCS CTL1
1920	1674	CD7E15		CMF FSTLIN
1930	1677	F008		BEG CTL2
1940	1679	3B		SEC
1950	167A	E901		SBC ##1
1960	167C	856D	CTL1	STA SCRHI
1970	167E	866C		STX SCRLO
1980	1680	60		RTS
1990	1681	AE7C15	CTL2	LDX LINEST
2000	1684	D0F6		BNE CTL1
2010				:
2020				:CURSOR DOWN CTRL D
2030				:
2040	1686	A56C	CTRLD	LDA SCRLO
2050	1688	1B		CLC
2060	1689	6940		ADC ##40
2070	168B	AA		TAX
2080	168C	A56D		LDA SCRHI
2090	168E	6900		ADC ##0
2100	1690	CD7F15		CMF LASLIN
2110	1693	D0E7		BNE CTL1
2120	1695	AE7C15		LDX LINEST
2130	1698	D0E2		BNE CTL1
2140				: END

Een hexdump van de Smart-Terminal.

Het smart-terminal programma uit OSI-POEL 10 is erg lang uitgedrukt in assembler. Vandaar dat OSI-bezitters met een kleiner geheugen problemen zullen hebben om het te assembleren.

De hierbij gegeven hexdump geeft het programma weer met \$1980 als startadres, dus bovenaan in de eerste 8k RAM. Het is mogelijk om zelf (handmatig) het programma te verplaatsen. De onderstreepte geheugencellen geven de Page-nummers aan. Als het programma dus verplaatst moet worden van \$1980 naar \$3980 (bovenaan in het 16 k gebied) dan hoeven alleen alle 19 (onderstreept) te worden veranderd in 39, alle 1A in 3A etc. tot en met 1F. Het is niet raadzaam om dit programma lager in het geheugen te gaan zetten, omdat het dan moeilijker of zelfs onmogelijk wordt BASICODE programma's binnen te halen of weg te zenden.

Er is 1 (een) verschil met het gepubliceerde programma in OSI-POEL. In plaats van <ctrl> plus een andere toets, moet nu eerst <esc> worden ingedrukt en daarna de betreffende letter. Sommige opbel-komputers moeten namelijk met een <ctrl>+letter worden bestuurd. (<ctrl>-s is vaak stoppen van de listing en <ctrl>-x het afbreken van de output).

Overigens kan ik na enige maanden intensief experimenteren met dit programma melden dat het prettig werkt en dat ik geen fouten meer vermoed.

Jos Burghouts
(040-528372)

```

      0 1 2 3 4 5 6 7 8 9 A B C D E F
1980 20 05 1A 20 23 1B 20 71 1B 4C 74 A2 1A 1C 2D 1C
1990 AE F3 69 FF 30 1D BA FF F1 1C 3A 48 28 48 A9 01
19A0 20 3E FC 20 C6 FC D0 12 0A D0 F5 A9 00 3D 16 02
19B0 3D 15 02 A9 04 3D 14 02 D0 46 4A 20 02 2A E0 21
19C0 D0 E6 A9 1B D0 21 20 C6 FD 23 8D 13 02 0A 0A 0A
19D0 38 ED 13 02 3D 13 02 8A 4A 20 C6 FD D0 CD 18 23
19E0 6D 13 02 A3 B9 CD FD CD 15 02 D0 C4 CE 14 02 F0
19F0 0C A0 05 A2 C8 CA D0 FD 88 D0 F8 F0 03 4C 63 FD
1A00 A9 FF 4C BB FD A2 06 CA 3D E1 1F 05 EF 3A D0 F7
1A10 A2 02 20 0B 1B C9 43 F0 03 C9 46 D0 F3 A2 60 85
1A20 EF A2 03 20 0B 1B C9 41 F0 03 C9 55 D0 F3 A2 60
1A30 85 F2 A2 04 20 0B 1B C9 45 D0 04 A9 80 D0 0E C9
1A40 4E D0 04 A9 00 F0 06 C9 4F D0 E7 A9 01 85 EC A2
1A50 05 20 0B 1B C9 31 D0 04 A9 02 D0 06 C9 32 D0 EF
1A60 A9 20 85 EE A2 06 20 0B 1B C9 37 D0 04 A9 01 D0
1A70 06 C9 38 D0 EF A2 05 85 ED 20 A7 1A B0 B4 A5 EC
1A80 29 7F 65 ED 65 EE C9 07 D0 07 AA A5 EC D0 01 CA
1A90 3A AA CA 8A 0A 0A 4A E8 8A 3D 00 F0 AD 24 1D AC
1AA0 25 1D 85 11 34 12 60 18 A5 ED 65 EC C9 01 F0 0D
1AB0 18 A5 ED 65 EE C9 05 D0 0E A5 EC F0 0A 20 1D 1B

