JOHN HERMANS	SCHETS EN SCROLL	EC 9
	PUZZLE-KUBUS	EC11
	VLUCHTSIMULATOR	EC15
	WEEFPATRONEN	EC21
HANS COENEN	DOBBELSTEEN (FORTH)	EC21
	SIMPLE GAME (FORTH)	EC22

ED. ZAKELIJK GEBRUIK

EE. UTILITIES

JOS COURBOIS E.A.	HOBBYSCOPP BASICLOADER VOOR OSI	EE 1
JAN EN BAS EDIXHOVEN	QUICKSAVE	EE 5
	QUICKSAVE (VERVOLG)	EE11
A.L. MATHLENER	DISASSEMBLER 4.0	EE19
HENK DE WAL	MORSE TUTOR	EE22
MARTIN GOERLITZ (BEW. J.A. V/D LINDEN		
	6502-DISASSEMPLER	EE23
JOHN HERMANS	FORTH-HEXDUMP OP HET SCHERM	EE31
HANS DE JONG	OSI MINISPACER	EE32
A. MATHLENER	TEXT-SEARCHER	EE33
	HEX-SEARCHER	E34
FORTH INTEREST GROUP	FORTH ASSEMPLER EDITOR	EE35
HANS COENEN	SCREEN EN PRINTER HANDLER (FORTH)	EE37
	RANDOM NUMBER GENERATOR (FORTH)	EE37
TON HELWIG	PROGRAMMA'S MET WAIT	EE38
FORTH INTEREST GROUP	FORTH LINE EDITOR	EE39
HENK SCHAER	MINIDATABASE	EE45

F. TALEN

FA. MACHINETAAL

JOS BURGHOUTS	WAT KAN IK MET MACHINETAAL	FA 1

FB. BASIC

FC. PASCAL

FD. FORTH

JOHN HERMANS	FORTH	FD 1
	FORTH INFO	FD 6
	DISK-LOZE FORTH	FD 7
JOS BURGHOUTS EN JOHN HERMANS	OPZET VAN EEN FORTH	
	T-APE O-PERATING S-YSTEEM	FD12
JOS BURGHOUTS	FORTH INTRODUKTIE	FD15

FE. LISP

OSI POEL

OSI POEL

HELAAS ZIJN ER BIJ HET OSIGG-BOEKJE 1/2 ENKELE FOUTEN OPGETREDEN IN DE PAGINA-NUMMERING.
OM DIT ZO VOLLEDIG MOGELIJK TE KUNNEN HERSTELLEN ZIJN IN DIT BOEKJE (NR. 6) EEN AANTAL PAGINA'S OPNIEUW OPGENOMEN.
DE OUDE PAGINA'S, WAARVOOR NU NIEUWE ZIJN OPGENOMEN, KUNT U HET BESTE BEPLAKKEN MET EEN STUKJE WIT PAPIER.

TEVENS MOET U DE VOLGENDE PAGINA'S EEN ANDER NUMMER GEVEN.

EF 1	T/M	EF 3	MOET ZIJN	EA 9	T/M	EA11
EM 1	T/M	EM 8	MOET ZIJN	EE23	T/M	EE30
FM 1	T/M	FM 4	MOET ZIJN	FA 1	T/M	FA 4

DE RUBRIEK FA ASSEMBLER HEET VOORTAAN FA MACHINETAAL \

A ALGEMEEN

AA Systeembeschrijvingen

AB Modificatie overzicht

AC Systeem bedienings info

AD Memory map

AE Diversen

OSI Uitbreidingen

OVERZICHT

Onze 6502 stelt ons in staat om een geheugengebied van 64K te adresseren. Of het daarbij gaat om schrijf-leesgeheugen of alleen-leesgeheugen doet niet ter zake, we kunnen 65536 adressen tot op een bit precies aanspreken.
We willen het nu niet hebben over de inhoud van deze adressen en hetgeen we daarmee doen. Eerst willen we maar eens onze geheugenkaart bekijken.
Fig.1 is een komplete plattegrond van onze 64K met daarin aangegeven de decimale en hexadecimale adressen per 1K.
Vergeleken met de plattegrond van een stad vinden we zelfs overeenkomsten.

er is een bestuurscentrum	-het keyboard
een afd.gemeentewerken	-Basic en Monitor
de woonwijken	-het schrijf-leesgeh.
in- en uitvoer faciliteiten	-de ACIA

Op de kaart zijn deze gebieden aangegeven, we vinden:

Basic	A000 - BFFF
Video	D000 - D7FF (voor de 48 char.versie)
Monitor	F800 - FFFF

maar ook zijn er adressengebiedjes voor:

Keyboard	DC00 - DFFF
ACIA	F000 - F3FF

Ho ho zegt U nu, mij zijn alleen maar twee adressen bekend:
DF00 voor het keyboard en F000 voor de ACIA.
Om U duidelijk te maken wat er aan het handje is, gaan we nog even terug naar het stad voorbeeld.

OSI zegt: het bestuurscentrum, dat hoeven we enkel maar aan te duiden, dat kent iedereen, preciese straat en huisnummer aanduiding zijn niet nodig.
Hetzelfde geldt voor de in- en uitvoerfaciliteiten. De haven weet iedereen te liggen.
OSI kan dit royale standpunt innemen omdat er op het SUPER-BOARD met 4 of 8K geheugen toch nog genoeg vrije adres ruimte is.
Een zeer aantrekkelijk voordeel voor OSI is echter ook, dat deze adressen niet tot op het laatste bit hoeft te worden uitgeveterd, maar slechts z.g. partieel gedecodeerd behoeft te worden.

DECODEREN

Waarom decoderen? Wel, omdat we uit 16 adreslijnen niet zonder meer 65536 adressen kunnen toveren.
In fig.2 vinden we een hele pagina geheugen weergegeven, dit zijn 256 geheugencellen.

We zien 0 t/m F genummerde kolommen en 0t/m F genummerde rijen. Het geheugengebied loopt van 00 t/m FF en kan met een byte worden gedefinieerd. Dit is dan de LOW byte.
Een heel adres bestaat uit een LOW en een HIGH byte.
De HIGH byte geeft aan welke van de mogelijk adresseerbare 256 blokken van 256bytes elk, aangesproken dient te worden.

CHIPS

Al die tijd praten we nu over geheugencellen die in matrix vorm gerangschikt zijn.
De RAM chips die we momenteel verwerken zijn doorgaans 2114's U heeft er steeds 2 nodig om 1K. geheugen te vormen. Dat komt omdat ze een matrix hebben van 1024 X 4, d.w.z. elke chip heeft maar 4 data uitgangen en van elk 1K paartje neemt dan ook één de datalijnen 0 t/m 3 en de ander 4 t/m 7 voor tijn rekening.
ROM en EPROM chips van 2K. hebben een matrix van 2048 x 8.
Om zo'n matrix uit te lezen hebben we adreslijnen nodig vanaf AØ (adreslijn 0) en opwaarts. Met FF kwamen we op 256 (AØ t/m A7) en elke volgende adreslijn verdubbelt het gebied.
Voor een 2114 zijn dan ook AØ t/m A9 aangesloten en een 2K. ROM of EPROM heeft er dan de A10 ook nog bij nodig.

Het mechaniek oftewel de logika die "de" bewuste cel op de chip selekteert, is op de chip aanwezig en varieert in snelheid van werken. De fabriek selekteert uit haar produktie de snellere chips en laat zich daar goed voor betalen.
In ieder geval hoeven we ons niet te bekommeren om de reeds op de chip aangesloten adreslijnen. De chip weet zelf wat daarmee aangevangen moet worden.

Iets anders is, dat alle geheugenchips op dezelfde AØ t/m A9 aangesloten zijn. Hoe wordt die ene (of dat paartje van 1K) tussen alle andere geselekteerd?

CHIP ENABLE

Geen nood, in dat akelig logische machien dat computer heet, en nou nooit eens weet wat je eigenlijk bedoelt, is ook dat exact uitgeknobbeld.
Als namelijk AØ t/m A9 aangesloten zijn op een 2114, hebben we nog een slordig bundeltje adreslijnen in de hand.
Uit A10 t/M A15 wordt dan ook een uniek Chip Enable signaal gevormd en aan het chip-paar aangesloten.
Dergelijke signalen worden opgetekend als:

CE en aangeduid als:"hoog" - "een" - "true" of als:

$\overline{CE}$ en aangeduid als:"laag" - "nul" - "false".

Van elk door de microprocessor opgegeven adres wordt dus eerst het"hoge" deel omgezet in een chip enable en daarna op de chip de bewuste geheugencel aangesproken.

wRITE ENABLE

Zolang we nog met RAM te doen hebben, is er echter een mogelijkheid dat de microprocessor iets wil schrijven in het geheugen. In dat geval moet nog een signaal gegeven worden.

Lezen gaat n.l. zonder meer, de cel wordt aangesproken en een kopie van de geheugencel wordt door de chip op de databus aangeboden. De inhoud van de cel verandert niet, ook niet als er honderden malen uit gelezen wordt.

Schrijven echter, verandert de cel-inhoud wel en dus benodigt de chip hiertoe het Write Enable signaal.
Als dit dan samen met CE en adres wordt aangeboden, kopieert de chip de data die op de databus worden aangeboden in de geheugencel.

GEHEUGENINDELING

Laten we fig.1 nogmaals bekijken. We zien dat het gebied tot aan A000 (BASIC) een mooi aaneengesloten gebied is, dat uitnodigt om met RAM gevuld te worden.
Misschien lukt het om de "Landman" 32K op 2MHz te laten draaien, dan zou dat een prima mogelijkheid zijn om tot en met 9FFF oftewel 40K. geheugen te creeren

Voor de"gewone jongens"onder ons die niet wat losse pegulanten hebben om een discdrive aante schaffen is het gebied"boven" de BASIC interressant om enkele EPROMS van 2K. (2716) in onder te brengen.

De"Extended Monitor" op E800 is een utility die al algemeen bekend is.
Zelf heb ik, als machinetaal fan, de Assembler-Editor in Eprom gezet, weliswaar nog op zijn oude adres 0240, maar een Move-opdracht van de Ext.Mon. brengt hem nu in een oogwenk op zijn plaats in RAM.

Dit zijn enkele voorbeelden van Eprom mogelijkheden. Het "branden" van een Eprom is,nu het goedkope programmertje van Zero SC op de markt is, geen kostbare zaak meer.

Weer bepalen we ons tot de geheugen kaart. De ogenschijnlijk vrij zijnde 5 stuks 2K. gebieden zijn er echter maar 4.
In D800 t/m DFFF bevindt zich n.l. het keyboard adres DF00.
Ook F000 t/m F7FF is een gebied waarin zich enkele adressen bevinden, n.l. de ACIA op F000 en van de 2P de PIA op F700.

Hier komt dan het verhaal over het partieel decoderen tevoorschijn. Met deze uitdrukking wordt bedoeld, alleen maar de allernodigste, hoogste adreslijnen te gebruiken om een adres te vormen.

OSI gebruikt in deze gevallen A15 t/m A10 en daar ze dan zo grof selekteren, kiezen ze hun adressen maar een goed eind uitelkaar.
De reden voor deze manier van werken is eenvoudig: elke chip meer op het board kost plaats en geld!

Hoe zit dat nou met die I/O units?
Het keyboard heeft een adres nodig, uit dit adres ontstaan, gecombineerd met het R/$\overline{W}$ (Read-Write) signaal, 2 andere signalen n.l. RKB (Read Key Board) en WKB (Write Key Board)

De ACIA heeft twee adressen nodig n.l. het basisadres ACS samen met de A11 en A10 als Chip Select F000 en daarnaast nog de AØ om F001 te kunnen vormen.

Een PIA heeft zelfs vier adressen nodig, per poort 2. Buiten het basisadres worden ook nog AØ en A1 gebruikt

Het stukje F000 t/m F7FF ga ik intensief voor dergelijke adresseringen gebruiken. Daartoe zal ik het beter moeten decoderen als OSI dat totnogtoe deed.

Voor mijzelf heb ik vastgesteld dat ik het D800 t/m DFFF stukje wil"hebben". Consequentie is dat ik het keyboard adres moet verhuizen. Grote consequentie daaruit is weer: Basic programmas waarin 57088 genoemd wordt, moet ik veranderen.

De hardware zijde van dit verhuizen is geen heksentoer. Het is wel een verhaal op zich, dat U nog wel van me krijgt.

Als U nu teleurgesteld bent over de mogelijkheden van Uw computer, n.l. dat deze "maar" 64K kan adresseren, dan kan ik U troosten. Er is op betrekkelijk gemakkelijke manier nog een of twee adreslijntjes "bij te maken" en U weet: elke adreslijn meer verdubbelt het adresgebied!

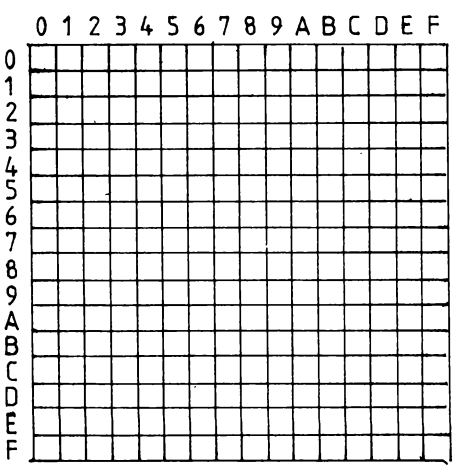


FIG 2

01 0000-03FF 0-1023	02 0400-07FF 1024-2047	03 0800-0BFF 2048-3071	04 0C00-0FFF 3072-4095
05 1000-13FF 4096-5119	06 1400-17FF 5120-6143	07 1800-1BFF 6144-7167	08 1C00-1FFF 7168-8191
09 2000-23FF 8192-9215	10 2400-27FF 9216-10239	11 2800-2BFF 10240-11263	12 2C00-2FFF 11264-12287
13 3000-33FF 12288-13311	14 3400-37FF 13312-14335	15 3800-3BFF 14336-15359	16 3C00-3FFF 15360-16383
17 4000-43FF 16384-17407	18 4400-47FF 17408-18431	19 4800-4BFF 18432-19455	20 4C00-4FFF 19456-20479
21 5000-53FF 20480-21503	22 5400-57FF 21504-22527	23 5800-5BFF 22528-23551	24 5C00-5FFF 23552-24575
25 6000-63FF 24576-25599	26 6400-67FF 25600-26623	27 6800-6BFF 26624-27647	28 6C00-6FFF 27648-28671
29 7000-73FF 28672-29695	30 7400-77FF 29696-30719	31 7800-7BFF 30720-31743	32 7C00-7FFF 31744-32767
33 8000-83FF 32768-33791	34 8400-87FF 33792-34815	35 8800-8BFF 34816-35839	36 8C00-8FFF 35840-36863
37 9000-93FF 36864-37887	38 9400-97FF 37888-38911	39 9800-9BFF 38912-39935	40 9C00-9FFF 39936-40959
41 A000-A3FF 40960-41983 Basic	42 A400-A7FF 41984-43007 Basic	43 A800-ABFF 43008-44031 Basic	44 AC00-AFFF 44032-45055 Basic
45 B000-B3FF 45056-46079 Basic	46 B400-B7FF 46080-47103 Basic	47 B800-BBFF 47104-48127 Basic	48 BC00-BFFF 48128-49151 Basic
49 C000-C3FF 49152-50175	50 C400-C7FF 50176-51199	51 C800-CBFF 51200-52223	52 CC00-CFFF 52224-53247
53 D000-D3FF Video 53248-54271 memory	54 D400-D7FF Video 54272-55295 memory	55 D800-DBFF 55296-56319	56 DC00-DFFF 56320-57343
57 E000-E3FF 57344-58367	58 E400-E7FF 58368-59391	59 E800-EBFF 59392-60415	60 EC00-EFFF 60416-61439
61 F000-F3FF 61440-62463	62 F400-F7FF 62464-63487	63 F800-FBFF 63488-64511	64 FC00-FFFF 64512-65535

F000-F001. ACIA cass port

DF00 - keybord.

OSI UITBREIDINGEN

HEXADECIMAAL

het 16 tallig stelsel dient als ezelsbrug voor het optekenen van grote aantallen eenen en nullen, zoals dat bij adressen nu eenmaal nodig is.
Het bekende tabelletje van 0 - 15 oftewel van Ø - F, dat hiernaast is afgebeeld, heb ik altijd bij de hand als ik op adressen en decoderingen ga zitten vossen.

HEX	BINAIR			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

DECODERINGSTECHNIEK

De moderne decoderhardware die ons ter beschikking staat maakt het gemakkelijk om meerdere adreslijnen te bundelen en daaruit unieke adressen te definieren.

Er zijn vrij gecompliceerde chips zoals:
- 74139 een tweevoudige - 2 lijn naar 4 lijn decoder
- 74138 een 3 lijn naar 8 lijn decoder
- 74154 een 4 lijn naar 16 lijn decoder

maar daarnaast kan de meervoudige-ingang NAND of NOR poort ook nog wel diensten bewijzen.

Om met de decoderchips goed te kunnen omgaan is het erg belangrijk om hun "waarheidstabellen" te kennen.

Aan de hand van enkele voorbeelden zoals die op het Superboard voorkomen, zal ik trachten met deze tabellen wat klaarheid in het decodergebeuren te brengen.

VOORBEELD I RAM DECODERING

Chip U23, een 74138 (3 naar 8 decoder) staat centraal in het decodeerproces van het S.B.
Deze chip heeft als ingang de drie hoogste adreslijnen A15 - A14 - A13 en verdeelt heel grof het hele 64K gebied in 8K. stukken.

Als U de waarheidstabel (Fig 4) en het schema met elkaar vergelijkt, ziet U dat aan alle enable voorwaarden op de meest directe manier is voldaan n.l. pin 6 hangt direct aan de +5V en pinnen 4 en 5 liggen aan GND.
De toestand van de lijnen A15-14-13 is nu verder bepalend ervoor welke uitgang Y van de 138 omlaag getrokken wordt. Lang niet alle uitgangen zijn benut, YØ waar de laagste 8K. dus OXXX mee geselekteerd wordt, regeert het hele, op het Superboard voorkomenede RAM; Y5 beslaat het hele BASIC gebied; Y6 is nodig om Video Ram en Keyboard te kunnen adresseren; Y7 tenslotte, hebben we nodig voor ACIA en Monitor adressering.

U weet, adresseren betekent, zoveel mogelijk adreslijnen bundelen, dus te precieser is het adres gedefinieerd.
We zien dat dan ook gebeuren als we de decodering van het RAM geheugen volgen.

De selektie op U23 was nog maar een heel grof gebeuren, slechts 3 van de 16 adresbits zijn bepaald.
Nog even kijken we welke adreslijnen vast op de 2114 geheugen chips aangesloten zijn: AØ t/m A9.
Dan missen we in ons complete adres nog de lijnen: A12 - A11 en A10.
Op chip U22 worden deze drie aangesloten. Zij verdelen nu de 1e 8K. weer in 1K. stukken, precies die mootjes geheugen die door een paartje 2114 beheerst worden.

Bekijkt U nu Fig 5 en het schema van aansluitingen op U22, dan ziet U daar al wat verschillen met de aansluitingen van U23.
Van de Enables ligt weliswaar pin 6 nog steeds aan +5V, pin 4 wordt nu door de uitgang van U23 gestuurd. Alleen als Y (de OXXX uitgang)"omlaag" gaat, wordt U23 ge-enabled.
Maar ook pin 5 moet"laag" zijn, anders werkt de chip nog niet. deze pin hangt aan de geinverteerde "Clock" ($\overline{Ø2}$)

Het waarom hiervan is als volgt: er is een bepaalde tijd nodig voor de microprocessor om een stabiel adres op de adresbus te zetten. Als U22 ge-enabled zou worden vóórdat dit adres stabiel is, zou een foutief adres worden gelezen en daardoor foutieve data op de databus worden gezet.
Door $\overline{Ø2}$ nu "hoog" te maken nadat het adres stabiel is, voorkomt de microprocessor deze troubles.

	INPUTS						OUTPUTS							
	ENABLE			SELECT			Y	Y	Y	Y	Y	Y	Y	Y
	G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	0	1	2	3	4	5	6	7
7 4	+ 5 V	G N D	G N D	A 15	A 14	A 13								
1	X	1	1	X	X	X	1	1	1	1	1	1	1	1
	0	X	X	X	X	X	1	1	1	1	1	1	1	1
3	1	0	0	0	0	0	0	1	1	1	1	1	1	1
	1	0	0	0	0	1	1	0	1	1	1	1	1	1
8	1	0	0	0	1	0	1	1	0	1	1	1	1	1
	1	0	0	0	1	1	1	1	1	0	1	1	1	1
	1	0	0	1	0	0	1	1	1	1	0	1	1	1
	1	0	0	1	0	1	1	1	1	1	1	0	1	1
	1	0	0	1	1	0	1	1	1	1	1	1	0	1
	1	0	0	1	1	1	1	1	1	1	1	1	1	0
PINOUT	6	4	5	3	2	1	15	14	13	12	11	10	9	7
16 = VCC 8 = GND												X = ONBEPAALD		

FIG **4**

	INPUTS						OUTPUTS							
	ENABLE			SELECT			Y	Y	Y	Y	Y	Y	Y	Y
	G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	0	1	2	3	4	5	6	7
7 4	+ 5 V	$\overline{0}$ X X X	$\overline{\phi}$ $\overline{2}$	$\overline{A}$ 12	$\overline{A}$ 11	$\overline{A}$ 10								
1	X	1	1	X	X	X	1	1	1	1	1	1	1	1
	0	X	X	X	X	X	1	1	1	1	1	1	1	1
3	1	0	0	0	0	0	0	1	1	1	1	1	1	1
	1	0	0	0	0	1	1	0	1	1	1	1	1	1
8	1	0	0	0	1	0	1	1	0	1	1	1	1	1
	1	0	0	0	1	1	1	1	1	0	1	1	1	1
	1	0	0	1	0	0	1	1	1	1	0	1	1	1
	1	0	0	1	0	1	1	1	1	1	1	0	1	1
	1	0	0	1	1	0	1	1	1	1	1	1	0	1
	1	0	0	1	1	1	1	1	1	1	1	1	1	0
PINOUT	6	4	5	3	2	1	15	14	13	12	11	10	9	7
16 = VCC 8 = GND												X = ONBEPAALD		

FIG 5

HEX	BINAIR			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

FIG 3

7 4 1 3 9	INPUTS			OUTPUT			
	ENABLE	SELECT		Y	Y	Y	Y
	$\overline{G}$	B	A	0	1	2	3
	$\overline{A}$ X X X	$\overline{A}_{12}$	$\overline{A}_{11}$				
	1	X	X	1	1	1	1
	0	0	0	0	1	1	1
	0	0	1	1	0	1	1
	0	1	0	1	1	0	1
	0	1	1	1	1	1	0

PIN	I	1	3	2	4	5	6	7
OUT	II	15	13	14	12	11	10	9

X = ONBEPAALD 16 = VCC 8 = GND

FIG **6**

HEX	BINAIR			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

FIG 3

7 4 1 3 9	INPUTS			OUTPUT			
	ENABLE	SELECT		Y	Y	Y	Y
	$\overline{G}$	B	A	0	1	2	3
	$\overline{F}$ X X X	A	A				
		9	8				
	1	X	X	1	1	1	1
	0	0	0	0	1	1	1
	0	0	1	1	0	1	1
	0	1	0	1	1	0	1
	0	1	1	1	1	1	0

PIN	I	1	3	2	4	5	6	7
OUT	II	15	13	14	12	11	10	9

X = ONBEPAALD 16 = VCC 8 = GND

FIG **7**

Aangezien het chip signaal "laag" moet zijn, wordt $\overline{\emptyset 2}$ geinverteerd met een 7404, die bovendien nog wat krachts-inspanning van de 6502 overneemt hiemee.
Het opnemen van de factor"Clock" in een adressering zult U regelmatig tegenkomen.

Selects op U22 zijn de geinverteerde adreslijnen. Noodzakelijk was dit niet, waarschijnlijk dient dit ter ontlasting van de microprocessor.
Elke TTL uitgang heeft een "fan out" van 10; d.w.z. er kunnen 10 TTL ingangen op worden aangesloten.
Op het schema kunt U dan ook zien dat van $\overline{A12}$ - $\overline{A11}$ en $\overline{A10}$ gretig gebruik gemaakt wordt.

Consequentie van deze invertering is wel, dat de uitgangs rangorde van U22 op zijn kop staat. Een A12 die "hoog" is, komt als "laag" bij U22 aan, hierdoor is YØ de hoogste en Y7 de laagste uitgang geworden.

Voorbeeld II BASIC en ACIA

Op de Basic chips zijn AØ t/m A10 aangesloten en tussen het adres AXXX van U23 en een Basic chip missen we dus alleen maar de lijnen A12 en A11.
Hier kan een 74139 (2 naar 4 decoder) volstaan om een compleet adres te verkrijgen. (Fig 6).

Enable signaal voor deze eerste helft van U17 is uitgang Y5 van U23 n.l. AXXX.
Onderweg naar U17 wordt aan AXXX nog een signaal toegevoegd n.l. $R/\overline{W}$.
Dit heeft practisch hetzelfde effect als de $\overline{\emptyset 2}$ aansluiting uit het vorige voorbeeld. Daar echter uit ROM alleen maar gelezen kan worden is aansluiting van $R/\overline{W}$ zinvoller.
Ook in dit geval is door de geinverteerde adreslijnen de uitgangsvolgorde weer omgekeerd.

De Basic chips hebben een "hoge" Chip Select en dus moeten de uitgangssignalen van de 139 worden geinverteerd.

De andere helft van de 139 heeft slechts een nuttige uitgang n.l. het signaal ACS (Fig 7).
Ook nu worden weer zoveel mogelijk adreslijnen samengevoegd Als enable van deze zijde van de 139 fungeert een signaal dat ontstaat uit Y7 van U23 ($\overline{EXXX}$) waaraan toegevoegd wordt $\overline{A12}$ waardoor FXXX ontstaat in de NOR poort U15. Dit signaal is "hoog" en moet dan ook geinverteerd worden in U16.
OK deel II van U17 is ge-enabled.

Geselekteerd wordt met A9 en A8, als deze beiden nul zijn, ontstaat signaal $\overline{ACS}$
Dit is het basis Chip Select signaal voor de ACIA en wordt aan pin 9 van de ACIA aangesloten.
Op de pinnen 8 en 10 worden ook nog resp. $\overline{A11}$ en $\overline{A10}$ aangesloten.

Nu zetten we even alles op een rijtje om te zien hoe en in hoeverre dit adres gedecodeerd is:

A15 - A14 - A13 werden op U23 gebundeld. Als ze alledrie"een"zijn, gaat uitgang Y7 laag (EXXX)

$\overline{A12}$ wordt in de NOR poort erbijgetrokken. De Nor poort is een nullenverzamelaar d.w.z. alleen als A12 een één is, kan de NOR poort A12 en EXXX samen voegen.

$\overline{A11}$ - $\overline{A10}$ zijn aan"hoge" enables aangesloten, d.w.z. alleen als A11 en A10 nul zijn kunnen $\overline{A11}$ en $\overline{A10}$ de ACIA chip enabelen.

A9 - A8 selecteren als ze beiden nul zijn de uitgang YØ van U17.

A15	A14	A13	A12	A11	A10	A9	A8	A7	t/m	AØ
I	I	I	I	Ø	Ø	Ø	Ø	X X X X X X X X		X
F				Ø				X		X

Alleen de "high byte" van het adres is gedecodeerd. Van de "low byte" adreslijnen wordt alleen AØ aangesloten op pin 11 die als RS wordt aangeduid, n.l. Register Select. Dit dient om de ACIA in de juiste vorm te initieeren

VOORBEELD III VIDEO EN KEYBOARD

Van de acht uitgangen van chip U20 worden er maar 4 gebruikt, zie Fig 8.
Twee stuks zijn Video signalen, n.l. RVE en WVE, zij bepalen of uit het videogeheugen gelezen, danwel of erin geschreven kan worden.
Meer wil ik over het videogebeuren niet zeggen, omdat ik me anders op glad ijs zou wagen, en dat is onverstandig, zoals U weet.

De andere twee uitgangen bepalen de Keyboardadressering.
De signalen zijn: RKB en WKB.

De manier waarop nú de 138 gebruikt wordt is vernuftig gevonden:
De enables
Aan pin 6 hangt de Ø2, dat klopt, want pin 6 heeft een hoog signaal nodig
Aan pin 4 is uitgang Y6 van U23 verbonden, die het bereik CXXX selecteert
Aan pin 5 hangt $\overline{A12}$ die duidelijk daardoor het selectie gebied tot DXXX bepaalt.

De selectiesignalen zijn resp. $\overline{R/W}$ - $\overline{A11}$ en $\overline{A10}$ waarmee dan voor de video weer een volledig adres wordt gedecodeerd.

Het Keyboard adres wordt eenvoudigweg niet verder uit gedecodeerd.

De waarheidstabel laat zien dat de adressen die door $\overline{A11}$ en $\overline{A10}$ worden geselecteerd, tweemaal in de uitgangen voorkomen echter, eenmaal als lees-adressering en eenmaal als schrijf-adressering.

7 4 1 3 8	INPUTS						OUTPUTS							
	ENABLE			SELECT			Y	Y	Y	Y	Y	Y	Y	Y
	G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	0	1	2	3	4	5	6	7
							READ				WRITE			
	Ø	$\overline{C}$	$\overline{A}$	R	A	A	D			D	D			D
	2	X	12	W	11	10	F			0	F			0
		X					X			X	X			X
							X			X	X			X
	X	1	1	X	X	X	1	1	1	1	1	1	1	1
	0	X	X	X	X	X	1	1	1	1	1	1	1	1
	1	0	0	0	0	0	0	1	1	1	1	1	1	1
	1	0	0	0	0	1	1	0	1	1	1	1	1	1
	1	0	0	0	1	0	1	1	0	1	1	1	1	1
	1	0	0	0	1	1	1	1	1	0	1	1	1	1
	1	0	0	1	0	0	1	1	1	1	0	1	1	1
	1	0	0	1	0	1	1	1	1	1	1	0	1	1
	1	0	0	1	1	0	1	1	1	1	1	1	0	1
	1	0	0	1	1	1	1	1	1	1	1	1	1	0
PINOUT	6	4	5	3	2	1	15	14	13	12	11	10	9	7

16 = VCC 8 = GND X = ONBEPAALD

FIG **8**

Rein Heesterman

> Het MOSI systeem

Hierbij willen wij u het MOSI systeem presenteren.
U zult zich misschien afvragen wat is MOSI ?
Wel MOSI staat voor OSI Middennederland.
Dit is bedacht door Martien Schot en Ko v.Ekeren ,die als doel hadden het Superboard in verbeterde modulaire vorm te herbouwen
Gekozen is uiteraard voor het elektuurbussysteem omdat hier al kaarten voor bestonden.
Hieronder volgt een opsomming van wat er reeds ontwikkeld is, waarvan bijgaand ook de schema's met printopstelling.

1) universele geheugenkaart

   Deze kaart is geschikt voor 2 t/m 16K statische ram (2Kx8) b.v. 6116 8416 5517 M58725 enz.enz.
   of 2 t/m 32K eprom 2716 en/of 2732.

2) video kaart

   Deze kaart geeft een zeer stabiel (50 hertz) beeld van:
   64 characters x 32 regels

3) processor-interface kaart

   Deze kaart bevat naast de 6502 een parallelpoort voor printer RS-232 en een cassette interface met een uitgebreide baudrate (75 t/m 9600 baud)
   en een nauwkeurige adresselectie voor keyboard en ACIA
   Doordat tijdens keyboard of ACIA selectie het bus r/w signaal in de write mode gehouden word, is het mogelijk om op de keyboard en ACIA adressen eproms te plaatsen

4) keyboard decoder kaartje

   Dit is een klein printje dat tussen keyboard en databus gemonteerd word, en is voorzien van electronische shiftlock en autoreset

5) bus kaart

   Het is mogelijk 8 kaarten op deze buskaart te steken

6) voedings kaart

   Het is mogelijk twee voedingen op deze kaart te monteren b.v. 5 en 12 volt welke volledig gescheiden zijn en overspannings beveiligd

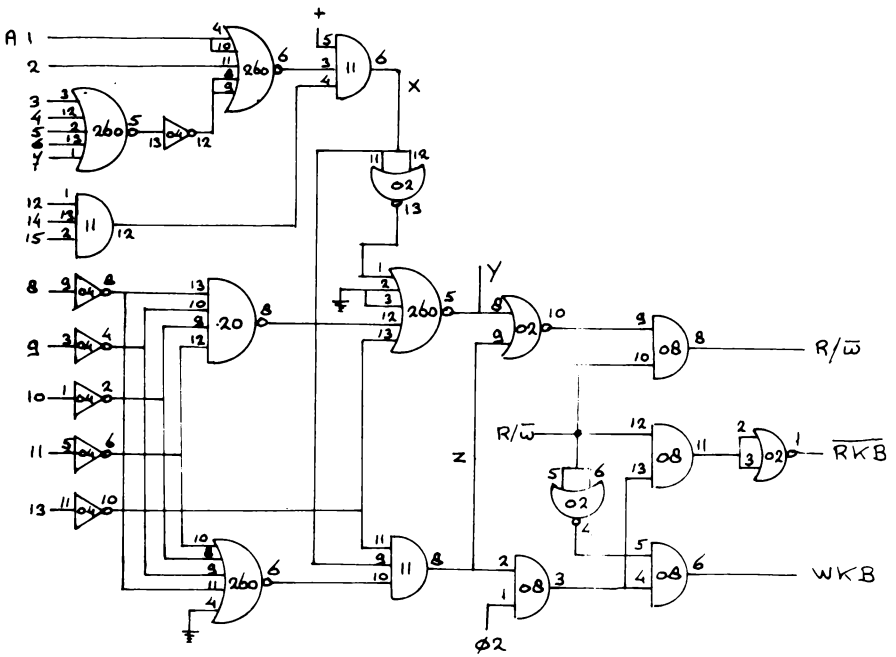
7) extender kaart

   Deze kaart maakt het mogelijk de kaarten buiten het frame te plaatsen zodat aan de kaarten gemeten kan worden.

8) keyboardprint

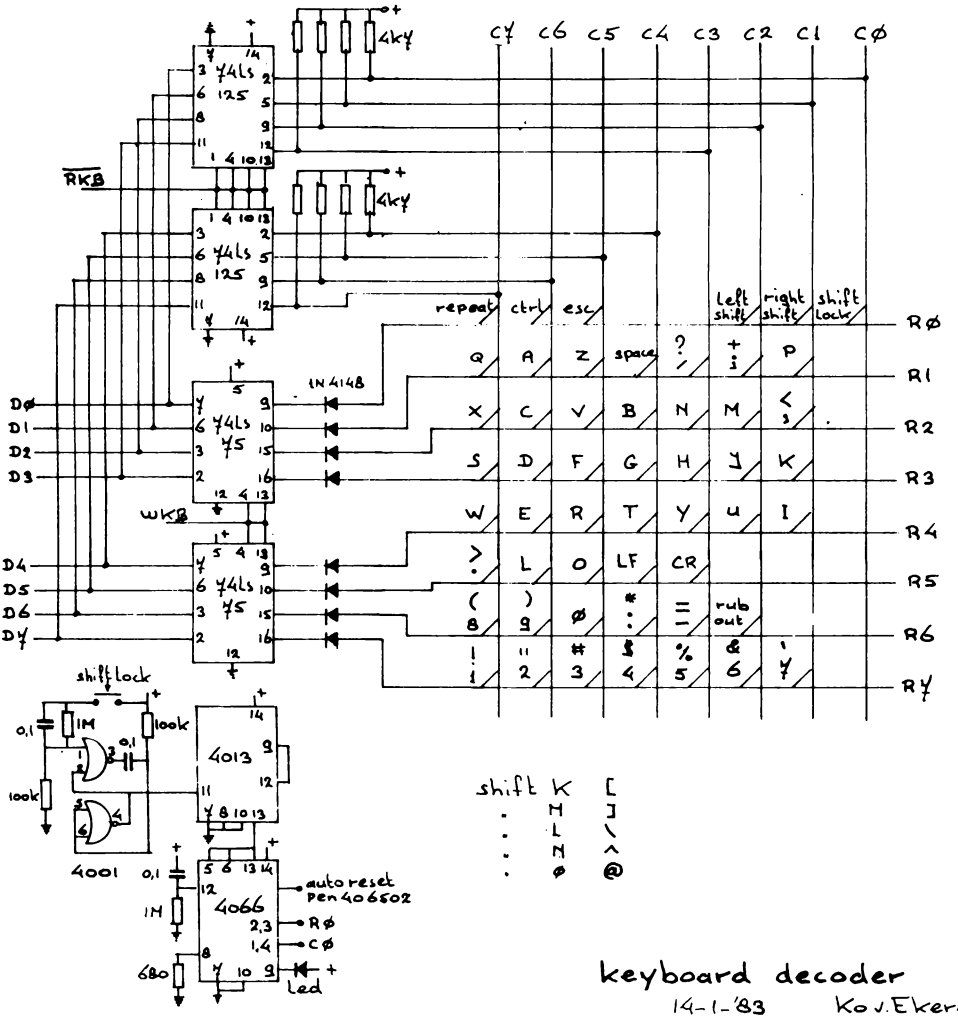
   Op deze print kunnen Tokay MM10L2X keyboard schakelaars gemonteerd worden, deze zijn leverbaar door o.a. Modelec in Ede

FEBRUARI 1983



acia en keyboard adressering
acia FØØØ - FØØ1
keyboard DFØØ - DFØ1

FEBRUARI 1983



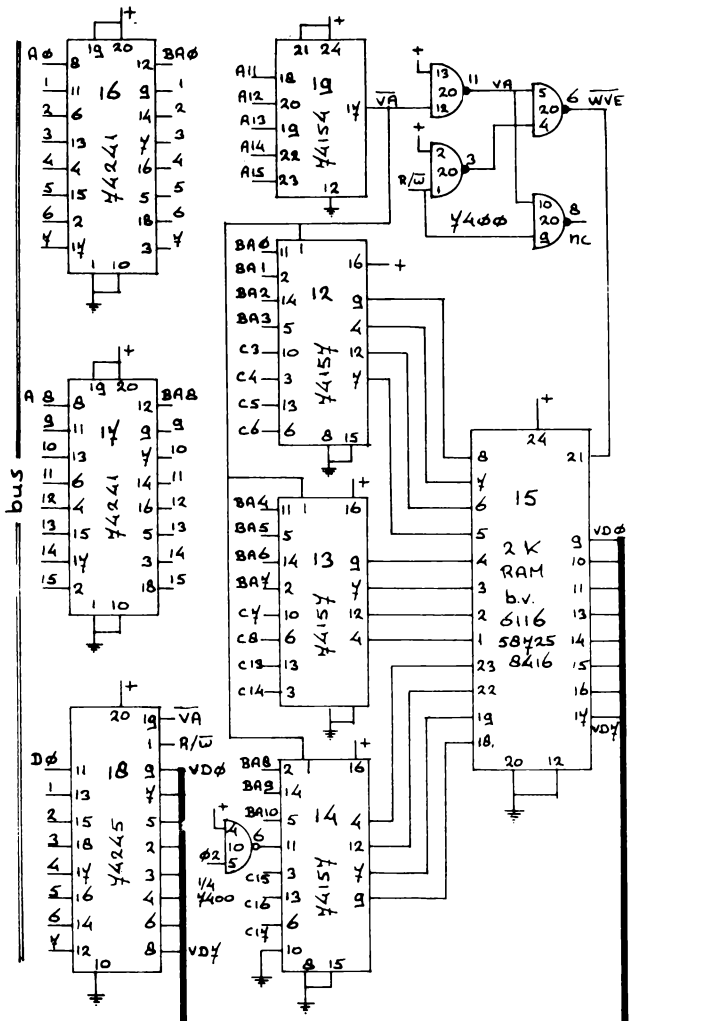
keyboard decoder

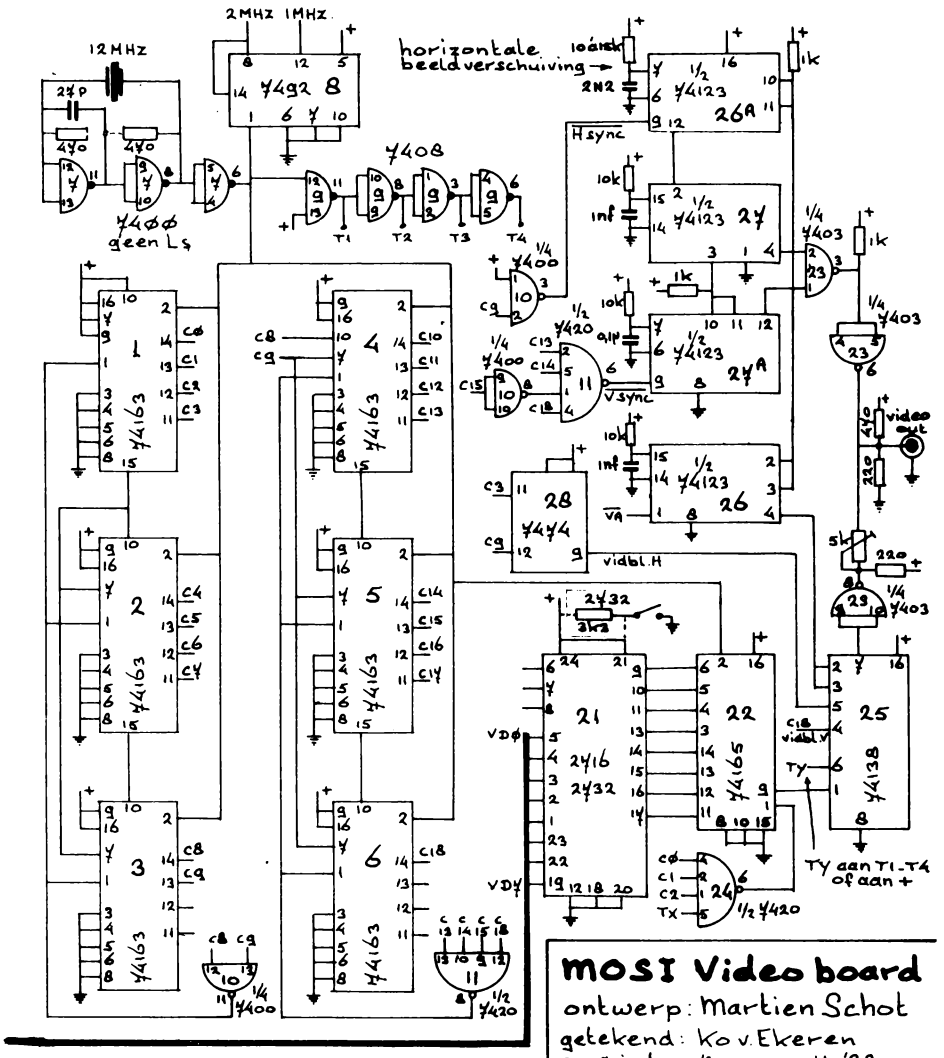
14-1-'83 Ko v. Ekeren

voorstel aansluiting toetsenbord aan systeem d.m.v. 15 polige D connector

1 WKB	9 $\overline{\text{RKB}}$
2 reset	10 massa
3 + 5V	11 D6
4 D7	12 D4
5 D5	13 D2
6 D3	14 DØ
7 D1	15 NC
8 NC	

FEBRUARI 1983


bus
74241
74154
74157
74245
74LS00
2K RAM b.v. 6116 5847258416
VA
R/W
VDØ
VDY
WVE
nc

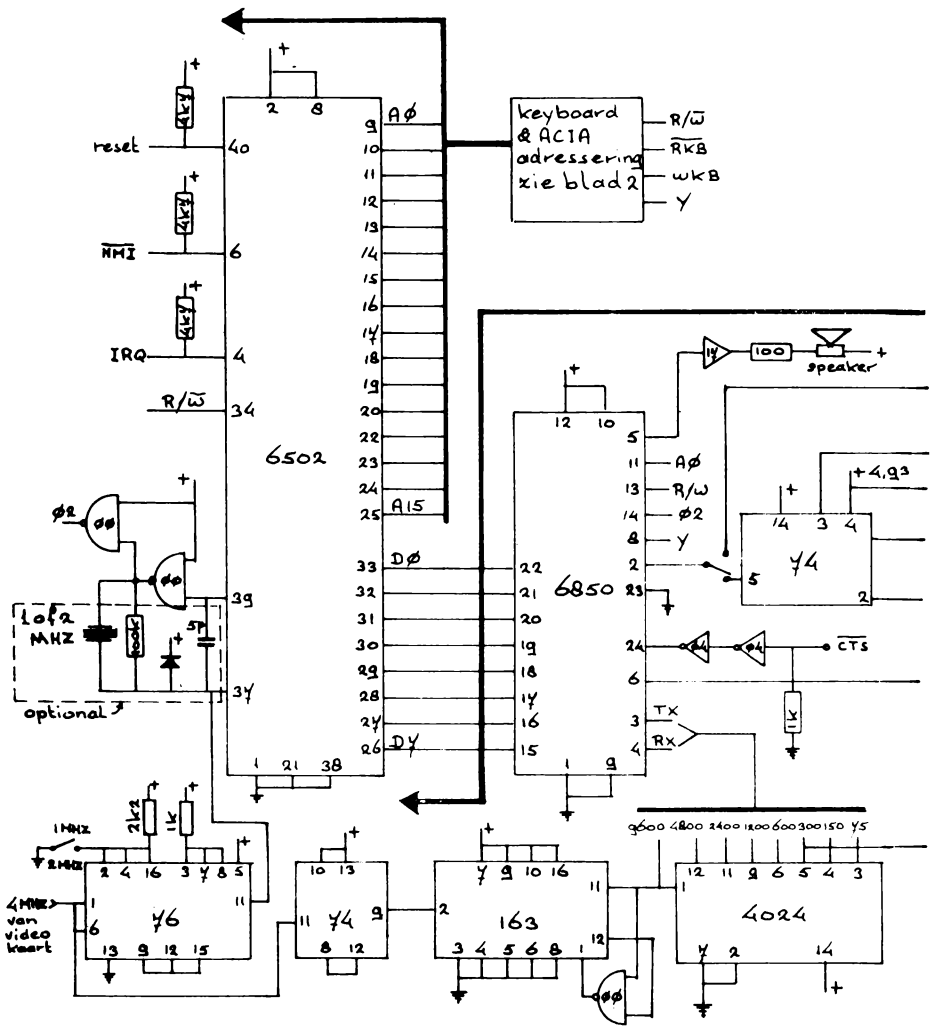


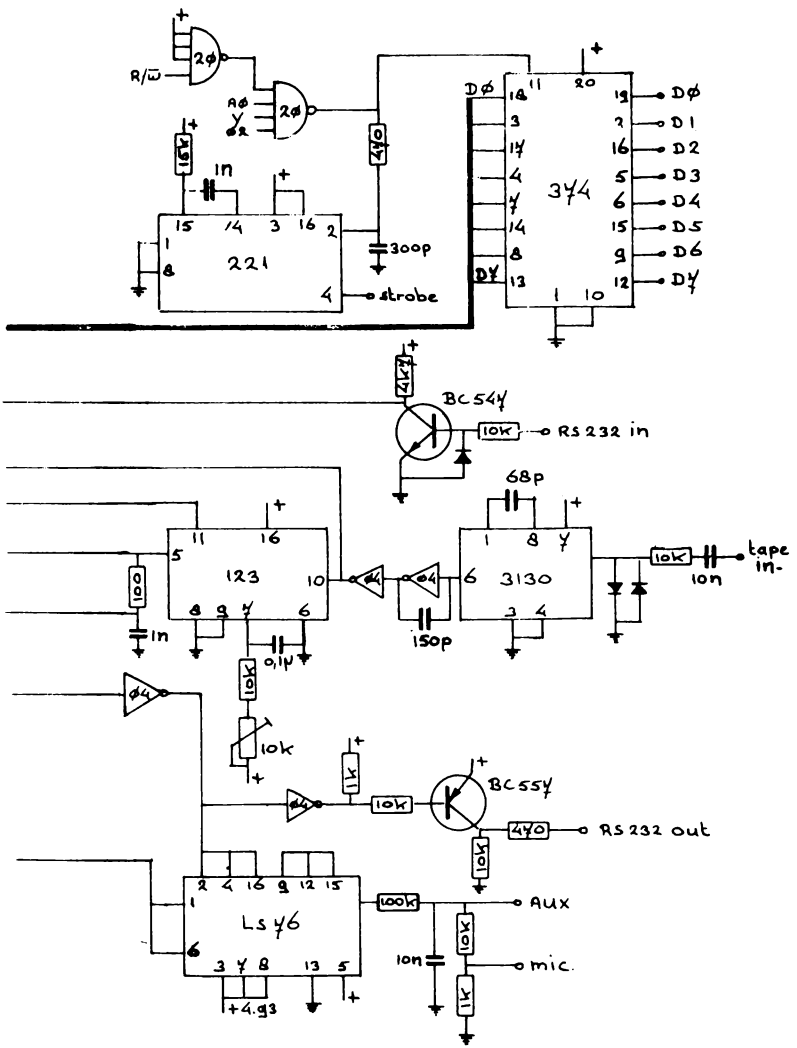
MOSI Video board

ontwerp: Martien Schot

getekend: Ko v. Ekeren

gewijzigd 12-1-'83 11-'82


keyboard & ACIA adressering zie blad 2
R/W
RKB
wKB
Y
reset
NMI
IRQ
R/W
6502
A0
A15
D0
D7
6850
speaker
100
+4,93
74
CTS
TX
RX
1,8 MHz
optional
2MHz
4MHz van video kaart
76
74
163
4024
9600 4800 2400 1200 600 300 150
Y5

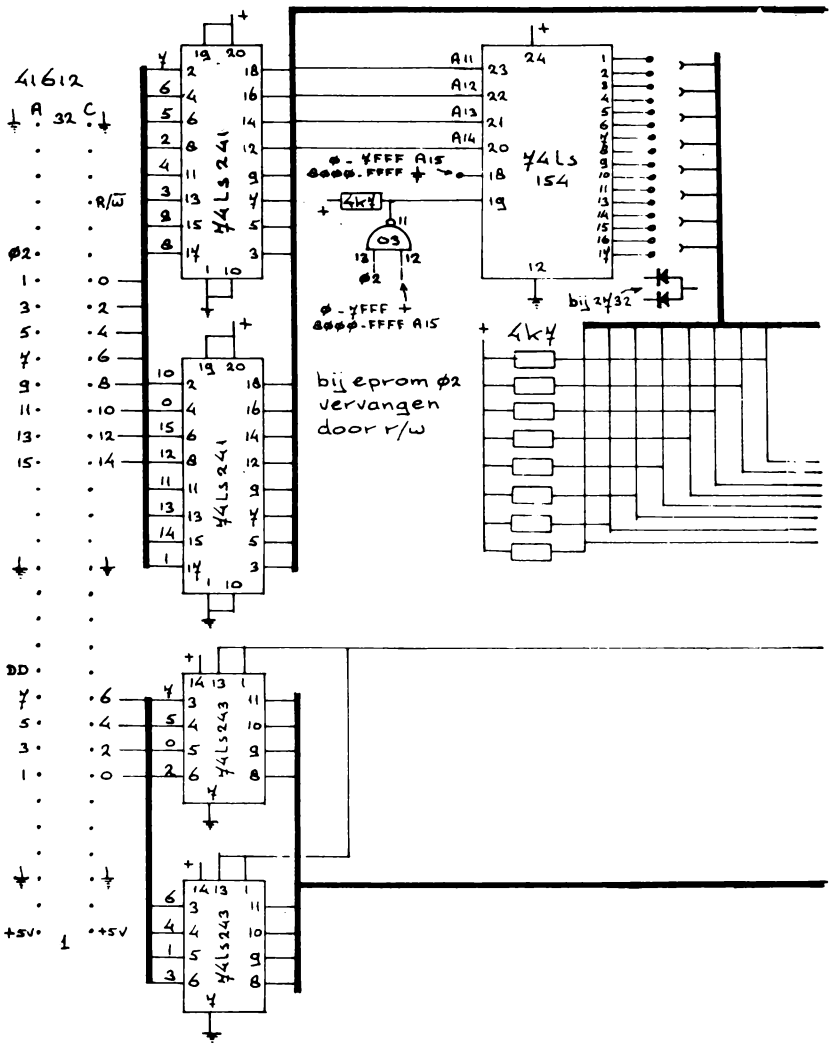


mosi

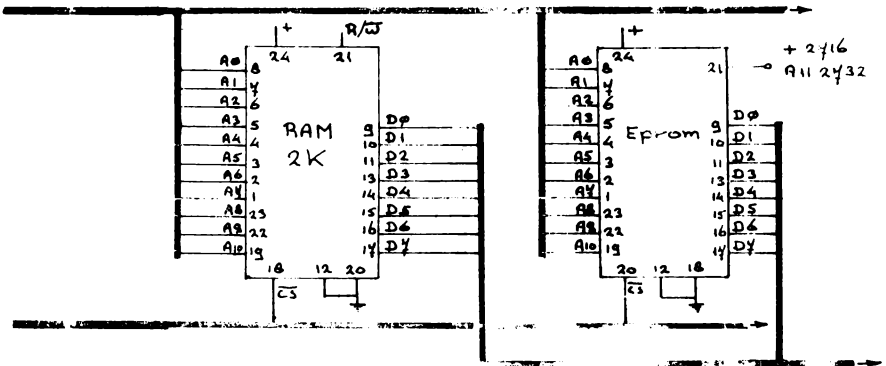
cpu en interface board

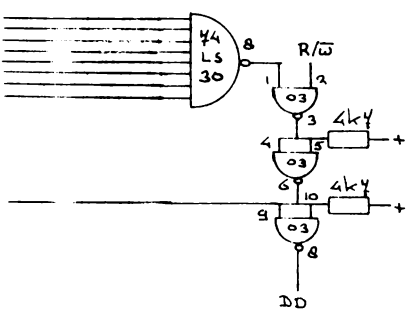
FEBRUARI 1983

Ko v. Ekeren 1.'83



FEBRUARI 1983





indien DD niet nodig is
kan 74ls03 vervangen worden
door 74ls00 en kunnen
pull up weerstanden vervallen

PINNR.	74154					
1	0000	-	07FF	8000	-	87FF
2	0800	-	0FFF	8800	-	8FFF
3	1000	-	17FF	9000	-	97FF
4	1800	-	1FFF	9800	-	9FFF
5	2000	-	27FF	A000	-	A7FF
6	2800	-	2FFF	A800	-	AFFF
7	3000	-	37FF	B000	-	B7FF
8	3800	-	3FFF	B800	-	BFFF
9	4000	-	47FF	C000	-	C7FF
10	4800	-	4FFF	C800	-	CFFF
11	5000	-	57FF	D000	-	D7FF
13	5800	-	5FFF	D800	-	DFFF
14	6000	-	67FF	E000	-	E7FF
15	6800	-	6FFF	E800	-	EFFF
16	7000	-	77FF	F000	-	F7FF
17	7800	-	7FFF	F800	-	FFFF

mosi memory board

16 K static ram
of 16 - 32 K eprom

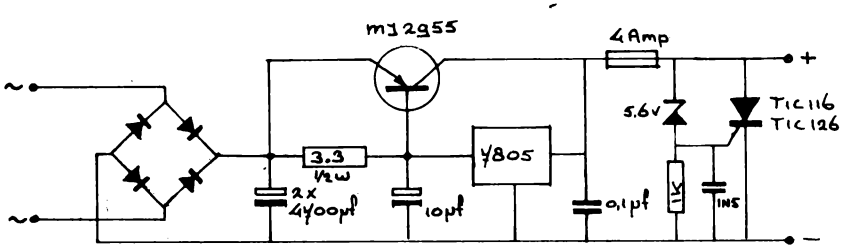
FEBRUARI 1983

Al deze printen met documentatie zijn te verkrijgen bij:

Ko van Ekeren Reijerscop 26 3454 HX De Meern
of
Martien Schot Hofstede 21 3902 CG Veenendaal

Eventueel zijn ook character eproms bij ons te verkrijgen
2716 eprom met OSI characters of
2732 eprom met OSI + viditel characters
Ook gewijzigde monitor eproms (voor 64 char.) kunnen wij verzorgen

Om het systeem nog completer te maken zouden wij mensen die ook reeds (euro)kaarten of ideeen hebben ontwikkelt willen verzoeken contact met ons op te nemen. (liefst schriftelijk)



Mosi dual power board

FEBRUARI 1983

EPROMS in het gebruik.
(Erasable Programmable Read Only Memory)

Het is met de Eproms zo gegaan als met zoveel van de huidige microcomputer-elektronica, als een bepaalde chip meer wordt gekocht door de hobbyist, dan stijgt daarvan het gebruik, doordat de produktie wordt verhoogd, gaat de prijs omlaag, de hobbyist koopt nog meer van die produkten en op het laatst krijg je, dat een produkt dat amper niet te betalen viel, nu plotseling onder ieders bereik komt.
De Eproms zie je overal om je heen gebruiken en een 2716 is op het ogenblik wel het populairste artikel en is ook zeer betaalbaar. Er zijn nog vrij grote verschillen in prijs bij de diverse handelaren maar na een korte tijd is te verwachten dat ook dit zal worden gladgestreken, totdat er een verzadigingspunt ontstaat, of een ander artikel in de belangstelling komt.
Natuurlijk sleept het gebruik van een bepaald artikel ook andere zaken met zich mee. Zo is bijvoorbeeld de laatste jaren met het stijgen van de populariteit van de Eprom ook de Eprom-programmer toenemend in de belangstelling gekomen.
Het bekende programmertje van Zero SC is op dit gebied erg bekend geworden.
Voor een lage prijs wordt een geraffineerde Eprom-programmer geleverd, die zonder enige fout of moeilijkheden is te gebruiken door wie dan ook, een erg handig stukje gereedschap.
De Eprom staat dan ook op die manier voor de computerist ter beschikking om zijn veel gebruikte, eventueel zelf gemaakte, programma in op te slaan en dit elk ogenblik te kunnen oproepen zonder omslachtige cassette laad-manoeuvres.
Daar niet alle computers de mogelijkheid bieden om Eproms bij te plaatsen op het bord, worden er door diverse firma's onder andere door bijvoorbeeld Elektuur, printjes in de handel gebracht waarop Eproms kunnen worden ondergebracht.
Al deze feiten werken mee om het gebruik van de Eproms te stimuleren.
In de praktijk blijkt dat ondanks de populairiteit van de Eprom, er wel mensen zijn die op de een of andere manier moeilijkheden hebben bij het bewerken of het lezen van een Eprom. Kort en goed, het is nodig gebleken om hier iets aan te doen. Een ander punt is ook nog dat als een Eprom eventueel geprogrammeerd is en er blijkt een fout in te staan, de Eprom kan worden gewist, maar dat het wissen van de Eprom een moeilijkheid vormt, doordat daar betrekkelijk dure apparatuur voor nodig is en men over het algemeen niet weet dat dit met een erg goedkoop lampje van Philips ook kan.
Op het wissen van de Eprom komen we in een ander hoofdstuk terug en doen we dat volledig uit de doeken.
Elke zich respecterende chip-fabrikant maakt vandaag de dag een aantal Eproms van verschillende capaciteit.
De grotere fabrikanten vanaf het 1 K byte type tot en met het 16 K byte type en er zijn anderen, die net voorzichtig meesnoepen van de populaire typen, namelijk de 2716 en de 2732. Er zijn dus vele merken Eproms in de handel op het ogenblik waar volgens ervaring van mensen die daar veel mee omgaan, toch nog wat eigenaardigheden aan kleven.
We zullen aan het eind van het artikel trachten een zo volledig mogelijk beeld te geven voor zover ik dat tenminste heb kunnen achterhalen, van de karakteristieke moeilijkheden van de verschillende chips.

Het Eprom principe
Van een 2716 zegt men dat hij een matrix heeft van 2048 x B, dat betekent dan dat er zich op die Eprom 2048 x 8 celletjes of te wel 16384 cellen bevinden die allemaal individueel kunnen worden geprogrammeerd.
Om u voor te stellen hoe de werking van een Eprom eigenlijk is kunt u het beste denken aan een sluis met sluisdeuren. Als de sluisdeuren open staan zal door de sluis water vloeien, er is dus een stroom. Zijn de sluisdeuren dicht, dan is er geen stroom van water door de sluis. Elke cel van de Eprom werkt ongeveer op dit principe. Zolang de cel openstaat zal er een stroompje doorvloeien en dus geeft die

cel zich zelf te kennen als zijnde een '1'.
Door het programmeren worden de sluisdeuren als het ware dicht gepompt, de drempel wordt zo hoog gemaakt dat er geen stroom meer kan vloeien en de cel geeft zich te kennen als een '0'.
Dat wil zeggen dat als u een niet geprogrammeerde Eprom bekijkt met een disassembler of iets dergelijks, u dan in die Eprom allemaal 'FFFF' zult aantreffen, in elke cel zal een '1' staan.
Een volledige geprogrammeerde Eprom waar elke cel is geprogrammeerd, zou dus kunnen neerkomen op een Eprom met alleen maar '0' erin. Door nu de ene cel wel en de andere cel niet te programmeren komt in de Eprom een patroon te staan van enen en nullen en u weet, dat is voor de computer het eten dat hij graag voor zich ziet.
Deze celletjes staan natuurlijk niet zomaar ordeloos op die chip, nee ze zijn volgens een matrix er op neergezet, dat wil zeggen we kunnen kolommen en rijen onderscheiden.
Die kolommen en rijen kunnen weer apart worden aangesproken door de adres-codering die zich op de chip bevindt.
Dat door zo'n cel niet een enorm vermogen kan worden verwerkt is waarschijnlijk begrijpelijk, de stroompjes die door de cel gaan zijn uiterst klein en moeten eerst worden versterkt voordat ze naar buiten toe kunnen worden uitgevoerd als DATA.
Vandaar dat op de chip ook buffers, dat zijn dan eindversterkers, aanwezig zijn.

Verschillende fabrikaten

Steeds meer fabrikanten van chips willen deelnemen aan de markt die voor deze chips is ontstaan. Daarbij zijn uiteraard ontzettend veel Japanse fabrieken die hoe langer hoe meer de kunst hebben afgekeken van de Amerikanen en nu zoals we dat gewend zijn, door een grotere perfectie de markt van de Amerikanen proberen weg te snoepen.
Verschillende grote Amerikaanse fabrieken zijn uiteraard nog steeds aan de markt en ze brengen hun produkten met succes. Er zijn natuurlijk altijd wel enkele spitsafbijters geweest, fabrieken die voor het eerst uitkwamen met zo'n chip.
Ook de chip-fabricagemethoden ondergaan steeds verbeteringen en we kunnen eigenlijk wel zeggen dat een generatie van chips nu is voorbij gegaan en een nieuwe generatie al reeds door de verschillende fabrikanten aan de markt is gebracht.
De oude generatie van chips was die, die drie spanningen nodig had om behoorlijk te kunnen werken. Dat was niet alleen het geval met de dynamische RAM-chips maar ook met de Eprom chips. Er was daar sprake van de spanningen +12 Volt, +5 Volt en —5 Volt.
Een voorbeeld van deze chips is de nog wel bekende 2708 die bij alle fabrikanten deze spanningen nodig heeft. Dan blijkt weer dat zo'n spitsafbijter zoals bijvoorbeeld Texas Instruments zich dan toch eigenlijk naar mijn bescheiden mening, heeft vergaloppeerd, want deze fabrikant heeft ook zijn 2716 gebracht met die drie spanningen. Zodoende is deze chip van Texas Instruments een soort van buitenbeentje geworden in de huidige single supply golf.
Trend-setter van de huidige Eproms is wel de firma INTEL. Deze firma heeft de eerste grote aanstoot gegeven aan de single supply of wel +5 Volt gevoede chips en praktisch alle andere fabrieken die hierna zijn uitgekomen met Eproms, hebben dit voorbeeld gevolgd.
Natuurlijk kennen al die fabrikanten weer eigen benamingen of kenmerken voor de chips en u zult in de tabel er ook wel enige vinden. Dat zijn de merken die op de chip staan gedrukt. Echter, het komt allemaal neer op de zelfde soort chip en de verschillen tussen deze huidige chips zijn toch wel miniem.

De chip in het systeem

Zoals ik reeds eerder opmerkte wordt op de chip een bepaalde cel geadresseerd door middel van decoderingsschakelingen die zijn ingebouwd.
Als u wilt weten welke adreslijnen nodig zijn om een chip met een matrix van 2048x8 oftewel 16384 bits te adresseren, dan vraagt u maar aan uw calculatortje om voor u uit te rekenen 2 tot de macht 11 en na enig aarzelen zal op het display dan ook 16384 verschijnen. U weet dus nu dat 11 adreslijnen namelijk de adreslijnen A0 tot en met A10, nodig zijn, om deze chip intern te adresseren.

Evenzo is dan voor een 2732 met een matrix van 4096x8 een adreslijn meer erbij nodig want elke adreslijn méér verdubbelt namelijk het aantal te adresseren bits.
Uit de overige hogere adreslijnen die uw computer rijk is, wordt dan een chipselect signaal gevormd dat op de pen die met CE of CS gemerkt is, wordt aangesloten. Dit zijn meestal laag actieve signalen zodat er een streep of bar boven staat en we spreken dan ook van bar-CE of bar-CS.
Evenzo is er op een andere pen meestentijds een signaal OE-bar aanwezig. De functie van deze signalen is tussen de chips onderling nog wel eens verschillend.
U vindt bij dit artikel twee tabellen per Epromsoort: een aparte tabel voor ten eerste de chips die aan elkaar gelijk zijn en ten tweede voor de daarvan afwijkende chips. Natuurlijk heeft u met al dit soort schakelingen niets te maken als u gebruik maakt van een Eprom-program apparaatje omdat daarop al die schakelingen doorgaans op een eenvoudige wijze zijn aan te brengen, maar het is voor de echte knutselaar toch van belang om te weten wat de verschillende signalen inhouden en hoe de chip er op reageert.
Buiten de adreslijnen en de controlelijnen zijn er natuurlijk ook nog datalijnen op de chip aanwezig en die kunnen zich in drie staten bevinden namelijk: De staat 'data-in', dat is dan bij het programmeren; de tweede-staat 'data-out', dat is bij het lezen, het normale gebruik van de Eprom. De derde is een staat waarin de chip zich praktisch scheidt van de bus. Dat is een staat van hoge impedantie en u vindt dat op vele datasheets aangegeven met de bemerking 'HIGH-Z'.
In deze laatste stand is de chip praktisch vrij van de bus en kan geen storingen ontvangen uit de bus of meegeven aan deze bus.

Wissen

Voordat een chip geprogrammeerd kan worden moet hij òf leeg van de fabrikant betrokken zijn òf anders, als wij een programma in de chip hebben staan, zal dit eerst uit de chip gewist moeten worden. Dat wissen kan gebeuren met een ultra-violet lamp die licht produceert met een golflengte van 2537 ingstrom. Als een chip onder zo'n lamp wordt gelegd en daarvan niet verder dan 2½ cm is verwijderd, duurt het wissen ongeveer 15 minuten. Deze tijdsduur zal ook ervan afhangen of het venstertje van de chip helemaal schoon is. Om dit schoon te maken kan alcohol gebruikt worden of een ander reinigingsmiddel dat de chip niet aantast. De lamp zal uiteraard van het juiste type moeten zijn. Er wordt door Philips een lamp geproduceerd die in ziekenhuizen wordt gebruikt voor sterilisatie van medische intrumenten. Dit sterilisatie-buisje is ongeveer 20 tot 25 cm lang, heeft een gewone, normale gloeilamp fitting en kan dan ook in een normale gloeilampfitting worden gedraaid en zo direct met de 220 Volt van het lichtnet werken. Er is geen voorschakel-weerstand nodig, het lamoje werkt helemaal zelfstandig. Het betreft een Philips lampje met de betekenis TUV-6W (een 6 Watt lampje) en het is niet het lampje dat het zogenaamde zwarte licht uitstraalt, dat lampje geeft niet de golflengte die nodig is. Uiterste voorzichtigheid moet worden geboden bij het gebruik van deze lamp, het zijn niet de zichtbare stralen die het oog kunnen beschadigen, maar het intensieve ultraviolette licht dat door het lampje wordt uitgestraald, kan zeer schadelijk zijn voor het netvlies van uw ogen. Wees dus uiterst voorzichtig en sluit de lamp **geheel** af. De lamp wordt niet warm, een koortsthermometer die op een centimeter afstand van de lamp gedurende 10 minuten werd neergelegd wees geen termperatuur aan. Met andere woorden u hoeft niet bang te zijn dat door warmte-ontwikkeling iets kan gebeuren met uw Eproms. Door de kleine omzet van deze lampen is de verkrijgbaarheid nogal gering, in feite zou elke elektrotechnische zaak, of installatiebedrijf dit lampje moeten kunnen leveren. Ze worden geleverd in een vrij groot aantal per verpakking en het zal wel duidelijk zijn dat men niet bereid is om bij het leveren van 1 of 2 lampjes gelijk een hele doos aan te schaffen.

Verschillende werktoestanden

De verschillende werktoestanden heten op zijn Engels MODE. We kennen in de eerste plaats de READ-MODE (lees-wijze) waarbij 8 bits worden geadresseerd (1 byte). De inhoud van deze byte wordt op de bus gezet door de databuffers van de chip en de computer (microprocessor) kan deze data lezen. Na dit lezen wordt de chip gedeselecteerd maar vervalt hierbij in een soort halve waak toestand die wordt aangegeven met STAND-BY of POWER-DOWN.

Bij het inschakelen van de READ-MODE of STAND-BY mode spelen de toestanden van de pennen 18 en 20 een grote rol en u kunt ten eerste zien uit de tabel hoe zich dit verhoudt maar ook komen we er nog later op terug. Ten derde kennen we de PROGRAM-MODE dat is de toestand waarin op een van de pennen 21 of 25 Volt wordt aangesloten en op een andere pen het programma naar binnen wordt gepulst.
In de mode STAND-BY is de data-zijde van de chip gescheiden van de databus, dus in de zogenaamde HIGH-Z staat, toestand waarbij de verbindingen van chip en databus een dergelijk hoge weerstand hebben dat praktisch de verbinding verbroken is. Om het stroomverbruik van de Eproms zo laag mogelijk te houden wordt de STAND-BY mode gebruikt, in deze toestand wordt slechts een kwart van het normale verbruik gebruikt. U zult begrijpen dat als een chip die in een STAND-BY mode staat plotseling wordt ge-enabled (ingeschakeld) een vrij groot meerverbruik ontstaat dat tot stroompieken op de voedingslijnen aanleiding kan geven. Het is daarom aan te bevelen dat de voedingslijnen direct bij de Eproms gebufferd worden door een laagspannings Elco van 47 of 100 micro farad als er enkele Eproms bij elkaar zitten. De gang van zaken is nu als volgt: De chip wordt geselecteerd (ge-enabled) door het uitzenden van een adres. Als dat adres 'stabiel' is. wordt de output-enable omlaag gebracht en worden de data op de databus gezet door de chip. De READ-MODE is dus een bedrijf in 2 fasen. De Output-enable zal meestal met succes kunnen worden aangesloten op het geïnverteerde clock (02) signaal. Uit deze beschrijving zal duidelijk zijn dat deze beide pennen: CE en OE niet met elkaar verwisseld mogen worden omdat dan leesfouten kunnen ontstaan.

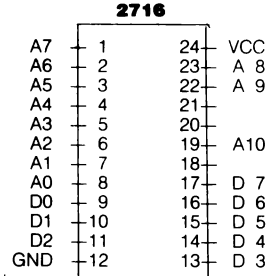
Verschillende categorieën Eproms

Met de afbeeldingen 1, 2 en 3 zijn de pin-outs gegeven van de diverse categorieën Eproms.
De in deze afbeelding niet benoemde pennen zijn in figuur 1 tot en met 3 apart vermeld met de door de diverse fabrikanten gebezigde benamingen. Tenslotte zijn in figuur 4 de verschillende werktoestanden weergegeven met daarbij van elke chip de logische signalen of Voltages die op de bewuste pennen aanwezig moeten zijn.

Chip pin out's

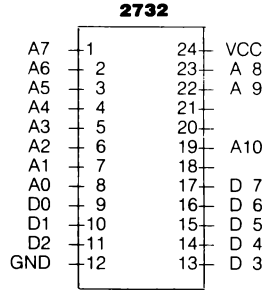
Afbeelding 1

2516
2716

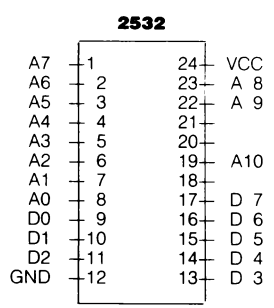


Afbeelding 2

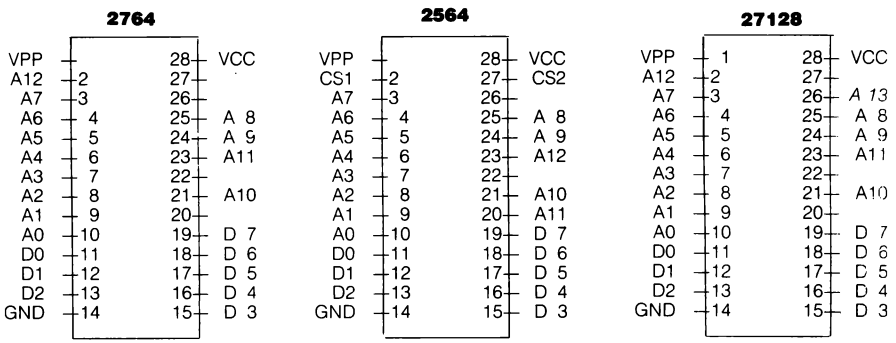
2732



2532



Afbeelding 3



	2716					2516	
Fabrikaat	Intel	Motorola	AMD	Fujitsu	Hitachi	Texas Instruments	
Chip kenmerk / Pennummer	2716	MCM 2716	AM 2716	MBM 2716	HN 462716G	TMS 2516	
21	VPP	VPP	VPP	VPP	VPP	VPP	
20	OE	G	OE	OE	OE	CS	
18	CE/PGM	E/PGR	CE/PGM	CE/PGM	CE/PGM	PD/PGM	

Figuur 1 (tabel + pin out)

	2732				2532		
Fabrikaat	Intel	AMD	Fujitsu	Hitachi	Texas Instruments	Motorola	Hitachi
Chip kenmerk / Pennummer	2732A	AM 2732	MBM 2732A	HN 462732G	TMS 2532	MCM 2532	HN 462532G
21	A11	A11	A11	A11	VPP	VPP	VPP
20	OE/VPP	OE/VPP	OE/VPP	OE/VPP	PD/PGM	E/PRG	CE/PGM
18	CE/PGM	CE/PGM	CE/PGM	CE/PGM	A11	A11	A11

Figuur 2 (tabel + pin out)

	2764				2564	27128	
Fabrikaat	Intel	AMD	Fujitsu	Hitachi	Texas Instruments	AMD	
Chip kenmerk / Pennummer	2764	AM 2764	MBM 2764	HN 482764G	TMS 2564	AM 27128	
27	PGM	PGM	PGM	PGM	CS2	PGM	
26	NC	NC	NC	NC	VCC	A13	
22	OE	OE	OE	OE	PD/PGM	OE	
20	CE	CE	CE	CE	A11	CE	

Figuur 3 (tabel + pin out)

Figuur 4

Mode	PEN	1	18	20	21	22	27
READ	2716-2516	—	laag	laag	+5V	—	—
	2732	—	laag	laag	—	—	—
	2532	—	—	laag	XX	—	—
	2764	+5V	—	laag	—	laag	hoog
	2564	+5V	—	—	laag	laag	laag
STANDBY	2716-2516	—	hoog	XX	+5V	—	—
	2732	—	hoog	XX	—	—	—
	2532	—	—	hoog	XX	—	—
	2764	+5V	—	hoog	—	XX	XX
	2564	+5V	—	—	XX	hoog	laag
PROGRAM	2716-2516	—	PULS	hoog	+25V	—	—
	2732	—	PULS	+25V	—	—	—
	2532	—	—	PULS	21/25V	—	—
	2764	VPP	—	laag	—	hoog	PULS
	2564	21/25V	—	—	laag	PULS	laag

Programmeerspanningen

De in de tabellen voorkomende aanduiding VPP staat voor programmeerspanning. De grootte of hoogte van deze spanning staat in directe relatie tot de celdichtheid van de eigenlijke chip. Zo is er voor de 2716 typen, die nog een relatief kleine dichtheid hebben, algemeen 25.0 V als VPP voorgeschreven. Vanaf de 2732A typen en vervolgens ook voor de 2764, 27128 en 27256 vindt u de VPP gegeven als 21.0 Volt.

Soms verdraagt een bepaald fabrikaat nog een mishandeling met 25.0 Volt, bijvoorbeeld Fujitsu, maar als u een Intel chip tegen de voorschriften in met 25.0 Volt wilt programmeren, blaast u hem gegarandeerd op. Neem daarom geen risico en houd u aan de voorschriften, het kan uw portemonnee alleen maar ten goede komen. Het antwoord op de vraag hoe u 21.0 Volt uit het Zero programmertje kunt halen, vindt u achteraan in dit artikel.

Speciale eigenaardigheden
2716/2516
Deze 2K categorie veroorzaakt wel de minste last. Alleen bij gebruik van de 2516 moet op de afwijkende aansluiting van pennen 18 en 20 worden gelet. De programmeerspanning VPP is 25.0 Volt. Er is van Texas Instruments ook een 2716 verkrijgbaar maar dit is een chip van de oudere, 3 supply-generatie, dat wil zeggen voor deze chip zijn nog +12V, +5V en −5V nodig.

2732/2532
Doordat ook deze chip nu voor een betaalbare prijs verkrijgbaar is en pin-compatible is met de 2716, is de populariteit ervan sterk gestegen. Uit figuur 2 kunt u zien dat adreslijn A11 een plaatsje opvraagt waardoor bij de 2732 pen 20 een dubbel-functie krijgt. Achter deze pen zit op de chip een voltage-sensor die de chip automatisch omschakelt naar de Program-mode als aan deze pen VPP wordt aangeboden.

Pas op met de 2732A
Voor alle 2732A typen geldt een programmeerspanning: VPP van 21.0 Volt.
De 2532 heeft geen aparte output-enable. Als de chip met pen 20 ge-enabled wordt verschijnen direct de data op de bus. Voor de 2532 geldt als VPP: 25.0V

2764-2564
Deze chip kan niet meer pin-compatible zijn met de voorgaande typen en heeft dan ook een 28 pens voet nodig. Met deze voet kunnen wel weer een hele reeks chips met nog grotere capaciteit worden bediend. De 27128 en 27256 zijn al aangekondingd in de vakpers, de 27128 is zelfs al verkrijgbaar. Bij de 2764 is pen 26 niet aangesloten en bij de 2564 is deze pen intern door verbonden met pen 28 (VCC). Het programmeer voltage voor de 2764 is VPP = 21.0 Volt en van de 2564 is VPP = 25.0 Volt.

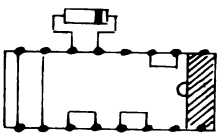
27128
Van deze chip was alleen wat 'Advanced Information' beschikbaar. De pin out staat in afbeelding 3. Het VPP voltage is VPP = 21.0 Volt.

Wissen
Niet elke Eprom wist even gemakkelijk. Bij hogere omgevingstemperaturen geeft een Eprom sneller aan 'leeg' te zijn, terwijl in een ruimte met lagere temparatuur toch blijkt dat in diezelfde Eprom nog wel wat informatie is 'blijven zitten'.
De harde UV-straling is de factor die de veroudering van een Eprom veroorzaakt, vele malen lang wissen zorgen dus voor het levenseinde van de Eprom. Er moet dan ook een compromis gesloten worden tussen zeker wissen en te lang wissen. Als u met één fabrikaat werkt gaat u natuurlijk uw pappenheimers wel kennen. Toch zijn er grote verschillen tussen de fabrikaten onderling. Een Fujitsu Eprom is al in ongeveer 10 minuten gewist maar het duurt bij een NEC Eprom wel 40 minuten voor alle informatie eruit is verdwenen. Ook Texas Instruments Eproms hebben een langere wis-tijd nodig.

Statische Lading
Een enkele keer schijnt het te zijn voorgekomen dat onder het quartz venstertje van de Eprom zich een statische lading bevond waardoor volkomen onregelmatige leesfouten optraden. Als remedie hiervoor werd gevonden, een hele korte wis-tijd van ongeveer 10 seconden. Hierdoor wordt de lucht onder het venster geïoniseerd en de lading kan afvloeien.

Zero Programmer aanpassingen voor VPP = 21.0 Volt

Er zijn verschillende uitvoeringen van dit populaire gereedschapje. Voor de moderne versies bestaat er een ombouwmethode die ook het programmeren van de 2764-27128 enzovoorts mogelijk maakt. Allereerst echter een gemakkelijke methode om het programmeren van de 2732A zonder grote kattesprongen mogelijk te maken. Hiertoe moet u een twee selectie voetje maken dat wordt bedraad zoals het originele voetje, met uitzondering van het draadbruggetje tussen pennen 4 en 5. U soldeert hier een zener diode van 3.9 Volt 400 mW tussen zó dat het zwarte ringetje (kathode) aan pen 4 ligt. Het tekeningetje van pagina 10 van de Handleiding Eprom programmer gaat er dan als volgt uitzien:



2732A

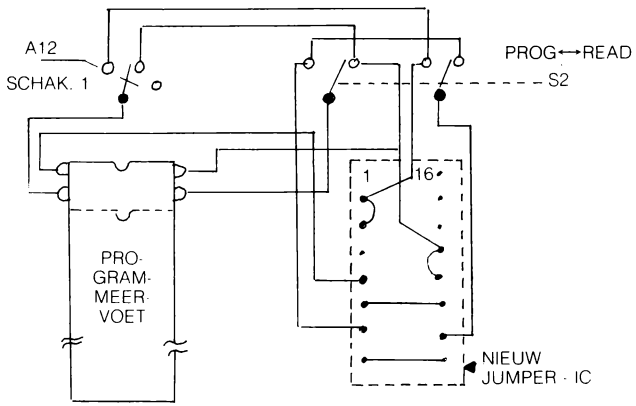
Dit voetje gebruikt u dan uitsluitend voor de 2732A. Een meer uitgebreide ombouw wordt in het volgende, door Zero SC gepubliceerde stukje beschreven.

Aanpassing van de ZERO EPROM programmer, zodanig dat er 2764 EPROM's mee geprogrammeerd kunnen worden

Benodigdheden:
1 28 pins socket
1 schakelaar dubbel om
1 schakelaar enkel om
1 zener 4v7 400mW
1 zener 5v7 400mW
1 spannings regelaar 78L15

Indien uw programmer reeds met een 78L15 is uitgerust, dan zijn de laatste 3 componenten niet nodig. U vindt naast het IC NE555 eerst 4 weerstanden en dan 3 diodes. De eerste diode vervangt u door een zener van 4v7 (streepje richting programmeersocket) en de tweede diode vervangt u door een zener van 5v6 (richting idem). Vervolgens vervangt u het component direct naast de schakelaar (78L24) door een 78L15. Het soldeer-eilandje aan de onderkant van de print, onder de eerste zener van 4v7 kan nu doorverbonden worden voor 2732A en 2764 EPROM's, of worden opgengelaten voor 2716, 2732 en 2532 EPROM's.

Verwijder nu het jumper IC en vervang dit door nevenstaande schakeling, inclusief nieuwe 28 pin programmeervoet. De A12 schakelaar is om de te programmeren helft in te stellen. Schakelaar S2 moet in de stand 'read' staan. Programmeren: Bij de melding 'switch VPP to 25 Volt', eerst S2 in de stand '25 Volt' zetten en dan pas de spatiebalk indrukken. Bij terugschakelen: Eerst 25 Volt afschakelen, daarna S2 omzetten en pas daarna spatie geven. Het is normaal dat de programmer in de 2732 mode staat.



Het was de bedoeling van dit artikel om het onderwerp zo uitputtend mogelijk te behandelen. De vele importeurs die mij aan data-sheets hebben geholpen, waarvoor ik hen zeer dankbaar ben, stelden mij in staat de begeleidende tabellen te maken. Ook de prettige medewerking die ik van Zero SC mocht ontvangen, waardoor vele waardevolle tips aan dit artikel konden worden toegevoegd, heb ik zeer op prijs gesteld.
Rein Heesterman

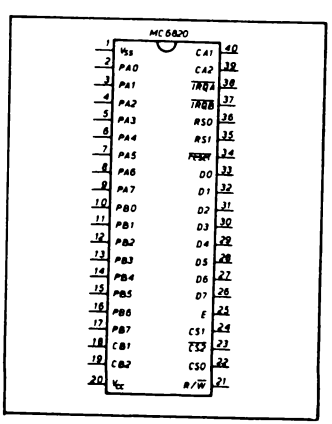
*** UITBREIDING M O S I SYSTEEM ***

Het MOSI systeem is uitgebreid met een PIA kaart.
Deze kaart kan 4 stuks PIAs herbergen van het type
6821 6820 of 6520.
Net als alle andere MOSI kaarten is deze ook weer volledig compatible met de elektuurbus en is volledig gebufferd door middel van TTL-LS met P.n.P inputs, welke er voor zorgen dat de data en adres lijnen minimaal worden belast.
OP de Print is nog wat ruimte gereserveerd om nog wat schakelingen op te bouwen.
De 80 !! in/outputs worden naar buiten gebracht d.m.v 2 stuks 40 polige DIL voeten,waarin 40 polige DIL konnektoren met bandkabel kunnen worden geplaatst.
De adressen van de registers van de PIAs vindt men op hex C110-C11F.

TOEPASSING: Machine besturing.
Modeltrein besturing.
Bewaking en beveiliging.
en natuurlijk jou toepassing.