```

1AC0 A2 07 20 C9 1A 38 60 18 60 20 19 1B A0 FF CA 30
 1AD0 08 C8 B9 E6 1E 10 FA 30 F5 C8 39 E6 1E 30 05 20
 1AE0 69 FF D0 F5 29 7F 4C 69 FF A2 03 20 FF 1A 0A 0A
 1AF0 0A 0A 95 E2 20 FF 1A 15 E2 95 E2 CA 10 ED 60 20
 1B00 00 FD 20 69 FF 20 93 FE 30 F5 60 20 C9 1A 20 00
 1B10 FD 4C 69 FF A2 08 4C C9 1A A9 0A 20 00 F3 A9 0D
 1B20 4C 00 F8 A0 00 20 47 FF 20 52 FF A9 0D 3D 24 02
 1B30 A0 00 B9 05 1E 30 06 20 69 FF C8 D0 F5 27 7F 20
 1B40 69 FF A2 00 86 0D 9E 05 02 20 C9 1A 20 E9 1A A2
 1B50 01 20 0B 13 C9 59 D0 F1 A5 E2 95 E6 95 F5 A5 E3
 1B60 85 E7 85 F6 A5 E4 95 E8 95 F7 A5 E5 95 E9 85 E9
 1B70 60 20 9A 19 C9 FF F0 3B C9 13 D0 31 20 00 FD C9
 1B80 4E F0 29 C9 53 D0 03 4C E0 1B C9 50 D0 03 4C 5E
 1B90 1C C9 56 D0 03 4C A1 1C C9 57 D0 06 20 23 1D 4C
 1BA0 74 A2 C9 54 D0 07 20 05 1A 4C 71 1B 60 20 EF 00
 1BB0 20 B1 FC AD 00 F0 4A 90 0E AD 01 F0 22 7F 20 CA
 1BC0 1B 20 F2 00 20 D2 1C 4C 71 1B 29 7F C9 7F 00 0F
 1BD0 C9 1A F0 0B C9 19 F0 07 C9 18 F0 03 20 00 F3 60
 1BE0 A5 79 C9 01 D0 C6 A5 7A C9 03 D0 C0 AD 9C 12 AC
 1BF0 8D 19 8D 1A 02 8C 1B 02 AD 9E 19 AC 9F 19 85 04
 1C00 84 05 A9 99 A0 00 85 13 84 14 A9 32 95 ED 20 B1
 1C10 FC A2 12 86 C3 84 C4 4C F6 A5 C9 0A F0 0E 02 80
 1C20 48 45 ED 85 ED 68 20 B1 FC 20 2B 02 60 48 98 48
 1C30 AD 90 19 AC 91 19 85 04 84 05 AD 22 19 AC 23 12
 1C40 8D 1A 02 8C 1B 02 A9 0D 20 1A 1C A9 03 20 1A 1C
 1C50 A5 ED 20 B1 FC 20 14 1B 68 A8 68 4C 71 1B A2 00
 1C60 86 0D 8E 05 02 20 C9 1A 20 E9 1A A2 01 20 0B 1B
 1C70 C9 59 F0 03 4C 71 1B A0 00 B1 E4 22 7F 20 31 FC
 1C80 20 2B 02 A5 E4 C5 E2 D0 10 A5 E5 C5 E2 0A 23
 1C90 8D 05 02 20 14 1B 4C 71 1B E6 E4 D0 DC E6 85 D0
 1CA0 D8 A5 F7 85 EA A5 F8 85 EB 20 14 1D 20 CB 1C 80
 1CB0 0F 20 69 FF 20 A6 FC F0 FB 20 2A 12 C9 16 D0 EC
 1CC0 A5 EA 85 F7 A5 EB 85 F8 4C 71 1B A0 00 B1 F7 18
 1CD0 90 0C C9 00 F0 1A C9 7F F0 16 A0 00 21 F7 A8 E6
 1CE0 F7 D0 02 E6 F8 A5 F8 C5 F6 20 04 A5 F7 C5 F5 20
 1CF0 60 20 CB 1C 30 01 60 20 19 1B A6 EE 20 C9 1A AD
 1D00 96 12 AC 97 19 8D 18 02 8C 19 02 A5 F8 A4 EB 85
 1D10 F7 84 EA 60 A5 E9 85 F8 A5 E8 85 F7 20 D6 FB 20
 1D20 D6 F9 60 A5 F7 85 EA A5 F8 85 EB A9 00 8D 03 02
 1D30 A2 09 20 0B 1B C9 4C F0 06 C9 4F D0 F3 85 EC 20
 1D40 BC 1D A9 00 85 ED A9 0E 20 03 1B C9 4F 54 C9
 1D50 4E D0 F3 20 F4 FF AD 03 02 D0 04 A9 0D 00 35 20
 1D60 EB FF A4 EC D0 06 C9 82 D0 EC 35 EC 48 45 ED 85
 1D70 ED 68 22 7F 20 23 02 20 DA 1C 08 C9 03 F0 07 28
 1D80 90 D4 A9 0B D0 0E 28 20 ER FF C5 ED F0 04 A9 0C
 1D90 D0 02 A9 08 85 EE A9 00 8D 03 02 A5 F7 A4 F8 85
 1DA0 F5 84 F6 AD 98 19 AC 99 12 9D 19 02 9C 19 02 A5
 1DB0 E4 A4 E5 85 F7 84 F8 A9 00 85 ED 60 A2 0A 20 0B
 1DC0 1B C9 4E D0 17 A5 85 A4 86 85 F5 84 F6 85 E2 84
 1DD0 E3 A5 79 A4 7A C8 95 E4 84 E5 D0 24 C9 4F D0 D0
 1DE0 20 19 1B A2 00 20 C9 1A 20 E9 1A A2 01 20 0B 1B
 1DF0 C9 59 D0 EC A5 E2 A4 E3 95 F5 84 F6 A5 E4 A4 E5
 1E00 85 F7 84 F8 60 45 53 43 M X=ESC
 1E08 2B 0A 0D 2B 4E 3D 54 65 + +N=TE
 1E10 72 75 67 20 6E 61 61 72 PUG NAAR
 1E18 20 62 61 73 69 63 0A 0D BASIC
 1E20 2B 53 3D 5A 65 6E 64 74 +S=ZENDT
 1E28 20 62 61 73 69 63 28 6F BASIC00

1E30 64 65 22 70 72 6F 67 72 DE)PR2CD
 1E38 2E 20 75 69 74 0A 0D 23 . MIT +
 1E40 50 2D 5A 65 6E 64 74 20 P=7ENOT
 1E48 67 65 68 65 75 67 65 6E CEVENGEN
 1E50 64 75 6D 70 20 6E 61 61 DIMP NAA
 1E58 72 20 4D 4F 44 45 4D 0A P MODEM
 1E60 0D 2B 56 3D 54 6F 6F 6E +U=T02M
 1E68 20 62 69 6E 6E 65 6E 67 BINJENC
 1E70 65 6B 6F 6D 65 6E 20 74 EK2MEN T
 1E78 65 68 65 6E 73 20 76 61 EKENS MA
 1E80 6E 61 66 20 62 65 67 69 NAF BECI
 1E88 6E 0A 0D 20 20 20 53 70 N SP
 1E90 61 74 69 65 3D 72 74 6F ATIE=ST0
 1E98 70 2F 3C 43 54 50 4C 3E P/<CTRL>
 1EA0 2D 56 3D 65 69 6E 64 65 -U=EINDE
 1EA8 0A 0D 22 57 3D 48 61 61 +U=HAA
 1EB0 6C 20 62 61 73 60 63 6F L BASIC0
 1EB8 64 65 20 70 72 6F 67 72 DE PR0CP
 1EC0 2E 20 62 62 6E 6E 65 6E . BINNEN
 1EC8 0A 0D 23 54 3D 56 65 72 +T=VER
 1ED0 61 6E 64 65 72 20 7A 65 ANDER ZE
 1ED8 6E 64 70 72 6F 74 6F 6B NDRR0TZK
 1EE0 6F 6C 0A 0A 0A 83 46 67 0L FI
 1EE8 72 73 74 2F 4C 61 73 74 RST/LAST
 1EF0 20 3F 87 52 65 61 64 72 ? READY
 1EF8 3F 87 48 61 6C 66 2D 6F ? HALF-0
 1F00 66 20 46 75 6C 6C 64 75 F FULLDU
 1F08 70 6C 65 78 20 48 2F 46 PLEX H/F
 1F10 2F 87 45 63 68 6F 6D 6F ? ECH0M0
 1F18 64 65 20 41 61 6E 2F 55 DE AAN/U
 1F20 62 74 3F 87 50 61 72 69 IT? PARI
 1F28 74 72 20 45 76 65 6E 2F TY EVEN/
 1F30 4E 6F 6E 65 2F 4F 64 64 NONE/0DD
 1F38 2F 37 53 74 6F 70 62 69 ? ST0PBI
 1F40 74 73 20 32 20 6F 66 20 TS 2 0F
 1F48 31 3F 87 27 20 6F 66 20 I? 7 0F
 1F50 38 20 62 69 74 73 3F 87 8 BITS?
 1F58 46 6F 75 74 20 6F 70 6E F0NT 0PN
 1F60 62 65 75 77 87 52 65 61 IEMV REA
 1F68 64 72 2E 07 07 07 0A 8D DV.
 1F70 42 61 73 62 63 6F 64 65 BASIC0DE
 1F78 20 4C 61 64 65 6E 20 6F LADEN 0
 1F80 66 20 4F 70 6E 65 6D 65 F 0PNEME
 1F88 6E 3F 87 47 65 63 65 75 N? GEHEM
 1F90 67 65 6E 61 64 72 65 73 GENADRES
 1F98 20 4E 6F 72 6D 61 61 6C N0RMAAL
 1FA0 20 6F 66 20 4F 70 67 65 0F 0DCE
 1FA8 76 65 6E 3F 87 47 65 68 VEN? GEH
 1FB0 65 75 67 65 6E 20 76 6F EUGEN V0
 1FB8 6C 87 43 68 65 63 6B 73 L CHECKS
 1FC0 75 6D 20 66 6F 75 74 87 UM F0UT
 1FC8 41 66 67 65 62 72 6F 6B AFGE880?
 1FD0 65 6E 87 4F 75 64 20 6F EN 0UD 0
 1FD8 66 20 6E 69 65 75 77 3F F NIEUW?
 1FE0 87 4C CA LB 4C B1 FC 24 LJ LI S

OSI POEL

FEBRUARI 1985

Gebruik van de FORTH ASSEMBLER voor de 6502-microprocessor.

Deze ASSEMBLER (pag. EE 35 e.v.) is gemaakt door William F. Ragsdale en met een uitgebreide beschrijving gepubliceerd in FORTH DIMENSIONS III/5 pag.143-150.

Hier volgen in het kort enkele aantekeningen, die van nut kunnen zijn bij een eerste gebruik van deze ASSEMBLER.

De 6502-microprocessor kent een-byte, twee-byte en drie-byte instructies.

Een-byte instructies.

Als de instructie een manipulatie is van de accumulatorinhoud, maar waarvan de mnemonics ook op de inhoud van een geadresseerde byte zou kunnen worden uitgevoerd, moet de opcode vooraf worden gegaan door .A (zie pag. FA 6 scr.42 regel 5: A. ASL,).

Twee- en drie-byte instructies.

In een twee- en drie-byte instructie wordt onderscheid gemaakt tussen opcode en operand:

opcode = operatie code = wat er wordt gedaan

operand = waarmee wat wordt gedaan

adres of data ($0 \leq \text{data} < 256$).

De ASSEMBLER verwacht, dat de operand op de getallenstack staat, als hij de mnemonics van een opcode tegenkomt. De operand moet dus op de stack gezet worden, alvorens de instructie wordt verwerkt.

De ASSEMBLER compileert alleen de ASSEMBLER-woorden. Kent de ASSEMBLER het woord niet, dan geeft hij het door aan FORTH. Komt FORTH een getal tegen, dan zet hij het op de getallenstack. De ASSEMBLER, die weer wordt aangeroepen, als er een ASSEMBLER-woord gecompileerd moet worden, haalt het getal weer van de stack af.

Als de operand gevolgd wordt door #, dan is het data (zie scr.42 regel 2: 6 # LDX,), anders wordt de operand behandeld als een adres, ook voor page zero.

Wordt een CONSTANT gebruikt als operand, dan komt de waarde van deze CONSTANT op de stack, en die moet dan weer al of niet gevolgd worden door # (zie pag. FA 7 scr.31 regel 4: LSF STA,).

Wordt een VARIABLE gebruikt als operand, dan wordt het adres van het parameterveld (PFA) van die VARIABLE op stack gezet (en niet zijn waarde!) (scr.42 regel 3: RIJ Y, LDA,).

Als de waarde van VARIABLE opgenomen moet worden in de machinetaalroutine, dan de VARIABLE-naam laten volgen door @, en als dat als data wordt gebruikt ook nog door #.

Als het PFA van de VARIABLE wordt gebruikt, kan met die VARIABLE gemanipuleerd worden, b.v. gebruikt worden als teller (scr.30 regel 12: LEN DEC,).

Subroutines.

Als een subroutine wordt aangeroepen (met JSR,), moet het startadres van die subroutine op de getallenstack staan. Zou alleen de naam van de subroutine in de code opgenomen worden, dan wordt bij compilatie het code field address (CFA) van de routine op de stack gezet en loopt bij uitvoering van het machinecode-woord het systeem vast. Een machinecode-subroutine begint op de plaats van het PFA van een gewoon FORTH-woord. Dus moet het PFA van de subroutine op de stack staan. Dat wordt bereikt met de naam van de subroutine te laten voorafgaan door ' (TICK). Dus ' naam JSR,.

Het woord INKEY komt niet voor in FIG-FORTH, maar wel in de woordenschat van b.v. de JUPITER ACE.
KEY wacht op een toetsindruk.
INKEY kijkt of een toets is ingedrukt, maar wacht niet.

INKEY bevat een lus met Y in een lus met X als variabele

De ASCII-waarde van de ingedrukte toets staat in variabele TEMPA.

SCR # 41

```

0 ( INKEY 1 )
1 DECIMAL 26 EMIT ." INKEY 1" CR
2 HEX 0 VARIABLE MATRIX 3E ALLOT 0 VARIABLE TEL
3 : VUL.MATRIX 0 TEL ! 29 BLOCK 2FF + DUP 100 + SWAP
4     DO I NUMBER DROP TEL @ MATRIX + C!
5     TEL @ 1+ TEL !
6     4 +LOOP ;
7 VUL.MATRIX
8 0 VARIABLE RIJ 6 ALLOT 0 VARIABLE TEMPA
9 : VUL.RIJ FE FD FB F7 EF DF BF 7F
10     B 0 DO RIJ I + C! LOOP ;
11 VUL.RIJ -->
12 31 32 33 34 35 36 37 20 38 39 30 3A 2D 5F 20 20
13 2E 4C 4F 0A 0D 20 20 20 57 45 52 54 59 55 49 20
14 53 44 46 47 48 4A 4B 20 58 43 56 42 4E 4D 2C 20
15 51 41 5A 20 2F 3B 50 20 20 20 20 20 20 20 20 20

```

*41 DECIMAL = 29 HEX
2FF → REGEL 12
100 → 4 REGELS*

SCR # 42

```

0 ( INKEY 2 )
1 ." INKEY 2" CR HEX ASSEMBLER
2 CODE INKEY PHA, TXA, PHA, TYA, PHA, 6 # LDX,
3     BEGIN, RIJ ,X LDA, DFOO STA, 6 # LDY,
4     BEGIN, RIJ ,Y LDA, DFOO CMP, 0=
5     IF, TXA, .A ASL, .A ASL, .A ASL,
6     TEMPA STA, TYA, CLC, TEMPA ADC, TAY,
7     MATRIX ,Y LDA, TEMPA STA, 0 # LDY, 0 # LDX,
8     THEN, DEY, 0<
9     UNTIL, DEX, 0<
10    UNTIL,
11    PLA, TAY, PLA, TAX, PLA,
12    NEXT JMP,
13 END-CODE
14 FORTH DECIMAL ;S
15

```

```

: KIKJ 10000 0 DO
    0 TEMPA !
    INKEY
    TEMPA @ TEMPB @ =
    IF
    ELSE TEMPA @ DUP TEMPB ! EMIT
    THEN
    TEMPA @ TEMPB !
    LOOP ;

```


SCR # 30

```

0 ( TONEN 1 )
1 26 EMIT ." TONEN 1" CR
2 0 VARIABLE DUUR 10 VARIABLE LEN 0 VARIABLE TIMBRE1
3 0 VARIABLE TIMBRE2 0 VARIABLE TEMP 0 VARIABLE LENGTE
4 HEX ASSEMBLER
5 CODE WACHT PHA, TXA, PHA, TYA, PHA,
6     BEGIN, FO # LDX,
7     BEGIN, 10 # LDY,
8     BEGIN, NOP, NOP, NOP, DEY, 0=
9     UNTIL,
10    DEX, 0=
11    UNTIL,
12    LEN DEC, 0=
13    UNTIL,
14    PLA, TAY, PLA, TAX, PLA, NEXT JMP,
15    END-CODE -->

```

SCR # 31

```

0 ( TONEN 2 )
1 ." TONEN 2" CR F000 CONSTANT LSP
2 CODE NOOT PHA, TYA, PHA, TXA, PHA,
3     BEGIN, LENGTE LDA, TEMP STA,
4     BEGIN, 51 # LDA, LSP STA, TIMBRE1 LDX, TIMBRE2 LDY,
5     BEGIN, NOP, NOP, NOP, NOP, DEY, 0=
6     UNTIL,
7     11 # LDA, LSP STA,
8     BEGIN, NOP, NOP, NOP, NOP, DEX, 0=
9     UNTIL,
10    TEMP DEC, 0=
11    UNTIL,
12    DUUR DEC, 0=
13    UNTIL,
14    PLA, TAX, PLA, TAY, PLA, NEXT JMP,
15    END-CODE --> 33 LOAD ;S