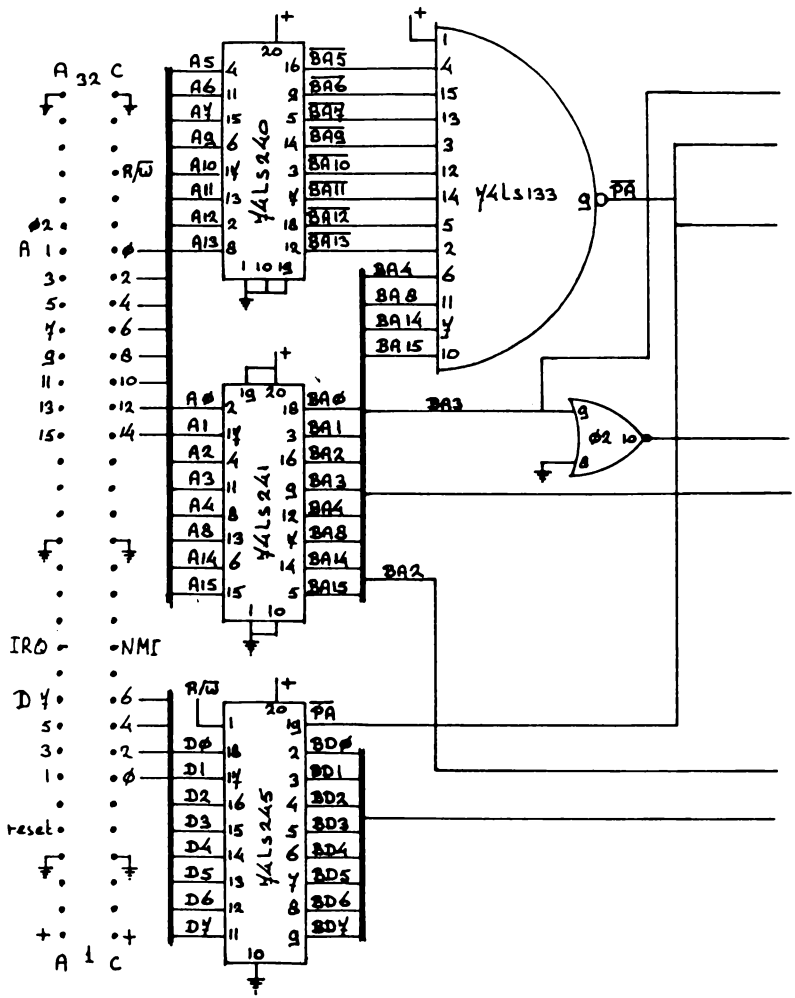
De Print is te bestellen bij het MOSI team.

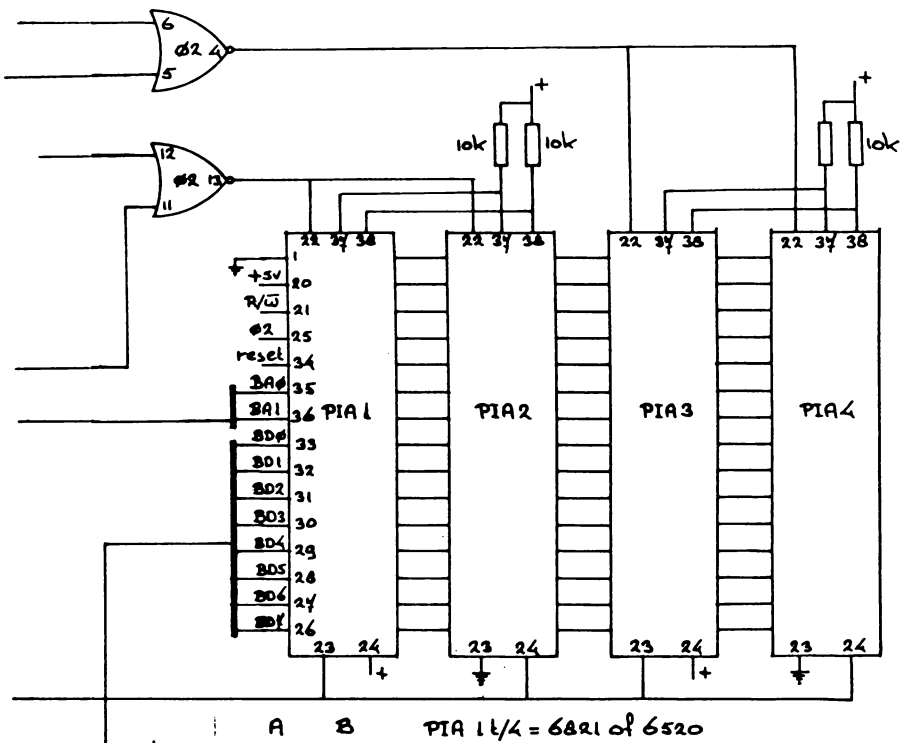
Ko van Ekeren	Martien Schot
Reyerscop 26	Hofstede 21
3454HX De Meern	3902CG Veenendaal



KONNEKTOR PIA 2/4	I/O	KONNEKTOR PIA 1/3
1	CA2	40
2	CA1	39
3	PA0	38
4	PA1	37
5	PA2	36
6	PA3	35
7	PA4	34
8	PA5	33
9	PA6	32
10	PA7	31
11	PB0	30
12	PB1	29
13	PB2	28
14	PB3	27
15	PB4	26
16	PB5	25
17	PB6	24
18	PB7	23
19	CB1	22
20	CB2	21

OSI POEL


A 32 C
R/W
Ø2
A 1
Ø
2
4
6
8
10
12
14
IRQ
NMI
D 7
reset
A 1 C
74LS240
74LS241
74LS245
74LS133
A5
A6
A7
A9
A10
A11
A12
A13
BA5
BA6
BA7
BA9
BA10
BA11
BA12
BA13
BA4
BA8
BA14
BA15
A0
A1
A2
A3
A4
A8
A14
A15
BA0
BA1
BA2
BA3
BA4
BA8
BA14
BA15
PA
Ø2
R/W
D0
D1
D2
D3
D4
D5
D6
D7
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7



	A	B
contr. PIA 1	C119	C11B
" " 2	C11D	C11F
" " 3	C111	C113
" " 4	C115	C117
I/O reg PIA 1	C118	C11A
" " 2	C11C	C11E
" " 3	C110	C112
" " 4	C114	C116

PIA 1 t/4 = 6821 of 6520

Mosi I/O kaart

GETEKEND: KD VAN EKEREN

ONTWERP: Martien Schot 8-'83

OSI POEL

WAIT WATCHER

Het WAIT-statement wordt weinig in de OSI-programma's gebruikt. In "The 6502 journal" (febr. '81) vroeg R.L. Elm zich ook al af waarop we nog wachten.
Toch is het een bekijken waard en ik hoop danook dat dit artikeltje je tenminste op een idee brengt. Als het maar niet zoveel ergernis geeft dat je daar zo broodmager van wordt dat het lijkt alsof je ge-"weight-watched" hebt!
Hoe werkte WAIT ook al weer ?
WAIT I,J,K leest de decimale waarde op adres I, exclusive or't (EOR) dit met K en AND het resultaat met J. WAIT wordt "gepasseerd" als de laatste berekening 0 is. Als je K weglaat wordt hiervoor 0 verondersteld.
EOR doet mij overigens denken aan de zgn. hotelschakeling die wordt gebruikt bij gangen en trappen. Staan de schakelaars in een ongelijke stand dan brandt de lamp !
6 EOR 5 geeft dus 3 omdat binair 110 EOR 101 = 011.
AND zou je kunnen vergelijken met een zgn. serieschakeling: allebei de schakelaars aan en de lamp brandt.
6 AND 5 = 4 omdat binair 110 AND 101 = 100.
Nu weer terug naar WAIT. Als je van te voren weet welke waarde op een bepaald adres staat of komt, levert I EOR K -> 0 op.
Twee praktische toepassingen:
vervang (bijv. in Hobbyscoop-programma's)
INPUT "Druk voor vervolg op <RET>";A$ door
PRINT "Druk voor vervolg op <ESC>":WAIT 57088,32,254
dan springt de OSI niet meer uit zijn programma maar gaat geluidloos verder. (Zie ook 4e 50 pag.boekje, pag.44 (EC10) Rn.10760)
10000 WAIT 61440,1:PRINT CHR$(PEEK(61441));:GOTO 10000
is een handige VIEW-routine om op een bandje iets op te zoeken met gebruik van het beeldscherm, terwijl het programma in de computer niet wordt verknoeid.
Voor experimenten zie bijgaande programma's. (zie EE 38)

Ton Helwig, Wageningen

PROGRAMMASTRUKTUREN IN BASIC EN FORTH.

Bij het gestruktureerd programmeren gaat men er vanuit dat elk programma kan worden samengesteld uit een beperkt aantal basisstrukturen.
Deze strukturen worden in het algemeen als volgt aangeduid:

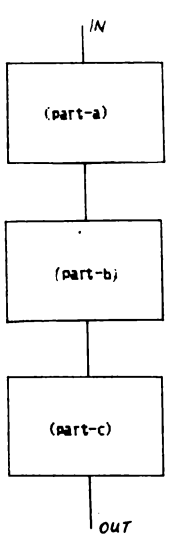
1. De sequence of volgorde
2. De IF...THEN...ELSE... konstruktie
3. De REPEAT...UNTIL... konstruktie
4. De WHILE...DO... konstruktie

In het nuvolgende zal voor elk van deze strukturen het stroomdiagram worden aangegeven, alsmede de implementatie van de struktuur in zowel BASIC als FORTH.
Voor praktische voorbeelden wordt verwezen naar de, elders in dit boekje opgenomen, programma's:

-FORTH-KLOK

-Russisch Roulette

1. De sequence of volgorde



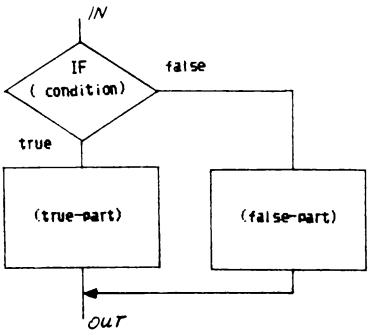
BASIC

```
100REM (part-a)
-
-
200 REM (part-b)
-
-
300 REM (part-c)
-
-
```

FORTH

```
: PROGRAM (part-a part-b part-c) ;
```

2. De IF...THEN...ELSE... construktie



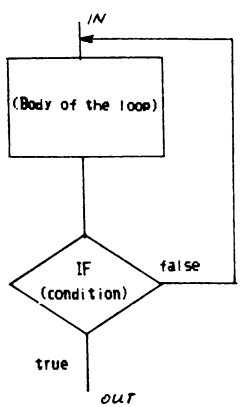
BASIC

```
100IF (conditie) THEN 200
110REM (false-part)
-
-
-
190GOTO300
200REM (true-part)
-
-
-
300REM (next construct)
```

FORTH

```
( condition) IF ( true part)
            ELSE ( false part)
            THEN ( next construct)
```

3. De REPEAT...UNTIL...construktie



BASIC

```
100FOR D=0 TO 1
110REM (Body of the loop)
-
-
-
200D=(condition)
210NEXT D
220REM ( next construct)
```

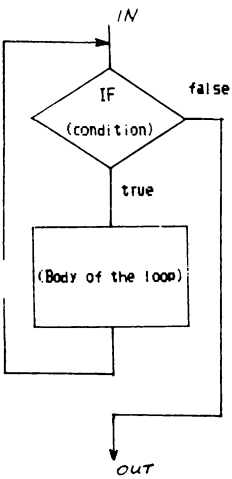
FORTH

```
BEGIN ( Body of the loop)
      ( condition)
UNTIL ( next construct)
```

Opmerking:

als het aantal malen dat de lus doorlopen wordt bekend is, kan ipv. REPEAT...UNTIL de FOR...NEXT (BASIC) of de DO...LOOP (FORTH) construktie worden toegepast.

.. De WHILE...DO...construktie



BASIC

```
100IF ( inversed condition) THEN 200
110REM (Body of the loop)
-
-
-
190GOTO100
200REM ( next construct)
```

FORTH

```
BEGIN ( condition)

WHILE ( Body of the loop)

REPEAT ( next construct)
```

RØUTINE 5 FATAL ERRØRS

NAAR DEZE RØUTINE WØRDT GESPRØNGEN BIJ EEN FØUT WAARBIJ BASIC MØET AFBREKEN. DE FØUT WØRDT GEMELD & DE BASIC SPRINGT NAAR DE COMMAND-MØDE

ØNVØLDØENDE GENEMGEN FØUT

```
A24C- A2 0C     LDX #$0C ;LAAT X NAAR DE 'ØM' WIJZEN
```

FØUT AFDRUK

DEZE RØUTINE DRUKT EEN FØUTMELDING AF. AAN DE HAND VAN DE INHØUD VAN HET X-REG. WØRDT IN EEN TABEL DE FØUT ØPGEZØCHT TWEE LETTERS PER FØUTMELDING WØRDEN AFGEDRUKT ØM DE SØØRT FØUT AAN TE GEVEN (WEL MET BIT7=1 ZØDAT ER EEN GRAFISCH SYMBØØL VERSCHIJNT I.P.V. EEN LETTER.

```
A24E- 46 64     LSR $64      ;ZØRG DAT ER GEPRINT KAN WØRDEN
A250- 20 6C A8  JSR $A86C    ;PRINT EEN CR/LF
A253- 20 E3 A8  JSR $A8E3    ;PRINT EEN VRAAGTEKEN '?'
A256- BD 64 A1  LDA $A164,X  ;HAAL EERSTE LETTER ØP
A259- 20 E5 A8  JSR $A8E5    ;DRUK HET AF
A25C- BD 65 A1  LDA $A165,X  ;DAN DE TWEEDE LETTER
A25F- 20 E5 A8  JSR $A8E5    ;ØØK AFDRUKKEN
A262- 20 91 A4  JSR $A491    ;ZET AL DE BASIC VERWIJZINGEN TERUG
A265- A9 86     LDA #$86     ;LAAT A+Y WIJZEN NAAR DE PLAATS WAAR
A267- A0 A1     LDY #$A1     ;DE 'ERRØR' STRING TE VINDEN IS.
A269- 20 C3 A8  JSR $A8C3    ;DRUK DIE AF.
A26C- A4 88     LDY $88      ;NEEM HET HØGE BYTE V/H HUIDIGE REGEL-
A26E- C8        INY          ;NUMMER, VERHØØG MET EEN.
A26F- F0 03     BEQ $A274    ;ALS 0, DAN IN DE DIRECT MØDE, SPRINGEN
A271- 20 53 B9  JSR $B953    ;DRUK 'IN' AF MET HET REGELNUMMER
```

RØUTINE 6 WARME BASIC START

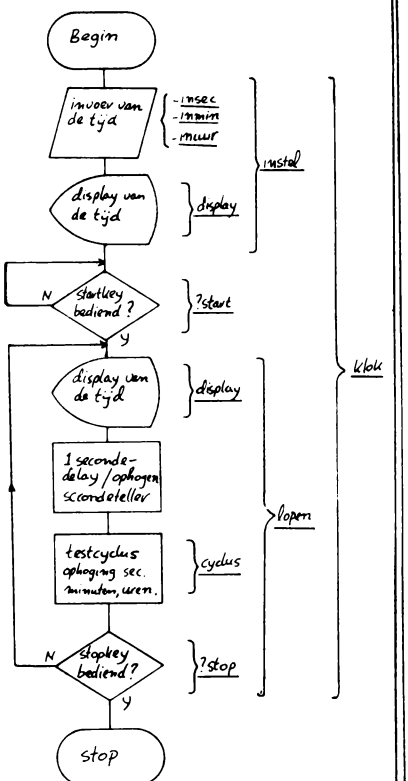
NA HET ØPSTARTEN IS DIT HET ALGEMENE BEGINPUNT.

```
A274- 46 64     LSR $64      ;ZØRG DAT ER GEPRINT KAN WØRDEN.
A276- A9 92     LDA #$92     ;LAAT A+Y WIJZEN NAAR HET ADRES WAARMEE
A278- A0 A1     LDY #$A1     ;'ØK' WØRDT AFGEDRUKT
A27A- 20 03 00  JSR $0003    ;DØE DAT DAN ØØK
A27D- 20 57 A3  JSR $A357    ;VUL DE INPUT-BUFFER
A280- 86 C3     STX $C3      ;ZET HET ADRES VAN DE BUFFER IN DE
A282- 84 C4     STY $C4      ;VERBINDINGSRØUTINE (LØW-HIGH ADRES)
A284- 20 BC 00  JSR $00BC    ;HAAL EERSTE LETTER BUITEN 'N SPATIE
A287- F0 F4     BEQ $A27D    ;ZØLANG DE INPUTBUFFER LEEG IS, TERUG
A289- A2 FF     LDX #$FF     ;ZET DE BASIC IN DE DIRECT-MØDE,
A28B- 86 88     STX $88      ;(DAN REAGEERT IE ZØNDER REGELNUMMERS)
A28D- 90 06     BCC $A295    ;ALS EERSTE LETTER=CIJFER NAAR EDITRØUT.
A28F- 20 A6 A3  JSR $A3A6    ;ANDERS 'N KØMMANDØ D'US ØMZETTEN (RØUT.11)
A292- 4C F6 A5  JMP $A5F6    ;GA HET KØMMANDØ UITVØEREN.
```

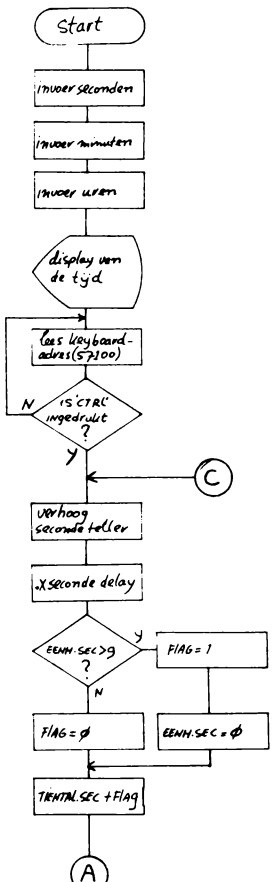
FORTH-klok, listing

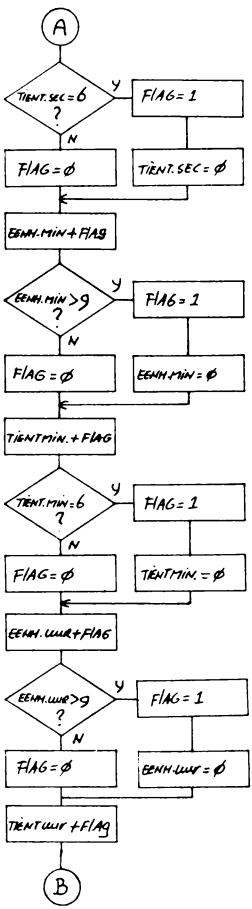
```
25 LIST
SCR # 25
  0 ( FORTH-KLOK)
  1 0 VARIABLE S1 0 VARIABLE S10 0 VARIABLE MIN1 0 VARIABLE MIN10
  2 0 VARIABLE UUR1  0 VARIABLE UUR10 0 VARIABLE FLAG
  3 0 VARIABLE D1 : P DUP . ;
  4 : KLOKRESET 0 S1 ! 0 S10 ! 0 MIN1 ! 0 MIN10 ! 0 UUR1 !
  5          0 UUR10 ! ;
  6 : CLS 3 EMIT ;
  7 : W1  ." EENHEDEN " ;
  8 : W10 ." TIENTALLEN " ;
  9 : SC   ." SECONDEN: " ;
 10 : MT  ." MINUTEN: " ;
 11 : UR  ." UREN : " ;
 12 : KOM   KEY 48 - ;
 13 : INSEC W1 SC KOM P S1 ! CR W10 SC KOM P S10 ! CR ;
 14 : INMIN W1 MT KOM P MIN1 ! CR W10 MT KOM P MIN10 ! CR ;
 15 —)
OK
26 LIST
SCR # 26
  0 ( KLOK VERVOLG )
  1 : INUUR W1 UR KOM P UUR1 ! CR W10 UR KOM P UUR10 ! CR ;
  2 : ASC  @ 48 + ;
  3 : SET 1 FLAG ! ;
  4 : RESET 0 FLAG ! ;
  5 : DELAY D1 @ 0 DO LOOP ;
  6 HEX
  7 : DISSEC S1 ASC D2DF C! S10 ASC D2DE C! ;
  8 : DISMIN MIN1 ASC D2DC C! MIN10 ASC D2DB C! ;
  9 : DISUUR UUR1 ASC D2D9 C! UUR10 ASC D2D8 C! ;
 10 DECIMAL
 11 : DISPLAY   DISSEC DISMIN DISUUR ;
 12 : INSTEL CLS INSEC INMIN INUUR CLS DISPLAY ;
 13 : ?START  BEGIN 57100 @ -16706 = UNTIL ; ( START IF 'CTRL')
 14 : ?STOP  57100 @ -8482 = ; ( TRUE IF 'ESC')
 15 —)
OK
27 LIST
SCR # 27
  0 ( KLOK VERVOLG2)
  1 : TIM1 S1 @ 9 ) IF SET 0 S1 ! ELSE RESET THEN
  2   FLAG @ S10 +! ;
  3 : TIM2  S10 @ 6 = IF SET 0 S10 ! ELSE RESET THEN
  4  FLAG @ MIN1 +! ;
  5 : TIM3 MIN1 @ 9 ) IF SET 0 MIN1 ! ELSE RESET THEN
  6  FLAG @ MIN10 +! ;
  7 : TIM4  MIN10 @ 6 = IF SET 0 MIN10 ! ELSE RESET THEN
  8  FLAG @ UUR1 +! ;
  9 : TIM5 UUR1 @ 9 ) IF SET 0 UUR1 ! ELSE RESET THEN
 10   FLAG @ UUR10 +! ;
 11 : URTST  UUR1 @ 4 = UUR10 @ 2 = AND ; ( -- TRUE IF 24 )
 12 : ?24-UUR  URTST IF KLOKRESET THEN ;
 13 : CYCLUS TIM1 TIM2 TIM3 TIM4 TIM5 ?24-UUR ;
 14 : LOPEN  BEGIN DISPLAY 1 S1 +! DELAY CYCLUS ?STOP UNTIL ;
 15 : KLOK  INSTEL ?START LOPEN ;  ;S
OK
```

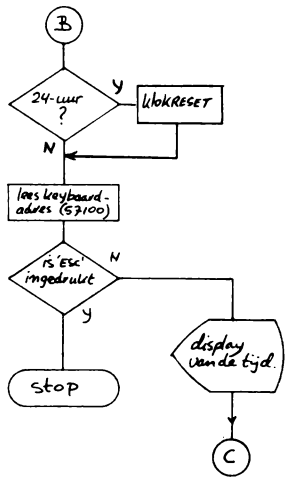
FORTH-klok, programmastructuur.



FORTH-klok, Flowchart







opmerkingen:

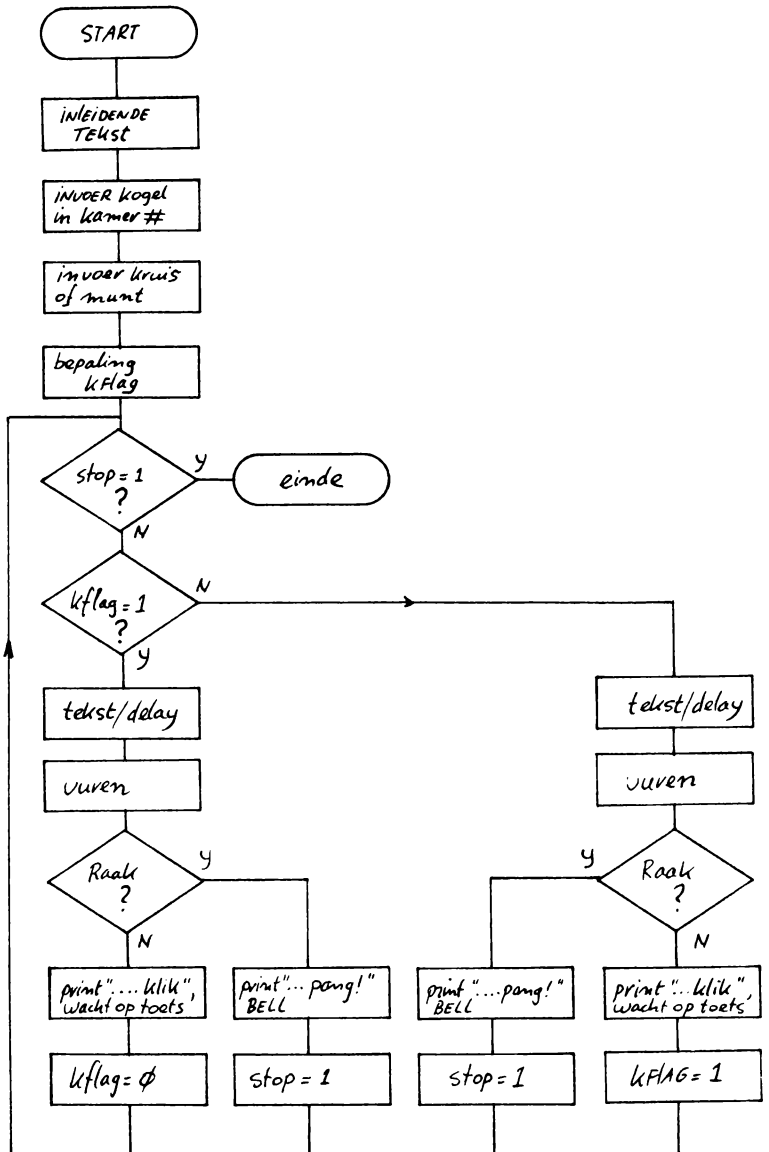
- Voor een superboard met klokfreq. van 1 MHz moet D1 een waarde hebben van circa 8500 (decimaal). De exacte waarde dient proefondervindelijk te worden vastgesteld.
- De onderstreepte woorden verwijzen naar de betreffende FORTH-woorden.

H. coenen, nov '82

Russisch Roulette, listing

```
60 LIST
SCR # 60
  0 ( RUSSIAN ROULETTE, DRUK NA ELK SCHOT OP EEN TOETS!)
  1 ( LAAD RANDOMNUM.GENERATOR,ZIE 50P-BOEKJE EE37 SCR#31)
  2 : CLS 3 EMIT ;
  3 : DELAY @ DO LOOP ;
  4 0 VARIABLE KFLAG  0 VARIABLE STOP  0 VARIABLE KAMER
  5 : IN  KEY 48 - ;
  6 : TEKST CLS ." DIT IS RUSSISCH ROULETTE, EEN LEVENSGEVAARLIJK
  7  SPEL. " CR ." BENODIGD: 1 COMPUTER, 1 HUMAN (YOU!) EN" CR
  8   ." 1 REVOLVER MET 6 KAMERS" CR CR ;
  9 : VR1 ." WE TOSSEN WIE BEGINT,KRUIS(1) OF MUNT(0)"  ;
 10 : KINV ." IN WELKE KAMER (1-6) MOET DE KOGEL ?" IN KAMER ! ;
 11 : KFLAGKEUZE CR CR VR1 IN 2 CHOOSE = KFLAG ! ;
 12 : ?RAAK KAMER @ = ; ( TRUE IF RAAK)
 13 : VUREN 7 CHOOSE 13000 DELAY ;
 14 : ?STOP STOP @ ;
 15 : INIT. TEKST KINV KFLAGKEUZE ; —)
OK
61 LIST
SCR # 61
  0 ( RUSSIAN ROULETTE VERVOLG)
  1 : TH1 CLS ." HET IS JOUW BEURT, ZWEET...." ;
  2 : TC1 CLS ." HET IS MIJN BEURT, ZWEET...." ;
  3 : TH2 ." ...KLIK!!" KEY ;
  4 : TC2 TH2 ;
  5 : TH3A ." ...PANG!!!" 10 0 DO 7 EMIT LOOP ;
  6 : TC3A TH3A ;
  7 : TC3B CR ." JAMMER, COMPUTER DOOD!" CR ;
  8 : TH3B CR ." HUMAN DOOD!! " CR ;
  9 : TH3 TH3A TH3B ;
 10 : TC3 TC3A TC3B ;
 11 : COMPUTER TC1 VUREN ?RAAK IF TC3 1 STOP ! ELSE TC2 1
 12  KFLAG ! THEN ;
 13 : HUMAN TH1 VUREN ?RAAK IF TH3 1 STOP ! ELSE TH2
 14  0 KFLAG ! THEN ;
 15 : KEUZE KFLAG @ IF HUMAN ELSE COMPUTER THEN ; —)
OK
62 LIST
SCR # 62
  0 ( RUSSIAN ROULETTE VERVOLG)
  1 : RESET 0 STOP ! ;
  2 : RUSSIAN RESET INIT. BEGIN KEUZE ?STOP UNTIL ;
  3 ;S
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
OK
```

Russisch Roulette, Flowchart



Artikelen voor OSI bezitters uit de HCC Nieuwsbrief

HCCN nr.		
10	H.J.Wevers	Geheime programma's
10	H.J.Wevers	Gebruikersrapport OSI SUPERBOARD en chall. 1P
12	H.J.Wevers	Geheime programma's (nogmaals)
14	R.Heesterman	Systeem software OSI
14	H.J.Wevers	50 Hz video voor Superboard
14	R.Jansen	Spiralen voor de OSI C1P en Superboard
14	H.J.Wevers	Basic trace voor de OSI
14	F.vd Bosch	Video-ingang
15	R.Heesterman	Geheugenorganisatie
		Programmeren in machinetaal
15	H.J.Wevers	Videoroutine 32*30
15	P.Broers	Opzoeken van Programma's
16	H de Jong	Heathkit H14 aan de Challenger 2P
		Hexdump 65V
16	J.Burghouts	Demonstratie karakterset
		Poke adressen
		Fruitautomaat
16	H.de Jong	OSI 8 K basic manual
17	P.Broers	Machinetaal op cassette
17	R.Jansen	Random vullen van het scherm
		Lissajous en Cirkel
17	H.J.Wevers	Meer tekens op het scherm
18	A.Hilgersom	Warme start na BREAK
18	C.Romijn	Verbeterde grafieken op beeldscherm
18	H.Coenen	Outputpoort voor het OSI-Superboard (zie ook HCCN20)
18	H.J.Wevers	OHIO Challenger C4P MF testrapport
19	P.Broers	Superboard toetsenbord-perikelen
19	K.Romijn	Teletype 33 aan Superboard
20	N.Awater	Weer een tipje van de OSI-Basic sluier
20	Ton vd Bosch	Korrektie op outputpoort van HCCN 18
21	J.Burghouts	Spiralen tekenen
		Funktie teken programma
21	P.Hinderks	Hardware wijzigingen Superboard en 2P
21	H.de Wal	Toonladders met het Superboard
22	J.Burghouts	Cassette-kijker
22	A.Hilgersom	Machinetaal op cassette, maar anders
22	H.de Wal	Help, Help, hoe krijg ik mijn programma terug
22	H.Kuska	Random number generator
23	N.Awater	Superboard autoreset
23	P.Broers	Singlist, single step lister
24	J.Hermans	OSI artikelen / OSI users groep
25	P.Hinderks	OSI Superboard als orgel
25	J.Burghouts	Het onderbreken van Basic
25	L.van Essen	Baudrate schakeling OSI
25	J.Hermans	OSI boek nr 1
26	H.J.Wevers	Binnen in de Microsoft Basic 1
		Joystick voor 1P en Superboard II

27	H.Coenen	OSI g.g.
27	R.Heesterman	Volgende OSI dag
27	J.Landman	Superboard autoreset
27	H.Bosma	Programmeerbare karaktergenerator voor OSI
27	T.Meyering	Vertikale baan op Philips TV
28	H.J.Wever	Microsoft Basic 2
28	J.Oelp	Lichtkrant OSI
28	P.Broers	Videoram test
28	J.Hermans	OSI boek nr 2 / OSI basic
28	J.Ros	Autoreset voor C1p / Superboard
29	J.vd Linden	Hernummeren
29	J.Glaser	Viditel voor OSI
29	J.Hermans	OSI artikelen
30	J.vd Linden	Labels voor OSI basic
30	J.Cornelissen	Doolhof OSI
30	A.Mathlener	OSI beeldloper
30	H.J.Wevers	Koning OSI
30	J.Oelp	De voetbaltoto invuller
30	J.Glaser	Lotto kontroleur OSI
30	G.Wiselius	Wijziging Elektuur Ram/Eprom kaart voor OSI
30	A.Megens	Addendum 2e 50 pag.boek OSI
30	J.Hermans	OSI artikelen
31	T.Helwig	Ook pas aan de OSI ?
31	P.Broers	Selective loader OSI
31	J.vd Linden	Autonumber OSI /funktieinvoer in lopend programma
32	J.Verrij	Handshake voor de printer OSI C1p
32	J.Hartog	OSI weer op poten / grappen op het Superboard
32	P.Broers	Lister subroutine OSI
33	H.J.Zimberlin	Serial interface OSI-Epson MX 80
33	B.Heesbeen	Datarecorder en het OSI Superboard
33	P.Broers	Poging tot call of sys OSI
33	J.Hermans	"Piep" voor C1P en Superboard II
33	B.Heesbeen	Klok OSI
34	T.Helwig	OSI 1p Basicode, eenvoudige baudrate schakeling Druk(te) fouten OSI, Autonum HCCN31,Singlist HCCN23
34	H.J.Wevers	OSI Basicode
34	P.Broers	USR(X) en OSI Challenger
35	J.Burghouts	OSI software standaard / 2e en 3e OSI Cassette
36	H.Kooy	OSI gg overal
37	J.vd Linden	Ook pas aan de OSI ? (2) /erratum funktieinvoer lopend programma HCCN31
37	?????	Nogmaals hernummeren
37	H.Barten	Erratum OSI hernummer
38	C.Romijn	Harmonischen OSI
38	J.vd Linden	OSI grafiek

Bijgewerkt tot 1-4-1982

Jos Burghouts

OSI PRINTSERVICE

Vele computeristen zouden wel eens graag van "die ene schakeling" een printje willen (laten) maken.

Daar dan bij navraag de moeilijkheden te groot blijken te zijn, gaat het stuk niet door.

Welnu, wij willen U de mogelijkheid bieden om voor een redelijke prijs toch een eenmalige - of klein aantal print(en) te laten maken.

De prijs wordt bepaald door:

a- kwaliteit van het printontwerp
b- enkelzijdige of dubbelzijdige print
c- afmetingen van de print
d- geboord of ongeboord

ad a als de koperbanen te dicht tegen elkaar liggen, kan het best zijn dat een printje mislukt.
Dat mislukken is voor rekening van de ontwerper!

ad b en c materiaal kosten.

ad d zelf gaten boren spaart veel geld, waarom dat zo is, weet U wel nadat U zo'n paar honderd gaten geboord heeft.

ONTWERP EN LAY-OUT

Gebruik voor Uw ontwerp papiermet een raster van 0,1 inch, in verband met de internationale standaardisatie van onderdelen.

Bij Uw ontwerp moet U voor ogen houden dat de gaten van een tweezijdige print niet doorgemetaliseerd zijn.
Dit vraagt extra zorg, n.l. waar U naar de andere zijde wilt doorvoeren, moet U twee, precies boven elkaar liggende, cirkels plakken, waarin later een doorverbinding gesoldeerd kan worden.

Als U een bewerkelijk ontwerp gemaakt heeft, teken dan alles in een schaal 2:1. Zo plakt U dan ook de film of folie, dit vergemakkelijkt het werk aanzienlijk.

Het verwerken van een connector paar is goedkoper dan vernikkeld - vergulde uitlopers + een connector.

HET PLAKKEN OP DOORZICHTIG MATERIAAL

Gebruik plakmaterialen, geen wrijfmaterialen. Wrijfmateriaal kan aanleiding geven tot haarscheurtjes die met het blote oog niet waar te nemen zijn maar door de camera haarfijn weergegeven worden en straks in Uw ontwerp als onderbrekingen roet in het eten gooien.

Plakmateriaal is eventueel door de printservice te leveren.

Het doorzichtige materiaal kan zijn: z.g. calkeer papier, een licht melkachtig witte folie of kleurloze plastic film van krimp- en rekarm materiaal van minimaal 0.2 mm dikte.

Snij folie of film krap af, zodanig dat er maximaal een rand van 1 cm buiten de tekening ontstaat.

Geef de afmetingen van het plaatje aan met kruisen van (op schaal 1:1) 0,5 mm breedte. De maat van de plaat wordt door de binnenkant van de kruisen aangegeven.

Zorg dat minimaal 1½ mm binnen de rand van het printje geen koperbanen voorkomen.

Tracht zoveel mogelijk brede kopersporen te plakken. Ga indien mogelijk niet beneden 0,5 mm op ware grootte.
De ~~tussenruimte tussen~~ de kopersporen en/of cirkels mag niet minder zijn dan 0,5 mm o.w.gr.(denk aan de prijs van mislukte prints!)

Bij plakken op schaal 2:1 dient U zodanig materiaal te benutten dat na verkleinen tot 1:1 nog werkbare eilandjes ontstaan.
Een cirkel zou na verkleinen niet minder dan 1,5 mm diam. mogen hebben.

De geplakte lay-outs voor tweezijdige prints moeten zijn voorzien van enkele referentie punten. Deze moeten bestaan uit cirkels, geen kruisen hiervoor gebruiken. Deze cirkels moeten binnen het printontwerp liggen.
Ook bevestigingsgaten moeten met een cirkel worden aangegeven. Houdt U dan ook daaromheen rekening met de kop van het boutje?

Verdeel de koperbezetting zo regelmatig mogelijk, vul eventuele open plekken op met tekst. Gebruik voor tekst een minimale dikte van 0,35 mm.
Ook zou U op open plekken een losse voet kunnen aanbrengen. Dat kan soms makkelijk zijn.

Indien op de gehele printzijde geen tekst voorkomt, zet dan op de film of folie met een zwarte viltstift "koperzijde".

Bestel Uw prints in rechthoekige vorm. Moeten er nog hoekjes uit, zaag die er dan later zelf uit.

Voeg bij Uw bestelling ook de tekening of een kopie daarvan.

De printfabrikant heeft een negatief van de lay-out nodig voor zijn procede. Als U dit zelf kunt (laten) maken (koperbanen wit, de achtergrond zwart) dan geldt ook hiervoor: U moet instaan voor de kwaliteit van de opname.

Lever alle materialen vlak in, d.w.z. rol niet de hele bubs in een koker en verstuur die.

Stuur Uw zending naar:

PROEFPRINT GAALMAN B.M.A. tel: 015 - 134269
POSTBUS 150 giro: 245790
2624 AP DELFT

Sluit een briefje bij waarop U vermeldt:

aantal printjes
volledig adres
telefoonnummer
HCC lidmaatschaps nr.

Het kan zijn dat de printfabrikant zodanige aanmerkingen op Uw ontwerp heeft dat hij het niet verantwoord vindt om de print te maken. U krijgt dan Uw materiaal teruggestuurd met de nodige opmerkingen.

De printjes worden zo snel mogelijk geleverd. De levertijd is echter afhankelijk van de bezetting met professionele opdrachten.

U krijgt de printjes thuisgestuurd vergezeld van een faktuur. Deze faktuur is strikt binnen 30 dagen te betalen.
Mocht men hier niet de hand aan houden, dan zal tot rembourszendingen (erg duur) worden overgegaan.

Verdere inlichtingen over deze service bij Uw OSI voorzitter.

GOEDKOOP GROEN TV/MONITOR BEELDSCHERM

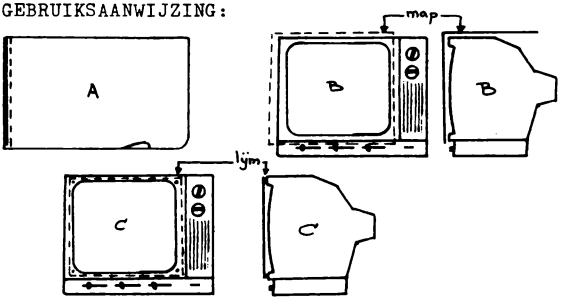
EEN IDEE VOOR DE SMALLE BEURS

Een tijdje geleden was ik opzoek naar een monitor met een groene beeldbuis, ik had een bibberende TOKYO-portable-TV. Na enig "marktonderzoek" vond ik een echte monitor toch te duur.

Op advies van een andere OSI-SBII bezitter kon ik mijn TV ombouwen naar monitor. En door externe voeding was het beeld zeer aanvaardbaar maar nog steeds zwart/wit !!

In een ruimgesorteerde kantoor/boekhandel vond ik TRANSPARANTE PLASTICMAPPEN in diverse kleuren oa. geel en groen (afm ±35x22 cm), kosten ƒ0,80.

GEBRUIKSAANWIJZING:



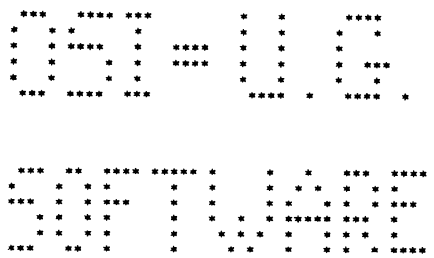
afb. A snij de map open op de ---------

afb. B hang één zijde voor de TV/Monitor probeer eventueel ook andere kleuren.

afb. C snij na definitieve keuze één zijde op maat en plak dit stuk met bv. dubbelzijdig kleefband voor het scherm.

Ziezo klaar is Kees voor een paar kwartjes !!!!!

Ton Graafmans
Bronkhorststr. 41
2316 SZ LEIDEN
071 - 132598



S P E L R E G E L S

== PRØGRAMMA'S KUNNEN ALLEEN WØRDEN BESTELD DØØR HCC EN ØSI-UG LEDEN.

== SØFTWARE MAG WELWØRDEN GEKØPIEERD, EN ØP ELKE MØGELIJKE MANIER WØRDEN VERSPREID.

== HET AANLEVEREN VAN PRØGRAMMA'S GEEFT VEEL VØØRDELEN, TEGØED-BØNNEN BIJ TØEKØMSTIGE BESTELLINGEN EN JE NAAM IN DE Ø.S.I. U.G. SØFTWARE ANALEN !!

== BESTELLINGEN KUNNEN HET BESTE WØRDEN GEDAAN MET BEHULP VAN EEN ØVERSCHRIJVING ØP DE ØSI GIRØREKENING: 2012367
T.N.V. HCC ØSI GEBRUIKERS GRØEP
MØNTGØMERYLAAN 112
2625 PR DELFT

B E T A L I N G

== DE PRIJS VAN EEN BANDJE IN DE DIREKTE VERKØØP IS 7.50, ALS ZE VERZØNDEN WØRDEN MØET 11.50 WØRDEN BETAALD.

== EEN FLØPPY KØST BIJ DE RECHTSTREEKSE VERKØØP 11.00 EN BIJ VERZENDING 15.00

== ALS NIET VIA DE ØSI-GIRØ WØRDT BESTELD, MØET ALS BETALINGSBEWIJS ØFWEL;
- KØNTANT GELD
- EEN BETAAL(ØF EURØ)-CHEQUE
) EEN GIRØ-BETAALKAART
(NATUURLIJK ØNDERTEKEND !!!!)
WØRDEN MEEGESTUURD.

V Ø Ø R W A A R D E N

== BESTELLINGEN DIE NIET AAN DE VØØRWAARDEN VØLDØEN DIE HIERVØØR VERMELD STAAN WØRDEN ØNVERBIDDELIJK TERUG-GESTUURD (ALS ER WEL GELD WAS MEEGESTUURD), ØF VERNIETIGD !!!! JUIST INVULLEN EN GØED BETALEN VØØRKØMT DUS VEEL NARIGHEID VØØR IEDEREEN !!!!

D Ø K U M E N T A T I E

== DØKUMENTATIE KAN MAAR BIJ UITZØNDERING WØRDEN BESTELD.
NEEM DAARVØØR KØNTAKT ØP MET DE SØFTWARE-KØØRDINATØR.

ØP HET EERSTE BANDJE STAAT ;

1. BASIC BASIC INTERPRETER
ZIE DE SERIE IN DE HCC N
2. BATTLESHIP
ZØALS HET BEKENDE ZEESLAG
3. BLACK JACK
21-EN GRAFIES
4. BØMBER
EEN GRAFIES BØMMENWERPER-NEERSCHIET SPEL
5. BØTER KAAS EN EIEREN
BEKEND ØUD-HØLLANDS SPEL
6. 3D BØTER KAAS EN EIEREN
EEN INGEWIKKELDE VERSIE !!
7. DESTRØYER
EEN GRAFIES SCHIET-SPELLETJE
8. GØBANG
EEN ØØSTERS INTELLIGENTIE-SPEL

BAND NUMMER TWEE

1. FRUITAUTØMAAT
EEN EENARMIGE BANDIET IN UW KØMPUTER
2. ØSI ØRGEL
ELEKTØNISCH ØRGEL, GRAFISCH WEERGEGEVEN
3. MASTERMIND
BREEK DE KØDE, GEEF ZELF ØØK EEN KØDE!
4. STARTREK I
SNEL GRAFISCH RUIMTESPEL, ZEER STERK
5. DØØLHØF
ØNTSNAP UIT EEN VERANDEREND DØØLHØF
6. BASIC HERNUMMER
GØED WERKEND, NUTTIG PRØGRAMMA
7. MAGISCHE VIERKANTEN
AARDIGE WISKUNDIGE TØEPASSING

BAND DRIE

1. BREAK ØUT
GRAFISCH T.V. TENNISSPEL (MAAR LEUKER)
2. FRUSTRATIØN
SØØRT GRAFISCH SØLITAIRE, DENKSPEL
3. GALGJE
WØØRDEN-RAAD-SPELLETJE
4. HALLØ
PRAAT 'INTELLIGENT' MET DE KØMPUTER
5. HAMMURABI
BEZØRG UW REPUBLIEK WELVAART EN WELZIJN
6. KØNING
DIEN UW LAND ALS LEVENDE KØNING !!
7. LIFE V 2.0
GRAFISCH SIMULATIE-SPEL
8. NEW YØRK TAXI
HØUDT ZØNDER KLEERSCHEUREN EEN TAXI AAN

BAND NUMMER VIER

1. MØRIA
HAAL DE SCHAT UIT DE ØNDERWERELD
2. MØRSECURSUS
EEN 'MUST' VØØR STARTENDE ZENDAMATEURS
3. MØRSE ZEND/ØNTVANGER
STUUR EN ØNTVANG ECHTE MØRSE
4. REKENEN
SCHØØLVØØRBEELD ØM REKENEN TE LEREN
5. RTTY ZEND/ØNTVANGEN
STUUR EN ØNTVANG TELETYPE KØDE
6. MØRSECØDE GEVER
TYPE ASCII IN, ER KØMT MØRSE UIT
7. MØRSE ØMZETTER
ER GAAT MØRSE ØN, KØMT ASCII UIT

BAND VIJF
NØG IN BEWERKING

1. FØRTH
DE ECHTE FIG-FØRTH MET EDITØR !!
2. QUICKSAVE
LAAD EN LØS MACHINETAAL VIA UW VIDEØ, SNEL EN ZEKER ZØNDER EXPRA GEHEUGEN

BAND ZES

1. DUMB TERMINAL V2.1
   ØM MET GRØTE KØMPUTERS TE KUNNEN PRATEN
2. DUNGEØNS
   BEVECHT FANTASTISCHE SPRØØKJESFIGUREN
3. ETCH A SKETCH
   UW KØMPUTER ALS GRAFISCH TEKENBØEK
4. PAARDERENNEN
   VERANTVØØRD (GRAFISCH) GØKKEN TØT S MAN
5. WUMPUS II
   ZØEK EN DØØD HEM IN DE DØNKERE GRØTTEN
6. MELØDIE 1
   MUZIEK UIT DE KØMPUTER
7. VØGELTJESDANS
   MEER MUZIEK UIT UW KØMPUTER

BAND NUMMER ACHT
IN BEWERKING

1. WARI
   ØUD AFRIKAANS DENKSPEL
2. CØNDØT
   ØER HØLLANDS VERHUIZERTJE SPELEN
3. EENENTWINTIGEN
   SPEEL GRAFISCH KAART TEGEN DE KØMPUTER
4. REVERSI
   VERSLA HEM IN DIT GRAFISCH BØRDSPEL
5. MIJNENVELD
   PRØBEER ERØVERHEEN TE KØMEN
6. MAANLANDER
   LAND ØP UW GRAFISCH BEELDSCHERM

BAND NUMMER ZEVEN

1. CRAZY BALL
   EEN GRAFISCH T.V. TENNISSPEL
2. RENDIER (ALLEEN IK VIDEØ !)
   EEN ØSI KERSTGRØET
3. SEAWØLF
   BRENG SCHEPEN (GRAFISCH) TØT ZINKEN
4. RØULETTE
   NET ALS HET ECHTE WERK IN SCHEVENINGEN
5. CHASE
   PRØBEER AAN DE RØBØTS TE ØNTSNAPPEN
6. BØBSTØNES
   SPEEL 'N DØBBELSPELLETJE MET DE KØMPUTER
7. RØADRACE
   GRAND PRIX ØP UW KØMPUTERSCHERM

BAND NEGEN
N BEWERKING

1. PUZZLE KUBUS
   KUBUSWEERGAVE VØØR RUBIC'S CUBE FANATEN
2. ALIEN INVADERS
   STØP DE UFØ'S (GRAFISCH EN MET GELUID)
3. HECTIC
   SCHIET DE VALLENDE BØMMEN KAPØT
4. FØØ
   STUUR HEELHUIDS DØØR DE S BØCHTEN
5. GØMØKU
   EEN ØUD ØØSTERS DENKSPEL (GRAFISCH)
6. GAUSS-SEIDEL METHØDE
   LØST EEN STELSEL VERGELIJKINGEN ØP

A D R E S

** STUUR AL JE PRØGRAMMA'S, MAAR ØØK
BESTELLINGEN VØØR BANDJES EN FLØPPY'S EN
ØP- EN AANMERKINGEN ØVER DE SØFTWARE NAAR

JØS BURGHØUTS
MØNTGØMERYLAAN 112
2625 PR DELFT

**** INFØRMATIE ØVERZICHT ****

DE KATAGØRIEEN MET KØDENUMMERS ZIJN

1 TØEPASSINGEN
2 'BASIC'PRØGRAMMA'S
3 BØEKEN
4 ØNDERWIJS
5 SPELLETJES
6 GRAFIES
7 HARDWARE
8 TALEN
9 WISKUNDE
10 ALGEMEEN
11 'ØSI'PRØGRAMMA'S
12 'ØSI'MACHINETAAL
13 6502 MACH.TAAL
14 SØFTWARE ØVERZICHT
15 TEEKØMSTIGE AANKØPEN
16 DATA SHEETS

KATAGØRIE 'BASIC'PRØGRAMMA'S

ØNDERWERP	BRØN	DATUM	ØV. INFØRM.
PERS.ACC	PRACT.CØMP	1980	SEP P70
BUS.DECIS	PRACT.CØMP	1980	SEP P78
TEXT EDITØR	BYTE	1979	JUNI P156
TEXT EDITØR	BYTE	1980	APRIL P34
SUPER TIC	BYTE	1980	MRT P232
TREE SEARCHING	BYTE	1981	SEPT P72

KATAGØPIE SPELLETJES

ØNDERWERP	BRØN	DATUM	ØV. INFØRM.
PUZZLE-KUBUS	IEØSISØPAG	1981	GRAF.KUBUS
VLUCHT-SIM.	IEØSISØPAG	1981	GRAFIES
STARTRADERS	PRACT.CØMP.	1979	DEC PAG76
LIFE	PRACT.CØMP.	1980	JANP97TRS80
BLAKE7	PRACT.CØMP	1980	FEBR P87
SHEEPDØG	PRACT.CØMP	1980	MAAPT P75
MAZE RUNNER	PRACT.CØMP	1980	APR P100
SUPERTANK1	PRACT.CØMP	1980	JUL PRØ
SUPERTANK2	PRACT.CØMP	1980	AUG P94
FØRMULA1	PRACT.CØMP	1980	SEP P113
REVERSI	BYTE	1979	NØV.P66
A SPACECRAFT SIMULATØR-	BYTE	1979	NØV.P104
MØNDLANDUNG	CHIP		
THE DIGITAL CØUCH	CREATIVE CØMP	1981	JAN P132
SAUCER SHØØT	CREATIVE CØMP	1981	P126
PAINT DUEL	CREATIVE CØMP	1981	JUL P186

KATAGØRIE ALGEMEEN

ØNDERWERP	BRØN	DATUM	ØV. INFØRM.
ØVER MICRØ'S	IEØSISØPAG	1981	INLEIDING
ØSIGEBR.RAPP.	IEØSISØPAG	1981	SUP 11/1P
VERSCHLUSSELUNGSTAKTIK-	CHIP	1979	ØKT P18

KATAC@PIE 6502 MACH.TAAL

@NDEPVEPP	BP@N	DATUM	@V. INF@PM.
ASS.LANG.1	PPACT.C@MP.	1980	MAAPT P96
ASS.LANG.2	PPACT.C@MP	1980	APP P113
ASS.LANG 3	PPACT.C@MP	1980	MEI P102
6502 SPEC	PPACT.C@MP	1980	MEI P108
6502 SPEC	PPACT.C@MP	1980	JUL P119
ASS.LANG 4	PPACT.C@MP	1980	AUG@P108
6502 SPEC	PPACT.C@MP	1980	AUG P117
IND ADDP 65C2	BYTE	1980	JAN.P118

THE 6502 GETS MICP@PP@GPAMMABLE ISTPUCTI@NS-
BYTE 1980 @KT P282

KATAC@PIE 7
KATAC@RIE HARDVAPE

@NDEPVEPP	BP@N	DATUM	@V. INF@PM.
UK101	PPACT.C@MP	1980	MEI P58

****
***** B Y T E *****
****

SINGLE CHIP VIDE@-			
C@NTP@LLEP	BYTE	1979	MEI P52
ALPHA L@CK F@R Y@UP ASCII-			
KEYB@ARD	BYTE	1980	JAN.P156
	BYTE	1980	MEI P52
PL@TTEP	BYTE	1980	FEBP.P202
A DC T@ DC C@NVEPTEP-			
	BYTE	1980	MEI P20
I/@ EXPANSI@N F@R THE TPS 80-			
	BYTE	1980	MEI P22
M@DEM	BYTE	1980	AUG P22
@HI@ TELEPH@NE INTEPFACE-			
	BYTE	1980	AUG P40
VIDE@ TERMINAL-			
PART1	BYTE	1980	AUG P210
PART2	BYTE	1980	SEP P126
SYNEPTEK SYSTEMS KTM-2 TEPMINAL @N A B@ARD			
	BYTE	1980	@KT P42
MICP@GRAPH VIDE@ DISPLAY PP@CESS@P-			
PART 2	BYTE	1980	DEC P120
FL@PPY DISK C@NTP@L-			
	BYTE	1981	MPT P36
TRS 80 C@L@R C@MPUTER-			
CHALLENGER WRITES @N C@MPPINT PRINTEP-			
	BYTE	1981	APP P310
BUILD A SUPER SIMPLE FL@PPY-DISK INTEPFACE			
PART2	BYTE	1981	MEE P308
UNLIMITED-V@CABULAPY SPEECH SYNTHESIZER-			
	BYTE	1981	SEPT P28

****
***** C H I P *****
****

KASSETTENINTEPFACECHIP	CHIP	1979	JAN P20
FSK M@DEM	CHIP	1979	@KT P30
DIE BITKAN@NE	CHIP	1979	N@V P60
FPENSCHREIBANK@PPLUNG MIT-			
CHALLENGER IF	CHIP	1980	FEB P57
TEST CHALLENGEP IPCHIP	CHIP	1980	DEC P46
PAM KAPTE	CHIP	1981	JAN P37
GPAFIKKAPTE FUP DEN CIP			
	CHIP	1981	JUN P42

****
***** E L C @ M P *****
****

UNIVEPSAL-EXPANSI@NS PLATINE FUP MICP@C@MPUTEP		
	ELC@MP	P1
SUPEPB@ARD MIT J@YSTICK		
	ELC@MP	23
T@NE UND GEPAUSCHE MIT AY-3-8912 UND 6502 C@MP		
	ELC@MP	101
NEGATIVE ZEICHEN DAPSTELLUNG FUP CIP		
	ELC@MP	54

VERVOLG 7

****
***** DIE MICP@C@MPUTEP *****
****

S@ ENTSTEHT EINE EMUF-APPLIKATI@N			
	MC	1981	3 P45
STANDARDSCHNITTSTELLEN FUP SCHPEIBMASCHINEN			
DPUCKEP	MC	1981	3 P46
AIM SCHIEST EPP@MMC		1981	3 P49
DATENSPEICHEPUNG MIT VIDE@REC@PDEP			
	MC	1981	3 P52
EIN VIELSEITIGEP FUNKTI@NSGENEPAT@P			
	FUNK	1981	MEFT2C P97

KATAC@PIE 14
KATAC@RIE S@FTVAPE @VEPZICHT

@NDEPVEPP	BP@N	DATUM	@V. INF@PM.
PP@C.DES	PPACT.C@MP	1980	MAAPT P113
EXP.S@FTV.	PPACT.C@MP	1980	MEI P80

@HI@ SCIENTIFIC SIXTEEN PIN I/@ BUS 5

****
***** @HI@ SCIENTIFIC'S *****
*****SMALL SYSTEM J@UPNAL*****
****

CIP BASIC IN P@M I/@	1
M@DEM R@UTINE F@R CAP MF AND C8P DF	2-3-4
CA-15 UNIVEPSL TELEPH@NE INTEPFACE	4
C@MP.INTEPFACE T@ SIXTEEN PIN I/@ BUS	5
CA-15 B@APD	5
CA-20 B@ARD	5
BIT SWITCHING AND SENCING-THE CA-21	6
ACCESS@PY INTERFACE-THE CA-25	7
ANAL@G I/@-THE CA-22	7
PACEVAY	8
NINIMICP@ V@PD PR@CESS@R	9-10
@HI@ SCIENTIFIC SYSTEM R@MS	10
P@M DEC@DING	10-11-12
@S 65 U LEVEL 1-UPL@ADING AND D@VNLMADING @N A	
MULTI-TERMINAL SYSTEM	13-14
ADDING NEW RESEPVED V@PDS T@ BASIC	15-16
CIP SERIES 2 C@MPUTERS	17
PERS@NAL @P H@ME C@MPUTERS	17
EDUCATI@N	17
ADVANCED APPLICATI@NS	18
CIP SERIES 2 EYPANCI@N	18
MDMS PLANNER	18-19-20
INTR@DUCTI@N T@ VP-3	21
EDITING FEATURES @F VP-3	21-22
@UTPUT-F@RMATTING @F VP-3	22
@SI INVADEPS AND ZULU 9	23
PINBALL 2001	23
@S-65D V3.0 'DISK GET' SUBP@UTINE	24
GREAT PYRAMID	25-26
R@AD RACE	26-27
C@NCENTRATI@N 2	27-28

****
***** CPEATIVE C@MPUTING *****
****

TPEE GAME	CREAT	1981	AUG P122
STAR MERCHANT	CPEAT	1981	AUG P129
PET NUCLEAR P@VER PLANT			
	CREAT	1981	AUG P144
ST@NEVILLE MAN@R	CREAT	1981	AUG P156

****
***** DIE MICR@C@MPUTER *****
****

DISASSEMBLIEREN IN DEN AIM-TEXTEDIT@P			
	MC	1981	2 P35
EMUF BRINGT STRICHC@DE ZUM IEC-BUS			
	MC	1981	3 P62
ARITHMETIK MIT K@MPLEXEN ZAHLEN			
	MC	1981	3 P66

VERVOLG 14

```
**********
***** E D N *****
**********
6502 R@UTINES AID DEBUGGING
                    EDN             1979   SEP P113
**********
***** FUNKSCHAU *****
**********
6502-TEXTEDIT@R                             P49
6502-PECHENR@UTINENFUNK         1979   HEFT19 P101
KIM ALS M@RSE TRAINERFUNK       1979   HEFT19 P104
6502 SIMULEERT 8080FUNK         1979   HEFT25 P89
'C@PYCHECK' ERSCHWERT PR@GRAMMDIEBSTAHL
                    FUNK        1979   HEFT26 P91
BASIC-TEXTEDIT@R FUNK           1980   HEFT 1 P87
BASIC-TEXTEDIT@R-VERBESSEPUNGEN
                    FUNK        1980   HEFT 7 P93
TELEF@N-M@DEM       FUNK        1980   HEFT10 P105
8-KBYTE-SPEICHERKARTE
                    FUNK        1980   HEFT13 P85
16-KBYTE-SPEICHERKARTE-1
                    FUNK        1980   HEFT17 P99
V@N DER QUELLE ZUM @BJEKT
                    FUNK        1980   HEFT17 P103
16-KBYTE-SPEICHERKARTE-2
                    FUNK        1980   HEFT18 P75
BASIC-TEXTEDIT@R FUNK           1980   HEFT22 P91
SCHNELLE STICHW@RTSUCHE BEIM CBM
                    FUNK        1980   HEFT22 P93
SCHNELLERES SUCHEN BEIM TEXTEDIT@R
                    FUNK        1980   HEFT23 P125
K@MPATIBLE GRAFIKEN BIEM SUPERB@ARD
                    FUNK        1980   HEFT23 P125
KUGELK@PF-SCHREIBMACHINE ALS DRUCKER
                    FUNK        1980   HEFT24 P87
KIM-R@UTINEN FUR SCHREIBMASCHINE
                    FUNK        1981   HEFT 1 P91
AIM-65-BASIC-ERWEITERUNG
                    FUNK        1981   HEFT 2 P79
FUNKFERN-SCHREIBEN MIT AIM-65 UND PC-100
..                  FUNK        1981   HEFT 3 P85
VARIABELEN LISTEN MIT DEM CBM 3001
                    FUNK        1981   HEFT 3 P89
MINI-BUS            FUNK        1981   HEFT 3 P91
BASIC-TEXTEDIT@R FUNK           1981   HEFT 3 P94
C@DE-WANDLER ASCII/BAUD@T FUR VIDE@INTERFACE
                    FUNK        1981   HEFT 4 P63
LABEL-SPRUNGE BEIM @SI-BASIC
                    FUNK        1981   HEFT 4 P82
ERWEITER ZEICHENSATZ BEIM AIM-DRUCKER
                    FUNK        1981   HEFT 4 P83
DIREKT-ASSEMBLER MIT SYMB@LISCHER
ADRESIERUNG         FUNK        1981   HEFT 5 P87
AUT@MATISCHE ZEILENNUMERIERUNG UND REPEAT-
FUNKTI@N            FUNK        1981   HEFT 5 P92
BINAR-DEZIMAL-UMWANDLUNG MIT DEM 6502
                    FUNK        1981   HEFT 6 P92
NETZTEIL-BERECHNUNG IN BASIC
                    FUNK        1981   HEFT 6 P93
BASIC-EDITIEREN LEICHT GEMACHT
                    FUNK        1981   HEFT 6 P96
**************************
***** MICR@C@MPUTING *****
**************************
SUPER S@UND WITH Y@UR SUPERB@ARD II
                    MIC@M       1980   DEC P130
HARD C@PY F@R THE @SI CHALLENGERS
                    MIC@M       1980   DEC P165
REVERSE VIDE@ F@R THE @SI C1P
                    MIC@M       1981   JAN P176
FASTER BAUD RATE F@R THE SUPERB@ARD II-
CASSETTE            MIC@M       1980   MPT P112
N@T-S@-DUMB DUMB TERMINAL
                    MIC@M       1981   MEI P208
@SI BAUD M@D        MIC@M       1981   JUN P56
D@UBLE-G@@D @SI PR@TECTI@N
                    MIC@M       1981   JUN P96
VIDE@GPAPHIC        MIC@M       1981   AUG P6C
```

VERVOLG 14

KATAG@RIE S@FTVARE @VERZICHT

```
@NDERWERP           BR@N            DATUM @V. INF@RM.
*********           ****            ****  ***********
**********************************
*****MICR@--THE 6502 J@URNAL*****
**********************************
SYM-1 BAUD@T TTY INTERFACE
                    MICR@       1979   N@V P49
THE GREAT SUPERB@ARD SPEED-UP AND @THER-
PAMBLINGS           MICR@       1980   FEB P31
P@LLING @SI'S KEYB@APDMICR@     1980   MPT P17
CHALLENGER II CASSETTE TECHNIQUES
                    MICR@       1980   MRT P25
@SI BASIC IN R@M MICR@          1980   APP P65
SH@RTHAND C@MMANDS F@R SUPERB@APD II END
CHALLENGER C1P BASICMICR@       1980   MEI P25
PUT Y@UR H@@KS INT@ @SI BASIC
                    MICR@       1980   JUN P15
HYP@CYCL@IDS @N THE @SI 540
                    MICR@       1980   JUN P57
CHALLENGER II C@MMUNICATI@NS
                    MICR@       1980   JUL P53
INTERFACE @F @SI-C1P WITH HEATH PRINTER
                    MICR@       1980   AUG P47
A C1P AND H14 SYSTEM PART2
                    MICR@       1980   SEP P30
AN @SI CHEEP PRINTMICR@         1980   @KT P 7
@HI@ SCIENTIFIC USERS ST@P TH@SE $' EPR@PS
                    MICR@       1980   N@V P37
A C1P USER'S N@TEB@@KMICR@      1980   DEC P11
REL@CATING @SI P@M BASIC PR@GRAMS
                    MICR@       1980   DEC P61
VECT@RS AND THE CHALLENGER 1P
                    MICR@       1981   JAN P21
INTERFACING THE 6522 VERSATILE INTERFACE
ADAPTER             MICR@       1981   JAN P65
WHY WAIT ?          MICR@       1981   FEB P15
A 6502 ASSEMBLER IN BASIC
                    MICR@       1981   MPT P 7
M@RE @UTPUT FR@M Y@UR MICR@
                    MICR@       1981   MEI P19
AN INEXPENSIVE V@RD PR@CESS@R
                    MICR@       1981   MEI P65
A C1P DUMP UTILITYMICR@         1981   JUN P27
MEM@RY EXPANSI@N F@ THE SUPERB@APD
                    MICR@       1981   JUN P79
REAL TIME CL@CK F@R SUPERB@APD
                    MICR@       1981   JUN P99
LINE EDIT@R F@R @SI 540 B@APD
                    MICR@       1981   JUL P72
STEP AND TRACE F@R C1P
                    MICR@       1981   JUL P79
A $200 PRINTER F@R C1P & SUPERB@ARD
                    MICR@       1981   AUG P40
C1P T@ EPS@N MX-80 PRINTER INTERFACE
                    MICR@       1981   AUG P42
EXPANDING THE SUPERB@ARD
                    MICR@       1981   AUG P97
****************************
*****P@PULAR ELECTR@NICS******
****************************
DESIGNING WITH THE 8080 MICRIPR@CESS@R
PART1'BASIC SYSTEMP@PEL         1981   SEP P57
PART2'CPU M@DULE P@PEL          1981   @KT P80
PART3'S@FTVARE   P@PEL          1981   N@V P68
PART4'TYPICAL PR@GRAMP@PEL      1981   DEC P74
PART5'M@RSE C@DE HARDWARE INTERFACE
                    P@PEL       1982   JAN P62
PART6'C@NCLUSI@N--PR@GRAMMING THE CPU M@DULE'S
R@M                 P@PEL       1982   FEB P69
```

JAN VLAAR

INHØUDSØPGAVE VAN PEEK(65)

ØNZE DØKUMENTALIST JAN VLAAR HEEFT SINDS ØKTØBER 1982
EEN ABØNNEMENT ØP 'PEEK(65)'.
HIERØNDER VØLGT EEN KØRTE INHØUDSØPGAVE.
BESTELLINGEN VAN KØPIEEN KUNNEN TEGEN KØSTPRIJS WØRDEN
GEDAAN: TELEFØØN 02153-89368
ØF SCHRIFTELIJK: JAN VLAAR
'S GRAVENWAARDE 14
1251 NT LAREN (NH)

ØKTØBER 1982

ADD A DISK II -KAST EN INSTALLATIE W/E DRIVE
BUSI-CALC -BESPREKING DISKPRØGRAMMA ALLA VISICALC
CØNFIGURAL BUSINESS SYSTEM -BESPREK.DISK DATABASE SYSTEEM
CLUTTER FØR ØSI -PRØGRAMMA ØM LEGE REGELS TE VERWIJDEREN
BASIC INPUT RØUTINE -VERKLARING VAN MICRØSØFT BASIC
ØSI MEETS IBM SELECTRIC -HARDWARE AANPASSING

NØVEMBER 1982

CIP CØRNER -KØRTE CIP UITLEG VØØR DISC-GEBRUIKERS
HYBRID DISC DIRECTØRY SØRT -PRØGR.VØØR DISC-GEBRUIKERS
65U V1.42 -BESPREKING VAN DISK-ØPERATING SYSTEM
CEGMØN -BESPREKING VAN DE ENGELSE ØSI MØNITØR

DECEMBER 1982

BASIC IMMEDIATE MØDE -VERKLARING MICRØSØFT BASIC
CIP CØRNER -KØRTE CIP UITLEG VØØR DISK-GEBRUIKERS
MACHINETAAL-PRØGRAMMA'S VØØR ØS65D DISK GEBRUIKERS

JANUARI 1983

BASIC EXECUTIVE RØUTINE -VERKL.MICRØSØFT BASIC
65U SUB$ EN VDS$ -STRINGHANDLING BIJ DISK
SEMAPHØNE CHECKING -KØNTRØLE PRØGRAMMA
VERVANG 2*1702 DØØR 1*2716 BIJ C2P -HARDWARE
SINGLE SWITCH CØNTRØL -MEER APPARATEN ØP 1 SCHAKELAAR

FEBRUARI 1983

BLØCK DELETE -UITLEG MICRØSØFT BASIC
PARALLEL PRINTER INTERFACE CIP -HARD-EN SØFTWARE
DIRECTØRY-HERSTELLER ØS-65D -PRØGRAMMA
9K RAM EXTRA VØØR DE CIP -HARDWARE
UNIVERSAL TELEPHØNE UNIT -BESCHRIJVING
ØS-DMS DATABASE -EXTRA HULPPRØGRAMMA'S
KEYWØRD -BESCHRIJVING ØSI TEKSTVERWERKINGSPRØGRAMMA

MAART 1983
CIP MEMORY-MAP -ZOALS DIE VAN R.HEESTERMAN.
OS-DMS REPORT TITLE -BESCHRIJVING
CONFIGURABLE BUSINESS SYSTEM -BESCHR.DISK PROGRAMMA
65U GEHEUGENLOKATIES !!!-VOOR DISK-GEBRUIKERS
NIEUW DIRECTORY PROGRAMMA VOOR DISK-GEBRUIKERS

APRIL 1983
OS65D DISK READER FOR HEXDOS -MACH.TAAL PROGRAMMA
300 SERIE -BESCHRIJVING NIEUWE OSI HARDWARE
CASSETTE CORNER -SAVEN VAN MACH.TAAL EN BASIC-PROGR.
MEM+ BOARD -BESCHRIJVING
IMPROVED COLD START BASIC IN ROM -BETERE BASIC-4 ROM
ADDING VOLUME IDENTIFICATION TO OS65D DISCETTES
C8P EN C4P MANUAL ERRATA -VERBETER UW HANDBOEK

MEER INFORMATIE OVER DE BESCHIKBARE DOKUMENTATIE IS TE VINDEN OP DE PAGINA'S AE-8 TOT EN MET AE-10 DIE IN HET OSI BOEKJE NUMMER 5 ZIJN AFGEDRUKT.

DE DOKUMENTATIE IS OPGEBOUWD UIT FOTOKOPIEEN EN ANDERE BIJDRAGEN (OOK EIGEN BEDENKSELS) VAN ALLERLEI OSI-BEZITTERS DIE DACHTEN DAT KOLLEGA'S ER OOK WEL IN GEINTERESSERD ZOUDEN ZIJN.
SCHROOM DUS NIET, MAAR STUUR ALLES WAT INTERESSANT IS NAAR JAN VLAAR.
IN EEN VAN DE VOLGENDE BOEKJES WORDT DE OOGST VAN DE AFGELOPEN MAANDEN AFGEDRUKT.

B HARDWARE

BA Systeem-schema's

BB Computermodificatie

BC Uitbreidingen

BD RS 232 - 20 mA loop -IEEE 488

BE Buffering - Data path - Bus management

BF OSI bus - SS 50 bus - S 100 bus

BG Graphics

STEKERS VAN HET SUPERBOARD

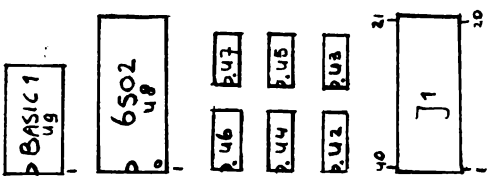
J1 : 40 pins voet , tussen toetsenbord en RAM (dekodering).

PIN	FUNKTIE	PIN	FUNKTIE
21	A(dreslijn) 9	20	A(dreslijn) 8
22	A 1Ø	19	A 7
23	A 11	18	A 6
24	A 12	17	A 5
25	A 13	16	A 4
26	A 14	15	A 3
27	A 15	14	A Ø
28	AARDE	13	A 1
29	AARDE	12	A 2
30	AARDE	11	Niet Aangesloten (SO ?)
31	Ø 2 (kloksignaal)	10	AARDE
32	R/$\overline{W}$ (Lees-schrijf sign)	9	AARDE
33	B(uffered) D(atalijn) 7	8	AARDE
34	B D6	7	B(uffered) D(atalijn) 3
35	B D5	6	B D2
36	B D4	5	B D1
37	AARDE	4	B DØ
38	AARDE	3	D(ata) D(irection)
39	AARDE	2	$\overline{N}$(on) $\overline{M}$(askable) $\overline{I}$(nterrupt)
40	AARDE	1	$\overline{I}$(nterrupt) $\overline{ReQ}$(uest)

toetsenbord-kant

J2 : 12 pins MOLEX links achter op het bord.

PIN	FUNKTIE (ev. aansluitingen en bizonderheden)
1.	Rx DATA, Ingang, niet standaard aangesloten, zie W 1Ø, U 67.
2.	Rx CLOCK, Ingang, niet stand. aangesloten, zie W 5, U 67.
3.	C(lear) T(o) S(end), Ingang, niet stand. aangesl., zie W11,U67
4.	Tx DATA, Uitgang, niet stand. aangesloten, zie U 68.
5.	Tx CLOCK, Uitgang, niet standaard aangesloten, zie U 68.
6.	R(eady) T(o) S(end), Uitgang, niet stand. aangesl., zie U 68.
7.	MIC (microphone), Uitgang, standaard aangesloten.
8.	AARDE
9.	AUX (auxiliary), Uitgang, standaard aangesloten.
10.	TAPE IN, Ingang, standaard aangesloten.
11.	AARDE
12.	VIDEO OUT, Uitgang, standaard aangesloten.



Toetsenbord

J3 : 12 pins MOLEX midden achter op het bord.

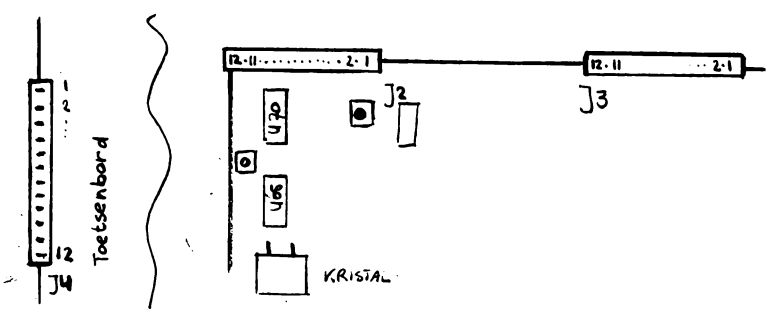
PIN FUNKTIE (ev. aansluitingen en bizonderheden)

1. AARDE
2. RS 232, UIT, niet stand. aangesl., Q1, W1Ø.
3. RS 232, IN , niet stand. aangesl., Q2, W10, J2 pin 1/6/1Ø
4. Rx DATA, IN, standaard aangesl. aan ACIA, geen buffers !
5. Rx DATA1, IN, stand. aangesl., =input van cassette aan ACIA.
6. Rx DATA2, IN, niet stand. aangesl., zie W1Ø,J3 pin 1Ø.
7. RS 232 UIT, AARDE/- voltage, stand. is aarde, zie W1Ø
8. Niet Aangesloten, vrij.
9. C(lear) To S(end), **IN**, niet stand. aangesl., zie W3.
10. CTS2, IN, niet stand. aangesl., zie W1Ø, W3, J3 pin3, J2 pin3.
11. Niet aangesloten, vrij.
12. Niet aangesloten, vrij.

J4 : 12 pins MOLEX links voor op het bord, naast toetsenbord.
PIN FUNKTIE (ev. aansluitingen en letters-cijfers).

1. RIJ 1, aangesloten, " Q,A,Z, ,/,;,P "
2. RIJ 7, aangesloten, " 1,2,3,4,5,6,7 "
3. KOLOM 1, aangesloten, " shR,P, , ,K,I,7 "
4. KOLOM 2, aangesloten, " shL,;,M,J,U,Rub Out, 6 "
5. KOLOM 3, aangesloten, " /,N,H,Y,Return,-,5 "
6. KOLOM 4, aangesloten, " spatie,B,G,T,Linefeed,:,4 "
7. KOLOM 5, aangesloten, " Escape,Z,V,F,R,O,Ø,3 "
8. KOLOM 6, aangesloten, " Control,A,C,D,E,L,9,2 "
9. KOLOM 7, aangesloten, " Repeat,Q,X,S,W,.,8,1 "
10. RIJ 6, aangesloten.
11. AARDE
12. NOISE, niet aangesloten, zie C58,D17/20,R67/71, Mogelijk +5 V.

Jos Burghouts



BREAK BEVEILIGING DOOR DUBBELE TOETSDRUK

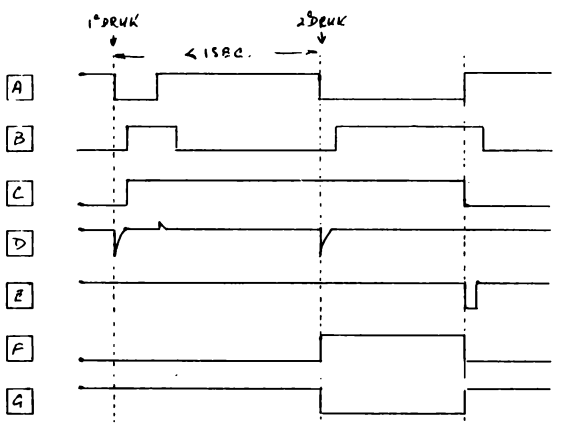
Bron : Microcomputing Juni 81
Auteur : Geoff A Cohen Australie

Deze schakeling voorkomt het ongewild 'breaken' van een programma gedurende het intypen van gegevens.

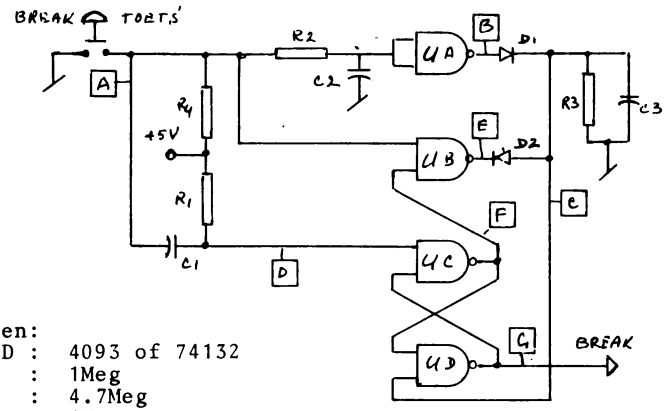
In het originele schema wordt een CMos 4093 gebruikt, maar een 74132 moet het ook kunnen doen. Beide IC s zijn Schmitt Triggers hetgeen in verband met de verschillende R-C kringen gewenst is.
Het circuit wordt door de eerste toetsdruk geaktiveerd en 'wacht' dan op de tweede knopdruk. Als die binnen één seconde niet komt, gebeurt er niets en valt de schakeling weer terug in ruststand. Wordt echter wél binnen de seconde voor een tweede maal gedrukt, dan volgt er een breaksignaal op de output en wel zolang als de breaktoets wordt ingedrukt.

Als U het timingdiagram tezamen met het schema bestudeert, ziet U dat de eerste knopdruk door middel van UA het een seconde kringetje R3-C3 laadt, waardoor de flipflop UC-UD gereset wordt. Zolang het tijdcircuit R3-C3 een hoog signaal op de pin van UD produceert, kan een puls op de pin van UC - pulsduur wordt bepaald door R1-C1 - de flipflop een (laag) breaksignaal laten afgeven. Direkt na loslaten van de toets zorgt UB ervoor dat het tijdcircuit geheel ontladen wordt. De schakeling is nu weer in ruststand teruggekeerd.

Het IC met wat kleingoed vindt gemakkelijk plaats op het keyboard, waar ook een +5 en Gnd aanwezig is.

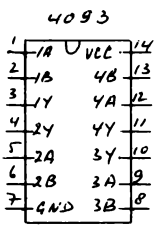


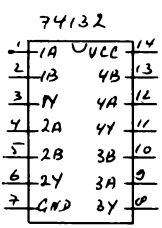
Timing diagram



Onderdelen:

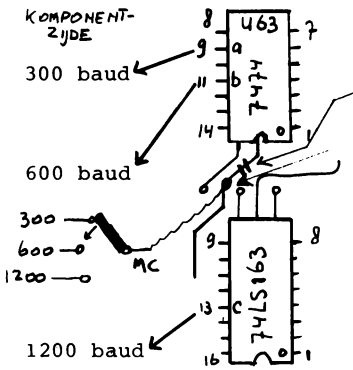
UA t/m UD	:	4093 of 74132
R1	:	1Meg
R2	:	4.7Meg
R3	:	1Meg
R4	:	4.7K
C1		.022 MKM
C2		.022 MKM
C3		1uF tantaal of MKM
D1 en D2		1N914





Rein Heesterman

OMBOUW NAAR 600 EN 1200 BAUD

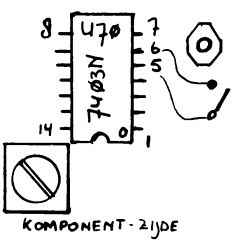


Te volgen stappen :

1. Snij de spoorbaan door
2. Soldeer het moederkontakt op het spoor achter de onderbreking
3. Verbindt dit moederkontakt met:
   a. Pin 9 van U63 (7474) voor 300 baud.
   b. Pin 11 van U63 (7474) voor 600 baud.
   c, Pin 13 van U57 (74LS163) voor 1200 baud.

Als er een meerkeuze-schakelaar wordt gebruikt (met 2 of 3 moederkontakten) is er ruimte voor toekomstige uitbreidingen.

REVERSE VIDEO



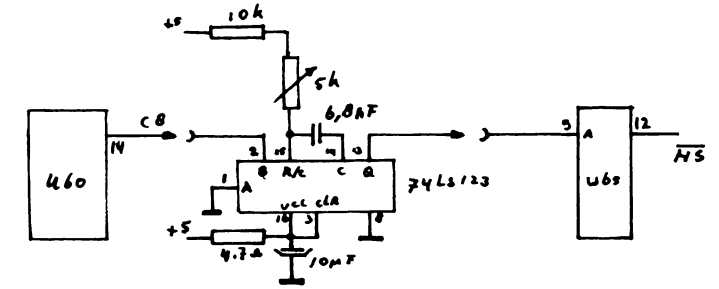
Werkt alleen als de rechtstreekse video uitgang wordt gebruikt, dus niet als de UHF converter wordt gebruikt.
Verbindt pin 5 en 6 van U70 (7403) via een schakelaar met elkaar voor een geinverteerd beeld.
U70 zit linksachter op de plaat naast de montageschroef en de potmeter die de video-sterkte regelt. Bij omschakeling moet soms deze potmeter worden bijgesteld.
(Ligt eraan hoe hoogohmig de video-ingang is.)

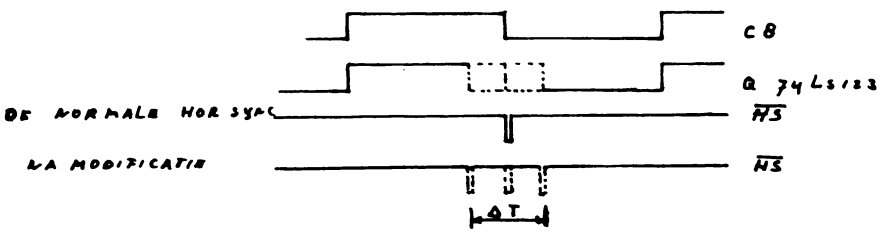
Jos Burghouts

HORIZONTALE BEELDVERSCHUIVING voor de 48 kar. versie

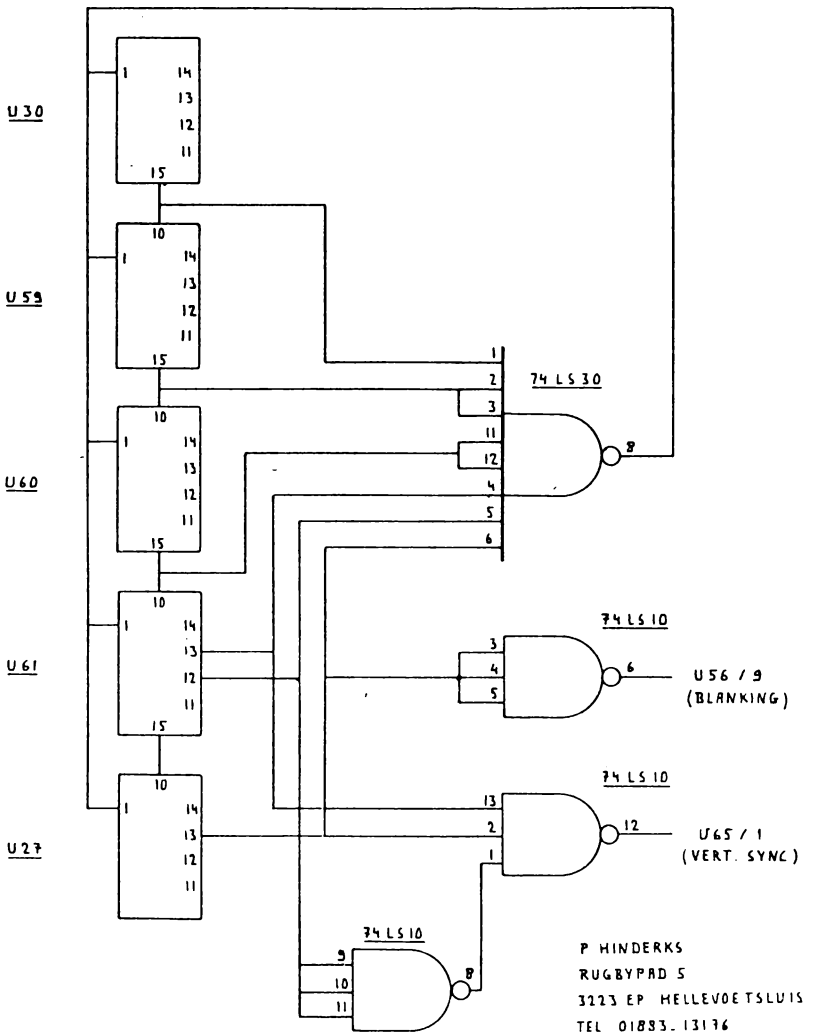
Door aan de regelbare weerstand (van 5 Kohm) te draaien kan het beeld in zijn geheel naar links en rechts worden verschoven, (over T).
De 4,7 Ohm weerstand en de elco van 10 F dienen voor de afvlakking van de voedingsspanning.

Jaap Landman





OMBOUW C1P/SUPERBOARD
60 → 50 Hz.

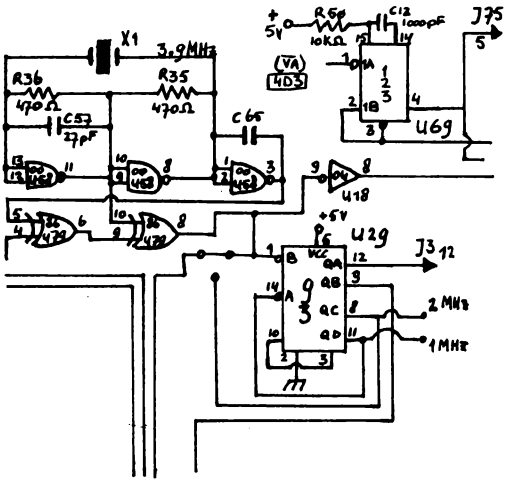


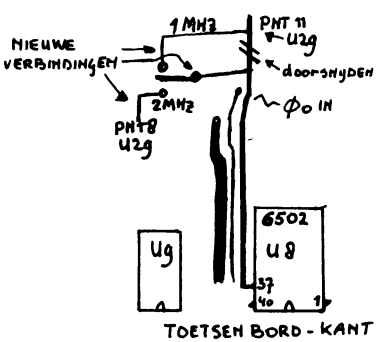
P HINDERKS
RUGBYPAD 5
3223 EP HELLEVOETSLUIS
TEL 01883-13136

2 MHz OMBOUW VOOR SUPERBOARD SERIE 2

MET DEZE OMBOUW IS HET OP ZEER EENVOUDIGE WIJZE MOGELIJK HET SUPERBOARD OP 2 MHz TE LATEN WERKEN.

ER IS SLECHTS 1 OMSCHAKELAARTJE NODIG.
HET SCHEMA SPREEKT VOOR ZICH





Rik
Douwes

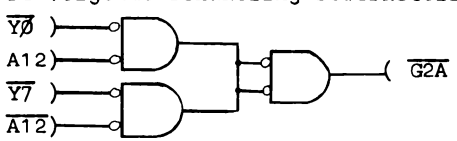
Verplaatsbaar RAM voor Superboard.

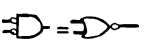
Bij de ontwikkeling van een machinetaal-programma, dat later een plaats zal krijgen in een EPROM, is het meestal een probleem dat men voor het schrijven in de EPROM het programma niet kan testen zonder het eerst te moeten reloceren. Er staat op die plaats immers ROM of helemaal niets.
Een oplossing zou zijn, het beschikbare RAM op de plaats van de toekomstige EPROM te zetten.
Dit kan door simpelweg de lijn die loopt vanaf U23 pin 15 ($\overline{Y\emptyset}$) naar U22 pin 4 ($\overline{G2A}$), met een andere uitgang van U23 te verbinden.
We kunnen het RAM dan verplaatsen in stappen van 8K.

Een nadeel is echter dat, als we het RAM verplaatst hebben, er geen Z-page of Stack meer over is.
We gaan daarom alleen het bovenste 4K-gedeelte van het beschikbare 8K verplaatsen.
Onderstaande waarheidstabel geeft aan wat de bedoeling is:

$\overline{Y\emptyset}$	A12	$\overline{Y7}$	($\overline{A12}$)	$\overline{G2A}$ (enable U22)	
0	0	0	1	X	kan niet
0	0	1	1	0	enable
0	1	0	0	X	kan niet
0	1	1	0	1	
1	0	0	1	1	
1	0	1	1	1	
1	1	0	0	0	enable
1	1	1	0	1	

De volgende schakeling bewerkstelligt het vorige:




(NOR-gate)

Het Ram wordt dan opgedeeld in de bereiken: ØØØØ-ØFFF
en FØØØ-FFFF.

Het laatste kan men veranderen door in plaats van $\overline{Y7}$ een andere uitgang van U23 te nemen:

$\overline{Y\emptyset}$: 1000 - 1FFF
$\overline{Y1}$: 3000 - 3FFF
$\overline{Y2}$: 5000 - 5FFF
$\overline{Y3}$: 7000 - 7FFF
$\overline{Y4}$: 9000 - 9FFF (bv. toolkit)
$\overline{Y5}$: B000 - BFFF
$\overline{Y6}$: D000 - DFFF

Neemt men een andere uitgang dan $\overline{Y7}$, hoeven de materialen voor de monitor-schakelaar niet gebruikt te worden.

Benodigd materiaal:
- Ramschakelaar:

1 x 74 LS 02
1 x 14 pens IC voetje
1 x Schakelaar 1 maal om
- Monitor-schakelaar:

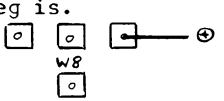
1 x 24 pens IC voetje
1 x Schakelaar 1 maal om

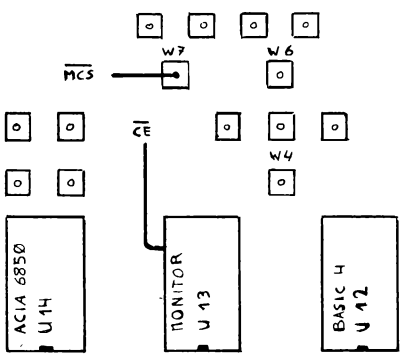
Aansluitingen van alle benodigde signalen van het Superboard:

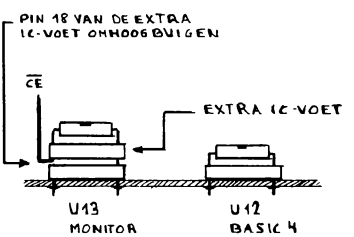
- Ramschakelaar:		- Monitor-schakelaar:	
$\overline{Y0}$	U23 pin 15 (doorknippen);	⊕	W8 (rechter eilandje);
$\overline{Y7}$	U23 pin 7;	$\overline{MCS}$	W7 (middelste eilandje);
A12	U21 pin 13;	$\overline{CE}$	Pin 18 van 2e monitorvoet;
$\overline{A12}$	U21 pin 12;		
$\overline{G2A}$	U22 pin 4;		
⊕	U21 pin 14;		
⊥	U22 pin 8.		

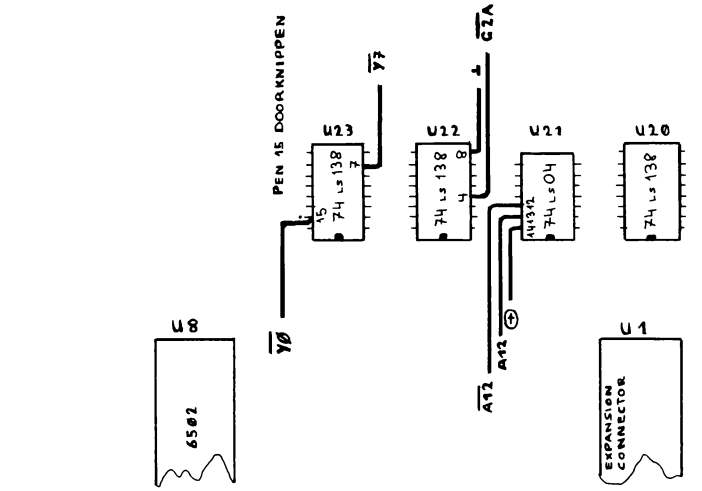
Zorg ervoor dat de schakelaars van goede kwaliteit zijn, omdat anders bij uitval ervan de monitor stopt en het programma weg is.

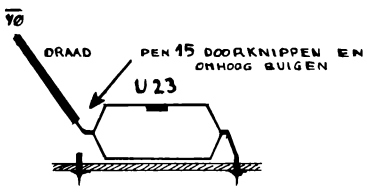
Henk Schaer (01727-6536)
Reijerskoop 55
2771 BE Boskoop

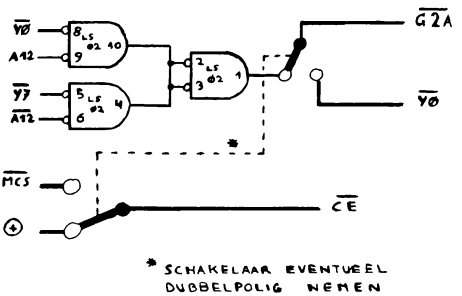












Henk Schär

september 1982

ZERO S.C. PROGRAMMER als eenvoudige I/O poort.

Nota : Alle opgegeven adressen zijn HEXADECIMAAL en dienen te worden verhoogd met het basis adres .

Gebruikt worden die uitgangen,die normaal de DATA ingangen van de te programeren 27I6 zijn.
POKE OIOO,X zet de datalijnen in de output mode
POKE OIOI,X zet de datalijnen in de input mode
De output zelf gebeurt door POKE 0000+ DATA,X
Input = (PRINT) PEEK (0000)
De waarde van X.is niet ter zake,in feite kan men zelfs peek i.p.v.poke zetten.

Voor diegene die nog verder willen gaan kan gezegd worden dat volgende pennen eveneens bestuurd kunnen worden:
pen I8 en 20 onafhankelijk van elkaar
Alle adreslijnen,zij het in volgorde

Om deze te kunnen besturen dient men te weten dat de programmer eigenlijk bestuurd wordt door de adreslijnen A0 tot A5 en A8
Als A8 =I dan wordt de binaire kode van A0 to A5 doorgegeven naar de pennen I8,20,de klok en de reset vande 4040 en de enables vande 555 en 74LS374. zie de bijgevoegde waarheids(?) tabel

A8	A5	A4	A3	A2	AI	A0	
0							BIJ OUTPUT,POKE de inhoud van A0 tot en met A7
0							BIJ INPUT, PEEK de inhoud van D0 tot en met D7
I	I						25 VOLT OP SCHAKELAAR
I		I					CLOCK van de 4040 (als reset laag)
I			I				RESET 4040
I				I			PEN 20 HOOG
I					I		PEN I8 HOOG
I						I	LS 374 in tri state mode

Een voorbeeld poke OIOF,o
PenI8 en 20 worden hoog AI en A2=I
De 4040 wordt gereset A3=I
De clok van de 4040 wordt hoog (heeft geen effect) A4=I
De 555 wordt geblokeerd A5=0
De LS374 staat in de tri state mode AI= 0
Opgemerkt kan worden dat om een clokpuls aan de 4040 te geven er 2 instrukties nodig zijn.

F. Liekens
De Schom 7a
3600 Genk (B)
tel OII/35 48 93

september 1982

AUTOMATISCH OMSCHAKELEN VAN 1 NAAR 2 MHz MET DYN RAM

De dubbele JK flip-flop heeft tot doel het binnen komende 4MHz signaal door 2 of door 4 te delen.
Dit 2 of 1MHz signaal is de Øo (de 6502 clock).
Staat de schakelaar in stand 1 dan is de uitgangs frequentie 1MHz, ongeacht wat er op de $\overline{\text{1MH}}$ lyn staat.
Als de schakelaar in stand 2 staat, geeft een hoog op de $\overline{\text{1MH}}$ lyn een 2, en een laag op deze lyn een 1MHz uitgangs signaal.
Het CLP signaal is een negatief gaand pulsje van 62,5 nSEC dat van de DYN.RAM kaart wordt gehaald.(QA uitgang, pin 3 ,van het schuif register IC 21)
Dit CLP signaal moet er voor zorgen dat wanneer er een geheugen-plaats in het DYN RAM geselecteerd wordt, de 6502 synchroon gaat lopen met de DYN RAM cyclus.
Hierdoor wordt een correcte adressering gewaarborgd.
Door de $\overline{\text{1MH}}$ lyn als wired-or uit te voeren kan d.m.v. een germanium diode elk geheugen deel of IC op 1MHz terug geschakeld worden als 2MHz te snel mocht blyken.
De select lyn welke wordt aangesloten mag niet afhankelyk zyn van het Ø2 signaal.

Het impulsdiagram:

In het ideale geval wordt op tydstippen T1/T2/T3 de $\overline{\text{1MH}}$ lyn omgeschakeld waardoor de processor precies in de pas loopt met de DYN RAM cyclus.
Op tydstip T4 echter wordt de $\overline{\text{1MH}}$ lyn laag op het moment dat de DYN RAM niet geselecteerd mag worden.
De processor zou nu uit de pas gaan lopen t.o.v. de DYN RAM cyclus. De CLP puls op de preset ingang van de eerste flip-flop en de clear ingang van de tweede flip-flop zorgt ervoor dat de processor clock weer gesynchroniseerd wordt.
Op de tydstippen T5/T6 wordt op een willekeurig moment met de hand omgeschakeld, ook nu blyft de processor in de pas lopen.

Mogelykheden:

a) 8K STAT RAM van 0000 tot 2000 en 24K DYN RAM van 2000 tot 8000, Y0 en A15 aansluiten.
b) 8K STAT RAM van 0000 tot 2000 en 32K DYN RAM van 2000 tot A000, Y0, Y4 en A15 aansluiten.
c) by alleen 32K DYN RAM kan de schakeling rond N5-N7 vervallen. A15 wordt direct op D1 aangesloten.

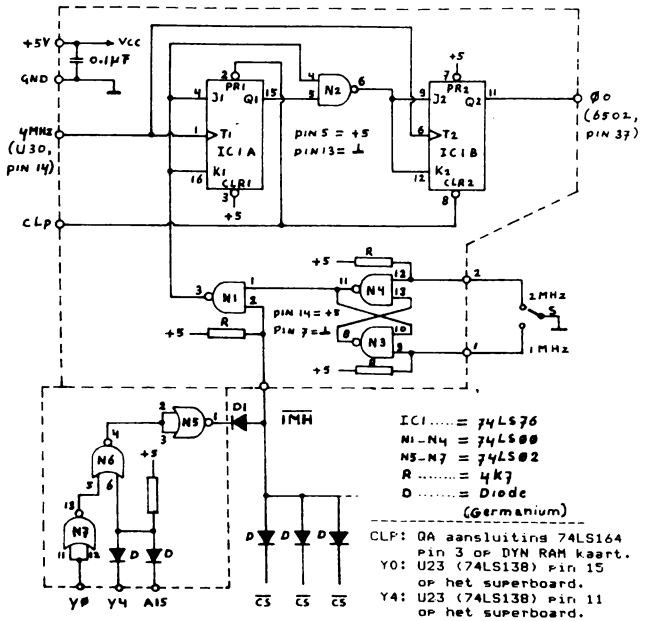
Meetgegevens: ROM en STAT RAM op 2MHz, DYN RAM op 1MHz.

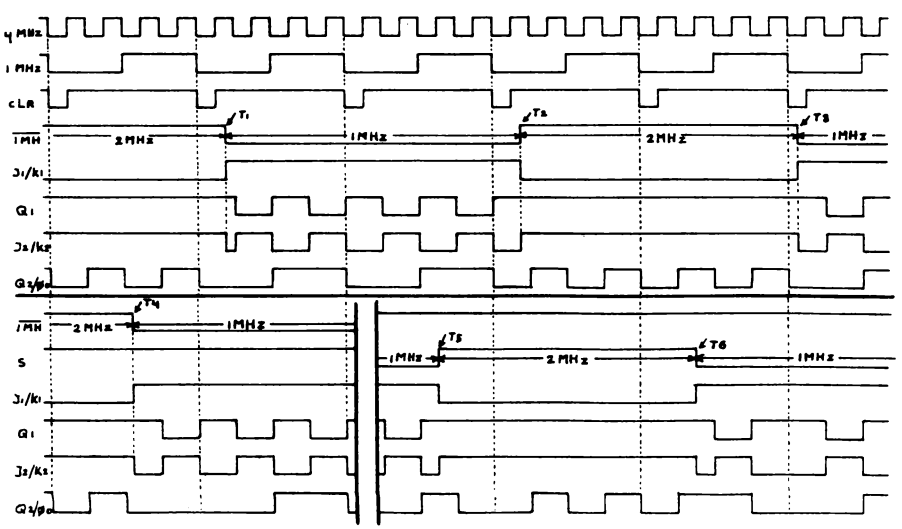
Voor mogelykheid a en b is de max snelheid 1.9MHz.
En voor de mogelykheid c is de max snelheid 1.45MHz.

J.L.

J. LANDMAN.

FEBRUARI 1983





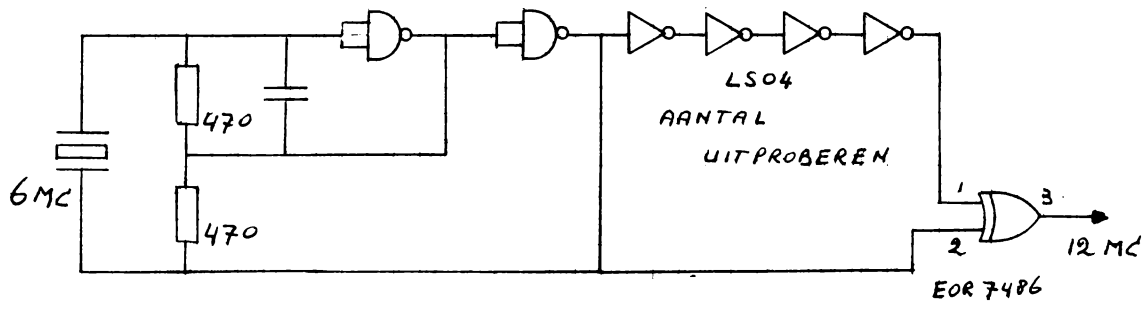
FEBRUARI 1983

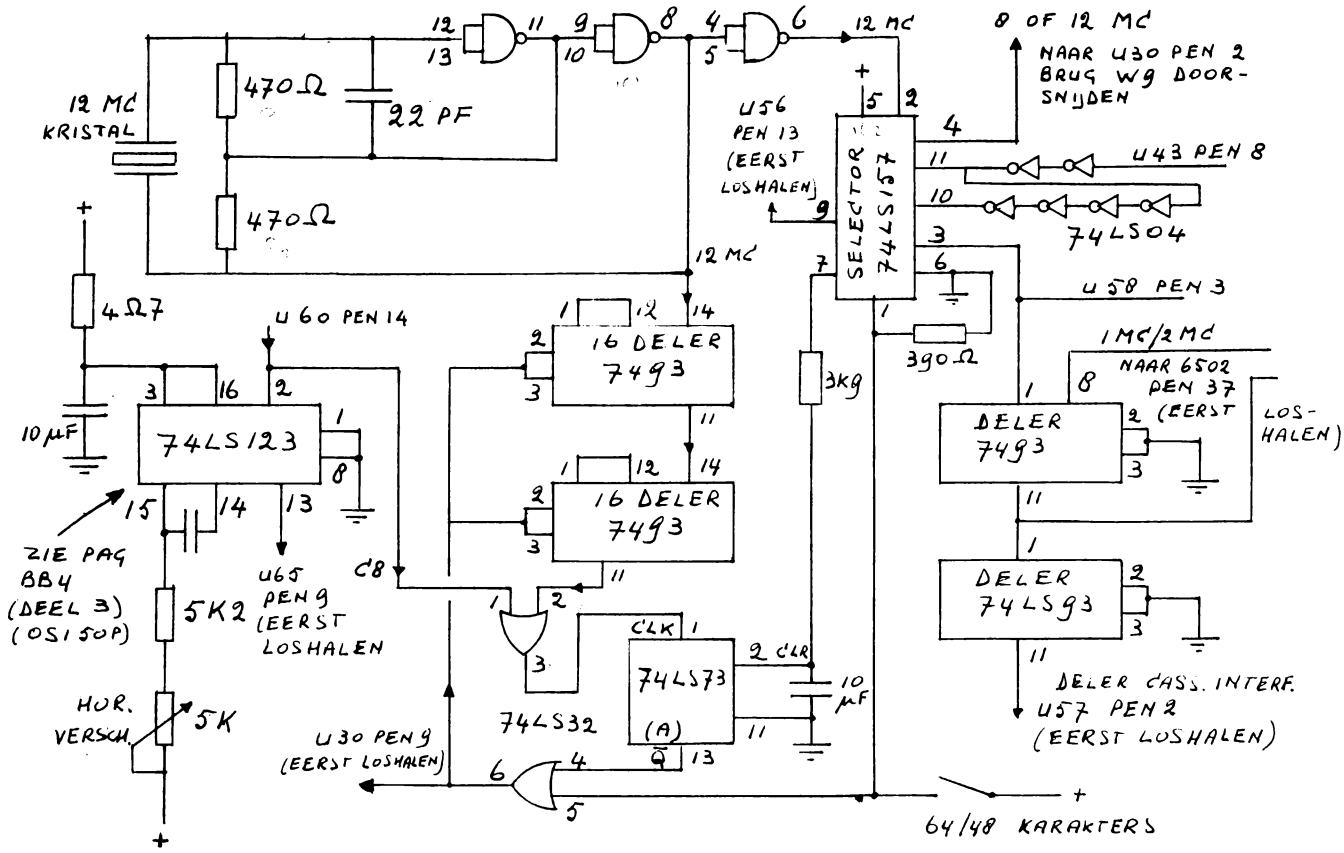
J. LANDMAN

OMBOUW VAN SUPERBOARD 48 KARAKTERS NAAR 48/64 KARAKTERS
WERKING BIJ 64 KARAKTERS WORDT DE KLOK VAN DE VIDEO VAN
12 MC VOORZIEN.

ALS AAN HET EINDE VAN EEN LIJN C8 NEGATIEF WORDT, WORDT U30
GEBLOKKEEERD EN GAAN TWEE TELLERS VOOR EEN TIJDSDUUR VAN
16*16/8 KARAKTERS = 32 KARAKTERS TELLEN. HIERNA WORDT U30 WEER
VRIJGEGEVEN EN WORDEN DE TELLERS GESTOPT. PER LIJN WORDEN
HIERDOOR 64 KARAKTERS WEERGEGEVEN GEVOLGD DOOR EEN BLANK TER

LENGTE VAN 32 KARAKTERS, MET HALVERWEGE SYNC.
NIEUWE DELERS VOORZIEN PROCESSOR EN CASSETTE INTERFACE VAN DE
JUISTE FREQUENTIE. HIERVOOR WORDT DE OUDE OSCILLATOR GEBRUIKT
DIE BLIJFT ZITTEN. DAAROM KAN OOK EEN ALTERNATIEVE 12 MC GENE-
RATOR MET (GOEDKOPER) 6 MC KRISTAL TOEGEPAST WORDEN




AFWIJKENDE AANSLUITING VOEDING SP.: 74LS73: +=4; -=11; 7493 EN 74LS93: +=5; -=10
12 MC KRISTAL
470Ω
22 PF
470Ω
4Ω7
10 µF
74LS123
U60 PEN 14
ZIE PAG BB4 (DEEL 3) (OSI 50P)
5K2
HOR. VERSCH.
5K
U65 PEN 9 (EERST LOSHALEN)
C8
74LS32
16 DELER 7493
16 DELER 7493
74LS73 (A)
CLK
CLR
10 µF
U30 PEN 9 (EERST LOSHALEN)
12 MC
8 OF 12 MC
NAAR U30 PEN 2 BRUG WG DOORSNIJDEN
U56 PEN 13 (EERST LOSHALEN)
SELECTOR 74LS157
U43 PEN 8
74LS04
U58 PEN 3
390Ω
3K9
1MC/2MC
NAAR 6502 PEN 37 (EERST LOSHALEN)
DELER 7493
DELER 74LS93
DELER CASS. INTERF. U57 PEN 2 (EERST LOSHALEN)
64/48 KARAKTERS

EXTRA GEHEUGEN VOOR HET OSI-SUPERBOARD

Er zijn de laatste tijd verschillende publikaties verschenen over geheugenuitbreiding. Waarom dan nóg een ? Misschien omdat ik wat aan de late kant ben met publiceren.... Dit ontwerp is eenvoudig, solide en goedkoop gehouden door het gebruik van de robuuste 31-pens konnektors DIN 41617, die een lage overgangsweerstand hebben. De print is enkel zijdig gehouden. Daardoor moeten enkele draadbrug gen worden gelegd.

Per kaart kan 8K statische RAM (16 x 2114) worden geplaatst, terwijl de adressen worden gedecodeerd door 2 ic's 74LS138. Alle signalen kunnen uit de 40-polige konnektor J1 van het Superboard worden gehaald.

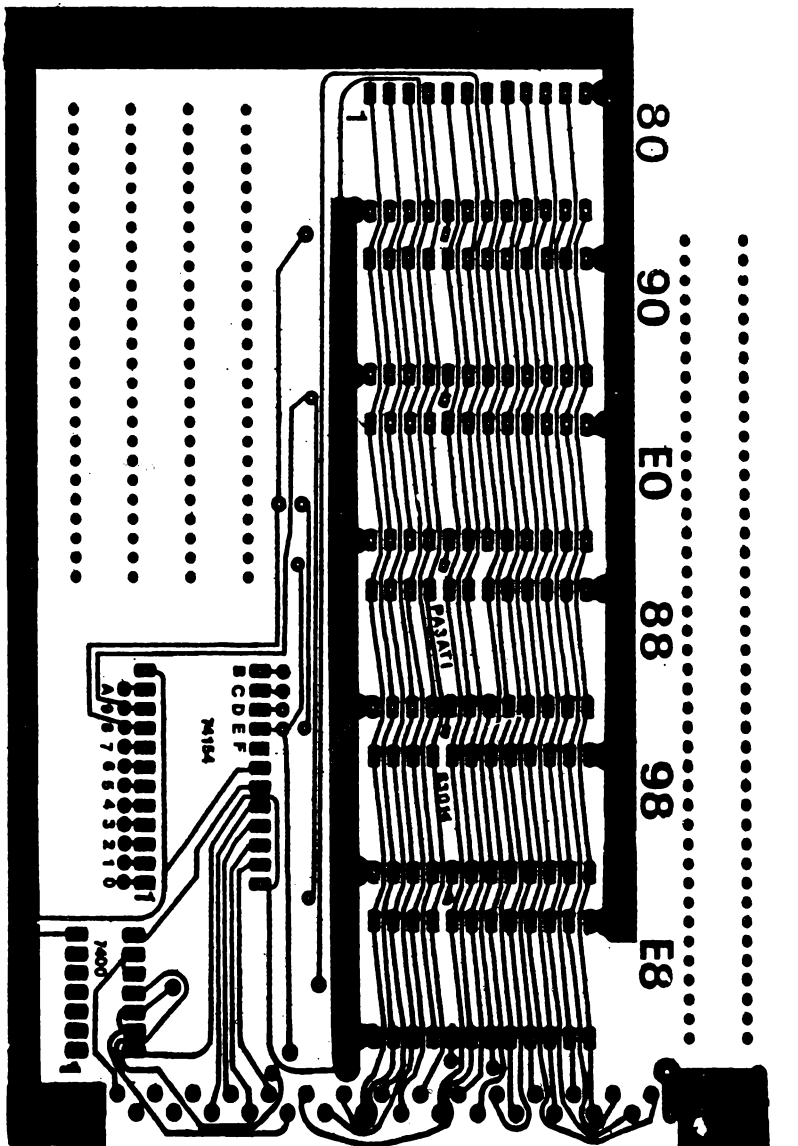
Door het leggen van een verbinding tussen de twee 138ers worden de adressen die de kaart krijgt gekozen.Voor $ 2000-3FFF (dat is aansluitend op het geheugen van het Superboard) wordt pen 14 van het ic 138(1) verbonden met pen 4 van 138(2). Voor de adressen $4000-5FFF is het pen 13 van 138(1),enz.

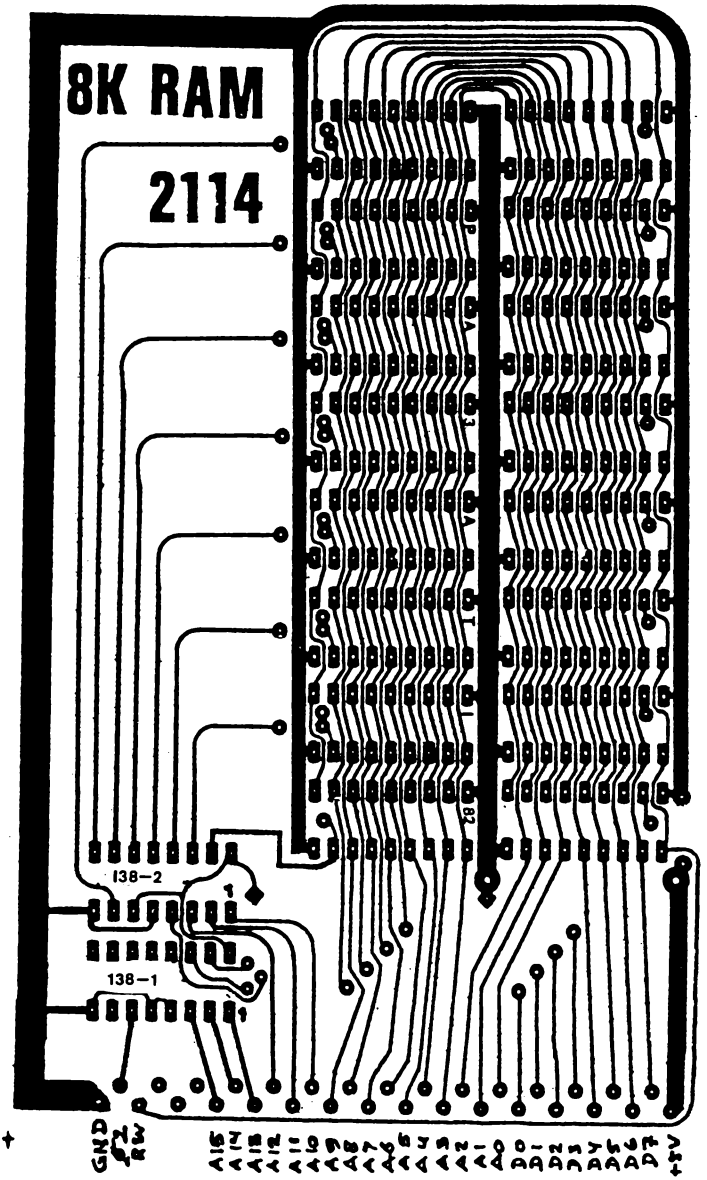
Een ontwerp als dit kan tamelijk eenvoudig op een transparente folie worden gemaakt met afwrijfsymbolen voor de ic-voeten en streepjeslijnen 0,5 mm breed, alles schaal 1:1.Hoewel tekenen op grotere schaal,bijv.1:4, gemakkelijker is wordt het door de hoge kosten van fotografisch verkleinen minder aantrekkelijk als het om kleine aantallen kaarten gaat.

De EPROM-kaart past uiteraard op dezelfde bus.

A.W. RIJKEBOER
VINKENKROCHTLAAN 13
1951 KK VELSEN NOORD
02510-26520

6X2K EPROM2716

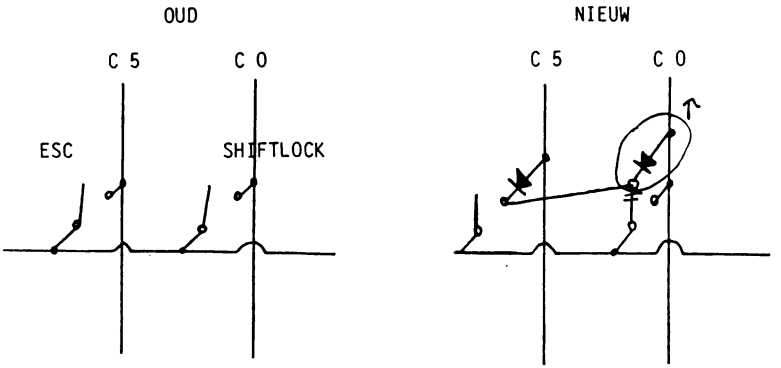



8K RAM
2114
I38-2
138-1
GND
R/W
+5V

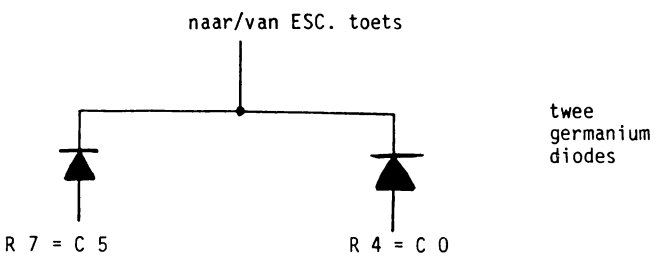
WIJZIGING KEYBOARD VOOR HW MONITOR ROM

ESC. toets detektie ontbreekt in lowercase-stand (shiftlock-up)

In de nieuwe stand lijkt het wanneer de ESC. toets wordt ingedrukt alsof de shiftlock ingedrukt staat dus:



de twee germaniumdiodes komen te zitten bij de weerstanden ter hoogte van de 0 toets.
(zie manual voor juiste positie)



Printspoor van de ESC. toets naar de Z toets doorsnijden, draadje aanbrengen van ESC. toets naar diodes.

VEEL SUCCES, John Glaser

GEHEUGEN UITBREIDING MET DYNAMISCHE RAM's (4116)

Dynamische ram's hebben een aantal voordelen boven statische ram's :

- ze verbruiken minder stroom
- ze hebben een grotere dichtheid per chip
- ze zijn per bit (byte) $\pm$ 4 maal goedkoper

Er zijn echter ook nadelen :

- ze vragen meer hardware om te "refreshen"
- ze hebben 3 voedingsspanningen nodig (i.p.v. 1)
- het is moeilijker ze op 2 MHz te gebruiken

Het geheugenelement van een dynamische ram is opgebouwd uit een condensator met transistor. Omdat de lading van de kondensator weglekt, moet minstens 1 keer in de 10 mS een z.g. refresh-cyclus plaatsvinden. Bij die refreshing wordt eerst het refresh-adres op de adreslijnen gezet (door de 74LS393) en daarna de $\overline{RAS}$ (=Row Adres Strobe = rijadressignaal) laag gemaakt. Door het selekteren van de rij geheugenelementen, wordt ze gerefreshed.

Het lezen en schrijven van de geheugenchip zelf gebeurt ook op een aparte manier :

1. eerst wordt het rij-adres (AØ t/m A6) aangeboden (via de buffer 81LS95 en het $\overline{ROWB}$ signaal).
2. de $\overline{RAS}$ wordt laag gemaakt, het geheugen I.C. selekteert (en refreshed automatisch een aantal v/d) 7 onderste lijnen.
3. dan wordt hetkolomadres (A7 t/m A13) aangeboden (weer via een 81LS95 maar nu met $\overline{COLB}$).
4. het $\overline{CAS}$ signaal wordt laag gemaakt
5. afhankelijk van het R/$\overline{W}$ signaal wordt de uitgekozen geheugenplaats beschreven of uitgelezen.

Per 4116 wordt zo 1 bit geselekteerd, om 8 bits te krijgen (=1 byte) zijn dus 8 stuks 4116 nodig, die samen 16K leveren.

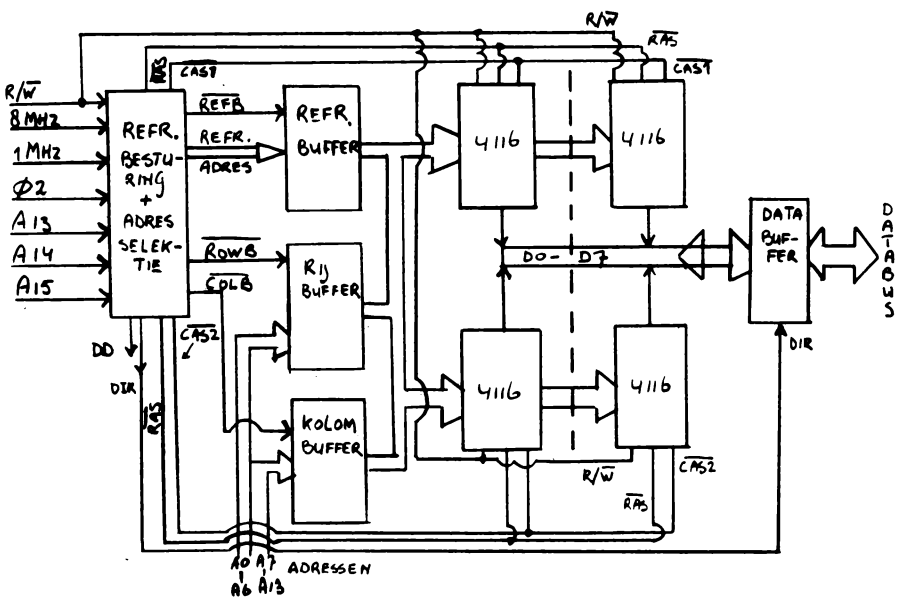
Nu het schema zelf.

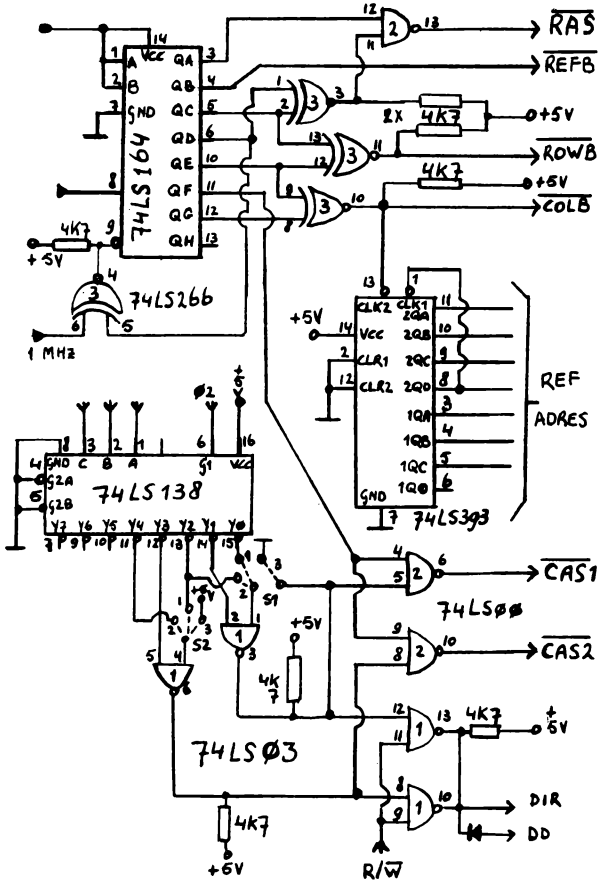
Het 8 MHz en 1 MHz signaal worden van deler U30 afgetakt. (resp. pin 2 (of W9) en pin 12)

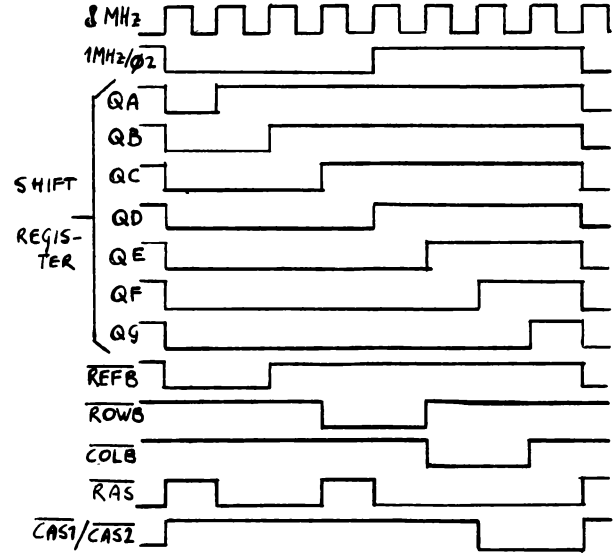
De adres- en datalijnen evenals de R/$\overline{W}$ en het Ø2 signaal worden van de 40-polige uitbreidingsplug (J1) gehaald.

Het DD signaal zorgt ervoor dat wanneer er op het Superboard databuffers zitten, deze omgeschakeld worden als er gelezen dan wel geschreven wordt.
De +12 volt, +5 volt, -5volt en aarde-aansluiting komen rechtstreeks uit een voeding (extra te maken !)
De synchronisatie van de refresh-logika gebeurt met het 1 MHz signaal en de QD uitgang van het schuifregister(74LS164)
De $\overline{RAS}$ en de R/$\overline{W}$ lijnen van de bovenste en onderste geheugenrij zijn met elkaar doorverbonden.Door middel van de $\overline{CAS1}$ en $\overline{CAS2}$ worden de verschillende geheugenrijen geaktveerd.
Zoals uit het schema te lezen valt is de adresselektie per 8 K. omschakelbaar. De hoogste geheugenrij kan worden gebruikt tot 40 K. ($9FFF) of tot 32 K. ($7FFF), waardoor er nog 8 K. overblijft voor eventuele EPROM's.
Een print-tekening was bij het maken van dit 50 pag. boekje nog niet (helemaal) af, mogelijk kan hij op 21 november 1981 worden gepresenteerd.

BLOKSCHEMA VAN DE 32 K. UITBREIDING





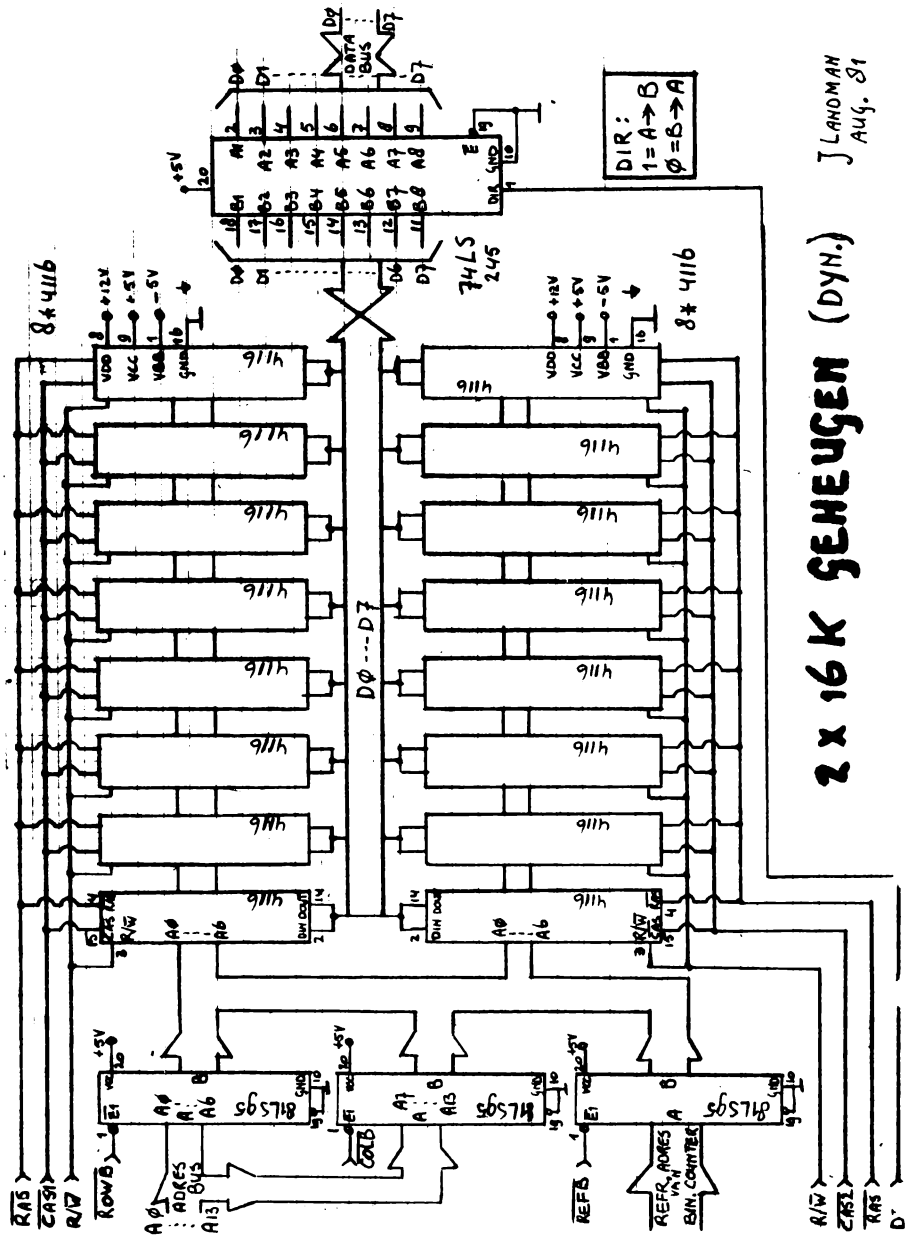


S1 CAS1		
	1	Ø - 16 K
	2	8 - 24 K
	3	OFF

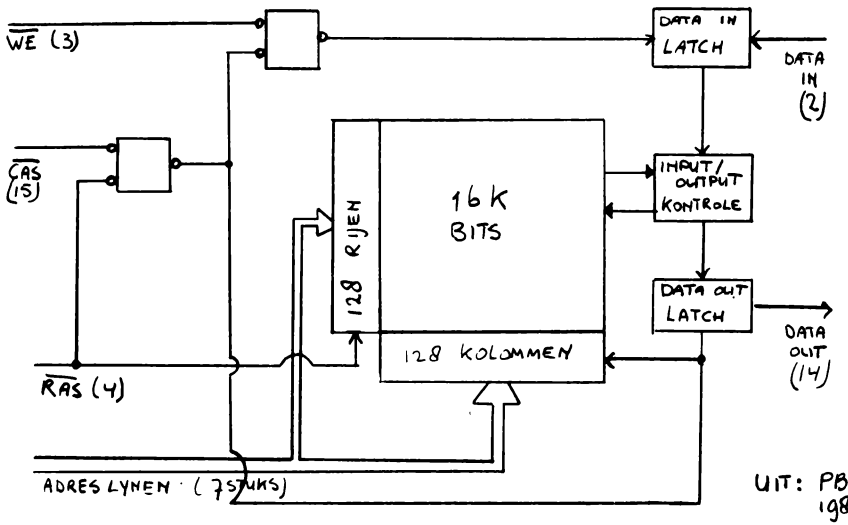
S2 CAS2		
	1	16 - 32 K
	2	24 - 40 K
	3	24 - 32 K ↳ 8 K

REFRESH LOGICA EN
ADRES SELECTIE

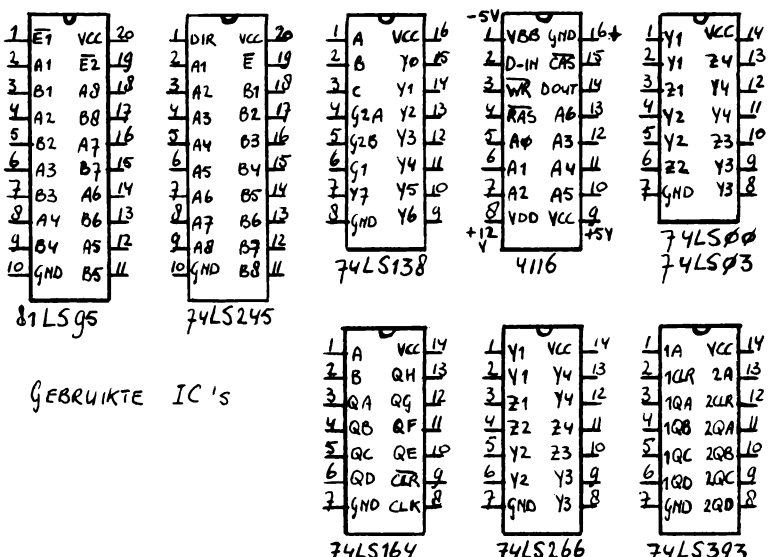
J LANDMAN
Aug. 81


2 x 16 K GEHEUGEN (DYN.)
J. LANDMAN AUG. 81
DIR: 1=A→B Ø=B→A
74LS 245
8 * 4116
DATA BUS
D0---D7
RAS
CAS1
R/W
ROW B
ADRES BUS
COL B
REFB
REFR. ADRES VAN TELLER
A/W
CAS1
RAS
D'

OPBOUW VAN DE 4116



UIT: PBE 1980-2



GEBRUIKTE IC 's

OKT 81 11.

2716/2732 EPROMPROGRAMMER

DE PROGRAMMER DIE OORSPRONKELIJK VOOR DE SYM WAS BEDOELD, IS IN HET VOORBEELD AANGEPAST VOOR OSI, MAAR IS OP IEDER 6502 SYSTEEM AAN TE SLUITEN, WAARIN 6532-6522, OF 6532-6532 AANWEZIG IS. ALLEEN HET GEDEELTE RECHTS VAN DE STIPPELLIJN IS DAN NODIG EN DE ADRESSEN VAN DE SYMBOLTABLE MOETEN WORDEN AANGEPAST.

HET LINKERGEDEELTE IS NODIG VOOR COMPUTERS ZONDER 6532-6522. IN HET SCHEMA ZIJN DE AANSLUITINGEN VAN DE CHIP-SELECT EN DE REGISTER-SELECTLIJNEN AANGEGEVEN VOOR OSI.

DE SIGNALEN UIT DE POORTLIJNEN PBD OTM 3 WORDEN IN HET PROGRAMMA ZODANIG GECONFIGUREERD DAT PIN 18, 20 EN 21 DE SPANNING VOERT DIE NODIG IS, AFHANKELIJK VAN TYPE EPROM EN FAZE VAN HET PROGRAMMEREN.

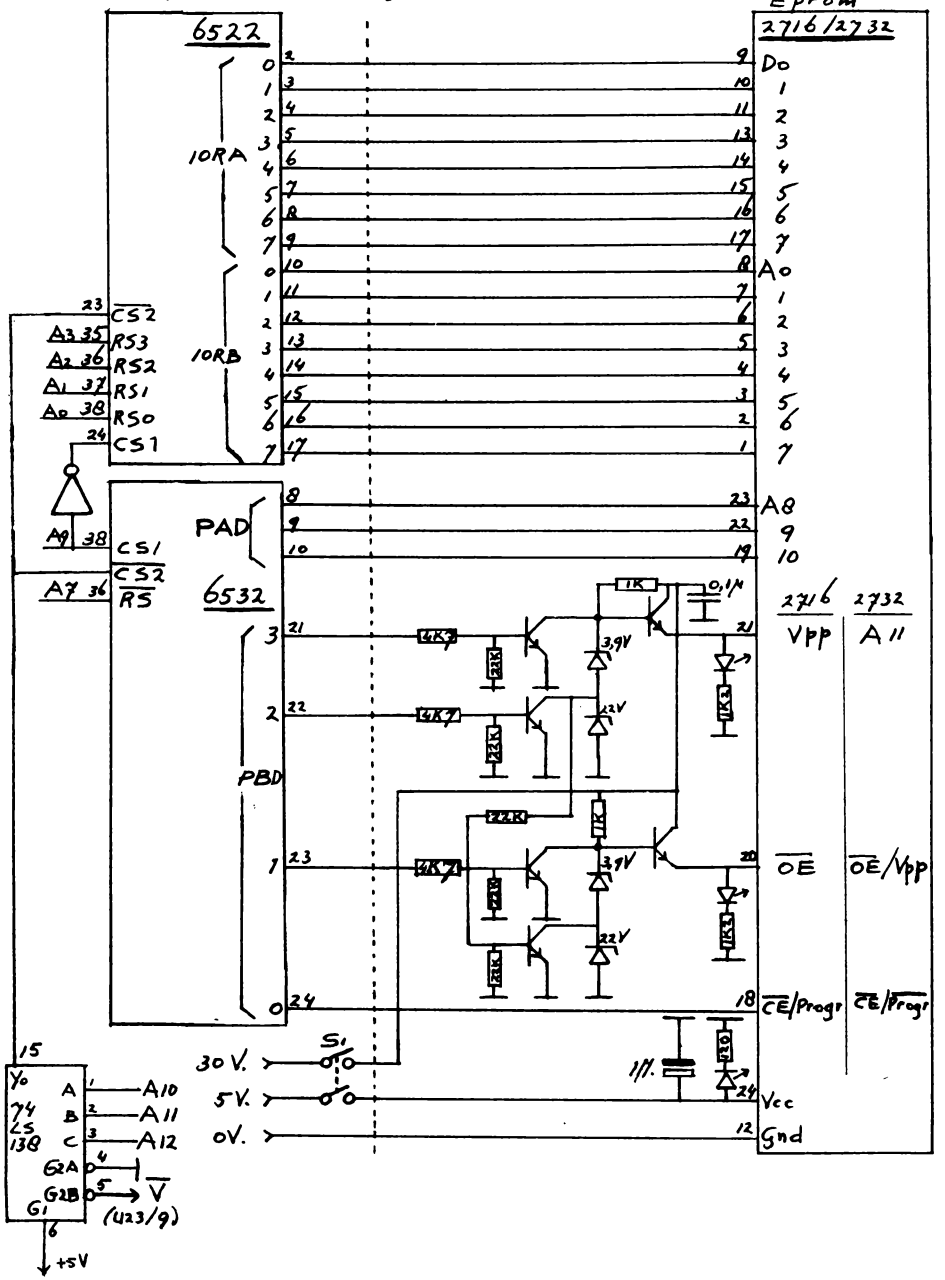
AAN HET OPLICHTEN VAN DE LED'S IS TE ZIEN, OF ER GEPROGRAMMEERD WORDT EN WAT ER GEPROGRAMMEERD WORDT: (2716 OF 2732 EERSTE 2K OF 2732 TWEEDE 2K).

HET PROGRAMMEREN VAN ENKELE BYTES IS MOGELIJK, DOOR IN DE BUFFER ALLEEN DIE BYTES TE VULLEN MET DATA EN DE NIET TE PROGRAMMEREN ADRESSEN TE VULLEN MET FF.

Type	Functie	PB3	PB2	PB1	PB0	PBD
2716	STA/STP	H	X	H	L	$0E
	Read	L	H	H	L	$06
	Program	L	L	L	⊓	$0%/$01
2732 1ste 2K	STA/STP	H	X	H	L	$0E
	Read	H	X	H	L	$0E
	Program	H	H	L	⊔	$0%/$0C
2732 2de 2K	STA/STP	H	X	H	L	$0E
	Read	L	H	H	L	$06
	Program	L	H	L	⊔	$0%/$08

ZIE VOLGENDE PAGINA

2716/2732 Programmer met 25 volt sturing.



```
 10 0000          ;**************************************
 20 0000          ;* EPROM PROGRAMMER VLG'S EDN 9/81    *
 30 0000          ;* AANGEPAST VOOR OSI EN UITGEBREID   *
 40 0000          ;* VOOR 2716 EN 2732 EPROMS DOOR      *
 50 0000          ;* KEES HOOGELAND                     *
 60 0000          ;**************************************
 70 0000          ;
 80 0000          ;PROGRAMMEERMOGELIJKHEDEN:
 90 0000          ;
100 0000          ;S=START/STOP:INITIALISEER VIA EN PIA,
110 0000          ;              VERWIJDEREN EPROM TOEGESTAAN.
120 0000          ;              (SPANNING AFZETTEN)
130 0000          ;B=BLANK:CONTROLEER OF EPROM GEWIST IS.
140 0000          ;V=VERIFY:VERGELIJK INHOUD VAN RAMBUFFER
150 0000          ;         MET INHOUD VAN EPROM (GEDEELTE).
160 0000          ;M=MONITOR:TERUGKEER NAAR EXT.MON
170 0000          ;C=COPY:COPIEER INHOUD EPROM NAAR RAMBUFFER
180 0000          ;P=PROGRAM:COPIEER INHOUD RAMBUFFER
190 0000          ;          NAAR EPROM
200 0000          ;
210 0000                 INCHAR=$E853
220 0000                 OUTCHR=$E861
230 0000                 PRBYT=$EAAC
240 0000                 CRLF=$EB07
250 0000                 CR=$F9B9
260 0000                 BELL=$FBD0
270 0000                 STROUT=$A8C3
280 0000                 BUFFER=$3800
290 0000                 BYTADR=$0090
300 0000                 PAGNR=BYTADR+1
310 0000                 LSTPG1=BYTADR+2
320 0000                 XMON=$E800
330 0000                 IORB=$C000   VIA ADRESSEN
340 0000                 IORA=IORB+1
350 0000                 DDRB=IORB+2
360 0000                 DDRA=IORB+3
370 0000                 RDFLAG=$C2D5
380 0000                 CLK1KT=$C2F7
390 0000                 PAD=$C280
400 0000                 PADD=PAD+1
410 0000                 PBD=PAD+2
420 0000                 PBDD=PAD+3
430 0000                 HCONRD=$C200
440 0000                 HCONPR=HCONRD+1
450 0000                 HPROG1=HCONRD+2
460 3000                 *=$3000
470 3000 2007EB  START   JSR CRLF
480 3003 205931          JSR STASTP
490 3006                 H0=MSG0/256*256
500 3006 A92B            LDA #MSG0-H0
510 3008 A030            LDY #MSG0/256
520 300A 20C3A8          JSR STROUT PRINT MESSAGE 0
530 300D 2007EB          JSR CRLF
540 3010 20D0FB          JSR BELL
550 3013 2053E8          JSR INCHAR WAT VOOR EPROM (GEDEELTE)?
560 3016 C931            CMP #'1    2716?
```

```
570 3018 D003           BNE *+5     NEE!
580 301A 4CCE30         JMP IN2716 JA, HULPLOC'S VULLEN
590 301D C932           CMP #'2     EERSTE 2K VAN 2732?
600 301F D003           BNE *+5     NEE!
610 3021 4CE030         JMP IN2K32 JA, HULPLOC'S VULLEN
620 3024 C933           CMP #'3     TWEEDE 2K VAN 2732?
630 3026 D0D8           BNE START   NEE, PROBEER OPNIEUW
640 3028 4CF230         JMP IN4K32 JA, HULPLOC'S VULLEN
650 302B 2A       MSGO  .BYTE '**************************',$0A,$0D
650 302C 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
650 303C 2A 2A 2A 2A 2A 2A 2A 2A 0A 0D
660 3046 2A             .BYTE '**** EPROMPROGRAMMER ****',$0A,$0D
660 3047 2A 2A 2A 20 45 50 52 4F 4D 50 52 4F 47 52 41 4D
660 3057 4D 45 52 20 2A 2A 2A 2A 0A 0D
670 3061 2A             .BYTE '* TYPE VOOR 2716    :1  *',$0A,$0D
670 3062 20 54 59 50 45 20 56 4F 4F 52 20 32 37 31 36 20
670 3072 20 20 20 3A 31 20 20 2A 0A 0D
680 307C 2A             .BYTE '*       VOOR 2732/2K :2  *',$0A,$0D
680 307D 20 20 20 20 20 20 56 4F 4F 52 20 32 37 33 32 2F
680 308D 32 4B 20 3A 32 20 20 2A 0A 0D
690 3097 2A             .BYTE '*       VOOR 2732/4K :3  *',$0A,$0D
690 3098 20 20 20 20 20 20 56 4F 4F 52 20 32 37 33 32 2F
690 30A8 34 4B 20 3A 33 20 20 2A 0A 0D
700 30B2 2A             .BYTE '**************************',$0A,0D
700 30B3 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
700 30C3 2A 2A 2A 2A 2A 2A 2A 2A 0A 0D
710 30CD 00             .BYTE $00
720 30CE A906     IN2716 LDA #%00000110  BITPATRONEN IN POORT B
730 30D0 8D00C2         STA HCONRD      VOOR 2716
740 30D3 A900           LDA #%00000000
750 30D5 8D01C2         STA HCONPR
760 30D8 A901           LDA #%00000001
770 30DA 8D02C2         STA HPROG1
780 30DD 4C0431         JMP CMND
790 30E0 A90E     IN2K32 LDA #%00001110  BITPATRONEN IN POORT B
800 30E2 8D00C2         STA HCONRD      VOOR EERSTE 2K VAN 2732
810 30E5 A90D           LDA #%00001101
820 30E7 8D01C2         STA HCONPR
830 30EA A90C           LDA #%00001100
840 30EC 8D02C2         STA HPROG1
850 30EF 4C0431         JMP CMND
860 30F2 A906     IN4K32 LDA #%00000110  BITPATRONEN IN POORT B
870 30F4 8D00C2         STA HCONRD      VOOR TWEEDE 2K VAN 2732
880 30F7 A905           LDA #%00000101
890 30F9 8D01C2         STA HCONPR
900 30FC A904           LDA #%00000100
910 30FE 8D02C2         STA HPROG1
920 3101 4C0431         JMP CMND
930 3104 A200     CMND  LDX #$00   CLR PAGE ZERO
940 3106 8A             TXA
950 3107 9500           STA $00,X
960 3109 E8             INX
970 310A D0FB           BNE *-3
980 310C 2007EB         JSR CRLF
```

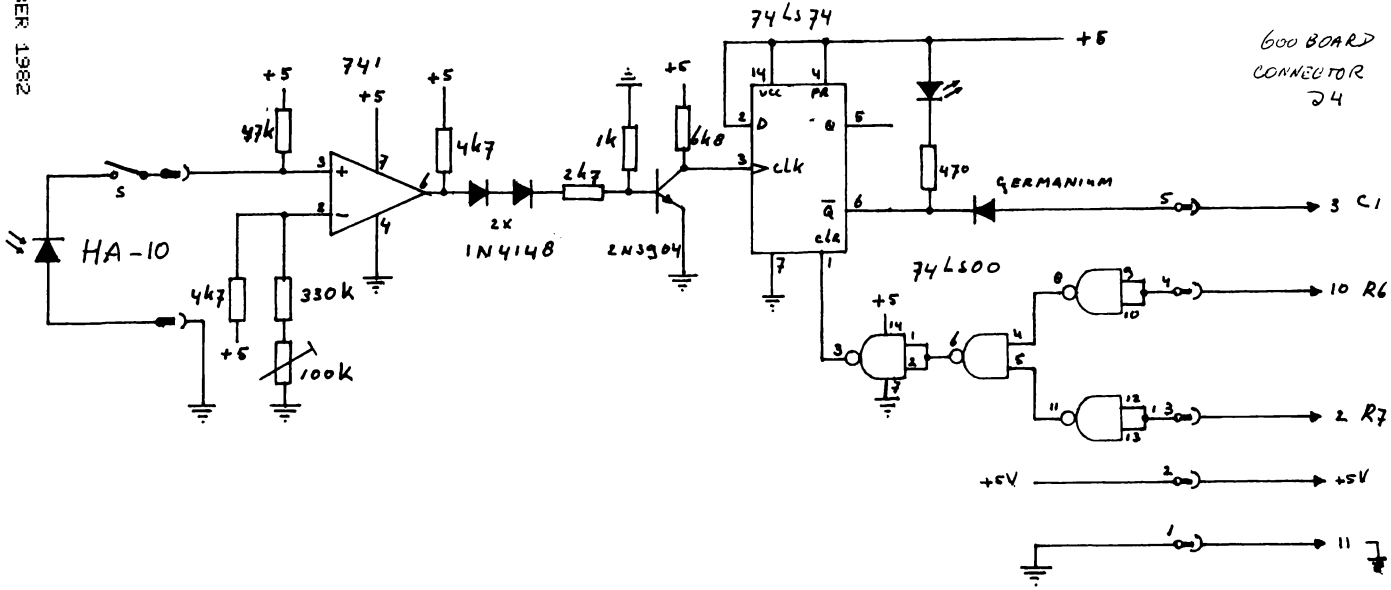
```
 990 310F                H1=MSG1/256*256
1000 310F A949           LDA #MSG1-H1
1010 3111 A031           LDY #MSG1/256
1020 3113 20C3A8         JSR STROUT  PRINT S/B/V/M/C/P
1030 3116 20D0FB         JSR BELL
1040 3119 2053E8         JSR INCHAR  WELK COMMMANDO?
1050 311C C956           CMP #'V     VERIFY?
1060 311E D003           BNE *+5     NEE!
1070 3120 201032         JSR VERIFY  JA, DOE HET
1080 3123 C942           CMP #'B     CONTROLE OP FF?
1090 3125 D003           BNE *+5     NEE!
1100 3127 205C32         JSR BLANK   JA,CONTROLEER EPROM
1110 312A C943           CMP #'C     COPIEREN NAAR BUFFER?
1120 312C D003           BNE *+5     NEE!
1130 312E 20C431         JSR COPY    JA, DOE HET
1140 3131 C94D           CMP #'M     TERUG NAAR EXT.MON?
1150 3133 D003           BNE *+5     NEE!
1160 3135 4C00E8         JMP XMON    JA,TERUG N. XMON
1170 3138 C950           CMP #'P     PROGRAMMEREN?
1180 313A D003           BNE *+5     NEE!
1190 313C 206F31         JSR PROGRM  JA, DOE HET
1200 313F C953           CMP #'S     START/STOP CONFIGURATIE?
1210 3141 D003           BNE *+5     NEE!
1220 3143 205931         JSR STASTP  JA,CONFIGUREER
1230 3146 4C0431         JMP CMND    PROBEER OPNIEUW
1240 3149 20       MSG1  .BYTE ' S/B/V/M/C/P ? ',$00
1240 314A 53 2F 42 2F 56 2F 4D 2F 43 2F 50 20 3F 20 00
1250 3159 A90F     STASTP LDA #%00001111
1260 315B 8D83C2          STA PBDD PBD 0T/M3=OUTPUT
1270 315E A90E            LDA #%00001110
1280 3160 8D82C2          STA PBD  PIN 18,20 EN 21 0 VOLT
1290 3163 A900            LDA #$00
1300 3165 8D03C0          STA DDRA    =INPUT
1310 3168 8D02C0          STA DDRB    IDEM
1320 316B 8D81C2          STA PADD    IDEM
1330 316E 60              RTS
1340 316F A2FF     PROGRM LDX #$FF    CODE VOOR OUTPUT
1350 3171 20E531          JSR CONFPR REG'S EN ADR.VOORBEREIDEN
1360 3174 A591     PROGR1 LDA PAGNR   PAGINA ADRES
1370 3176 8D80C2          STA PAD     NAAR EPROM
1380 3179 A590     PROGR2 LDA BYTADR  BYTE ADRES
1390 317B 8D00C0          STA IORB    NAAR EPROM
1400 317E B190            LDA (BYTADR),Y BUFFERINHOUD
1410 3180 C9FF            CMP #$FF    DIT BYTE PROGRAMMEREN?
1420 3182 F019            BEQ PROGR4 NEE!VOLGEND BYTE
1430 3184 8D01C0          STA IORA    JA SHRIJF IN EPROM
1440 3187 A931            LDA #49     =50 M.SEC.VOOR 1MHZ KLOK
1450 3189 8DF7C2          STA CLK1KT 6532 TIMER
1460 318C AD02C2          LDA HPROG1
1470 318F 8D82C2          STA PBD     PROGRAMMEER!
1480 3192 2CD5C2   PROGR3 BIT RDFLAG 50 M.SEC. AL OM?
1490 3195 10FB            BPL PROGR3 NEE!PROBEER OPNIEUW
1500 3197 AD01C2          LDA HCONPR JA,
1510 319A 8D82C2          STA PBD     EINDE PROGR.PULS
```

```
1520 319D E690     PROGR4 INC BYTADR VOLGEND BYTE
1530 319F D0D8            BNE PROGR2 PAGINA VOL? NEE!VOLGEND BYTE
1540 31A1 E691            INC PAGNR  JA,VOLGENDE PAGINA
1550 31A3 A591            LDA PAGNR
1560 31A5 C592            CMP LSTPG1 ALLE PAGINA'S KLAAR?
1570 31A7 D0CB            BNE PROGR1 NEE!VOLGENDE
1580 31A9 A920            LDA #$20   SPATIE
1590 31AB 2061E8          JSR OUTCHR PRINT!
1600 31AE                 H2=MSG2/256*256
1610 31AE A9B8            LDA #MSG2-H2
1620 31B0 A031            LDY #MSG2/256
1630 31B2 20C3A8          JSR STROUT KLAAR MELDING
1640 31B5 20D0FB          JSR BELL
1650 31B8 80       MSG2   .BYTE 128,150,161,'DONE',161,150,128,0
 1650 31B9 96 A1 44 4F 4E 45 A1 96 80 00
1660 31C3 60              RTS
1670 31C4 A200     COPY   LDX #$0C   CODE VOOR INPUT
1680 31C6 20EE31          JSR CONFRD REG'S EN ADR. VOORBEREIDEN
1690 31C9 A591     COPY1  LDA PAGNR  PAGINA ADRES
1700 31CB 8D80C2          STA PAD    NAAR EPROM
1710 31CE A590     COPY2  LDA BYTADR BYTE ADRES
1720 31D0 8D00C0          STA IORB   NAAR EPROM
1730 31D3 AD01C0          LDA IORA   LEES EPROMDATA
1740 31D6 9190            STA (BYTADR),Y SHRIJF IN BUFFER
1750 31D8 E690            INC BYTADR BYTADRES VERHOGEN
1760 31DA D0F2            BNE COPY2  PAGINA VOL?NEE!VOLGEND BYTE
1770 31DC E691            INC PAGNR  JA,VOLGENDE PAGINA
1780 31DE A591            LDA PAGNR
1790 31E0 C592            CMP LSTPG1 ALLE PAGINA'S GEDAAN?
1800 31E2 D0E5            BNE COPY1  NEE!VOLGENDE
1810 31E4 60              RTS        JA,TERUG NAAR COMMANDMODE
1820 31E5 AD01C2  CONFPR  LDA HCONPR PROGRAMMEERCONFIGURATIE
1830 31E8 8D82C2          STA PBD    PIN 18,20 EN 21
1840 31EB 18              CLC
1850 31EC 9006            BCC CNFRD1 VERDERGAAN
1860 31EE AD00C2  CONFRD  LDA HCONRD LEESCONFIGURATIE
1870 31F1 8D82C2          STA PBD    PIN 18,20 EN 21
1880 31F4 8E03C0  CNFRD1  STX DDRA   IN C.Q UIT-CODE VOOR DATAPOORT
1890 31F7 A9FF            LDA #%11111111 BITPATROON VOOR OUTPUT
1900 31F9 8D02C0          STA DDRB       A0 T/M A7
1910 31FC A907            LDA #%00000111 BITPATROON VOOR OUTPUT
1920 31FE 8D81C2          STA PADD       A8 T/M A10
1930 3201 A900            LDA #$00
1940 3203 A8              TAY
1950 3204 8590            STA BYTADR BUFFER BEGINADRES (BYTE)
1960 3206 18              CLC
1970 3207 A938            LDA #$38
1980 3209 8591            STA PAGNR  BUFFER BEGINADRES (PAGINA)
1990 320B 6908            ADC #$08
2000 320D 8592            STA LSTPG1 BUFFER EINDADRES+1(PAGINA)
2010 320F 60              RTS
```

```
2020 3210 A200    VERIFY LDX #$00   CODE VOOR INPUT
2030 3212 20EE31         JSR CONFRD REG'S EN ADR. VOORBEREIDEN
2040 3215 A591    VERFY1 LDA PAGNR  PAGINA ADRES
2050 3217 8D80C2         STA PAD    NAAR EPROM
2060 321A A590    VERFY2 LDA BYTADR BYTE ADRES
2070 321C 8D00C0         STA IORB   NAAR EPROM
2080 321F B190           LDA (BYTADR),Y HAAL BUFFERDATA
2090 3221 CD01C0         CMP IORA   GELIJK AAN EPROMDATA?
2100 3224 F003           BEQ VERFY3 JA,VOLGENDE
2110 3226 203632         JSR OUTPUT NEE!PRINT ADRES MET FOUT
2120 3229 E690    VERFY3 INC BYTADR VOLGEND BYTE
2130 322B D0ED           BNE VERFY2 PAGINA VOL?NEE,VOLGEND ADRES
2140 322D E691           INC PAGNR  JA, VOLGENDE PAGINA
2150 322F A591           LDA PAGNR
2160 3231 C592           CMP LSTPG1 ALLE PAGINA'S GEDAAN?
2170 3233 D0E0           BNE VERFY1 NEE!VOLGENDE
2180 3235 60             RTS        JA,TERUG NAAR COMMAND MODE
2190 3236 48      OUTPUT PHA    BUFFERDATA NAAR STACK
2200 3237 20B9F9         JSR CR
2210 323A A591           LDA PAGNR
2220 323C 20ACEA         JSR PRBYT ;PRINT ADR.HI
2230 323F A590           LDA BYTADR
2240 3241 20ACEA         JSR PRBYT ;PRINT ADR.LO
2250 3244 A92F           LDA #'/
2260 3246 2061E8         JSR OUTCHR
2270 3249 68             PLA
2280 324A 20ACEA         JSR PRBYT ;PRINT BUFFERINH
2290 324D A92F           LDA #'/
2300 324F 2061E8         JSR OUTCHR
2310 3252 AD01C0         LDA IORA
2320 3255 20ACEA         JSR PRBYT ;PRINT EPROMINH.
2330 3258 2007EB         JSR CRLF
2340 325B 60             RTS
2350 325C A200    BLANK  LDX #$00   CODE VOOR INPUT
2360 325E 20EE31         JSR CONFRD REG'S EN ADR.VOORBEREIDEN
2370 3261 A591    BLANK1 LDA PAGNR  PAGINA ADRES
2380 3263 8D80C2         STA PAD    NAAR EPROM
2390 3266 A590    BLANK2 LDA BYTADR BYTE ADRES
2400 3268 8D00C0         STA IORB   NAAR EPROM
2410 326B A9FF           LDA #$FF
2420 326D CD01C0         CMP IORA   IS EPROMDATA $FF?
2430 3270 F003           BEQ BLANK3 JA,CHECK VOLGEND BYTE
2440 3272 203632         JSR OUTPUT NEE!PRINT AFWIJKING
2450 3275 E690    BLANK3 INC BYTADR VERHOOG BYTADRES
2460 3277 D0ED           BNE BLANK2 PAGINA VOL?NEE,VOLGEND BYTE
2470 3279 E691           INC PAGNR  JA,VERHOOG PAGINA ADRES
2480 327B A591           LDA PAGNR
2490 327D C592           CMP LSTPG1 ALLE PAGINA'S GEHAD?
2500 327F D0E0           BNE BLANK1 NEE!VOLGENDE
2510 3281 60             RTS        JA,TERUG NAAR COMMANDMODE
2520 3282                .END
```

NOVEMBER 1982

LIGHT PEN (MOD) *AARDVARK*



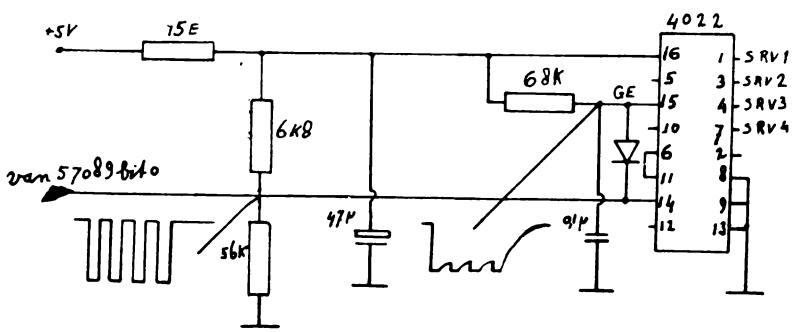
JLANDMAN

OSI POEL UITBREIDINGEN EC 20

SERVO-STURING

J. VAN DE BIESEBOS

```
 10 REM SERVO  FIJN   GROF
 20 REM   1    24087  24102
 30 REM   2    24117  24132
 40 REM   3    24147  24162
 50 REM   4    24177  24192
 55 REM
 60 REM LOW BYTE EN HIGH BYTE SERVO-UITGANG
 65 REM
 70 REM OP 1 EN 223 (57089 DEC) BIT 0
 80 REM
100 POKE 8955 , 0 : POKE 8956 , 94
110 FOR R = 24064 TO 24204
120 READ E   POKE R,E   NEXT
125 REM
130 DATA169,1  ,141,1,223,160,253,162,175,232,208,253,200,208,248
140 DATA169,255,141,1,223,160,255,162,175,232,208,253,200,208,248
150 DATA169,1  ,141,1,223,160,253,162,175,232,208,253,200,208,248
160 DATA169,255,141,1,223,160,255,162,175,232,208,253,200,208,248
170 DATA169,1  ,141,1,223,160,253,162,175,232,208,253,200,208,248
180 DATA169,255,141,1,223,160,255,162,175,232,208,253,200,208,248
190 DATA169,1  ,141,1,223,160,253,162,175,232,208,253,200,208,248
200 DATA169,255,141,1,223,160,255,162,175,232,208,253,200,208,248
210 DATA169,1  ,141,1,223,160,253,162,175,232,208,253,200,208,248
220 DATA169,255,141,1,223,96
```



```
5E00 A901    LDA ##$01
5E02 8D01DF  STA $DF01
5E05 A0FD    LDY ##$FD
5E07 A2AF    LDX ##$AF
5E09 E8      INX
5E0A D0FD    BNE $5E09
5E0C C8      INY
5E0D D0F8    BNE $5E07
5E0F A9FF    LDA ##$FF
5E11 8D01DF  STA $DF01
5E14 A0FF    LDY ##$FF
5E16 A2AF    LDX ##$AF
5E18 E8      INX
5E19 D0FD    BNE $5E18
5E1B C8      INY
5E1C D0F8    BNE $5E16
5E1E A901    LDA ##$01
5E20 8D01DF  STA $DF01
5E23 A0FD    LDY ##$FD
5E25 A2AF    LDX ##$AF
5E27 E8      INX
5E28 D0FD    BNE $5E27
5E2A C8      INY
5E2B D0F8    BNE $5E25

5E2D A9FF    LDA ##$FF
5E2F 8D01DF  STA $DF01
5E32 A0FF    LDY ##$FF
5E34 A2AF    LDX ##$AF
5E36 E8      INX
5E37 D0FD    BNE $5E36
5E39 C8      INY
5E3A D0F8    BNE $5E34
5E3C A901    LDA ##$01
5E3E 8D01DF  STA $DF01
5E41 A0FD    LDY ##$FD
5E43 A2AF    LDX ##$AF
5E45 E8      INX
5E46 D0FD    BNE $5E45
5E48 C8      INY
5E49 D0F8    BNE $5E43
5E4B A9FF    LDA ##$FF
5E4D 8D01DF  STA $DF01
5E50 A0FF    LDY ##$FF
5E52 A2AF    LDX ##$AF
5E54 E8      INX
5E55 D0FD    BNE $5E54
5E57 C8      INY
5E58 D0F8    BNE $5E52

5E5A A901    LDA ##$01
5E5C 8D01DF  STA $DF01
5E5F A0FD    LDY ##$FD
5E61 A2AF    LDX ##$AF
5E63 E8      INX
5E64 D0FD    BNE $5E63
5E66 C8      INY
5E67 D0F8    BNE $5E61
5E69 A9FF    LDA ##$FF
5E6B 8D01DF  STA $DF01
5E6E A0FF    LDY ##$FF
5E70 A2AF    LDX ##$AF
5E72 E8      INX
5E73 D0FD    BNE $5E72
5E75 C8      INY
5E76 D0F8    BNE $5E70
5E78 A901    LDA ##$01
5E7A 8D01DF  STA $DF01
5E7D A0FD    LDY ##$FD
5E7F A2AF    LDX ##$AF
5E81 E8      INX
5E82 D0FD    BNE $5E81
5E84 C8      INY
5E85 D0F8    BNE $5E7F

5E87 A9FF    LDA ##$FF
5E89 8D01DF  STA $DF01
5E8C 60      RTS
```

GEHEUGENUITBREIDING MET BANKS.

Kees Hoogeland

MET DE GEGEVEN SCHAKELING IS HET MOGELIJK OM NAAST HET WERKGEHEUGEN VAN 64 K NOG 8 MAAL 64 K AAN EXTRA GEHEUGEN TE ADRESSEREN. IN HET BIJBEHORENDE PROGRAMMA IS AANGEGEVEN HOE EEN BLOK MET DATA OF EEN PROGRAMMA WORDT OVERGEBRACHT VOORZIEN VAN EEN AUTOSTART-ADRES.

AAN DE HAND VAN DE 4 INSTRUCTIES IN HET VOORBEELD WORDT DE WERKING VERKLAARD. HET VERLOOP VAN DE DIVERSE SIGNALEN IS IN HET TIJDVOLGORDEDIAGRAM TE VOLGEN. DE VIERDE INSTRUCTIE IN HET VOORBEELD IS TOEGEVOEGD OM DE SIGNALEN MET DE SCOOP TE KUNNEN CONTROLEREN.

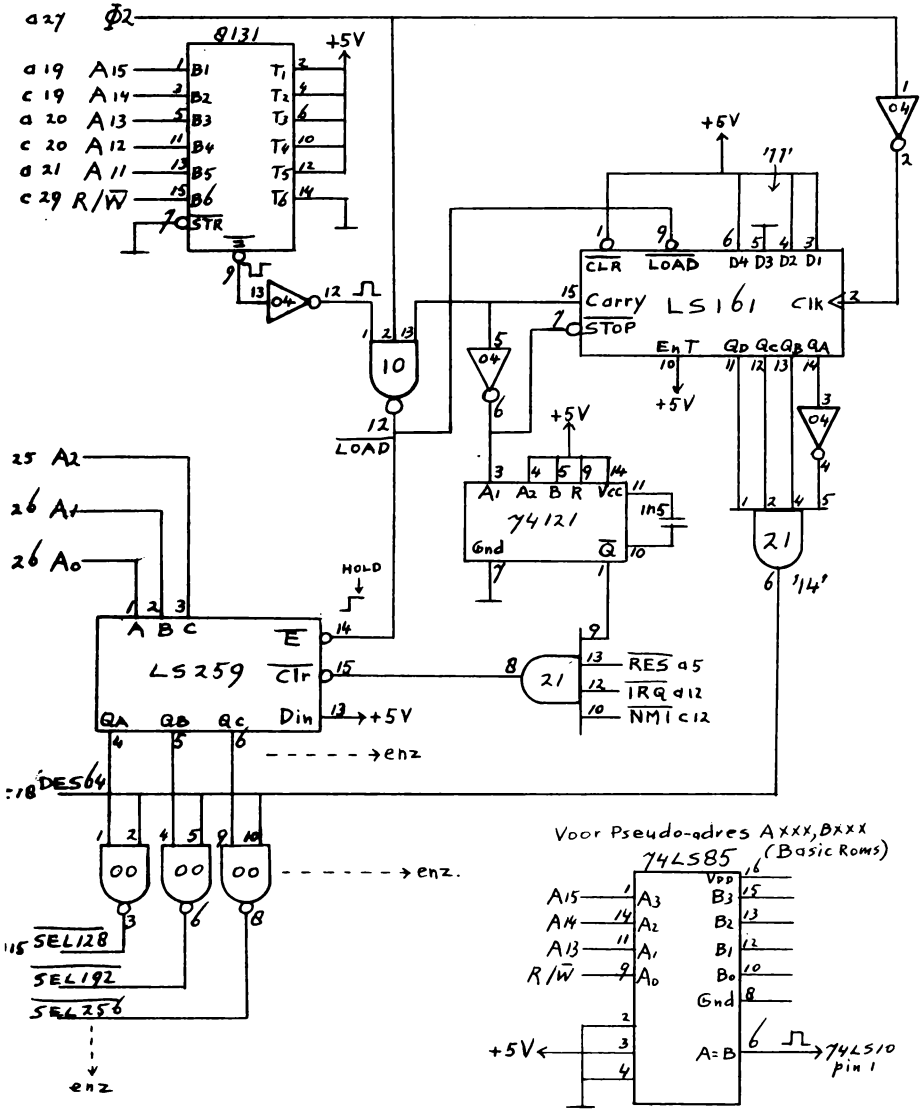
DE EERSTE INSTRUCTIE: STA FF00 IS DE ZGN. PSEUDO-INSTRUCTIE DIE DE BANKADRESERING MOET VOORBEREIDEN. DE COMPARATOR 8131 HERKEND HET ADRES F8XX OF HOGER, ALS DIT GECOMBINEERD IS MET HET LAAG ZIJN VAN R/W, DUS MET B.V. EEN STA-INSTRUCTIE.

DE PSEUDOINSTRUCTIE BEVAT HET ADRES VAN EEN WILLEKEURIG ROM OF EPROM IN HET WERKGEHEUGEN. DAAROM HOEFT ER GEEN ADRES GERESERVEERD TE WORDEN. ER WORDT IMMERS NIET NAAR EEN ROM GESCHREVEN.
(ONDER IN DE FIGUUR IS AANGEGEVEN HOE M.B.V. DE 7485 HET PSEUDO-ADRES MAG KOMEN IN HET ADRESSENBEREIK VAN DE BASIC ROMS. (AXXX-BXXX).

ALS DE PSEUDO INSTRUCTIE HERKEND WORDT ZAL ER OP HET HOOG GAAN VAN $\overline{\Phi 2}$, (ALS OOK DE TELLER LS161 OP 15 STAAT) EEN LOAD-PULS ONTSTAAN .