```

SCR # 32

```

0 ." NOTEN 1" CR
1 FORTH DECIMAL
2 : MAAT LEN ! WACHT ;
3 : TOON TIMBRE2 ! TIMBRE1 ! LENGTE ! NOOT ;
4 : G1 DUUR ! 53 98 DUP TOON ; : G2 DUUR ! 106 49 DUP TOON ;
5 : G3 DUUR ! 212 25 DUP TOON ; : A1 DUUR ! 30 174 DUP TOON ;
6 : A2 DUUR ! 60 87 DUP TOON ; : A3 DUUR ! 120 44 DUP TOON ;
7 : A4 DUUR ! 240 22 DUP TOON ;
8 : B1 DUUR ! 33 154 DUP TOON ; : B2 DUUR ! 67 77 DUP TOON ;
9 : B3 DUUR ! 134 39 DUP TOON ; : C1 DUUR ! 35 145 DUP TOON ;
10 : C2 DUUR ! 70 73 DUP TOON ; : C3 DUUR ! 140 36 DUP TOON ;
11 : D1 DUUR ! 40 129 DUP TOON ; : D2 DUUR ! 80 65 DUP TOON ;
12 : D3 DUUR ! 160 32 DUP TOON ; : E1 DUUR ! 44 116 DUP TOON ;
13 : E2 DUUR ! 89 58 DUP TOON ; : E3 DUUR ! 180 29 DUP TOON ;
14 : F1 DUUR ! 47 109 DUP TOON ; : F2 DUUR ! 94 55 DUP TOON ;
15 : F3 DUUR ! 188 27 DUP TOON ; -->

```

SCR # 33

```

0 ." NOTEN 2" CR
1
2 : C#1 DUUR ! 37 137 DUP TOON ; : C#2 DUUR ! 73 69 DUP TOON ;
3 : C#3 DUUR ! 149 34 DUP TOON ; : D#1 DUUR ! 42 122 DUP TOON ;
4 : D#2 DUUR ! 83 61 DUP TOON ; : D#3 DUUR ! 169 30 DUP TOON ;
5 : F#1 DUUR ! 47 104 DUP TOON ; : F#2 DUUR ! 94 52 DUP TOON ;
6 : F#3 DUUR ! 188 26 DUP TOON ; : G#1 DUUR ! 56 92 DUP TOON ;
7 : G#2 DUUR ! 111 46 DUP TOON ; : G#3 DUUR ! 223 23 DUP TOON ;
8 : A#2 DUUR ! 62 82 DUP TOON ; : A#3 DUUR ! 125 41 DUP TOON ;
9 : A#1 DUUR ! 31 164 DUP TOON ;
10 : LADDER1 6 C1 6 D1 6 E1 6 F1 6 G1 6 A2 6 B2 6 C2 ;
11 : LADDER# 6 C#1 6 D#1 6 F1 6 F#1 6 G#1 6 A#2 6 C2 6 C#2 ;
12 -->
13
14
15

```

SCR # 34

```

0 ." MELODIE EN DEUN" CR
1 : MELO
2      1 C1 2 F1 1 F1 1 A2 2 F1 1 F1
3      1 C1 2 G1 1 G1 1 A#2 2 G1 1 G1
4      1 C1 2 A2 1 A2 1 A#2 2 C2 1 A2
5      1 F1 2 G1 1 G1 1 A2 2 A#2 2 G1
6      2 A2 1 F1 1 C1 2 D2 1 C1
7      1 F1 2 C1 1 D1 1 E1 2 F1
8 ;
9 : EVEN 12000 0 DO LOOP ;
10 : MELODIE MELO EVEN ;
11 : MEL1 1 A2 2 A2 1 A2 2 A2 1 A#2 2 C2 1 C2 2 C2
12      1 A2 2 G1 1 G1 2 F1 1 E1 4 F1 ;
13 : MEL2 1 C2 2 C2 1 C2 1 C2 1 D2 1 E2 2 F2 1 C2 2 C2
14      1 C2 2 A#2 1 A#2 2 C2 1 D2 4 C2 ;
15 : DEUN MEL1 MEL1 MEL2 MEL1 EVEN ; -->

```

SCR # 35

```

0 ." VW (VIER WEVERKENS)" CR
1
2 : VW1  2 C1 2 F1 2 A2 2 F1 2 E1 2 G1
3      2 E1 2 D1 2 F1 2 D1 4 C1
4      2 E1 2 F1 2 E1 2 F1 4 D1 2 G1 6 E1 4 F1 ;
5
6 : VW2  2 C1 2 A#2 2 G1 2 G1 2 A2 2 F1 2 C1 6 F1
7      1 G1 1 A2 2 A#2 2 G1 2 G1 2 A2 2 F1 2 C1 6 F1 ;
8
9 : VW3  2 F1 4 D1 2 D1 4 G1 2 G1 6 E1 6 F1 ;
10
11 : VW VW1 VW1 VW2 VW3 EVEN ;
12 ;S
13
14
15

```

Deze FORTH ASSEMBLER werkt zonder labels. Het is daarom alleen mogelijk te springen naar een ander woord, dat ook een machinecode programma(tje) is, b.v. een machinecode subroutine of het woord NEXT. (Met NEXT JMP. wordt bij uitvoering, execution, van het machinecode woord teruggesprongen naar FORTH)

De ASSEMBLER laat voorwaardelijke sprongen toe zonder gebruik van labels. Deze kunnen (en moeten) geconstrueerd worden met:

- a. BEGIN, ... UNTIL,
- b. IF, ... ELSE, ... THEN,

(Denk aan de komma's aan het einde van de ASSEMBLER woorden!)

Het type a kan genest worden in het type b, maar dan wel in zijn geheel tussen IF, en ELSE, of in zijn geheel tussen ELSE, en THEN, (ELSE, mag weggelaten worden)

Het type b kan genest worden in a. Maar ook hier moet een binnenlus beëindigd zijn voor de buitenlus (BEGIN, UNTIL,) wordt afgesloten.

Voor voorbeelden van nesting met deze structuren zie de screens 30, 31 en 42.

De algemene regel bij machinecode routines gemaakt m.b.v. deze FORTH ASSEMBLER (overigens ook van toepassing op colon-definities) is:

elkaar snijdende programmalijnen zijn niet toegestaan, of wat hetzelfde is:

alleen onderbroken lussen zijn toegestaan.

Het parool is dus:

GESTRUCTUREERD PROGRAMMEREN !!!

Van het woord INKEY (scr.42) volgt hieronder een disassembly listing.

De woorden INKEY WACHT en NOOT maken bij lange na niet gebruik van alle eigenschappen en mogelijkheden van FORTH en ASSEMBLER. Ze zijn het resultaat van eerste pogingen.

FORTH dwingt tot 'top down' werken bij het opzetten van een programma. Maar het geeft ook de mogelijkheid om een programma van onderaf op te bouwen. Dat is zo, omdat elk woord (elke module) die als bouwsteen van een programma is opgezet, onafhankelijk getest kan worden. Met goed werkende modules kunnen dan weer grotere delen van een programma geschreven en getest worden, etc.

Bij het testen van een ML-subroutine moet U er wel aan denken om RTS, voor de test te vervangen door NEXT JMP,

Omdat bij het testen van modules problemen zouden kunnen ontstaan wegens het gebruik van de CPU-registers is het goed te weten, dat, als FORTH het woord NEXT verlaat om een CODE woord uit te voeren:

- het Y-register nul is;
- het X-register wijst naar de low byte van het onderste (laatst geplaatste) data stack item, gerekend vanaf het adres \$0000;
- de CPU stack pointer wijst naar de byte, die zich bevindt direct onder de low byte van de onderste cel van de return stack. PLA brengt de inhoud van die byte naar de accumulator maar verandert ook de stack pointer;
- de accumulator vrij te gebruiken is;
- de processor in de binaire toestand is (CLD is van kracht).

```

289E 48      PHA
289F 8A      TXA
28A0 48      PHA
28A1 98      TYA
28A2 48      PHA
28A3 A206    LDX  #06
28A5 BD3E28 LDA  $283E, X ←
28A8 BD00DF STA  $DF00
28AB A006    LDY  #06
28AD B93E28 LDA  $283E, Y ←
28B0 CD00DF CMP  $DF00
28B3 D017    BNE  $28CC
28B5 BA      TXA
28B6 0A      ASL  A
28B7 0A      ASL  A
28B8 0A      ASL  A
28B9 8D5028 STA  $2850
28BC 98      TYA
28BD 18      CLC
28BE 6D5028 ADC  $2850
28C1 AB      TAY
28C2 B99B27 LDA  $279B, Y
28C5 8D5028 STA  $2850
28C8 A000    LDY  #00
28CA A200    LDX  #00
28CC 88      DEY
28CD 10DE    BPL  $28AD ←
28CF CA      DEX
28D0 10D3    BPL  $28A5 ←
28D2 68      PLA
28D3 AB      TAY
28D4 68      PLA
28D5 AA      TAX
28D6 68      PLA
28D7 4C4404 JMP  $0444
    
```

EXTENDED BASIC VOOR O.S.I.