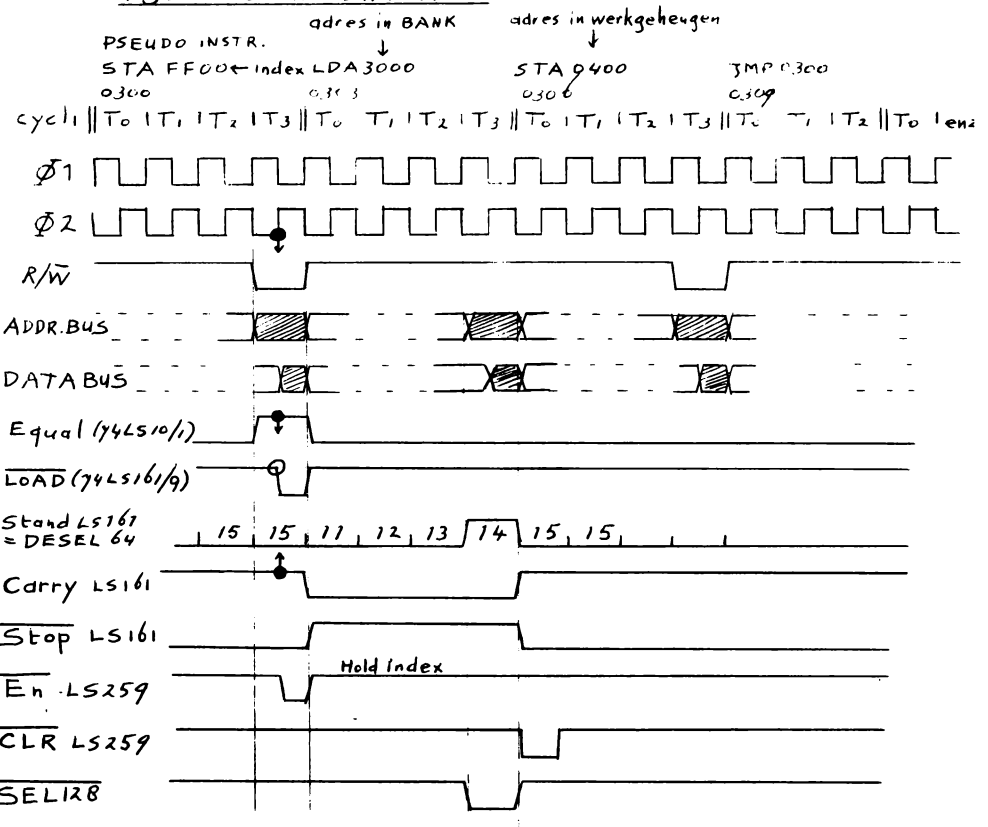
DEZE PULS OP LS10 PIN12 LAADT DE '1' VAN Din IN EEN VAN DE 8 LATCHES IN DE LS259 . DE KEUZE IS BEPAALD DOOR DE LAAGSTE DRIE BITS VAN HET PSEUDOADRES. DIT IS DE 'INDEX' WAARMEE HET NUMMER VAN DE GESELECTEERDE BANK WORDT AANGEGEVEN.

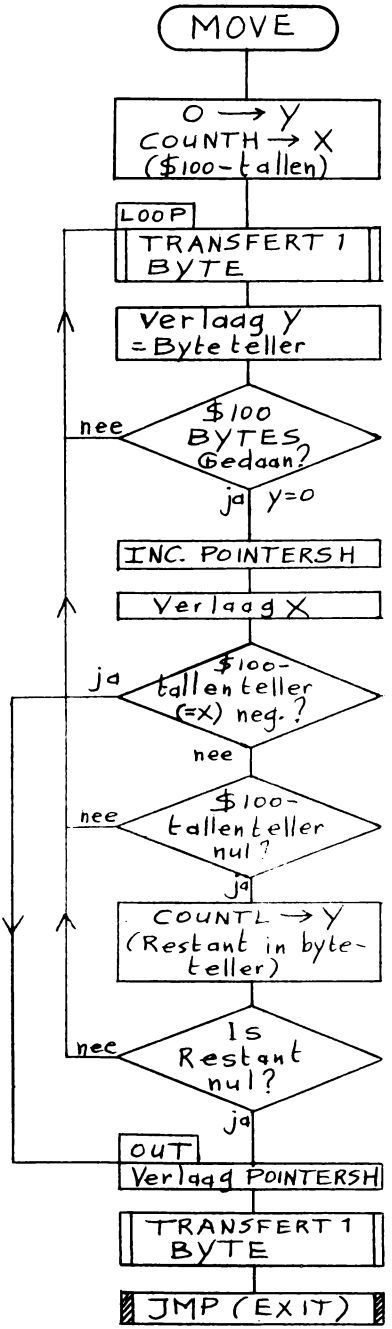
DE LOADPULS MAAKT OOK DAT DE TELLER LS161 OP 11 SPRINGT, IN OVEREENSTEMMING MET DE NIVEAU'S OP D1 T/M D4. DE TELLER STAAT NU NIET MEER GEBLOKKEERD DOOR DE GEINVERTEERDE CARRY, ZODAT DE NU VOLGENDE KLOKPULSEN DE TELLER VERHOGEN. DEZE PULSEN BRENGEN DE TELLER OP '14' TERWIJL DE TWEEDE INSTRUCTIE DAN NET OP T3 IS AANGEKOMEN. DIT IS DE CYCLUS IN DE 'LDA' INSTRUCTIE WAARIN DE DATA WORDT OVERGEBRACHT. HET SIGNAAL 'DES64' (='14') IS ECHTER HOOG GEWORDEN WAARDOOR DE CHIPSELECT VAN HET WERKGEHEUGEN IS ONDERDRUKT. DIT SIGNAAL WORDT IN HET SUPERBORD AANGESLOTEN OP U23/74LS138 PIN 4 OF 5. HEEFT MEN EEN DYN. RAMKAART VAN ELEKTUUR DAN KOMT DIT SIGNAAL OOK OOK HIEROP N.L. OP IC11 PIN 18 OF 19 EN OP DE RAM-EPROMKAART IS DIT IC5 PIN 18 OF 19.
DOOR HET SIGNAAL '14' TE COMBINEREN MET DE UITGANGEN VAN DE LS259 ONTSTAAT OP HET JUISTE MOMENT HET CHIPSELECT SIGNAAL VOOR DE GESELECTEERDE BANK, AAN TE SLUITEN OP DERGELIJKE PUNTEN ALS 'DES 64'.
DE BANKS KUNNEN ZOWEL ROMS ALS EPROMS OF RAMS ZIJN
VOORWAARDE IS WEL DAT DE INSTRUCTIE DIE
NA DE PSEUDOINSTRUCTIE KOMT STEEDS EENZELFDE CYCLUS-VERLOOP HEEFT.
LITERATUUR: ELEKTRONIK 6/26.3.1982


a27 Φ2
Q131
+5V
a19 A15
c19 A14
a20 A13
c20 A12
a21 A11
c29 R/W
STR
04
10
LOAD
'77'
CLR
LOAD
D4 D3 D2 D1
Carry
STOP
LS161
Clk
ENT
QD QC QB QA
21
'14'
74121
A1 A2 B R Vcc
Gnd
Q
INH
25 A2
26 A1
26 A0
HOLD
A B C
E
LS259
Clr
Din
QA QB QC
RES a5
IRQ a12
NMI c12
enz
DES64
00
SEL128
SEL192
SEL256
Voor Pseudo-adres AXXX, BXXX
(Basic Roms)
74LS85
A15 A14 A13 R/W
A3 A2 A1 A0
B3 B2 B1 B0
Vpp
Gnd
A=B
74LS10 pin 1

VOORBEELD	Tn	Addr. Bus	Data bus	R/$\bar{W}$
0300 8D00FF	T_0	PC	OPCODE	1
	T_1	PC +1	ADL	1
STA PSEUDO	T_2	PC +2	ADH	1
	T_3	ADH,ADL	DATA	0
0303 AD0030	T_0	idem	OPCODE	1
	T_1		ADL	1
LDA ABS1	T_2		ADH	1
	T_3		DATA	1
0306 8D0004	T_0	idem	OPCODE	1
	T_1		ADL	1
STA ABS2	T_2		ADH	1
	T_3		DATA	0
0309 4C0003	T_0	—	—	1
JMP BEGIN	T_1	—	—	1
	T_2	—	—	1

TIJDVOLGORDE DIAGRAM




MOVE
0 → Y
COUNTH → X
($100-tallen)
LOOP
TRANSFERT 1 BYTE
Verlaag Y
=Byte teller
nee
$100 BYTES Gedaan?
ja
y=0
INC. POINTERSH
Verlaag X
ja
$100-tallen teller (=X) neg.?
nee
nee
$100-tallen teller nul?
ja
COUNTL → Y
(Restant in byte-teller)
nee
Is Restant nul?
ja
OUT
Verlaag POINTERSH
TRANSFERT 1 BYTE
JMP (EXIT)


ENTRY
FROM → E0/E1
TO → E2/E3
COUNT → E4/E5
EXIT → D7D9/D7DA
STA PSEUDO
LDA BANKADR
RTS
D7DB t/m D7E1
JMP MOVE
TRANSFERT
ABSOLUUT
(Maak met Y, FROMH- en FROML-pointers de absolute adressen en plaats deze in subr. HAALBYTE)
HAALBYTE
(in Videoram)
BRENG WEG BYTE
RTS in MOVE
ABSOLUUT
FRPOINTERL + Y
Naar subr. HAALBYTE
FRPOINTERH + CARRY
Naar subr. HAALBYTE
RTS in TRANSFERT

```
 10 0000          ; ***********************************
 20 0000          ; **** MOVE VAN (NAAR) MEMORY-     ***
 30 0000          ; **** BANKS  MET INITIALISATIE-   ***
 40 0000          ; **** TABEL IN APART              ***
 50 0000          ; **** ENTRY-PROGRAMMA             ***
 60 0000          ; **** Kees Hoogeland Beverwijk    ***
 70 0000          ; **** 2 FEBR.1983                 ***
 80 0000          ; ***********************************
 90 0000          ;
100 0000                 FRPONT=$E0        FROM-POINTER
110 0000                 TOPONT=FRPONT+2   TO-POINTER
120 0000                 COUNT=FRPONT+4    AANTAL BYTES
130 0000                 EXIT=$D7D9        BEVAT STARTADRES
140 0000          ;
150 3000                 *=$3000
160 3000 A6E5    MOVE    LDX COUNT+1       IN X DE $100-TALLEN
170 3002 A000            LDY #0            Y TELT $FF BYTES
180 3004 202130  LOOP    JSR TRANSF        BRENG 1 BYTE OVER
190 3007 88              DEY               $FF BYTES GEHAD?
200 3008 D0FA            BNE LOOP          NEE, VOLGENDE!
210 300A E6E1            INC FRPONT+1      JA, $100-TALLEN
220 300C E6E3            INC TOPONT+1        POINTERS VERHOGEN
230 300E CA              DEX               $100-TALLEN TELLER ?
240 300F 3006            BMI OUT           NEGATIEF, DAN UIT
250 3011 D0F1            BNE LOOP          NUL ??
260 3013 A4E4            LDY COUNT         DAN RESTANT HALEN
270 3015 D0ED            BNE LOOP          DE LAATSTE RONDE
280 3017 C6E1    OUT     DEC FRPONT+1      DIT IS NODIG WANT
290 3019 C6E3            DEC TOPONT+1      BYTE 0 MOET NOG
300 301B 202130          JSR TRANSF
310 301E 6CD9D7          JMP (EXIT)        AUTOSTART
320 3021 202730  TRANSF  JSR ABSOL         HAAL MBV. ABS. ADRES
330 3024 91E2            STA (TOPONT),Y    BRENG BYTE WEG
340 3026 60              RTS
350 3027 98      ABSOL   TYA               FRPONT+Y=FROM LOW
360 3028 18              CLC
370 3029 65E0            ADC FRPONT
380 302B 8DDFD7          STA EXIT+6
390 302E A5E1            LDA FRPONT+1      FRPONT+1=FROM HIGH
400 3030 8DE0D7          STA EXIT+7
410 3033 9003            BCC ABSOL1        PAG. OVERSCHREDEN ?
420 3035 EEE0D7          INC EXIT+7        JA, VERHOOG ABS. ADRES
430 3038 20DBD7  ABSOL1  JSR EXIT+2        HAAL, EVT. UIT BANK
440 303B 60              RTS
450 303C EA      END     NOP
460 303D          ;
470 303D          ;
480 303D          ;
490 3040                 *=MOVE+$40
500 3040 A206    ENTRY   LDX #6            FROM)) FRPONT
510 3042 BD5730  OVER1   LDA TABEL-1,X     TO)) TOPONT
520 3045 95DF            STA FRPONT-1,X    COUNTS)) COUNT
530 3047 CA              DEX
540 3048 D0F8            BNE OVER1
550 304A A209            LDX #9            EXIT)) $D7D9
560 304C BD5D30  OVER2   LDA TABEL+5,X     SUBR.)) $D7DB
570 304F 9DD8D7          STA EXIT-1,X      PSEUDO-ADR.)) $D7DC
580 3052 CA              DEX               TOTAAL 9 BYTES
590 3053 D0F7            BNE OVER2
600 3055 4C0030          JMP MOVE          TABEL IS VOORBEELD!
```

```
610 3058 40       TABEL  .BYTE $40,$82,$40,$02,$51,$11,$00,$13
610 3059 82
610 305A 40
610 305B 02
610 305C 51
610 305D 11
610 305E 00
610 305F 13
620 3060 8D              .BYTE $8D,$00,$FF,$AD,0,0,$60
620 3061 00
620 3062 FF
620 3063 AD
620 3064 00
620 3065 00
620 3066 60
630 3067          ;
640 3067          ;
650 3067          ;             TABEL
660 3067          ;        I-------------I
670 3067          ;$00E0 I  FRPOINTL    I
680 3067          ;        I-------------I
690 3067          ;$00E1 I  FRPOINTH    I
700 3067          ;        I-------------I
710 3067          ;$00E1 I  TOPOINTL    I
720 3067          ;        I-------------I
730 3067          ;$00E2 I  TOPOINTH    I
740 3067          ;        I-------------I
750 3067          ;$00E3 I  COUNTL      I
760 3067          ;        I-------------I
770 3067          ;$00E4 I  COUNTH      I
780 3067          ;        I-------------I
790 3067          ;        I             I
800 3067          ;     ----------------------
810 3067          ;        I             I
820 3067          ;        I             I
830 3067          ;        I-------------I
840 3067          ;$D7D9 I    EXITL     I
850 3067          ;        I-------------I
860 3067          ;$D7DA I    EXITH     I
870 3067          ;        I-------------I  (-- SUBR."HAAL BYTE"
880 3067          ;$D7DB I 8D (STA)     I  (-- PSEUDO INSTRUCTIE
890 3067          ;        I-------------I
900 3067          ;$D7DC I 00(--Index   I
910 3067          ;        I-------------I
920 3067          ;$D7DD I FF           I  (-- PSEUDO ADRES
930 3067          ;        I-------------I
940 3067          ;$D7DE I AD (LDA)     I EVT.UIT BANK
950 3067          ;        I-------------I
960 3067          ;$D7DF I 00 (Dummy)   I PLAATS VOOR FROML
970 3067          ;        I-------------I
980 3067          ;$D7E0 I 00 (Dummy)   I PLAATS VOOR FROMH
990 3067          ;        I-------------I
1000 3067         ;$D7E1 I 60 (RTS)     I
1010 3067         ;        I-------------I
1020 3067         ;        I             I
1030 3067                  .END
```

E E N S Y S T E E M B U S V O O R D E O. S. I.

Gebaseerd op de elektuurbus (en Junior)

In dit schema staan : de basisbus (in het midden)
de junior-uitbreidingen (buitenste rijen)
overige speciale uitbreidingen (benoemd)

jun.	elekt.	a	c	elekt.	jun.
	+5V	1	1	+5V	
		2	2		
		3	3		CLR (Brabosi)
	GND	4	4	GND	
RESET		5	5		RDY
(OSI)DD		6	6		WAIT (OSI 2P)
	D1	7	7	DØ	
	D3	8	8	D2	
	D5	9	9	D4	
	D7	10	10	D6	
		11	11		
	IRQ	12	12	NMI	
(eigen toe -12V		13	13		
voeg.) K7		14	14		S.O.
K5		15	15		K6
GND		16	16		GND
K4		17	17		+12V
-5V		18	18		K3.
	A15	19	19	A14	
	A13	20	20	A12	
	A11	21	21	A10	
	A9	22	22	A8	
	A7	23	23	A6	
	A5	24	24	A4	
	A3	25	25	A2	
	A1	26	26	AØ	
	Ø2	27	27		
		28	28		K2
K1		29	29	R/$\overline{W}$	
	Ø1	30	30		Ex
RAM R/$\overline{W}$		31	31		
	GND	32	32	GND	

Speciale toevoegingen zijn :

3c...CLR	Brabosi	15a..min(-)12V	Eigen toevoeging
6a...DD	OSI		
6c...WAIT	OSI 2P	16ac.GND	Junior
14c..S.O	Junior	17c..+12V	Junior
5a...RESET	Junior	18a..min(-)5V	Junior
5c...READY	Junior		

Op de kontakten met een K. staat een voor de Junior belngrijke adresselektie.

Rein Meesterman

C INTERFACING

CA Joy sticks

CB Parallel I/O

CC Serial I/O

CD Modems

CE Printers

CF Plotters

CENTRONICS PARRALLEL-INTERFACE VOOR DE OSI.

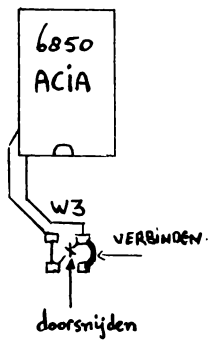
De meest voor de hand liggende oplossing voor het aansluiten van een printer op het superboard is gebruik te maken van de RS232 serie-interface. Aangezien de meeste nu gangbare printers standaard over een Centronics parrallel-interface beschikken dient er nog een (dure) aparte interface te worden aangeschaft.
Bijgaand schema geeft een goedkoper alternatief (ca. f50 incl. kabels). De schakeling wordt aangesloten op 3 signalen welke op de OSI aanwezig zijn nl. TX-data; TX-clock en CTS. De uitgang kan direct op de centronics standaard worden aangesloten. (EPSON bezitters zie HCCN 33 blz. 7)
Via een zespolige DIN-plug gaan we door een 3-aderige afgeschermde kabel naar de schakeling. Aan de andere kant verlaten we de interface met een 13-aderige flatcable.
De schakeling zelf is opgebouwd rond de UART AY-3-1015; het schema spreekt voor zich.

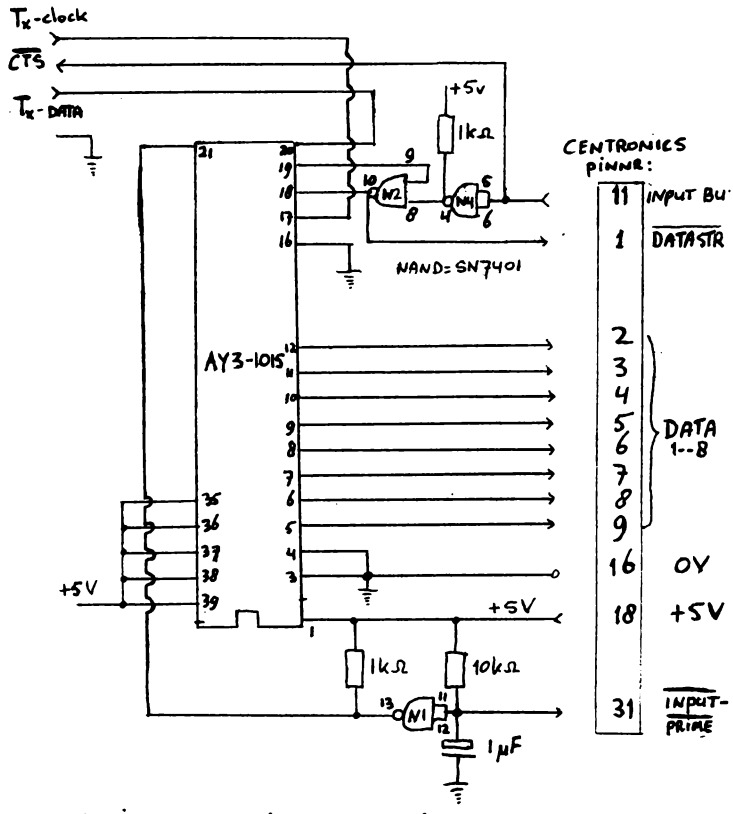
Aanpassingen in het superboard bestaan uit de aangegeven jumperverbindingen en het plaatsen van de IC's U67 en U68 met bijbehorende weerstanden. (zie manual en schema)

Diegenen die de baudrate van de cassette verhoogd hebben dienen ervoor te zorgen dat TX-clk 'naar buiten' dezelfde is als TX-clk naar de ACIA. Hiertoe printspoor naar pin11 U68 doorsnijden en pin11 U68 verbinden met pin3 en pin4 van de ACIA (Jumper W5).
De schakeling kan op een stukje VERO-board gemonteerd worden.
Als behuizing voldoet een leeg cassette-doosje uitstekend!
De kabellengten kunnen 1.5 a 3 meter bedragen.

Vragen of problemen bel R. Douwes tel 070-904006 of
H. Coenen tel 070-994566

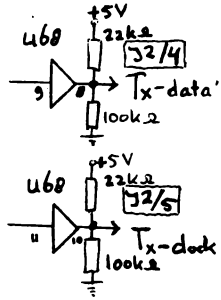
Wijzigingen bovenzijde superboard print:

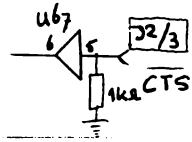




SERIE → CENTRONICS PARRALLEL INTERFACE

SUPERBOARD





6-polige DIN-CONNECTOR:

1 - NC
2 - Tx-DATA
3 - N.C.
4 - CTS
5 - Tx-clock
6 - ⏚ (0V)

TELETYPE OP SUPERBOARD.

Er moeten 2 dingen gebeuren om de teletype op het Superboard te kunnen gebruiken. Deze 2 zijn:

1. Aanpassen van de transmissiesnelheid. Dit is de snelheid waarmee de ACIA data naar buiten stuurt.
2. Aanpassen signaal niveaus. RS232 naar 20 mA Current Loop.

1) Aangenomen is dat het Superboard omgebouwd is naar 24x48 karakters. Hierbij is de kristalfrequentie 7,86 (of 8) MHz geworden. De snelheid waarmee data naar buiten wordt gestuurd wordt bepaald door de frequentie aangeboden op pin 4 v.d. ACIA en is genaamd TxClk. Deze frequentie is (bij OSI) 16x de gewenste baudrate. Zie onderstaande tabel:

BAUDRATE	FREQUENTIE	WERKELIJKE FREQUENTIE (7.86MHz)
110 Bd	1760 Hz	1754 Hz
300 Bd	4800 Hz	4723 Hz
600 Bd	9600 Hz	9446 Hz

Omgekeerd geldt ook dat de ACIA data inleest afhankelijk van de freq. op pin 3 genaamd RxClk.
Zoals op tekening 1 is te zien wordt door verandering van W5 Tx resp. RxClk ingevoerd op J2/2 en niet meer via (TxClk). Dit oude signaal wordt nu naar buiten gestuurd via J2/5. Staat de schakelaar op normaal (N) dan verandert er eigenlijk niets. Staat de schakelaar op TTY dan wordt er een 1754 Hz signaal aangeboden, waardoor de ACIA zend op 110 baud. Dit signaal wordt opgewekt door een externe frequentie deler (:70) welke clockfrequentie C5 (122,8 KHz) deelt.
Dit schema wijkt iets af van wat er in HCCN heeft gestaan. In de HCCN bleef RxClk altijd op stand N staan. Het voordeel van de nieuwe manier is dat per ongeluk op TTY (110 Bd) geSAVEde banden ook nog op stand TTY weer geLOAD kunnen worden. Bovendien is het al een gedeelte van een modificatie om b.v. het TTY toetsenbord of reader te kunnen gebruiken.

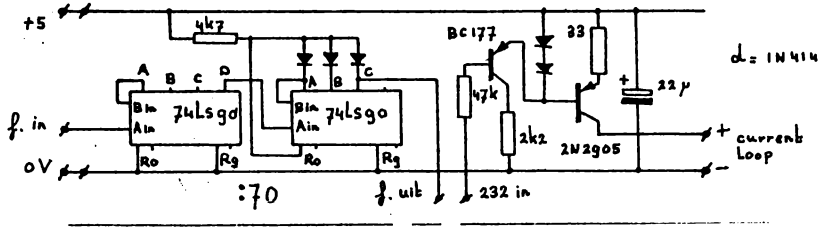
2) RS232 naar 20 mA Current Loop. Defenitie:

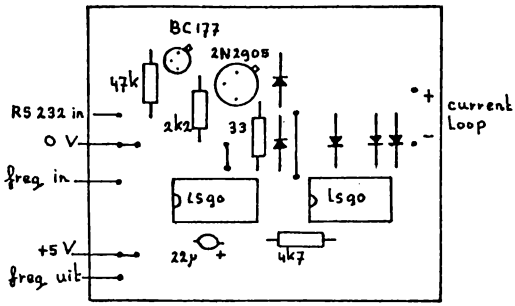
	TTL	RS232	Curr.Loop	
Logische "1"	+5V	-V	20 mA	(mark)
Logische "0"	0V	+V	0 mA	(space)

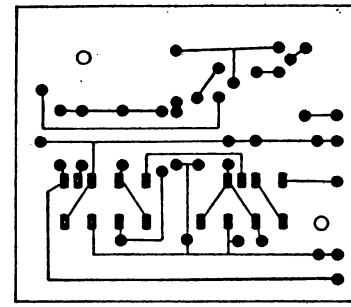
Bij RS232 dient vermeld te worden dat de spanning 'V' meestal 12 Volt is, maar in elk geval moet liggen tussen de 3 en 25 Volt.
Bij OSI ligt de logische '0' op +5V en dit voldoet aan de norm . De logische '1' komt echter niet lager dan OV en voldoet niet aan de norm. Door verbinding W10 te onderbreken en op J3/7 een negatieve spanning aan te brengen van b.v. -5V is dit ook opgelost. In de praktijk neemt de meeste apparatuur met Rs232 interface het niet zo nauw enwerkt ook zonder neg. spanning.
Dit stukje alleen ter informatie.
De zgn. Current loop is een schakeling welke eenstroom van 20 mA afgeeft onafhankelijk van de belasting:

Stroom	belastingsweerstand	spanning over bel.weerst.
20 mA	Oohm (kortsluiting)	0 V
20 mA	50 ohm	1 V
20 mA	200 ohm	4 V
??	1000 ohm	max 5 V

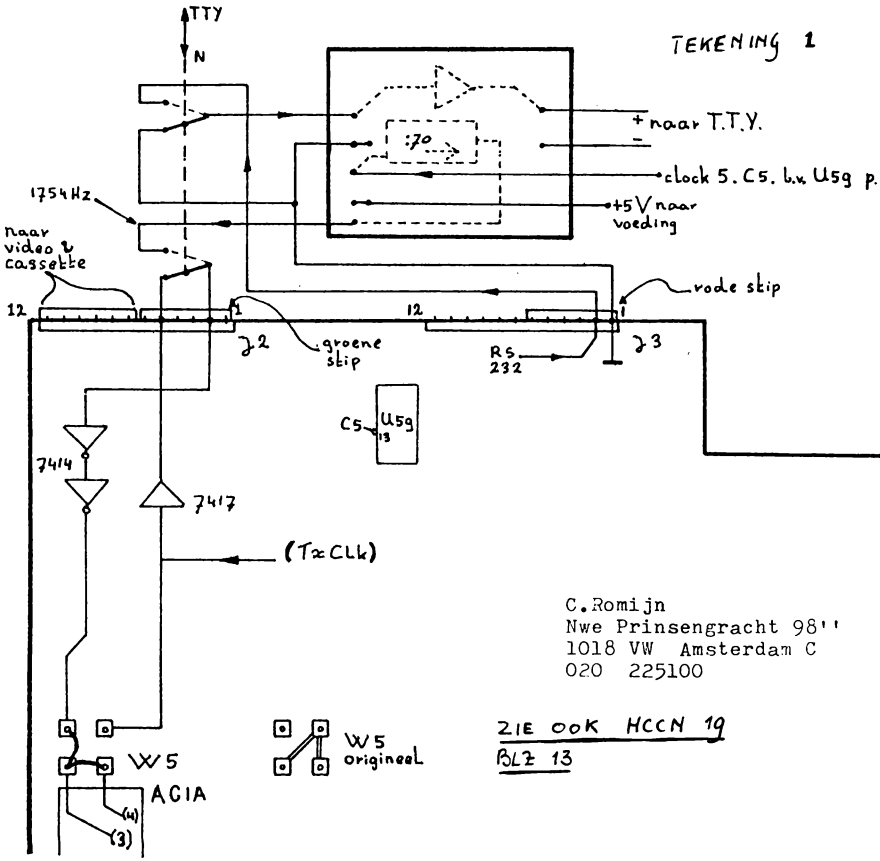
De spanning over de bel. weerst. kan nooit hoger worden dan de voedingsspanning van de stroombron, bij OSI dus 5 V. De TTY heeft een weerstand van ca 50 à 100 ohm.
De ingang van de RS232 naar 20 mA converter krijgt alleen data aangeboden in de stand TTY. Inde N stand wordt de ingang van de converter aan de OV gehangen. Hierdoor wordt er constant een mark /20 mA uitgezonden. Wanneer de schakelaar op N staat en de TTY staat aan krijgt men hierdoor geen wartaal op het papier.







TEKENING 1



C.Romijn
Nwe Prinsengracht 98''
1018 VW Amsterdam C
020 225100

ZIE OOK HCCN 19
BLZ 13

Krijg uw OSI aan de praat !

In de HCCn 37 stond een artikel over hoe men zeer eenvoudig spraaksignalen op kan slaan in het geheugen van een Apple. Het spraaksignaal werd versterkt aan de cassettepoort aangeboden waardoor het al bijna uit blokgolven bestond. De cassettepoort werd daarna heel snel achter elkaar afgevraagd en het bitpatroon werd opgeslagen in het geheugen. Vaderhand kon het dan weer afgespeeld worden over een zgn. toggleuitgang van de Apple.

Nu heeft het superboard geen geschikte cassetteingang en ook geen toggle-uitgang.De alternatieven moesten dus gevonden worden in de keyboardingang en de bell-speaker van het superboard.
We beginnen met de ingang:
Verbind de speaker uitgang van bijv. een cassetterecorder met pin 9 van J4 (links van het toetsenbord) en de massa met pin 11 van J4.
Pin 9 nu, is het hoogste bit van de keyboard-poort nl.C7 en kan daardoor makkelijker uit een byte 'gefilterd'worden dan de andere bits.

De uitgang:
Dat is de bell-speaker op het superboard. Hij wordt geaktiveerd door in het control-register van de ACIA getallen te zetten. De toggle-uitgang wordt nagebootst in software om de hardware zo eenvoudig mogelijk te houden.

Het bijstaande programmaatje komt als machinetaal-routine in het 'ongebruikte' stukje geheugen te staan en wordt vanuit BASIC aangeroepen met de USR-instruktie.
De opnameroutine begint op $Ø23Ø, de weergave op $ Ø28Ø.

```
 10 0000          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 20 0000          ;                              ;
 30 0000          ;  KRIJG UW ØSI AAN DE PRAAT  ;
 40 0000          ;                              ;
 50 0000          ; GEMAAKT DØØR HENK SCHAER     ;
 60 0000          ;                              ;
 70 0000          ; SLUIT VØØR ØPNAME DE SPEAKER;
 80 0000          ; UITGANG V/D VERSTERKER (B.V.;
 90 0000          ; DIE V/D CASSETTE) AAN AAN    ;
100 0000          ; PIN 9 VAN J4 (=C7).          ;
110 0000          ; WEERGAVE GEBEURT MET DE IN- ;
120 0000          ; TERNE 'BELL' PEAKER         ;
130 0000          ;                              ;
140 0000          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
150 0000          ;
160 0000          ;
170 0000          MEML0W=$3C  ;WERKADRES MET
180 0000          MEMHIG=$3D  ; HUIDIGE BITPATRØØN
190 0000          MAXMEM=$20  ;HØØGSTE PAGE-NR.
200 0000          KEYB  =$DF00  ;TØETSTENBØRD
210 0000          PATR  =$2F    ;ADRES PATRØØNHERKENNING KEYB.
220 0000          ACIAC =$F000  ;CØNTRØL REGISTER ACIA
230 0230               *=$0230  ;BEGINPUNT V/H PRØGRAMMA
240 0230 A93C     ØPNAME LDA #$3C
250 0232 852F          STA PATR
260 0234 A900          LDA #$00
270 0236 853C          STA MEML0W    ;ZET HET BEGINADRES ØP
280 0238 A910          LDA #$10
290 023A 853D          STA MEMHIG    ;VANAF DE 4K GRENS
300 023C A000          LDY #$00
310 023E 205802   NEXTCH JSR HAALØP
320 0241 913C          STA (MEML0W),Y  ;ZET IN GEHEUGEN WEG
330 0243 E63C          INC MEML0W
340 0245 D009          BNE VERTR1    ;GEEN NIEUWE PAGE
350 0247 E63D          INC MEMHIG    ;WEL VØLGENDE PAGE
360 0249 A53D          LDA MEMHIG    ;GA VERDER MITS NIET BØVEN
370 024B C920          CMP #MAXMEM   ;HET HØØGSTE RAM-GEBIED
380 024D D0EF          BNE NEXTCH
390 024F 60            RTS           ;ANDERS NAAR BASIC-CALLER
400 0250 4C5302   VERTR1 JMP VØLG1     ;EEN KLEINE VERTRAGING
410 0253 EA       VØLG1  NØP
420 0254 EA              NØP
430 0255 EA              NØP
440 0256 D0E6            BNE NEXTCH    ;SPRING ALTIJD
450 0258 A208     HAALØP LDX #$08      ;AANTAL BITS
460 025A D008            BNE NEXTBI    ;SPRING ALTIJD
470 025C A007     VERTR2 LDY #$07
480 025E 88       VØLG2  DEY
490 025F D0FD            BNE VØLG2     ;WEER EEN VERTRAGING
500 0261 EA              NØP
510 0262 EA              NØP
520 0263 EA              NØP
530 0264 48       NEXTBI PHA           ;RØL VØLGENDE CARRYBIT IN
540 0265 206E02          JSR CARRYB    ;GA DAT BIT ØPHALEN
550 0268 68              PLA           ;HAAL ØUDE SITUATIE ØP
560 0269 2A              RØL A         ;NEEM BIT IN LSB ØP
570 026A CA              DEX           ;WEER EEN BIT AF
```

```
 580 026B D0EF           BNE VERTR2   ;NAAR VØLGEND BITT
 590 026D 60             RTS          ;ANDERS BYTE AF
 600 026E AD00DF CARRYB  LDA KEYB     ;HAAL EEN CARRYBIT ØP
 610 0271 452F           EØR PATR     ;PULS ØP J4/C7 ?
 620 0273 1003           BPL NEECAR   ;NEE, ZET CARRY=0
 630 0275 38             SEC          ;JA, ZET CARRY=1
 640 0276 B002           BCS JACARR   ;SPRING ALTIJD
 650 0278 18       NEECAR CLC         ;ZET CARRY=0
 660 0279 EA             NØP
 670 027A 452F     JACARR EØR PATR    ;HERSTEL PATRØØN
 680 027C 852F           STA PATR     ;ØP DE JUISTE PLAATS
 690 027E 60             RTS
 700 0280              *=$0280
 710 0280 A93C     WEERGA LDA #$3C
 720 0282 852F           STA PATR
 730 0284 A900           LDA #$00
 740 0286 853C           STA MEML0W   ;GA WEER BEGINADRES
 750 0288 A910           LDA #$10
 760 028A 853D           STA MEMHIG   ; ØPBØUWEN
 770 028C A000           LDY #$00
 780 028E B13C     NEXTBY LDA (MEML0W),Y  ;HAAL VØLGEND BYTTE ØP
 790 0290 20A802         JSR ZETWEG   ;ZET DAT ØP DE SPEAKER
 800 0293 E63C           INC MEML0W   ;VØLGENDE GEHEUGENPLAATS
 810 0295 D009           BNE VERTR3   ;GEEN NIEUWE PAGE=SPRINGEN
 820 0297 E63D           INC MEMHIG   ;WEL EEN NIEUWE PAGE
 830 0299 A53D           LDA MEMHIG   ;MAG NIET BØVEN HØØGSTE
 840 029B C920           CMP #MAXMEM  ; GEHEUGENPLAATS
 850 029D D0EF           BNE NEXTBY   ;ZØNIET VØLGENDE BYTE
 860 029F 60             RTS          ;ANDERS KLAAR !!
 870 02A0 4CA302   VERTR3 JMP VØLG3   ;VERTRAGING
 880 02A3 EA       VØLG3  NØP
 890 02A4 EA              NØP
 900 02A5 EA              NØP
 910 02A6 D0E6           BNE NEXTBY   ;SPRING ALTIJD TERUG
 920 02A8 A208     ZETWEG LDX #$08     ;AANTAL BITS
 930 02AA D008           BNE VERTR5   ;SPRING ALTIJD
 940 02AC A006     VERTR4 LDY #$06     ;WEER EEN VERTRAGING
 950 02AE 88       VØLG4  DEY
 960 02AF D0FD           BNE VØLG4
 970 02B1 EA             NØP
 980 02B2 EA             NØP
 990 02B3 EA             NØP
1000 02B4 A003     VERTR5 LDY #$03     ;KØRTE VERTRAGING
1010 02B6 88       VØLG5  DEY
1020 02B7 D0FD           BNE VØLG5
1030 02B9 2A             RØL A         ;LEES BIT/BIT IN CARRY
1040 02BA 900D           BCC GEENCA    ;CARRYBIT = 0
1050 02BC 48             PHA           ;CARRY=1,BEWAAR REST
1060 02BD A52F           LDA PATR
1070 02BF 4960           EØR #$60
1080 02C1 852F           STA PATR
1090 02C3 8D00F0         STA ACIAC    ;NAAR SPEAKER
1100 02C6 68             PLA
1110 02C7 B00A           BCS WELCAR    ;SPRING ALTIJD
1120 02C9 A42F     GEENCA LDY PATR
```

```
1130 02CB 8C00F0          STY ACIAC     ;NAAR SPEAKER
1140 02CE A003      VERTR6 LDY #$03
1150 02D0 88        VØLG6  DEY
1160 02D1 D0FD             BNE VØLG6
1170 02D3 CA        VELCAR DEX
1180 02D4 EA               NØP
1190 02D5 EA               NØP
1200 02D6 D0D4             BNE VERTR4    ;NØG NIET ALLE BITS GEHAD
1210 02D8 60               RTS           ;ALLE BITS AFGEWERKT
```

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0230   A9 3C 85 2F A9 00 85 3C A9 10 85 3D A0 00 20 58
0240   02 91 3C E6 3C D0 09 E6 3D A5 3D C9 20 D0 EF 60
0250   4C 53 02 EA EA EA D0 E6 A2 08 D0 08 A0 07 88 D0
0260   FD EA EA EA 48 20 6E 02 68 2A CA D0 EF 60 AD 00
0270   DF 45 2F 10 03 38 B0 02 18 EA 45 2F 85 2F 60 00
0280   A9 3C 85 2F A9 00 85 3C A9 10 85 3D A0 00 B1 3C
0290   20 A8 02 E6 3C D0 09 E6 3D A5 3D C9 20 D0 EF 60
02A0   4C A3 02 EA EA EA D0 E6 A2 08 D0 08 A0 06 88 D0
02B0   FD EA EA EA A0 03 88 D0 FD 2A 90 0D 48 A5 2F 49
02C0   60 85 2F 8D 00 F0 68 B0 0A A4 2F 8C 00 F0 A0 03
02D0   88 D0 FD CA EA EA D0 D4 60
:
```

Het volgende BASIC programma start de routines.

```
10 PRINT CHR$(26)
20 POKE 11,48 : POKE 12,2 : REM beginadres opnameroutine
30 PRINT" Opname.......";  : FOR T = 1 TO 1000 : NEXT
40 PRINT" ........NU !!" : X=USR(X) : REM roep opname op
50 PRINT" Weergave....." : POKE 11,128 : POKE 12,2 : REM adres weerg.
60 X=USR(X) : REM roep weergave op
70 GOTO 10
```

De regels 10 tm 40 zijn voor de opname, 50 tm 60 voor de weergave.
De spraaksignalen worden opgeslagen vanaf 4096_{dec} tm 8191_{dec}
en beslaan dus 4K ram,maak uw programma's dus niet te lang.

Henk Schaer (01727 - 6536)
Reijerskoop 55
2771 BE Boskoop

UITBREIDING/VERANDERINGEN AAN ELEKTUUR-FLOPPY-DISC-INTERFACE

(GUIDO DE CUYPER) T. B. V. TOEPASSING MET OSI SUPERBOARD II / C1

DOOR CEES VAN LUIJPEN.

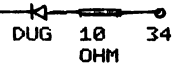
HET ELECTUUR-PRINTJE IS, EVENALS DE "OUDERE" PRINTJES VAN GUIDO, NIET ZONDER MEER OP EEN SUPERBOARD AAN TE SLUITEN. ONDERSTAANDE BESCHRIJVING GEEFT EEN OVERZICHT VAN DE MOGELIJK AAN TE BRENGEN WIJZIGINGEN EN UITBREIDINGEN OP DIT, OVERIGENS VOORTREFFELIJKE, INTERFACE-PRINTJE.

A) CREEREN DD-SIGNAAL:

A1) SNIJ DOOR VERBINDING C6 (CONNECTOR) - IC 13. 19
A2) SNIJ DOOR VERBINDING IC 8. 9 - IC 8. 10
A3) SNIJ DOOR VERBINDING IC 12. 5 - RESTERENDE POOTJES IC 12
A4) PLAATS OP IC 12 (SN7407N) EEN SN7400N
DIT WORDT IC 12A
BUIG DE POOTJES 1, 2, 3, 4, 5, 8, 9, 10, 11, 12 EN 13 ENIGSZINS NAAR BUITEN
A5) VERBIND IC 12A. 7 MET IC 12. 7
1C 12A. 14 MET IC 12. 14, EN LET OP !!!
---> IC 12A. 6 MET IC 12. 5 <---
A6) LEG EEN DRAAD TUSSEN IC 12. 6 EN C6 (CONNECTOR)
A7) LEG EEN DRAAD TUSSEN IC 5. 22 EN IC 12A. 5
A8) LEG EEN DRAAD TUSSEN IC 8. 11 EN IC 12A. 4

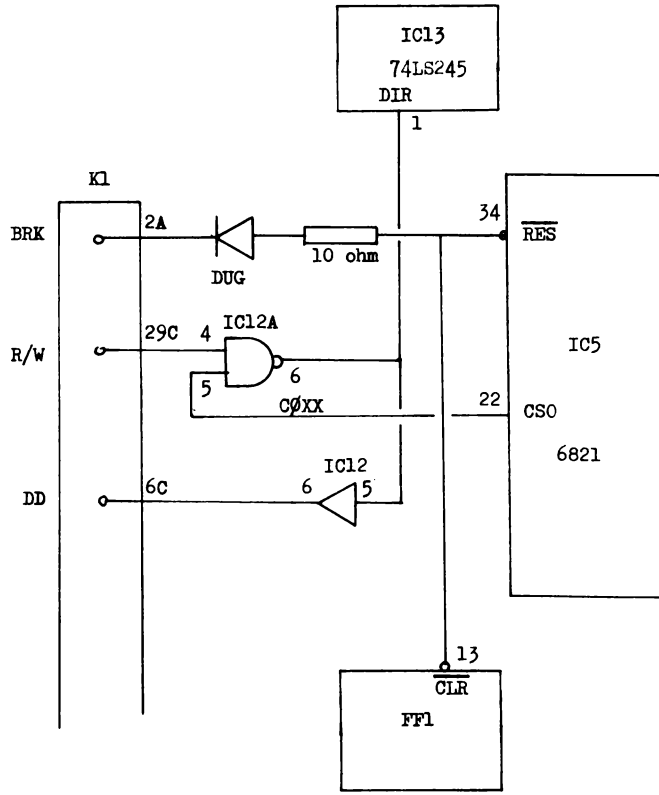
B) RESET SELECTIE DRIVE(S) DOOR <BREAK>

B1) OP SUPERBOARD: LEG EEN DRAAD TUSSEN U8. 40 (PROCESSOR) EN J1. 11 (EXPANSION CONNECTOR)
B2) BEKABELING: LEG EEN VERBINDING VAN J1. 11 NAAR DE FLOPPY-CONNECTOR
B3) OP FLOPPY-KAART: VERBIND CONNECTOR PUNT A2 VIA EEN GE-DIODE MET IN SERIE EEN WEERSTAND VAN 10 OHM MET IC 5.34

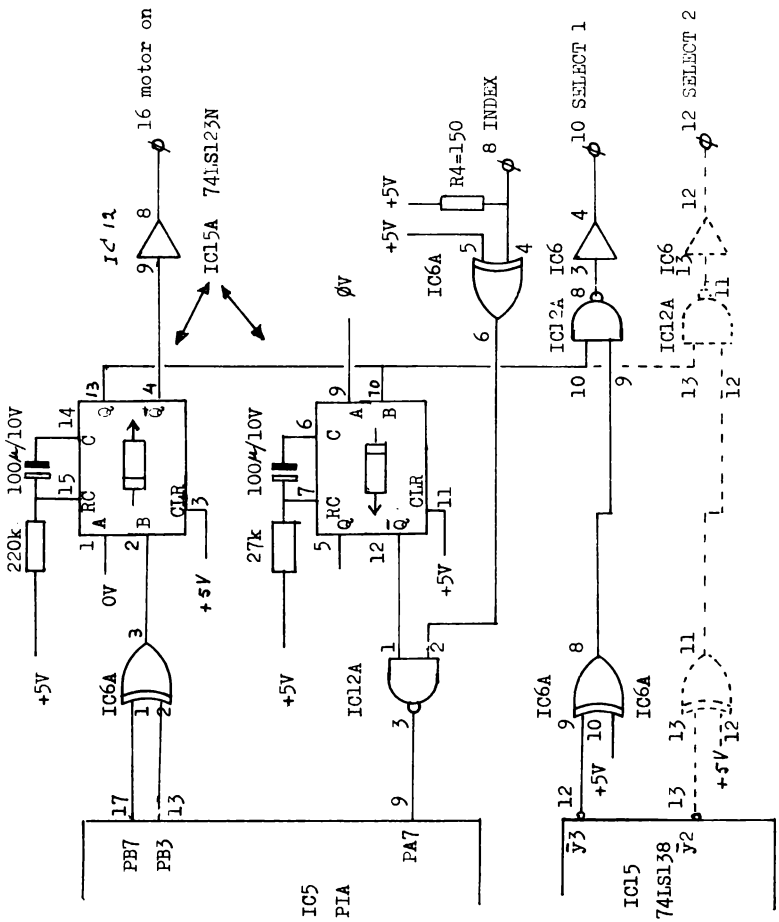


B4) SNIJ DOOR IC 14. 14 EN IC 14. 13
B5) VERBIND IC 14. 13 MET IC 5. 34

BRON: N. E. INGERSOLL: THE DISK SWITCH; MICRO 9/81 P. 15-17
1. REFERENTIES TYPENUMMER CONFORM ELEKTUUR
2. IC 12A SECTIE 5-6-7 REEDS GEBRUIKT VOOR DD-LIJN.
3. IC 6A : 74LS86N
IC 12A: 74LS00N
IC 15A: 74LS123N

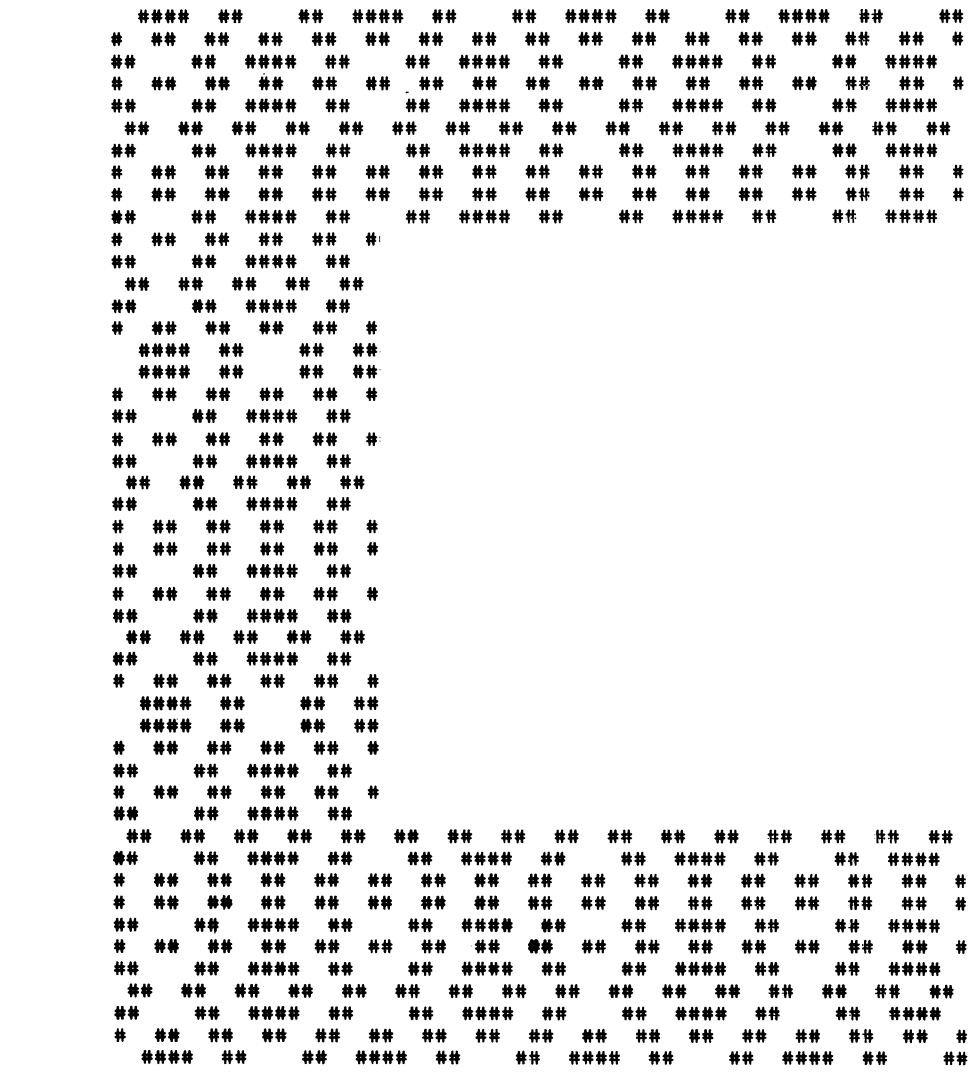


Wijzigingen Elektuur disk drive interface



D I S C M O T O R S C H A K E L A A R K. van Luijpen

Bron: MICRO september 1981

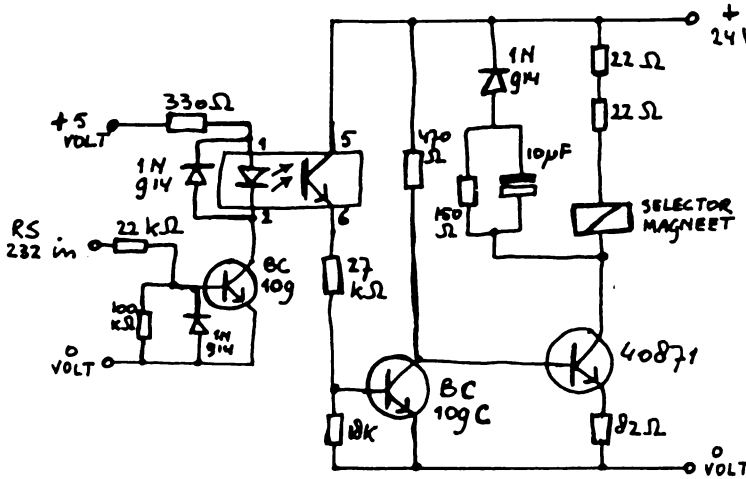


FEBRUARI 1983

EEN OPTISCHE KOPPELING TUSSEN UW SUPERBOARD EN UW PRINTER

C. van Luipen heeft voor J. Burghouts de onderstaande optische koppeling gemaakt. Daarmee wordt de storingrijke Teletype printer elektrisch volledig gescheiden van het superboard. Vroeger werd het RS232 signaal rechtstreeks via de 27K weerstand naar de basis van de BC 109C gestuurd. De enige toevoeging is dus de +24 Volt aansluiting aan pin 5 van de optocoupler.
In een drie-aderige kabel (PTT bijv.) wordt van het SB het +5V, het RS232 signaal en het aardesignaal gestuurd.

Al meer dan één jaar werkt de schakeling volledig foutloos !!!!



HET 'DISK OPERATING SYSTEM'.

EEN MICROCOMPUTER HEEFT NAAST HET REKEN- EN BESTURINGSORGAAN ALTIJD MINSTENS EEN TOETSENBORD EN EEN DISPLAY UNIT (B.V. EEN VIDEO MONITOR).
HET TOETSENBORD IS EEN INVOER- EN HET DISPLAY EEN UITVOERORGAAN. ELK APPARAAT, DAT VOOR INVOER OF UITVOER ZORGT, STAAT MET HET BESTURINGSORGAAN IN VERBINDING VIA EEN INTERFACE (HARDWARE) OOK WEL CONTROLLER GENOEMD EN WORDT BESTUURD MET EEN PROGRAMMA (SOFTWARE).
ZO OOK DE DISK DRIVE.
IN DE HCCN 8 STAAT EEN ZEER LEERZAAM ARTIKEL VAN GERRIT SLOT OVER DE FLOPPY DISK CONTROLLER.
DE FILES (PROGRAMMA'S EN DATA) WORDEN SERIEEL OP DISK GESCHREVEN MET EEN SNELHEID VAN 125 KBAUD (125000 BITS/SEC).
OM TECHNISCHE REDENEN WORDEN DE DATA BITS VOLGENS HET PRINCIPE VAN FREQUENTIE MODULATIE GEMENGD MET KLOKSIGNALEN (CLOCK BITS) MET EEN FREQUENTIE VAN 125 KHZ. DEZE CLOCK BITS LIGGEN TUSSEN DE DATA BITS IN.
DE KLOKSIGNALEN WORDEN GEGENEREERD BIJ HET SCHRIJVEN NAAR DISK, ELKE 8 USEC EEN PULS VAN CA 1 USEC.
BIJ HET LEZEN VAN DISK MOETEN DEZE TWEE TYPEN SIGNALEN WEER GESCHEIDEN WORDEN. DIT GEBEURT MET BEHULP VAN EEN DATA SEPERATOR.
ELKE BYTE, DIE OVERGESEIND WORDT, BEGINT MET EEN STARTBIT (0) EN EINDIGT MET EEN PARITY BIT (IS DE SOM VAN DE BITS IN DE BYTE EVEN DAN IS BIJ AFSPRAAK HET PARITY BIT B.V. 0)
AAN HET EINDE VAN DE BYTE BEVINDT ZICH NOG HET STOPBIT (1).
OM EEN BYTE OVER TE SEINEN ZIJN ER DUS 11 BITS NODIG. DAARBIJ KOMEN DAN NOG EENS DE 11 CLOCK BITS.
DE O.S.I. -DOS DOET HET ZO.
EEN ANDERE MOGELIJKHEID ZOU ZIJN OM IN PLAATS VAN PARITY BITS OM DE ZOVEEL BYTES EEN OF MEER CONTROLE BYTES TE PLAATSEN, B.V. NA ELKE PROGRAMMAREGEL TWEE BYTES ZOU GENOEG ZIJN, OMDAT BASIC VOOR EEN GROOT DEEL UIT ASCII-TEKENS (7 BITS VAN DE BYTE KUNNEN 1 ZIJN) BESTAAT, ZEKER ALS DE REGEL NIET MEER DAN 72 KARAKTERS MAG BEVATTEN, ZOALS BIJ HET SUPERBOARD HET GEVAL IS.
HOE ZO'N DISK INTERFACE WERKT, IS O.A. TE LEZEN IN DE ARTIKELEN VAN GUIDO DE CUYPER IN DE ELEKTUUR-NUMMERS VAN NOVEMBER EN DECEMBER 1982. DE INTERFACE, DIE DAARBESCHREVEN STAAT, IS AFGELEID VAN HET PRINTJE, DAT GUIDO ENIGE TIJD GELEDEN OP DE OSI-DAGEN HEEFT GEDEMONSTREERD.
HOE HET ELEKTUUR INTERFACE GESCHIKT GEMAAKT KAN WORDEN VOOR HET SUPERBOARD, KUNT U LEZEN IN HET ARTIKEL VAN CEES VAN LUYPEN OP PAGINA CC 9 E.V.
DE DISK-ORGANISATIE VAN OS65D WORDT BESCHREVEN DOOR T.R. BERGER (A SMALL OPERATING SYSTEM: OS65D, THE DISK ROUTINES) IN COMPUTE! VAN JANUARI EN FEBRUARI 1982. DEZE ARTIKELEN GEVEN EEN GOED BEELD VAN DE WIJZE WAAROP OS65D WERKT.
EEN COMPILATIE VAN DE WIJZE, WAAROP DE DISKS BESCHREVEN WORDEN, VINDT U IN DE HIERNA VOLGENDE TABEL.

DISK-ORGANISATIE VAN OS65D

T R A C K 1 ETC.

DATA-CLOCK IS VOOR ALLE SYSTEMEN (0. S. I.) HETZELFDE.

TRACK START (NA 'INDEX HOLE')

		CUMULATIEF
WACHT	1000 US	1000 US 2)
BYTES: $43 EN $57	88	1088
TRACK NUMMER	44	1132
STOP BYTE $58	44	1176
WACHT 4*1600	6400	7576

TRACK EINDE (NA LAATSTE SECTOR)

ERASE + WRITE CONTINUE	600*R	'TRAILING TIME' 1)
ERASE CONTINUE	525	

SECTOR-BEZETTING

BYTE $76	44	44
SECTOR NR (BCD)	44	88
AANTAL PAGINA'S (P)	44	132
DATA 44*256 + P =	11264*P	11246*P + 132
BYTES: $47 EN $53	88	11246*P + 220
WACHTTIJD	1600*P+215	12864*P + 435

DE TOTALE 'RECORDING TIME' T IN US:

T = 7576 + 600*R + 525 + 11264*Q + 435*N + (Q-R)*1600 1)

T = 8101 + 12864 * Q - 1000 * R + 435 * N

T R A C K 0

WACHT 1000 US
HIGH BYTE) TRACK ZERO WORDT
LOW BYTE) HIER GELADEN
AANTAL PAGINA'S IN TRACK 0
DATA

TRACK 0 WORDT BIJ 'BOOT'-EN IN RAM GELADEN TE BEGINNEN
BIJ $2200 OF EEN ANDER INGELEZEN STARTADRES.
DE COMPUTER SPRINGT HIERNA NAAR HET OPGEGEVEN ADRES.

1) R = AANTAL PAGINA'S IN LAATSTE SECTOR
N = AANTAL SECTOREN OP DE TRACK
P = AANTAL PAGINA'S OP DE VORIGE SECTOR
Q = AANTAL PAGINA'S OP DE TRACK

2) BEPAALD DOOR MICROPRESSOR-CLOCK.

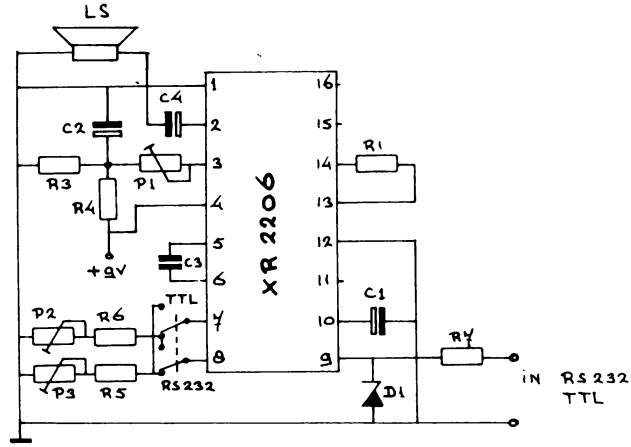
contra viditel modem

John Glaser

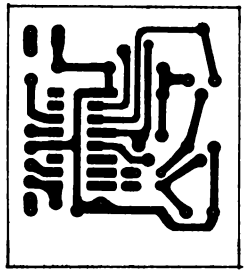
R1 = 200Ω 1%
R3,4 = 4k7
R5 = 27k
R6 = 15k
R7 = 270Ω

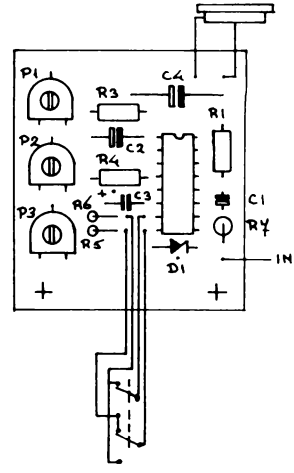
C1 = 1µf 16V
C2 = 10µf 16V
C3 = 22 Nf
C4 = 47µf 16V

D1 = 8,2V 400mW



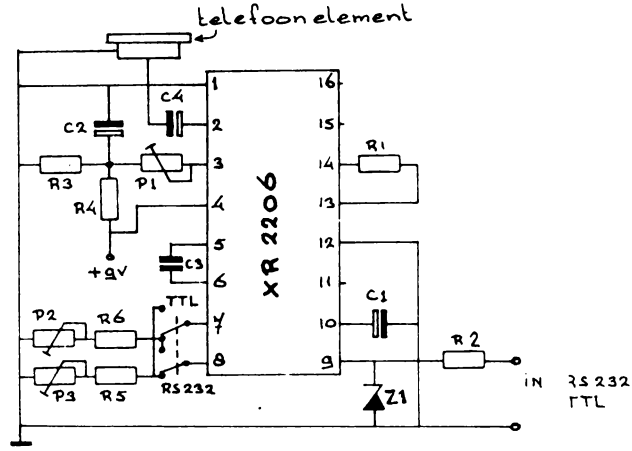
P1 volume
P2 2100 HZ ingang open
P3 1300 HZ " kort gesloten
RS 232





NOVEMBER 1982

FSK-Modulator

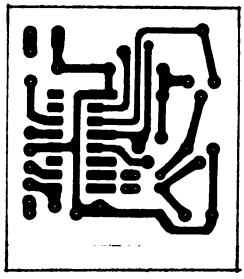


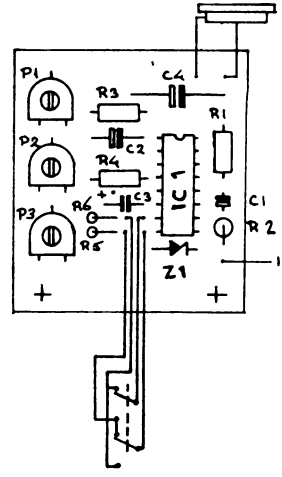
R1 = 200 Ω 1%
R3,4 = 4k7
R5 = 27k
R6 = 15k
R2 = 270 Ω

C1 = 1 µf 16V
C2 = 10µf 16V
C3 = 22 Nf
C4 = 47µf 16V

Z1 = 8,2 V 400mW
P1, P2, P3 = 10 K lin
Telefoonel. = 350 Ω
IC 1 = XR 2206

P1 volume
P2 2100 HZ ingang open
P3 1300 HZ " kort gesloten } : S 232





FEBRUARI 1983

Hobby Computer Club 30155236a 1c

***** Contra Viditel Modem *****

Een idee van John Glaser.Tel.033-751472

Dit is een acoustisch modem,dat dezefde frequenties uitstuurt als de Viditel-computer.
Het is te gebruiken voor het oversturen van computerprogramma's en werkt enkel-zijdig.(simplex)
De bouwkosten zijn erg laag omdat alleen de modulator(=zender) is uitgevoerd.
Als demodulator(=ontvanger) wordt het Viditel-modem gebruikt.
De te gebruiken baudrates zijn vrij tot een maximum van 1200 Baud.
Iedere computer die over een RS-232 uit-gang beschikt,kan hierop aangesloten worden.Ook computers met een uitgang op TTL-niveau zijn hier te gebruiken.

Hobby Computer Club 30155236b 1c

Vervolg Contra Viditel modem:

Gekozen is voor een 9-polige D connector die pin-compatible is met de PTT modem.
Een kwestie van een steker verplaatsen!
De modem werkt op een 9 Volt alkaline-batterij,wat voldoende is voor 100 uur.
Verder is het apparaat voorzien van drie schakelaars S1,S2 en S3.(Resp. Aan/Uit, batterij test en RS232/TTL)
Aan de bovenkant bevindt zich een rubber manchet waar het microfoon-gedeelte van de hoorn ingeklemd wordt.

De gebruikte frequenties zijn:

1300 Hz = MARK "1"

2100 Hz = SPACE"0"

Hobby Computer Club 30155236c 1c

Vervolg Contra Viditel modem

Het niveau op de telefoonlijn mag niet hoger als -6dB zijn.(imp.600 Ohm)

Als het apparaat gebouwd is moet een type-goedkeuring aangevraagd worden bij de afd.CATG van de Centrale Directie van PTT in Den Haag. Kosten 100 gulden.

De bouwkosten bedragen zo'n 50 gulden.

Op de bladzijden die hierna volgen staan een schema en aansluitgegevens.

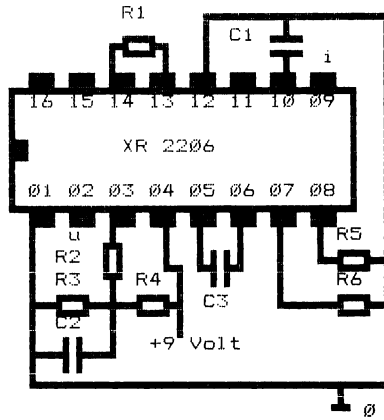
Hobby Computer Club 30155236e 1c

Aansluitingen IC XR 2206 voor FSK-modem

Punt Ø1 Ø Volt (massa)
,, Ø2 FSK uit
,, Ø3 naar R2=47K(instelpotmeter)
,, Ø4 +9 Volt
,, Ø5 naar C3=22nF
,, Ø6 naar C3 (zie Ø5)
,, Ø7 naar R6=15K+1ØK(instelpotmeter)
,, Ø8 naar R5=27K+1ØK(instelpotmeter)
,, Ø9 RS232/TTL in
,, 1Ø naar C1=1uF(elco)
,, 11 niet aangesloten
,, 12 Ø Volt (massa)
,, 13 naar R1=2ØØ Ohm(1%)
,, 14 naar R1 (zie 13)
,, 15 niet aangesloten
,, 16 niet aangesloten
R3=R4=4K7
C2=1Ø uF(elco)

Hobby Computer Club 30155236d 1c

FSK-modulator



Hobby Computer Club 30155236f Øc

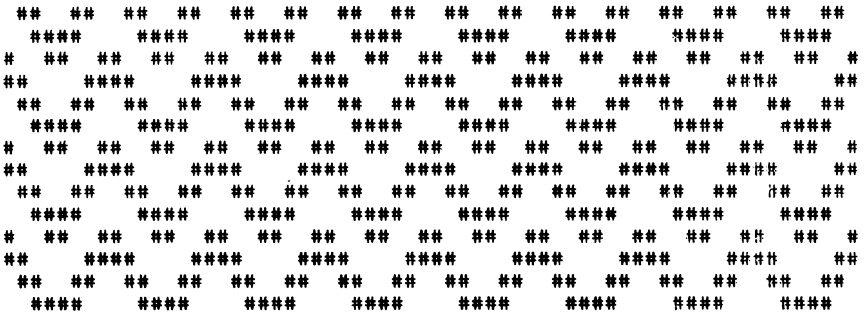
Punt Ø9 is de ingang en wordt beveiligd me een weerstand van 27Ø Ohm,gevolgt door een zenerdiode van 8,2 Volt naar massa.(anode aan massa)

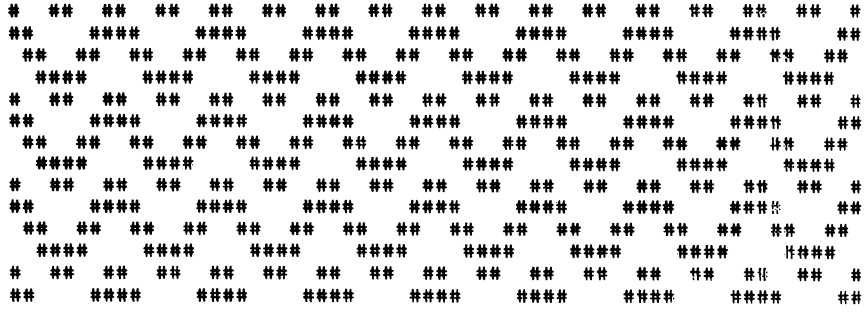
Punt Ø2 is de uitgang en gaat via een elco van 47 uF (+aan punt Ø2) naar een telefoonelement (zo'n ding wat in een hoorn zit).

Acoüstisch geeft 12ØØ Baud problemen.

Aanbevolen snelheid is 3ØØ Baud.

Veel succes John Glaser.





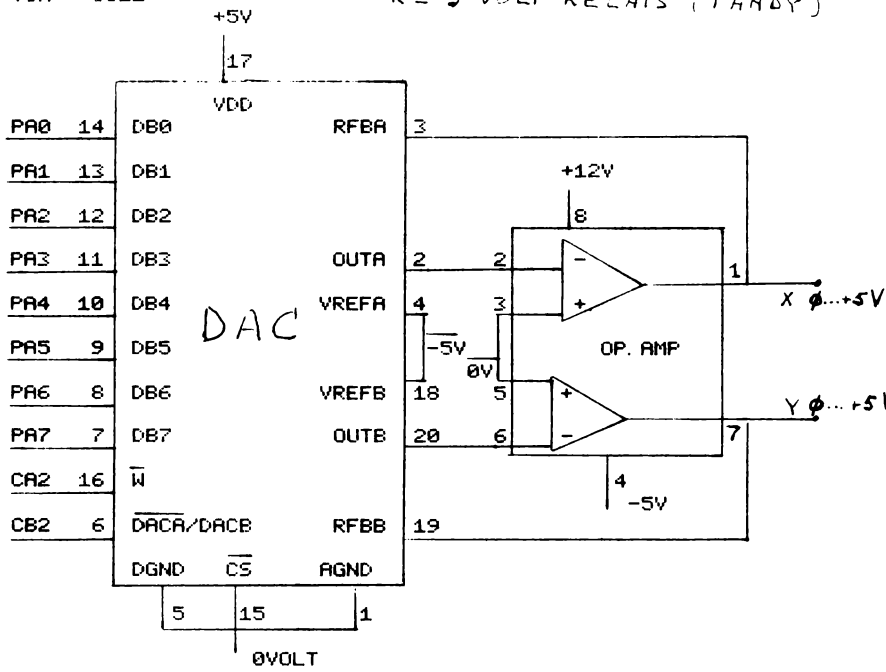
FEBRUARI 1983

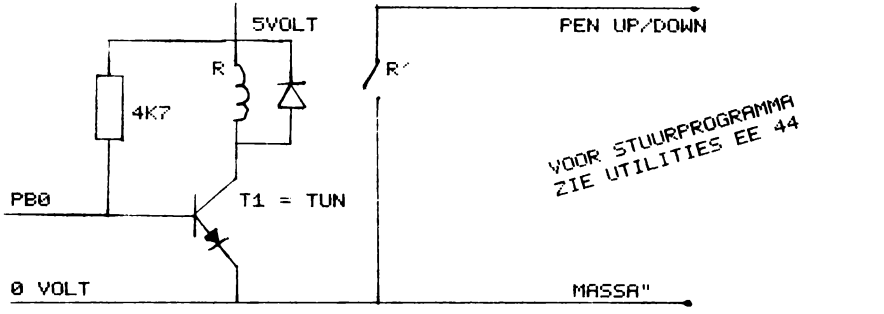
DUAL D/A CONVERTER MET BUFFER

O.A. VOOR AANSTURING VAN X-Y SCHRIJVER
RESOLUTIE 8 BITS (= 256 STAPPEN)

DAC = AD7528 VAN ANALOGE DEVICES TEL 01620-51080 (FL 35.-)
OP. AMP = 1458
VIA = 6522

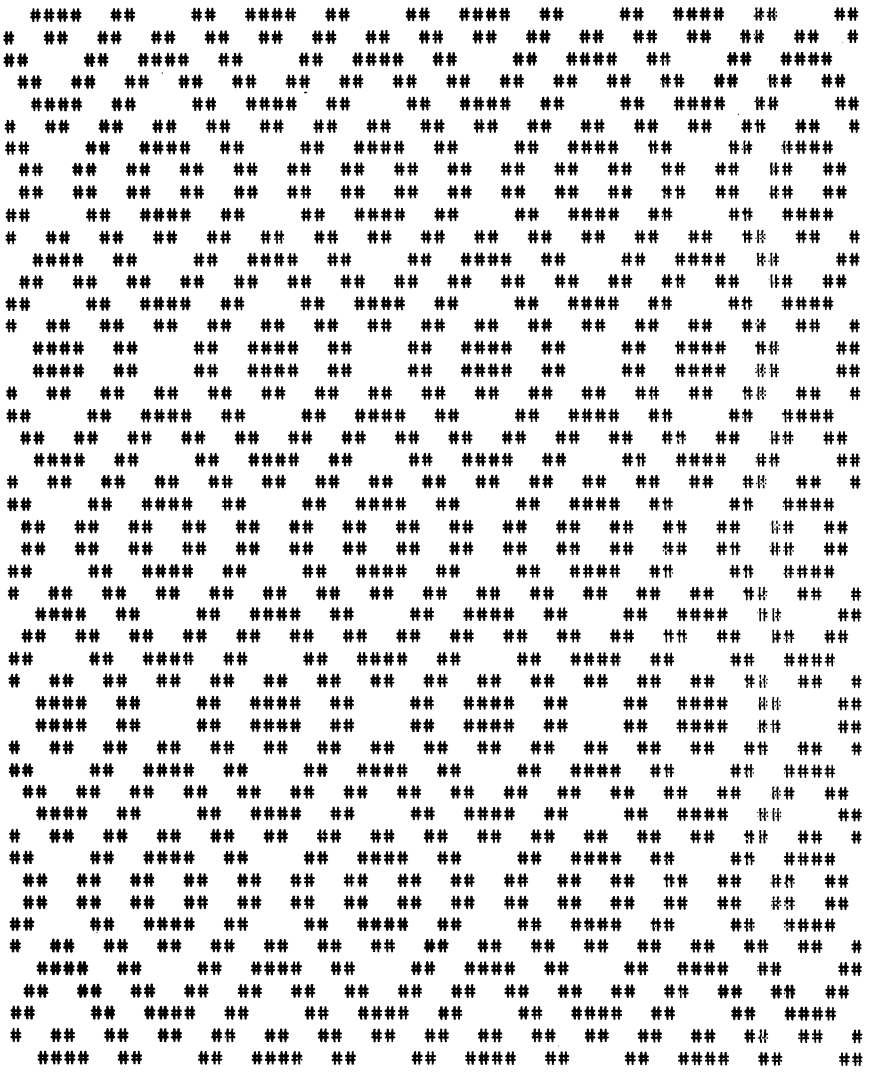
R = 5 VOLT RELAIS (FANDY)





CA2 GEEFT NEG PULS NA EEN WRITE NAAR DE VIA.
CB2 = "0" (POKE VIA+12,202) ="1" (POKE VIA+12,234) BIT 5 !
PB0 "0" = PEN UP "1" = PEN DOWN

FEBRUARI 1983 JOHN GLASER 033-751472



FEBRUARI 1983

D FIRMWARE

DA OSI Monitor

DB Microsoft Basic

DC H.W. Monitor

DD Toolkit - Assembler Editor - Ext. Monitor

DE Text Editor

OSI MICROSOFT BASIC IN ROM

DE VOLGENDE BESCHRIJVING VAN DE OSI MICROSOFT BASIC IN ROM IS GEMAAKT DOOR HENK VEVERS.
DE OORSPRONKELIJK ENGELSTALIGE TEKST IS DOOR JOS BURGHOUTS VERTAALD EN VAT BIJGEWERKT.

TABEL 1 SPRONGADRESSEN VAN BASIC OPDRACHT-ROUTINES

DE VERMELDE ADRESSEN (LINKS) ZIJN MET 1 VERLAAGD, OMDAT DE ROUTINES VIA EEN <RTS> VORDEN OPGEROEPEN

A000- 39 A6	.$A639 END
A002- 55 A5	.$A556 FOR
A004- 3F AA	.$AA40 NEXT
A006- 0B A7	.$A70C DATA
A008- 22 A9	.$A923 INPUT
A00A- 00 AD	.$AD01 DIM
A00C- 4E A9	.$A94F READ
A00E- B8 A7	.$A7B9 LET
A010- B8 A6	.$A6B9 GOTO
A012- 90 A6	.$A691 RUN
A014- 3B A7	.$A73C IF
A016- 19 A6	.$A61A RESTORE
A018- 9B A6	.$A69C GOSUB
A01A- E5 A6	.$A6E6 RETURN
A01C- 4E A7	.$A74F REM
A01E- 37 A6	.$A638 STOP
A020- 5E A7	.$A75F ON
A022- 7A A6	.$A67B NULL
A024- 31 B4	.$B432 WAIT
A026- F3 FF	.$FFF4 LOAD
A028- F6 FF	.$FFF7 SAVE
A02A- DD AF	.$AFDE DEF
A02C- 28 B4	.$B429 POKE
A02E- 2E A8	.$A82F PRINT
A030- 60 A6	.$A661 CONT
A032- B4 A4	.$A4B5 LIST
A034- 8B A6	.$A68C CLEAR
A036- 60 A4	.$A461 NEW

TABLE 2 VECTORS FOR FUNCTIONS

```
A038- D8 B7                    .$B7D8 SGN
A03A- 62 B8                    .$B862 INT
A03C- F5 B7                    .$B7F5 ABS
A03E- 0A 00                    .$000A USR
A040- AD AF                    .$AFAD FRE
A042- CE AF                    .$AFCE POS
A044- AC BA                    .$BAAC SQR
A046- C0 BB                    .$BBC0 RND
A048- BD B5                    .$B5BD LOG
A04A- 1B BB                    .$BB1B EXP
A04C- FC BB                    .$BBFC COS
A04E- 03 BC                    .$BC03 SIN
A050- 4C BC                    .$BC4C TAN
A052- 99 BC                    .$BC99 ATN
A054- 1E B4                    .$B41E PEEK
A056- 8C B3                    .$B38C LEN
A058- 8C B0                    .$B08C STR$
A05A- BD B3                    .$B3BD VAL
A05C- 9B B3                    .$B39B ASC
A05E- FC B2                    .$B2FC CHR$
A060- 10 B3                    .$B310 LEFT$
A062- 3C B3                    .$B33C RIGHT$
A064- 47 B3                    .$B347 MID$
```

TABLE 3 VECTORS FOR OPERATORS AND THEIR PRIORITIES
(addresses -1 because accessed by RTS)

```
A066- 79                       .
A067- 6E B4                    .$B46F +
A069- 79                       .
A06A- 57 B4                    .$B458 -
A06C- 7B                       .
A06D- FD B5                    .$B5FE *
A06F- 7B                       .
A070- CC B6                    .$B6CD /
A072- 7F                       .
A073- B5 BA                    .$B5BB ^ BABB
A075- 50                       .
A076- 68 AC                    .$AC69 AND
A078- 46                       .
A079- 65 AC                    .$AC66 OR
A07B- 7D                       .
A07C- EE BA                    .$BAEF NEGATE(-)
A07E- 5A                       .
A07F- D7 AB                    .$ABD8 NOT
A081- 64                       .
A082- 95 AC                    .$AC96 < = >
```

FEBRUARI 1983

TABLE 4 KEYWORDS

(last charcter has bit 7 set)

```
A084- 45 4E C4                .80 END
A087- 46 4F D2                .81 FOR
A08A- 4E 45 58 D4             .82 NEXT
A08E- 44 41 54 C1             .83 DATA
A092- 49 4E 50 55 D4          .84 INPUT
A097- 44 49 CD                .85 DIM
A09A- 52 45 41 C4             .86 READ
A09E- 4C 45 D4                .87 LET
A0A1- 47 4F 54 CF             .88 GOTO
A0A5- 52 55 CE                .89 RUN
A0A8- 49 C6                   .8A IF
A0AA- 52 45 53 54 4F 52 C5    .8B RESTORE
A0B1- 47 4F 53 55 C2          .8C GOSUB
A0B6- 52 45 54 55 52 CE       .8D RETURN
A0BC- 52 45 CD                .8E REM
A0BF- 53 54 4F D0             .8F STOP
A0C3- 4F CE                   .90 ON
A0C5- 4E 55 4C CC             .91 NULL
A0C9- 57 41 49 D4             .92 WAIT
A0CD- 4C 4F 41 C4             .93 LOAD
A0D1- 53 41 56 C5             .94 SAVE
A0D5- 44 45 C6                .95 DEF
A0D8- 50 4F 4B C5             .96 POKE
A0DC- 50 52 49 4E D4          .97 PRINT
A0E1- 43 4F 4E D4             .98 CONT
A0E5- 4C 49 53 D4             .99 LIST
A0E9- 43 4C 45 41 D2          .9A CLEAR
A0EE- 4E 45 D7                .9B NEW
A0F1- 54 41 42 A8             .9C TAB(
A0F5- 54 CF                   .9D TO
A0F7- 46 CE                   .9E FN
A0F9- 53 50 43 A8             .9F SPC(
A0FD- 54 48 45 CE             .A0 THEN
A101- 4E 4F D4                .A1 NOT
A104- 53 54 45 D0             .A2 STEP
A108- AB                      .A3 +
A109- AD                      .A4 -
A10A- AA                      .A5 *
A10B- AF                      .A6 /
A10C- DE                      .A7 ^
A10D- 41 4E C4                .A8 AND
A110- 4F D2                   .A9 OR
A112- BE                      .AA >
A113- BD                      .AB =
A114- BC                      .AC <
A115- 53 47 CE                .AD SGN
A118- 49 4E D4                .AE INT
A11B- 41 42 D3                .AF ABS
```

```
A11E- 55 53 D2              .B0 USR
A121- 46 52 C5              .B1 FRE
A124- 50 4F D3              .B2 POS
A127- 53 51 D2              .B3 SQR
A12A- 52 4E C4              .B4 RND
A12D- 4C 4F C7              .B5 LOG
A130- 45 58 D0              .B6 EXP
A133- 43 4F D3              .B7 COS
A136- 53 49 CE              .B8 SIN
A139- 54 41 CE              .B9 TAN
A13C- 41 54 CE              .BA ATN
A13F- 50 45 45 CB           .BB PEEK
A143- 4C 45 CE              .BC LEN
A146- 53 54 52 A4           .BD STR$
A14A- 56 41 CC              .BE VAL
A14D- 41 53 C3              .BF ASC
A150- 43 48 52 A4           .C0 CHR$
A154- 4C 45 46 54 A4        .C1 LEFT$
A159- 52 49 47 48 54 A4     .C2 RIGHT$
A15F- 4D 49 44 A4           .C3 MID$
A163- 00                    .BYTE 00, END OF TABLE

TABLE 5 ERROR MESSAGES

A164- 4E C6                 .NF
A166- 53 CE                 .SN
A168- 52 C7                 .RD
A16A- 4F C4                 .OD
A16C- 46 C3                 .FC
A16E- 4F D6                 .OV
A170- 4F CD                 .OM
A172- 55 D3                 .US
A174- 42 D3                 .BS
A176- 44 C4                 .DD
A178- 2F B0                 ./0
A17A- 49 C4                 .ID
A17C- 54 CD                 .TM
A17E- 4C D3                 .LS
A180- 53 D4                 .ST
A182- 43 CE                 .CN
A184- 55 C6                 .UF

TABLE 6 OTHER MESSAGES

A186- 20 45 52 52 4F 52 00  .ERROR
A18D- 20 49 4E 20 00        .IN
A192- 0D 0A 4F 4B 0D 0A 00  .OK
A199- 0D 0A 42 52 45 41 4B 00 .BREAK
```

HET STACK-GEBRUIK VAN DE BASIC

BASIC GEBRUIKT (BEHALVE HET DIREKTE GEBRUIK VAN DE INTERPRETER DE STACK VOOR DE <FOR NEXT> EN <GOSUB> KOMMANDO'S.
ZODRA ER EEN <FOR> OF <GOSUB> KOMMANDO WORDT GEGEVEN WORDT ER EEN VAST PAKKETJE MET GEGEVENS OP DE STACK GEZET, OM DAAR WEER VANAF TE WORDEN GEHAALD BIJ DE OVEREENKOMSTIGE <RETURN> OF <NEXT>.
DE OPZET VAN ZO'N PAKKETJE STAAT HIERONDER.
DE BYTES STAAN BESCHREVEN ZOALS ZE VAN DE STACK KOMEN.

FOR BLOCK

```
byte 1 - FOR token ($81)
byte 2 - pointer to FOR-variable -low-
byte 3 -                         -high-
byte 4 - STEP (FLP) -byte 1-
byte 5 -            -byte 2-
byte 6 -            -byte 3-
byte 7 -            -byte 4-
byte 8 - STEP -sign-
byte 9 - TO   (FLP) -byte 1-
byte 10-            -byte 2-
byte 11-            -byte 3-
byte 12-            -byte 4-
byte 13- statementnumber begin of loop -low-
byte 14-                               -high-
byte 15- characterpointer  begin of loop -high-
byte 16-                                 -low-
```

GOSUB BLOCK

```
byte 1 - GOSUB token ($8C)
byte 2 - GOSUB statementnumber -low
byte 3 -                       -high-
byte 4 - characterpointer saved -low-
byte 5 -                        -high-
```

DE OPBOUW VAN EEN BASIC-REGEL

BYTE 1 -ADRES V/D VOLGENDE REGEL (LOW B.)(LAATSTE REGEL=0)
BYTE 2 -ADRES V/D VOLGENDE REGEL (HIGH B.)(LAATSTE REGEL=0)
BYTE 3 -LAGE BYTE V/H REGELNUMMER
BYTE 4 -HOGE BYTE V/H REGELNUMMER
BYTE 5 EN VERDER DE OMGEZETTE TEKST VAN DE REGEL MET ALS LAATSTE BYTE EEN $00

RØUTINE 1 GETFØR

DEZE SUBRØUTINE ZØEKT NAAR EEN FØR-BLØK ØP DE STACK.

INPUT : ADRES VAN HET GEWENSTE FØR-BLØK IN $97-98
ALS $98=$00 NEEM DAN HET HØØGSTE FØR-BLØK
ØUTPUT: MET Z=1(STATUSREGISTER) IS HET BLØK GEVØNDEN
MET Z=0 IS HET BLØK NIET GEVØNDEN
WANNEER HET HØØGSTE BLØK WAS GEVRAAGD STAAT HET ADRES
IN $97-98

```
A1A1- BA       TSX           ;HAAL DE STACKPØINTER ØP
A1A2- E8       INX           ;SPRING ØVER DE TERUGKEER-ADRESSEN
A1A3- E8       INX           ;VAN DE TWEE SUBRØUTINES
A1A4- E8       INX           ;
A1A5- E8       INX           ;TØT BIJ HET HØØGSTE FØRNEXT BLØK
A1A6- BD 01 01 LDA $0101,X   ;HAAL HET TEKEN ØP
A1A9- C9 81    CMP #$81      ;WAS DAT EEN FØR-BLØK TEKEN ($81) ?
A1AB- D0 21    BNE $A1CE     ;ZØNEE, WEGSPRINGEN MET Z=0 NAAR RTS
A1AD- A5 98    LDA $98       ;ZØJA, HØØGSTE BLØK GEVENST ?
A1AF- D0 0A    BNE $A1BB     ;ZØNEE, (NØG) NIET BEWAREN, WEGSPRINGEN
A1B1- BD 02 01 LDA $0102,X   ;ZØJA, NEEM LØW-BYTE HØØGSTE BLØK
A1B4- 85 97    STA $97       ;EN ØPBERGEN
A1B6- BD 03 01 LDA $0103,X   ;NET ALS HET HIGH-BYTE
A1B9- 85 98    STA $98       ;
A1BB- DD 03 01 CMP $0103,X   ;IS DIT HET GEVENSTE BLØK ?
A1BE- D0 07    BNE $A1C7     ;ZØNEE, VERDERSPRINGEN (VØLGEND BLØK)
A1C0- A5 97    LDA $97       ;ANDERS ØØK NØG HET HØGE BYTE CHECKEN
A1C2- DD 02 01 CMP $0102,X   ;
A1C5- F0 07    BEQ $A1CE     ;GEVØNDEN, GA NAAR RTS MET Z=1
A1C7- 8A       TXA           ;NIET GEVØNDEN,
A1C8- 18       CLC           ;
A1C9- 69 10    ADC #$10      ;SPRING ØVER DIT BLØK HEEN
A1CB- AA       TAX           ;EN NEEM HET VØLGENDE BLØK ZØLANG
A1CC- D0 D8    BNE $A1A6     ;NØG NIET ØP HET EINDE V/D STACK
A1CE- 60       RTS           ;ANDERS TERUGKEREN..
```

RØUTINE 2 BLØCKMØVE

ER ZIJN TWEE INGANGEN IN DEZE GEHEUGENVERPLAATS-RØUTINE.
HET EERSTE DEEL ($A1CF-A1D5) KØNTRØLEERT ØF ER VØLDØENDE RUIMTE
IS, BREEKT AF MET EEN FØUTMELDING ALS DAT NIET LUKT.
HET TWEEDE DEEL (VANAF $A1D6) IS DE EIGENLIJKE RØUTINE.
DE HØØGSTE BYTES WØRDEN HET EERST VERPLAATST

INPUT : BEGINADRES V/H TE VERPLAATSEN BLØK ØP $AA-AB
EINDADRES V/H TE VERPLAATSEN BLØK ØP $A6-A7
EINDADRES V/H VERPLAATSTE BLØK ØP $A4-A5
ØUTPUT: BEGINADRES V/H VERPLAATSTE BLØK ØP $A7-A8
X=0;Y=0;$A6-A7 IS VERANDERD IN $AA-AB

```
A1CF- 20 1F A2 JSR $A21F  ;KØNTRØLEER ØF ER GENØEG VRIJ GEHEUGEN IS
A1D2- 85 7F    STA $7F    ;JA, PAS BEGINADRES VRIJ GEHEUGEN AAN
A1D4- 84 80    STY $80    ;EN ØØK HET HØØGSTE BYTE

REF 2B

A1D6- 38       SEC        ;BEREKEN DE LENGTE V/H TE VERPLAATSEN BLØ
A1D7- A5 A6    LDA $A6    ;
A1D9- E5 AA    SBC $AA    ;
A1DB- 85 71    STA $71    ;BEWAAR HET LAAGSTE BYTE
A1DD- A8       TAY        ;Y REG IS LØW-BYTE V/D BLØKLENGTE
A1DE- A5 A7    LDA $A7    ;
A1E0- E5 AB    SBC $AB    ;
A1E2- AA       TAX        ;X REG IS HIGH-BYTE V/D BLØKLENGTE
A1E3- E8       INX        ;PAS X AAN
A1E4- 98       TYA        ;ALS Y 'N VEELVØUD IS VAN 256,
A1E5- F0 23    BEQ $A20A  ;SPRING DAN VERDER
A1E7- A5 A6    LDA $A6    ;BEREKEN ANDERS HET BEGIN V/D REST
A1E9- 38       SEC        ;
A1EA- E5 71    SBC $71    ;
A1EC- 85 A6    STA $A6    ;SLA DAT ØP
A1EE- B0 03    BCS $A1F3  ;
A1F0- C6 A7    DEC $A7    ;ØP $A6-A7 STAAT HET BEGINBESTEMMINGSADRE
A1F2- 38       SEC        ;
A1F3- A5 A4    LDA $A4    ;BEREKEN HET BEGIN V/D BESTEMMINGSADRES
A1F5- E5 71    SBC $71    ;VAN DIT GEDEELTE
A1F7- 85 A4    STA $A4    ;
A1F9- B0 08    BCS $A203  ;
A1FB- C6 A5    DEC $A5    ;
A1FD- 90 04    BCC $A203  ;ZET DAT IN $A4-A5
A1FF- B1 A6    LDA ($A6),Y;HAAL 'N BYTE UIT HET HERKØMSTBLØK
A201- 91 A4    STA ($A4),Y;VERPLAATS DAT
A203- 88       DEY        ;PAS DE TELLET AAN
A204- D0 F9    BNE $A1FF  ;GA ZØ DØØR TØT HET EINDE
A206- B1 A6    LDA ($A6),Y;VERPLAATS ØØK NØG HET LAATSTE BYTE
A208- 91 A4    STA ($A4),Y;DAT ANDERS WØRDT VERGETEN
A20A- C6 A7    DEC $A7    ;PAS ØØK NØG DE VERWIJZINGEN AAN
A20C- C6 A5    DEC $A5    ;VAN HET HERKØMST EN NIEUWE BLØK
A20E- CA       DEX        ;ZIJN ALLE PAGINA'S (VAN 256 BYTES) KLAAR ?
A20F- D0 F2    BNE $A203  ;ZØNEE, NØGMAALS
A211- 60       RTS        ;ZØJA, KLAAR
```

RØUTINE 3 CHECKSTACK

DEZE SUBRØUTINE TEST ØF VØLDØENDE BYTES IN DE STACK VRIJ ZIJN.
MINIMUM IS 2*HET AANTAL IN DE ACCUMULATØR + 51 BYTES, VØØR
NMI VERWERKING. DE TEST IS NIET GEHEEL GØED, ZØDAT DE
BASIC BIJ NMI VERWEKING KAN BLIJVEN HANGEN.