Naast de grote verscheidenheid aan allerlei hardware modificaties voor de diverse O.S.I. systemen, wordt het eens tijd om ook de software zijde eens aan de stand van deze tijd aan te passen. Bedoeld wordt met name de toch vrij simpele basic.

Hiervoor bestaat nu de mogelijkheid om dankzij de EXTENDED BASIC of kortweg EXBASIC de OSI weer eens wat extra kracht in te blazen.

Deze EXBASIC versie 1 vervangt een klein aantal van de standaard basic routines maar voegt EXTRA zo'n 39 commando's en statements toe. Een klein aantal van deze commando's waren in het verleden al via de diverse uitvoeringen toolkits bijgemaakt, maar maken nu dus deel uit van deze EXBASIC versie.

De uitbreiding met deze EXBASIC is bedoeld voor alle Superboard C1P systemen, al of niet gemodificeert, en de op het superboard gebaseerde ontwerpen zoals MOSI e.d. Met geringe aanpassing is deze EXBASIC ook bruikbaar op de andere OSI ontwerpen.

Hieronder volgt een uiteenzetting van de 39 EXTRA commando's en statements zoals deze in de EXBASIC versie 1. voorkomen.

CLS : Snelle scherm schoonmaak (volledig) en cursor links boven.
CLW : Schoonmaken van het momenteel in gebruik zijnde window, en de cursor naar linksboven (Home)
HOME : Stuurt de cursor naar linksboven.
FILLS : Dit commando vult elke schermpositie met een van te voren opgegeven karakter. Bv. FILLS 43 vult het hele scherm met '+' tekens.
FILLW : Dit commando vult elke schermpositie van het huidige window met het van tevoren opgegeven karakter. Bv. FILLW 43 vult het huidige window met '+' tekens.
GET : GET X wacht totdat er een toets gedrukt wordt en zet het toets symbool in de variabele X.
De variabele X kan van elk type zijn.
INKEY : Dit is een GET key routine zonder dat de programmaloop onderbroken wordt. INKEY X plaatst de symbool-waarde van de laatstelijk gedrukte toets in de variabele X.
De variabele X kan weer van elk type zijn.
OLD : Wanneer een programma abusievelijk verwijderd is met NEW, kan het met OLD weer teruggehaald worden.
Het programma wordt dan teruggeplaatst, maar geen variabelen.
Opmerking: Het programma werkt niet meer als het reeds overschreven is met een ander programma of variabelen.
FIND : Het commando FIND zoekt door het gehele programma, en list regels waarin de opgegeven combinatie van karakters voorkomt. Wanneer er meerdere malen deze combinatie voorkomt wacht de computer totdat er een toets gedrukt wordt, alvorens de volgende regel te listen. Eventuele spaties in het FIND commando worden genegeerd.
Bv. FIND POKE. Het programma wordt nu afgezocht, en alle regels waarin POKE voorkomt worden nu afgedrukt.

- TRACE** : Dit is een bijzonder sterk de-bugging commando wat u in staat stelt om de loop van het programma te aanschouwen en op het display, linksboven, de lijnummers afdrukt van de laatstelijk uitgevoerde regel van het basic-programma.
TRACE X, wanneer X>0 schakelt het monitoren in en het lijnnummer wordt links bovenaan het scherm afgedrukt.
Wanneer X>0, X zal de snelheid waarmee de monitoring wordt uitgevoerd beïnvloeden. (Een grotere waarde van X beïnvloedt de snelheid) Wanneer X=0 wordt TRACE uitgeschakeld.
X moet in de range van 0-255 zijn.
Deze uitvoering van TRACE is NIET destructief, uw normale programma-uitvoering op het scherm blijft behouden.
- ELSE** : Het commando na ELSE zal uitgevoerd worden als een eerder statement IF ... THEN niet resulteert in een actie.
ELSE mag NIET op dezelfde regel geplaatst worden als het IF... THEN statement.
Bijvoorbeeld :
10 GET K\$: IF K\$="G" THEN PRINT"DAT WAS TOETS G"
20 ELSE PRINT"DIT WAT NIET TOETS G"
- VIEW** : Laat zien wat er op een cassettebandje staat zonder het programma te laden.
Het indrukken van de spatie-balk stopt deze routine.
- GO** : GO werkt op dezelfde wijze als het oude statement GOTO.
GOTO is ge-updated om samen te kunnen werken met labels en expressies. (zie hiervoor hoofdstuk LABELS)
- GS** : GS werkt op dezelfde wijze als het oude statement GOSUB.
GOSUB is nu ook ge-updated om samen te kunnen werken met labels en expressies. (zie hiervoor hoofdstuk LABELS)
- WIN** : WIN X,Y,Z zet het huidige window op een breedte van X+1 karakters, met de linksboven positie op Y, en de linksonder positie op Z. (Dus snel veranderen van een window)
- NORM** : Dit statement reset het window naar zijn normaal formaat, en maakt tevens het scherm schoon.
- DOKE** : Dit commando werkt als een dubbel-byte POKE instructie.
DOKE 16384,1234 zet de lage byte van 1234 in 16384 en de hoge byte in 16385.
- DEEK** : Dit commando werkt als een dubbel-byte PEEK instructie.
Bijvoorbeeld; DEEK X,A zet de inhoud in X (lage byte), en X+1 (hoge byte) in de variabele A.
Elk type variabele kan toegepast worden.. ook strings..!
DEEK X,A\$ zet het karakter equivalent van de inhoud van adres X in A\$.
- REPEAT...UNTIL** : Dit nieuwe statement stelt u in staat om meer ge-structureerde programma-loops te maken.
Bijvoorbeeld;
REPEAT:GET K : UNTIL K=32 wacht totdat de spatie-balk gedrukt wordt.
REPEAT bepaalt de start van de loop, en het statement tussen REPEAT en UNTIL zal herhaald worden totdat de expressie achter UNTIL waar (trew) is.
Er kan tot een maximum van 8 loops genest worden...
- MON** : Snelle toegang tot de CEGMON monitor, het springt in de monitor op adres F97E (hex).
- PAUSE** : Ideaal voor een wachtlus... PAUSE X (X moet in de range zijn van 0-65535) wacht voor X/100 seconden alvorens met het programma verder te gaan (Op een 1Mhz. machine) Op een 2Mhz. machine is dit gelijk aan X/200 seconden.
Opmerking :CNTL-C is gedurende dit statement uitgeschakeld.

PUTAT : Bv. PUTAT X,Y,"hallo!" zet het woord 'hallo!' op het scherm met de 'H' op de coördinaten X,Y. Na de tweede komma is het commando hetzelfde als PRINT.De afstand van de coördinaten X,Y zijn gerekend vanaf de linker- en bovenkant van het huidige window. 0,0 is dus linksboven op het scherm.

INAT : zet de inhoud van een bepaalde schermpositie in een variabele.Bv. INAT X,Y,A verplaatst de cursor naar de positie aangegeven door X,Y en zet de inhoud van deze schermpositie in de variabele A. Na de tweede komma werkt het commando hetzelfde als INPUT.De coördinaten worden ook weer bepaald vanaf de linksboven positie van het huidige window.

CALL : CALL X springt naar de machinetaal-routine op adres X en voert de routine uit.Wanneer deze routine gereed is springt de computer terug naar het BASIC programma.
Onjuist aanroepen van een machinetaal-routine zal resulteren in een "hang-up" van het systeem.

MC : Een sterk statement om vanuit BASIC machinetaal programma's in het geheugen te zetten. Bv. MC 16384,"2000FD8DDCDD060" zet een machinetaal programma aangewezen door de string,in het geheugen op adres 16384.De string moet bestaan uit 2 cijferige hexadecimale getallen,welke achter elkaar geplaatst worden. In het bovenstaande voorbeeld wordt het getal 20(hex) (de eerste twee cijfers),op adres 16384 geplaatst;het volgende getal (DD) op adres 16385;het volgende getal(FD) op adres 16386;en zo verder.
Merk op dat er geen komma's nodig zijn tussen de getallen.

PUSR : PUSR X zet de pointer van de USR routine op X.
(alleen getallen tussen 0 en 65535 toepassen)

EGOTO : EGOTO l, waarbij l kan bestaan uit een lijnnummer,label of expressie.(zie LABELS),zet de pointer het lijnnummer waar het programma naar toe springt,wanneer er een error optreedt in een BASIC programma of in de directe mode.
Wanneer de jump naar een denkbeeldig lijnnummer gemaakt wordt,zal de computer even stoppen wanneer er een error optreedt.CNTL-C verhelpt dit.EGOTO 0 zet de error-jump af.

ELINE : ELINE X zet het lijnnummer in welke de error optreedt in de variabele X.Elk type variabele kan toegepast worden.

ETYPE : ETYPE X zet het code-nummer van de error die optreedt in de variabele X. Elk type variabele kan toegepast worden.
Hier volgt een opsomming van de error-code numbers :
0=NF, 2=SN, 3=NR (no repeat), 4=RG, 6=OD, 8=FC, 10=OV,
12=OM, 14=US, 16=OS, 18=DD, 20=/D, 22=ID, 24=TM, 26=LS,
28=ST, 30=CN, 32=UF.

VDU : VDU X,Y,C plaats het karakter,aangewezen door C,op de coördinaten aangegeven door X,Y (afstand in karakters vanaf links,en afstand gerekend van de onderkant van het window.
De startpositie 0,0 (linksonder) wordt ge-set door BLG.

BLG : BLG X zet het 0,0 coördinaat van het grafisch window gelijk aan de schermpositie X. Bij koude start is de locatie 0,0 gelijk aan de uiterste linksonder positie van het scherm.

TEST : TEST X,Y,A zet de waarde van het karakter op de positie X,Y in de variabele A.Elk type variabele kan gebruikt worden, zelfs strings!.

HLIN : HLIN X,Y,A,C tekent een horizontale lijn van A karakters lang, met karakter C en het uiterste linker karakter op het coördinaat aangegeven door X,Y

VLIN : VLIN X,Y,B,C tekent een verticale lijn van B karakters lang, met karakter C en het onderste karakter op het coördinaat aangegeven door X,Y

BLK : BLK X,Y,A,B,C tekent een blok met karakters aangegeven door C, op een dusdanige wijze dat het blok A karakters lang is, B karakters hoog en het linker-onder karakter op het coördinaat aangegeven door X,Y

DRAW : DRAW maakt gebruik van de mogelijkheid om strings in een bepaalde vorm te verpakken op een dusdanige wijze dat er besturingstekens kunnen worden meegegeven. De mogelijkheid bestaat hiermede om met behulp van karakters bepaalde voorprogrammaerde vormen te tekenen op het scherm. Men noemt dit het gebruik maken van "SHAPE STRINGS". Elke voorgevormde string bestaat dan uit een aantal normale karakters en pijl-tekens welke de richting aangeven. Hier volgt een korte beschrijving hoe een en ander in zijn werk gaat ;

Het commando zal eerst een denkbeeldige cursor op het startpunt projecteren. Bv. DRAW X,Y,A\$ houdt in dat de cursor begint op het coördinaat X,Y Van hier uit wordt de string afgewerkt. Wanneer er in de string een pijl-teken voorkomt, gaat de cursor in de richting van die pijl. Elk ander teken wat voorkomt in de string zal dan op de plaats van de cursor worden afgedrukt. Wanneer het eind van de string bereikt is, stopt het commando DRAW.

De pijl-richting kan alleen omhoog, omlaag, links en rechts zijn, en dus NIET diagonaal...

Elk pijl-teken is vanaf het keyboard in te geven ; omhoog=CNTL-P, omlaag=CNTL-T, links=CNTL-V, en rechts=CNTL-R.

Een voorbeeld :

A\$="A(rechts)B(omlaag)C(links)D", DRAW 10,10,A\$ tekent dan een vierkantje op het scherm, op een zodanige manier dat de A op het coördinaat 10,10 gezet wordt, B een positie rechts hiervan, C staat onder de B en D staat dan links van C en onder de A.

Wanneer deze zgn. "Shape strings" correct gebruikt worden kunnen zij zeer veel POKE commando's vervangen om tekenwerk op het scherm te verrichten...

Zij doen hetzelfde werk, maar veel sneller en... met minder geheugenruimte....!!

UNDRAW : Los van het feit dat wanneer er in een string een normaal karakter gevonden wordt er een spatie op het scherm gezet wordt, werkt dit commando UNDRAW op dezelfde wijze als DRAW. Het kan uitgelegd worden als een commando dat elk getekend karakter van het scherm verwijdert.

Bijvoorbeeld; UNDRAW 10,10,A\$ (A\$ is dezelfde als gebruikt bij DRAW) zal het eerder getekende karakter vierkant weer verwijderen.

STORE : Wanneer dit statement samen gebruikt wordt met DRAW en UNDRAW, bestaat de mogelijkheid om de "Shapes" niet destructief over het gehele scherm te verplaatsen....

STORE X,Y,A\$ (waarbij A\$ een "shape" string is), plaatst het gevormde patroon op het scherm in de "shape string". Het kan eenvoudigweg gezien worden als het omgekeerde van het statement DRAW, m.a.w. in plaats van het plaatsen van karakters uit de string op het scherm, worden nu de karakters zelf vanaf het scherm gehaald en in de string gezet.

Ik hoop met onderstaand programma een en ander te verduidelijken.

```

10 A$="A(rechts)B(omlaag)C(links)D"
20 T$=A$+" " :REM anders verandert de BASIC het programma en niet
   alleen de variabele
30 FOR X=1 TO 20
40 STORE X,10,T$: REM slaat elk karakter dat gevonden wordt op
50 DRAW X,10,A$ : REM tekent in de "shape"
60 PAUSE 10      : REM om het geheel wat te vertragen...
70 DRAW X,10,T$ : REM zet het opgeslagen karakter weer terug
80 NEXT X

```

Dit programma schuift de gevormde karakters van links naar rechts op het scherm, zonder de oorspronkelijke teksten af te breken.

LABELS

Dit extra gegeven voegt een geheel nieuwe dimensie aan uw programmering toe.