```
A212- 0A       ASL          ;BEREKEN TWEE MAAL DE ACCUMULATOR
A213- 69 33    ADC #$33     ;PLUS 51 BYTES EXTRA VOOR NMI
A215- B0 35    BCS $A24C    ;BIJ STACK-OVERFLOW ..OM ERROR
A217- 85 71    STA $71      ;
A219- BA       TSX          ;
A21A- E4 71    CPX $71      ;VOLDOENDE RUIMTE ??
A21C- 90 2E    BCC $A24C    ;NEE, DAN OM ERROR GEVEN
A21E- 60       RTS          ;JA, KLAAR
```

ROUTINE 4 CHECKMEMORY

DEZE SUBROUTINE KONTROLEERT OF VOLDOENDE GEHEUGEN AANWEZIG IS.
BIJ DE KONTROLE WORDEN A+Y VERGELEKEN MET HET STARTADRES VAN
DE STRINGRUIMTE. BIJ ONVOLDOENDE GEHEUGEN WORDT DE <GARBAGE-
COLLECT> ROUTINE AANGEROEPEN, EN DAARNA OPNIEUW GETEST.
ALS ER DAN NOG NIET GENOEG GEHEUGEN IS WORDT DAT GEMELD MET
EEN (FATALE) FOUTMELDING.

INPUT : EINDE VAN DE ARRAY-RUIMTE IN A+Y
OUTPUT: NIEUWE EINDE IN A+Y, BEGINADRES VAN DE VRIJE
         GEHEUGENRUIMTE AANGEPAST IN $81-82

```
A21F- C4 82     CPY $82     ;VOLDOENDE RUIMTE BIJ HET HOGE BYTE ?
A221- 90 28     BCC $A24B   ;JA DAN KLAAR
A223- D0 04     BNE $A229   ;GA DE GARBAGE VERZAMELEN
A225- C5 81     CMP $81     ;KONTROLEER OOK HET LAGE BYTE
A227- 90 22     BCC $A24B   ;DAT IS GOED DUS KLAAR
A229- 48        PHA         ;NIET GOED, BEWAAR DE ACCUMULATOR
A22A- A2 08     LDX #$08    ;
A22C- 98        TYA         ;EN HET Y-REGISTER
A22D- 48        PHA         ;BEWAAR $A4-AC
A22E- B5 A3     LDA $A3,X   ;
A230- CA        DEX         ;
A231- 10 FA     BPL $A22D   ;
A233- 20 47 B1  JSR $B147   ;GA DE STRING RUIMTE OPRUIMEN
A236- A2 F8     LDX #$F8    ;EN ZET $A4-AC WEER TERUG.
A238- 68        PLA         ;
A239- 95 AC     STA $AC,X   ;
A23B- E8        INX         ;
A23C- 30 FA     BMI $A238   ;
A23E- 68        PLA         ;
A23F- A8        TAY         ;HAAL Y TERUG
A240- 68        PLA         ;EN DE ACCU
A241- C4 82     CPY $82     ;IS ER NU VOLDOENDE RUIMTE ?
A243- 90 06     BCC $A24B   ;JA, TERUGSPRINGEN NAAR DE OPROEPER
A245- D0 05     BNE $A24C   ;NEE, DAN EEN OM ERROR GEVEN
A247- C5 81     CMP $81     ;ONDUIDELIJK, KONTROLEER HET LAGE BYTE
A249- B0 01     BCS $A24C   ;JAMMER, TOCH OM ERROR
A24B- 60        RTS         ;KLAAR
```

ROUTINE 5 FATAL ERRORS

NAAR DEZE ROUTINE WORDT GESPRONGEN BIJ EEN FOUT WAARBIJ BASIC MOET AFBREKEN. DE FOUT WORDT GEMELD & DE BASIC SPRINGT NAAR DE COMMAND-MODE.

```
A24C- A2 0C     LDX #$0C  LAAT X NAAR DE 'OM' WIJZEN
```

FOUT AFDRUK

DEZE ROUTINE DRUKT EEN FOUTMELDING AF. AAN DE HAND VAN DE INHOUD VAN HET X-REG. WORDT IN EEN TABEL DE FOUT OPGEZOCHT TWEE LETTERS DER FOUTMELDING WORDEN AFGEDRUKT OM DE SOORT FOUT AAN TE GEVEN (WEL MET BIT7=1 ZODAT ER EEN GRAFISCH SYMBOOL VERSCHIJNT I. P. V. EEN LETTER.

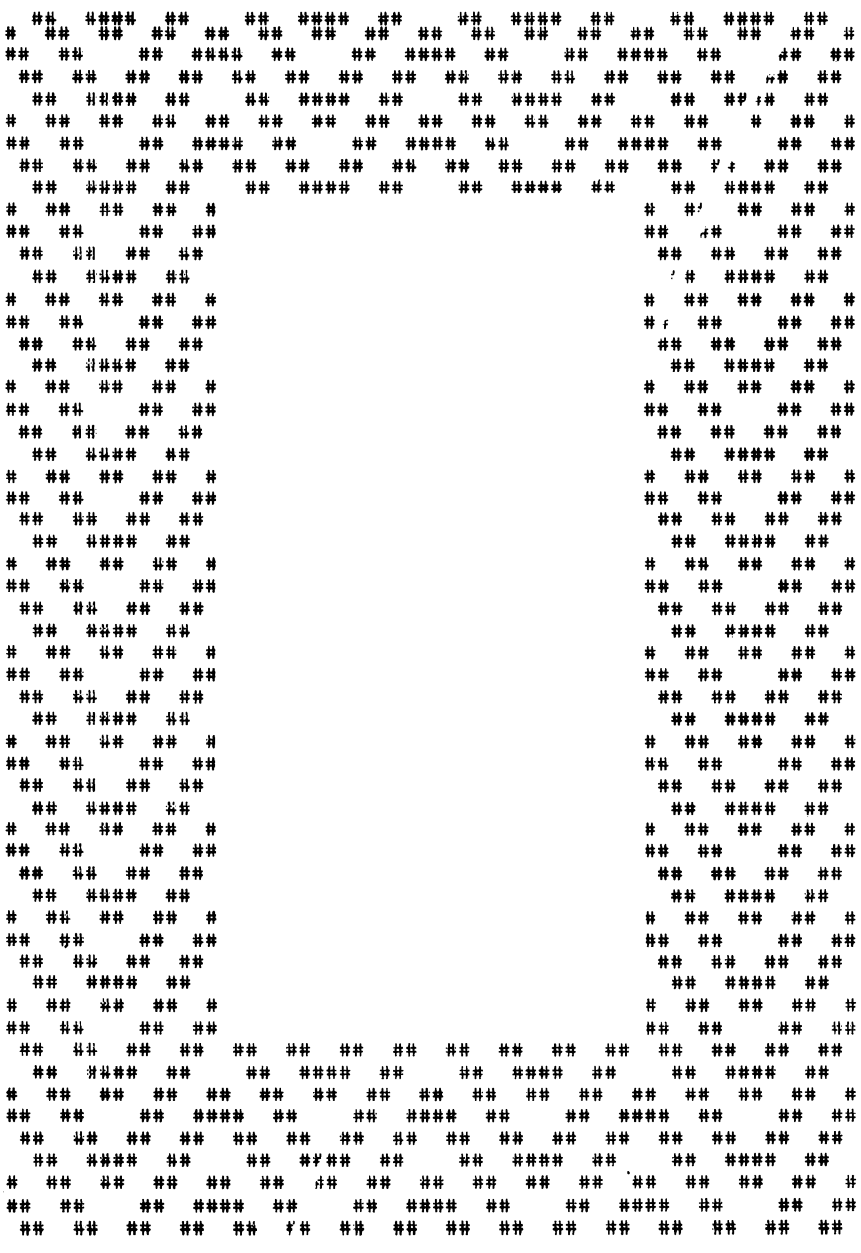
```
A24E- 46 64     LSR $64       ZORG DAT ER GEPRINT KAN WORDEN
A250- 20 6C A8  JSR $A86C     PRINT EEN CR/LF
A253- 20 E3 A8  JSR $A8E3     PRINT EEN VRAAGTEKEN '?'
A256- BD 64 A1  LDA $A164&X   HAAL DE EERSTE LETTER OP
A259- 20 E5 A8  JSR $A8E5     DRUK HET AF
A25C- BD 65 A1  LDA $A165&X   DAN DE TWEEDE LETTER
A25F- 20 E5 A8  JSR $A8E5     OOK AFDRUKKEN
A262- 20 91 A4  JSR $A491     ZET AL DE BASIC VERWIJZINGEN TERUG
A265- A9 8S     LDA #$86      LAAT A+Y WIJZEN NAAR DE PLAATS WAAR
A267- A0 A1     LDY #$A1      DE 'ERROR' STRING TE VINDEN IS
A269- 20 C3 A8  JSR $A8C3     DRUK DIE AF
A26C- A4 88     LDY #$88      NEEM HET HOGE BYTE V/H HUIDIGE
A26E- C8        INY           REGELNUMMER& VERHOOG MET EEN
A26F- F0 03     BEQ $A274     ALS 0& DAN IN DIRECT MODE& SPRINGEN
.                             SPRINGEN
A271- 20 53 B9  JSR $B953     DRUK 'IN' AF MET HET REGELNUMMER
```

ROUTINE 6 WARME BASIC START

NA HET OPSTARTEN IS DIT HET ALGEMENE BEGINPUNT

```
A274- 46 64     LSR $64       SORG DAT ER GEPRINT KAN WORDEN
A276- A9 92     LDA #$92      LAAT A+Y WIJZEN NAAR ADRES WAARMEE
A278- A0 A1     LDY #$A1      'OK' WORDT AFGEDRUKT
A27A- 20 03 00  JSR $0003     DOE DAT DAN OOK
A27D- 20 57 A3  JSR $A357     VUL DE INPUT-BUFFER
A280- 86 C3     STX $C3       ZET HET ADRES VAN DE BUFFER IN DE
A282- 84 C4     STY $C4       VERBINDINGSROUTINE (LOW-HIGH ADRES)
A284- 20 BC 00  JSR %00BC     HAAL EERSTE LETTER BUITEN 'N SPATIE
A287- F0 F4     BEQ $A27D     ZOLANG DE INPUTBUFFER LEEG IS& TERUG
A289- A2 FF     LDX #$FF      ZET DE BASIC IN DE DIRECT-MODE
A28B- 86 88     STX $88       (DAN REAGEERT IE ZONDER REGELNUMMERS
A28D- 90 06     BCC $A295     ALS EERSTE LETTER=CIJFER DAN NAAR
.                             EDITROUTINE
A28F- 20 A6 A3  JSR $A3A6     ANDERS 'N COMMANDO DUS OMZETTEN
.                             (ROUTINE 11)
A292- F6 A5     JMP $A5F6     GA HET COMMANDE UITVOEREN
```

```
10              ; BAS4.3, TOM MEIJERING, EMMEN 830210
20              ;
30              ; NIEUW DEEL VOOR BASIC 4 ROM, WERKT IN KOMBI-
40              ; NATIE MET NIEUWE MONITOREPROM.
50              ;
60              ;
70              ; GEWIJZIGDE KOUDE START EN AANVULLENDE
80              ; ROUTINES VOOR HEXADECIMAAL BEWERKIN-
90              ; GEN VOOR DE TOOLKIT, AAN TE ROEPEN VAN-
100             ; UIT BASIC.
110             ;
120             ;
130 E014=               TOOLTB=$E014 ; START TOOLKITTABEL,
140                                  ; (EVENTUEEL WIJZIGEN)
150             ;
160             ;
170             ;
180             ; VAN BCEE T/M BD09 WORDT BIJ KOUDE START GE-
190             ; COPIEERD NAAR 00BC T/M 00D7 (ZIE IN3).
200             ; EEN PAAR LOKATIES ZIJN VERANDERD.
210             ;
220 BCED                *=$BCED
230 BCED 00       BCED  BRK
240 BCEE E6C3     BCEE  INC $C3
250 BCF0 D002           BNE BCF4
260 BCF2 E6C4           INC $C4
270 BCF4 AD       BCF4  .BYTE $AD,$12,$00
270 BCF5 12
270 BCF6 00
280 BCF7 C93A           CMP #':
290 BCF9 B00A           BCS BD05
300 BCFB C920           CMP #$20
310 BCFD F0EF           BEQ BCEE
320 BCFF 38             SEC
330 BD00 E930           SBC #$30
340 BD02 38             SEC
350 BD03 E9D0           SBC #$D0
360 BD05 60       BD05  RTS
370 BD06 80             .BYTE $80,$4F,$C7,$52
370 BD07 4F
370 BD08 C7
370 BD09 52
380             ;
390             ;
400             ; VAN BD09 T/M BD11 LOZE RUIMTE OPGEVULD
410             ; MET EEN SUBROUTINE, NAMELIJK ROUTINE OM
420             ; DE GET NEXT CHARACTER ROUTINE OP 00BC
430             ; TE RESETTEN NAAR HET BEGIN VAN DE INPUT-
440             ; BUFFER!
450             ;
460 BD0A A900     RESBC LDA #$00
470 BD0C 85C4           STA $C4
480 BD0E 4C2DA4         JMP $A42D ; #$12 NAAR $C3
```



FEBRUARI 1983

RØUTINE 7 EDIT

MET DEZE VERZAMELING RØUTINES WØRDEN GENUMMERDE REGELS:
INGEVØEGD, GEWIST, VERVANGEN ØF TØEGEVØEGD DØØR BASIC.

```
A295- 20 7F A7 JSR $A77F  ;ZET DECIMAAL REGELNR ØM IN HEX-GETAL
A298- 20 A6 A3 JSR $A3A6  ;ZET DE REGEL IN DE BUFFER ØM IN TØKENS
A29B- 84 5D    STY $5D    ;BEWAAR DE LENGTE VAN DE REGEL.
A29D- 20 32 A4 JSR $A432  ;BESTAAT HET REGELNUMMER AL ?
A2A0- 90 44    BCC $A2E6  ;ZØNEE, VØEG HEM DAN IN
```

RØUTINE 7A DELETE LINE

DE ØUDE REGEL WØRDT GEWIST DØØR EEN MET HETZELFDE NUMMER
IN DE INPUTBUFFER.

```
A2A2- A0 01    LDY #$01   ;ZET PØINTER KLAAR
A2A4- B1 AA    LDA ($AA),Y;NEEM HØGE BYTE (ZIE ØØK RØUTINE 2)
A2A6- 85 72    STA $72    ;BEWAREN IN TIJD.ØPSLAGPLAATS NR.1
A2A8- A5 7B    LDA $7B    ;HAAL HET EINDADRES V/H PRØGRAMMA ØP
A2AA- 85 71    STA $71    ;BEWAAR (LAAG BYTE)
A2AC- A5 AB    LDA $AB    ;HAAL VERWIJZING NAAR DE TE WISSEN REGEL
A2AE- 85 74    STA $74    ;BEWAAR HET HØGE BYTE
A2B0- A5 AA    LDA $AA    ;EN HET LAGE BYTE
A2B2- 88       DEY        ;TREK VAN ELKAAR AF HET ADRES VAN HET
A2B3- F1 AA    SBC ($AA),Y;HUIDIGE REGELNUMMER EN DE VØLGENDE REGEL
A2B5- 18       CLC        ;ZØDAT SAMEN MET HET EINDADRES V/H PRØGR.
A2B6- 65 7B    ADC $7B    ;HET NIEUWE EINDADRES RESULTEERT.(LØW BYTE)
A2B8- 85 7B    STA $7B    ;BEWAREN ØP TIJD.ØPSLAGPLAATS 2
A2BA- 85 73    STA $73
A2BC- A5 7C    LDA $7C    ;PAS ØØK HET HØGE BYTE VAN HET EINDADRES
A2BE- 69 FF    ADC #$FF   ;AAN, EN
A2C0- 85 7C    STA $7C    ;BEWAAR DAT.
A2C2- E5 AB    SBC $AB    ;MINUS DE VERWIJZING NAAR DE TE WISSEN REGEL
A2C4- AA       TAX        ;STAAT HET AANTAL TE VERPLAATSEN BYTES IN X
A2C5- 38       SEC        ;BEREKEN ØØK NIEUWE LØW BYTE
A2C6- A5 AA    LDA $AA    ;LØW ADRES VAN HET PRØGRAMMAEINDE MINUS
A2C8- E5 7B    SBC $7B    ;'T ADRES V/D TE WISSEN REGEL, GEEFT
A2CA- A8       TAY        ;HET AANTAL TE VERPLAATSEN BYTES IN Y
A2CB- B0 03    BCS $A2D0  ;ALS DE REST NUL IS VERDERSPRINGEN
A2CD- E8       INX        ;ANDERS X-REG. (HIGH BYTE) AANPASSEN
A2CE- C6 74    DEC $74    ;PAS NU DE VERWIJZINGEN AAN
A2D0- 18       CLC        ;EERST HET LAGE BYTE
A2D1- 65 71    ADC $71    ;ALS ER GEEN REST IS VERDERSPRINGEN
A2D3- 90 03    BCC $A2D8  ;ANDERS HET HØGE BYTE AANPASSEN
A2D5- C6 72    DEC $72    ;
A2D7- 18       CLC        ;HAAL HET TE VERPLAATSEN BYTE ØP
A2D8- B1 71    LDA ($71),Y;ZET HET ØP DE NIEUWE PLAATS
A2DA- 91 73    STA ($73),Y;IS DEZE PAGINA (256 BYTES) KLAAR ?
A2DC- C8       INY        ;ZØNIET TERUGSPRINGEN EN NØG EENS
A2DD- D0 F9    BNE $A2D8  ;ZØJA BLØKKENTELLER HERKØMST VERHØGEN
A2DF- E6 72    INC $72    ;EN ØØK DIE VAN DE BESTEMMING
A2E1- E6 74    INC $74    ;ZIJN ALLE PAGINA'S AFGEWERKT ?
A2E3- CA       DEX        ;ZØNIET TERUG
A2E4- D0 F2    BNE $A2D8
```

```
490             ;
500             ;
510             ; ENTREE KOUDE START MET ORIGINELE VRAAG-
520             ; STELLING TEN BEHOEVE VAN PICODOS:
530             ;
540 BD11 A2FF   BD11   LDX #$FF
550 BD13 8688          STX $88
560 BD15 9A            TXS
570 BD16 20D2BD        JSR INIT  ; INITIALISEER ZEROPAGE
580 BD19 48            PHA
590 BD1A F049          BEQ OUD   ; GA ALTIJD NAAR OUDE VRAAG-
600                               ; STELLING
610 BD1C FF            .BYTE $FF
620             ;
630             ;
640             ; LOZE RUIMTE TOT BD22 OPGEVULD MET SUBROU-
650             ; TINE VOOR HET UITPRINTEN VAN EEN MESSAGE:
660             ;
670 BD1D A0BE   MSOUT  LDY #MESS1/256 ; ALLE MESSAGES MOETEN
680                                   ; IN DEZELFDE PAGE STAAN
690 BD1F 4CC3A8        JMP $A8C3
700             ;
710             ;
720             ; ENTREE KOUDE START VANUIT MONITOR NA BREAK:
730             ;
740 BD22 20D2BD BD22   JSR INIT
750 BD25 48            PHA
760 BE00=              H1=MESS1/256*256
770 BD26 A979   NIEUW  LDA #MESS1-H1
780 BD28 20C9BD        JSR VRAAG ; FIRST?
790 BD2B F010          BEQ TEST  ; DIREKT RETURN:
800                               ; BEGIN VAN BASIC OP 0300 LA-
810                               ; TEN STAAN EN TESTEN WAAR
820                               ; EINDE RAM IS VOOR EINDE BASIC.
830 BD2D 2002BE        JSR HEXIN
840 BD30 D0F4          BNE NIEUW ; GEEN GELDIG HEXADRES OPGEGEVEN.
850 BD32 8579          STA $79
860 BD34 847A          STY $7A
870 BE00=              H2=MESS2/256*256
880 BD36 A983   BASEND LDA #MESS2-H2
890 BD38 20C9BD        JSR VRAAG  ; END?
900 BD3B D021          BNE HEXEND ; NIET DIREKT RETURN, HAAL
910                               ; OPGEGEVEN ADRES OP.
920             ;
930             ;
940             ; TEST WAAR EINDE RAM IS VOOR EINDE BASIC:
950             ;
960 BD3D A579   TEST   LDA $79
970 BD3F A47A          LDY $7A
980 BD41 8511          STA $11
990 BD43 8412          STY $12
1000 BD45 A000         LDY #$00
1010 BD47 E611  TEST2  INC $11
1020 BD49 D002         BNE TEST3
1030 BD4B E612         INC $12
1040 BD4D B111  TEST3  LDA ($11),Y
1050 BD4F 49FF         EOR #%11111111
```

```
1060 BD51 9111          STA ($11),Y
1070 BD53 D111          CMP ($11),Y
1080 BD55 D017          BNE GET     ; EINDE RAM GEVONDEN.
1090 BD57 49FF          EOR #%11111111
1100 BD59 9111          STA ($11),Y ; HERSTEL INHOUD.
1110 BD5B 4C47BD        JMP TEST2
1120              ;
1130              ;
1140              ; HAAL OPGEGEVEN ADRES VOOR EINDE BASIC OP:
1150              ;
1160 BD5E 2002BE  HEXEND0 JSR HEXIN
1170 BD61 D0D3          BNE BASEND  ; ONGELDIG, OPNIEUW.
1180 BD63 F00D          BEQ STORE   ; GELDIG, VUL POINTERS IN.
1190              ;
1200              ;
1210              ; OUDE VRAAGSTELLING:
1220              ;
1230 BD65 2046A9  OUD     JSR $A946   ; PRINT ? EN VUL BUFFER.
1240 BD68 20BC00          JSR $00BC   ; HAAL EERSTE LETTER OP.
1250 BD6B 207FA7          JSR $A77F   ; ZET DECIMAAL OM NAAR HEX.
1260 BD6E A511    GET     LDA $11
1270 BD70 A412            LDY $12
1280              ;
1290              ;
1300              ; VUL POINTERS VOOR EINDE BASIC IN:
1310              ;
1320 BD72 8585    STORE   STA $85
1330 BD74 8486            STY $86
1340 BD76 8581            STA $81
1350 BD78 8482            STY $82
1360 BD7A A000            LDY #$00
1370 BD7C 98              TYA
1380 BD7D 9179            STA ($79),Y
1390 BD7F E679            INC $79
1400 BD81 D002            BNE STOR2
1410 BD83 E67A            INC $7A
1420 BD85 A579    STOR2   LDA $79
1430 BD87 A47A            LDY $7A
1440 BD89 201FA2          JSR $A21F   ;  OM?
1450 BD8C 2063A4          JSR $A463   ;  NEW
1460              ;
1470              ;
1480              ; ENTREE VOOR MELDING
1490              ; FREE $XXXX-$YYYY VANUIT TOOLKIT.
1500              ;
1510 BE00=                H3=MESS3/256#256
1520 BD8F A98B    FREE    LDA #MESS3-H3
1530 BD91 201DBD          JSR MSOUT
1540 BD94 A27C            LDX #$7C
1550 BD96 202BBE          JSR HEXOU2  ; PRINT INHOUD 7B/7C
1560 BD99 A92D            LDA #'-
1570 BD9B 20EEFF          JSR $FFEE
1580 BD9E A286            LDX #$86
1590 BDA0 202BBE          JSR HEXOU2  ; PRINT INHOUD 85/86
```

```
1600              ;
1610              ;
1620              ; ENTREE VOOR WARME START MET INSCHAKELING
1630              ; VAN DE TOOLKIT:
1640              ;
1650 BDA3 A94C    WARM1  LDA #$4C
1660 BDA5 85C5           STA $C5
1670 BE00=               I=INPUT/256*256
1680 BDA7 A94C           LDA #INPUT-I
1690 BDA9 85C6           STA $C6
1700 BDAB A9BE           LDA #INPUT/256
1710 BDAD 85C7           STA $C7
1720              ;
1730              ;
1740              ; ENTREE VOOR WARME START ZONDER TOOLKIT:
1750              ;
1760 BDAF A2FE    WARM2  LDX #$FE
1770 BDB1 8E2802         STX $0228
1780 BDB4 9A             TXS
1790 BDB5 4C74A2         JMP $A274
1800              ;
1810              ;
1820              ; GEDEELTE DAT BIJ KOUDE START NAAR 0000 T/M
1830              ; 0010 GECOPIEERD WORDT:
1840              ;
1850 BDB8 4CA3BD  TABEL  JMP WARM1   ; WARME-START VECTOR MET IN-
1860                                 ; SCHAKELEN TOOLKIT.
1870 BDBB 4CAEFB         JMP $FBAE   ; PROMPTMELDING READY IPV. OK.
1880 BDBE 05AE           ORA $AE
1890 BDC0 C1AF           CMP ($AF,X)
1900 BDC2 4C0EBF         JMP USR     ; A=USR(8) GEEFT WACHT OP
1910                                 ; TOETS EN GEEF WAARDE AAN A.
1920 BDC5 00             BRK
1930 BDC6 00             BRK
1940 BDC7 FF             .BYTE $FF   ; TERMINAL WIDTH.
1950 BDC8 FC             .BYTE $FC   ; TERMINAL WIDTH KOLOMMEN.
1960              ;
1970              ;
1980              ; SUBROUTINE OM MESSAGE TE PRINTEN EN INPUT
1990              ; TE VRAGEN:
2000              ;
2010 BDC9 201DBD  VRAAG  JSR MSOUT
2020 BDCC 2057A3         JSR $A357   ; VUL INPUTBUFFER.
2030 BDCF A513           LDA $13     ; HAAL EERSTE LETTER OP.
2040 BDD1 60             RTS
2050              ;
2060              ;
2070              ; SUBROUTINE OM ZEROPAGE TE INITIALISEREN:
2080              ;
2090 BDD2 A210    INIT   LDX #$10
2100 BDD4 BDB8BD  IN2    LDA TABEL,X
2110 BDD7 9500           STA $00,X   ; COPIEER 0000 T/M 0010.
2120 BDD9 CA             DEX
2130 BDDA 10F8           BPL IN2
2140 BDDC A21C           LDX #$1C
2150 BDDE BDEDBC  IN3    LDA BCED,X
```

```
2160 BDE1 95BB          STA $BB,X   ; COPIEER 00BC T/M 00D7.
2170 BDE3 CA            DEX
2180 BDE4 D0F8          BNE IN3
2190 BDE6 A903          LDA #$03
2200 BDE8 857A          STA $7A     ; DEFAULT BEGIN BASIC 0300.
2210 BDEA 85A0          STA $A0
2220 BDEC A94C          LDA #$4C
2230 BDEE 85A1          STA $A1
2240 BDF0 A92C          LDA #$2C
2250 BDF2 8512          STA $12
2260 BDF4 A968          LDA #$68
2270 BDF6 8565          STA $65
2280 BDF8 8A            TXA
2290 BDF9 8564          STA $64
2300 BDFB 8567          STA $67
2310 BDFD 8579          STA $79
2320 BDFF 85B2          STA $B2
2330 BE01 60            RTS
2340              ;
2350              ;
2360              ; SUBROUTINE OM INHOUD INPUTBUFFER OM TE ZETTEN
2370              ; IN EEN HEXADRES, NA RESET 00BC NAAR BEGIN
2380              ; VAN DE INPUTBUFFER (UNIVERSEEL):
2390              ;
2400 BE02 200ABD  HEXIN  JSR RESBC
2410              ;
2420              ;
2430              ; ALS HEXIN, MAAR ZONDER RESET 00BC:
2440              ;
2450 BE05 A204    HEXIN1 LDX #4
2460 BE07 86E0           STX $E0
2470 BE09 A200           LDX #0
2480 BE0B 86FC           STX $FC
2490 BE0D 86FD           STX $FD
2500 BE0F 20BC00  HEXIN2 JSR $00BC   ; HAAL CIJFER UIT INPUTBUFFER.
2510 BE12 2093FE         JSR $FE93   ; ZET OM IN BINAIR.
2520 BE15 300A           BMI KLAAR   ; ONGELDIG.
2530 BE17 20DAFE         JSR $FEDA   ; ROL IN FC/FD
2540 BE1A C6E0           DEC $E0
2550 BE1C D0F1           BNE HEXIN2  ; NOG GEEN 4 CIJFERS.
2560 BE1E 20BC00         JSR $00BC   ; ZET INPUTROUTINE GOED.
2570 BE21 A1C3    KLAAR  LDA ($C3,X)
2580 BE23 AA             TAX         ; ZET LAATSTE KAR. IN X
2590                                 ; (DIT MOET 0 ZIJN, ANDERS
2600                                 ; IS EEN ONGELDIG ANTWOORD
2610                                 ; OPGEGEVEN).
2620 BE24 A5FC           LDA $FC     ; LO
2630 BE26 A4FD           LDY $FD     ; HI
2640 BE28 E000           CPX #$00    ; GELDIG ANTWOORD?
2650 BE2A 60      RET    RTS
```

```
2660              ;
2670              ;
2680              ; SUBROUTINE OM DE INHOUD VAN ADRES X EN X-1
2690              ; ALS ADRES MET $ TE PRINTEN (UNIVERSEEL)!
2700              ;
2710 BE2B A924    HEXOU2 LDA #'$
2720 BE2D 20E5A8         JSR $A8E5   ; OUTPUT ACC.
2730              ;
2740              ;
2750              ; ALS HEXOU2, ECHTER ZONDER $!
2760              ;
2770 BE30 2034BE  HEXOUT JSR HEX1
2780 BE33 CA             DEX
2790 BE34 B500    HEX1   LDA $00,X
2800              ;
2810              ;
2820              ; SUBROUTINE OM INHOUD VAN Acc HEX UIT TE PRINTEN
2830              ; (UNIVERSEEL)!
2840              ;
2850 BE36 48      HEX2   PHA
2860 BE37 4A             LSR A
2870 BE38 4A             LSR A
2880 BE39 4A             LSR A
2890 BE3A 4A             LSR A
2900 BE3B 203FBE         JSR OUT
2910 BE3E 68             PLA
2920 BE3F 290F    OUT    AND #$0F
2930 BE41 0930           ORA #$30
2940 BE43 C93A           CMP #$3A
2950 BE45 3002           BMI DISP
2960 BE47 6906           ADC #6
2970 BE49 4CE5A8  DISP   JMP $A8E5   ; OUTPUT ACC.
2980              ;
2990              ;
3000              ; CHECK # TEKEN VOOR TOOLKIT!
3010              ;
3020 BE4C C923    INPUT  CMP #'#
3030 BE4E F007           BEQ TOOL
3040 BE50 C93A           CMP #':
3050 BE52 B0D6           BCS RET
3060 BE54 4CC900         JMP $C9
3070              ;
3080              ;
3090              ; ENTREE TOOLKIT!
3100              ;
3110 BE57 20BC00  TOOL   JSR $BC
3120 BE5A C941           CMP #'A
3130 BE5C 9018           BCC ERROR
3140 BE5E C95B           CMP #$5B
3150 BE60 B014           BCS ERROR
3160 BE62 291F           AND #$1F
3170 BE64 0A             ASL A
3180 BE65 A8             TAY
3190 BE66 B912E0         LDA TOOLTB-2,Y
3200 BE69 85A2           STA $A2
3210 BE6B B913E0         LDA TOOLTB-1,Y
3220 BE6E 85A3           STA $A3
3230 BE70 20A100         JSR $A1
3240 BE73 4CBC00         JMP $BC
```

```
1250              ;
1260 BE76 4C62A2  ERROR  JMP $A262
1270              ;
1280              ;
1290              ; MESSAGES (MOETEN IN DEZELFDE PAGE STAAN):
1300              ;
1310 BE79 1A      MESS1  .BYTE 26,10,'FIRST $',0
1310 BE7A 0A      .
1310 BE7B 46
1310 BE7C 49
1310 BE7D 52
1310 BE7E 53
1310 BE7F 54
1310 BE80 20
1310 BE81 24
1310 BE82 00
1320 BE83 20      MESS2  .BYTE          END $',0
1320 BE84 20
1320 BE85 45
1320 BE86 4E
1320 BE87 44
1320 BE88 20
1320 BE89 24
1320 BE8A 00
1330 BE8B 1A      MESS3  .BYTE 26,10,'FREE ',0
1330 BE8C 0A
1330 BE8D 46
1330 BE8E 52
1330 BE8F 45
1330 BE90 45
1330 BE91 20
1330 BE92 00
1340              ;
1350              ;
1360              ; NIEUWE HEXASIGNMENT TEN BEHOEVE VAN TOOLKIT:
1370              ;
1380 BE00=              H=HEXAS/256*256
1390 BE93 68      HEXASG PLA
1400 BE94 68             PLA          ; VERWIJDER RETURNADRES.
1410 BE95 A9A4           LDA #HEXAS-H
1420 BE97 85C6           STA $C6      ; ZET 00BC OM NAAR HEXAS.
1430 BE99 A5C3           LDA $C3
1440 BE9B 85E0           STA $E0
1450 BE9D A5C4           LDA $C4
1460 BE9F 85E1           STA $E1      ; BEWAAR INSTELLING 00BC.
1470 BEA1 A9B0           LDA #$B0     ; TOKEN VOOR USR.
1480 BEA3 60             RTS
1490              ;
1500              ; NB. DE USR-FUNKTIE WORDT NU DOORLOPEN, TOTDAT
1510              ; DEZE WEER 00BC AANROEPT EN DUS NAAR HEXAS GE-
1520              ; SPRONGEN WORDT.
1530              ;
1540              ; HEXAS MOET IN DEZELFDE PAGE BEGINNEN ALS INPUT.
1550              ;
1560 BEA4 68      HEXAS  PLA
1570 BEA5 68             PLA          ; VERWIJDER RETURNADRES.
1580 BEA6 A94C           LDA #INPUT-I
1590 BEA8 85C6           STA $C6      ; HERSTEL 00BC.
```

```
3600 BEAA A5E1          LDA $E1
3610 BEAC 85C4          STA $C4
3620 BEAE A5E0          LDA $E0
3630 BEB0 85C3          STA $C3
3640 BEB2 2005BE        JSR HEXIN1  ; HAAL ADRES UIT INPUTBUFFER.
3650 BEB5 85AE          STA $AE     ; LO
3660 BEB7 84AD          STY $AD     ; HI
3670 BEB9 A290          LDX #$90
3680 BEBB 38            SEC
3690 BEBC 20E8B7        JSR $B7E8   ; ZET OM NAAR DECIMAAL EN
3700 BEBF 4CB0AA        JMP $AAB0   ; GA DOOR ALS NA USR.
3710                ;
3720                ;
3730                ; VERBETERDE LEES DECIMAAL ROUTINE, TEVENS
3740                ; AANGEPAST IVM. REVHEX!
3750                ;
3760 BEC2 20D0BE  E076  JSR IN
3770 BEC5 85E0          STA $E0
3780 BEC7 84E1          STY $E1
3790 BEC9 20BC00        JSR $00BC
3800 BECC C92C          CMP #',     ; 2e GETAL, GESCHEIDEN DOOR ,?
3810 BECE D005          BNE FOUT
3820 BED0 20BC00  IN    JSR $00BC
3830 BED3 D005          BNE GOED
3840 BED5 A900    FOUT  LDA #$00    ; VUL 0 IN VOOR CIJFER.
3850 BED7 A8            TAY
3860 BED8 F009          BEQ END     ; ALTIJD.
3870                ;
3880 BEDA 20C1AA  GOED  JSR $AAC1   ; NAAR FLOATING POINT.
3890 BEDD 2008B4        JSR $B408   ; NAAR 0011/0012.
3900 BEE0 A8            TAY         ; HI
3910 BEE1 A511          LDA $11     ; LO
3920 BEE3 85E2    END   STA $E2     ; LO
3930 BEE5 84E3          STY $E3     ; HI
3940                ;
3950                ;
3960                ; SUBROUTINE OM 00BC EEN LETTER TERUG TE ZETTEN!
3970                ;
3980 BEE7 A6C3    DECBC LDX $C3
3990 BEE9 D002          BNE END2
4000 BEEB C6C4          DEC $C4
4010 BEED C6C3    END2  DEC $C3
4020 BEEF 60            RTS
4030                ;
4040                ;
4050                ; SUBROUTINE VOOR TOOLKIT, TEGENGESTELDE VAN HEX-
4060                ; ASIGNMENT!
4070                ;
4080 BEF0 20C2BE  REVHEX JSR E076
4090 BEF3 A2E1          LDX #$E1
4100 BEF5 4C2BBE        JMP HEXOU2
4110                ;
4120                ;
4130                ; SUBROUTINE OM INPUTVECTOR TE RESETTEN
4140                ; (TEN BEHOEVE VAN DE TOOLKIT)!
4150                ;
4160 BEF8 A9BA    RESINP LDA #$BA
4170 BEFA A0FF          LDY #$FF
```

```
4180                ;
4190                ;
4200                ; SUBROUTINE OM INPUTVECTOR TE SETTEN:
4210                ;
4220 BEFC 8D1802    SETINP STA $021B
4230 BEFF 8C1902           STY $021C
4240 BF02 60               RTS
4250                ;
4260                ;
4270                ; SUBROUTINE OM OUTPUTVECTOR TE RESETTEN:
4280                ;
4290 BF03 A969      RESOUT LDA #$69
4300 BF05 A0FF             LDY #$FF
4310                ;
4320                ;
4330                ; SUBROUTINE OM OUTPUTVECTOR TE SETTEN:
4340                ;
4350 BF07 8D1A02    SETOUT STA $021A
4360 BF0A 8C1B02           STY $021B
4370 BF0D 60               RTS
4380
4390                ;
4400                ; SUBROUTINE OM BIJ A=USR(0) OP KEYBOARD
4410                ; TE WACHTEN EN WAARDE AAN A TE GEVEN:
4420                ;
4430 BF0E 2000FD    USR    JSR $FD00
4440 BF11 A8               TAY
4450 BF12 A900             LDA #$00
4460 BF14 4CC1AF           JMP $AFC1
4470                ;
4480                ;
4490                ; SUBROUTINE VOOR ERRORMELDING, TEN
4500                ; BEHOEVE VAN DE TOOLKIT:
4510                ;
4520 BF17 206CA8    ERR    JSR $A86C   ; LINEFEED.
4530 BF1A A986             LDA #$86
4540 BF1C A0A1             LDY #$A1
4550 BF1E 20C3A8           JSR $A8C3
4560 BF21 4C00FB           JMP $FB00   ; BELL.
4570                ;
4580                ;
4590                ; SUBROUTINE VOOR ZELFMOORDUITGANG
4600                ; VAN DE TOOLKIT:
4610                ;
4620 BF24 A21C     EX     LDX #$1C
4630 BF26 BDEDBC   EX2    LDA BCED,X
4640 BF29 95BB             STA $BB,X
4650 BF2B CA               DEX
4660 BF2C D0F8             BNE EX2
4670 BF2E 4CAFBD           JMP WARM2
```

```
4600                ;
4690                ;
4700                ; SUBROUTINE RESET WINDOW VOOR TOOLKIT:
4710                ;
4720 BF31 A004      RESWIN LDY #$04
4730 BF33 B9FAFE    RESW2  LDA $FEFA,Y
4740 BF36 992202           STA $0222,Y
4750 BF39 88               DEY
4760 BF3A 10F7             BPL RESW2
4770 BF3C 60               RTS
4780                ;
4790                ;
4800 BFB4                  *=BCED+$02C7
4810 BFB4 FF               .BYTE $FF,$FF,$FF
4810 BFB5 FF
4810 BFB6 FF
4820                ;
4830                ;
4840                ; JUMPTABEL:
4850                ;
4860 BFB7 4C31BF           JMP RESWIN ; RESET WINDOW
4870 BFBA 4C24BF           JMP EX     ; ZELFMOORD EXIT
4880 BFBD 4C17BF           JMP ERR    ; ERROR MELDING
4890 BFC0 4C4CBE           JMP INPUT  ; CONTROLE # TOOLKIT
4900 BFC3 4C0ABD           JMP RESBC  ; RESET 00BC NAAR BEGIN
4910 BFC6 4CE7BE           JMP DECBC  ; ZET 00BC EEN LETTER TERUG
4920 BFC9 4C93BE           JMP HEXASG ; HEXASIGNMENT
4930 BFCC 4CF0BE           JMP REVHEX ; INVERSE HEXASIGNMENT
4940 BFCF 4C8FBD           JMP FREE   ; PRINT FREE $XXXX-$YYYY
4950 BFD2 4C07BF           JMP SETOUT ;   SET OUTPUTVECTOR
4960 BFD5 4C03BF           JMP RESOUT ; RESET OUTPUTVECTOR
4970 BFD8 4CFCBE           JMP SETINP ;   SET  INPUTVECTOR
4980 BFDB 4CF8BE           JMP RESINP ; RESET  INPUTVECTOR
4990 BFDE 4C02BE           JMP HEXIN  ;  MET RESET BC BEGIN BUFFER
5000 BFE1 4C05BE           JMP HEXIN1 ; GEEN RESET BC BEGIN BUFFER
5010 BFE4 4C30BE           JMP HEXOUT ; HEXOU2 ZONDER $
5020 BFE7 4C36BE           JMP HEX2   ; DISPLAY Acc-WAARDE IN HEX
5030 BFEA 4CC2BE           JMP E076   ; DATAIN
5040 BFED 4C43BD           JMP WARM1  ;  MET TOOLKIT-OMZETTING
5050 BFF0 4CAFBD           JMP WARM2  ; GEEN TOOLKIT-OMZETTING
5060                ;
5070                ;
5080                ; OUDE SUBROUTINE DIE DOOR MINITOR NAAR RAM
5090                ; VERPLAATST WORDT EN DERHALVE MOET BLIJVEN
5100                ; STAAN:
5110                ;
5120 BFF3 B900D0           LDA $D000,Y
5130 BFF6 9900D0           STA $D000,Y
5140 BFF9 C8               INY
5150 BFFA 60               RTS
5160
5170 BFFB FF               .BYTE $FF,$FF,$FF,$FF,$FF
5170 BFFC FF
5170 BFFD FF
5170 BFFE FF
5170 BFFF FF
```

TOELICHTING SOURCE BAS 4.3:

Dit programma past in BASIC 4, als deze rom door een eprom wordt vervangen. Door een vernieuwde Koude-start-routine en doordat bij gebruik van een nieuwe monitor de routines aan het einde van deze rom vervangen zijn, is hier voldoende ruimte vrij gekomen en blijft zelfs een gedeelte buiten gebruik.
Eventuele oude software waarin deze oude routines nog gebruikt wordt kan eenvoudig aangepast worden voor de vervangende routines in de monitor.

De voordelen zijn:

1e Verbeterde Koude start, met de mogelijkheid Basicprogramma's bijv. boven in het geheugen te zetten. Gevraagd wordt het adres van het begin (FIRST $----) en van het eind (LAST $----). Als direkt met Cr wordt geantwoord, worden de pointers van $0300 tot einde ram gezet, na een niet-destructieve test van het geheugen. Gevonden waarden worden gemeld met 'FREE $0303-$2000' oid. Het geheel werkt dus hexadecimaal.

2e Met behulp van de toolKit (welke steeds automatisch wordt ingeschakeld na een warme of Koude start) kan vanuit basic worden aangeroepen:
   a. Hexasignment: bijv. POKE #HD400,#H31.
      (werkt ook in berekeningen en zelfs bij het opgeven van een getal bij INPUT).
   b. Inverse hexasignment: bijv. PRINT #I(12*15) geeft $00B4.
   c. Free-melding: bijv. #F geeft FREE $0456-$2000 oid.
   De toolKit-tabel moet hiervoor natuurlijk wel voor de betreffenletters worden aangepast, zodat ze resp. naar HEXASG,REVHEX en FREE verwijzen.

Verder zijn er nog diverse kleine routines opgenomen die nu in de toolKit staan, en kan deze alsmede de Extended Monitor behoorlijk ingekort worden.
De vroegere 'Lees Decimaal' routine ($E076) is verbeterd.
De nieuwe versie heet 'E076' en accepteerd alleen 2 getallen, gescheiden door een Komma.
Een tabel van de bruikbare routines is aan het einde van het programma opgenomen.

Het geheel werkt bij mij probleemloos (ook in Kombinatie met de Picodos) met een monitor versie 4.0. Andere versies moeten echter ook mogelijk zijn.
Voor liefhebbers kan ik eproms programmeren, maar de diverse gedeelten kunnen ook als idee dienen om weer op verder te borduren.

TOM MEIJERING
CALTHORNERBRINK 3
7812 HS EMMEN
tel.05910-10769

```
 10 A2E6                *=$A2E6
 20 A2E6                ;RØUTINE 7B  INSERT LINE
 30 A2E6           ;
 40 A2E6           ; ER WØRDT NU EEN NIEUWE REGEL TØEGEVØEGD
 50 A2E6           ;DE LAGE BYTES MØETEN HET EERST WØRDEN VERPLAATST
 60 A2E6           ;ZØDAT DE <BLØCKMØVE> VAN $A1CF NIET KØN WØRDEN
 70 A2E6           ;GEBRUIKT.
 80 A2E6           ;
 90 A2E6 A513      LDA $13       ;IS DE REGEL LEEG (DELETE) ?
100 A2E8 F02F      BEQ $A319     ;ZØJA, DAN BEN JE KLAAR
110 A2EA A585      LDA $85       ;ZØNEE, HAAL <HIMEM> HET EINDADRES
120 A2EC A486      LDY $86       ;V/H BESCHIKBAAR GEHEUGEN
130 A2EE 8581      STA $81       ;EN GEEF AAN VANWAAR STRINGRUIMTE
140 A2F0 8482      STY $82       ;BESCHIKBAAR IS (CLEAR DUS STRINGS)
150 A2F2 A57B      LDA $7B       ;HAAL HET EINDE V/H PRØGR. ØP
160 A2F4 85A6      STA $A6       ;IN EINDADR.V/D HERKØMSTPLEK
170 A2F6 655D      ADC $5D       ;TEL DE REGELLENGTE ERBIJ ØP
180 A2F8 85A4      STA $A4       ;DIT WØRDT HET EINDE V/H TE VER-
190 A2FA A47C      LDY $7C       ; PLAATSEN BLØK. DØE DAT NU ØØK
200 A2FC 84A7      STY $A7       ;VØØR DE HØGE ADRESBYTES (DE LAGE
210 A2FE 9001      BCC $A301     ; STAAN AL IN $A4+$A6)
220 A300 C8        INY
230 A301 84A5      STY $A5
240 A303                         BLKMØV=$A1CF
250 A303 20CFA1    JSR BLKMØV    ;VERPLAATS NU HET BLØK
260 A306 A57F      LDA $7F       ;ER IS EEN GAT,PAS ADMINISTRATIE AAN
270 A308 A480      LDY $80       ;EN ØØK HET HØGE ADRESBYTE
280 A30A 857B      STA $7B       ;EN PAS DE VERWIJZING NAAR HET EINDE
290 A30C 847C      STY $7C       ;V/H PRØGRAMMA AAN (LAAG/HØØG)
300 A30E A45D      LDY $5D       ;HAAL DE REGELLENGTE ØP
310 A310 88        DEY           ;GEBRUIK Y ALS TELLER
320 A311 B90F00    LDA $000F,Y   ;HAAL HET TE VERPLAATSEN BYTE
330 A314 91AA      STA ($AA),Y   ;ZET HET IN DE ØPEN RUIMTE
340 A316 88        DEY
350 A317 10F8      BPL $A311     ;GA DØØR TØT ALLES KLAAR IS
360 A319           ;
370 A319           ;
380 A319                 ;  RØUTINE 8 PAS DE REGELVERWIJZERS
390 A319                 ;           VAN IEDERE BASICREGEL AAN.
400 A319           ;
410 A319           ; NU ALLE TEKSTEN ZIJN AANGEPAST ZULLEN DE VER-
420 A319           ;WIJZINGEN VAN ALLE BASICREGELS NAAR DE VØLGENDE
430 A319           ;REGEL MØETEN WØRDEN AANGEPAST. IN DEZE RØUTINE
440 A319           ;WØRDEN ZE ALLEMAAL ØPNIEUW BEREKEND.
450 A319           ;
460 A319                        RESTBA=$A477
470 A319 2077A4    JSR RESTBA    ;RESET DE BASICPØINTERS.
480 A31C A579      LDA $79       ;HAAL DE VERWIJZING NAAR HET BEGIN
490 A31E A47A      LDY $7A       ; V/H PRØGRAMMA ØP EN
500 A320 8571      STA $71       ;BEWAAR DIE EVENTJES
510 A322 8472      STY $72
520 A324 18        CLC           ;MET EEN SCHØNE LEI BEGINNEN
530 A325 A001      LDY #$01      ;GEBRUIK Y ALS TELLER EN INDEX
540 A327 B171      LDA ($71),Y   ;HAAL DE BYTE ØP UIT DE B.REGEL
550 A329 D003      BNE $A32E     ;INDIEN GEEN 0 VERDERSPRINGEN
560 A32B 4C7DA2    JMP $A27D     ;ANDERS PRØGRAMMAEINDE,KLAAR !
570 A32E A004      LDY #$04      ;SLA REGELADMINISTRATIE ØVER
580 A330 C8        INY           ;INDEX VERHØGEN
590 A331 B171      LDA ($71),Y   ;HAAL DE INHØUD V/D REGEL
600 A333 D0FB      BNE $A330     ;TØT EEN 0 = REGELEINDE
```

```
610 A335 C8       INY           ;REGELEINDE GEVØNDEN, PAS Y AAN
620 A336 98       TYA           ;VØØR DE VØLGENDE REGELSTART
630 A337 6571     ADC $71       ;BEREKEN DE VERWIJZING NAAR DE
640 A339 AA       TAX           ; VØLGENDE REGEL, BEWAAR IN Y
650 A33A A000     LDY #$00      ;BEGIN WEER AAN HET BEGIN VAN
660 A33C 9171     STA ($71),Y   ;DEZELFDE REGEL
670 A33E A572     LDA $72       ;ØØK HET HØGE ADRESBYTE
680 A340 6900     ADC #$00      ;TEL 'N MØGELIJKE CARRY ERBIJ
690 A342 C8       INY           ;VØLGENDE ADMINISTRATIE-PLEK
700 A343 9171     STA ($71),Y   ;VERWIJZING IS NU BEREKEND.
710 A345 8571     STX $71       ;VØLGENDE REGEL LØW-BYTE
720 A347 8572     STA $72       ;EN HIGH-BYTE
730 A349 90DA     BCC $A325     ;SPRING ALTIJD TØT PRØGR.EINDE
740 A34B          ;
750 A34B          ;
760 A34B                ; RØUTINE 8  VUL DE INPUTBUFFER
770 A34B          ;
780 A34B          ; DE INPUTBUFFER WØRDT GEVULD. DE KARAKTERS
790 A34B          ;WØRDEN IN HET BUFFER GEZET TØT EEN <RETURN> IS
800 A34B          ;INGETØETST, MAAR ER WØRDT GEEN CR WEGGEZET.
810 A34B          ;IN PLAATS DAARVAN VØEGT EEN ANDERE RØUTINE EEN
820 A34B          ;0 TØE EN WØRDT ER EEN CR/LF UITGEVØERD.
830 A34B          ;ALLEEN EEN <BACKSPACE> EN <APESTAART> KUNNEN
840 A34B          ;WØRDEN GEBRUIKT ØØM DE TEKST TE KØRRIGEREN.
850 A34B          ;ALS DE BUFFER VØL IS KLINKT DE <BELL> (D'S
860 A34B          ;CHR$(7) ) EN EEN GRAFIES TEKEN BIJ DE ØRIGI-
870 A34B          ;NELE ØSI MØNITØR (THANKS TØ HENK WEVERS).
880 A34B          ;NØRMALE START IS RØUTINE 8C. HET X-REGISTER
890 A34B          ;IS DE PLAATSAANDUIDER IN DE BUFFER, EN ALLEN
900 A34B          ;GELDIGE ASCII SYMBØLEN WØRDEN GEACCEPTEERD.
910 A34B          ;
920 A34B                ; RØUTINE 8A  BACKSPACE
930 A34B          ;
940 A34B 20E5A8   JSR $A8E5     ;PRINT DE INHØUD V/D ACCUMULATØR
950 A34E CA       DEX           ;VERMINDER BUFFERVERWIJZER
960 A34F 1008     BPL $A359     ;GA VERDER MITS BUFFER NIET LEEG
970 A351          ;
980 A351                ; RØUTINE 8B  PRINT ACCU + CR/LF
990 A351          ;
1000 A351 20E5A8  JSR $A8E5     ;PRINT INHØUD V/D ACCUMULATØR
1010 A354 206CA8  JSR $A86C     ;PRINT EEN CR/LF
1020 A357         ;
1030 A357               ; RØUTINE 8C  VUL DE BUFFER TØT DE
1040 A357               ;             <RETURN> INTØETSING
1050 A357         ;
1060 A357 A200    LDX #$00      ;INDEXTELLER NAAR BEGIN BUFFER
1070 A359 2086A3  JSR $A386     ;HAAL EEN KARAKTER ØP (RØUT.9)
1080 A35C C907    CMP #$07      ;WAS DAT EEN BELL ?
1090 A35E F014    BEQ $A374     ;ZØJA VERDER SPRINGEN
1100 A360 C90D    CMP #$0D      ;WAS HET EEN CARRIAGE RETURN ?
1110 A362 F01F    BEQ $A383     ;ZØJA NAAR EINDE INPUT
1120 A364 C920    CMP #$20      ;ANDERE KØNTRØLEKARAKTERS ?
1130 A366 90F1    BCC $A359     ;ZØJA NEGEREN EN ØPNIEUW
1140 A368 C97D    CMP #$7D      ;BUITEN HET ASCII GEBIED ?
1150 A36A B0ED    BCS $A359     ;ZØJA, NIET TØEGESTAAN, ØPNIEUW
1160 A36C C940    CMP #$40      ;WAS HET EEN <APESTAART> ?
1170 A36E F0E1    BEQ $A351     ;ZØJA HELEMAAL ØPNIEUW BEGINNEN
1180 A370 C95F    CMP #$5F      ;WAS HET EEN <BACKSPACE> ?
1190 A372 F0D7    BEQ $A34B     ;ZØJA UITVØEREN (RØUT.8A)
1200 A374 E047    CPX #$47      ;IS HET BUFFER VØL ?
1210 A376 B004    BCS $A37C     ;ZØJA AANGEVEN
1220 A378 9513    STA $13,X     ;ANDERS KARAKTER TØEVØEGEN
1230 A37A E8      INX           ;INDEXTELLER VERHØGEN
```

```
1240 A37B          ; 2C TRUUK ØM DØØR HET PRØGRAMMA TE GAAN
1250 A37B          ; 2C A9 07 IS EEN NEP-ØPDRACHT NL BIT $07A9
1260 A37B 2C       .BYTE $2C
1270 A37C A907     LDA #$07        ;LAAT DE BELL HØREN
1280 A37E 20E5A8   JSR $A8E5       ;STUUR HET KARAKTER WEG
1290 A381 D0D6     BNE $A359       ;SPRING ALTIJD TERUG
1300 A383 4C66A8   JMP $A866       ;SPRING NAAR EINDE INPUT
1310 A386          ;
1320 A386          ;
1330 A386                ; RØUTINE 9  HAAL EEN KARAKTER ØP
1340 A386          ;
1350 A386          ; DIT IS EEN ALGEMENE RØUTINE ØM EEN BYTE UIT EEN
1360 A386          ;INVØERAPPARAAT (TØETSENBØRD,CASSETTEINGANG)
1370 A386          ;ØP TE HALEN. DE VELE NØP'S ZIJN EEN ØVERBLIJFSEL
1380 A386          ;UIT ØUDERE (TELETYPE) VERSIES V/D INTERPRETER.
1390 A386          ;
1400 A386                          INPUT=$FFEB
1410 A386 20EBFF   JSR INPUT       ;HAAL KARAKTER ØP
1420 A389 EAEAEA..NØP NØP NØP ..14 MAAL TØTAAL
1560 A397 297F     AND #$7F        ;STRIP HØØGSTE BIT
1570 A399 C90F     CMP #$0F        ;WAS HET EEN <CTRL-Ø> ?
1580 A39B D008     BNE $A3A5       ;ZØNEE VERDER NAAR KLAAR
1590 A39D 48       PHA             ;ZØJA, BEWAREN ØP DE STACK
1600 A39E A564     LDA $64         ;HAAL DE ØUTPUTFLAG ØP
1610 A3A0 49FF     EØR #$FF        ;KEER DE ØUTPUTFLAG ØM
1620 A3A2 8564     STA $64         ;EN TERUGZETTEN
1630 A3A4 68       PLA             ;EN HAAL DE INHØUD VAN A TERUG
1640 A3A5 60       RTS             ;KLAAR
1650 A3A6          ;
1660 A3A6          ;
1670 A3A6                ; RØUTINE 10  TØKENISE BUFFER
1680 A3A6          ;
1690 A3A6          ; IN DEZE RØUTINE WØRDEN DE BASICREGELS IN-
1700 A3A6          ;GEKØRT ØM ZØDØENDE RUIMTE TE SPAREN.
1710 A3A6          ;ALLE BASIC-CØMMANDØ'S WØRDEN VERVANGEN DØØR
1720 A3A6          ;HUN 'TØKEN', EEN KØDE-BYTE MET EEN 1 ALS
1730 A3A6          ;MEEST SIGNIFIKANTE BIT. DE WAARDE VAN DE
1740 A3A6          ;TØKEN WØRDT BEPAALD DØØR HAAR PLAATS IN
1750 A3A6          ;TABEL 4. HET TØKEN VØØR TØ IS $9D (HET
1760 A3A6          ;30-E TØKEN ALS JE BEGINT TE TELLEN MET 0,
1770 A3A6          ;PLUS $80).
1780 A3A6          ;HET X-REGISTER WØRDT GEBRUIKT ALS INDEXTELLER
1790 A3A6          ;VØØR DE INPUT, Y ALS TELLER VØØR DE ØUTPUT.
1800 A3A6          ;ØMDAT DE INGEKØRT KØDE KØRTER IS DAN DE ØRI-
1810 A3A6          ;GINELE KØDE WØRDT DE ØUTPUTSTRING WEGGESCHREVEN
1820 A3A6          ;ØP DE PLAATS VAN DE INPUTSTRING !
1830 A3A6          ;
1840 A3A6 A6C3     LDX $C3         ;ZET DE INPUTPØINTER KLAAR
1850 A3A8 A004     LDY #$04        ;EN DE ØUTPUTPØINTER.
1860 A3AA 8460     STY $60         ;IN DATATYPE-VLAG 'GEEN DATA'
1870 A3AC B500     LDA $00,X       ;HAAL EEN KARAKTER UIT INPUT
1880 A3AE C920     CMP #$20        ;IS HET EEN SPATIE ?
1890 A3B0 F03A     BEQ $A3EC       ;ZØJA KØPIEREN
1900 A3B2 855C     STA $5C         ;MØGELIJK EEN STRINGAFSLUITER
1910 A3B4 C922     CMP #$22        ;IS HET EEN '"' ?
1920 A3B6 F058     BEQ $A410       ;ZØJA DIT EN WAT VØLGT KØPIEREN
1930 A3B8 2460     BIT $60         ;BEZIG IN EEN DATA-ØPDRACHT ?
1940 A3BA 7030     BVS $A3EC       ;ZØJA GAAN KØPIEREN.
1950 A3BC C93F     CMP #$3F        ;IS HET EEN '?' ?
1960 A3BE D004     BNE $A3C4       ;ZØNEE VØLGENDE ØPDRACHT ØVERSLAAN
1970 A3C0 A997     LDA #$97        ;VERVANG ? DØØR PRINT-TØKEN
```

```
1980 A3C2 D028     BNE $A3EC     ;ALTIJD SPRINGEN EN AFDRUKKEN
1990 A3C4 C930     CMP #$30      ;IS HET EEN GETAL ?
2000 A3C6 9004     BCC $A3CC     ;ZØNIET VERDER SPRINGEN
2010 A3C8 C93C     CMP #$3C      ;MØGELIJK, NØGMAALS TESTEN
2020 A3CA 9020     BCC $A3EC     ;HET IS EEN GETAL, KØPIEREN DUS
2030 A3CC 84BA     STY $BA       ;BEWAAR ØUTPUTINDEXTELLER
2040 A3CE A000     LDY #$00      ;GA TABEL MET NAMEN DØØRLØPEN
2050 A3D0 845D     STY $5D       ;ZET NAMEN TELLER KLAAR
2060 A3D2 88       DEY           ;EEN PLEK VØØR DE TABEL
2070 A3D3 863C     STX $3C       ;BEWAAR INPUTINDEXTELLER
2080 A3D5 CA       DEX           ;EEN PLEK VØØR DE INPUTSTRING
2090 A3D6 C8       INY           ;VØLGENDE PLAATS IN TABEL 5
2100 A3D7 E8       INX           ;VØLGENDE PLAATS IN INPUTSTRING
2110 A3D8 B500     LDA $00,X     ;HAAL VØLGENDE KARAKTER ØP
2120 A3DA C920     CMP #$20      ;IS DAT EEN SPATIE ?
2130 A3DC F0F9     BEQ $A3D7     ;ZØJA VØLGENDE INPUT KARAKTER
2140 A3DE 38       SEC           ;GA REKENEN
2150 A3DF F984A0   SBC $A084,Y   ;GELIJK AAN KARAKTER IN DE TABEL ?
2160 A3E2 F0F2     BEQ $A3D6     ;ZØJA DE VØLGENDE PRØBEREN
2170 A3E4 C980     CMP #$80      ;GELIJK AAN TØKEN NR 1?
2180 A3E6 D02F     BNE $A417     ;ZØNEE VØLGENDE TØKEN PRØBEREN
2190 A3E8 055D     ØRA $5D       ;BEREKEN DE TØKENWAARDE
2200 A3EA A4BA     LDY $BA       ;BEWAAR ØUTPUT-INDEX-TELLER
2210 A3EC E8       INX           ;VØLGENDE PLEK IN DE INPUTBUFFER
2220 A3ED C8       INY           ;VØLGENDE PLEK IN ØUTPUTSTRING
2230 A3EE 990E00   STA $000E,Y   ;KØPIEER KARAKTER ØF TØKEN
2240 A3F1 B90E00   LDA $000E,Y   ;HERSTEL STATUSREGISTER
2250 A3F4 F034     BEQ $A42A     ;EEN 0? DAN EINDE INPUTSTRING
2260 A3F6 38       SEC           ;GA KIJKEN ØF HET EEN ':' IS
2270 A3F7 E93A     SBC #$3A
2280 A3F9 F004     BEQ $A3FF     ;ZØJA DATAVLAG HERSTELLEN
2290 A3FB C949     CMP #$49      ;IS HET EEN 'DATA' TØKEN ?
2300 A3FD D002     BNE $A401     ;NEE, LAAT DATA-VLAG ZELFDE
2310 A3FF 8560     STA $60       ;JA, VERANDER DE DATAVLAG
2320 A401 38       SEC
2330 A402 E954     SBC #$54      ;IS HET EEN 'REM' TØKEN ?
2340 A404 D0A6     BNE $A3AC     ;ZØNEE, ØPNIEUW NAAR BEGIN
2350 A406 855C     STA $5C       ;ZET STRING-AFSLUITER ØP 0
2360 A408 B500     LDA $00,X     ;HAAL EEN TEKEN UIT INPUTSTRING
2370 A40A F0E0     BEQ $A3EC     ;BIJ EEN 0 EINDE , KØPIEREN
2380 A40C C55C     CMP $5C       ;KØNTRØLEER STRINGAFSLUITER
2390 A40E F0DC     BEQ $A3EC     ;ØØK KØPIEREN
2400 A410 C8       INY           ;VØLGENDE TEKEN IN ØUTPUTSTRING
2410 A411 990E00   STA $000E,Y   ;KØPIEER TEKEN EVEN
2420 A414 E8       INX           ;VØLGENDE PLEK IN INPUTSTRING
2430 A415 D0F1     BNE $A408     ;HERHALEN
2440 A417 A6C3     LDX $C3       ;GEEN KEYWØRD GEVØNDEN TØTNUTØE
2450 A419 E65D     INC $5D       ;VØLGENDE KEYWØRD-TELLER
2460 A41B C8       INY
2470 A41C B983A0   LDA $A083,Y   ;EINDE HUIDIGE KEYWØRD ?
2480 A41F 10FA     BPL $A41B     ;ZØNEE TERUGSPRINGEN
2490 A421 B984A0   LDA $A084,Y   ;EINDE TABEL MET KEYWØRDS
2500 A424 D0B2     BNE $A3D8     ;NEE, GA VERDER ZØEKEN
2510 A426 B500     LDA $00,X     ;JA,BEWAAR INPUTTEKEN
2520 A428 10C0     BPL $A3EA     ;ALTIJD RETUGSPRINGEN
2530 A42A 991000   STA $0010,Y   ;KLAAR,ZET 0 ALS EINDE ØUTPUT
2540 A42D A912     LDA #$12      ;HERSTEL PARSING RØUTINE
2550 A42F 85C3     STA $C3       ;NAAR START VAN ØMGEZETTE STRING
2560 A431 60       RTS           ;EN TERUG NAAR ØPRØEPER
2570 A432                        ;
2580 A432                        ;
2590 A432                        ;
2600 A432                        ;
```

DE VØLGENDE SUBRØUTINE KAN WØRDEN GEBRUIKT IN
KØMBINATIE MET BASICØDE. MET WAT VERNUFT IS HIJ
ØØK VEL VØØR ALGEMEEN GEBRUIK ØM TE BØUVEN.
DØØR DE REGELS 211 EN 212 TE VERANDEREN, IS HET
SAMEN MET DE SUBRØUTINE ØP 20000- MØGELIJK ØM
DØØR <ESC> IN TE TØETSEN HET GEHELE SCHERM UIT TE
LATEN PRINTEN ØP EEN PRINTER, ØF NAAR EEN MØDEM
TE STUREN. DØØRDAT DE ZENDRØUTINE (ØP 22900 EN 22910)
ZELF VEER EEN SUBRØUTINE VØRMT IS AANPASSING AAN
ANDERE VØRMEN VAN VEGZENDEN (ZØALS PARALLEL) ZØNDER
VEEL PRØBLEMEN MØGELIJK

```
20000 REM REGELPRINTER ØSI SUPERBØARD
20010 REM LET ØP: DE ALLERBØVENSTE REGEL VAN HET
20020 REM SCHERM IS DE ØPDRACHTREGEL.
20030 VE=-1:H0=00:G0SUB23110
20040 PRINT"HET HELE SCHERM ?";:G0SUB210:PRINTIN$;
20050 IFIN$="J"THENT1=0:T2=02:G0T020200
20060 IFIN$<>"N"THENG0SUB23000:RETURN
20070 G0SUB23000
20080 INPUT"VANAF WELKE REGEL VAN BØVEN ";T1
20090 T1=INT(T1):IFT1<00RT1>02THEN20070
20100 VE=-1:H0=00:G0SUB23110
20110 G0SUB23000
20120 INPUT"TØT AAN REGELNUMMER ";T2
20130 T2=INT(T2):IFT2<T10R02<T2THEN20100
20200 VE=-1:H0=00:G0SUB23110
20210 G0SUB23000
20220 PRINT"ZET EERST PRINTER AAN,TØETS <RETURN>";
20230 G0SUB210:IFASC(IN$)<>13THEN20230
20240 G0SUB23000
20250 07=64:P0KE55262,20:IFPEEK(55262)<>20THEN07=32
20260 04=53248+07*(01-1)+00:09=13:G0SUB22910
20270 F0R05=T1T0T2
20280 F0R06=0T003
20290 0R=04+05*07+06
20300 G0SUB22900:NEXT06
20310 09=13:G0SUB22910:09=10:G0SUB22910:NEXT05
20320 VE=-1:H0=00:G0SUB23110
20330 G0SUB23000:PRINT"AFDRUKKEN GEREED"CHR$(7);
20340 F0R05=1T02000:NEXT05:G0SUB23000
20350 RETURN
22900 09=PEEK(08)
22910 WAIT61440,2:P0KE61441,09:RETURN
23000 PRINTCHR$(13);CHR$(5);:RETURN
23110 P0KE14,1:P0KE553,00+H0:P0KE554,01-1+VE
23111 PRINTCHR$(10);:RETURN
```

```
READY
LIST 210-219

 210 G0SUB 200:IF IN$="" THEN 210
 211 IFASC(IN$)=27THENG0SUB20000
 212 RETURN

READY
```

DE REKENWIJZE VAN SBC VERKT NØGAL INGEVIKKELD,
VANDAAR DEZE UITLEG.
1. BEPAAL HET KØMPLEMENT VAN HET GETAL
2. TEL DAARBIJ DE CARRYBIT
3. EN TEL DIT RESULTAAT ØP BIJ DE INHØUD VAN DE
   ACCUMULATØR.
BIJ WIJZE VAN VØØRBEELD VØRDEN EEN AANTAL VAARDES
UITGEREKEND VØØR SBC #$30 EN SBC #$D0.

```
          $30 = 0011 0000   $D0 = 1101 0000
KØMPLEMENTAIR  1100 1111         0010 1111
MET CARRYBIT            1                1
               ---------+       ---------+
               1101 0000         0011 0000

               1101 0000         0011 0000
MET ACCU=$39   0011 1001         0000 1001
               ---------+       ---------+
RESULTAAT C=1  0000 1001    C=0  0011 1001

               1101 0000         0011 0000
MET ACCU=$30   0011 0000         0000 0000
               ---------+       ---------+
RESULTAAT C=1  0000 0000    C=0  0011 0000

               1101 0000         0011 0000
MET ACCU=$2F   0010 1111         1111 1111
               ---------+       ---------+
RESULTAAT C=0  1111 1111    C=1  0010 1111

               1101 0000         0011 0000
MET ACCU=$1F   0001 1111         1110 1111
               ---------+       ---------+
RESULTAAT C=0  1110 1111    C=1  0001 1111
```

JØS BURGHØUTS

DE <CHARGET> SUBRØUTINE ($00BC-00D3)

HAALT HET VØLGENDE TEKEN ØP IN DE BASIC-REGEL (ØF UIT
DE INPUT-BUFFER ØF VAAR ØØK VANDAAN), ZØDRA DAT GRØTER
ØF GELIJK IS ALS $3A DAN KLAAR.
SLA DE SPATIES SIMPELWEG ØVER.
DE REKENPARTIJ, DIE HIERØNDER VERDER VØRDT UITGELEGD,
ZØRGT ERVØØR DAT ALTIJD DE CARRY-VLAG VØRDT GEZET,
BEHALVE BIJ DE TEKENS $30 T/M $39 (DAT ZIJN TØEVALLIG
DE ASCII-WAARDEN VØØR DE CIJFERS 0 T/M 9).
DE INHØUD VAN DE GEHEUGENPLAATS VØRDT IN DE ACCUM.
ØPGESLAGEN. DE X- EN Y-REGISTERS VERANDEREN NIET.
DE RØUTINE VØRDT GEKØPIEERD VAN $BCEE T/M $BD05 MET
BEHULP VAN DE RØUTINE ØP $BD4C-BD54, DIE BIJ DE
KØUDE START VØRDT DØØRLØPEN (EN NIET ALS SUBRØUTINE
AANSPREEKBAAR IS).

DAN NU DE RØUTINE;

```
$00BC E6C3    INC $C3       NEEM HET VØLGENDE ADRES (LØW)
$00BE D002    BNE $00C2     SPRING ALS GEEN PAGINA VØRDT
                            ØVERSCHREDEN, ANDERS
$00CD E6C4    INC $C4       PAGINANUMMER VERHØGEN
$00C2 AD**** LDA $****     HAAL HET TEKEN ØP
$00C5 C93A    CMP #$3A ':  KØNTRØLEER ØF HET EEN : IS
$00C7 B00A    BCS $00D3     GRØTER ØF GELIJK $3A=KLAAR
$00C9 C920    CMP #$20      WAS HET EEN SPATIE ?
$00CB F0EF    BEQ $00BC     ZØJA,VØLGENDE TEKEN HALEN
$00CD 38      SEC           ANDERS DE CARRYVLAG CLEAREN
$00CE E930    SBC #$30      VØØR DE GETALLEN VAN
$00D0 38      SEC           $30 T/M $39
$00D1 E9D0    SBC #$D0
$00D3 60      RTS           KLAAR
```

DE ØØRSPRØNKELIJKE ØSI-EXTENDED MØNITØR VULDE HET
GEHEUGENGEBIED VANAF $D0. DAARDØØR KØN BASIC NIET MEER
ØPSTARTEN NA GEBRUIK VAN DE E.M. ZØNDER KØUDE START.
IN DE VEELGERØEMDE 'TØØLKIT' VØRDT DIT PRØBLEEM ØPGE-
LØST DØØR $D0 T/M $D3 SIMPEL NIET TE GEBRUIKEN.
DE 'TØØLKIT' GEBRUIKT ØVERIGENS DE PARSER-RØUTINE ØP
$00BC-$00D3 VEL. ØP $00C5 STAAT IN PLAATS VAN
CMP #$3A, JMP $9048, VAARMEE EEN ØMLEIDING VØRDT
GEMAAKT ØM TE KIJKEN ØF EEN '#' IS GEVØNDEN.
MET DE 'SUICIDE EXIT', HET #X KØMMANDØ ØP $9FF0
VØRDT DE NØRMALE PARSER WEER HERSTELD.

SUBRØUTINES UIT DE MØNITØR
==========

```
      $F800  'TVØUT'; KØNTRØLEER ØF ER EEN KØNTRØLE KARAKTER IS
                      INGETØETST; ZØJA UITVØEREN
      >>> IN; MØGELIJK EEN SCHERMKØNTRØLEKARAKTER IN A
      <<<UIT; A; X EN Y ØNVERANDERD
      +++    ; IN $0202=TEMP1 STAAT WAARDE V/D ACCUMULATØR

F800 8D0202 STA $0202   'TVØUT' INGEDRUKTE TØETS WØRDT BEWAARD(TEMP1)
F803 48     PHA         +
F804 8A     TXA         +
F805 48     PHA         + REGISTERS BEWAARD
F806 98     TYA         +
F807 48     PHA         +
F808 AD0202 LDA $0202
F80B A000   LDY #$00
F80D EA     NØP             KØNTRØLE KARAKTES WØRDEN UITGEDEKØDEERD
F80E C919   CMP #$19
F810 D003   BNE $F815
F812 2058FF JSR $FF58   ] GA NAAR SCHERMØPBØUW 48*12
F815 C918   CMP #$18
F817 D003   BNE $F81C
F819 2047FF JSR $FF47   ] GA NAAR <SET24>
F81C C90A   CMP #$0A
F81E D003   BNE $F823
F820 20D5F8 JSR $F8D5   ] GA NAAR <LINE-FEED>
F823 C90D   CMP #$0D
F825 D003   BNE $F82A
F827 20B9F9 JSR $F9B9   ] GA NAAR <CR> CARRIAGE RETURN
F82A C95F   CMP #$5F
F82C D003   BNE $F831
F82E 20E0F9 JSR $F9E0   ] GA NAAR <BACK-SPACE>
F831 C903   CMP #$03
F833 D003   BNE $F838
F835 201BFA JSR $FA1B   ] GA NAAR <CLS>
F838 C902   CMP #$02
F83A D003   BNE $F83F
F83C 2093F9 JSR $F993   ] GA NAAR <HØME>
F83F C901   CMP #$01
F841 D003   BNE $F846
F843 2054F9 JSR $F954   ] GA NAAR <EØS> END ØF SCREEN
F846 C905   CMP #$05
F848 D003   BNE $F84D
F84A 20C5F9 JSR $F9C5   ] GA NAAR <EØL> END ØF LINE
F84D C912   CMP #$12
F84F D003   BNE $F854
F851 2038FA JSR $FA38   ] GA NAAR <RIGHT>
F854 C915   CMP #$15
F856 D003   BNE $F85B
F858 2027FA JSR $FA27   ] GA NAAR <UP>
F85B C90C   CMP #$0C
F85D D003   BNE $F862
F85F 2021FA JSR $FA21   ] GA NAAR <LEFT>
```

```
F862 C91A    CMP #$1A
F864 D003    BNE $F869
F866 2052FF  JSR $FF52   ] GA NAAR <WINDØW SCHØØN> + <HØME>
F869 C907    CMP #$07
F86B D003    BNE $F870
F86D 20D0FB  JSR $FBD0   ] GA NAAR <BELL>
F870 C909    CMP #$09
F872 D003    BNE $F877
F874 2044FA  JSR $FA44   ] GA NAAR <INSERT>
F877 C904    CMP #$04
F879 D003    BNE $F87E
F87B 207CFA  JSR $FA7C   ] GA NAAR <DELETE>
F87E C920    CMP #$20
F880 9003    BCC $F885
F882 203EFA  JSR $FA3E   ] GA NAAR <SPATIE>
F885 68      PLA         +
F886 A8      TAY         + HAAL ØUDE REGISTERINHØUDEN ØP
F887 68      PLA         +
F888 AA      TAX         +
F889 68      PLA                 EINDE UITDEKØDERING KØNTRØLE-KARAKTES
F88A 60      RTS
```

```
$F88B  'BERCUR'; BEREKEN DE ABSØLUTE VIDEØPLAATS V/D KURSØR
                 EN ZET DIE IN $022C/D (LØW/HIGH BYTE)
                 DEZE PLEK WØRDT BEREKEND UIT $0229/A
                 RESP. -RØW- EN -LINE-
>>> IN; -RØW- EN -LINE-
<<<UIT; X=0; Y=ØNVERANDERD; A=PAGENR. VIDEØADRES = $022D
```

```
F88B A900    LDA #$00    'BERCUR' BEREKEN DE KURSØRPLAATS
F88D 8D2D02  STA $022D       ABSØLUTE PLAATS IN $022D EN $022C
F890 AD2A02  LDA $022A       EN RELATIEVE PLAATS IN \RØW\ $0229
F893 8D2C02  STA $022C                          EN \LINE\ $022A
F896 A205    LDX #$05
F898 0E2C02  ASL $022C
F89B 2E2D02  ROL $022D
F89E CA      DEX
F89F D0F7    BNE $F898       HAAL \VIDTYPE\ ØP EN
F8A1 AD2602  LDA $0226       SPRING BIJ EEN 24*24 SCHERMØPBØUW
F8A4 F006    BEQ $F8AC
F8A6 0E2C02  ASL $022C
F8A9 2E2D02  ROL $022D
F8AC 18      CLC             HAAL \RØW\ ØP
F8AD AD2902  LDA $0229
F8B0 6D2C02  ADC $022C
F8B3 8D2C02  STA $022C
F8B6 A9D0    LDA #$D0        HET PAGE-NR. WAS BEREKEND IN $022D
F8B8 6D2D02  ADC $022D
F8BB 8D2D02  STA $022D
F8BE 60      RTS
```

```
$F8BF  'RESTCU'; ZET HET KARAKTER IN ØLDCH($0201) ØP DE PLAATS
                VAN DE KURSØR.
($F8C2 'RESTCU1'; ZET HET KARAKTER IN A ØP DE PLEK V/D KURSØR)
>>> IN; EVENTUEEL KARAKTER IN $0201
<<<UIT; Y=0; X=ØNVERANDERD; A=-ØLDCH- =$0201
+++    ;VIDEØADRES ØP $022C/D HEEFT INHØUD VAN A


F8BF AD0102 LDA $0201  'RESCU' ZET HET ØUDE KARAKTER ØP DE PLAATS
F8C2 A08D   LDY #$8D   = STA ABSØLUUT       \\          V/D KURSØR
F8C4 8C2B02 STY $022B  'RESCU1' ZET KARAKTER IN ACCU ØP KURSØRPLEK
F8C7 A060   LDY #$60   = RTS  WAARMEE IN BAG.2 EEN VIDEØ-HULPRØU-
F8C9 8C2E02 STY $022E          TINE IS ØPGEBØUVD.
F8CC A000   LDY #$00
F8CE 4C2B02 JMP $022B  ) GA NAAR DIE RØUTINE !


$F8D1  'VIDGET'; HAAL HET KARAKTER IN DE VIDEØ ØP
>>> IN;
<<<UIT; Y=0; X=ØNVERANDERD; A=INHØUD VIDEØADRES ØP $022C/D
+++    ; ØP $022B STAAT $AD; STA ABSØLUUT

F8D1 A0AD   LDY #$AD   = LDA ABSØLUUT VØØR ØPBØUW VIDEØHULPRØUTINE
F8D3 D0EF   BNE $F8C4  ) SPRING ALTIJD NAAR <RESCU>


$F8D5  'LF' ; GEEFT NIEUWE REGEL EN SCRØLLT INDIEN NØDIG
>>> IN;
<<<UIT; INDIEN ER GEEN SCRØLLING NØDIG WAS
                A=$0201; X=0; Y=0
.      ; INDIEN DE ØNDERKANT V/H SCHERM WAS BEREIKT;
                A=0; X=$0201; Y=0

F8D5 20BFF8 JSR $F8BF  'LF' GEEF EEN LINE-FEED EERST NAAR <RESCU>
F8D8 AD2A02 LDA $022A      VERGELIJK HUIDIGE REGELNR MET
F8DB CD2502 CMP $0225      MAXIMALE REGELNUMMER
F8DE B006   BCS $F8E6      MAXIMUM BEREIKT ? DAN NAAR <SCRØLL>
F8E0 EE2A02 INC $022A      ANDERS HUIDIG REGELNR. VERHØGEN EN
F8E3 4CE9F8 JMP $F8E9      KURSØR WEGZETTEN MET <VRCUR>
F8E6 4C07F9 JMP $F907  ) GA NAAR <SCRØLL>


$F8E9  'WRCUR' ZET DE KURSØR ØP DE HUIDIGE KURSØRPLEK
>>> IN;
<<<UIT; A=0; X=$0201; Y=0
+++    ; ØP $0201 STAAT HET GØEDE KURSØRTEKEN $5F ØF $BB

F8E9 A95F   LDA #$5F   'WRCUR' ZET KURSØR ØP HUIDIGE KURSØRPLEK
F8EB AE2702 LDX $0227      KØNTRØLEER MET DE \EDITFL\ ØF DAT EEN
F8EE 3002   BMI $F8F2      CØMMAND-KURSØR ØF
F8F0 A9BB   LDA #$BB       EDIT-KURSØR MØET ZIJN
```

```
$F8F2  'WRCHAR'; SCHRIJFT KARAKTER IN A ØP HET SCHERM ZØNDER
               DE KURSØRPLAATS TE VERHØGEN.
>>> IN; KARAKTER IN A
<<<UIT;X=$0201; Y=0; A=0


F8F2 8D0102 STA $0201  'WRCHAR' SCHRIJF KARAKTER IN ACCU ØP SCHERM
F8F5 2089F8 JSR $F889  'WRTEK ' ØF TEKEN IN ACCU ; MET <BERCUR>
F8F8 20D1F8 JSR $F8D1      HAAL HET HUIDIGE TEKEN ØP
F8FB AA     TAX            BEWAAR DAT IN HET X-REG.
F8FC AD0102 LDA $0201      HAAL HET TE PRINTEN KARAKTER ØP
F8FF 20C2F8 JSR $F8C2      ZET DAT ØP HET SCHERM MET <RESCU>
F902 8E0102 STX $0201      BERG IN X BEWAARDE TEKEN ØP IN \ØLDCH\
F905 98     TYA
F906 60     RTS



$F907  'SCRØLL'; DE TEKST-WINDØW EEN REGEL ØMHØØG
>>> IN;
<<<UIT;X=0; Y=0; A=$0201 (TEMP1)


F907 18     CLC        'SCRØLL' DE TEKSTWINDØW 1 REGEL ØMHØØG
F908 A207   LDX #$07       HAAL DE SCRØLLING RØUTINE UIT BASIC RØM
F90A BDF3BF LDA $BFF3,X    EN ZET ØP PAG.2  \\ LDA $D000;Y \\
F90D 9D0702 STA $0207,X    STA $D000;Y \\ INY \\ RTS \\
F910 CA     DEX
F911 10F7   BPL $F90A
F913 AD2402 LDA $0224 HAAL DE BEGINREGEL \LINES\ ØP
F916 8D2A02 STA $022A      EN ØP REGELNR V/D KURSØR \LINES\ ZETTEN
F919 AD2202 LDA $0222      BEGINKØLØMPLAATS IN WINDØW \RØWS\ ØPHALEN
F91C 8D2902 STA $0229      EN ØP \RØW\ ZETTEN (=KØLØMNR V/D KURSØR)
F91F 208BF8 JSR $F88B      BEREKEN DE ABSØLUTE (=$) KURSØRPLEK IN <BERCUR>
F922 AD2C02 LDA $022C      + HAAL HET ABSØLUTE KURSØR ADRES ØP
F925 AE2D02 LDX $022D      +
F928 8D0B02 STA $020B      + PLAATS DAT IN HULPSUBRØUTINE BIJ SCRØLLEN
F92B 8E0C02 STX $020C      +                              ($022B)
F92E AD2A02 LDA $022A      HAAL REGELNR V/D KURSØR ØP \LINE\
F931 CD2502 CMP $0225      IS HET DE LAATSTE REGEL TØEGESTAAN ? (\LINEE\)
F934 B018   BCS $F94E      ZØJA; WEGSPRINGEN
F936 EE2A02 INC $022A      NEEM VØLGEND REGELNR VØØR KURSØRPLAATS
F939 208BF8 JSR $F88B      BEREKEN ABSØLUTE KURSØRPLEK IN <BERCUR>
F93C AD2C02 LDA $022C      + HAAL DAT ADRES ØP
F93F AE2D02 LDX $022D      +
F942 8D0802 STA $0208      + EN ZET IN SCRØLL-HULPSUBRØUTINE
F945 8E0902 STX $0209      +
F948 2082F9 JSR $F982  1 NAAR <SCRØLL EEN REGEL>
F94B 4C19F9 JMP $F919  1 NAAR DE VØLGENDE REGEL
F94E AD2202 LDA $0222 HET HELE WINDØW IS ØPGESCRØLLED; NEEM \RØWS\ ØP
F951 8D2902 STA $0229 DAT WØRDT KURSØRKØLØMNR. \RØW\
```

```
$F954  'E0S'; END 0F SCREEN MAAK HET SCHERM SCH00N BINNEN
               DE WINDOW T0T HET EINDE DAARVAN
>>> IN;
<<<UIT;X=$0201 (TEMP1); Y=0; A=0

F954 AD2A02 LDA $022A  'E0S' MAAK WINDOW SCH00N VANAF DE KURS0R
F957 48     PHA            + BEWAAR DE REGEL- EN K0L0M-NUMMER
F958 AD2902 LDA $0229      + \LINE\ EN \R0W\ 0P DE STACK
F95B 48     PHA            ++++++
F95C A920   LDA #$20       DRUK EEN SPATIE AF 0NDER DE KURS0R
F95E 20F2F8 JSR $F8F2      MET <WRCHAR>
F961 AD2902 LDA $0229      KIJK 0F DE KURS0R 0P DE
F964 CD2302 CMP $0223      LAATSTE PLAATS (VAN DE REGEL) STAAT
F967 9008   BCC $F971      Z0NIET; VERDER SPRINGEN
F969 AD2A02 LDA $022A      DE LAATSTE K0L0M IS BEREIKT; HAAL REGELNR.
F96C CD2502 CMP $0225      IS HET DE LAATSTE REGEL ?          0P
F96F B006   BCS $F977      Z0JA; VERDERSPRINGEN
F971 20A5F9 JSR $F9A5      Z0NIET NAAR <INCCUR>
F974 4C5CF9 JMP $F95C      ) EN HET TEKEN SCHRIJVEN
F977 68     PLA            + DE LAATSTE REGEL IS BEREIKT; DUS
F978 8D2902 STA $0229      + HAAL DE BEWAARDE KURS0RPLAATSEN 0P
F97B 68     PLA            +
F97C 8D2A02 STA $022A      ++++++
F97F 4CE9F8 JMP $F8E9      ) EN SPRING NAAR <WRCUR>


$F982  'SCR0LL EEN REGEL' VAN DE TEKSTWINDOW 0MH00G
>>> IN; KURS0RPLAATS 0P $0229/A
<<<UIT; X=0NVERANDERD; Y=TELLER($0223;R0WE); A=R0WE($0223)
+++   ; R0W($0229)=R0WE($0223)

F982 A000   LDY #$00   'SCR0LL EEN REGEL'
F984 200702 JSR $0207      ) <SCR0LL EEN TEKEN>
F987 EE2902 INC $0229      V0LGENDE K0L0MNR. V/D KURS0R \R0W\
F98A AD2302 LDA $0223      + LAATSTE K0L0M IS BEREIKT
F98D CD2902 CMP $0229      ++++++
F990 B0F2   BCS $F984      Z0NIET; N0G EEN KEER
F992 60     RTS


 $F993  'H0ME'; ZET DE KURS0R 0P DE EERSTE PLEK V/D TEKSTWINDOW
 >>> IN;
 <<<UIT; X=0NVERANDERD; Y=0; A=$0201 (TEMP1)

F993 20BFF8 JSR $F8BF  'H0ME'; ) GA NAAR <RESTCU>
F996 AD2402 LDA $0224      + ZET DE KURS0R 0P DE EERSTE REGEL
F999 8D2A02 STA $022A      ++++++
F99C 4CBCF9 JMP $F9BC      ) ZET KURS0R 0P EERSTE K0L0M; GA NAAR
                                                      <WRCUR>

    44 43 57 4D 07 00 ; DCWM<BELL>
```

```
$F9A5  'INCCUR'; ZET DE KURSØR EEN PLAATS VERDER
>>> IN;
<<<UIT; X=$0201 (TEMP1); Y=0; A=0
+++    ; $0229 (-RØW-) VERANDERT EN $022A(-LINE-) BIJ SCRØLLING

F9A5 AD2902 LDA $0229'INCCUR' VERHØØG KURSØRPLEK; ZET KURSØR I PLEK
F9A8 CD2302 CMP $0223     LAATSTE KØLØMPLAATS BEREIKT ?      VERDER
F9AB B006   BCS $F9B3     ALS DAT LAATSTE KØLØM IS; SPRINGEN
F9AD EE2902 INC $0229     ANDERS VØLGENDE KØLØM
F9B0 4CE9F8 JMP $F8E9] NAAR <WRCUR>
F9B3 4CD5FC JMP $FCD5] KØNTRØLEER \LØADVLAG\; GEEF <LF/CR>
F9B6 4CD5F8 JMP $F8D5] GEEF <LF>


 $F9B9  'CR'; ZET DE KURSØR ØP DE EERSTE PLAATS V/D REGEL
 >>> IN;
 <<<UIT; X=0201 (TEMP1); Y=0; A=0
 +++    ; $0229 (-RØW-) VERANDERT (WØRDT $0222;-RØWS-)

F9B9 20BFF8 JSR $F8BF 'CR' ZET KURSØR ØP EERSTE PLAATS V/D ZELFDE
F9BC AD2202 LDA $0222     + ZET KURSØR ØP DE EERSTE KØLØM   REGEL
F9BF 8D2902 STA $0229     +
F9C2 4CE9F8 JMP $F8E9 ] GA NAAR <WRCUR>


$F9C5  'EØL'; MAAK DE REGEL TØT EINDE SCHØØN VANAF DE KURSØR
>>> IN;
<<<UIT; X=$0201; Y=0; A=0
+++    ; $0229/A VERANDEREN.

F9C5 AD2A02 LDA $022A 'EØL' MAAK DE REGEL SCHØØN VANAF KURSØRPLAATS
F9C8 48     PHA           + BEWAAR DE ABSØLUTE KURSØRPLAATS
F9C9 AD2902 LDA $0229     + ØP DE STACK
F9CC 48     PHA           +
F9CD A920   LDA #$20      PRINT EEN SPATIE MET
F9CF 20F2F8 JSR $F8F2     <WRCAR>
F9D2 AD2902 LDA $0229     + AL BIJ DE LAATSTE KØLØM AANGEKØMEN ?
F9D5 CD2302 CMP $0223     +
F9D8 B09D   BCS $F977     ZØJA; NAAR DE <EINDE V/D REGEL> SUB
F9DA 20A5F9 JSR $F9A5     ANDERS NAAR <INCCUR>
F9DD 4CCDF9 JMP $F9CD     EN NØG EEN SPATIE AFDRUKKEN

$F9E0  'BACKSPACE';  ZET DE KURSØR 1 PLAATS TERUG EN ZET
                ØP BEIDE PLAATSEN EEN SPATIE
>>> IN;
<<<UIT; A=0; X=$0201; Y=0
+++    ; $0229 (-RØW-) EN SØMS $022A (-LINE-) MET 1 VERLAAGD

F9E0 A920   LDA #$20   'BACKSPACE' ; HAAL EEN SPATIE ØP EN
F9E2 20F2F8 JSR $F8F2      ] NAAR <WRCHAR>
F9E5 20F0F9 JSR $F9F0      ] NAAR <DECCUR>
F9E8 A920   LDA #$20       HAAL NØG 'N SPATIE ØP
F9EA 20F2F8 JSR $F8F2      ] NAAR <WRCHAR>
F9ED 4CE9F8 JMP $F8E9      ] TENSLØTTE NAAR <WRCUR>
```

```
   $F9F0  'DECCUR';ZET DE KURSØR EEN PLAATS TERUG
                  VERPLAATS DE KURSØR
   >>> IN;
   <<<UIT; X=$0201 (TEMP1); Y=0; A=0
   +++    ; $0229 (-R0W-) EN SØMS $022A (-LINE-) MET 1 VERLAAGD


   F9F0 AD2202 LDA $0222  'DECCUR';TEGENØVERGESTELDE VAN INCCUR
   F9F3 CD2902 CMP $0229     IS DE KURSØR ØP DE EERSTE KØLØM ?
   F9F6 B006   BCS $F9FE     ZØJA; VERDER
   F9F8 CE2902 DEC $0229     VERMINDER KURSØRKØLØMPLAAATS
   F9FB 4CE9F8 JMP $F8E9    ] EN GA NAAR <WRCUR>
   F9FE AD2302 LDA $0223     \ ZET KURSØR ØP DE
   FA01 8D2902 STA $0229     / LAATSTE KØLØM
   FA04 AD2402 LDA $0224     \ IS DE KURSØR ØP DE
   FA07 CD2A02 CMP $022A     / EERSTE REGEL ?
   FA0A B006   BCS $FA10      ZØJA; VERDER SPRINGEN
   FA0C CE2A02 DEC $022A      ZET KURSØR 1 REGEL ØMH00G
   FA0F 4CE9F8 JMP $F8E9     ] GA NAAR <WRCUR>
   FA12 AD2202 LDA $0222      KURSØR ØP EERSTE REGEL, ZET DE
   FA15 8D2902 STA $0229      KURSØR ØP DE EERSTE KØLØM
   FA18 4CE9F8 JMP $F8E9     ] GA NAAR <WRCUR>


   $FA1B  'CLS'; MAAK DE TEKSTWINDØW SCHØØN
   >>> IN;
   <<<UIT; A=0; X=$0201; Y=0
   +++    ; KURSØR STAAT IN DE 'HØME' PØSITIE


   FA1B 2093F9 JSR $F993  'CLS'; GA EERST NAAR HØME
   FA1E 4C54F9 JMP $F954   ] GA NAAR <E0S>


   $FA21  'LEFT'; BEWEEG DE KURSØR EEN PLAAATS NAAR LINKS
                  ZØNDER HET TEKEN ERØNDER AAN TE TASTEN
   >>> IN;
   <<<UIT; A=0; X=$0201; Y=0
   +++    ; ZIE <DECCUR>


   FA21 20BFF8 JSR $F8BF  'LEFT'; GA EERST NAAR <RESTCU>
   FA24 4CF0F9 JMP $F9F0   ] GA NAAR <DECCUR>


   $FA27  'UP'; PLAATS DE KURSØR EEN REGEL ØMH00G
   >>> IN;
   <<<UIT; A=0; X=$0201; Y=0;
   +++    ; ZIE <DECCUR>


   FA27 20BFF8 JSR $F8BF  'UP'; GA EERST NAAR <RESTCU>
   FA2A AD2402 LDA $0224   \ IS DE KURSØR ØP DE
```

```
FA2D CD2A02 CMP $022A   / BØVENSTE REGEL
FA30 B003   BCS $FA35    ZØJA; VERDER SPRINGEN
FA32 CE2A02 DEC $022A    ZØNEE KURSØR REGEL ØMHØØG
FA35 4CE9F8 JMP $F8E9   ) GA NAAR <WRCUR>


$FA38  'RIGHT'; VERPLAATS KURSØR 1 PLAATS NAAR RECHTS, ZØNDE
                HET TEKEN ERØNDER TE VERANDEREN
>>> IN;
<<<UIT; X=$0201; Y=0; A=0;
+++   ; ZIE ØØK <INCCUR>

FA38 20BFF8 JSR $F8BF  'RIGHT'; GA EERST NAAR <RESTCU>
FA3B 4CA5F9 JMP $F9A5   ) GA NAAR <INCCUR>


$FA3E  'WRCHA'; SCHRIHFT KARAKTER IN ACCU ØP HET SCHERM
                EN VERHØØGT DE KURSØRPLAATS
>>> IN; TEKEN IN ACCU
<<<UIT; X=$0201; Y=0; A=0;
+++   ; ZIE ØØK <INCCUR>

FA3E 20F2F8 JSR $F8F2  'WRCHA'; GA EERST NAAR <WRCHAR>
FA41 4CA5F9 JMP $F9A5   ) EN NAAR <INCCUR>


$FA44  'INSERT'; MAAK EEN SPATIE PLAATS ØP DE KURSØRPLEK
>>> IN;
<<<UIT; A=0; X=$0201; Y=0;
+++   ; ZIE ØØK <WRCUR>

FA44 A920   LDA #$20   'INSERT'; HAAL EEN SPATIE ØP
FA46 20C2F8 JSR $FAC2   ) NAAR <RESTCU>
FA49 AD2A02 LDA $022A   \ BEWAAR HUIDIGE KURSØRPLAATS
FA4C 48     PHA         / (ABSØLUUT)
FA4D AD2902 LDA $0229   /
FA50 48     PHA         /
FA51 AD2902 LDA $0229   \ IS DE KURSØR ØP DE LAATSTE
FA54 CD2302 CMP $0223   / KØLØMPLAATS ?
FA57 B009   BCS $FA62   ZØJA; VERDER SPRINGEN
FA59 EE2902 INC $0229   ZØNEE; VERHØØG KØLØMNR V/D KURSØR
FA5C 20F5F8 JSR $F8F5   ) GA NAAR <WRTEK>
FA5F 4C51FA JMP $FA51   ) NAAR JEZELF TERUGSPRINGEN
FA62 AD2202 LDA $0222   \ ZET KURSØR ØP DE
FA65 8D2902 STA $0229   / EERSTE KØLØM
FA68 AD2A02 LDA $022A   \ IS DE KURSØR ØP DE
FA6B CD2502 CMP $0225   / LAATSTE REGEL ?
FA6E B009   BCS $FA79   ZØJA; VERDER SPRINGEN
FA70 EE2A02 INC $022A   ZØNEE; VERHØØG REGELNR V/D KURSØR
```

```
FA73 CE2902 DEC $0229   ZET KURSØR 1 PLAATS TERUG
FA76 4C51FA JMP $FA51   ) SPRING NAAR JEZELF TERUG
FA79 4C77F9 JMP $F977   ) KURSØR WAS ØP DE LAATSTE REGEL
                         HERSTEL ØØRSPR. KURSØR NAAR <WRCUR>
```

```
$FA7C  'DELETE'; VERWIJDER KARAKTER ØNDER DE KURSØR
                 EN PAS DE REST V/H SCHERM AAN
>>> IN;
<<<UIT; A=0; Y=$0201; Y=0;
+++    ; ZIE ØØK <WRCUR>
```

```
FA7C AD2902 LDA $0229  'DELETE'
FA7F 48     PHA         \ BEWAAR HET HUIDIG
FA80 AD2A02 LDA $022A   / KURSØRADRES
FA83 48     PHA         / (ABSØLUUT)
FA84 AD2302 LDA $0223   \ ZET KURSØR ØP LAATSTE KØLØM
FA87 8D2902 STA $0229   /
FA8A AD2502 LDA $0225   \ ZET KURSØR ØP LAATSTE REGEL
FA8D 8D2A02 STA $022A   /
FA90 A220   LDX #$20    HAAL EEN SPATIE ØP
FA92 8EFF02 STX $02FF   EVEN BEWAREN (MØET DIT NIET 022F ZY
FA95 ADFF02 LDA $02FF   HAAL BEWAARDE TEKEN ØP
FA98 8EFF02 STX $02FF   BEWAAR TELLER
FA9B 20F2F8 JSR $F8F2   ) NAAR <WRCAR>
FA9E 68     PLA         HAAL ØUDE REGELNR V/D KURSØR ØP
FA9F CD2A02 CMP $022A   AL ØP DE ØUDE REGEL ?
FAA2 F007   BEQ $FAAB   ZØJA; VERDERSPRINGEN
FAA4 48     PHA         ZØNEE; WEER ØP DE STACK BEWAREN
FAA5 20F0F9 JSR $F9F0   ) NAAR <DECCUR>
FAA8 4C95FA JMP $FA95   ) EN NAAR JEZELF SPRINGEN
FAAB AA     TAX         ØP DE ØUDE REGEL;REGELNR BEWAREN
FAAC 68     PLA         HAAL KØLØMNR ØUDE KURSØRPLEK ØP
FAAD CD2902 CMP $0229   IS DE KURSØR AL DAAR ?
FAB0 D003   BNE $FAB5   ZØNIET; VERDER SPRINGEN
FAB2 4CE9F8 JMP $F8E9   ZØJA; KLAAR !! NAAR <WRCUR>
FAB5 48     PHA         BEWAAR ØUDE KØLØMPLAATS WEER
FAB6 8A     TXA         HAAL ØUDE REGELNR TERUG
FAB7 4CA4FA JMP $FAA4   ) SPRING NAAR JEZELF TERUG
```

HIERNA VØLGT EEN LØSSE VERZAMELING RØUTINES DIE IN HET ALGE WØRDEN GEBRUIKT BIJ HET 'EDITEN' VAN EEN REGEL.

```
FABA EA     NØP         'EDIT RØUTINES'
FABB A488   LDY $88     HAAL DE HØGE BYTE V/H HUIDIGE REGEL
FABD C8     INY
FABE D009   BNE $FAC9   HØØGSTE BYTE GEEN FF (DIREKT MØDE)
FAC0 AC2702 LDY $0227   IN DE DIREKT MØDE, HAAL EDITVLAG
FAC3 1005   BLP $FACA   ER IS EDIT MET REGELNR ØF GEEN E.MØ
```

```
FAC5 C90D   CMP #$0D    ER IS EDIT ZØNDER REGELNR. CR IN AC
FAC7 F074   BEQ $FB3D   ZØJA NAAR BASICPRØGR.HERSTELLER
FAC9 60     RTS
FACA E8     INX
FACB C90D   CMP #$0D    CR INGETØETST ?
FACD F00A   BEQ $FAD9   GA UIT EDITMØDE ALS 0227<>0
FACF C91B   CMP #$1B    ESC INGETØETST ?
FAD1 F00B   BEQ $FADE   ZØJA; ALTIJD UIT EDIT MØDE
FAD3 2000F8 JSR $F800   HAAL DE VØLGENDE TØETS ØP
FAD6 4CB3FF JMP $FFB3   ] NAAR <KEYBA> EN MØDE KØNTRØLE
FAD9 AD2702 LDA $0227   HAAL DE EDIT-VLAG ØP
FADC D005   BNE $FAE3   ER STØND 01 EN GEEN 0
FADE A9FF   LDA #$FF    GA UIT DE EDIT-MØDE
FAE0 8D2702 STA $0227   ZET DE -NØ EDIT- MØDE
FAE3 68     PLA
FAE4 68     PLA
FAE5 68     PLA     HAAL AL DE SUBRØUTINE-CALLS ØP
FAE6 68     PLA
FAE7 68     PLA
FAE8 68     PLA
FAE9 2093F9 JSR $F993   ] NAAR <HØME>
FAEC E020   CPX #$20    WAS HET ØUDE KARAKTER EEN SPATIE ?
FAEE D018   BNE $FB08   NEE;SCHERMREGEL NAAR INPUTBUFFER
FAF0 20A5F9 JSR $F9A5   ZØJA; NAAR <INCCUR>
FAF3 AD2902 LDA $0229   \ IS DE KURSØR ØP DE
FAF6 CD2302 CMP $0223   / LAATSTE KØLØM ?
FAF9 D0F1   BNE $FAEC   ZØNEE; NAAR JEZELF TERUGSPRINGEN
FAFB AD2A02 LDA $022A   \ ZØJA; IS DE KURSØR ØP DE
FAFE CD2502 CMP $0225   / LAATSTE REGEL ?
FB01 D0E9   BNE $FAEC   ZØNEE; NAAR JEZELF TERUGSPRINGEN
FB03 A200   LDX #$00    ZØJA; INPUTBUFFER=LEEG
FB05 4C2CFB JMP $FB2C   EN NAAR AFSLUITING SPRINGEN
FB08 A900   LDA #$00    'SCHERMREGEL NAAR INPUTBUFFER'
FB0A 48     PHA         HØUDT DE TELLING GØED
FB0B 68     PLA         TELLER ØPHALEN
FB0C AA     TAX         EN IN INPUTBUFFER-INDEX ZETTEN
FB0D AD0102 LDA $0201   HAAL -ØLDCH- ØP
FB10 E047   CPX #$47    WAS X AL 71 DEC. (=LENGTE INP.BUFF.)
FB12 B00E   BCS $FB22   ZØJA; VERDERSPRINGEN
FB14 9513   STA $13,X   ZØNEE GA DE INPUTBUFFER VULLEN
FB16 E8     INX         VØLGENDE PLEK
FB17 8A     TXA         TELLER BEWAREN
FB18 48     PHA         ØP DE STACK
FB19 20BFF8 JSR $F8BF   NAAR <RESTCU>
FB1C 20A5F9 JSR $F9A5   NAAR <INCCUR>
FB1F 4C0BFB JMP $F90B   EN HET VØLGENDE KARAKTER HALEN
FB22 CA     DEX         'VERWIJDER SPATIES UIT BUFFER'
FB23 F006   BEQ $FB2B   SPRINGEN BY BEGIN V/D BUFFER
FB25 B413   LDY $13,X   ANDERS LAATSTE KARAKTER HALEN
FB27 C020   CPY #$20    WAS DAT EEN SPATIE ?
FB29 F0F9   BEQ $FB22   ZØJA; TELLER AANPASSEN EN TERUG
FB2B E8     INX         ZØNEE;ZET AAN HET EINDE V/D
FB2C A000   LDY #$00    BUFFER EEN 0 NEER
FB2E 9413   STY $13,X
FB30 201BFA JSR $FA1B   ] GA NAAR <CLS>
```

```
FB33 20C7FB JSR $FBC7   ) NAAR FØUTMELDINGHERSTELLER
FB36 A212   LDX #$12    MAAK ALLE REGISTERS
FB38 A000   LDY #$00    BEDRIJFSKLAAR VØØR DE
FB3A 4C80A2 JMP $A280   ) <LEEG DE BUFFER> RØUTINE
FB3D 68     PLA
FB3E 68     PLA
FB3F 68     PLA         HAAL AL DIE SUBRØUTINE-CALLS ØP
FB40 68     PLA
FB41 68     PLA
FB42 68     PLA
FB43 2066A8 JSR $A866    ZET INPUTBUFFER KLAAR (??)
FB46 86C3   STX $C3      \ ADRES V/D VARIABELE
FB48 84C4   STY $C4      / DIE VØRDT BEVERKT
FB4A 4CE7FC JMP $FCE7   HAAL VØLGEND TEKEN, KØM HIER TERUG
FB4D D003   BNE $FB52   NIET EINDE BUFFER(=0) VERDERSPRINGEN
FB4F 4C7DA2 JMP $A27D   ANDERS TERUG NAAR BASIC (VUL BUFFER)
FB52 A2FF   LDX #$FF    \ HET TEKEN WAS 0 (VAN FB4D)
FB54 8688   STX $88     / ZET DE BASIC IN DE CØMMANDMØDE
FB56 B003   BCS $FB5B
FB58 4C95A2 JMP $A295   ) NAAR BASIC REGEL-EDIT
FB5B 20A6A3 JSR $A3A6   ) GA INPUTBUFFER IN TØKENS ØMZETTEN
FB5E 20BC00 JSR $00BC   ) NAAR <GETCHAR>
FB61 C999   CMP #$99    IS DAT 'LIST' ?
FB63 F03B   BEQ $FBA0   ZØJA; NAAR BASIC MET <CLS>
FB65 C989   CMP #$89    IS DAT 'RUN' ?
FB67 F037   BEQ $FBA0   ZØJA; NAAR BASIC MET <CLS>
FB69 C99B   CMP #$9B    IS DAT 'NEW' ?
FB6B F033   BEQ $FBA0   ZØJA; NAAR BASIC MET <CLS>
FB6D C945   CMP #$45    IS HET EEN 'E' ?
FB6F D032   BEQ $FBA3   ZØNIET; NAAR BASIC ZØNDER <CLS>
FB71 20BC00 JSR $00BC   ZØJA; NAAR <GETCHAR>
FB74 D00A   BNE $FB80   VERDER ZØLANG NIET EEN '0'
FB76 A901   LDA #$01    \ EDIT MET EEN REGELNUMMER
FB78 8D2702 STA $0227   /
FB7B 201BFA JSR $FA1B   ) NAAR <CLS>
FB7E F0CF   BEQ $FB4F   ) ALTIJD NAAR INPUTBUFF.VULLEN+BASIC
FB80 B021   BCS $FBA3
FB82 C930   CMP #$30    IS HET EEN '0'
FB84 F01D   BEQ $FBA3   ZØJA; NAAR BASIC ZØNDER <CLS>
FB86 20BC00 JSR $00BC   ZØNEE; NAAR <GETCHAR>
FB89 F004   BEQ $FB8F   INDIEN 0 NAAR EDIT ZØNDER REGELNR.
FB8B 90F9   BCC $FB86   GEEN CARRYBIT?VØLGEND KARAKTER HALEN
FB8D B014   BCS $FBA3   ALTIJD INPUTBUFFER LEGEN ZØNDER <CLS>
FB8F A900   LDA #$00    \ EDIT ZØNDER EEN
FB91 8D2702 STA $0227   / REGELNUMMER
FB94 A9FB   LDA #$FB    \ LAAT DE VØØRVERWERKER
FB96 A0B5   LDY #$B5     ) WIJZEN NAAR DE RØUTINE DIE
FB98 8505   STA $05      ) BASIC IN DE CØMMAND-MØDE
FB9A 8404   STY $04     / ZET
FB9C A999   LDA #$99    ZET 'LIST'TØKEN AAN HET
FB9E 8513   STA $13     BEGIN V/D INPUT-BUFFER
FBA0 201BFA JSR $FA1B   ) NAAR <CLS>
FBA3 A000   LDY #$00    \ ZET IN HET VARIABELE
FBA5 A212   LDX #$12     ) ADRES HET STARTADRES
FBA7 86C3   STX $C3      ) VAN DE INPUTBUFFER
FBA9 84C4   STY $C4     /
```

```
FBAB 4CF6A5 JMP $A5F6   GA DE INPUTBUFFER LEGEN
FBAE A9BA   LDA #$BA    'READY' ZET READY ØP HET SCHERM
FBB0 A0FB   LDY #$FB
FBB2 20C3A8 JSR $A8C3   VIA <MSGPRNT> V/D BASIC
FBB5 A9FF   LDA #$FF    \ ZET DE BASIC IN DE
FBB7 8588   STA $88     / COMMAND-MØDE
FBB9 60     RTS
FBBA 0D 0A 72 65 61 64  <CR><LF>'READ
FBC0 79 0D 0A 07 07 00  Y'<CR><LF><BELL><BELL> BREAK
FBC6 00     BRK
FBC7 A9AE   LDA #$AE    \ ZET 'READY'PRINTMELDER
FBC9 8504   STA $04      ) IN DE WØØRVERWERKER
FBCB A9FB   LDA #$FB     )
FBCD 8505   STA $05      )
FBCF 60     RTS          /
FBD0 AC0302 LDY $0203   'BELL',HAAL BASICLØADVLAG ØP
FBD3 F001   BEQ $FBD6   MITS KEYBØARD VERDER SPRINGEN
FBD5 60     RTS         ANDERS KLAAR
FBD6 A030   LDY #$30    ZET TØØN-TELLERS KLAAR
FBD8 A2C0   LDX #$C0
FBDA 20F0FB JSR $FBF0   KIES EERSTE PIEPRØUTINE
FBDD CA     DEX
FBDE D0FD   BNE $FBDD
FBE0 A2C0   LDX #$C0
FBE2 20F3FB JSR $FBF3   KIES TWEEDE PIEPRØUTINE
FBE5 CA     DEX
FBE6 D0FD   BNE $FBE5
FBE8 88     DEY
FBE9 D0ED   BNE $FBD8
FBEB 98     TYA         ZET DE 0 V/D Y IN DE ACCU
FBEC 60     RTS         IN A,X EN Y STAAT EEN 0
FBED A953   LDA #$53    'RESAC',RESET ACIA SUPERB.
FBEF 2CA911 BIT $11A9
FBF2 2CA951 BIT $51A9
     FBF0   LDA #$11    EERSTE PIEPRØUTINE
     FBF2   BIT $51A9
            FBF3   LDA #$51   TWEEDE PIEPRØUTINE
FBF5 8D00F0 STA $F000   IN CØNTRØLEREGISTER ACIA
FBF8 60     RTS
FBF9 A000   LDY #$00
FBFB 4CA6FC JMP $FCA6   ) GA NAAR <TESTSP>
FBFE A000   LDY #$00    'DISCB' DISKBØØTTRAP (VERSIE 4.0)
FC00 8C01C0 STY $C001   INITIALISEER DISC-PIA
FC03 8C00C0 STY $C000
FC06 A204   LDX #$04
FC08 8E01C0 STX $C001
FC0B 8C03C0 STY $C003
FC0E 88     DEY
FC0F 8C02C0 STY $C002
FC12 8E03C0 STX $C003
FC15 8C02C0 STY $C002
FC18 A9FB   LDA #$FB
FC1A D009   BNE $FC25   SPRING ALTIJD
FC1C A902   LDA #$02
FC1E 2C00C0 BIT $C000
FC21 F01C   BEQ $FC3F
```

```
FC23 A9FF    LDA #$FF
FC25 8D02C0  STA $C002
FC28 2099FC  JSR $FC99    ) SPRING NAAR RTS
FC2B 29F7    AND #$F7
FC2D 8D02C0  STA $C002
FC30 2099FC  JSR $FC99
FC33 0908    ORA #$08
FC35 8D02C0  STA $C002
FC38 A218    LDX #$18
FC3A 2085FC  JSR $FC85    ) NAAR VERTRAAGRØUTINE
FC3D F0DD    BEQ $FC1C    SPRING ALTIJD
FC3F A27F    LDX #$7F
FC41 8E02C0  STX $C002
FC44 2085FC  JSR $FC85    ) NAAR VERTRAAGRØUTINE
FC47 AD00C0  LDA $C000
FC4A 30FB    BMI $FC47
FC4C AD00C0  LDA $C000
FC4F 10FB    BPL $FC4F
FC51 A903    LDA #$03     ) MASTER-RESET DISK-ACIA
FC53 8D10C0  STA $C010
FC56 A958    LDA #$58
FC58 8D10C0  STA $C010
FC5B 2090FC  JSR $FC90    ) NAAR <LEESDISK>
FC5E 85FE    STA $FE      HIGHBYTE VAN STARTADRES ØPSLAAN
FC60 AA      TAX          BEWAAR ØØK IN X
FC61 2090FC  JSR $FC90    ) NAAR <LEESDISK>
FC64 85FD    STA $FD      LØWBYTE VAN STARTADRES ØPSLAAN
FC66 2090FC  JSR $FC90    ) NAAR <LEESDISK>
FC69 85FF    STA $FF      AANTAL TE LEZEN PAGINA'S
FC6B A000    LDY #$00
FC6D 2090FC  JSR $FC90    ) NAAR <LEESDISK>
FC70 91FC    STA ($FD),Y  VUL GEHEUGEN V.A. STARTADRES
FC72 C8      INY
FC73 D0F8    BNE $FC6D
FC75 E6FE    INC $FE      VEHØØG PAGENUMMER
FC77 C6FF    DEC $FF      VERLAAG AANTAL NØG TE LEZEN PAGES
FC79 D0F2    BNE $FC6D    ALLE PAGES GELEZEN ?
FC7B 86FE    STX $FE      ZØJA; STARTADRES HERSTELLEN
FC7D A9FF    LDA #$FF
FC7F 8D02C0  STA $C002
FC82 6CFD00  JMP ($00FD)  EN ØVERGESEINDE PRØGR.UITVØEREN
FC85 A0F8    LDY #$F8     'VERTRAAGRØUTINE'
FC87 88      DEY
FC88 D0FD    BNE $FC87
FC8A 55FF    EØR $FF,X
FC8C CA      DEX
FC8D D0F6    BNE $FC85
FC8F 60      RTS          X=0, Y=0
FC90 AD10C0  LDA $C010    'LEESDISK' BEKIJK ACIA KØNTR.REG.
FC93 4A      LSR A
FC94 90FA    BCC $FC90    ZØLANG GEEN TEKEN, TERUGSPRINGEN
FC96 AD11C0  LDA $C011    ANDERS TEKEN LEZEN (IN ACC ZETTEN)
FC99 60      RTS
FC9A EA      NØP
FC9B 206CA8  JSR $A86C    NAAR<EINDE V/D REGEL>(CRLF NULLS)
FC9E 205389  JSR $8953    PRINT 'IN'+HUIDIG REGELNUMMER
```

```
FCA1 209BFF JSR $FF9B   NAAR <CTRL>C ROUTINE
FCA4 EA     NOP
FCA5 EA     NOP
FCA6 A9FD   LDA #$FD    'TESTSP' TEST SPATIEBALK INTOETSING
FCA8 8D00DF STA $DF00
FCAB A910   LDA #$10
FCAD 2C00DF BIT $DF00
FCB0 60     RTS         7=1 DAN INGEDRUKT
FCB1 48     PHA         'ACIA-UIT'
FCB2 AD00F0 LDA $F000   BEKIJK ACIA KONTROLE-REGISTER
FCB5 4A     LSR A
FCB6 4A     LSR A
FCB7 90F9   BCC $FCB2   ZOLANG GEEN TIJD, TERUGSPRINGEN
FCB9 68     PLA         ANDERS KARAKTER TERUGHALEN
FCBA 8D01F0 STA $F001   IN DATA-REGISTER ZETTEN
FCBD 60     RTS
FCBE 49FF   EOR #$FF    ZET GEKOZEN REGEL OP 1 (INVERTEREN)
FCC0 8D00DF STA $DF00   SELEKTEER TOETSENBORD REGEL
FCC3 49FF   EOR #$FF    HERSTEL ACCUMULATOR
FCC5 60     RTS
FCC6 48     PHA
FCC7 20CFFC JSR $FCCF   HAAL DE TOETSENBORDKOLOM OP
FCCA AA     TAX         ZET TELLER IN X (= KOLOM)
FCCB 68     PLA         HAAL REGELTELLER OP
FCCC CA     DEX
FCCD E8     INX         LAAT STATUSREGISTER NAAR X KIJKEN
FCCE 60     RTS
FCCF AD00DF LDA $DF00   HAAL KOLOM VAN TOETSENBORD OP
FCD2 49FF   EOR #$FF    HAAL DE 1 ERUIT
FCD4 60     RTS
FCD5 AD0302 LDA $0203   HAAL DE LOAD-VLAG OP
FCD8 3003   BMI $FCDD   ALS BIT 7=1 WEGSPRINGEN
FCDA 1005   BPL $FCE1   ANDERS VERDER NAAR <CR/LF>
FCDC 60     RTS
FCDD C9FF   CMP #$FF    STAAT ER FF IN DE LOAD-VLAG ?
FCDF D0FB   BNE $FCDC   ZONIET; KLAAR, ANDERS
FCE1 20BCF9 JSR $F9BC   1 NAAR <CR>
FCE4 4CD5F8 JMP $F5D5   1 NAAR <LF>
FCE7 20C7FB JSR $FBC7   SPRING NAAR VOORVERWERKER-HERSTELLER
FCEA 20BC00 JSR $00BC   HAAL VOLGENDE KARAKTER OP
FCED 4C4DFB JMP $FB4D   TERUG NAAR CALLER.
FCF0 8D00DF STA $DF00
FCF3 2C00DF BIT $DF00
FCF6 60     RTS
FCF7 E000   CPX #$00    'KEYBA';LUIDT DE BELL ALS X=0
FCF9 D005   BNE $FD00   GA ANDERS INGETOETST KARAKTER HALEN
FCFB A907   LDA #$07    'KEYBB';LUIDT DE BELL ALTIJD
FCFD 2000F8 JSR $F800   KONTROLE-KARAKTER VERWERKEN
FD00 8A     TXA         'KEYB'
FD01 48     PHA         BEWAAR X
FD02 98     TYA         EN Y
FD03 48     PHA         OP DE STACK
FD04 A901   LDA #$01    GA REGEL VOOR REGEL TOETSENB.SCANNEN
FD06 20BEFC JSR $FCBE   SELEKTEER DE REGEL V/H TOETSENBORD
FD09 20C6FC JSR $FCC6   KIJK OF EEN KOLOM REAGEERT
```

```
FD0C D012    BNE $FD20    ZØJA; VERDERSPRINGEN
FD0E 0A      ASL A        ZØNEE; NEEM VØLGENDE REGEL
FD0F D0F5    BNE FD06     SCAN DE KØLØMMEN ØPNIEUW
FD11 A900    LDA #$00     DE HELE MATRIX ZØNER RESULTAAT GESCAND
FD13 8D1602  STA $0216    GEEF AAN DAT SCANNING ZØNDER
FD16 8D1502  STA $0215    RESULTAAT WAS
FD19 A902    LDA #$02     AANTAL KEREN DAT HETZELFDE
FD1B 8D1402  STA $0214    KARAKTER MØET WØRDEN HERKEND
FD1E D0E4    BNE $FD04    BEGIN ALTIJD ØPNIEUW MET SCANNEN
FD20 4A      LSR A        ER IS EEN TØETSINDRUK GEVØNDEN !!
FD21 9009    BCC $FD2C    ZØEK DE KØLØM ØP
FD23 2A      RØL A
FD24 E021    CPX #$21
FD26 D0E6    BNE $FD0E    ZØNIET, VØLGENDE REGEL
FD28 A918    LDA #$1B     -ESC- ?
FD2A D021    BNE $FD4D    WAS -ESC- EERDER INGEDRUKT ?
FD2C 20C6FD  JSR $FDC6    HAAL EERSTE 1 IN DE ACCUMULATØR
FD2F 98      TYA          NAAR DE ACCU
FD30 8D1302  STA $0213    BEWAAR PLAATS VAN TØETSINDRUK
FD33 0A      ASL A        \ VERMENIGVULDIG
FD34 0A      ASL A         ) ACCUMULATØR
FD35 0A      ASL A        / MET 8
FD36 38      SEC
FD37 ED1302  SBC $0213
FD3A 8D1302  STA $0213    RESULTAAT TØETSINDRUK IN ASCII
FD3D 8A      TXA
FD3E 4A      LSR A
FD3F 20C6FD  JSR $FDC6    GA VØLGENDE TØETSINDRUK ØPHALEN
FD42 D0CD    BNE $FD11    BEGIN ANDERS MET SCHØNE LEI.
FD44 18      CLC
FD45 98      TYA
FD46 6D1302  ADC $0213
FD49 A8      TAY
FD4A B9CDFD  LDA $FDCD,Y HAAL GEVØNDEN LETTER ØP IN TABEL
FD4D CD1502  CMP $0215    STØND DIE AL EERDER GEREGISTREERD ?
FD50 D0C4    BNE $FD16    ZØNIET ØPNIEUW SCANNEN
FD52 CE1402  DEC $0214    ZØJA;AANTAL SCANRØNDES VERLAGEN
FD55 F00C    BEQ $FD63    VØLDØENDE GESCAND ? VERDERGAAN
FD57 A005    LDY #$05     ANDERS WACHTEN
FD59 A2C8    LDX #$C8
FD5B CA      DEX
FD5C D0FD    BNE $FD5B
FD5E 88      DEY
FD5F D0F8    BNE $FD59
FD61 F0A1    BEQ $FD04    ALTIJD WEER ØPNIEUW TØETSENBØRD SCANNEN
FD63 A296    LDX #$96     ER IS EEN LETTER GEVØNDEN
FD65 CD1602  CMP $0216    STØND ØP 0216 $96 ?
FD68 D002    BNE $FD6C    ZØNIET ØP 0214 ZETTEN
FD6A A214    LDX #$14     ANDERS $14 WEGZETTEN
FD6C 8E1402  STX $0214    ØP 0214
FD6F 8D1602  STA $0216    GEVØNDEN LETTER ØP 0216 BEWAREN
FD72 A901    LDA #$01     BEGIN MET REGEL 1
FD74 20BEFC  JSR $FCBE    ZET REGEL ØP TØETSENBØRD-SELEKTIE
FD77 20CFFC  JSR $FCCF    HAAL KØLØMNUMMER IN DE ACCU
FD7A AA      TAX          ZET DIE IN X
FD7B A0A0    LDY #$A0
```

```
FD7D 4A      LSR A
FD7E C920    CMP #$20
FD80 F029    BEQ $FDAB
FD82 A000    LDY #$00
FD84 C910    CMP #$10
FD86 F023    BEQ $FDAB
FD88 8A      TXA
FD89 4A      LSR A
FD8A 900B    BCC $FD97
FD8C 29FF    AND #$FF
FD8E D019    BNE $FDA9
FD90 AD1502 LDA $0215
FD93 3016    BMI $FDAB
FD95 100B    BPL $FDA2
FD97 AE1502 LDX $0215
FD9A E051    CPX #$51
FD9C 90EE    BCC $FD8C
FD9E 29FF    AND #$FF
FDA0 F009    BEQ $FDAB
FDA2 A0E0    LDY #$E0
FDA4 AD1502 LDA $0215
FDA7 1002    BPL $FDAB
FDA9 A0F0    LDY #$F0
FDAB AD1502 LDA $0215
FDAE C921    CMP #$21
FDB0 9009    BCC $FDBB
FDB2 8C1302 STY $0213
FDB5 18      CLC
FDB6 6D1302 ADC $0213
FDB9 297F    AND #$7F     ZET MSB ØP 0
FDBB 8D1302 STA $0213     ZET GEVØNDEN TØETS ØP 0213
FDBE 68      PLA          HERSTEL ØUDE REGISTERS
FDBF A8      TAY          EERST Y
FDC0 68      PLA
FDC1 AA      TAX          DAN X
FDC2 AD1302 LDA $0213     ZET DE LETTER IN DE ACCUMULATØR
FDC5 60      RTS
FDC6 A008    LDY #$08     SCHUIF ACCU MAX.8 PLAATSEN NAAR LINKS
FDC8 88      DEY
FDC9 0A      ASL A
FDCA 90FC    BCC $FDC8    ZØLANG GEEN 1 IN DE ACCU; ØPNIEUW
FDCC 60      RTS          ANDERS IN Y DE PLAATS ØF CARRYCLEAR

       F  0  1  2  3  4  5  6
FDCF  4F 20 7A 61 71 4C 6D 6E   Ø ZAQLMN
FDD7  62 76 63 78 6B 6A 68 67   BVCXKJHG
FDDF  66 64 73 69 75 79 74 72   FDSIUYTR
FDE7  76 77 00 00 0D 0A 6F 6C   EW    ØL
FDEF  4E 00 DF 4D BA 50 B9 B8   N  M:P98
FDF7  B7 B6 B5 B4 B3 B2 B1 EA   7654321J
FDFF  EA                        J
```

```
FE00 A228   LDX #$28     'MONIT'
FE02 9A     TXS
FE03 D8     CLD
FE04 D006   BNE $FE0C    SPRING ALTIJD
FE06 A2D8   LDX #$D8     'CLRSC',BOVENSTE VIDEOPAGE
FE08 A9D0   LDA #$D0           ,ONDERSTE VIDEOPAGE
FE0A D021   BNE $FE2D    SPRING ALTIJD
FE0C 2006FE JSR $FE06    DOE EEN SCREEN CLEAR
FE0F F032   BEQ $FE43    SPRING ALTIJD,LAATSTE Y WAS 0
FE11 20E9FE JSR $FEE9    'ADRESMODE',HAAL INPUT(TOETS/CASS.)
FE14 C92F   CMP #$2F     WAS DAT EEN '/' ?
FE16 F037   BEQ $FE4F    JA; NAAR <DATAMODE>
FE18 C947   CMP #$47     WAS DAT EEN 'G' ?
FE1A F030   BEQ $FE4C    JA; NAAR <MONITORSTART>
FE1C C94C   CMP #$4C     WAS DAT EEN 'L' ?
FE1E F05C   BEQ $FE7C    JA; NAAR <MONITORLOAD>
FE20 2093FE JSR $FE93    NAAR <HEX-ASCII>
FE23 30EC   BMI $FE11    BIJ 'N ONGELDIG KAR. NAAR <ADRESMODE>
FE25 A202   LDX #$02     \ VERVISSEL NIBBLES H/D
FE27 20DAFE JSR $FEDA    / ADRESSEN
FE2A 4C43FE JMP $FE43    NAAR WARME MONITORSTART
FE2D 85FF   STA $FF      ADRES HAAR MONITOR VERKT (HIBYTE)
FE2F A900   LDA #$00
FE31 85FE   STA $FE      LAAG BYTE
FE33 85FB   STA $FB      KONTROLE MONITORLOAD(0=KEYB;1-FF=REST)
FE35 A8     TAY
FE36 A920   LDA #$20     \ VUL
FE38 91FE   STA ($FE),Y   ) EEN PAGE
FE3A C8     INY           ) MET
FE3B D0FB   BNE $FE38    / SPATIES
FE3D E6FF   INC $FF      NEEM VOLGENDE PAGINA
FE3F 4C3CFE JMP $FE3C    KONTROLEER OF HOOGSTE PAGE
FE42 EA     NOP          PL'S 1 IS BEREIKT
FE43 B1FE   LDA ($FE),Y HAAL INHOUD VAN MONIT.ADRES
FE45 85FC   STA $FC      ZET OP MONITOR-ADRES-INHOUD
FE47 20ACFE JSR $FEAC    ZET MONIT.ADRES+INHOUD OP SCHERM
FE4A D0C5   BNE $FE11    SPRING ALTIJD NAAR <DATAMODE>
FE4C 6CFE00 JMP ($00FE) 'MONITORSTART'
FE4F 20E9FE JSR $FEE9    'DATAMODE',NAAR HAAL KARAKT.BINNEN
FE52 C92E   CMP #$2E     WAS DAT EEN PUNT '.' ?
FE54 F0BB   BEQ $FE11    JA:NAAR <ADRESMODE>
FE56 C90D   CMP #$0D     WAS DAT EEN CARRIAGE-RETURN ?
FE58 D00F   BNE $FE69    JNEE;NAAR <MONITORINPUT>
FE5A E6FE   INC $FE      JA;MONITORADRES VERHOGEN LAAGBYTE
FE5C D002   BNE $FE60    PAGE-GRENS OVERSCHREDEN ?
FE5E E6FF   INC $FF      JA; OOK HOOG-BYTE OPHOGEN
FE60 A000   LDY #$00
FE62 B1FE   LDA ($FE),Y HAAL INHOUD V/H MONITORADRES OP
FE64 85FC   STA $FC      ZET DAT NAAR INHOUDMONITORADRES
FE66 4C77FE JMP $FE77    TEGELIJKERTIJD OP HET SCHERM
FE69 2093FE JSR $FE93    NAAR <HEX-ASCII>
FE6C 30E1   BMI $FE4F    'MONIT.INPUT';SPRING BIJ ONGELDIG
FE6E A200   LDX #$00                 T TEKEN (BIT 7=1)
FE70 20DAFE JSR $FEDA    VERVISSEL NIBBLES
FE73 A5FC   LDA $FC      HAAL OUDE INHOUD OP
FE75 91FE   STA ($FE),Y ZET DAT IN VERKELIJKE ADRES
FE77 20ACFE JSR $FEAC    NAAR <ADRES+INHOUD OP VIDEO>
```

```
FE7A D0D3   BNE $FE4F   SPRING ALTIJD NAAR <DATAMØDE>
FE7C 85F8   STA $F8     'MØNIT.LØAD KØNTRØLE'
FE7E F0CF   BEQ $FE4F   INDIEN 0(=KEYB.) NAAR <DATAMØDE>
FE80 AD00F0 LDA $F000   'ACIA-IN';HAAL KARAKT. VAN ACIA
FE83 4A     LSR A
FE84 90FA   BCC $FE80   ?ØLANG KARAKT.NIET BINNEN TERUG
FE86 AD01F0 LDA $F001   ?ØJA; INHØUD ACIA ØPHALEN
FE89 297F   AND #$7F    BIT 7 ØP 0 ZETTEN
FE8B 60     RTS
FE8C E4FF   CPX $FF     HØØGSTE VIDEØPAGE+1 BEREIKT ?
FE8E D0A8   BNE $FE38   ?ØNEE;VØLGENDE PAGINA VULLEN
FE90 84FF   STY $FF     ?ØJA;ZET 'N 0 IN MØNITØR-ADRES
FE92 60     RTS
FE93 C930   CMP #$30    'HEX-ASCII';WAS DAT EEN '0' ?
FE95 3012   BMI $FEA9   <$30 ?;NIET UITVØERBAAR;SPRINGEN
FE97 C93A   CMP #$3A    LAGER ØF GELIJK AAN '9' ?
FE99 300B   BMI $FEA6   ?ØJA; NAAR BIJNA KLAAR
FE9B C941   CMP #$41    \  LAGER ALS 'A'
FE9D 300A   BMI $FEA9    ) ØF HØGER ALS
FE9F C947   CMP #$47     ) 'F' ?
FEA1 1006   BPL $FEA9   /  ?ØJA; WEGSPRINGEN
FEA3 38     SEC
FEA4 E907   SBC #$07    BEREKEN ASCII LETTER
FEA6 290F   AND #$0F    BIJNA KLAAR; EN HET GETAL
FEA8 60     RTS
FEA9 A980   LDA #$80    ZET BIT 7 ØP 1;FØUTUITGANG
FEAB 60     RTS
FEAC A203   LDX #$03    'ADRES+INHØUD ØP VIDEØ'
FEAE A000   LDY #$00
FEB0 B5FC   LDA $FC,X
FEB2 4A     LSR A
FEB3 4A     LSR A
FEB4 4A     LSR A
FEB5 4A     LSR A
FEB6 20CAFE JSR $FECA   NAAR <HEX-ASCII> LINKER NIBBLE
FEB9 B5FC   LDA $FC,X
FEBB 20CAFE JSR $FECA   NAAR <HEX-ASCII> RECHTER NIBBLE
FEBE CA     DEX
FEBF 10EF   BPL $FEB0
FEC1 A920   LDA #$20    \ ZET 2 SPATIES
FEC3 8D53D1 STA $D153    ) TUSSEN ADRES
FEC6 8D54D1 STA $D154   / EN INHØUD
FEC9 60     RTS
FECA 290F   AND #$0F    'HEX-ASCII' MET WEERGAVE ØP SCHERM
FECC 0930   ØRA #$30
FECE C93A   CMP #$3A
FED0 3003   BMI $FED5
FED2 18     CLC
FED3 6907   ADC #$07
FED5 994FD1 STA $D14F,Y
FED8 C8     INY
FED9 60     RTS
FEDA A004   LDY #$04    'WERUISSEL NIBBLES'
FEDC 0A     ASL A       \  BRENG MINST SIGNIFIKANTE
FEDD 0A     ASL A        ) NIBBLE NAAR LINKS
FEDE 0A     ASL A        ) RECHTER NIBBLE WØRDT 0
FEDF 0A     ASL A       /
```

```
FEE0 2A      ROL A       \  BRENG LINKER NIBBLE
FEE1 36FC    ROL $FC,X    ) NAAR RECHTS
FEE3 36FD    ROL $FD,X    ) RECHTER NIBBLE
FEE5 88      DEY          ) VERSCHUIFT NAAR
FEE6 D0F8    BNE $FEE0    ) LINKS
FEE8 60      RTS         /
FEE9 A5FB    LDA $FB      'HAAL KARAKTER BINNEN',KEYBOARD?
FEEB D093    BNE $FE80    BIJ GEEN KEYB.NAAR LOAD-ROUTINE
FEED 4C00FD  JMP $FD00    VEL KEYB.;NAAR <KEYB>
FEF0 BAFF    ADRES $FFBA 'IN'    NAAR $0218
FEF2 69FF    ADRES $FF69 'OUTCH' NAAR $021A
FEF4 9BFF    ADRES $FF9B 'CNTRL' NAAR $021C
FEF6 8BFF    ADRES $FF8B 'LOAD'  NAAR $021E
FEF8 96FF    ADRES $FF96 'SAVE'  NAAR $0220
FEFA 09      =ROWS=$0222 VIDEOPARAMETERS 2K VIDEO
FEFB 39      =ROWE=$0223 WOOR CIP MET 48 KARAKTERS
FEFC 04      =LINES=$0224
FEFD 1D      =LINEE=$0225
FEFE FF      =VIDTYP=$0226, CIP 2K VIDEO
FEFF FF      =EDITFL=#0227, EDITVLAG,GEEN EDITMODE


FF00 D8      CLD          'RESET',SPRONGADRES NA -BREAK-
FF01 A228    LDX #$28
FF03 9A      TXS
FF04 20EDFB  JSR $FBED    ) NAAR <RESAC>,MASTERRESET ACIA
FF07 A000    LDY #$00     \  VUL PAGE 2 MET NULLEN
FF09 98      TYA           ) (VAN $0200 TOT EN
FF0A 9D0002  STA $0200,X   )  MET $0228)
FF0D CA      DEX           )
FF0E D0FA    BNE $FF0A    /
FF10 2047FF  JSR $FF47    NAAR <SET 24>,Y-UIT =0
FF13 B99FF9  LDA $F99F,Y \ ZET 'DOWN' OP HET SCHERM
FF16 F006    BEQ $FF1E    )
FF18 2000F8  JSR $F800    )
FF1B C8      INY          )
FF1C D0F5    BNE $FF13    /
FF1E 20BAFF  JSR $FFBA    NAAR <IN>, ALGEMENE INPUTROUTINE
FF21 C94D    CMP #$4D     IS EEN 'M' INGETOETST ?
FF23 D003    BNE $FF28    ZONEE;VERDERSPRINGEN
FF25 4C00FE  JMP $FE00    ZOJA; NAAR MONITOR <MONIT>
FF28 C957    CMP #$57     IS EEN 'W' INGETOETST ?
FF2A D006    BNE $FF32    ZONEE;VERDERSPRINGEN
FF2C 8D2802  STA $0228    ZOJA;ZET 'W' OP =BASFL=
FF2F 4C0000  JMP $0000    NAAR WARME START W/D BASIC
FF32 C943    CMP #$43     IS ER EEN 'C' INGEDRUKT ?
FF34 D00A    BNE $FF40    ZONEE;VERDERSPRINGEN
FF36 8D2802  STA $0228    ZOJA;ZET 'C' OP =BASFL=
FF39 A000    LDY #$00     HUIDIG REGELNR. WORDT 0
FF3B 8A88    STY $88      ZET IN HOOG-BYTE VAN HUID.REGELNR.
FF3D 4C22BD  JMP $BD22    NAAR KOUDE START BASIC,ZE INIT.
FF40 C944    CMP #$44     IS ER EEN 'D' INGETOETST ?
FF42 D0BC    BNE $FF00    ZONEE;GEEN GELD. TOETS;NAAR <RESET>
FF44 4CFEFB  JMP $FBFE    ZOJA; NAAR <DISCB>,DE DISCBOOTSTRAP
FF47 B9F0FE  LDA $FEF0,Y 'SET 24';VUL PAGINA 2 MET
FF4A 991802  STA $0218,Y  VEKTOREN EN VIDEO-
FF4D C8      INY          PARAMETERS.
```

```
FF4E C010    CPY #$10     ------
FF50 D0F5    BNE $FF47    'SET 48'
FF52 2006FE JSR $FE06    NAAR <SCHERM SCHØØN>
FF55 4C96F9 JMP $F996    NAAR <HØME>
FF58 B965FF LDA $FF65,Y 'SCHERM 48*12' VØØR SERIE 2 S.B.
FF5B 990202 STA $0202,Y VUL 4 PLAATSEN IN PAGE 2
FF5E C8      INY
FF5F C005    CPY #$05     KLAAR ?
FF61 D0F5    BNE $FF58    ZØNIET; TERUG
FF63 F0ED    BEQ $FF52    ANDERS ALTIJD NAAR <CLS>+<HØME>
FF65 0B      =RØWS=$0222; PARAMETERS VØØR 48*12 SCHERM
FF66 3A      =RØWE=$0223
FF67 02      =LINES=$0224
FF68 0D      =LINEE=$0225
FF69 2000F8 JSR $F800    'ØUTCH';ALGEMENE ØUTPUTRØUTINE
FF6C 48      PHA         ZET KARAKTER ØP SCHERM EN BEWAREN
FF6D AD0502 LDA $0205    KØNTRØLEER =BASIC-SAVE-VLAG=
FF70 F017    BEQ $FF89   VERDERSPRINGEN ALS VLAG NIET GEZET
FF72 68      PLA         ANDERSKAPAKTER TERUGHALEN
FF73 20B1FC JSR $FCB1    EN NAAR ACIA STUREN MET <ACIA-UIT>
FF76 C90D    CMP #$0D    WAS DAT EEN CARRIAGE-RETURN ?
FF78 D010    BNE $FF8A   ZØNEE; KLAAR (NAAR EEN RTS)
FF7A 48      PHA         'NULL'; ZØJA-
FF7B 8A      TXA         BEWAAR A EN X REGISTERS
FF7C 48      PHA
FF7D A20A    LDX #$0A    \  EN SCHRIJF 10 MAAL
FF7F A900    LDA #$00     ) EEN NULL (=$00)
FF81 20B1FC JSR $FCB1    ) NAAR DE ACIA MET <ACIA-UIT>
FF84 CA      DEX         )
FF85 D0FA    BNE $FF81   /
FF87 68      PLA         EN HERSTEL
FF88 AA      TAX         X-REGISTER EN
FF89 68      PLA         ACCUMULATØR
FF8A 60      RTS
FF8B 48      PHA         'LØAD' BEWAAR ACCU
FF8C CE0302 DEC $0203    ZET =BASIC-LØAD-VLAG= AAN (<>$00)
FF8F A900    LDA #$00    ZET DE
FF91 8D0502 STA $0205    =BASIC-SAVE-VLAG= UIT (=$00)
FF94 68      PLA         HAAL ACCU TERUG
FF95 60      RTS
FF96 48      PHA         'SAVE';BEWAAR ACCU
FF97 A901    LDA #$01    ZET =BASIC-SAVE-VLAG- AAN (<>$00)
FF99 D0F6    BNE $FF91   SPRING ALTIJD
FF9B AD1202 LDA $0212    'CNTRL';HAAL =CNTRL-VLAG= ØP
FF9E D0F5    BNE $FF95   ALS DIE UIT STAAT BEN JE KLAAR
FFA0 20A6FC JSR $FCA6    AAN ? DAN NAAR <TESTSP>
FFA3 F0FB    BEQ $FFA0   SPRING TERUG BY SPATIEBALK INGEDRUKT
FFA5 20E3FF JSR $FFE3    ZET $FE ØP TØETSENBØRD
FFA8 70EB    BVS $FF95   GEEN REAKTIE ?;KLAAR (NAAR RTS)
FFAA 20E6FF JSR $FFE6    ZET $FB ØP TØETSENBØRD
FFAD 70E6    BVS $FF95   GEEN REAKTIE ?;KLAAR (NAAR RTS)
FFAF 38      SEC         ER WAS WEL EEN REAKTIE
FFB0 4C3DA6 JMP $A63D    GA NAAR <END> RØUTINE
FFB3 20F7FC JSR $FCF7    NAAR <KEYBA>
FFB6 4CBBFA JMP $FABB    NAAR KØNTRØLE DIRECT-MØDE/EDIT-MØDE
FFB9 EA      NØP
```

```
FFBA 2C0302 BIT $0203   'IN';ALGEMENE INPUT RØUTINE
                          HAAL =BASIC-LØAD-VLAG= ØP
FFBD 1018   BPL $FFD7   KEYB.STAAT AAN;KØNTRØLEER ØP BASIC
FFBF EA     NØP
FFC0 EA     NØP
FFC1 EA     NØP
FFC2 20F9FB JSR $FBF9   NAAR <TESTSP>
FFC5 F00D   BEQ $FFD4   SPATIEBALK INGEDRUKT?;WEGSPRINGEN
FFC7 AD00F0 LDA $F000   'ACIA-IN';KARAKTER BINNEN ?
FFCA 4A     LSR A       KØNTØLEER KØNTRØLE REGISTER
FFCB 90F5   BCC $FFC2   ZØLANG NIETS BINNEN;TERUGSPRINGEN
FFCD EA     NØP
FFCE EA     NØP
FFCF EA     NØP
FFD0 AD01F0 LDA $F001   HAAL HET KARAKTER ØP
FFD3 60     RTS         TERUG NAAR CALLER
FFD4 8C0302 STY $0203   ZET WEG ØP =BASIC-LØAD-VLAG=
FFD7 AD2802 LDA $0228   HAAL =BASFL= ØP
FFDA D0D7   BNE $FFB3   ALS HET BASIC IS SPRINGEN
FFDC 4C00FD JMP $FD00   ANDERS NAAR <KEYB>
FFDF EA     NØP
FFE0 492F01 ??????
FFE3 A9FE   LDA #$FE
FFE5 2CA9FB BIT $FBA9
     FFE5 A9FB LDA #$FB
FFE8 4CF0FC JMP $FCF0   ZET KARAKTER ØP TØETSENBØRD
FFEB 6C1802 JMP ($0218) 'INPUT';SPRING INDIREKT
FFEE 6C1A02 JMP ($021A) 'ØUTPUT';SPRING INDIREKT
FFF1 6C1C02 JMP ($021C) 'CTRL-C';SPRING INDIREKT
FFF4 6C1E02 JMP ($021E) 'LØAD'; SPRING INDIREKT
FFF7 6C2002 JMP ($0220) 'SAVE'; SPRING INDIREKT
FFFA 3001   ADRES $0130 'NMI'-VEKTØR
FFFC 00FF   ADRES $FF00 'RESET'-VEKTØR
FFFE C001   ADRES $01C0 'IRQ'-VEKTØR
```

Jos Burghouts
Montgomerylaan 112
2625 PR Delft

PAGE ZERØ ADRESSEN

```
ADR. INHØUD TØELICHTING
HEX.
 00 4C74A2 SPRING NAAR DE WARME START V/D BASIC $A274
 03 4CAEFB SPRING NAAR DE VØØRVERWERKER (FØUTMELDING ETC.)
           + PØKE 96(DEC.) ØP $03 ØM 'READY' TE ØNDERDRUKKEN
 06 05AE   ADRES VAN INVAR & SUBR ØMZETT. REAL NAAR INTEGER
 08 C1AF   ADRES VAN ØUTVAR & SUBR ØMZETT. INTEGER NAAR REAL
 0A 4C.... SPRING NAAR USR() SUBRØUTINE
 0D **     AANTAL NULL'S NA EEN CARRIAGE RETURN (START=0)
 0E **     KURSØRPLAATS ØP HUIDIGE ØUTPUTREGEL
 0F **     TERMINAL WIDTH=REGELLENGTE ØUTPUTAPPARAAT
           + EEN 0 GEEFT EEN EXTRA CARRIAGE RETURN
 10 **     KØMMA-INDIKATØR EN TERMINALBREEDTE (KØMMAKØLØMMEN)
           + AANTAL KAR=14*(INT(REGELLENGTE/14)-1)
 11 ****   HUIDIGE REGELNUMMER V/D BASIC (ØF ADRES)
 13 TØT 5A INPUT BUFFER VØØR TØETSENBØRD & CASSETTE
 5B **     STØPINDIKATØR BIJ STRINGZØEKEN
           & EVEN/ØNEVEN VLAG BY ØMZETT. INTEGER/REAL
           & TYDELYK ADRES VØØR HUIDIGE REGELNUMMER
           & TYDELYK ADRES BY 'AND' EN 'ØR' RØUTINES
 5C **     TWEEDE STØP-INDIKATØR BY HET STRINGZØEKEN
 5D **     KØMPLEMENT BY 'AND' EN 'ØR' RØUTINES;AND=Ø;OR=FF
           & AANTAL FEITELYKE DIMENSIES BY ARRAY-VERWYZING
 5E **     VERWYZINGSINDIKATØR VØØR 'N ARRAYVERWYZING
 5F **     STRINGINDIKATØR (MSB=1) BY ZØEKEN VAN VARIABELEN
 60 **     DATA-STMNT VLAG VØØR ØMZETT. VAN KØMMANDØ NAAR TØKEN
 61 **     VARIABELE INDIK.;<>0=GEEN INTEGERS & STR.TØEGESTAAN
 62 **     INPUTTYPE INDIK.;BIT 7=0 DAN INPUT/BIT 7=1 DAN READ
 63 **     PARAMETERTYPE BY RELATIØNELE ØPERATØREN (<=>)
 64 **     GEEN ØUTPUT ALS BIT 7=1 IN WERKING MET <CTRL>Ø
 65 **     STRINGBESCHRIJVER-STACK; INDEXWAARDE
 66 **     STRINGBESCHRIJVER-STACK; LAGE BYTE
 67 **     STRINGBESCHRIJVER-STACK; HØGE BYTE
 68 TØT 70 STRINGBESCHRIJVER-STACK
 71 ****   TYDELYK ADRESØPSLAGPLAATS NR.1 (BASIC)
 73 ****   TYDELYK ADRESØPSLAGPLAATS NR.2 (BASIC)
 75 ****** EXTRA REEEL GETAL REGISTER BY '*' & '/' (3 BYTES)
 78 **     NIET IN GEBRUIK BY DE BASIC
 79 0103   BEGINADRES V/H BASICPRØGRAMMA (MEESTAL $0301)
 7B ****   BEGINADRES V/D VARIABELENTABEL=EINDE BASICPRØGRAMMA
 7D ****   BEGINADRES V/D GEDIMENSIØNEERDE VARIAB.TABEL
           + IS EINDE ENKELVØUDIGE VARIABELENTABEL
 7F ****   EINDADRES GEDIM.VARIABELENTABEL=BEGIN VRY GEHEUGEN
 81 ****   ADRES VANWAAR DE STRINGS WØRDEN ØPGESLAGEN
           + VANAF HET EINDE V/D RAM-RUIMTE NAAR VØREN
 83 ****   EINDADRES TØT WAAR SRINGS WØRDEN ØPGESLAGEN
 85 ****   HIMEM=EINDADRES V/H BESCHIKBAAR RAM-GEHEUGEN
 87 ****   HUIDIGE REGELNUMMER (ALS MSB=1 DAN CØMMAND-MØDE)
 89 ****   LAATSTE REGELNUMMER (BY 'END' ØF 'STØP')
 8B ****   ADRESVERWYZING NAAR VØØRLAATSTE REGELNUMMER
 8D ****   HUIDIG REGELNUMMER BY EEN 'READ' ØPDRACHT
 8F ****   ADRES WAAR HET VØLGENDE 'DATA'STMNT TE VINDEN IS
 91 ****   ADRES WAAR HET VØLGENDE ELEMENT IN DE DATA STAAT
           + WØRDT BY EEN 'RESTØRE' ØP 0 GEZET
```

```
93 ****    DE ØMGEZETTE VARIABELE-NAAM (IN ASCII)
95 ****    ADRES V/D HUIDIGE VARIAB. IN DE VARIAB.TABEL
           + WØRDT MET SUBRØUTINE $AD0B ØPGEZØCHT
97 ****    ADRES V/E VARIABELE BY 'INPUT'
99 **      ØPBERGPLAATS VØØR HUIDIGE RELAT.ØPERATØR (INDEX)
9A **      NIET IN GEBRUIK DØØR BASIC
9B **      ØPBERGPLAATS VØØR HUIDIGE RELATIØNELE ØPERATØR
9C TØT 9F  ØPBERGPLAATS VØØR REELE GETALLEN
           & TYDELYKE ADRESVERWYZING 3 EN 4
A0 03      STAPGRØØTTE V/D GARBAGE-CØLLECTØR
A1 4C****  JMP NAAR ADRES IN TE VULLEN IN $A2/A3
A2 **      TYDELYKE ØPBERGPLAATS VØØR GARBAGE-CØLLECTØR
A3 **      BEWAKINGS-BYTE BY REEEL GETAL '+' EN '-'
A4 TØT A7  ØPBERGPLAATS VØØR REELE GETALLEN
A4 ****    EINDADRES V/D BESTEMMING BY GEHEUGENVERPLAATSING
A6 ****    EINDADRES V/D HERKØMST BY GEHEUGENVERPLAATSING
A8 TØT AB  TWEEDE ØPBERGPLAATS VØØR REELE GETALLEN
A8 **      AANTAL CYFERS VØØR DE '.' BY REEEL GETAL NAAR
           + STRING ØMZETT. & AANTAL NA DE '.'
A9 **      EXPØNENTWAARDE BIJ REEEL GETAL NAAR STRING ØM-
           + ZETTING EN TERUG (DE 06 IN 1.23E06)
AA ****    TYDELYKE ADRESVERWYZING NR.5(BV.BY 'LIST')
           & STARTADRES V/D HERKØMST BY GEHEUGENVERPLAATSING
AC TØT B0  BELANGRYKSTE REEEL GETAL ØPBERGPLAATS
AC **      EXPØNENT
AD TØT AF  MANTISSE = EIGENLYKE GETAL ZØNDER PUNT
AD **      GETAL;MEEST SIGNIFIKANTE BYTE;(BIT7=1)
AE **      GETAL
AF **      GETAL;MINST SIGNIFIKANTE BYTE
AE ****    ØPBERGPLAATS VØØR INTEGERWAARDE VAN 'INVAR';
           + (SUBRØUT. ØP $AE05) =15 BITS MET TEKEN
B0 *       TEKEN ØP BIT 7 ('N 1=NEGATIEF GETAL)
B1 **      HULPØPBERGPLAATS BY REEEL GETAL BEWERKINGEN
B2 **      ØPVULBYTE BY RECHTSVERPLAATSING V/H REEEL GETAL REG.
B3 TØT B7  TWEEDE REEEL GETAL REGISTER (ZIE $AC/AF)
B8 **      EXCL.ØR V/D TEKENS REEEL GETAL REG. 1 & 2
B9 **      BEWAKINGSBYTE VØØR REEEL GETAL REGISTER
BA ****    TYDELYKE ADRESVERWYZING 6 (ØMZETT.REAL NAAR STR.)
BC TØT D3  SUBRØUTINE ØM HET VØLGENDE TEKEN ØP TE HALEN
           + WAT ØP $C3/C4 TE VINDEN IS
           + HEET 'CHARGET' EN WØRDT GEKØPIEERD VAN $BCEE/BD05
BF TØT D3  SUBRØUTINE ØM DE ACCU TE TESTEN ØP CYFERS
           + LAAT DE CARRY-VLAG 0 INDIEN 0 T/M 9
C2 TØT D3  SUBRØUTINE ØM DE LAATST BEWERKTE BYTE ØP TE HALEN
C3 ****    ADRES V/D VARIABELE DIE WØRDT BEWERKT
D4 TØT D7  REEEL RESULTAAT V/D LAATSTE RND ØPRØEP
E0 TØT FA  VRYE PAGE ZERØ RUIMTE BEH. $E6 EN SØMS $F0/F1
FB **      KØNTRØLE MØNITØR BY LADEN;0=KEYBØARD;1/FF ANDERS
FC **      INHØUD ADRES WAAR MØNITØR NET WERKT
FD **      KARAKTER DAT WØRDT WEGGESCHREVEN NAAR SCHERM
FE ****    ADRES WAAR MØNITØR(EPRØM) NET BEZIG IS


EINDE ZERØ-PAGE ØVERZICHT
( MET DANK AAN HENK WEVERS )
```

PAGINA TWEE-ADRESSEN
==========

```
ADR. NAAM     TØELICHTING
HEX.
0200 CURDIS   KURSØRPLEK Ø/D REGEL BIJ ØUDE VID.RØUTINE($BF2D)
0201 ØLDCH    HET TEKEN ØNDER DE KURSØR
0202 TEMP1    HULPØPBERGPLAATS VØØR DE VIDEØRØUTINE
0203 LØADFL   BASIC-LØAD-VLAG;0=KEYBØARD;MSB=1 ACIA
0204 TEMP2    TIJDELIJKE ØPBERGPLAATS
0205 SAVEFL   BASIC-SAVE-VLAG;0=SCHERM;01=ØØK ACIA
0206 PRNTTIM  PRINTTIMER;0=NØRMAAL;255=XEER LANGZAAM
0207 - 0211   TIJDELIJKE HULPRØUTINE VØØR HET SCRØLLEN
              0207 LDA $D709;Y
              020A STA $D709;Y
              020D INY
              020E RTS
0212 CNTRLC   <CTRL> C EN DYNAMISCHE HALT-VLAG
              + 0=AAN EN 1=UIT
0213 GETKEYR  RESULTAAT VAN TØETSINDRUK UIT DE GETKEY-
              + RØUTINE ØP $FD00 EN $FEED
0214 - 0217   TIJDELIJKE ØPBERGPLAATS VØØR KEYBØARD-RØUTINE
0218 INPUTR   ADRES MET INPUTRØUTINE (VØØR IP $FFBA)
021A ØUTPUTR  ADRES MET ØUTPUTRØUTINE (VØØR IP $FF69)
021C CRTLR    ADRES MET <CTRL>C RØUTINE (VØØR IP $FF9B)
021E LØADR    ADRES MET LØADRØUTINE (VØØR IP $FF8B)
0220 SAVER    ADRES MET SAVERØUTINE (VØØR IP $FF96)
0222 RØWS     BEGINPLAATS V/D KØLØM IN DE TEKSTWINDØW
0223 RØWE     EINDPLAATS VAN DE KØLØM
0224 LINES    BEGINPLAATS V/D REGEL IN DE TEKS WINDØW
0225 LINEE    LAATSTE REGEL V/D TEKSTWINDØW
0226 VIDTYP   VIDEØTYPE 0=24*25; FF=CIP 2K VIDEØ; ØVERIGE 12*48
0227 EDITFL   EDITVLAG 0=EDIT ZØNDER REGELNR;01 MET REGELNR.
              $FF= GEEN EDITMØDE
0228 BASFL    BASICVLAG 0=GEEN BASIC; ANDERS=BASIC
0229 RØW      PLAATS V/D KURSØR; KØLØMNUMMER
022A LINE     PLAATS V/D KURSØR; REGELNUMMER
022B - 022E   VIDEØ-HULP-SUBRØUTINE
              022B $8D STA-ABSØLUUT
              . ØF $AD LDA-ABSØLUUT
              022C ABSØLUUT VIDEØADRES MET KURSØR (LØW/HIGH)
              WØRDT BEREKEND MET <BERCUR> $F88B
              022E RTS
022F NIET GEBRUIKT; DØØR FØUT WØRDT $02FF GEBRUIKT !!
0230 - 02FE   VRIJ RAM-GEHEUGEN

 EINDE PAGINA-TWEE ØVERZICHT


READY
```

E SOFTWARE

EA Spelen

EB Subroutines

EC Grafisch

ED Zakelijk gebruik

EE Utilities

```
48 PRINT" RICHTEN. DE TEKENS WORDEN MET DE JOYS
TICKS BE-":PRINT
50 PRINT" WOGEN: NOORD=OMHOOG, OOST=RECHTS, ZUI
D=OMLAAG":PRINT
52 PRINT" WEST=LINKS, NOORDOOST=OMHOOG EN NAAR
RECHTS ETC.":PRINT
54 FORG=1TO40000:NEXTG:PRINTCHR$(3)
60 PRINTCHR$(3):PRINT" WIE SPELEN ER? (NAAM,NAA
M)"
65 INPUTA$,B$
70 POKE11,6:POKE12,254:A=USR(A):PRINTCHR$(2)
75 PRINT"           SCORE O ";A$" ";S1
80 PRINT"           SCORE X ";B$" ";S2
85 FORT=53778TO53809:POKET,27:NEXTT
90 FORT=53393TO53809:POKET,27:NEXTT
95 FORT=53809TO53425:POKET,27:NEXTT
100 FORT=53385TO53433:POKET,27:NEXTT
110 FORT=53433TO55289:POKET,27:NEXTT
120 FORT=55289TO55241STEP-1:POKET,27:NEXTT
130 FORT=55241TO53385STEP-64:POKET,27:NEXTT
140 P=53450:L=55224:POKEP,188:POKEL,226
150 POKE530,1:X=0:T=57088:GOTO1110
210 IFPEEK(T)=127THEN500
220 IFPEEK(T)=111THEN510
230 IFPEEK(T)=239THEN520
240 IFPEEK(T)=175THEN530
250 IFPEEK(T)=191THEN540
260 IFPEEK(T)=159THEN550
270 IFPEEK(T)=223THEN560
```

```
280 IFPEEK(T)=95THEN570
290 IFPEEK(T)=255THEN210
500 IFPEEK(R-64)=32THEN600
505 GOTO700
510 IFPEEK(R-63)=32THEN610
515 GOTO700
520 IFPEEK(R+1)=32THEN620
525 GOTO700
530 IFPEEK(R+65)=32THEN630
535 GOTO700
540 IFPEEK(R+64)=32THEN640
545 GOTO700
550 IFPEEK(R+63)=32THEN650
555 GOTO700
560 IFPEEK(R-1)=32THEN660
565 GOTO700
570 IFPEEK(R-65)=32THEN670
575 GOTO700
600 R0=R:R=R-64:GOTO1000
610 R0=R:R=R-63:GOTO1000
620 R0=R:R=R+1:GOTO1000
630 R0=R:R=R+65:GOTO1000
640 R0=R:R=R+64:GOTO1000
650 R0=R:R=R+63:GOTO1000
660 R0=R:R=R-1:GOTO1000
670 R0=R:R=R-65:GOTO1000
```

```
700 IFPEEK(R-64)<>32THENIFPEEK(R-63)<>32THENIFP
EEK(R+1)<>32THEN710
705 GOTO750
710 IFPEEK(R+65)<>32THENIFPEEK(R+64)<>32THENIFP
EEK(R+63)<>32THEN720
715 GOTO750
720 IFPEEK(R-1)<>32THENIFPEEK(R-1)<>32THENIFPEE
K(R-65)<>32THEN730
725 GOTO750
730 IF(INT(X/2))x2<>XTHEN740
735 S1=S1+1:GOTO1200
740 S2=S2+1:GOTO1200
750 IFPEEK(R)=188THEN760
755 FORQ=1TO10:POKER,4:POKER,32:NEXTQ:POKER,226
:GOTO210
760 FORQ=1TO10:POKER,4:POKER,32:NEXTQ:POKER,188
:GOTO210
1000 X=X+1
1010 IF(INT(X/2))x2<>XTHEN1020
1015 L=R:POKER0,187:POKEL,226:PRINTCHR$(7):PRIN
TCHR$(2):GOTO1110
1020 P=R:POKER0,187:POKEP,188:PRINTCHR$(7):PRIN
TCHR$(2):GOTO1100
1100 POKE53705,27:R=L:POKET,253:GOTO210
1110 R=P:POKET,127:GOTO210
1200FORT=1TO25:PRINTCHR$(7):NEXTT:PRINTCHR$(2):
GOTO70
```

LISTING "SOLITAIRE"

```
100  CLEAR : GOSUB 1140
110  PRINT TAB(14); "SOLITAIRE"
120  A = 111
130  PRINT : PRINT : PRINT
140  PRINT "spelregels nodig ?";: GOSUB 1070
150  GOSUB 1140
160  PRINT : IF I = Ø THEN 360
170  PRINT "De spelregels zijn bijna"
180  PRINT "hetzelfde als bij dammen."
190  PRINT "U mag slaan, mits er"
200  PRINT "achter het ruitje dat U"
210  PRINT "wilt slaan een leeg"
220  PRINT "veldje bevindt."
230  PRINT
240  PRINT"Doordat de geslaagde"
250  PRINT "ruitjes van het veld"
260  PRINT "verdwijnen, blijven er op"
270  PRINT "het laatst weinig"
280  PRINT "ruitjes over."
290  PRINT
300  PRINT "Het is de kunst om een"
310  PRINT "minimum aantal ruitjes"
320  PRINT "over te laten."
330  PRINT : PRINT : PRINT
340  PRINT "VERDERGAAN ?"; : GOSUB 1070
350  GOSUB 1140 : PRINT
360  FOR F = 53454 TO 53966 STEP 64
370  FOR G = F TO F + 5 STEP 2
380  POKE G,232 : NEXT G,F
390  FOR F = 53640 TO 53800 STEP 64
400  FOR G = F TO F + 16 STEP 2
410  POKE G,232 : NEXT G,F
420  POKE 53712,A
430  POKE 54090,49 : POKE 54092, 61 : POKE 54094,22
440  POKE 54154,50 : POKE 54156,61 : POKE 54158,18
```

```
450          POKE 54098,51 : POKE 54100,61 : POKE 54102,16
460          POKE 54162,52 : POKE 54164,61 : POKE 54166,20
470          D$ = "dit ruitje verschuiven ?" : GOSUB 1010
480          FOR F = 53454 TO 53582 STEP 64
490          FOR G = F TO F + 5 STEP 2
500          M = PEEK(G) : IF M = A THEN 550
510          POKE G,187
520          FOR T = 1 TO 120 : NEXT T
530          GOSUB 1070 : IF I = 1 THEN 730
540          POKE G,M
550          NEXT G,F
560          FOR F = 53640 TO 53800 STEP 64
570          FOR G = F TO F + 16 STEP 2
580          M = PEEK(G) : IF M = A THEN 630
590          POKE G,187
600          FOR T = 1 TO 120 : NEXT T
610          GOSUB 1070 : IF I = 1 THEN 730
620          POKE G,M
630          NEXT G,F
640          FOR F = 53838 TO 53998 STEP 64
650          FOR G = F TO F + 5 STEP 2
660          M = PEEK(G) : IF M = A THEN 710
670          POKE G,187
680          FOR T = 1 TO 120 : NEXT T
690          GOSUB 1070 : IF I = 1 THEN 730
700          POKE G,M
710          NEXT G,F
720          GOTO 470
730          GOSUB 1000
740          POKE 530,1 : POKE P,127
750          IF PEEK(P) = 127 THEN R = 1 : GOTO 800
760          IF PEEK(P) = 191 THEN R = 2 : GOTO 800
770          IF PEEK(P) = 223 THEN R = 3 : GOTO 800
780          IF PEEK(P) = 239 THEN R = 4 : GOTO 800
790          GOTO 750
800          POKE 530,0 : OK = ø
810          ON R GOTO 820, 840, 860, 880
```

```
820 IF PEEK(G-2) = 232 AND PEEK(G-4) = A THEN OK = 1
830 GOTO 890
840 IF PEEK(G+2) = 232 AND PEEK(G+4) = A THEN OK = 1
850 GOTO 890
860 IF PEEK(G-64) = 232 AND PEEK(G-128) = A THEN OK = 1
870 GOTO 890
880 IF PEEK(G+64) = 232 AND PEEK(G+128) = A THEN OK = 1
890 IF OK = 1 THEN 930
900 D$ = "onmogelijk" : GOSUB 1010
910 FORF = 1 TO 900 : NEXT F : POKE G,M
920 FOR F = 53347 TO 53380 : POKE F,32 : NEXT F : GOTO 470
930 POKE G,A
940 ON R GOTO 950, 960, 970, 980
950 POKE G-2,A : POKE G-4,232 : GOTO 990
960 POKE G+2,A : POKE G+4,232 : GOTO 990
970 POKE G-64,A : POKE G-128,232 : GOTO 990
980 POKE G+64,A : POKE G+128,232
990 FOR F = 53347 TO 53380 : POKE F,32 : NEXT F : GOSUB
    1200 : GOTO 470
1000 D$ = "welke richting ?"
1010 FOR F = 53348 TO 53380 : POKE F,32 : NEXT F
1020 C = (26 - LEN(D$)) / 2 + 53348
1030 FOR F = C TO C + LEN(D$) - 1
1040 POKE F,ASC(MID$(D$, F - C + 1, 1))
1050 NEXT F
1060 RETURN
1070 P = 57088 : POKE 530,1
1080 POKE P,247
1090 IF PEEK(P) = 251 THEN I = 1 : GOTO 1130
1100 POKE P,251
1110 IF PEEK(P) = 247 THEN I = 0 : GOTO 1130
1120 GOTO 1080
1130 POKE 530,0 : RETURN
1140 V = PEEK(129) : V1 = PEEK(130)
1150 POKE 129,0 : POKE 130,212
1160 S$ = " "
1170 FOR T = 1 TO 7 : S$ = S$ + S$ + " " : NEXT T
```

```
1180        POKE 129,V : POKE 130,V1
1190        RETURN
1200        N = ø
1210        FOR F1 = 53448 TO 53960 STEP 64
1220        FOR G1 = F1 TO F1 + 16 STEP 2
1230        K = PEEK(G1)
1240        IF K = A OR K = 32 THEN 1300
1250        N = N + 1
1260        IF PEEK(G1-2) = 232 AND PEEK(G1-4) = A THEN IK = 1
1270        IF PEEK(G1+2) = 232 AND PEEK(G1+4) = A THEN IK = 1
1280        IF PEEK(G1+64) = 232 AND PEEK(G1+128) = A THEN IK = 1
1290        IF PEEK(G1-64) = 232 AND PEEK(G1-128) = A THEN IK = 1
1300        NEXT G1,F1
1310        IF IK = 1 THEN IK = 0 : RETURN
1320        FOR F = 54019 TO 54171 : POKE F,32 : NEXT F
1330        PRINT"U heeft nog";N;"rondjes"
1340        PRINT"over."
1350        PRINT : PRINT "Mijn mening :"
1360        PRINT : PRINT : PRINT
1370        IF N < 3 THEN PRINT "UITMUNTEND !!!!!!!" : END
1380        IF N < 5 THEN PRINT "ZEER GOED GEDAAN !!!" : END
1390        IF N < 7 THEN PRINT "GOED ZO ......." : END
1400        IF N < 9 THEN PRINT "KAN ER MEE DOOR !!!" : END
1410        IF N<11 THEN PRINT "---- KON BETER ----" : END
1420        IF N<13 THEN PRINT "SLECHT HOOR !!!!" : END
1430        PRINT "--- VRESELYK SLECHT ---" : END
```

A.L. Mathlener
Kustweg 470
9933 BW Delfzijl
tel: 05960 - 15326

B R E E K : een soort BREAKOUT-spel, maar nu in FORTH.

Het commentaar staat tussen haakjes.