U bent nu in staat om gestructureerd programma's te maken. Met andere woorden; het totaal nietszeggende commando GOSUB 1670 wat de instructie laat zien, kan nu vervangen worden met een veel begrijpelijker commando, Bv. : GOSUB display instructies. De LABELS mogen zolang zijn als ze ingevoerd kunnen worden... De enige voorwaarden zijn :

- De label moet beginnen met een lower-case letter,
- Er mogen geen punt-komma's (;) in voorkomen.
- Los hiervan kunnen zij verder elk karakter bevatten...!
- Het lijnnummer waar het label naar verwijst moet met dezelfde label beginnen, en zouden er meerdere regels met hetzelfde label voorkomen, dan springt het programma naar de regel met het laagste lijnnummer.
- GOTO 'print'; springt naar een lijn beginnende met het label 'print', of naar elk ander label dat begint met 'print'

Bijvoorbeeld ; 'print menu'
Eventuele spaties in een label worden overgeslaan.
Labels kunnen gebruikt worden in de volgende statements ;
GOTO, GOSUB, RESTORE, ON..GOTO, ON..GOSUB, EGOTO

RESTORE

Dit commando werkt nu nog als met de oude BASIC, maar werkt nu ook met lijnnummers uitgevoerd als labels, getallen of expressies. Bijvoorbeeld; RESTORE 100 zet de DATA-pointer op de lijnnummer 100 'RESTORE data' zet de DATA-pointer op de regel beginnende met dat label. Bevat de regel, waar naar verwezen wordt, geen DATA commando, dan zal de computer vanaf die regel gaan zoeken, regel voor regel, totdat een DATA-commando gevonden wordt.

EXPRESSIES

Om EXBASIC goed te laten werken, mogen er in de nieuw toegevoegde commando's en statements geen expressies gebruikt worden welke in de nieuwe commando's zelf voorkomen. Een uitzondering hierop is het statement UNTIL en ELSE, welke nu ook voorzien zijn van tokens. Het is zeer onwaarschijnlijk dat dit een hinder zal opleveren.

TOEVOEGEN VAN EXTRA COMMANDO'S

Het toevoegen van extra commando's is erg gemakkelijk, omdat EXBASIC het begin van zijn commando-tabel op adres \$02B0 (hex) heeft, en al de instructie-decoding gebeurt via de EXBASIC.

U hoeft alleen op te geven wat de naam van het toe te voegen commando is, en waar de respectievelijke routine in het geheugen is opgeslagen.

Na koude-start ziet de tabel er als volgt uit; \$02B0 FF 96 E2

(De zelfde manier van werken is gebruikt als in de CEGMON-monitor voor het displayen van hexadecimale getallen)

In EXBASIC geeft het getal # \$FF aan dat dit het eind van de tabel is, en de getallen # \$96 en # \$E2 geven aan waar de volgende tabel staat.

In dit geval is het \$96E2 wat de tabel is in de EXBASIC.

Opmerking; wanneer er een nieuwe tabel aangemaakt wordt, dient deze te beginnen met het getal # \$AA (zie evt. de tabel op \$96E2.)

Zoals eerder aangehaald, om een commando toe te voegen dienen naam en adres toegevoegd te worden in de tabel.

Dit wordt als volgt opgebouwd; eerst de nieuwe naam, daarna het getal # \$00 om aan te geven dat dit het eind van de commando-naam is, hierna het adres min 1, wat het startadres van de routine aangeeft.

Bijvoorbeeld;

Om het commando 'KEY' bij te maken (dit wacht totdat er een toets gedrukt wordt) dient de tabel middels de CEGMON-monitor aangemaakt te worden.

.02B0 'KEY' 00,FC,FF,FF,96,E2 dit verandert de tabel en wordt gelezen als; \$02B0 4B 45 59 00 FC FF 96 E2.

De getallen 4B, 45 en 59 zijn de letters K,E,Y en het getal 00 geeft aan dat dit het eind van de naam is.

Het opgegeven adres is \$FCFF+1 (dit is \$FD00) wat in de CEGMON-monitor de keyboard routine is om te wachten op een toetsaanslag.

De laatste drie getallen geven aan dat dit het eind van deze tabel is, (FF 96 E2) en tevens de start van de volgende tabel op \$96E2.

Een goede kennis van de 6502 machinetaal (instructie-set) en de werking van de interpreter is eigenlijk wel vereist. Maar door de diverse routine's zoals gebruikt in deze EXBASIC te bestuderen (bijvoorbeeld het 'VDU' commando op \$92EE) wordt de manier van toevoegen van commando's welk duidelijk.

ANDERE VOORDELEN VAN EXBASIC

1. Het start zichzelf volledig op, er hoeft niet, zoals bij andere systemen, ge-POKED te worden.
2. List een regelnummer wanneer er een error geconstateerd wordt. Wanneer de EGOTO routine niet gebruikt wordt, dan zorgt het programma ervoor dat naast de vermelding van de error OOK nog eens de VOLLEDIGE REGEL waar de error in optreedt afgedrukt wordt.
3. Ook in deze EXBASIC zijn de foutmeldingen aangepast en is de foutmelding in leesbare tekst uitgevoerd.
4. GRAFISCHE karakters kunnen nu middels de REPEAT toets ingegeven worden....
5. Het commando LIST kan nu IN een programma gebruikt worden
6. Terugmeldingen in de directe mode zijn nu wat vriendelijker.. met name OK is nu vervangen door Ready.
7. Het NULL commando is uitgebreid en kan nu gebruikt worden van D t/m 255.

VERSCHILLEN TEN OPZICHT VAN DE OUDE BASIC

Om de diverse nieuwe zaken in EXBASIC goed te laten functioneren, zijn er enkele kleine dingen veranderd.

1. CNTL-0 stopt nu niet meer het printen (was al)
2. Verschillende page-2 locaties zijn nu in gebruik door EXBASIC. Het gebied waar het om gaat is \$0235 - \$02CF exclusief de extra commando tabel, wordt dit gebied nu gebruikt voor het decoderen van commando's en labels, opslag van de diverse pointers en de REPEAT...UNTIL stack. Enkele locaties zijn niet gebruikt, en die kunnen gebruikt worden voor een eventueel verdere uitbreiding van EXBASIC.
3. Dit geldt alleen voor degenen die gebruik maken van vernieuwde SAVE en LOAD routines, waarbij het programma niet ge-list wordt. Wanneer een pre-EXBASIC programma geladen wordt middels deze vernieuwde routine dan kunnen er twee commando's verwisseld worden, dit zijn NULL en CLEAR. Hiervoor komt dan resp. ELSE en UNTIL voor in de plaats... Bij het runnen van het programma levert dit natuurlijk fouten op.
Om dit te omzeilen kunt u het best het programma gewoon laden, en daarna met bv. behulp van FIND de betreffende regels opzoeken en hierna verbeteren.

DE HARDWARE WIJZIGINGEN.....

Uitgaande van de standaard BASIC-ROM'S van de OSI (wie heeft dit nog) bestaande uit 4 maal een 2K ROM resp. op de adressen A000-A7FF, A800-AFFF, B000-B7FF en B800-BFFF wordt hiervan de EERSTE ROM vervangen (A000-A7FF) en komt er op adres \$9000 een 2K Eprom bij. Dit houdt gelijk in dat personen die op adres \$9000 de Toolkit hebben zitten, deze hier weg moeten halen.

De Eprom die er op adres \$9000 bijkomt is een 2716 (2K) en bevat op de adressen \$9000-97FF de EXBASIC toevoegingen.

Het "gat" wat nu overblijft van adres \$9800-9FFF is bedoeld voor een latere uitbreiding van de EXBASIC. (Zit er aan te komen !)

Voor degenen die reeds eerder de oude BASIC-ROMS vervangen hebben door 2 maal een 4K Eprom resp. A000-AFFF en B000-BFFF, zal dus de eerste 4K Eprom aangepast moeten worden voor EXBASIC.

BELANGRIJK.....!!!!

DEZE EXBASIC WERKT ALLEEN GEZAMELIJK MET DE CEGMON-MONITOR-ROM

Dit komt omdat velerlei commandos welke nu nieuw toegevoegd zijn gebruik maken van vernieuwde routine's in de CEGMON-monitor

U dient dus voor gebruikmaking van deze EXBASIC ook de monitor-Rom te vervangen door de CEGMON-versie.