```
HEX                          ( getallen zijn hexadecimaal )
48Ø VARIABLE SPEED           ( bepaalt laagste speelsnelheid )
  Ø VARIABLE BAL             ( nieuwe positie speelbal )
  Ø VARIABLE BALL            ( positie speelbal )
  Ø VARIABLE TEL             ( aantal verspeelde ballen )
  Ø VARIABLE AANTAL          ( aantal gewenste ballen )
  Ø VARIABLE UUR             ( krijgt waarde gekozen vertraging )
  Ø VARIABLE SNV             ( verticale snelheid )
  Ø VARIABLE SNH             ( horizontale snelheid )
  Ø VARIABLE SCORE           ( aantal behaalde punten )
  Ø VARIABLE RECORD          ( het gestelde record )
  Ø VARIABLE PADDLE          ( positie van de paddle )
D766 VARIABLE BASTA          ( startpositie van de bal )
  Ø VARIABLE VELD            ( controle-vlag )
  Ø VARIABLE WEER            ( controle-vlag )
: NAAM 1 224 C! 1F 225 C! HOME  ( clearscreen en naam )
      A Ø DO CR LOOP
      A SPACES ." "B R E E K"
      2ØØØ Ø DO LOOP ;       ( delay )
: VR1 CR CR ." "GEWENSTE SNELHEID 1-9 (9=SNELSTE):"
      KEY 3Ø - 1 MAX 9 MIN DUP . SPEED SWAP 5Ø * - TO UUR ;
      ( KEY: wacht op toets en zet ASCII-waarde op de stack )
      ( MAX: verwijder laagste van 2 bovenste getallen van de stack )
      ( MIN: idem, maar nu hoogste )
      ( . = PRINT    SWAP: verwissel bovenste twee getallen van stack )
: VR2 CR CR ." "GEWENSTE AANTAL BALLEN (1-9):"
      KEY 30 - 1 MAX 9 MIN DUP . TO AANTAL
      2ØØØ Ø DO LOOP ;
: VR3 CR CR CR
      ." "LAAT <SHIFT LOCK>'TOETS OMHOOG KOMEN" CR CR
      ." "EN DRUK OP EEN TOETS" CR CR GR
      ." "L-<SHIFT> BEWEEGT DE PADDLE NAAR LINKS" CR CR
      ." "R-<SHIFT> BEWEEGT DE PADDLE NAAR RECHTS"
      KEY DROP ;             ( het resultaat van KEY mag niet op de
                               stack blijven staan )
: VID1 1 224 C! HOME 2 225 C! ( het speelveld wordt klaargemaakt )
      3Ø Ø DO A1 I D149 + C!
              A1 I D189 + C!
           LOOP ;
: VID2 1B Ø DO A1 I 4Ø * D1C9 + C!
              A1 I 4Ø * D1CA + C!
              A1 I 4Ø * D1F7 + C!
              A1 I 4Ø * D1F8 + C!
           LOOP ;
: VID3 5 Ø DO
          2C Ø DO
               BB J 40 * I + D1CB + C!
               LOOP
          LOOP ;
: START NAAM VR1 VR2 VR3 VID1 VID2 VID3
      D7AØ TO PADDLE
      Ø TO TEL
      Ø TO SCORE ;
```

```
: RESULT 9 222 C! HOME
      ." "BREEK    RECORD:" RECORD DECIMAL .
      ." "BAL:" TEL . HEX ;
: RESCORE 28 222 C! HOME
      ." SCORE:" SCORE DECIMAL . HEX ;
: ZETPA C4 PADDLE C! 84 PADDLE 1+ C!   ( de paddle krijgt zijn vorm )
       84 PADDLE 2 + C! C6 PADDLE 3 + C! ;
: ZETPAL ZETPA
       20 PADDLE 4 + C! ;          ( als paddle naar links schuift )
: ZETPAR ZETPA
       20 PADDLE 1 - C! ;          ( als paddle naar rechts schuift )
: TYD UUR Ø DO LOOP ;              ( vertraging )
: WAPA PADDLE DUP                  ( zijkantbewaking voor paddle )
       D7B3 > IF D7B3 TO PADDLE ENDIF
       D78B < IF D78B TO PADDLE ENDIF ;
: PALI DFØØ C@ FB =                ( is L-<SHIFT>-toets ingedrukt ? )
      IF -1 +TO PADDLE WAPA ZETPAL ENDIF ;
: PARE DFØØ C@ FD =                ( is R-<SHIFT>-toets ingedrukt ? )
       IF  1 +TO PADDLE WAPA ZETPAR ENDIF ;
: PIEP 8 Ø DO 4Ø FØØØ C!
          2 Ø DO LOOP
             Ø FØØØ C!
          3 Ø DO LOOP
          LOOP ;
: HOR SNH MINUS TO SNH ;           ( MINUS keert teken om van bovenste )
: VER SNV MINUS TO SNV ;           ( getal in de stack )
: BALVE BAL TO BALL SNV SNH + +TO BAL ;  ( bal beweegt )
: BALKA BAL C@ A1 =                      ( bal komt tegen de wand )
          IF BAL D1CØ <
             IF VER ELSE HOR ENDIF
             BALL TO BAL PIEP
          ENDIF ;
: BALO 2Ø BALL C! E2 BAL C! BAL TO BALL ; de bal loopt verder )
: SCOORT BAL C@ BB =                     ( schiet de bal een steen weg?)
          IF 1 +TO SCORE RESCORE PIEP VER ENDIF ;
: OMHOOG Ø TO SNH -4Ø TO SNV ;
: TPLØ SNH Ø = IF VER -1 TO SNH ENDIF ;   ( bal tegen linkerzijde paddle )
: TPL1Ø SNV Ø = IF OMHOOG ENDIF ;
: TPL1 SNH 1 =
        IF SNV Ø =
           IF OMHOOG
           ELSE HOR Ø TO SNV
           ENDIF
        ENDIF ;
: TPRØ SNH Ø = IF VER 1 TO SNH ENDIF ; ( bal tegen rechterzijde paddle )
: TPR1Ø SNV Ø = IF OMHOOG ENDIF ;
: TPR1 SNH -1 =
        IF SNV Ø =
           IF OMHOOG
           ELSE HOR Ø TO SNV
           ENDIF
        ENDIF ;
```

```
: BALPALI BAL C@ C4 -                  ( bal tegen linkerzijde paddle ? )
        IF TPLØ TPL1Ø TPL1 PIEP ENDIF ;
: BALPARE BAL C@ C6 -                  ( bal tegen rechterzijde paddle ? )
        IF TPRØ TPR1Ø TPR1 PIEP ENDIF ;
: BALPAMI BAL C@ 84 -                  ( bal op midden paddle ? )
        IF VER PIEP ENDIF ;
: BALPA BALPALI BALPAMI BALPARE ;      ( bal tegen paddle ? )
: WEGBAL BAL D8ØØ >                    ( bal verlaat speelveld ? )
        IF 1 +TO TEL 1 TO WEER ENDIF ;
: WACHT BEGIN DFØØ C@ 7F = UNTIL ;     ( is  REPEAT  ingedrukt ? )
: OPNIEUW 1E 224 C! 1F 225 C! HOME
        ." "VOOR NIEUW SPEL DRUK OP <REPEAT>"
        WACHT ;
: BETER? SCORE RECORD >                ( is record verbeterd ? )
        IF SCORE TO RECORD ENDIF
        Ø TO SCORE ;
: WEERBAL RESULT RESCORE OMHOOG        ( er komt een nieuwe bal )
        BASTA DUP TO BALL TO BAL
        Ø TO WEER
        UUR 2Ø - TO UUR                ( kleinere vertraging )
        UUR 4Ø < IF 4Ø TO UUR ENDIF ;
: LEEG? Ø TO VELD                      ( alle stenen weggeschoten ? )
        5 Ø DO                         ( dan nieuwe stenen )
            2C Ø DO                    ( en nieuwe ballen )
                J 4Ø * I + D1CB + C@ BB -
                IF  1 TO VELD  ENDIF
                LOOP
            LOOP
        VELD Ø = IF VID3 Ø TO TEL ENDIF ;
: SPRING BEGIN                         ( doe dit tot bal weg is )
        PALI PARE BALVE BALPA BALKA SCOORT TYD WEGBAL BALO  ZETPA
        WEER 1 = UNTIL ;
: BREUK BEGIN                          ( doe dit zolang er nog ballen )
       WEERBAL LEEG? SPRING            ( zijn )
       TEL AANTAL = UNTIL ;
: BREEK BEGIN                          ( het spel vam begin tot U op )
       START BREUK 9 222 C!            ( BREAK duwt )
       BETER? OPNIEUW
       AGAIN ;
```

Dit programma is geschreven in PE-FORTH V Ø.1
Als U een andere versie van FORTH heeft, controleer dan eerst of dezelfde standaard-woorden daarin voorkomen (b.v. misschien wordt i.p.v. ENDIF het woord THEN gebruikt).
Het programma is geschreven voor Superboard II met 48 karakter-scherm (of meer).

Er zit een klein foutje in het programma: de paddle wil pas bewegen, als de eerste bal verloren is gegaan.
Kunt U het foutje vinden ?

John Hermans

FORTH - HEXDUMP op het scherm.

```
( 64 bytes worden op het scherm geschreven )
HEX
  68 VARIABLE CC1
   8 VARIABLE CC2
D1D8 VARIABLE CC3
D1D8 VARIABLE VIDEO
2000 VARIABLE BYTE
: HAAL BYTE C@ DUP ;   ( 1 byte wordt opgehaald en 2x op stack gezet )
: HOOG 10 / DUP ;      ( 4 hoge bits extra op stack )
: LAAG 10 * - ;        ( 4 lage bits van byte in TOS )
: KARAKTER DUP 9 > IF 7 + ENDIF 30 + ;
                       ( van hex-waarde naar ASCII-waarde )
: HOGERVIDEO 1 +TO VIDEO ;  ( hoog schermadres op )
: HOGERBYTE 1 +TO BYTE ;    ( hoog geheugenadres op )
: WEERVIDEO CC3 TO VIDEO ;  ( herstel schermadres voor nieuwe pagina )
: SCHRIJF KARAKTER VIDEO C! HOGERVIDEO ;  ( schrijf ASCII-karakter
                  op het scherm en hoog schermadres op )
: VIDEOBIJ CC1 +TO VIDEO ;  ( sla regel over en begin op nieuwe regel )
: SPATIE 20 VIDEO C! HOGERVIDEO ;  ( schrijf spatie  op het scherm
                                en hoog schermadres op )
: EENBYTE HAAL HOOG SCHRIJF LAAG SCHRIJF SPATIE HOGERBYTE ;
                  ( schrijf byte plus spatie op het scherm )
: EENREGEL CC2 0 DO EENBYTE LOOP VIDEOBIJ ;
                  ( schrijf regel en sla regel over )
: EENPAGINA WEERVIDEO 8 0 DO EENREGEL LOOP ;
                  ( schrijf 8 regels op het scherm )
: 1P  HOME BYTE . EENPAGINA ;  ( maak scherm schoon - schrijf adres
                    eerste byte - schrijf 64 bytes )
: HEDU TO BYTE BEGIN 1P KEY 20 = UNTIL HOME ;
                  ( HEDU verwacht het adres van het eerste byte
                    dat op scherm gezet moet worden - toets geeft
                    nieuwe pagina - spatiebalk beeindigt programma )
```

HEDU is geen geweldig programma, want het gebruikt lang niet alle mogelijkheden van FORTH. Maar het werkt .
Om het te laten lopen moet U intikken:
HHLL HEDU return
HHLL is het adres vanaf waar de dumping geschiedt.

```
5 PRINTCHR$(26)
10 POKE11,213:POKE12,252:X=USR(X)
15 PRINTSPC(8);"SPACE REACTION TEST"
20 PRINT:PRINT"WRITTEN BY E. SONNEVELD & C. TOTTE":PRINT
25 TT=57088
30 LG=53478
35 RG=53498
40 BG=INT((RG+LG)/2)
45 FORI=0TO38
50 READJ:POKE586+I,J:NEXT
55 RESTORE
60 INPUT"DESIRED LEVEL (0-255): ";LE:PRINT
65 IFLE>255ORLE<0THEN60
70 POKE518,LE
75 INPUT"DESIRED DEPTH (0-20): ";DE:PRINT
80 DE=ABS(INT(DE))
85 IFDE>20THEN80
90 INPUT"INPUT RANDOM NUMBER (0-255): ";RN:PRINT
95 RN=ABS(INT(RN))
100 IFRN>255THEN90
105 POKE682,RN
110 BB=21
120 STER=42:SPATIE=32
130 LG=LG+32*DE
140 RG=RG+32*DE
150 BG=BG+32*DE
160 POKE11,213:POKE12,252:X=USR(X)
163 POKE11,74:POKE12,2
165 FORI=0TO16:X=USR(X)
170 R=INT(PEEK(682)/256*BB)
175 X=USR(X)
180 IFPEEK(682)/256<.95THENPRINTTAB(R);"*":GOTO200
185 X=USR(X)
190 PRINTTAB(INT(R*9/10));INT(PEEK(682)/256*9+1)
200 J=PEEK(TT)
210 DS=(J=250)-(J=252)
250 IFBG+DS<LGORBG+DS>RGTHENDS=0
260 BG=BG+DS:SC=PEEK(BG)
270 IFSC=SPATIETHEN300
280 IFSC=STERTHEN310
290 BO=BO+10*(SC-48)
300 POKEBG,238:NEXTI
310 FORJ=1TO230
320 POKEBG,J:NEXTJ
330 PRINTTAB(20); "SCORE: ";I;"   BONUS: ";BO:PRINT
340 I=I+BO:BO=0
360 PRINTTAB(20);"TOTAL: ";I:PRINT
370 IFI>HSTHENHS=I
380 PRINTTAB(20);"HIGH SCORE: ";HS:PRINT
390 PRINTTAB(20):INPUT"PLAY AGAIN (Y,N,O): ";A$:PRINT
400 IFLEFT$(A$,1)="Y"THEN160
410 IFLEFT$(A$,1)="O"THEN5
420 IFLEFT$(A$,1)<>"N"THEN390
430 END
440 DATA72,8,138,72,162,8,173,173,2,42
450 DATA42,42,42,77,170,2,42,42,42,110
460 DATA170,2,110,171,2,110,172,2,110,173
470 DATA2,202,208,228,104,170,40,104,96
```

```
024A 48      PHA
024B 08      PHP
024C 8A      TXA
024D 48      PHA
024E A208    LDX #$08
0250 ADAD02  LDA $02AD
0253 2A      ROL A
0254 2A      ROL A
0255 2A      ROL A
0256 2A      ROL A
0257 4DAA02  EOR $02AA
025A 2A      ROL A
025B 2A      ROL A
025C 2A      ROL A
025D 6EAA02  ROR $02AA
0260 6EAB02  ROR $02AB
0263 6EAC02  ROR $02AC
0266 6EAD02  ROR $02AD
0269 CA      DEX
026A D0E4    BNE $0250
026C 68      PLA
026D AA      TAX
026E 28      PLP
026F 68      PLA
0270 60      RTS
0271 EA      NOP
0272 EA      NOP
0273 EA      NOP
0274 EA      NOP
0275 EA      NOP
0276 EA      NOP
0277 EA      NOP
0278 EA      NOP
0279 EA      NOP
027A EA      NOP
027B EA      NOP
027C EA      NOP
027D EA      NOP
027E EA      NOP
027F EA      NOP
0280 EA      NOP
0281 EA      NOP
0282 EA      NOP
0283 EA      NOP
0284 EA      NOP
0285 EA      NOP
0286 EA      NOP
0287 EA      NOP
```

PROGRAMMA WERKING:
RUN, INITIALISATIE BESTAAT UIT:
LEVEL (0-255): SNELHEID VAN SCHRIJVEN OP HET SCHERM (MEESTAL 0-50)
DEPTH (0- 20): INDRINGDIEPTE VAN HET VLIEGTUIG
RANDOM (0-255): BEGINGETAL VOOR DE RANDOM GENERATOR

N. B. 518 = ADRES PRINT-TIMER VIDEO.

M. B. V. SHIFT RIGHT EN SHIFT LEFT IS HET VLIEGTUIG TE BESTUREN.
DE BEDOELING IS OM DE STERREN TE ONTWIJKEN EN DE GETALLEN TE RAKEN OM ZODOENDE BONUSPUNTEN TE VERKRIJGEN.
HET PROGRAMMA IS GESCHREVEN OP EEN 1K VIDEO, WAARBIJ DOOR MIJ ENIGE VIDEOPOINTERS ZIJN AANGEPAST AAN MIJN T. V. -BEELD. ZODOENDE ZULLEN DE REGELS 30-35 EXPERIMENTEEL VAST GESTELD MOETEN WORDEN.
L6 = LINKER BOVENGRENS
R6 = RECHTER BOVENGRENS - 1

OM HET PROGRAMMA SNELLER TE DRAAIEN: A) REGEL 170 BB INVULLEN
OM HET PROGRAMMA SNELLER TE DRAAIEN: A) REGEL 170 BB INVULLEN
B) REGEL 200 TT INVULLEN
C) REGEL 270/280 ASCII VAN EEN SPATIE EN EEN STER INVULLEN

VARIABELEN: TT = ADRES TOETSEN SHIFT LEFT EN SHIFT RIGHT
L6, R6 = LINKER RESP. RECHTER BOVENGRENS VIDEO SCHERM
B6 = START POSITIE EN ACTUELE POSITIE VLIEGTUIG
BB = BEELDSCHERMBREEDTE - 1 (BIJ PRINTEN VAN EEN TEKEN VOLGT ER SPATIE, DIE OP DE LAATSTE POSITIE VAN EEN REGEL EEN CR EXTRA GENEREERD)

NOVEMBER 1982

```
10 PRINTCHR$(26)
20 PRINT"#####   SATELLIETBAAN   #####":PRINT:PRINT
30 REM  VAN JOHN HERMANS   SCHIPLUIDEN
40 REM  AUGUSTUS 1982
50 REM  DE SATELLIETBAAN START OP 0 GR N.B.  EN 0 GR O.L.
60 REM  ELKE KEER ALS DE SATELLIET PASSEERT
70 REM  TUSSEN 48 EN 56 GR N.B.  EN 2 EN 10 GR O.L.
80 REM  WORDT DE DOORKOMSTTIJD GEGEVEN
90 REM  IN DAGEN UREN EN MINUTEN VANAF DE START
100 INPUT"GEEF OMLOOPTIJD IN MINUTEN: ";O:PRINT
110 INPUT"GEEF HOEK BAAN MET EVENAAR IN GR. : ";AL:PRINT
120 PI=3.141592:DA=8:DB=1:T=0
150 PRINTCHR$(26)
160 POKE517,1
170 PRINTCHR$(9);"SATELLIETBAAN":PRINT:PRINT
190 PRINT"STARTTIJD: DAG 0  UUR 0  MINUUT 0":PRINT
195 PRINT"POSITIE OP STARTTIJD:  0 GR.  N.B.  0 GR.  O.L. ":PRINT
200 PRINT"HOEK MET EVENAAR = ";AL;" GRADEN":PRINT
210 PRINT"OMLOOPTIJD       = ";O;" MINUTEN":PRINT
220 AL=AL*PI/180
230 PRINT"TIJDSTIP VAN PASSEREN:":PRINT
240 PRINT"  DAG   UUR  MINUUT   N.B.       O.L. ":PRINT
250 POKE517,0
260 C=2*PI*T/O:A=SIN(AL)*SIN(C):B=SQR(1-A*A):CC=COS(AL)
270 TA=SIN(C)/COS(C):BB=ATN(CC*TA)-T*PI/720
280 IFBB<-2*PITHENBB=BB+2*PI:GOTO280
290 ONM+1GOTO300,320,340,360
300 IFBB>=0THEN380
310 M=1
320 IFBB<0THENBB=BB+PI:GOTO380
330 M=2
340 IFBB>=0THENBB=BB+PI:GOTO380
350 M=3
360 IFBB<0THENBB=BB+2*PI:GOTO380
370 M=0
380 AA=ATN(A/B)*180/PI:BB=BB*180/PI
390 IFAA<48ORAA>56THENT=T+1:GOTO260
400 IFBB<20RBB>10THENT=T+1:GOTO260
410 GOSUB420:T=T+1:GOTO260
420 HH=INT(T/1440):HH$=STR$(HH)+" "
430 IFLEN(HH$)<5THENHH$=" "+HH$:GOTO430
440 U=T-HH*1440:UU=INT(U/60):UU$=STR$(UU)+" "
450 IFLEN(UU$)<5THENUU$=" "+UU$:GOTO450
460 MM=T-HH*1440-UU*60:MM$=STR$(MM)+" "
470 IFLEN(MM$)<5THENMM$=" "+MM$:GOTO470
480 A$=HH$+CHR$(58)+UU$+CHR$(58)+MM$
490 POKE517,1
500 PRINTA$;
510 DC=AA:GOSUB34000:PRINTSPC(DH);DB$;
520 DC=BB:GOSUB34000:PRINTSPC(DH);DB$
530 POKE517,0
540 RETURN
```

NOVEMBER 1982

```
 34000 REM GETAL-'FORMAT'-SUBROUTINE
 34001 REM GEBRUIKTE VARIABELEN ZIJN DA T/M DH EN DA$,DB$,DC$
 34002 REM ALVORENS SUBROUTINE AAN TE ROEPEN IN HET PROGRAMMA OPGEVEN
 34003 REM DA = TOTAAL AANTAL POSITIES VAN DE UITVOER MET EV. MINTEKEN
 34004 REM DB = GEWENST AANTAL CIJFERS ACHTER DE KOMMA
 34005 REM DC = WAARDE VAN DE VARIABELE, DIE GEPRINT MOET WORDEN
 34006 REM IN PROGRAMMA NA SUBROUTINE-AANROEP 'PRINTSPC(DH);DB$' ETC.
 34007 DD=10^DB:DE=0:IFDC<0THENDE=1:DC=-DC
 34008 DF=INT(DC+.5/DD):DG=INT((DC-DF)*DD+.5)
 34009 DC$=STR$(DF):DA$=STR$(DG):IFDB=0THENDB=1
 34010 DC$=RIGHT$(DC$,LEN(DC$)-1):DA$=RIGHT$(DA$,LEN(DA$)-1)
 34011 IFLEN(DA$)>DBTHENDA$=LEFT$(DA$,DB)
 34012 IFLEN(DA$)<DBTHENDA$="0"+DA$:GOTO34012
 34013 DB$=DC$+". "+DA$:DH=DA+1-LEN(DB$)
 34016 IFDE=1THENDB$="-"+RIGHT$(DB$,DA-1):DH=DH-1
 34017 RETURN
OK
RUN

#####   SATELLIETBAAN   #####

GEEF OMLOOPTIJD IN MINUTEN: ? 91.5

GEEF HOEK BAAN MET EVENAAR IN GR. : ? 55

SATELLIETBAAN

STARTTIJD: DAG 0  UUR 0  MINUUT 0

POSITIE OP STARTTIJD:  0 GR. N.B.  0 GR. O.L.

HOEK MET EVENAAR =  55  GRADEN

OMLOOPTIJD       =  91.5  MINUTEN

TIJDSTIP VAN PASSEREN:

  DAG   UUR  MINUUT   N.B.      O.L.

   0 :   3 :  20      48.9      3.3
   0 :   3 :  21      50.7      8.5
   0 :   4 :  56      54.6      6.6
   0 :  18 :  43      54.1      3.6
   0 :  18 :  44      53.2      9.8
   1 :   3 :  45      50.7      2.5
   1 :   3 :  46      52.2      8.1
   1 :   5 :  21      55.0      7.2
   1 :  19 :   8      53.2      3.8
   1 :  19 :   9      51.9      9.7
   2 :   4 :  10      52.2      2.1
   2 :   4 :  11      53.4      8.0
   2 :  17 :  58      54.9      4.8
   2 :  19 :  33      51.9      3.6
   2 :  19 :  34      50.3      9.1
   3 :   4 :  35      53.4      2.0

 NOVEMBER 1982
```

MATRIX BEWERKINGEN

Dit zijn standaard subroutines met allerlei soorten matrix-bewerkingen. De REM statements dienen alleen ter instruktie en kunnen bij gebruik het beste worden weggelaten.
De gebruikte variabelen zijn: (afhankelijk van het gebruikte deel !)
IA(IR,IK) , IB(,),IC(,),ID(,),IE(,),IF(,),IG(,) en IH , II , IJ , IK , IL , IM , IR , IS , IT , IU.

```
40000 REM LEES MATRIX UIT EEN DATA-STMNT
40002 REM IN HET HOOFDPROGRAMMA MOET IA(IR,IK) ZIJN GEDIMENSIONEERD
40004 REM IA(IR,IK) IS DE MATRIX MET IR RIJEN EN IK KOLOMMEN
40006 REM IJ EN II WORDEN ALS TELLERS GEBRUIKT.
40010 FORII=1TORA:FORIJ=1TOKA:READ(II,IJ):NEXTIJ,II:RETURN
40019 ##########################################
40020 REM LEES MATRIX MET INPUT-STMNT
40022 REM IN HET HOOFDPROGRAMMA MOET IA(IR,IK) ZIJN GEDIMENSIONEERD
40024 REM IA(IR,IK) IS DE MATRIX MET IR RIJEN EN IK KOLOMMEN
40026 REM IJ EN II WORDEN ALS TELLERS GEBRUIKT.
40030 FORII=1TOIR:PRINT"RIJ NR"II:FORIJ=1TOIK:PRINT"KOLOM "IJ;
40035 INPUTA(II,IJ):NEXTIJ,II:RETURN
40039 ##########################################
40040 REM MATRIX-OPTELLING
40042 REM IN HET HOOFDPROGRAMMA MOETEN IA(IR,IK),IB(IR,IK) EN IC(IR,IK)
40043 REM ZIJN GEDIMENSIONEERD.GEBRUIK (IA),(IB) EN (IC) LIEFST ALLEEN
40044 REM ALS HULPMATRICES OM MEE TE REKENEN !!
40045 REM (IA),(IB) EN (IC) ZIJN MATRICES MET IR RIJEN EN IK KOLOMMEN
40046 REM IJ EN II WORDEN ALS TELLERS GEBRUIKT.
40050 FORII=1TOIR:FORIJ=1TOIK:IC(II,IJ)=IA(II,IJ)+IB(II,IJ)
40055 NEXTIJ,II:RETURN
40059 ##########################################
40060 REM MATRIX-AFTREKKING
40062 REM IN HET HOOFDPROGRAMMA MOETEN IA(IR,IK),IB(IR,IK) EN IC(IR,IK)
40063 REM ZIJN GEDIMENSIONEERD.GEBRUIK (IA),(IB) EN (IC) LIEFST ALLEEN
40064 REM ALS HULPMATRICES OM MEE TE REKENEN !!
40065 REM (IA),(IB) EN (IC) ZIJN MATRICES MET IR RIJEN EN IK KOLOMMEN
40066 REM IJ EN II WORDEN ALS TELLERS GEBRUIKT.
40070 FORII=1TOIR:FORIJ=1TOIK:IC(II,IJ)=IA(II,IJ)-IB(II,IJ)
40075 NEXTIJ,II:RETURN
40079 ##########################################
40080 REM MATRIX VERMENIGVULDIGING
40082 REM IN HET HOOFDPROGRAMMA MOETEN IA(IR,IK),IB(IS,IL) EN IC(IT,IM)
40083 REM ZIJN GEDIMENSIONEERD.GEBRUIK (IA),(IB) EN (IC) LIEFST ALLEEN
40084 REM ALS HULPMATRICES OM MEE TE REKENEN !!
40085 REM (IA),(IB) EN (IC) ZIJN MATRICES MET IR RIJEN EN IK KOLOMMEN
40086 REM IJ EN II WORDEN ALS TELLERS GEBRUIKT.
40087 REM LET WEL: IK=IS, IR=IT EN IL=IM, ANDERS VOLGT FOUTMELDING
40088 REM GEEF IK,IS,IR,IT,IL EN IM OP VOORDAT DE SUBR. WORDT GEROEPEN
40089 REM IN IEDER GEVAL IM,IT EN IS ALS REGEL 40090 WORDT WEGGELATEN.
40090 IFIK()ISORIR()ITORIL()IMTHEN40098
40091 IU=0:FORIH=1TOIT:FORII=1TOIM:FORIJ=1TOIS
40092 IU=IU+(IA(IH,IJ)*IB(IJ,II)):NEXTIJ:IC(IH,II):IU=0
40093 NEXTII,IH:IU=0:RETURN
40098 PRINT"DE VOLGENDE WAARDEN MOETEN GELIJK ZIJN"
40099 PRINTIK"="IS,IR"="IT,IL"="IM:STOP:##################################
```

```
40100 REM MATRIX DELING
40102 REM IN HET HOOFDPROGRAMMA MOETEN IA(IR,IK),IB(IS,IL) EN IC(IT,I
40103 REM ZIJN GEDIMENSIONEERD.GEBRUIK (IA),(IB) EN (IC) LIEFST ALLEEN
40104 REM ALS HULPMATRICES OM MEE TE REKENEN !!
40105 REM (IA),(IB) EN (IC) ZIJN MATRICES MET IR RIJEN EN IK KOLOMMEN
40106 REM IJ EN II WORDEN ALS TELLERS GEBRUIKT.
40107 REM LET WEL: IK=IS, IR=IT EN IL=IM, ANDERS VOLGT FOUTMELDING
40108 REM GEEF IK,IS,IR,IT,IL EN IM OP VOORDAT DE SUBR. WORDT GEROEPEN
40109 REM IN IEDER GEVAL IM,IT EN IS ALS REGEL 40090 WORDT WEGGELATEN.
40110 IFIK()ISORIR()ITORIL()IMTHEN40098
40111 IU=0:FORIH=1TOIT:FORII=1TOIM:FORIJ=1TOIS
40112 IU=IU+(IA(IH,IJ)/IB(IJ,II)):NEXTIJ:IC(IH,II):IU=0
40113 NEXTII,IH:IU=0:RETURN
40118 PRINT"DE VOLGENDE WAARDEN MOETEN GELIJK ZIJN"
40119 PRINTIK"="IS,IR"="IT,IL"="IM:STOP:##############################
40120 REM EENHEIDSMATRIX
40122 REM DAT IS EEN MATRIX MET ENEN ( 1'en ) OP DE HOOFDDIAGONAAL
40123 REM EN VERDER NULLEN.
40124 REM (IE) HEEFT DE DIMENSIES IE(IK,IR)
40125 REM II EN IJ ZIJN DE TELLERS, IK EN IR MOETEN ZIJN BEPAALD
40126 REM LET WEL IK MOET GELIJK ZIJN AAN IR !! (TEST IN 40129)
40129 IFIK()IRTHEN40138
40130 FORII=1TOIK:FORIJ=1TOIR:IE(II,IR)=0:NEXTIJ,II
40132 FORII=1TOIK:IE(II,II)=1:NEXTII:RETURN
40138 PRINT"HET AANTAL RIJEN EN KOLOMMEN MOET GELIJK ZIJN !"
40139 PRINT"ECHTER IK="IK" EN IR="IR:STOP:#########################
40140 REM MATRIXTRANSFORMATIE
40142 REM DE KOLOM- EN RIJ- ELEMENTEN WORDEN VERWISSELD
40143 REM IA(IK,IR) EN IB(IR,IK) WORDEN GEBRUIKT
40144 REM LET OP DE DIMENSIONERING VAN RIJEN EN KOLOMMEN
40145 REM MATRIX (A) WORDT GETRANSPONEERD IN (B)
40150 FORII=1TOIK:FORIJ=1TOIR:IB(IJ,II)=IA(II,IJ):NEXTIJ,II:RETURN
40159 ###########################################
40160 REM MATRIX KOPIEEREN
40162 REM VOOR IEDERE MATRIX IS EEN KORRESPONDERENDE MATRIX ALS
40164 REM TIJDELIJKE OPBERGPLAATS.
40165 REM GEBRUIK TELKENS EEN ANDERE INGANG, AL NAAR GELANG DE KOPIE,
40166 REM B.V. (A)-)(B),(A)-)(C),(A)-)(D) ETC.
40167 REM LET ZELF OP DE DIMENSIES, ER WORDT NIET OP GETEST
40170 FORII=1TOIK:FORIJ=1TOIR:IA(II,IJ)=IB(II,IJ):NEXTIJ,II:RETURN
40171 FORII=1TOIK:FORIJ=1TOIR:IA(II,IJ)=IC(II,IJ):NEXTIJ,II:RETURN
40172 FORII=1TOIK:FORIJ=1TOIR:IA(II,IJ)=ID(II,IJ):NEXTIJ,II:RETURN
40173 FORII=1TOIK:FORIJ=1TOIR:IA(II,IJ)=IE(II,IJ):NEXTIJ,II:RETURN
40174 FORII=1TOIK:FORIJ=1TOIR:IA(II,IJ)=IF(II,IJ):NEXTIJ,II:RETURN
40175 FORII=1TOIK:FORIJ=1TOIR:IA(II,IJ)=IG(II,IJ):NEXTIJ,II:RETURN
40176 FORII=1TOIK:FORIJ=1TOIR:IB(II,IJ)=IA(II,IJ):NEXTIJ,II:RETURN
40177 FORII=1TOIK:FORIJ=1TOIR:IB(II,IJ)=IC(II,IJ):NEXTIJ,II:RETURN
40178 FORII=1TOIK:FORIJ=1TOIR:IB(II,IJ)=ID(II,IJ):NEXTIJ,II:RETURN
40179 FORII=1TOIK:FORIJ=1TOIR:IB(II,IJ)=IE(II,IJ):NEXTIJ,II:RETURN
40180 FORII=1TOIK:FORIJ=1TOIR:IB(II,IJ)=IF(II,IJ):NEXTIJ,II:RETURN
40181 FORII=1TOIK:FORIJ=1TOIR:IB(II,IJ)=IG(II,IJ):NEXTIJ,II:RETURN
40182 FORII=1TOIK:FORIJ=1TOIR:IC(II,IJ)=IA(II,IJ):NEXTIJ,II:RETURN
40183 FORII=1TOIK:FORIJ=1TOIR:IC(II,IJ)=IB(II,IJ):NEXTIJ,II:RETURN
40184 FORII=1TOIK:FORIJ=1TOIR:IC(II,IJ)=ID(II,IJ):NEXTIJ,II:RETURN
40185 FORII=1TOIK:FORIJ=1TOIR:IC(II,IJ)=IE(II,IJ):NEXTIJ,II:RETURN
```

```
40186 FORII=1TOIK:FORIJ=1TOIR:IC(II,IJ)=IF(II,IJ):NEXTIJ,II:RETURN
40187 FORII=1TOIK:FORIJ=1TOIR:IC(II,IJ)=IG(II,IJ):NEXTIJ,II:RETURN
40188 FORII=1TOIK:FORIJ=1TOIR:ID(II,IJ)=IA(II,IJ):NEXTIJ,II:RETURN
40189 FORII=1TOIK:FORIJ=1TOIR:ID(II,IJ)=IB(II,IJ):NEXTIJ,II:RETURN
40190 FORII=1TOIK:FORIJ=1TOIR:ID(II,IJ)=IC(II,IJ):NEXTIJ,II:RETURN
40191 FORII=1TOIK:FORIJ=1TOIR:ID(II,IJ)=IE(II,IJ):NEXTIJ,II:RETURN
40192 FORII=1TOIK:FORIJ=1TOIR:ID(II,IJ)=IF(II,IJ):NEXTIJ,II:RETURN
40193 FORII=1TOIK:FORIJ=1TOIR:ID(II,IJ)=IG(II,IJ):NEXTIJ,II:RETURN
40194 FORII=1TOIK:FORIJ=1TOIR:IE(II,IJ)=IA(II,IJ):NEXTIJ,II:RETURN
40195 FORII=1TOIK:FORIJ=1TOIR:IE(II,IJ)=IB(II,IJ):NEXTIJ,II:RETURN
40196 FORII=1TOIK:FORIJ=1TOIR:IE(II,IJ)=IC(II,IJ):NEXTIJ,II:RETURN
40197 FORII=1TOIK:FORIJ=1TOIR:IE(II,IJ)=ID(II,IJ):NEXTIJ,II:RETURN
40198 FORII=1TOIK:FORIJ=1TOIR:IE(II,IJ)=IF(II,IJ):NEXTIJ,II:RETURN
40199 FORII=1TOIK:FORIJ=1TOIR:IE(II,IJ)=IG(II,IJ):NEXTIJ,II:RETURN
40200 FORII=1TOIK:FORIJ=1TOIR:IF(II,IJ)=IA(II,IJ):NEXTIJ,II:RETURN
40201 FORII=1TOIK:FORIJ=1TOIR:IF(II,IJ)=IB(II,IJ):NEXTIJ,II:RETURN
40202 FORII=1TOIK:FORIJ=1TOIR:IF(II,IJ)=IC(II,IJ):NEXTIJ,II:RETURN
40203 FORII=1TOIK:FORIJ=1TOIR:IF(II,IJ)=ID(II,IJ):NEXTIJ,II:RETURN
40204 FORII=1TOIK:FORIJ=1TOIR:IF(II,IJ)=IE(II,IJ):NEXTIJ,II:RETURN
40205 FORII=1TOIK:FORIJ=1TOIR:IF(II,IJ)=IG(II,IJ):NEXTIJ,II:RETURN
40206 FORII=1TOIK:FORIJ=1TOIR:IG(II,IJ)=IA(II,IJ):NEXTIJ,II:RETURN
40207 FORII=1TOIK:FORIJ=1TOIR:IG(II,IJ)=IB(II,IJ):NEXTIJ,II:RETURN
40208 FORII=1TOIK:FORIJ=1TOIR:IG(II,IJ)=IC(II,IJ):NEXTIJ,II:RETURN
40209 FORII=1TOIK:FORIJ=1TOIR:IG(II,IJ)=ID(II,IJ):NEXTIJ,II:RETURN
40210 FORII=1TOIK:FORIJ=1TOIR:IG(II,IJ)=IE(II,IJ):NEXTIJ,II:RETURN
40211 FORII=1TOIK:FORIJ=1TOIR:IG(II,IJ)=IF(II,IJ):NEXTIJ,II:RETURN
40215 REM HET IS VERSTANDIG OM NIET GEBRUIKTE KOMBINATIES
40216 REM WEG TE LATEN, ( OM WAT GEHEUGEN TE SPAREN).
40219 ##########################################
62999 END
63000 SAVE:PRINT:PRINT:PRINT:PRINT"Matrix subroutines 40000-";
63001 PRINTTAB(48)CHR$(64):PRINT:PRINT:PRINT:PRINT:PRINT
63002 PRINT:PRINT:PRINT"MATINR"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63003 POKE4,194:POKE5,165:LIST40000-40019:POKE4,195:POKE5,168
63004 PRINT:PRINT:PRINT"MATINP"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63005 POKE4,194:POKE5,165:LIST40020-40039:POKE4,195:POKE5,168
63006 PRINT:PRINT:PRINT"MATPLS"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63007 POKE4,194:POKE5,165:LIST40040-40059:POKE4,195:POKE5,168
63008 PRINT:PRINT:PRINT"MATMIN"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63009 POKE4,194:POKE5,165:LIST40060-40079:POKE4,195:POKE5,168
63010 PRINT:PRINT:PRINT"MATMUL"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63011 POKE4,194:POKE5,165:LIST40080-40099:POKE4,195:POKE5,168
63012 PRINT:PRINT:PRINT"MATDEE"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63013 POKE4,194:POKE5,165:LIST40100-40119:POKE4,195:POKE5,168
63014 PRINT:PRINT:PRINT"MATEEN"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63015 POKE4,194:POKE5,165:LIST40120-40139:POKE4,195:POKE5,168
63016 PRINT:PRINT:PRINT"MATTRS"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63017 POKE4,194:POKE5,165:LIST40140-40159:POKE4,195:POKE5,168
63018 PRINT:PRINT:PRINT"MATKOP"TAB(48)CHR$(64):PRINT:PRINT:PRINT
63019 POKE4,194:POKE5,165:LIST40160-40219:POKE4,195:POKE5,168
63020 PRINT:PRINT:PRINT"EINDE MATRIXSUBROUTINES"CHR$(64):PRINT:PRINT
63025 PRINT:PRINT:PRINT:PRINT"Matrix-save programma.."TAB(48)CHR$(64)
63030 POKE4,194:POKE5,165:LIST62999-:POKE4,195:POKE5,168
63040 PRINT:PRINT:PRINT:PRINT"POKE515,0":PRINT:PRINT:POKE517,0
```

Jos Burghouts

PIEPJE voor C1P en Superboard.

Dit is een machinetaal-subroutine voor C1P en Superboard II met de nieuwe supportprom, waarmee de luidspreker een kort piepje laat horen. Het ML-programma wordt geladen in pagina 2 vanaf $0240. (Ook $023F wordt gebruikt.)

```
$0240  50              toon1
    1  50              toon2
    2  50              lengte
    3  AD4202          LDA $0242    laad lengte
    6  8D3F02  loop1   STA $023F    store lengte'
    9  AE4002          LDX $0240    laad toon1
    C  AC4102          LDY $0241    laad toon2
    F  A951            LDA  $51     tokkel
$0251  8D00F0          STA $F000    luidspreker
    4  CA      loop2   DEX          toon1=toon1-1
    5  D0FD            BNE loop2    niet nul dan terug
    7  A911            LDA  $11     tokkel
    9  8D00F0          STA $F000    luidspreker
    C  88      loop3   DEY          toon2=too2-1
    D  D0FD            BNE loop3    niet 0 ? dan terug
    F  CE3F02          DEC $023F    lengte'=lengte'-1
$0262  D0E5            BNE loop1    niet 0 ? dan nog een keer
$0264  60              RTS          klaar
```

```
44000 REM GELUIDSEFFECT (PIEPJE) ML-DEEL WORDT GELADEN MET GOSUB44011
44001 REM GELUIDSEFFECT IN PROGRAMMA OPROEPEN MET GOSUB44011
44002 REM GEBRUIKT WORDEN VARIABELEN MA-ME EN VAN 0-PAGE E0-E3
44003 REM ML-DEEL KAN OVERAL IN RAM GELADEN WORDEN
44004 MA=576:REM STARTADRES ML-DEEL. KAN VERANDERD WORDEN.
44005 READMB:IFMB()567THEN44005
44006 MC=INT(MA/256):MD=MA-256*MC
44007 READMB:IFMB(0THEN44009
44008 POKEMA,MB:MA=MA+1:GOTO44007
44009 POKE225,50:POKE226,50:POKE227,50
44010 RETURN
44011 MA=PEEK(11):MB=PEEK(12):POKE11,MD:POKE12,MC
44012 ME=USR(ME):POKE11,MA:POKE12,MB:RETURN
44013 DATA567,165,227,133,224,166,225,164,226,169,64,141,0,240,202
44014 DATA208,253,169,0,141,0,240,136,208,253,198,224,208,232,96,-1
```

John Hermans

32000 - 32035 PLOT SUBROUTINE

IN CA = snijpunt X en Y as op scherm
53248 < CA < 54272 ⇒ 24 tekens
53248 < CA < 55296 ⇒ 48 tekens
CB = X richting met 0 < ABS(CB) < 48
CC = Y richting

GEBRUIKT CA, CB, CC, CD, CE, CF, CG, CH, CI, CJ, CL.

```
32000 REM PLOT-SUBROUTINE VOOR SUPERBOARD CA 900 BYTE
32001 REM GEBRUIKTE VARIABELEN CA T/M CL
32002 REM CA,CB,CC KRIJGEN IN PROGRAMMA GEHEEL GETAL (!!!) ALS WAARDE
32003 REM CA IS SNIJPUNT X- EN Y-AS 53248(CA(54272 VOOR 24 KARAKTERS
32004 REM 53248(CA(55296 VOOR 48 KARAKTERS
32005 REM CB IS X-RICHTING  CC IS Y-RICHTING
32006 REM 0(ABS(CB)(48 EN 0(ABS(CC)(48 VOOR 24 KARAKTERS
32007 REM 0(ABS(CB)(64 EN 0(ABS(CC)(96 VOOR 48 KARAKTERS
32008 CD=64:REM CD=32 VOOR 24 KARAKTERS
32009 CE=CB-INT(CB/2)*2:CF=CC-INT(CC/2)*2
32010 CI=-(CE=0ANDCF=0):CJ=-(CE=0ANDCF()0)
32011 CK=-(CE()0ANDCF=0):CL=-(CE()0ANDCF()0)
32012 CG=CA+INT(CB/2)-INT(CC/2)*CD
32013 IFPEEK(CG)=161THENRETURN
32014 CH=ABS(PEEK(CG)-159):IFCH=0ORCH)19THENCH=1
32015 IFCH)15THENCH=CH-15:GOTO32019
32016 IFCH)6THENCH=CH-6:GOTO32018
32017 ONCHGOTO32020,32022,32023,32024,32025,32026
32018 ONCHGOTO32027,32028,32029,32030,32031
32019 ONCHGOTO32032,32033,32034,32035
32020 IFPEEK(CG)()32THENRETURN
32021 POKECG,167*CI+168*CJ+165*CK+166*CL:RETURN
32022 POKECG,176*CI+177*CJ+157*(CK+CL):RETURN
32023 POKECG,156*(CI+CJ)+178*CK+175*CL:RETURN
32024 POKECG,175*CI+177*CJ+155*(CK+CL):RETURN
32025 POKECG,154*(CI+CK)+178*CJ+176*CL:RETURN
32026 POKECG,154*CI+169*CJ+165*CK+157*CL:RETURN
32027 POKECG,170*CI+155*CJ+157*CK+166*CL:RETURN
32028 POKECG,167*CI+156*CJ+154*CK+170*CL:RETURN
32029 POKECG,156*CI+168*CJ+169*CK+155*CL:RETURN
32030 POKECG,178*CI+169*(CJ+CK)+177*CL:RETURN
32031 POKECG,170*(CI+CL)+175*CJ+176*CK:RETURN
32032 POKECG,161*CK-175*(CK=0):RETURN
32033 POKECG,161*CJ-176*(CJ=0):RETURN
32034 POKECG,161*CI-177*(CI=0):RETURN
32035 POKECG,161*CL-178*(CL=0):RETURN
```

John Hermans

Rechthoek Plotter (regels 32040-32070)

```
==Variabelen CD,CE,DF,CG,CH,CI,CJ,CK,CL,CM,RL
==In CD=aanvangspunt,nulpunt scher linksboven.
     CE=breedte, lengte
     CF=hoogte van de          rechthoek
     CG=horizontale positie
     CH=vertikale positie      van de linker-bovenhoek
     RL=regellengte van de video
==Uit geen variabelen.
==Auteur Hans Coenen.
```

```
32040 REM Rechthoek-plotter
32041 REM CI t/m CK en CM zijn hoekpunten, CL is een teller
32042 REM De waarden zijn voor een SUPERBOARD met 48 kar. breed video.
32050 CD = 53515
32052 IF CE<=1 OR CF<=1 THEN PRINT"Invoer te klein":RETURN
32054 CI = CD + RL*CH +CG: CJ = CI+CE-1: CK = CI+RL*(CF-1)
32056 CM = CJ + RL*(CF-1)
32058 POKE CI,210: POKE CJ,207: POKE CK,209: POKE CM,208:
      REM De hoeken.
32060 IF CE<=2 AND CF<=2 THEN RETURN
32062 IF CE<=2 AND CF>2 THEN 32068
32064 FOR CL =1 TO CE-2: POKECI+CL,135: POKECK+CL,128: NEXT CL
32066 IF CE>2 AND CF<=2 THEN RETURN
32068 FOR CL =1 TO CE-2: POKECI+CL*RL,136:POKECJ+CL*RL,143:NEXT CL
32070 RETURN
```

TEKST PLOTTER (regels 32080-32096)

```
==Variabelen CD,CE,CF,CG,CH,RL,CA$
==In CD=aanvangspunt,nulpunt scherm linksboven.
     CG=horizontale positie
     CH=vertikale positie      ofwel, begin tekststring t.o.v. CD
     CA$=te plotten regel   /  RL=regellengte
==Uit geen variabelen
==Auteur Hans Coenen.
```

```
32080 REM Tekst-plotter
32082 REM CF is een teller, CE is het startpunt van de string
32084 REM De waarden gelden voor een SUPERBOARD met 48 kar. video
32090 CD = 53515
32092 CE = CD + RL * CH + CG
32094 FOR CF=1TO LEN(CA$):POKE CE+CF,ASC(MID$(CA$,CF)):NEXT CF
32096 RETURN
```

40300-40308 SORTEERROUTINE
==========

GEBRUIKTE VARIABELEN: IN, IQ, IP, IQ
==========
IN IN = AANTAL ELEMENTEN IN MATRIX[IA], GESORTEERD
MOET WORDEN VAN LAAG NAAR HOOG.
IA = EEN-DIMENSIONALE MATRIX MET MINIMAAL
IN ELEMENTEN.
UIT IA = GESORTEERDE MATRIX

AUTEUR: A. SICCAMA
==========

```
40300 REM SORTEREN
40301 REM DOOR A. SICCAMA
40302 IP = 0
40303 IF IA(IQ) > IA(IQ+1) THEN GOSUB 40307
40304 NEXT IQ: IF IP THEN 40302
40306 RETURN
40307 IO = IA(IQ)   IA(IQ) = IA(IQ+1)   IA(IQ+1)=IO   IP = 1
40308 RETURN
```

40400-40424 LINEAIRE REGRESSIE
==========

GEBRUIKTE VARIABELEN: JA T/M JE, JF(), JG(), JH T/M JO
==========
IN JI = AANTAL PUNTEN X,Y
JF(JI) = ARRAY MET X-WAARDEN
JG(JI) = ARRAY MET BIJBEHORENDE Y-WAARDEN
UIT JM = RICHTINGS-COEFFICIENT
JN = Y-WAARDE VOOR X = 0
(VERGELIJKING: Y = JM * X + JN)
JO = CORRELATIE-COEFFICIENT (0 <= JO <= 1)
(BETERE AANPASSING NAARMATE JO DICHTER BIJ 1 LIGT)

AUTEUR: A. SICCAMA
==========

```
40400 REM LINEAIRE REGRESSIE
40401 REM DOOR A. SICCAMA
40410 JA = 0 : JB = 0 : JC = 0   JD = 0 : JE = 0
40411 FOR JH = 1 TO JI
40412 JA = JA + JF(JH) * JG(JH)
40413 JB = JB + JF(JH)
40414 JC = JC + JG(JH)
40415 JD = JD + JF(JH) * JF(JH)
40416 JE = JE + JG(JH) * JG(JH)
40417 NEXT JH
40418 JJ = JA - JB * JC / JI
40419 JK = JD - JB * JB / JI
40420 JL = JE - JC * JC / JI
40421 JM = JJ / JK
40422 JN = (JC - JM * JB) / JI
40423 JO = SQR(JJ * JJ / JK / JL)
40424 RETURN
```

40500-40524 INTEGREREN MET SIMPSON METHODE
==========

AUTEUR : A. SICCAMA
==========

GEBRUIKTE VARIABELEN : JA T/M JK
==========
IN JB = ONDERWAARDE INTEGRAAL
JE = BOVENWAARDE INTEGRAAL
JC = NAUWKEURIGHEID
FNJ() = TE INTEGREREN FUNCTIE
UIT JI = OPPERVLAKTE (INTEGRAAL)
(OPPERVLAKTEN ONDER X-AS WORDEN
NEGATIEF GETELD.)

OPMERKINGEN BIJ DE REGELS 40523 - 40526
-A- ABSOLUTE FOUT: HET ANTWOORD KAN MAXIMAAL JC
IN WAARDE AFWIJKEN
-B- RELATIEVE FOUT: HET ANTWOORD KAN MAXIMAAL
JC * HET ANTWOORD AFWIJKEN
-C- PROCENTUELE FOUT: HET ANTWOORD KAN MAXIMAAL
JC % AFWIJKEN

TIPS BIJ HET INTEGREREN VOLGENS SIMPSON-METHODE
-- ALS HET ANTWOORD DICHT BIJ 0 LIGT, NEEM DAN ALTIJD
DE ABSOLUTE FOUTEN-MARGE (ANDERS WORDT MOGELIJK
DOOR 0 GEDEELD)
-- NEEM VOOR FNJ() EEN VLOEIENDE FUNCTIE-INTERVAL,
(DUS NIET 1/X INTEGREREN VOOR -1 <= X <= 2)
-- OPPERVLAKTEN BENEDEN DE X-AS WORDEN NEGATIEF GETELD !!!

```
 40500 REM INTEGREREN VOLGENS SIMPSON-METHODE
 40501 REM DOOR A. SICCAMA
 40510 JA = (JE - JB) / 2
 40511 JD = FNJ(JB) + FNJ(JE)
 40512 JF = 0
 40513 JG = FNJ(JB + JA)
 40514 JH = (JD + 2 * JF + 4 * JG) * JA / 3
 40515 JF = JF + JG
 40516 JG = 0
 40517 JA = JA / 2
 40518 FOR JJ = JA + JB TO JE STEP 2 * JA
 40519 JG = JG + FNJ(JJ)
 40520 NEXT JJ
 40521 JI = (JD + 2 * JF + 4 * JG) * JA / 4
 40522 PRINT "TUSSENRESULTAAT: "; JH
 40523 IF ABS(JH - JI) > JC THEN JH = JI : GOTO 40515 : REM A
 40524 IF ABS((JH - JI) / JH) > JC THEN JH = JI : GOTO 40515 : REM B
 40525 JK = ABS (((JH - JI) / JH) * 100) - 100
 40526 IF JK > JC THEN JH = JI : GOTO 40515 : REM C
```

NOVEMBER 1982

```
34000 REM GETAL-'FORMAT'-SUBROUTINE
34001 REM DE GEBRUIKTE VARIABELEN ZIJN DA T/M DH EN DA$,DB$,DC$
34002 REM ALVORENS SUBROUTINE AAN TE ROEPEN IN HET PROGRAMMA OPGEVEN
34003 REM DA = TOTAAL AANTAL POSITIES VAN DE UITVOER MET EV. MINTEKEN
34004 REM DB = GEWENST AANTAL CIJFERS ACHTER DE KOMMA
34005 REM DC = WAARDE VAN DE VARIABELE, DIE GEPRINT MOET WORDEN
34006 REM IN PROGRAMMA NA SUBROUTINE-AANROEP 'PRINT SPC(DH);DB$' ETC.
34007 DD = 10 ^ DB : DE = 0 : IF DC < 0 THEN DE = 1 : DC = - DC
34008 DF = INT(DC + .5 / DD) : DG = INT((DC - DF) * DD + .5)
34009 DC$ = STR$(DF) : DA$ = STR$(DG) : IF DB = 0 THEN DB = 1
34010 DC$ = RIGHT$(DC$ , LEN(DC$) - 1) : DA$ = RIGHT$(DA$ , LEN(DA$) - 1)
34011 IF LEN(DA$) > DB THEN DA$ = LEFT$(DA$ , DB)
34012 IF LEN(DA$) < DB THEN DA$ = "0" + DA$ : GOTO 34012
34013 DB$ = DC$ + "." + DA$ : DH = DA + 1 - LEN(DB$)
34014 IF DE = 1 THEN DB$ = "-" + RIGHT$(DB$ , DA - 1)   DH = DH - 1
34015 RETURN
```

```
50000 REM BERG GETAL OP IN HOGE RAM
50001 SA$=STR$(SA)
50002 IFLEN(SA$)<12THENSA$=" "+SA$:GOTO50002
50003 SA=VAL(LEFT$(SA$,1))
50004 POKESC,SA+48
50005 FORSB=2TO11
50006 SA=VAL(MID$(SA$,SB,1))
50007 POKESC+SB-1,SA+48
50008 NEXTSB
50009 SA=VAL(RIGHT$(SA$,1))
50010 POKESC+11,SA+48
50011 RETURN
50012 REM GEBRUIKT WORDEN DE VARIABELENNAMEN SA, SB, SC EN SA$
50013 REM DE WAARDE VAN EEN VARIABELE WORDT MET VARIABELE SA
50014 REM NAAR DE SUBROUTINE MEEGENOMEN EN DAAR TOT STRING SA$
50015 REM OMGEWERKT DIE IN 12 BYTES WORDT OPGEBORGEN
50016 REM SC IS HET ADRES IN HOGE RAM VAN WAAR AF HET GETAL
50017 REM WORDT OPGEBORGEN
50018 REM DE SUBROUTINE-AANROEP IN HET HOOFDPROGRAMMA IS
50019 REM SC = STARTADRES (DUBBELE PUNT)
50020 REM SA = VARIABELE  (DUBBELE PUNT) GOSUB50000
50021 REM HAAL GETAL OP IN HOGE RAM
50022 SA$=""
50023 FORSB=0TO11:SA=PEEK(SB+SC):SA$=SA$+CHR$(SA):NEXTSB
50024 SA=VAL(SA$):RETURN
50025 REM GEBRUIKT WORDEN DE VARIABELENNAMEN SA,SB,SC EN SA$
50026 REM HET STARTADRES IN HOGE RAM (SC) VAN DE VARIABELE
50027 REM WORDT MEEGENOMEN NAAR DE SUBROUTINE
50028 REM DE WAARDE VAN EEN VARIABELE WORDT MET VARIABELE SA
50029 REM UIT DE SUBROUTINE MEE TERUGGEBRACHT
50030 REM SUBROUTINE-AANROEP GAAT ALS VOLGT
50031 REM SC=STARTADRES IN HOGE RAM (DUBBELE PUNT) GOSUB50021
50032 REM (DUBBELE PUNT) VARIABELE = SA
```

T R E F W O O R D E N

Heeft u dat ook ? Van die stapels met interessante papieren ? Het tot een minimum (vanwege het gewicht) beperkte stapeltje folders en brochures van de HCCdag en andere happenings, de kopieen van belangwekkende artikelen die je laatst zo graag wilde hebben, de listings die nog ingetikt moeten worden, de weliswaar ongevraagd in de bus gevallen informatie die toch van pas kan komen, ...
Wat doe je er mee ? Hoe vind je ooit iets terug ?
Heel eenvoudig: maak een nieuwe, nette stapel, het oudste onderop de meest recente dingen boven. Nummer de zaak van onder naar boven, doe bijgaand programma in de computer en voer per document een (of enkele) trefwoord(en) met bijbehorend nummer in. De stapel daarna in een verloren (maar niet vergeten) hoekje, en uw wetenschap is plotseling via uw computer bereikbaar geworden !
Het probleem dat ons clubblad geen uitgebreid register kent, kan op dezelfde manier worden opgelost.

"Trefwoorden" maakt het mogelijk alle ingevoerde, maar ook de geselecteerde informatie via een printer vast te leggen. Heeft u geen printer ? Dan kunnen (eventueel) regel 340,350,360,380,390,410,420,-740,750,760,780,790,810,820,890,960 en 993 achterwege blijven en regel 330 in GOTO 370 en regel 730 in GOTO 770 worden veranderd.
Als maximum voor het aantal trefwoorden is hier 500 gekozen (regel 40,50,70,180,580,590) maar afhankelijk van de geheugencapaciteit kan dit meer of moet dit minder zijn. Tik een bescheiden blok trefwoorden in, zet dit eerst via de SAVE-routine in regel 820 e.v. op de cassette en experimenteer dan door dit blok meermalen in te voeren. Bij een teveel gaan al gauw het programma (bijna) en de trefwoorden verloren: prettig als alles dan nog op de band staat !
Als bij uittikken een keer teveel op <RET> wordt gedrukt, "springt" de OSI uit de input-loop en geeft een "ready". Jammer als je dan niet weet dat hij met een CONT en <RET> weer "terugspringt" !
Een komma of een dubbele punt invoeren kan niet (tenzij begonnen wordt met "), want anders mikt hij alles na die komma eruit (EXTRA IGNORED). Nu bestaat er een mogelijkheid (zie blauwe FIRST OSI BOOK

p.5.4) om deze laatste twee ongerechtigheidjes te omzeilen, maar dat gaf zoveel strubbelingen (opschuiven van de machinetaalsubroutine in het geheugen) dat ik daar maar voorlopig van af heb gezien. Het programma draagt er overigens nog de sporen van: liever dubbel dan terugspringen. Voor experimenten geef ik hier het principe in BASIC:

```
10 FOR A= 4096 TO 4134: READ B: POKE A,B: NEXT
20 POKE 11,0: POKE 12,16 :A$= USR(X)
30 DATA 32,87,163,165,195,164,196,133,17,132,
40 DATA 18,169,19,160,0,133,195,132,196,133
50 DATA 92,133,91,32,180,176,165,17,164,18
60 DATA 133,195,132,196,104,76,194,0
```

Het programma zelf biedt overigens voldoende mogelijkheden om ongerechtigheden in de trefwoorden te corrigeren.
Om het opzoeken niet te lang te laten duren en om "variaties" in het trefwoord toe te laten, wordt bij het raadplegen van het bestand alleen op de eerste vijf tekens gelet (regel 1010).

Ton Helwig, Wageningen

TREFWOORDEN SEPT. 1981

```
 10 PRINTCHR$(3):POKE546,11:POKE548,0
 40 CLEAR:DIM A$(500)
 50 INPUT"Hoeveel trefwoorden laden (max.500)";N
 60 IF N=0 THEN 180
 70 IF N>500 THEN PRINT"MAXIMUM IS 500 !":GOTO50
 80 PRINT"Druk de PLAY-toets van de recorder in."
 85 FORX=0 TO 6000:NEXT:PRINT"   Daar gaat-ie !"
 90 LOAD:LOAD
 95 FORI=1TO N
130 INPUT A$(I)
140 NEXT:POKE515,0
150 INPUT"Direct bestand raadPle9en (J/N)";E$
160 PRINTCHR$(3)
170 IF E$="J" THEN 940
180 PRINT"Hoeveel trefwoorden moeten worden toe9evoe9d (max."500-N")"
190 INPUT H:IF H>500-N THEN 180
200 IF H=0 THEN PRINT CHR$(3):GOTO 440
230 FOR Y=N+1 TO N+H
240 INPUT A$(Y)
250 NEXT
260 PRINT CHR$(3):PRINT"1= eerst TOTALE inhoud Printen"
270 PRINT"2= alleen TOEVOEGING Printen"
280 PRINT"3= NIET Printen"
290 INPUT"Wat is uw keuze";B
300 IF B=1 THEN C=1:GOTO 330
310 IF B=2 THEN C=N+1:GOTO 330
```

```
320 IF B=3 THEN 440
330 INPUT"Eerst Printer aanzetten, type dan P <RET>";Z$
340 IF Z$<>"P" THEN 330
350 INPUT"Wat is de datum van vandaag";C$:PRINT CHR$(3)
360 SAVE:PRINTCHR$(14)
370 PRINT"   T R E F W O O R D E N L I J S T"
380 PRINTCHR$(15)CHR$(10)"                              PER:"C$:C$=""
390 PRINTCHR$(10)
400 FOR I=C TO N+H
410 IF LEN(A$(I))>34 THEN PRINT I" " A$(I):NEXT:GOTO440
420 IF I/2-INT(I/2)=0 THENPRINTTAB(38)I" "A$(I):NEXT:GOTO440
430 PRINT I" "A$(I);:NEXT
440 POKE517,0:PRINT:PRINT
450 N=N+H:INPUT"Hoeveel trefwoorden moeten er worden VERANDERD ";F
460 IF F=0 THEN 560
470 FOR T=1 TO F
480 PRINT"Trefwoord "T",Wat is het volgordenummer";
490 INPUT U:PRINT CHR$(3)
500 PRINT"OUD= "A$(U)
510 PRINT"Wat is de nieuwe tekst ?"
540 INPUT A$(U):NEXT T
560 PRINTCHR$(3)
570 PRINT"    Hoeveel trefwoorden moeten er alsnog ";
580 PRINT"         worden toegevoegd (max."500-N")";
590 INPUT H:IF H>500-N THEN 560
600 IF H=0 THEN PRINT CHR$(3):GOTO 660
630 FOR Y=N+1 TO N+H
640 INPUT A$(Y)
650 NEXT
660 PRINT CHR$(3):PRINT"1= eerst TOTALE inhoud Printen"
670 PRINT"2= alleen TOEVOEGING Printen"
680 PRINT"3= NIET Printen"
690 INPUT"Wat is uw keuze";B
700 IF B=1 THEN C=1:GOTO 730
710 IF B=2 THEN C=N+1:GOTO 730
720 IF B=3 THEN 840
730 INPUT"Eerst Printer aanzetten, type dan P <RET>";Z$
740 IF Z$<>"P" THEN 730
750 PRINTCHR$(3):INPUT"Wat is de datum van vandaag";C$
760 SAVE:PRINTCHR$(14)
770 PRINT"   T R E F W O O R D E N L I J S T"
780 PRINTCHR$(15)CHR$(10)"                              PER:"C$:C$=""
790 PRINTCHR$(10)
800 FOR I=C TO N+H
810 IF LEN(A$(I))>34 THEN PRINT I" "A$(I):NEXT:GOTO840
820 IF I/2-INT(I/2)=0 THENPRINTTAB(38)I" "A$(I):NEXT:GOTO840
830 PRINT I" "A$(I);:NEXT
840 PRINT:PRINT:PRINT:PRINT:PRINT:POKE517,0
845 N=N+H
850 PRINT"1= direct SAVE'n"
860 PRINT"2= eerst bestand RAADPLEGEN"
870 INPUT G:PRINTCHR$(3):ON G GOTO 880,950
880 PRINT"Is cassetterecorder klaar voor opname "
890 PRINT"en Printer uit ?"
900 INPUT"Type dan P en <RET>";D$
910 FOR X=0 TO 500:NEXT
```

```
920 NULL5:SAVE:FORD=1TO N:PRINTA$(D):NEXT
930 POKE517,0
940 PRINTCHR$(3)
950 PRINT"Indien cijfer VOOR trefwoord getypt:"
960 PRINT"1= selectie tevens Printen"
970 PRINT"2= bestand naar cassette "
980 PRINT"3= trefwoord veranderen"
985 PRINT"4= trefwoord(en) toevoegen"
990 INPUT" OP te zoeken trefwoord ";E$
993 IF VAL(E$)=1 THEN SAVE:E$=RIGHT$(E$,(LEN(E$)-1))
995 IF VAL(E$)=2 THEN 880
997 IF VAL(E$)=3 THEN 1035
998 IF VAL(E$)=4 THEN 560
1000 FOR E=1 TO N
1010 IF LEFT$(A$(E),5)=LEFT$(E$,5)THEN PRINTA$(E)
1020 NEXT:POKE517,0
1030 PRINT:PRINT:GOTO 950
1035 PRINT CHR$(3)
1040 INPUT "Hoe luidt te veranderen trefwoord Precies";F$
1050 FOR E=1 TO N
1060 IF F$<>A$(E) THEN NEXT:PRINT:PRINT"Niet gevonden !":GOTO 950
1070 INPUT"Type nieuwe tekst";A$(E):GOTO 900
```

MACHINETAAL IN BASIC DATA-STATEMENTS.

Inleiding

Het doel van bijgaand programma is wat te besparen op omrekenen typewerk bij het maken van BASIC programma's welke machinetaal subroutines gebruiken (X=USR(X)).
Dit programma genereert de DATA-statements,waarin het machinetaal programma staat (decimaal), alsmede de bijbehorende inleesroutine.
De invoer bestaat uit: - het startadres van het ML-programma (4 bytes HEX)
- het ML-programma zelf (steeds 2 bytes HEX gevolgd door return)

De invoer wordt beeindigd door het geven van een decimale punt.

Programma opbouw

regelnummer:

100-112	gebruiksaanwijzing
115	wachten op ESC toets
150-165	invoer en omzetting ML startadres
170-220	Idem voor ML-programma
240	regelnummer eerste datastatement (evt. zelf te wijzigen!)
251-340	uitvoer naar cassette van datastatements
400-420	Idem van inleesroutine
1000-1055	HEX-DEC omzetting

NB regel 270 voorkomt fouten door onnauwkeurigheid bij het weer loaden van de statements, kan ook worden weggelaten.

H.V.Coenen
tel. 070-994566

```
100 ?"Dit programma zet een in HEX in te voeren":?
105 ?"machinetaalprogramma om in basic datastatements":?
107 ?"Deze worden direct op een leeg stuk cassetteband":?
109 ?"geladen. Voer alles in HEX in!":?
110 ?"- Eindig datainvoer met '.' ":?
112 ?"- Druk nu op ESC."
115 IF PEEK(57100)<>222THEN115
130 DIM D(500)
140 POKE 11,6: POKE 12,254:X=USR(X)
145 ?CHR$(2);
150 ?:INPUT" Startadres:";S$
160 X$=S$: GOSUB 1000
165 SA=D(0)
170 REM DATA INVOER
180 N=1
```

```
190 INPUT "$";X$
200 IF X$="." THEN 230
210 GOSUB 1000
220 N=N+1:GOTO 190
230 REM UITVOER
240 RG=2000:Z=N-1:N=1
244 X=USR(X):?CHR$(2);
246 ?"- Plaats lege cassette en zet rec. op opname":?
250 ?"- Druk nu weer op ESC.":?
251 NULL4:SAVE
252 IF PEEK(57100)<>222 THEN252
270 ?"1999 REM SYNCHRONISATIE!!"
300 ? RG;"DATA";
310 FOR J=1 TO 10
312 IF J=1 THEN ?D(N);:GOTO320
314 ?",";D(N);
320 N=N+1:IF N>Z THEN 400
330 NEXT J
340 ?:?RG+10:GOTO 300
400 ?:?RG+10"FOR P=";SA;"TO";SA+Z-1
410 ?RG+20;"READD:POKE P,D: NEXT":?CHR$(13)
420 LOAD:POKE 515,0
430 END
1000 REM HEXGETAL=X$:DECIMAAL GETAL=D(N)
1010 L=LEN(X$)
1015 IF L=2 THEN X$="00"+X$
1020 FOR I=4 TO 1 STEP -1
1025 X$(I)=MID$(X$,I,1)
1030 IF X$(I)<"A" THEN1040
1035 F(I)=ASC(X$(I))-55:GOTO1045
1040 F(I)=VAL(X$(I))
1045 NEXT I
1050 D(N)=F(1)*4096+F(2)*256+F(3)*16+F(4)
1055 RETURN
```

```
100 REM ******* STRING VAN SCHERM *************
101 REM
102 REM   met deze routine kun je de input verleggen naar scherm
104 REM   lege input of input met komma's is ook mogelijk
106 REM   Men kan op elke plaats op het scherm beginnen en
108 REM   dan alles invoeren tot het eind van de regel, niet verder
110 REM
111 REM   Zoals de routine nu is, maakt elk controleteken een einde
113 REM   aan de invoer. Dit is niet nodig: men kan hier ook anders
115 REM  laten werken.
116 REM
117 REM   Het is wenselijk dat men IN$ als eerste variabele
119 REM   declareert: regels 1030/1040 kunnen dan vervallen
121 REM
123 REM   en er wordt iets vlugger gewerkt. Doet men het niet,
125 REM   dan moet men er voor zorgen dat alle variabelen die in de
127 REM   routine voorkomen, gedeclareerd zijn, zoals in regel 510
129 REM   ------------ ----------------------
131 REM   Op regel 2000 en volgende staat een gepackte versie
133 REM   Daarbij is er van uit gegaan dat IN$ de eerste variabele is
135 REM   en dat VT een keer in het hoofdprogramma is berekend
199 REM   ------------------------------ ---
500 IN$="":CL=CH=SL=VT=KEY     :REM GEBRUIKTE VARIABELEN
510 VT=PEEK(123)+256*PEEK(124) :REM  ADRES VARIABELENTABEL
520 PR$=CHR$(7)+CHR$(235)
530 GOSUB 1000:PRINT:PRINTIN$,"LENGTE"LEN(IN$)
540 GOSUB 2000:PRINT:PRINTIN$,"LENGTE"LEN(IN$)
550 GOTO530
1000 REM ------------------------------------
1002 REM     STRING VAN SCHERM ALS INPUT
1004 REM ------------------------------------
1006 REM    zoek IN$  in variabelentabel
1008 REM ------------------------------------
1010 REM
1020 VT=PEEK(123)+256*PEEK(124)
1030 IFPEEK(VT)=ASC("I")ANDPEEK(VT+1)=ASC("N")+128THEN1050
1040 VT=VT+6:GOTO1030
1045 REM ------------------------------------
1050 REM    IN$ gevonden in variabelentabel
1055 REM ------------------------------- --
1057 REM schrijf CURSORPOSITIE als begin IN$
1059 REM ------------------ -------- ---- -----
1060 PRINT";";
1070 CL=PEEK(556):POKEVT+3,CL
1080 CH=PEEK(557):POKEVT+4,CH
1090 SL=PEEK(553)     :
1095 REM ------------------------------------
1097 REM    schrijf op scherm via USR($FD00)
1099 REM ------------------------------------
1100 POKE11,0:POKE12,253
1110 KEY=USR(KEY):KEY=PEEK(531)
1120 IFKEY)31THENPRINTCHR$(KEY);:GOTO1110
```

```
1124 REM ------------------------------------
1126 REM    bereken de lengte van de invoer
1127 REM    en schrijf die in de var.tabel
1128 REM ------------------------------------
1130 SL=63AND(PEEK(553)-SL)
1140 POKE VT+2,SL
1144 REM ------------------------------------
1145 REM    copieer van scherm naar stringram
1146 REM ------------------------------------
1150 IN$=IN$+""
1160 RETURN
1199 REM
2000 REM ************************************
2001 REM *                                  *
2002 REM *   STRING VAN SCHERM  IN$         *
2003 REM *                                  *
2004 REM ************************************
2005 REM
2010 PRINTPR$;:CL=PEEK(556):CH=PEEK(557):POKEVT+3,CL:POKEVT+4,CH
2020 SL=PEEK(553):POKE11,0:POKE12,253
2030 KEY=USR(KEY):KEY=PEEK(531):IFKEY)31THENPRINTCHR$(KEY);:GOTO20
2040 SL=63AND(PEEK(553)-SL):POKEVT+2,SL:IN$=IN$+"":RETURN
```

Peter Broers

```
10000 REM SCHETS EN SCROLL
10001 REM MAAK EEN SCHETS OP HET SCHERM MET CHR$(161)
10002 REM EN LAAT DE SCHETS HORIZONTAAL SCROLLEN
10003 REM J.HERMANS SCHIPLUIDEN
10004 REM JULI 1981
10005 READR:IFR()456THEN10005
10008 GOSUB10770
10010 INPUT"INFORMATIE GEWENST (J/N) ";A$
10020 IFLEFT$(A$,1)="J"THENGOSUB10660
10030 PRINTCHR$(26)
10040 P=53711:N=187:POKEP,N:M=160:O=32
10050 J=200
10060 POKE530,1:POKE57088,127
10070 Z=PEEK(57088)
10080 IFZ=255THENPOKE530,0:GOTO10060
10090 Z=8-LOG(255-Z)/LOG(2)
10100 ONZGOSUB10120,10260,10370,10500,10500,10610,10520
10110 GOTO10060
10120 Q=PEEK(P):IFQ=MTHENS=1
10130 T=PEEK(P+1):IFT=MTHENU=1
10140 POKEP,O
10150 P=P+1
10160 FORI=0TO1792STEP64
10170 IFP=53561+ITHENR=1:I=1792
10180 NEXTI
10190 IFR=1THENR=0:P=P-1:POKEP,N:RETURN
10200 IFS=1THENS=0:POKEP-1,M
10210 IFU=1THENU=0:POKEP,M:GOTO10240
10220 POKEP,N
10230 J=200
10240 GOSUB10650
10250 RETURN
10260 Q=PEEK(P):IFQ=MTHENS=1
10270 T=PEEK(P+64):IFT=MTHENU=1
10280 POKEP,O
10290 P=P+64
10300 IFP)55291THENP=P-64:POKEP,N:RETURN
10310 IFS=1THENS=0:POKEP-64,M
10320 IFU=1THENU=0:POKEP,M:GOTO10350
10330 POKEP,N
10340 J=300
10350 GOSUB10650
10360 RETURN
10370 Q=PEEK(P):IFQ=MTHENS=1
10380 T=PEEK(P-1):IFT=MTHENU=1
10390 POKEP,O
10400 P=P-1
10410 FORI=0TO1792STEP64
10420 IFP=53513+ITHENR=1:I=1792
10430 NEXTI
10440 IFR=1THENR=0:P=P+1:POKEP,N:RETURN
```

```
10450 IFS=1THENS=0:POKEP+1,M
10460 IFU=1THENU=0:POKEP,M:GOTO10480
10470 POKEP,N
10480 GOSUB10650
10490 RETURN
10500 Q=PEEK(P):IFQ=MTHENS=1
10510 T=PEEK(P-64):IFT=MTHENU=1
10520 POKEP,0
10530 P=P-64
10540 IFP<53257THENP=P+64:POKEP,N:RETURN
10550 IFS=1THENS=0:POKEP+64,M
10560 IFU=1THENU=0:POKEP,M:GOTO10580
10570 POKEP,N
10580 GOSUB10650
10590 RETURN
10600 POKEP,M:GOSUB10650:RETURN
10610 POKEP,0:RETURN
10620 POKE57088,253
10630 IFPEEK(57088)=239THENRETURN
10640 GOTO10800
10650 FORI=1TOJ:NEXT:RETURN
10660 PRINTCHR$(26)
10670 PRINT"TOETS 1 = NAAR RECHTS":PRINT
10680 PRINT"      2 =       BENEDEN":PRINT
10690 PRINT"      3 =       LINKS":PRINT
10700 PRINT"      4 =       BOVEN":PRINT
10710 PRINT"      5 = ZET WIT VLAKJE":PRINT
10720 PRINT"      6 = WIS WIT VLAKJE":PRINT
10730 PRINT"      7 = ZET HOR. SCROLL IN WERKING"
10740 PRINT:PRINT"            (SPATIEBALK STOPT SCROLL)"
10750 PRINT:PRINT:PRINT"VOOR VERVOLG DRUK OP (ESC)"
10760 WAIT57088,32,254:PRINTCHR$(26):RETURN
10770 P=560
10780 READR:IFR<0THENRETURN
10790 POKEP,R:P=P+1:GOTO10780
10800 POKE11,48:POKE12,2
10810 A=USR(A)
10820 FORI=1TO2*J:NEXTI
10830 GOTO10620
10835 DATA456
10840 DATA169,0,133,224,133,226,169,1,133,228,169,208
10850 DATA133,225,169,208,133,227,133,229,133,231,169,63
10860 DATA133,230,169,80,133,232,169,0,133,233,162,0
10870 DATA160,4,161,224,129,232,230,232,24,169,64,101,224
10880 DATA133,224,136,208,240,160,4,169,64,133,234
10890 DATA161,228,129,226,230,226,230,228,198,234,208,244
10900 DATA24,169,0,101,226,133,226,24,169,0,101,228,133,228
10910 DATA136,208,223,169,80,133,232,160,4,161,232
10920 DATA129,230,230,232,24,169,64,101,230,133,230,136
10930 DATA208,240,230,225,230,227,230,229,230,231,169,216
10940 DATA197,225,208,157,96
10950 DATA-1
```

PUZZLE-KUBUS. (ALLEEN VOOR 48/64 KARAKTERS PER REGEL)

DIT PROGRAMMA IS EEN SIMULATIE VAN
RUBIK'S CUBE PUZZLE
DE KUBUS WORDT GEPROJECTEERD OP HET SCHERM IN OPENGEVOUWEN
TOESTAND. DE ACHTERZIJDE STAAT HELEMAAL RECHTS. DE VOORZIJDE
STAAT IN HET KRUISPUNT.
U WEET WAT DE BEDOELING IS.
DIT PROGRAMMA WERKT MET DE SUPPORT-PROM VERSIES 3.7 UP.
ZONDER DIE PROM WORDT HET ERG MOEILIJK DIT PROGRAMMA TE
LATEN LOPEN.
OMDAT HET PRINTEN VAN DE VERRICHTE BEWEGINGEN EEN SCROLL KAN
GEVEN VAN DE FIGUUR. WAT U DOET, MOET DAN DUS GEPOKED WORDEN ONDER
DE FIGUUR OF WORDEN WEGGELATEN (KLADPAPIERTJE).
HET PROGRAMMA GEEFT U NIET DE OPLOSSING VOOR DE PUZZLE.
====
LOAD-OMVANG: 5100 BYTES; RUN-OMVANG: 5600 BYTES.