OPMERKING VOOR GEBRUIKERS MET MEER DAN 32K RAM

Ondanks het feit dat de locatie van EXBASIC in het geheugen op adres \$9000 (hex) tot 97FF zit, vindt er van diverse zaken een echo-ing plaats in het gebied van \$8000-9FFF. Op een computer met meer dan 32 K RAM vrij beschikbaar geheugen (en dus in het gebied 8000-9FFF) kunnen er dus problemen ontstaan met deze extra RAM (>32K) en met de EXBASIC.

Voorwaarde is dus dat het gebied van \$8000-9FFF goed uitgecodeerd dient te worden.

Deze verbeterde uitcodering kan bijvoorbeeld geschieden middels de uitbreidingprint "HCT 1" zoals eerder beschreven in deze OSI boekjes (blz. AA36 - AA41).

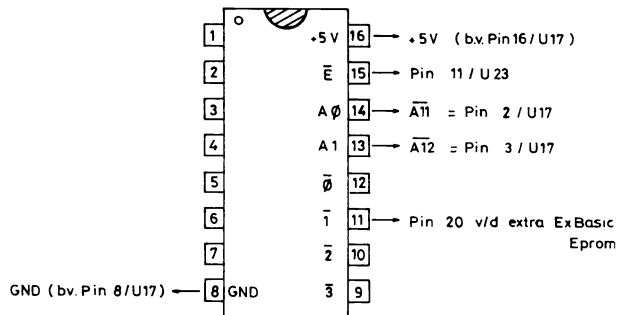
Een andere mogelijk is zelf een verbeterde uit-decoding te maken via de hieronder gegeven aanwijzingen.

Als u deze verbeterde manier van decoding toepast, kunt u toch nog tot zo'n 36 K RAM gaan voor BASIC-programma's.

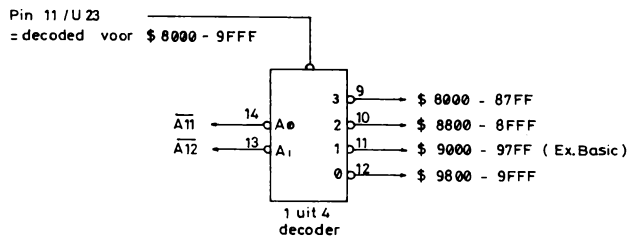
Eventueel kunt u zelfs nog de 2K extra toepassen op \$9800-9FFF (hex) als u niet voornemens bent om de uitbreiding met EXBASIC 2 ook toe te passen. Deze extra 2K kunt u dan gebruiken voor evt. machinetaal programma's in Eprom.

VERBETERDE ADRES DECODING (o.a. t.b.v. EXBASIC)

Benodigd is een extra 74LS139 of 74139 en een vrije proto plaats op het main-board of een klein stukje vero-board.



Circuit beschrijving;



WAT KOST DEZE EXBASIC UITBREIDING.....

Ja, dat is natuurlijk een vraag die eenieder wel interesseert... Welnu dat zal u meevallen.

Een totaal pakket aan Eproms bevat;

1x 2716 tbv de CEGMON-MONITOR (Die nodig is...)

1x 2732 tbv de uitbreiding op 9000(hex) (incl. ruimte voor EXBASIC versie 2) of in plaats hiervan 1x 2716

1x 2716 of 1x 2732 afhankelijk van de momentele uitvoering van de standaard BASIC-Roms (Eproms)

De kostprijs totaal is afhankelijk van de momentele prijs van de Eproms, en ligt globaal op Fl.50,- (all in)

U mag echter ook zelf de Eproms meebrengen en door ons laten programmeren.... In principe kan dat dan op de OSI dagen in Utrecht.

U betaalt dan slechts Fl.10,- als bijdrage in de door ons gemaakte kosten,voor het verkrijgen van deze uitbreidingen.

WAAR TE BEKOMEN

De distributie van deze CEGMON-MONITOR en EXBASIC wordt verzorgt door Ko van EKEREN (van de MOSI,weet u wel...)

Het adres is; Reijerscop 26, 3454 HX De MEERN.

Bellen mag ook maar dan graag na 18.00 uur.....

NABESCHOUWING...

Gedurende enkele maanden is er al druk ge-experimenteerd met deze CEGMON-MONITOR en EXBASIC versies door Cor Struyk en Ko van Ekeren, en wij zij tot de slotsom gekomen dat vooral het GRAFISCHE deel van de OSI nu wat beter uit de verf komt.

Ook de diverse handige zaken als REPEAT...UNTIL en IF..THEN..ELSE hebben al vollop toepassing gevonden,en zeer zeker niet te vergeten de toepassingsmogelijkheid van LABELS.....

Toch zullen er nog wel wat zaken uitgezocht moeten worden.

Het niet zo gemakkelijk meer kunnen gebruiken van de (verplaatste) TOOLKIT wordt al driftig bestudeert en inmiddels is ook Jos Burghouts al druk doende om een en ander te combineren zodat de nu ontbrekende zaken uit de Toolkit en de Extended monitor weer toegankelijk worden.

Al met al is het even wennen,maar we zijn er zeker van dat de diverse nieuwigheden zeker toepassing zullen vinden.

In elk geval succes er mee, en graag bereid voor meer tekst en uitleg op de OSI bijeenkomsten.

Groeten;
Cor struyk, Geertruidenberg.

OSI FOEL

EXTENDED BASIC VERSIE 2 LAATSTE ONTWIKKELINGEN.....
 =====

Doordat de EXBASIC versie 1 alleen samenwerkt met de CEGMON-monitor rom, blijkt in de praktijk dat een groot aantal van de oude programma's niet, of niet goed meer functioneert.

Ook de TOOLKIT en EXTENDED MONITOR blijken in de praktijk niet zo gemakkelijk meer gebruikt te kunnen worden.

Dit is in hoofdzaak te wijten aan de vele GEHEEL GEWIJZIGDE routines in de CEGMON monitor.

Een bijkomend voordeel is echter wel dat de CEGMON monitor veel SNELLER is dan de oude monitor versies.

Wij kunnen ons echter heel goed voorstellen dat u er weinig voor voelt om alle oude programmatuur om te schrijven naar de nieuwe video-routines.

Vandaar dat wij in samenwerking met Jos Burghouts een oplossing bedacht hebben om toch de EXBASIC toe te kunnen passen op een bestaande monitor-rom (versie Henk Wevers)

Jos is op dit moment vollop doende om alle NIEUWE commando's van de EXTENDED BASIC om te schrijven naar de reeds bekende routines van de STANDAARD (Henk Wevers) monitor.

Het enige wat we daarbij verliezen, is het snelheids voordeel wat met de CEGMON monitor verkregen werd.

Dit houdt tevens in dat, wanneer u geïnteresseerd bent in de EXBASIC uitbreiding, maar op de OUDE monitor wilt voortborduren dat u nog even geduld zal moeten hebben met de aanschaf van de EXBASIC uitbreiding....

Op dit moment is de omschrijving van EXBASIC versie 1 naar versie 2 (voor gebruik met de standaard monitor) voor ongeveer twee-derde uitgezocht.

Wij hopen dan ook op korte termijn met een volledig werkende versie 2 te kunnen presenteren.

Het staat u echter vrij om nu reeds de EXBASIC uitbreiding evt. met een CEGMON monitor aan te schaffen om al met de EXTRA (39) commando's vertrouwd te geraken...

Naderhand programmeren wij dan uw EPROMS gratis om naar de versie 2.

Graag zijn wij bereid om nadere informatie te verstrekken.....

Met vriendelijke OSI groet,

Cor Struyk
 Ko van Ekeren.