```
 1000 POKE11,6:POKE12,254:A=USR(A)
 1010 POKE546,12:POKE548,3
 1020 REM PUZZLE-KUBUS
 1030 REM J. HERMANS SCHIPLUIDEN
 1040 REM DECEMBER 1980
 1050 GOTO1670
 1060 A=PEEK(V(2,0,0)):B=PEEK(V(2,2,0))
 1070 POKEV(2,0,0),B:B=PEEK(V(2,2,2))
 1080 POKEV(2,2,0),B:B=PEEK(V(2,0,2)):POKEV(2,2,2),B:POKEV(2,0,2),A
 1090 A=PEEK(V(2,0,1)):B=PEEK(V(2,1,0))
 1100 POKEV(2,0,1),B:B=PEEK(V(2,2,1))
 1110 POKEV(2,1,0),B:B=PEEK(V(2,1,2)):POKEV(2,2,1),B:POKEV(2,1,2),A
 1120 A=PEEK(V(0,0,2)):B=PEEK(V(3,0,0))
 1130 POKEV(0,0,2),B:B=PEEK(V(4,2,0))
 1140 POKEV(3,0,0),B:B=PEEK(V(1,2,2)):POKEV(4,2,0),B:POKEV(1,2,2),A
 1150 A=PEEK(V(0,1,2)):B=PEEK(V(3,0,1))
 1160 POKEV(0,1,2),B:B=PEEK(V(4,1,0))
 1170 POKEV(3,0,1),B:B=PEEK(V(1,2,1)):POKEV(4,1,0),B:POKEV(1,2,1),A
 1180 A=PEEK(V(0,2,2)):B=PEEK(V(3,0,2))
 1190 POKEV(0,2,2),B:B=PEEK(V(4,0,0))
 1200 POKEV(3,0,2),B:B=PEEK(V(1,2,0)):POKEV(4,0,0),B:POKEV(1,2,0),A
 1210 IFC>0GOTO1240
 1220 PRINT"1";:W=W+1:IFW=5THENW=0:PRINT" ";
 1230 FORX=1TO1000:NEXTX
 1240 RETURN
 1250 A=PEEK(V(1,0,0)):B=PEEK(V(1,2,0))
 1260 POKEV(1,0,0),B:B=PEEK(V(1,2,2))
 1270 POKEV(1,2,0),B:B=PEEK(V(1,0,2)):POKEV(1,2,2),B:POKEV(1,0,2),A
 1280 A=PEEK(V(1,0,1)):B=PEEK(V(1,1,0))
 1290 POKEV(1,0,1),B:B=PEEK(V(1,2,1))
 1300 POKEV(1,1,0),B:B=PEEK(V(1,1,2)):POKEV(1,2,1),B:POKEV(1,1,2),A
 1310 FORI=0TO2:FORJ=0TO2
 1320 A=PEEK(V(0,I,J)):B=PEEK(V(2,I,J))
 1330 POKEV(0,I,J),B:B=PEEK(V(4,I,J))
 1340 POKEV(2,I,J),B:B=PEEK(V(5,I,J)):POKEV(4,I,J),B:POKEV(5,I,J),A
 1350 NEXTJ:NEXTI
```

```
3500 PRINTCHR$(3)
3510 PRINTTAB(12);"FINAL FLIGHT STATUS":PRINT:PRINT
3520 PRINTTAB(12);"TIME OF FLIGHT: ";INT(10*T4/6)/100;" MIN"
3530 PRINTTAB(12);"FUEL LEFT: ";INT(F9*2.2);" LBS"
3540 PRINTTAB(12);"FINAL SPEED: ";INT(V/P6);" K/H"
3550 PRINTTAB(12);"FINAL HORIZON: ";INT((D3+T1)*P4);" DEG"
3560 PRINT:PRINTTAB(12);"FINAL E-COORD. : ";INT(X1*100/P5)/100
3570 PRINTTAB(12);"FINAL N-COORD. :";INT(X2*100/P5)/100
3580 PRINT:PRINTTAB(12);"TRY AGAIN ? (Y/N)"
3590 FORI=1TO8:PRINT:NEXTI
3600 POKE530,1:POKE57088,239
3610 IFPEEK(57088)=247THENQA=1:POKE530,0:GOTO1060
3620 POKEEE,251
3630 IFPEEK(EE)=247THENPOKE530,0:GOTO3650
3640 POKE530,0:GOTO3600
3650 PRINT:PRINTTAB(12);"GOODBYE. COME AGAIN SOON."
3660 POKE546,9:POKE547,56:POKE548,4:POKE549,29:END
```

DOOR ALS EXTRA REGEL IN TE LASSEN:

```
2935 RETURN
```

WORDT DE LANDING UITGEVOERD BIJ WINDSTILTE.
VOOR '542 POLLED KEYBOARD' DIENEN DE VOLGENDE WIJZIGINGEN TE WORDEN AANGEBRACHT:

```
1160 E1=2:E2=128:E3=2:E4=4:E5=8:E6=16:E7=32:EE=57088
1170 E9=128:EF=54436:T3=3:YI=0
1240 253 WORDT 2 EN 239 WORDT 16
3600 251 WORDT 4
3610 239 WORDT 16
```

```
10 REM WEEFPATRONEN
20 REM JOHN HERMANS
30 REM ZIE OOK 'BYTE' VAN SEPTEMBER 1982
40 REM PAUL W. HEISER - A WEAVING SIMULATOR
50 REM SEPTEMBER 1982
100 READRK,RR,K,I
110 DIM A(RR,RK),B(K),C(I)
120 FORL=1TORR:FORM=1TORK:READA(L,M):NEXTM,L
130 READNK:FORL=1TOK:READB(L):NEXTL
140 READNI:FORL=1TOI:READC(L):NEXTL
150 FORL=1TONI
160 FORM=1TOK
170 F$=""
180 FORN=1TONK
190 FORO=1TOI
200 ONA(B(M),C(O))+1GOTO210,220
210 F$=F$+" ":GOTO230
220 F$=F$+"#"
230 NEXTO,N
240 PRINTF$
250 NEXTM,L
300 DATA5,5,19,20
310 DATA1,1,0,0,1
320 DATA1,0,0,1,0
330 DATA0,0,1,0,1
340 DATA0,1,0,1,1
350 DATA1,0,1,1,0
370 DATA2
380 DATA1,2,3,4,5,1,2,3,4,5
390 DATA1,2,3,4,5,4,3,2,1
400 DATA2
410 DATA1,2,3,4,5,1,2,3,4,5
420 DATA5,4,3,2,1,1,2,3,4,5
```

HANS COENEN: DOBBELSTEEN

```
SCR # 32
  0 ( DOBBELSTEEN)
  1 : BOVEN   53988 53981 DO 128 I C! LOOP ;
  2 : ONDER   54372 54365 DO 135 I C! LOOP ;
  3 : LINKER  54364 54044 DO 143 I C! 64 +LOOP ;
  4 : RECHTER 54372 54052 DO 136 I C! 64 +LOOP ;
  5 : BODY BOVEN ONDER LINKER RECHTER ;
  6 : EEN  BODY 226 54176 C! ;
  7 : TWEE BODY 226 54114 C! 226 54238 C! ;
  8 : DRIE EEN TWEE ;
  9 : VIER TWEE 226 54110 C! 226 54242 C! ;
 10 : VIJF EEN VIER ;
 11 : ZES VIER  226 54112 C! 226 54240 C! ;
 12 -->
 13
 14
 15
OK
```

```
SCR # 33
  0 ( DOBBELSTEEN VERVOLG)
  1 : REST DUP 4 = IF VIER ELSE DUP 5 = IF VIJF
  2                  ELSE ZES DROP THEN THEN ;
  3   TEST DUP 1 = IF EEN  ELSE DUP 2 = IF TWEE
  4                  ELSE DUP 3 = IF DRIE
  5                  ELSE REST THEN THEN THEN ;
  6   CIJFER 6 CHOOSE TEST ;
  7   DISPLAY CLS CIJFER SP! ;
  8 . DELAY 0 DO LOOP ;
  9 : WERP 20 0 DO DISPLAY 100 DELAY LOOP ;
 10 : REGEL CLS ." WERP: SPATIE, STOP: ANDERE TOETS" ;
 11 0 VARIABLE FF1  : RESET 0 FF1 ! ;
 12 : DOBBEL REGEL BEGIN KEY 32 = IF WERP
 13        ELSE 1 FF1 ! THEN FF1 @ UNTIL ;
 14 : DOBBEL RESET DOBBEL ;
 15 ;S
OK
```

HANS COENEN: SIMPLE GAME

```
SCR # 46
  0 ( SIMPLE GAME )   : DELAY 0 DO LOOP ;
  1   DOELEN 50 0 DO 240 54730 I + C! 5 +LOOP ;
  2   TEKST1  " EENVOUDIG SCHIETSPEL, BEWEEG MET" CR
  3   TEKST2 ." LINKER EN RECHTER SHIFT.  SCHIET MET REPEAT" CR
  4 . TEKST3 ." EN STOP MET SHIFT-LOCK UP " CR 25000 DELAY CLS
  5 : TEKST TEKST1 TEKST2 TEKST3 ;
  6 0 VARIABLE FLAG  0 VARIABLE POS
  7 : SCHERM CLS CR CR CR TEKST DOELEN ;
  8 1 POS !  0 FLAG !
  9 : STAPUN POS @ 53580 + 64 + ;
 10 : EINPUN STAPUN 20 66 * + ;
 11 : VUUR EINPUN STAPUN DO 198 I C! 100 DELAY 32 I C! 66 +LOOP
 12 : DOELTEST EINPUN C@ 240 = IF 32 EINPUN C! THEN ;
 13 : SCHIETEN VUUR DOELTEST ;
 14 -->
 15
OK


SCR # 47
  0 ( VERVOLG SIMPLE GAME)    TREKAF POS @ 1 - POS ! ;
  1 . TELOP 1 POS +! ;
  2 : ZET   POS @ 53580 + C! ;
  3 : WIS   32 ZET ;
  4 : VOOR  WIS TELOP 237 ZET ;
  5   ACHT  WIS TREKAF 239 ZET ;
  6   STOP 1 FLAG ! ;
  7   A DUP 126 = IF SCHIETEN ELSE DUP 255 = IF STOP THEN THEN
  8   B DUP 250 = IF ACHT ELSE DUP 252 = IF VOOR ELSE A DROP
  9     THEN THEN ;
 10 : GETKEY 57100 C@ ;
 11 : ENTER SP! GETKEY B ;
 12 : GAME 0 FLAG ! SCHERM BEGIN 100 DELAY ENTER FLAG @ UNTIL
 13 ;S
 14
 15
OK
```

**** M I N I — L O G O ****

MINI-LOGO IS GESCHREVEN IN BASIC.
HET HERKENT EN VERWERKT EEN AANTAL LOGO-ACHTIGE OPDRACHTEN
IN DE VORM VAN EEN PROGRAMMA MET REGELNUMMERS.
DE OPDRACHTEN BEWEGEN DE 'TURTLE' (SCHILDPAD, HIER EEN PUNT)
OVER HET SCHERM, WAARBIJ HIJ EEN SPOOR VAN DE OPGEGEVEN
KLEUR ACHTERLAAT.

DIE OPDRACHTEN ZIJN

MET REGELNUMMER:

- HERHAAL <AANTAL>: HERHAALT DE OPDRACHTEN, DIE STAAN TUSSEN HERHAAL EN TOT HIER HET OPGEGEVEN AANTAL MALEN
- TOT HIER
- STAP <AANTAL>: DE 'TURTLE' DOET HET OPGEGEVEN AANTAL STAPPEN OVER HET SCHERM IN DE AANGEGEVEN RICHTING
- DRAAI <AANTAL>: DE BEWEGINGSRICHTING WORDT OVER EEN HOEK, DIE GELIJK IS AAN AANTAL * PI / 4 MET DE WIJZERS VAN HET UURWERK MEE GEDRAAID
- KLEUR <WAARDE>: WAARDE IS DE ASCII-WAARDE VAN HET KARAKTER OF DE WAARDE VAN HET GRAFISCHE TEKEN, DAT OP HET SCHERM GEPRINT WORDT. 'KLEUR 0' PRINT SPATIES.

ZONDER REGELNUMMERS

- DOE: HEEFT HETZELFDE GEVOLG ALS <RUN> IN BASIC
- LIJST: GEEFT LISTING OP HET SCHERM
- PRINT: GEEFT LISTING VIA PRINTER
- STOP: HERSTART MINI-LOGO
- NIEUW: HEEFT HETZELFDE GEVOLG ALS<NEW> IN BASIC

```
990 PRINTCHR$(26):PRINT:PRINT:PRINT:PRINT
1000 LG$="****              M I N I - L O G O       ****":PRINTLG$
1005 REM *** M. BAETEN 1983 ***
1010 PRINT:INPUT"JE NAAM  :";AU$
1020 PRINT:INPUT"DATUM    :";DT$
1030 L1$="-----------------------------------------------":PRINTL1$
1040 GOSUB10000:REM INITIALISATIE
1500 GOSUB11000:IFA$=""THEN1500
1510 IFVAL(A$)THENGOSUB3000:GOTO1500
1520 IFASC(A$)=ASC("L")THENGOSUB5000
1530 IFASC(A$)=ASC("P")THENGOSUB2000
1540 IFASC(A$)=ASC("D")THENGOSUB6000:PRINTCHR$(2)
1970 IFA$="STOP"THENRUN
1980 IFA$="NIEUW"THENNL=0
1990 GOTO1500
```

```
2000 NULL8:POKE517,1:REM LISTING NAAR PRINTER
2010 GOSUB5000
2020 POKE517,0
2030 RETURN
2500 LN=VAL(A$):L=LEN(A$)
2510 IFASC(A$)<65THENL=L-1:IFLTHENA$=RIGHT$(A$,L):GOTO2510
2520 IFL=0THENA$=""
2530 LN$=RIGHT$("    "+STR$(LN),3)+".  "
2540 RETURN
3000 REM SCHRIJF A$ IN ARRAY A$(NL)
3010 GOSUB2500:REM LN=WAARDE,A$=TEKST
3020 IFLN<1ORLN>999THENERR=1:GOTO9990
3030 OG=1:BG=NL+1
3040 MI=INT((OG+BG)/2):IFOG>=BGTHEN3080
3050 IFLN<VAL(A$(MI))THENBG=MI:GOTO3040
3060 IFLN>VAL(A$(MI))THENOG=MI+1:GOTO3040
3070 GOTO3120
3080 IFA$=""THENRETURN
3090 NL=NL+1:CD=NL
3100 IFCD>OGTHENA$(CD)=A$(CD-1):CD=CD-1:GOTO3100
3110 MI=OG
3120 IFA$>""THENA$(MI)=LN$+A$:RETURN
3130 CD=MI
3140 IFCD<NLTHENA$(CD)=A$(CD+1):CD=CD+1:GOTO3140
3150 NL=NL-1:RETURN
5000 PRINTCHR$(26):REM LISTING OP HET SCHERM
5005 PR$=L1$:GOSUB5090
5010 PR$=LG$:GOSUB5090
5020 IFLEN(AU$)+LEN(DT$)<46THENAU$=AU$+", ":GOTO5020
5030 PR$=AU$+DT$:GOSUB5090
5040 PR$=L1$:GOSUB5090
5050 CL=0
5060 CL=CL+1:IFCL>NLTHENPR$=L1$:GOTO5090
5070 PR$=A$(CL):GOSUB5090
5080 GOTO5060
5090 PRINTPR$:RETURN
6000 REM MINI-LOGO COMPILER EN  EXEC.
6001 PRINTCHR$(26)
6005 X=0:Y=0:DR=2
6010 NC=0:CC=0:FORCL=1TONL:W1=1:A$=A$(CL)
6030 IFMID$(A$,W1,1)<>"/"THEN6060
6040 CO$=LEFT$(A$,W1):GOSUB7000
6045 GOSUB6090
6050 A$=MID$(A$,W1):W1=2
6060 W1=W1+1:IFW1<LEN(A$)THEN6030
6070 CO$=A$:GOSUB7000:NEXTCL
6090 ON(CC)GOSUB6120,6150,6190,6210,6300
6100 CC=CC+1:IFCC<=NCTHEN6090
6110 RETURN
6120 REM  ---------- HERHAAL -----------
6130 NI=NI+1:BE(NI)=CC:TE(NI)=WA(CC)
6140 RETURN
6150 REM    TOT HIER .(EINDE HERHALING)
6160 TE(NI)=TE(NI)-1:IFTE(NI)>0THENCC=BE(NI):RETURN
6170 NI=NI-1:IFNI>0THEN6160
6180 RETURN
```

```
6190 REM  ---------- KLEUR ------------
6200 KL=WA(CC):RETURN
6210 REM  ---------- STAP -------------
6220 WA=WA(CC)
6230 IFWA=0THENRETURN
6240 X=X+XI(DR):Y=Y+YI(DR)
6270 REM  TEKEN
6280 WA=WA-1:GOTO6230
6290 RETURN
6300 REM  ---------- DRAAI ------------
6310 DR=DR+WA(CC)
6320 IFDR>7THENDR=DR-8
6330 RETURN
7000 REM  ------ INTERPRETEER ---------
7010 L=LEN(CO$):IFCO$<"A"THENIFL>1THENCO$=MID$(CO$,2):GOTO7010
7020 CO=0:FORT=1TOLEN(SET$):IFASC(CO$)=ASC(MID$(SET$,T))THENCO=T:T=100
7030 NEXTT
7040 IFCO=0THENERR=2:GOTO9990
7050 NC=NC+1:CO(NC)=CO
7060 FORT=1TOLEN(CO$):WA(NC)=VAL(MID$(CO$,T)):IFWA(NC)THENT=100
7070 NEXTT
7080 RETURN
9000 REM  ---------- TEKEN ------------
9011 IFX<XOORX>XMTHENRETURN
9012 IFY<YOORY>YMTHENRETURN
9020 IFKLTHEN9040
9030 RETURN
9040 IFKL<32THENKL=161
9050 POKE14,1:POKE553,X+26:POKE554,Y+12:PRINTCHR$(10);
9060 PRINTCHR$(KL);:RETURN
9990 REM      FOUTMELDING
9991 ERR$(1)="REGELNUMMER"
9992 ERR$(2)="ONBEKENDE OPDRACHT"
9999 PRINTERR$(ERR)" FOUT !":GOTO1500
10000 REM  INITIALISATIE
10005 REM ----------------
10010 A$=AU$:BG=CC=CL=CO:CO$=D$:CV=DR
10020 FL$=DT$:ER=KL:IN$=", ":POKE11,87:POKE12,163
10030 LN$="":MI=NC=NI=NL=OG:PR$=SE$
10040 T=W1=WA:X=XI=XM:Y=YI=YM
10050 DIMA$(50),BE(10),CO(100),TE(10),XI(7),YI(7),WA(100)
10060 FORDR=0TO7:READXI(DR):NEXT
10070 FORDR=0TO7:READYI(DR):NEXT
10080 DATA0,1,1,1,0,-1,-1,-1
10090 DATA-1,-1,0,1,1,1,0,-1
10100 XM=27:YM=15:XO=-23:YO=-12
10110 SET$="HTKSD"
10200 RETURN
11000 PRINTIN$;:IN=USR(IN):A$=""
11010 FORI=1TO72:A=PEEK(I+18):IFA=0THEN11030
11020 A$=A$+CHR$(A):NEXTI
11030 RETURN
```

HET PRINTEN VAN EIGEN KARAKTERS

Dit kan leuke resultaten geven.

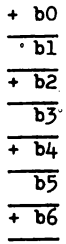

Waterman


Vissen


Ram


Stier

Op ruitjespapier maakt men, door vakjes op te vullen, het gewenste karakter. Deze tekening moet men nu in een programma opnemen. Daartoe wordt de tekening in stroken verdeeld (voor mijn printer worden dit stroken van 7 vakjes). De ingevulde vakjes worden nu verticaal gelezen en er wordt de binaire waarde aan toegekend (zie voorbeeld).



(b7) b6 b5 b4 b3 b2 b1 b0
1 1 0 1 0 1 0 1

Voor deze 7 bits wordt nu als achtste bit een 1 geplaatst, zodat de code loopt van 128 t/m 255.

Tot slot een klein voorbeeldprogramma van het onderstaande karakter:

```
10 DATA 255,129,129,249,137,137,137,137,249,129
20 DATA 129,129,129,129,129,129,129,129,255,128
30 DATA 255,128,128,255,128,128,240,144,255,144
40 DATA 144,176,160,224,160,160,128,128,255,128
50 DATA 255,128,128,129,129,129,143,137,137,136
60 DATA 136,248,128,255,128,128,128,128,255,128
70 DATA 255,192,192,192,192,192,194,194,194,194
80 DATA 194,203,200,207,200,200,192,192,255,128
90 POKE 517,1
100 PRINTCHR$(18);
110 Y=0:Z=0
120 Y=Y+1
130 READ A
140 PRINTCHR$(A);
150 IFY=20 THEN PRINT:Y=0:Z=Z+1
160 IFZ=4 THEN 180
170 GOTO 120
180 POKE 517,0
190 END
```

Toelichting:

```
 90 Schakelt de printer in.
100 Zet mijn printer in de grafische mode.
180 Schakelt de printer uit.
```

P.s.
Mijn printer (TRS-80 Line Printer VIII) vraagt bij het gaan naar een nieuwe strook een extra 128. Dit zal niet bij iedere printer nodig zijn. In dat geval vervalt op de regels 20,40,60 en 80 de laatste 128 en wordt regel 150 IFY=19 enz.

Leo Luijsterburg.

F O R T H K L O K

EEN PROGRAMMA VAN M. J. P. GALLAND HULKSTRAAT 2 1784 RL DEN HELDER

OM DIT PROGRAMMA TE DRAAIEN HET VOLGENDE OPGEVEN :
  XX LOAD XX IS HET BEGIN SCREEN NUMMER
  RUN

DE DELAY MOET AAN UW COMPUTER AANGEPAST WORDEN !!!

G R O T E T E K E N S I N F O R T H

DIT PROGRAMMA ZET OP EEN DOOR U AANGEGEVEN LIJN OP HET
SCHERM DE DOOR U AANGEGEVEN KARAKTERS. EEN UITSTEKEND
VOORBEELD VAN DE WERKWIJZE DAARVAN IS MIJN PROGRAMMA
FORTH KLOK.

OPMERKINGEN:
- MET DIT PROGRAMMA KUNNEN 9 KARAKTERS PER REGEL WORDEN GETOOND;
- ALS U DEZE ROUTINE VOOR ANDERE DOELEINDEN WILT GEBRUIKEN
  MOET U HET VOLGENDE DOEN:
  + BEGIN ADRES VAN DE REGEL IN STRT INVULLEN ;
  + EVENTUELE OFFSET IN OFFS PLAATSEN ;
  + HET KARAKTER-NUMMER (BEGINNEND MET 0) IN KARNR ZETTEN
  + EN DAN OPGEVEN B. V. 9 S1 ! S1 TEKEN
  + (9 IS HET TE PRINTEN KARAKTER)
- DE PRIMITIVES VAN SCREEN 13 MOETEN GELADEN ZIJN:
  DIT ZIJN: : CLS 3 EMIT ;
             : WAIT 0 DO LOOP ;
             : PRNT DUP . ;

```
 SCR # 1
  0 ( --- GROTE TEKENS --- )
  1 13 LOAD ( LAAD PRIMITIVES ) HEX
  2 11 VARIABLE OFFS 0 VARIABLE KARNR D280 VARIABLE STRT
  3 DECIMAL
  4 : ADR 64 * OFFS @ + KARNR @ 5 * + STRT @ + ;
  5 : STR DUP DUP ADR 1+ 148 SWAP C! ADR 2+ 148 SWAP C!
  6                ADR 3 + 148 SWAP C! ;
  7 : ZL ADR 147 SWAP C! ; : ZR ADR 4 + 146 SWAP C! ;
  8 : BOV 1 STR ;  : MID 4 STR ;  OND 7 STR ;
  9 : BL  2 ZL 3 ZL ;  : BR 2 ZR 3 ZR ;
 10 : OL  5 ZL 6 ZL ;  : OO 5 ZR 6 ZR ;
 11 : LK  BL OL ;      : RK BR OO ;
 12 : BMO BOV MID OND ;  ZETSP 32 SWAP C! ;
 13 : RESLYN DUP ADR ZETSP DUP ADR 1+ ZETSP DUP ADR 2+ ZETSP
 14          DUP ADR 3 + ZETSP DUP ADR 4 + ZETSP ;
 15  RESTEK 8 1 DO I RESLYN LOOP ; -->
```

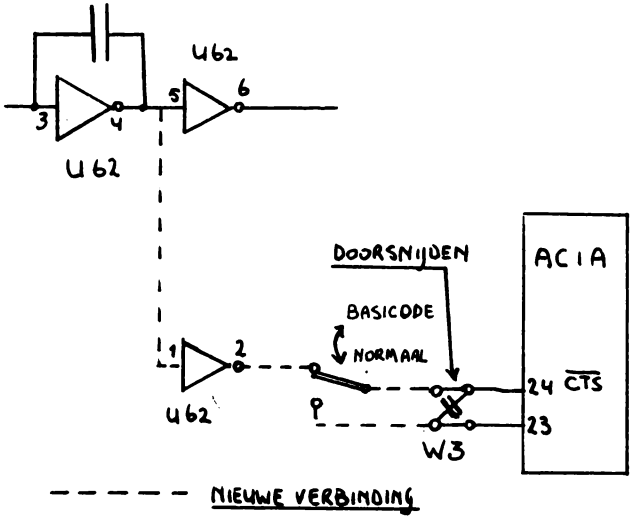
```
SCR # 2
  0 ( --- GROTE TEKENS ---  VERVOLG)
  1 : EEN     RK                  ;
  2 : TWEE    BR     BMO    OL    ;
  3 : DRIE    BMO    RK           ;
  4 : VIER    BL     MID    RK    ;
  5 : VYF     BL     BMO    OO    ;
  6 : ZES     LK     BMO    OO    ;
  7 : ZEVEN   BOV    RK           ;
  8 : ACHT    LK     BMO    RK    ;
  9 : NEGEN   BL     BMO    RK    ;
 10 : NUL     LK     BOV    OND   RK    ;
 11 : TEKEN @ RESTEK DUP 1 = IF EEN THEN DUP 2 = IF TWEE THEN
 12     DUP 3 = IF DRIE THEN DUP 4 = IF VIER THEN
 13     DUP 5 = IF VYF THEN DUP 6 = IF ZES THEN
 14     DUP 7 = IF ZEVEN THEN DUP 8 = IF ACHT THEN DUP 9 =
 15     IF NEGEN THEN DUP 0 = IF NUL THEN ; FORTH S;


SCR # 3
  0 ( -- FORTH KLOK -- )
  1 1 LOAD  ( LAAD SPECIALE PRIMITIVES EN TEKEN ROUTINES)
  2 0 VARIABLE S1  0 VARIABLE S10 0 VARIABLE M1
  3 0 VARIABLE M10 0 VARIABLE U1
  4 0 VARIABLE U10 0 VARIABLE D1
  5 : GETK KEY 48 - ;   : TK  TEKEN ;
  6 : DELAY D1 WAIT ;   : KR  KARNR ;
  7 : GET2C GETK DUP 10 * GETK DUP ROT + . ;
  8 : LEESTYD ." BEGIN TYD (UUMMSS) : " GET2C ." :" U1 ! U10 !
  9           GET2C ." :" M1 ! M10 ! GET2C ." :" S1 ! S10 ! ;
 10 : DISS1   7 KR ! S1 TK ;   : DISS10  6 KR ! S10 TK ;
 11 : DISM1   4 KR ! M1 TK ;   : DISM10  3 KR ! M10 TK ;
 12 : DISU1   1 KR ! U1 TK ;   : DISU10  0 KR ! U10 TK ;
 13 : DISPLAY DISS1 DISS10 DISM1 DISM10 DISU1 DISU10 ;
 14 : INSTEL CLS LEESTYD CLS DISPLAY ;
 15 -->


SCR # 4
  0 ( -- FORTH KLOK --  VERVOLG)
  1 : ?START  BEGIN 57100 @ -16706 = UNTIL ;
  2 : ?STOP   57100 @ -8482 = ;
  3 : TEST    DUP @ 1+ ROT OVER < IF DROP 0 THEN DUP ROT ! ;
  4 : SEC1    9 S1 TEST DISS1  ;   : SEC10   5 S10 TEST DISS10 ;
  5 : MIN1    9 M1 TEST DISM1  ;   : MIN10   5 M10 TEST DISM10 ;
  6 : UUR1    9 U1  TEST DISU1 ;   : UUR10   2 U10 TEST DISU10 ;
  7 : ?24UUR U10 @ 2 = IF U1 @ 4 = IF 0 U1 ! 0 U10 !
  8          DISU1 DISU10 THEN THEN ;
  9 : TYD SEC1 0= IF SEC10 0= IF MIN1 0= IF
 10       MIN10 0= IF UUR1 0= IF UUR10
 11       ELSE ?24UUR THEN THEN THEN THEN ;
 12 : LOPEN BEGIN TYD DELAY ?STOP UNTIL ;
 13 : RUN INSTEL ?START LOPEN ;
 14 FORTH
 15 ;S
```

```
 10 0000         ;**********************************
 20 0000         ;*     Hobbyscoop BASICLOADER     *
 30 0000         ;* Oorspronkelijk ontworpen voor  *
 40 0000         ;* PET/CBM door Jos Courbois      *
 50 0000         ;*              Benno Zuure       *
 60 0000         ;*              Hans Courbois     *
 70 0000         ;* Voor OSI 1P (2MHz) bewerkt     *
 80 0000         ;* door : Peter Hinderks          *
 90 0000         ;*        Rugbypad 5              *
100 0000         ;*        3223 EP Hellevoetsluis  *
110 0000         ;*        tel. 01883-13176        *
120 0000         ;*                                *
130 0000         ;* Gebruiksaanwijzing ;           *
140 0000         ;* Eerst laden met monitor daarna *
150 0000         ;* koude start, POKE 5,2 en dan   *
160 0000         ;* RETURN op de leader v/d band.  *
170 0000         ;**********************************
180 0000         ;
190 0000         ;**** Deklaraties
200 0000         ;
210 0230                 *=$0230
220 0230          NUL    =$D150  Videoram
230 0230          POS    =$00E0
240 0230          COUNT  =$00E2
250 0230          CHAR   =$0130
260 0230          TUSSEN =$0131
270 0230          GEMMID =$01C0
280 0230          OLDPER =$01C1
290 0230          INPUT  =$F000  Statusreg. ACIA
300 0230         ;
310 0230         ;*** Initialisatie (vervolg)
320 0230         ;
330 0230 A902     VRVOLG LDA #$02
340 0232 8D1902          STA $0219  Inputvector H
350 0235 8D2802          STA $0228  Basicflag
360 0238 A9FB            LDA #$FB
370 023A 8505            STA $05    Herstel readyprinter
380 023C 60              RTS
390 023D         ;
400 023D         ;*** Vindt de aanvangsleader en
410 023D         ;*** stel de rest v/h programma in
420 023D         ;*** op de gevonden snelheid.
430 023D         ;
440 023D 20EB02   START  JSR INIT Initialiseer
450 0240 20B002   FINDLE JSR PERIOD
460 0243 C92C            CMP #44  Tijd van 1
470 0245 90F9            BCC FINDLE
480 0247 E6E2            INC COUNT
490 0249 D0F5            BNE FINDLE
500 024B A000            LDY #0
510 024D 8CC001          STY GEMMID
```

```
 520 0250 20B002  OPNIEU JSR PERIOD
 530 0253 0A             ASL A  Tweemaal
 540 0254 18             CLC
 550 0255 6DC001         ADC GEMMID
 560 0258 8DC001         STA GEMMID
 570 025B 9001           BCC DOORGN
 580 025D C8             INY
 590 025E E6E2    DOORGN INC COUNT
 600 0260 D0EE           BNE OPNIEU
 610 0262 98             TYA  Deel door twee
 620 0263 E90E           SBC #$0E
 630 0265 8D50D1         STA NUL
 640 0268                ;
 650 0268                ;*** Wacht op eerste startbit
 660 0268                ;
 670 0268 20B302  FINDBE JSR HALFPE
 680 026B CD50D1         CMP NUL
 690 026E 90F8           BCC FINDBE
 700 0270                ;
 710 0270                ;*** Lees een byte
 720 0270                ;
 730 0270 A008   READBY  LDY #$08
 740 0272                ;
 750 0272                ;*** Lees 1 bit
 760 0272                ;
 770 0272 20B002  READ1  JSR PERIOD
 780 0275 CD50D1         CMP NUL
 790 0278 B006           BCS FOUND0
 800 027A                ;
 810 027A                ;*** Bit is 1
 820 027A                ;
 830 027A 20B002 FOUND1  JSR PERIOD
 840 027D 38             SEC
 850 027E B001           BCS READ2
 860 0280                ;
 870 0280                ;*** Bit is 0
 880 0280                ;
 890 0280 18     FOUND0  CLC
 900 0281                ;
 910 0281                ;*** Rol 'n bit in byte
 920 0281                ;
 930 0281 6E3001 READ2   ROR CHAR
 940 0284 88             DEY
 950 0285 D0EB           BNE READ1
 960 0287                ;
 970 0287                ;*** 1 byte binnen,haal B7 weg
 980 0287                ;*** en test op ETX (end of text)
 990 0287                ;
1000 0287 AD3001         LDA CHAR
1010 028A 297F           AND #$7F
1020 028C 91E0   ENDLOA  STA (POS),Y
1030 028E C903           CMP #$03  End of text?
```

```
1040 0290 F03A          BEQ BASIC
1050 0292         ;
1060 0292         ;*** Verhoog de pointer.
1070 0292         ;
1080 0292 E6E0    VERDER INC POS
1090 0294 D002          BNE FINDST
1100 0296 E6E1          INC POS+1
1110 0298         ;
1120 0298         ;*** Verhoog pointer
1130 0298         ;
1140 0298 A900    FINDST LDA #0
1150 029A 85E2          STA COUNT
1160 029C         ;
1170 029C         ;*** Wacht op startbit volgende byte
1180 029C         ;
1190 029C 20B302  LABEL  JSR HALFPE
1200 029F CD50D1        CMP NUL
1210 02A2 9002          BCC INCOUN
1220 02A4 B0CA          BCS READBY
1230 02A6 E6E2    INCOUN INC COUNT
1240 02A8 D0F2          BNE LABEL
1250 02AA A903          LDA #$03 End of text
1260 02AC D0DE          BNE ENDLOA
1270 02AE         ;
1280 02AE         ;*** Jump naar start
1290 02AE         ;
1300 02AE D08D          BNE START
1310 02B0         ;
1320 02B0         ;*** Laad een halve periode
1330 02B0         ;
1340 02B0 20B302  PERIOD JSR HALFPE
1350 02B3         ;
1360 02B3         ;*** Laad een halve periode
1370 02B3         ;
1380 02B3 A200    HALFPE LDX #0
1390 02B5 AD00F0        LDA INPUT
1400 02B8 8D3101        STA TUSSEN
1410 02BB         ;
1420 02BB         ;*** Wacht tot poort verandert
1430 02BB         ;
1440 02BB E8      HALF1  INX
1450 02BC AD00F0        LDA INPUT
1460 02BF CD3101        CMP TUSSEN
1470 02C2 F0F7          BEQ HALF1
1480 02C4 8A            TXA
1490 02C5 6DC101        ADC OLDPER
1500 02C8 8EC101        STX OLDPER
1510 02CB 60            RTS
1520 02CC         ;
1530 02CC         ;*** Naar BASIC inputroutine
1540 02CC         ;
```

```
1550 02CC 20EB02  BASIC  JSR INIT Initialiseer
1560 02CF 4C74A2         JMP $A274 Warme start BASIC
1570 02D2 A000    LOCA   LDY #$0
1580 02D4 B1E0           LDA (POS),Y
1590 02D6 C903           CMP #$03
1600 02D8 D00a           BNE GOON.
1610 02DA A9BA           LDA #$BA
1620 02DC 8D1802         STA $0218 Herstel vektor
1630 02DF A9FF           LDA #$FF
1640 02E1 8D1902         STA $0219
1650 02E4 E6E0    GOON   INC POS
1660 02E6 D002           BNE RETURN
1670 02E8 E6E1           INC POS+1
1680 02EA 60      RETURN RTS
1690 02EB         ;
1700 02EB         ;*** Initialisatie *
1710 02EB         ;
1720 02EB A900    INIT   LDA #0
1730 02ED 85E2           STA COUNT
1740 02EF 85E0           STA POS
1750 02F1 A904           LDA #$04 Opslag basicode
1760 02F3 85E1           STA POS+1
1770 02F5 A9D2           LDA #$D2
1780 02F7 8D1802         STA $0218  Inputvektor L
1790 02FA 4C3002         JMP VRVOLG
1800 02FD 4C00FF         JMP $FF00  Break
1810 0300                .END ;;;;;
```



HEX DUMP VAN DE QUICKSAVE (48 karakter-versie)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
D500 :	A0	00	20	DD	D5	A0	0D	A2	04	20	BE	D5	E8	88	20	BE
D510 :	D5	CA	CA	CA	88	88	10	F1	A0	10	20	DD	D5	A9	E8	8D
D520 :	AB	D1	A2	0A	A9	00	20	B1	FC	CA	D0	FA	A2	03	20	3E
D530 :	D6	A9	2F	20	B1	FC	A2	18	A0	10	B9	09	D3	20	1E	D6
D540 :	C8	CA	D0	F6	A2	01	20	3E	D6	A9	2F	20	B1	FC	A2	69
D550 :	A0	00	B9	00	D0	20	1E	D6	C8	CA	D0	F6	A2	05	20	3E
D560 :	D6	A9	2F	20	B1	FC	A9	54	20	B1	FC	A0	06	A2	00	B5
D570 :	F0	20	1E	D6	E8	88	D0	F7	A2	01	20	3E	D6	A9	47	20
D580 :	B1	FC	A9	B9	8D	00	F0	20	13	D6	B1	F0	20	B1	FC	A9
D590 :	01	4D	AB	D1	8D	AB	D1	A5	F0	C5	F2	D0	19	A5	F1	C5
D5A0 :	F3	D0	13	20	13	D6	A9	B1	8D	00	F0	A9	20	8D	AB	D1
D5B0 :	20	D0	FB	4C	00	D5	E6	F0	D0	D0	E6	F1	D0	CC	B9	09
D5C0 :	D3	20	93	FE	30	14	95	F0	88	B9	09	D3	20	93	FE	30
D5D0 :	09	0A	0A	0A	0A	15	F0	95	F0	60	4C	43	FE	A9	B7	99
D5E0 :	49	D3	A2	00	20	00	FD	C9	0D	D0	01	60	48	A9	20	99
D5F0 :	49	D3	68	C9	3C	F0	0F	E0	18	F0	11	C9	3E	F0	03	99
D600 :	09	D3	E8	C8	D0	06	E0	00	F0	02	CA	88	A9	B7	99	49
D610 :	D3	D0	D1	A2	00	A0	00	88	D0	FD	CA	D0	F8	60	48	4A
D620 :	4A	4A	4A	20	32	D6	68	29	0F	20	32	D6	A9	0D	20	B1
D630 :	FC	60	09	30	C9	3A	90	02	69	06	20	B1	FC	60	A9	2E
D640 :	20	B1	FC	BD	51	D6	20	1E	D6	CA	BD	51	D6	20	1E	D6
D650 :	60	00	D0	19	D3	F0	00									
D000 :	A9	00	85	FB	A9	20	20	60	D0	A9	FD	8D	00	DF	AD	00
D010 :	DF	C9	EF	D0	03	4C	43	FE	A2	00	A9	B9	8D	00	F0	A0
D020 :	00	AD	00	F0	4A	90	FA	2C	00	F0	50	10	A9	74	20	60
D030 :	D0	A9	B1	8D	00	F0	20	D0	FB	4C	43	FE	AD	01	F0	91
D040 :	F0	A5	F0	C5	F2	D0	11	A5	F1	C5	F3	D0	0B	A9	B1	8D
D050 :	00	F0	20	D0	FB	6C	F4	00	E6	F0	D0	C5	E6	F1	D0	C1
D060 :	A2	00	9D	88	D3	CA	D0	FA	60							

Hoe save ik quicksave zelf?

Quicksave is het gemakkelijkst te savenen met richard.
Voordeel: load-gedeelte wordt vanzelf voorafgeladen (de hoeveelheid je hoeft dan dus alleen het save-gedeelte te savenen.
Dat is van : tot :
24 kar: D1ØØ - D256
48 " : D5ØØ - D656.
Zelfstart : D1ØØ
resp D5ØØ.

NB. Bij source listing 48 kar. staat (handgeschreven):
. regel = $ D3Ø8 " .
Niet zijn: $ D3Ø9.

<<<< QUICKSAVE >>>>

Eerdere publikaties:

1e 50-p. boek, blz 9: eerste introduktie. Bevat omschrijving van de mogelijkheden, gebruiksaanwijzing en een dump van de machine-kode om in te typen als je 'm niet van iemand ander's kassette overneemt. Bevat verder de source-listing, zonder noemenswaardige toelichting.

HCCN 28, input no.7 (mei 1981, blz. 29). Bevat wat informatie voor alle gebruikers, die hier in grote lijnen herhaald zal worden, en een quicksave versie voor de C2P.

Deze publikatie:

- is bedoeld als vervolg op het artikel uit het 1e 50-p. boek,
- geeft voor de nieuwe machines OSI Superboard versie 2 (of hoe ze ook precies heten) aangepaste versies van Quicksave,
- gaat in op de principes van wat er in het programma gebeurt,
- geeft een programma "Crossload", dat meer mogelijkheden biedt bij het laden.

Quicksave voor de nieuwe versies.

We hebben nu voor alle ons bekende OSI's versies van Quicksave en Crossload. We geven hier welke machines welke versies nodig hebben.

1. De C2P, de voorouder van het Superboard. Quicksave uit 1e 50-p. boek, aangepast volgens HCCN 28, blz 29. Crossload: de C2P versie uit dit boek.

2. Het eerste Superboard, zoals ie voor 't eerst op de markt kwam: met 1K videoram, 24*24 kar's en de oude monitorrom (dus niet de EPROM). Quicksave V24.3: (=de 24 kar. versie uit 't 1e 50-p. boek). Crossload: versie V1 uit dit boek.

3. Dezelfde machine, maar met de nieuwe monitor EPROM. Q'save: VW1K24, hier voor 't eerst gepubliceerd. Crossload: V2, ook uit dit boek.

4. Dezelfde machine, maar videoram uitgebreid tot 2K, dus 24*48. Quicksave V48.3 (=de 48 kar. versie uit 't 1e 50-p. boek. Crossload: V1.

5. De nieuwe Superboard versie 2, met EPROM en 1K videoram. Quicksave: VW1K24 of VW1K48 alnaargelang de instelling van het aantal karakters per regel. Crossload: V2.

6. Dezelfde nieuwe, maar videoram uitgebreid tot 2K, dus 24*48. Quicksave: V48.3. Crossload: V1.

De principes van Quicksave.

Taak: op band zetten van een stuk geheugen. Op zich al vreemd dat de monitorrom hier geen routine voor bevat. Dan zou Quicksave niet nodig geweest zijn. Laden kan de monitor wel, anders waren we nog verder van huis.

Onze OSI's hebben de bijzondere eigenschap dat in- en uitvoer naar de band precies hetzelfde worden behandeld als die van toetsenbord resp. naar het scherm. Als er een karakter ingevoerd moet worden kijkt de invoerroutine naar de load-vlag. Staat deze uit, dan wordt het karakter via het toetsenbord gehaald, en anders van de band (via de acia). De uitvoer gaat net zo, alleen print de uitvoerroutine het karakter altijd. De monitor-mode is een uitzondering: deze kijkt wel naar een (eigen) load-vlag, maar saven kan hij niet. Voor de hele band-I/O zijn er dus maar 2 superkleine routinetjes, en de mogelijkheden voor band zijn precies die voor de gewone I/O. Vandaar dat in BASIC na een LOAD kommando het voor de BASIC net is alsof iemand het programma (razendsnel) intypt. Dat gaat goed omdat na een LIST precies tevoorschijn komt wat zo iemand zou moeten intypen. Een erg mooi principe, vinden wil: slechts een soort I/O. Schoonheidsfoutje: de prompt ("OK" of "ready" of ">") die na een LIST op de band komt, geeft bij laden een SYNTAX error.

Nu terug naar Quicksave: door het intypen van de "L" in de monitormode wordt de load-vlag aangezet. Voorzieningen voor weer uitzetten (zoals de spatiebalk in BASIC), zijn er niet: alleen de "nieuwe" invoer, de band, kan dat, of natuurlijk de break toets. Maar die is een doodsvijand van Quicksave, omdat het in het videoRAM staat.

Na het intypen van de "L" is de machine dus geheel in handen van de band, die nu alles kan wat normaal in de monitor met het toetsenbord kan. Dus onder andere programma's intypen met de karakters: ".", "/", en de hex cijfers, en zelfs programma's starten met de "G". Dit is het principe van de hexdump van Hans de Jong in HCCN 16. Nadeel: voor te laden byte zijn drie volledige 8-bits karakters op de band nodig: de twee hex cijfers en de <RETURN>. Het laden zou dus drie maal zo snel kunnen.

Crossload

De enige andere mogelijkheid is dat de band met de genoemde karakters zelf een machinetaal programmaatje "intypt", dat hij zelf start ("G"), en dat daarna voor hem de gewenste akties zelf uitvoert. Dit principe wordt in de checksum-save/load toegepast (verspreid door Rein Heesterman in de beginperiode van de OSI-GG), en uiteraard ook in Quicksave. Het programma dat voor de echte data geladen wordt wordt wel "header" genoemd. Wij (de auteurs) noemen het het "autoload"-programma.

Voordat we een opsomming geven van wat er tijdens het laden precies gebeurt moet je weten wat door Quicksave op de band wordt gezet. (hieruit blijkt al voor een deel wat er gebeurt):

. XXXX / XX XX ... XX (=identifikatie in ascii-kodes = in hexdump)

. D000 / XX XX ... XX (=autoload in hexdump)

. 00F0 / T XX XX XX XX XX XX (=de drie adressen, zie onder. De "T" doet niets, dient als merktekentje.)

. D000 / G (=starten van autoload)

XXXXXXXX ... XXX (=de data in Quicksave format).

We beginnen de opsomming na de "L" in de monitor-mode met de load-vlag aan.

- De identifikatie wordt in het videoRAM gezet (en dus zichtbaar gemaakt)

- Het autoload wordt op zijn plaats gezet (in alle machines op D000)

- De drie adressen worden opgeslagen (op 00F0). Deze zijn begin-, eind- en zelfstartadres, en geven voor autoload aan waar deze data moeten komen, en waar de processor door moet gaan na het laden . Deze adressen zijn bepaald toen de data met Quicksave op de band werden gezet, nl. je hebt ze toen zelf gekozen en ingetypt.

- Autoload wordt gestart.

Nu is de machine uit de monitor-mode: autoload heeft het nu voor het zeggen, en voert de volgende stappen uit:

- schoon schip maken (load-vlag uit e.d.)

- kontroleren of spatiebalk is ingedrukt, en zo ja: stoppen (=springen naar de warme monitor)

- alle data laden (inklusief parity kontrole en stoppen als fout)

- klaarmelding (=piep)

- springen naar het opgegeven zelfstartadres.

Dat was het verhaal over de principes waarmee Quicksave werkt. Het zal nu duidelijk zijn, dat alles wat er zal gebeuren nadat je eenmaal de "L" hebt ingetypt nu al op de band vastligt. Bijvoorbeeld begin-, eind- en zelfstartadres. Het is zelfs niet helemaal logisch dat het autoload, afkomstig van de machine waarop de data gesaved zijn, moet runnen op de ladende machine. (Tot nu toe gaat dat alleen maar mis bij uitwisselen tussen C2P en een ander type.)

In normale gevallen heeft deze situatie geen gevolgen, maar soms hebben we tijdens het laden meer vrijheid nodig. Daarom hebben we het programma "Crossload" geschreven. Het principe is: niet afwachten wat de band gaat doen, maar zelf een programma runnen, dat met de informatie op de band precies doet wat ik wil. De kern van Crossload is exact het autoload-gedeelte van Quicksave: alleen het programma eromheen is iets anders. De hier volledig geliste versie is gemaakt om precies te doen wat autoload ook doet, nl. gewoon laden. Die noemen we dan ook de standaard crossload. Hij is gemakkelijk aan te passen voor speciale taken.

We geven nu in dezelfde stijl als bij gewoon laden wat er gebeurt bij laden m.b.v. standaard crossload. (Wat op de band staat natuurlijk precies hetzelfde.)

- Crossload wordt gestart, nadat het op zijn plaats is gezet

Nu is de machine dus in handen van crossload. Dat doet het volgende:

- wachten op een toets. Typ je een "X" in dan springt het zonder uitstel naar de warme monitor, en je hebt het zelf weer in de hand. Typ je een willekeurige andere toets in dan gaat crossload naar de volgende stap:

- wachten tot op de band een "T" passeert. (Dat is het merktekentje waarmee Quicksave speciaal voor zulke doeleinden het begin van de "adressentabel" aangeeft.) Alles wat ondertussen passeert, dat zijn o.a. de identifikatie en autoload, wordt genegeerd.

- de drie adressen die nu passeren binnenhalen en opslaan. (Dit kost nogal wat rekenwerk: elk adres bestaat uit 2 bytes die ieder weer uit 2 ascii kodes zijn opgebouwd.)

- wachten tot op de band "G" passeert ("G" dient nu als merkteken voor het begin van de data.)

- Vanaf nu doet Crossload precies hetzelfde als het volledige autoload (zelfs het programma is precies hetzelfde), op twee kleinigheden na. Nl. spatiebalk kontrole en belletjes zijn weg.

Voordat we een stel kant en klare recepten geven, volgt hier een vrij gedetailleerde verklaring van de machinekode van Crossload. (Zie de source listing.) Die kan van pas komen als je zelf een receptje wilt bedenken, of als je de bestaande wilt begrijpen. (Hij lijkt me ook leerzaam voor bijna machinetaal leken om eens door te nemen.)

- wachten op toets (regel 190).

als "X" dan springen naar warme monitor (200-220)

- wachten tot "T" op band (230-250)

- adressen tabel binnenhalen en opslaan (270-430) (Begrip van wat hierbinnen gebeurt is niet nodig, en niet interessant voor de rest van het programma: kijk er tegenaan als tegen een "black box".)

- wachten tot "G" op band (440-460)

(Voor de belangstellenden: Wat nu nog volgt geldt vrijwel ook voor autoload).

- schoon schip maken: monitor load- vlag uitzetten (470-480) en eventuele vorige errormelding uitzetten (490-500)

- acia programmeren op 8 bit, met paritybit en 1 stopbit (=$B9) (520-530)

- "alle data ophalen" (zie onder) (540-750)

- acia terugprogrammeren op 8 bit - geen parity - 2 stopbits (=$B1) (760-770)

- springen naar het adres dat door de band als zelfstartadres werd opgegeven (=het derde adres uit de "adressen-tabel") (780)

Alleen "alle data ophalen" behoeft nu nog nadere uitwerking. Per byte data:

- wachten tot acia een byte van de band heeft gehaald (550-570)

- kontroleren of parity o.k. (580-590): als fout dan: zet errormelding (600-610), programmeer acia terug (620-630), spring naar warme monitor (640).

- zet het byte in het geheugen (680-690)

- kontroleer of dit het laatste byte was (700-750), als laatste dan is "alle data ophalen" afgelopen (dus naar volgende stap boven = regel 760)

- verhoog "wegzetadres" (790-820).

Na deze uitleg introduceren we diverse "special purpose" crossloads, en de recepten waarmee je die uit de standaard crossload kunt maken en gebruiken. De recepten kun je het best gewoon toepassen wanneer je ze nodig hebt. Wijzigingen in het geheugen aanbrengen gaat altijd als volgt: eerst standaard crossload laden, eruit door "X" te typen, dan (in de gewone monitor mode) de wijzigingen intypen, dan crossload herstarten op D260 <D080>. Het eerste adres geldt altijd voor crossloads versies V1 en VC2P, het tweede (tussen "<" en">") voor versie V2.

1. Zelfstart zelf kiezen.

Toe te passen als je wilt dat de processor na het laden op een ander adres doorgaat dan op het zelfstartadres van de band.

Principe: vervang in crossload de sprong naar zelfstart door een sprong naar gewenste adres.

In source: regel 780: JMP (SAL) wordt JMP eigen adres.

Recept: op D2DD <D0FD> intypen: 4C low high ipv 6C F4 00. (Low en high zijn minst en meest signifikante bytes van gewenste adres).

Voorbeeld: start op $FE43 (=warme monitor) gewenst. Dan intypen 4C 43 FE.

2. Begin-, eind- en zelfstartadres zelf kiezen

Principe: de drie adressen die op band staan gewoon NIET binnenhalen en opslaan, maar zelf voor starten van crossload invullen.

In source: op LA.11 komt JMP LA.10 (= tabel invullen overslaan).

Recept

- op D26A <D08A> intypen 4C 93 D2 <4C B3 D1> ipv 20 80 FE

- op 00F0 t/m 00F5 intypen gewenste begin-, eind- en zelfstartadres, ieder adres in low-high volgorde.

- laden als met standaard crossload.

3. Doorladen en errortabel aanleggen

Als, om welke reden dan ook, het niet lukt zonder fouten de data te lezen, mag het laden niet stoppen bij de eerste de beste parity-fout. Deze versie van crossload probeert dan ook ook de rest van de data te laden. Bovendien legt hij in het geheugen een tabel aan van alle adressen waarop een byte staat dat met parityfout binnenkwam. NB: dit is geen garantie dat ALLE foute bytes gedekteerd zijn: parity kontrole detekteert alleen bytes waarin een oneven aantal foute bits zit.

De plaats van de tabel is zelf te kiezen; uiteraard moet die plaats "vrij" zijn. Is er geen ruimte voor zo'n tabel, dan kun je een adres kiezen waar geen RAM zit. Bijvoorbeeld op de Basic: $A000. Dan kun je namelijk nog wel aflezen hoeveel fouten er waren. De adressen zitten in high-low (!) volgorde in de tabel. Hij begint op het gekozen adres, en na elk veelvoud van 128 foute bytes begint hij opnieuw, overschrijft hij dus zichzelf. Adres $00F6 bevat het aantal fouten * 2 + 1, modulo 128.

Principe: de errormelding vervangen.

Recept: op D2B7 <D0D7> intypen:

```
A5 F0        LDA $00F0
9D low high  STA tabel adres,X
E8           INX
A5 F1        LDA $00F1
9D low high  STA tabeladres,X
E8           INX
86 F6        STX $00F6
```
(low en high als in recept 1.)

- laden als met standaard crossload
- na afloop aantal fouten en tabel inspecteren.

4. Acia resetten na error

Als de acia op een bepaald moment niet (meer) weet bij welk bit op de band een nieuw byte begint, en hij gokt mis, dat resulteert een grote troep. Dat is bij ons een keer voorgekomen toen we probeerden met crossload data te laden waarvan het autoload e.d. en zelfs een stuk data waren uitgewist. Laden lukte toen absoluut niet. Een truuk is dan om de acia te resetten iedere keer dat hij misgokt, zodat hij dan opnieuw begint en dus opnieuw gokt. De kans is groot dat hij na een byte of 20 weer goed zit, zodat de rest goed binnenkomt (alleen hoogst waarschijnlijk niet op de goede adressen).

Principe: errormelding vervangen door reset van de acia.

Recept: op D2B7 <D0D7> intypen:

```
A9 03        LDA #$03
8D 00 F0     STA contro
A9 B9        LDA #$B9
8D 00 F0     STA contro
4C AC D2 <CC D0>  JMP loadad
```
(op de C2P: FC ipv F0)

- laden als met standaard crossload.

5. Laden van een "vreemde" machine

Mocht de autoload die bij de data zit, dus de autoload van de machine waarop gesaved is, niet werken op jouw machine, dan kun je altijd gewoon laden met de standaard crossload voor JOUW machine.

Eind van de recepten.

Heb je een probleem dat misschien met crossload opgelost kan worden, of heb je een suggestie, bel dan even: wie weet kunnen we een nieuw recept bedenken, dat dan door iedereen gebruikt kan worden.

Intypen, laden e.d. van Crossload: Crossload kan op alle machines gewoon met de Quicksave gesaved worden, behalve met de versie V24.3 (=de allereerste, =de 24-kar. versie uit het 1e 50-p. boekje. Daar moet voor het saven het busy tekentje verhuisd worden: D0 ipv D2 op D121, D193, D196 en D1AF.) Het loopt van D260 t/m D2F0 <D080 t/m D110>.

Nog een handigheidje: een "vieuwertje" voor Quicksave.

Het is wel een leuk of handig eens te kijken wat Quicksave nu precies op de band heeft gezet. Daarvoor dit programmaatje, waarin alles geprint wordt maar elke <CR> (=$0D) wordt vervangen door een spatie (=$20):

```
20 80 FE  JSR load
C9 0D     CMP # return
D0 02     BNE ..
A9 20     LDA # spatie
20 69 FF  JSR print
C9 47     CMP #'G
D0 F0     BNE ..
20 00 FD  JSR getkey
4C 00 FE  JMP monitor
```

Nog wat losse opmerkingen:

Veel mensen blijken te denken dat tijdens saven altijd als derde adres D500 moet worden ingetypt. Dit omdat dat adres toevallig staat bij "hoe save ik Quicksave zelf?". De werkelijkheid is dat D500 het startadres is van Quicksave, dat je tenslotte op dat moment aan het saven bent. Nu niet denken: Oh, dan moet daar zeker altijd het startadres. Nee, je mag zelf een adres kiezen. Soms is het startadres van het gesavede programma handig, soms (vaak) de warme of koude monitor (FE43 resp. FE00), of misschien wel de extended monitor, zie maar.

Het onderscheid "warme" en "koude" monitor hebben wij zelf verzonnen, naar analogie met "warme" en "koude" BASIC start. Via de warme monitor start kom je namelijk in de monitor mode, net als bij de koude monitor start of na <BREAK> "M", maar o.a. de initialisatie van het videoRAM blijft achterwege (en dat is natuurlijk van levensbelang voor Quicksave).

Omdat Quicksave en Crossload in het videoram staan, worden ze volledig uitgewist door het intypen van <BREAK>. Daarom opletten als je het programma voor jezelf intypt. Gevaarlijk zijn de "G", "L" en <BREAK> toetsen. Nog gevaarlijker is het programma starten als je (nog) niet zeker weet of je alles goed hebt ingetypt: als de processor verdwaalt kun je niet anders meer dan breaken. Daarom: gewoon geen fouten maken.

In de gebruiksaanwijzing van Quicksave (in het 1e 50 p-boek) staat op blz. 11 halverwege dat beginadres niet kleiner mag zijn dan het eindadres. Het is natuurlijk andersom.

Voor de duidelijkheid: er bestaan maar twee soorten programma's die je zelf zult "hanteren": Quicksave en Crossload. Autoload is namelijk een onderdeel van Quicksave, en komt nooit zelfstandig voor (daarvoor is Crossload juist).

```
 10 0000           ;CROSSLOAD V1
 20 0000           CONTRO=$F000
 30 0000           STATUS=CONTRO
 40 0000           DATAPO=CONTRO+1
 50 0000           BASE   =$F0
 60 0000           BAL    =BASE
 70 0000           BAH    =BASE+1
 80 0000           EAL    =BASE+2
 90 0000           EAH    =BASE+3
100 0000           SAL    =BASE+4
110 0000           SAH    =BASE+5
120 0000           START  =$D260
130 0000           GETKEY=$FD00
140 0000           MNLOAD=$FE80
150 0000           HEXBIN=$FE93
160 0000           WARMON=$FE43
170 0000           ERRMEL=$D105
180 D260           *      =START
190 D260 2000FD    START  JSR GETKEY
200 D263 C958             CMP #'X
210 D265 D003             BNE *+5
220 D267 4C43FE           JMP WARMON
230 D26A 2080FE    LA.11  JSR MNLOAD
240 D26D C954             CMP #'T
250 D26F D0F9             BNE LA.11
260 D271 A200             LDX #$00
270 D273 A006             LDY #$06
280 D275 2080FE    LOATAB JSR MNLOAD
290 D278 2093FE           JSR HEXBIN
300 D27B 30F8             BMI LOATAB
310 D27D 0A               ASL A
320 D27E 0A               ASL A
330 D27F 0A               ASL A
340 D280 0A               ASL A
350 D281 95F0             STA BASE,X
360 D283 2080FE           JSR MNLOAD
370 D286 2093FE           JSR HEXBIN
380 D289 30F8             BMI *-6
390 D28B 15F0             ORA BASE,X
400 D28D 95F0             STA BASE,X
410 D28F E8               INX
420 D290 88               DEY
430 D291 D0E2             BNE LOATAB
440 D293 2080FE    LA.10  JSR MNLOAD
450 D296 C947             CMP #'G
460 D298 D0F9             BNE LA.10
```

Het intypen van de machinekodes

Voor de meeste versies is er gewoon een dump in hex-cijfers, nl. voor Quicksave VW1K24 en Crossloads V1 en V2.

Quicksave VW1K48 is exact Quicksave VW1K24, maar met inter =$10, regel =$D30C en reglen =$0C. Daarom uitgaan van VW1K24, en wijzigen:

- 10 i.p.v. 40 op: D199, D1B9;
- 0C i.p.v. 05 op: D1BB, D23F, D24A, D280;
- 4C i.p.v. 25 op: D260, D270, D28F;
- 1C i.p.v.45 op: D2D3.

Crossload VC2P is exact Crossload V1, met alleen ander acia-adres (FC00 i.p.v. F000). Daarom uitgaan van V1, en wijzigen: FC i.p.v. F0 op: D2A9, D2AE, D2B4, D2C0, D2C9, D2DC.

```
470 D29A A900             LDA #$00
480 D29C 85FB             STA $FB
490 D29E A920             LDA #$20
500 D2A0 20E8D2           JSR SCREEN
510 D2A3 A200             LDX #$00
520 D2A5 A9B9             LDA #$B9
530 D2A7 8D00F0           STA CONTRO
540 D2AA A000             LDY #$00
550 D2AC AD00F0    LOADAT LDA STATUS
560 D2AF 4A               LSR A
570 D2B0 90FA             BCC LOADAT
580 D2B2 2C00F0           BIT STATUS
590 D2B5 5010             BVC NOTERR
600 D2B7 A974             LDA #'t
610 D2B9 20E8D2           JSR SCREEN
620 D2BC A9B1             LDA #$B1
630 D2BE 8D00F0           STA CONTRO
640 D2C1 4C60D2           JMP START
650 D2C4 EA               NOP
660 D2C5 EA               NOP
670 D2C6 EA               NOP
680 D2C7 AD01F0    NOTERR LDA DATAPO
690 D2CA 91F0             STA (BAL),Y
700 D2CC A5F0             LDA BAL
710 D2CE C5F2             CMP EAL
720 D2D0 D00E             BNE LA.5
730 D2D2 A5F1             LDA BAH
740 D2D4 C5F3             CMP EAH
750 D2D6 D008             BNE LA.5
760 D2D8 A9B1             LDA #$B1
770 D2DA 8D00F0           STA CONTRO
780 D2DD 6CF400           JMP (SAL)
790 D2E0 E6F0      LA.5   INC BAL
800 D2E2 D0C8             BNE LOADAT
810 D2E4 E6F1             INC BAH
820 D2E6 D0C4             BNE LOADAT
830 D2E8 A200      SCREEN LDX #$00
840 D2EA 9D05D1           STA ERRMEL,
850 D2ED CA               DEX
860 D2EE D0FA             BNE *-4
870 D2F0 60               RTS
880 D2F1           END
```

QUICKSAVE VW1K24

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
D180  A0 00 20 5D D2 A0 0D A2 04 20 3E D2 E8 88 20 3E
D190  D2 CA CA CA 88 88 10 F1 A0 40 20 5D D2 A9 E8 8D
D1A0  4C D1 A2 0A A9 00 20 B1 FC CA D0 FA A2 03 20 BE
D1B0  D2 A9 2F 20 B1 FC A2 18 A0 40 B9 05 D3 20 9E D2
D1C0  C8 CA D0 F6 A2 01 20 BE D2 A9 2F 20 B1 FC A2 69
D1D0  A0 00 B9 00 D0 20 9E D2 C8 CA D0 F6 A2 05 20 BE
D1E0  D2 A9 2F 20 B1 FC A9 54 20 B1 FC A0 06 A2 00 B5
D1F0  F0 20 9E D2 E8 88 D0 F7 A2 01 20 BE D2 A9 47 20
D200  B1 FC A9 B9 8D 00 F0 20 93 D2 B1 F0 20 B1 FC A9
D210  01 4D 4C D1 8D 4C D1 A5 F0 C5 F2 D0 19 A5 F1 C5
D220  F3 D0 13 20 93 D2 A9 B1 8D 00 F0 A9 20 8D 4C D1
D230  20 D0 FB 4C 80 D1 E6 F0 D0 D0 E6 F1 D0 CC B9 05
D240  D3 20 93 FE 30 14 95 F0 88 B9 05 D3 20 93 FE 30
D250  09 0A 0A 0A 0A 15 F0 95 F0 60 4C 43 FE A9 B7 99
D260  25 D3 A2 00 20 00 FD C9 0D D0 01 60 48 A9 20 99
D270  25 D3 68 C9 3C F0 0F E0 18 F0 11 C9 3E F0 03 99
D280  05 D3 E8 C8 D0 06 E0 00 F0 02 CA 88 A9 B7 99 25
D290  D3 D0 D1 A2 00 A0 00 88 D0 FD CA D0 F8 60 48 4A
D2A0  4A 4A 4A 20 B2 D2 68 29 0F 20 B2 D2 A9 0D 20 B1
D2B0  FC 60 09 30 C9 3A 90 02 69 06 20 B1 FC 60 A9 2E
D2C0  20 B1 FC BD D1 D2 20 9E D2 CA BD D1 D2 20 9E D2
D2D0  60 00 D0 45 D3 F0 00
```

CROSSLOAD V2.

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
D080  20 00 FD C9 58 D0 03 4C 43 FE 20 80 FE C9 54 D0
D090  F9 A2 00 A0 06 20 80 FE 20 93 FE 30 F8 0A 0A 0A
D0A0  0A 95 F0 20 80 FE 20 93 FE 30 F8 15 F0 95 F0 E8
D0B0  88 D0 E2 20 80 FE C9 47 D0 F9 A9 00 85 FB A9 20
D0C0  20 08 D1 A2 00 A9 B9 8D 00 F0 A0 00 AD 00 F0 4A
D0D0  90 FA 2C 00 F0 50 10 A9 74 20 08 D1 A9 B1 8D 00
D0E0  F0 4C 80 D0 EA EA EA AD 01 F0 91 F0 A5 F0 C5 F2
D0F0  D0 0E A5 F1 C5 F3 D0 08 A9 B1 8D 00 F0 6C F4 00
D100  E6 F0 D0 C8 E6 F1 D0 C4 A2 00 9D 80 D1 CA D0 FA
```

CROSSLOAD V1.

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
D260  20 00 FD C9 58 D0 03 4C 43 FE 20 80 FE C9 54 D0
D270  F9 A2 00 A0 06 20 80 FE 20 93 FE 30 F8 0A 0A 0A
D280  0A 95 F0 20 80 FE 20 93 FE 30 F8 15 F0 95 F0 E8
D290  88 D0 E2 20 80 FE C9 47 D0 F9 A9 00 85 FB A9 20
D2A0  20 E8 D2 A2 00 A9 B9 8D 00 F0 A0 00 AD 00 F0 4A
D2B0  90 FA 2C 00 F0 50 10 A9 74 20 E8 D2 A9 B1 8D 00
D2C0  F0 4C 60 D2 EA EA EA AD 01 F0 91 F0 A5 F0 C5 F2
D2D0  D0 0E A5 F1 C5 F3 D0 08 A9 B1 8D 00 F0 6C F4 00
D2E0  E6 F0 D0 C8 E6 F1 D0 C4 A2 00 9D 05 D1 CA D0 FA
D2F0  60
```

Als je een probleem tegenkomt, bel dan even: misschien is het zo opgelost.

Jan en Bas Edixhoven
Hoflandstraat 31
2641 JJ Pijnacker
tel. 01736-3743

september 1982

BASIC DISASSEMBLER

```
1 REM DISASSEMBLER 4.0
2 REM
3 REM  ALMAT SOFTWARE
4 REM
5 REM   23-10-1981
6 REM
7 DATA4,6,14,14,14,3,3,14,4,1,5,14,14,2,2,14
8 DATA12,7,14,14,14,8,8,14,4,11,14,14,14,10,10,14
9 DATA2,6,14,14,3,3,3,14,4,1,5,14,2,2,2,14
10 DATA12,7,14,14,14,8,8,14,4,9,14,14,14,10,10,14
11 DATA4,6,14,14,14,3,3,14,4,1,5,14,2,2,2,14
12 DATA12,7,14,14,14,8,8,14,4,11,14,14,14,10,10,14
13 DATA4,6,14,14,14,3,3,14,4,1,5,14,13,2,2,14
14 DATA12,7,14,14,14,8,8,14,4,11,14,14,14,10,10,14
15 DATA14,6,14,14,3,3,3,14,4,14,4,14,2,2,2,14
16 DATA12,7,14,14,8,8,9,14,4,11,4,14,14,10,14,14
17 DATA1,6,1,14,3,3,3,14,4,1,4,14,2,2,2,14
18 DATA12,7,14,14,8,8,9,14,4,11,4,14,10,10,11,14
19 DATA1,6,14,14,3,3,3,14,4,1,4,14,2,2,2,14
20 DATA12,7,14,14,14,8,8,14,4,11,14,14,14,10,10,14
21 DATA1,6,14,14,3,3,3,14,4,1,4,14,2,2,2,14
22 DATA12,7,14,14,14,8,8,14,4,11,14,14,14,10,10,14
23 DATABRK,ØRA,???,???,???,ØRA,ASL,???,PHP,ØRA,ASL,???,???,ØRA,ASL,???
24 DATABPL,ØRA,???,???,???,ØRA,ASL,???,CLC,ØRA,???,???,???,ØRA,ASL,???
25 DATAJSR,AND,???,???,BIT,AND,R0L,???,PLP,AND,R0L,???,BIT,AND,R0L,???
26 DATABMI,AND,???,???,???,AND,R0L,???,SEC,AND,???,???,???,AND,R0L,???
27 DATARTI,EØR,???,???,???,EØR,LSR,???,PHA,EØR,LSR,???,JMP,EØR,LSR,???
28 DATABVC,EØR,???,???,???,EØR,LSR,???,CLI,EØR,???,???,???,EØR,LSR,???
29 DATARTS,ADC,???,???,???,ADC,R0R,???,PLA,ADC,R0R,???,JMP,ADC,R0R,???
30 DATABVC,ADC,???,???,???,ADC,R0R,???,SEI,ADC,???,???,???,ADC,R0R,???
31 DATA???,STA,???,???,STY,STA,STX,???,DEY,???,TXA,???,STY,STA,STY,???
32 DATABCC,STA,???,???,STY,STA,STX,???,TYA,STA,TXS,???,???,STA,???,???
33 DATALDY,LDA,LDX,???,LDY,LDA,LDY,???,TAY,LDA,TAX,???,LDY,LDA,LDX,???
34 DATABCS,LDA,???,???,LDY,LDA,LDX,???,CLV,LDA,TSX,???,LDY,LDA,LDX,???
35 DATACPY,CMP,???,???,CPY,CMP,DEC,???,INY,CMP,DEX,???,CPY,CMP,DEC,???
36 DATABNE,CMP,???,???,???,CMP,DEC,???,CLD,CMP,???,???,???,CMP,DEC,???
37 DATACPX,SBC,???,???,CPX,SBC(INC,???,INX,SBC,NØP,???,CPX,SBC,INC,???
38 DATABEQ,SBC,???,???,???,SBC,INC,???,SED,SBC,???,???,???,SBC,INC,???
39 DATA0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
40 GØT044
41 IFPEEK(57100)=222THEN58
42 GØT041
43 PRINT:PRINT" DISASSEMBLY CØMPLETE":END
44 FØRHH=1TØ32:PRINT:NEXTHH
45 PRINT"# DISASSEMBLER 4.0 #":PRINT:PRINT
46 INPUT"SPATIE: JA=1 NEE=2";JL:PRINT:PRINT
47 T=15:DIMC$(T),W$(255),Ø$(255)
48 FØRW=ØTØ255:READØ$(W):NEXTW
49 FØRW=ØTØ255:READW$(W):NEXTW
50 FØRI=ØTØT:READC$(I):NEXTI
51 PRINT
```

september 1982

```
52 INPUT"STARTADRES HEX";AD$
53 AN$=AD$:GØSUB94
54 A=AD:PRINT:TEL=A-1
55 INPUT" EINDADRES HEX";AD$
56 PRINT:PRINT:PRINT
57 GØSUB94:B=AD
58 PRINT
59 MN=MN+1:HG=20:IFJL=1THENHG=10
60 U=MN/HG:IFINT(U)=UTHEN41
61 TEL=TEL+1:IFJL=1THENPRINT
62 IFTEL>BTHEN43
63 GØSUB101:GØSUB109
64 E=PEEK(TEL):GØSUB107:GØSUB112
65 AD=PEEK(TEL)
66 V=VAL(Ø$(AD))
67 ØNVGØSUB82,68,87,80,80,76,76,76,76,68,68,118,68,92:GØTØ59
68 REM CØDE A
69 TEL=TEL+1:E=PEEK(TEL):Q=E
70 GØSUB107:GØSUB112
71 TEL=TEL+1:E=PEEK(TEL)
72 GØSUB107:GØSUB112
73 GØSUB116
74 GØSUB115:E=Q:GØSUB107
75 GØSUB114:PRINT"":RETURN
76 REM CØDE B
77 TEL=TEL+1:E=PEEK(TEL)
78 GØSUB107:GØSUB112
79 GØSUB116:GØSUB114:PRINT"":RETURN
80 REM CØDE C
81 GØSUB116:PRINT"":RETURN
82 REM CØDE D
83 TEL=TEL+1:E=PEEK(TEL)
84 GØSUB107:GØSUB112
85 GØSUB116
86 PRINTCHR$(35);:GØSUB114:PRINT"":RETURN
87 REM CØDE E
88 TEL=TEL+1:E=PEEK(TEL)
89 GØSUB107:GØSUB112
90 GØSUB116
91 PRINTCHR$(48);CHR$(48);:GØSUB114:PRINT"":RETURN
92 REM FØUTMELDING
93 GØSUB116:PRINT"":RETURN
94 REM HEX - DEC
95 XI=4096:AD=0
96 FØRL=1TØ4:FØRL1=0TØ15
97 IFMID$(AD$,L,1)=C$(L1)THEN99
98 NEXTL1
99 AD=AD+L1*XI:XI=XI/16
100 NEXTL:RETURN
101 REM DEC - HEX ADRES
102 J=TEL
103 J1=INT(J/4096):J2=J-(J1*4096)
104 J3=INT(J2/256):J4=J2-(J3*256)
105 J5=INT(J4/16):J6=J4-(J5*16)
106 RETURN
107 REM
108 J4=E:GØTØ105
```

```
 109 REM
 110 PRINTC$(J1);C$(J3);C$(J5);C$(J6);CHR$(32);
 111 RETURN
 112 REM
 113 PRINTC$(J5);C$(J6);CHR$(32);:RETURN
 114 REM
 115 PRINTC$(J5);C$(J6);:RETURN
 116 REM
 117 PRINTTAB(T);V$(AD);CHR$(32);:RETURN
 118 REM CODE F
 119 TEL=TEL+1:E=PEEK(TEL)
 120 GOSUB107:GOSUB112:GOSUB116
 121 J=TEL+E+1
 122 IFE<127THEN124
 123 E=256-E:J=TEL-E+1
 124 GOSUB103
 125 PRINTC$(J1)C$(J3)CS(J5)C$(J6):RETURN
READY
```

A.L. MATHLENER
KUSTVEG 4472
DELFZIJL
05960 - 15326

VERANDERINGEN AAN HET PRØGRAMMA

M Ø R S E T U T Ø R
========= =========

TE VINDEN ØP ØSI-CASSETTE NUMMER 4

ØM HET PRØGRAMMA REDELIJK TE KUNNEN LATEN LØPEN;
MØETEN ØP DE VØLGENDE REGELS AANPASSINGEN WØRDEN GEMAAKT

== REGEL 110 WØRDT REGEL 1410
== REGEL 110 WØRDT '110 FØR T=0 TØ T2:NEXT'
== VERANDER IN 130 VARIABELE TT MØET ZIJN T5
== VERANDER IN 1805; VARIABELE T5 MØET ZIJN T
== VERANDER IN 1820; VARIABELE T5 MØET ZIJN T
== REGEL 1445 WØRDT '1445 T5=TT*5'

ØM DE 'PAØWAL' AANSLUITING TE KUNNEN TØEPASSEN
ZIJN SØFTWARE- EN HARDWARE-AANPASSINGEN NØDIG

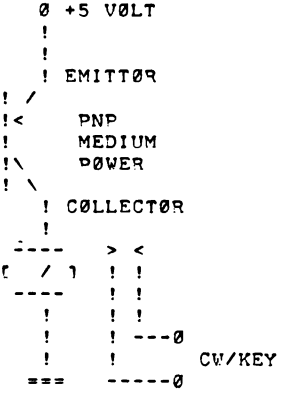
== SØFTWARE AANPASSINGEN..

---VERWIJDER ØP REGEL 15; 'GØSUB 2300'
---VERANDER ØP REGEL 70; 'Z1=64' EN 'Z2=0'
---HAAL DE 'REM'S WEG UIT DE REGELS 260 EN 310
---VERWIJDER DE REGELS; 262, 263, 265, 315, 2300
2310, 2320, 2350
---DE WAARDE '2000' ØP REGEL 1400 GELDT VØØR 2 MHZ
MACHINES, 1 MHZ MACHINES ZETTEN HIER '1000' NEER

== HARDWARE AANPASSINGEN..

---SLUIT EEN TØNE-ØSCILLATØR ØF ZENDER AAN AAN
KØNNEKTØR J2 PIN 6 (RTS)

---SCHEMA;



september 1982

6502 DISASSEMBLER

De hier afgedrukte disassembler is in principe geschikt voor iedere 6502 computer.Er hoeven slechts 5 adressen aangepast te worden.Maar eerst het programma.

Na intypen of laden via cassette van het programma wordt het gestart bij adres Ø376.
Het beeldscherm wordt gewist,en er wordt BA? _ afgedrukt.
Men moet nu het beginadres van het te disassembleren programma opgeven.Als dit gebeurt is wordt er EA? _ afgedrukt.Nu kan men het eindadres opgeven.
Tenslotte wordt er P/V? _ afgedrukt. Men heeft de keuze tussen uitvoer naar printer of video.

Direct hierna begint de uitvoer.Indien de uitvoer naar de video wordt weggeschreven zal er na 16 regels worden gewacht op een toets.Dit geeft u de gelegenheid het beeldscherm te bekijken.Na een willekeurige toets verschijnen de volgende 16 regels.Als er een CR (RETURN) wordt gegeven wordt er uit het programma gesprongen, en wel naar het indirecte adres van FFFC en FFFD.

Als de uitvoer naar een printer gaat,wordt de informatie via de SAVE-routine en de ACIA naar de printer gestuurd.Er wordt continue doorgegaan met uitvoer totdat het eindadres bereikt is.

In bovenstaande procedure kan men vrij eenvoudig wijzigingen aanbrengen om het programma aan te passen aan het eigen computersysteem.
Daartoe volgt nu een lijst van belangrijke adressen, subroutines,en zero-page geheugen,met aanwijzingen hoe één en ander werkt.

De disassembler bestaat in principe uit 3 delen;

- 1- Initialisatie deel (Ø376-Ø3DB)
- 2- Uitvoerings deel (Ø3DC-Ø54C)
- 3- Tabellen (Ø54D-Ø6FF)

1) Initialisatie: hier worden begin en eind adressen in zero-page geheugen gezet,m.b.v. de sub-routines; tekst (Ø3b5-Ø3C1)
getadres byte (Ø3C2-Ø3DB)
tabel (Ø35E-Ø375)
Verder wordt er gevraagd of de informatie naar een printer moet of naar een video. Als het de printer is geworden,wordt de SAVE-flag (Ø2Ø5) gezet,en wordt de regel-teller (ØØDB) negatief gemaakt.
Voor een video als uitvoer medium wordt er alleen 1Ø in ØØDB gezet.Dit bepaald de 16 regels op het scherm. Dit is te veranderen in adres Ø3AF en Ø536.

2) Uitvoerings deel.In dit deel wordt een stuk geheugen gedisassembleerd.Daarvoor wordt een buffer gebruikt (ØØE3-ØØEF).
Deze wordt voor iedere nieuwe regel gewist, (Ø4Ø1-Ø4ØA).Vervolgens wordt het adres in ØØEØ en ØØE1 omgezet naar ASCII,en in de buffer gezet.Subroutine (05AB-05c4) zet een hex byte in A om in twee ASCII-codes in Y en A.
Nu pas begint de eigenlijke disassembler. De werking komt globaal hierop neer;de inhoud van het adres van de adrespointer (E0,E1) wordt vergeleken met de tabellen, hier wordt alle informatie uitgehaald, zoals;lengte van de instructie,mnemonic, en de adresseermethode.Dit gebeurt allemaal tussen (Ø421-Ø5AA).Het programma van (Ø4ØF-

04B8) zet de juiste tekens i.v.m. de adresseermethode in de buffer.Vervolgens wordt de mnemonic in de buffer geplaatst, (Ø4CB-Ø4DC).De opcode,en eventuele operands maken het geheel compleet.

De afdrukroutine (0510-051F) drukt de hele buffer af. Dan wordt de adrespointer opgehoogd,aan de hand van de byte-teller (ØØDE).Dit gebeurt bij adres (Ø539-Ø551). (Ø52Ø-Ø534) houdt het aantal regels bij,en wacht eventueel op een toets.
Tenslotte zijn er nog een aantal hulproutines,zoals voor het omrekenen van relatieve adressen naar absolute adressen, (Ø552-Ø581).Ook is er nog een routine die een byte die geen instructie is met een vraagteken afdrukt.(Ø3DC-Ø4ØØ).

Het is mogelijk het programma naar een ander stuk geheugen te schuiven.Alleen moet men wel zorgen dat alle low-bytes op dezelfde plaats blijven staan. Dit moet,omdat men anders de indirecte sprongadressen in de tabel ook moet aanpassen, en dit brengt erg veel veranderingen in het programma met zich mee.Voor wie het toch wil,deze low bytes staan in de tabel (Ø6FØ-Ø6FF).Het high-byte staat in Ø442.

De input buffer kan overal in RAM worden geplaatst,omdat er van absolute adressering gebruik is gemaakt.

3)Tabel; (Ø5c5-Ø6ØØ) nodig om juiste entrypoints voor andere tabellen te vinden.
(0608-06AF) Letters van de mnemonics
(06B0-06DF) nodig om juiste entrypoints voor andere tabellen te vinden.
(06E0-06EF) lengte van een instructie
(06F0-06FF) low byte indirecte sprongadres.

Ten slotte nog de 5 subroutines die eventueel aangepast moeten worden;

FA1B geeft een clear screen
FDØØ wacht op toets,en geeft de ascii waarde aan A
FF69 general output routine
FE93 zet een hex getal in ASCII om naar hex.
FFFC indirecte sprong adres (reset)

Zero-page geheugen; D6 eindadres low
D7 eindadres high
D8 hulp geheugen
D9 idem
DA byte teller
DB regelteller
DD hulp geheugen
DE byteteller
DF hulp geheugen
EØ start adres low
E1 start adres high
E2 hulp geheugen
E3
- buffer
FF

Totaal is het programma 93Ø bytes lang.

Bron: Programma für Kleincomputer und Taschenrechner,
Sonderheft 31 der Funkschau, Martin Görlitz (disaasembler)
6502 disassembler für KIM.

Aangevuld en aangepast voor OSI-gebruikers door

J.A. van der Linden
Veestraat 6
8921 NC Leeuwarden

```
0376 201BFA JSR $FA1B
0379 A210   LDX #$10
037B 20B503 JSR $03B5
037E 20C203 JSR $03C2
0381 85E1   STA $E1
0383 20C203 JSR $03C2
0386 85E0   STA $E0
0388 A208   LDX #$08
038A 20B503 JSR $03B5
038D 20C203 JSR $03C2
0390 85D6   STA $D6
0392 20C203 JSR $03C2

0395 85D7   STA $D7
0397 A200   LDX #$00
0399 20B503 JSR $03B5
039C 2000FD JSR $FD00
039F C950   CMP #$50
03A1 D00B   BNE $03AE

03A3 A9FF   LDA #$FF
03A5 85DB   STA $DB
03A7 A901   LDA #$01
03A9 8D0502 STA $0205
03AC D004   BNE $03B2
03AE A910   LDA #$10
03B0 85DB   STA $DB
03B2 4C0104 JMP $0401
03B5 BD5E03 LDA $035E,X
03B8 C9FF   CMP #$FF
03BA F015   BEQ $03D1
03BC 2069FF JSR $FF69
03BF E8     INX
03C0 D0F3   BNE $03B5
03C2 20D203 JSR $03D2
03C5 0A     ASL A
03C6 0A     ASL A
```

```
03C7 0A      ASL A
03C8 0A      ASL A
03C9 85E2    STA $E2
03CB 20D203  JSR $03D2
03CE 18      CLC
03CF 65E2    ADC $E2
03D1 60      RTS
03D2 2000FD  JSR $FD00
03D5 2069FF  JSR $FF69
03D8 2093FE  JSR $FE93
03DB 60      RTS
03DC A219    LDX #$19
03DE A920    LDA #$20
03E0 9DE600  STA $00E6,X
03E3 CA      DEX
03E4 D0F8    BNE $03DE
03E6 A93F    LDA #$3F
03E8 8DFF00  STA $00FF
03EB A000    LDY #$00
03ED B1E0    LDA ($E0),Y
03EF 20AB05  JSR $05AB
03F2 8CEA00  STY $00EA
03F5 8DEB00  STA $00EB

03F8 A901    LDA #$01
03FA 85DE    STA $DE
03FC 85DA    STA $DA
03FE 4C1005  JMP $0510
0401 A21D    LDX #$1D
0403 A920    LDA #$20
0405 9DE200  STA $00E2,X
0408 CA      DEX
0409 D0F8    BNE $0403
040B A5E1    LDA $E1
040D 20AB05  JSR $05AB
0410 8CE300  STY $00E3
0413 8DE400  STA $00E4
0416 A5E0    LDA $E0
0418 20AB05  JSR $05AB
041B 8CE500  STY $00E5
041E 8DE600  STA $00E6
0421 A200    LDX #$00
0423 A000    LDY #$00
0425 B1E0    LDA ($E0),Y
0427 3DC805  AND $05C8,X
042A 5DD805  EOR $05D8,X
042D F00D    BEQ $043C

042F E8      INX
0430 E010    CPX #$10
0432 D0EF    BNE $0423
0434 A93F    LDA #$3F
0436 8DFF00  STA $00FF
0439 4CDC03  JMP $03DC

043C BDE006  LDA $06E0,X
043F 85DE    STA $DE
0441 A904    LDA #$04
0443 85DD    STA $DD
0445 BDF006  LDA $06F0,X
0448 85DC    STA $DC
044A A6DE    LDX $DE
044C CA      DEX
044D F01D    BEQ $046C
044F CA      DEX
0450 F00D    BEQ $045F
0452 A002    LDY #$02
0454 B1E0    LDA ($E0),Y
0456 20AB05  JSR $05AB
0459 8CF900  STY $00F9
045C 8DFA00  STA $00FA
045F A001    LDY #$01

0461 B1E0    LDA ($E0),Y
0463 20AB05  JSR $05AB
0466 8CFB00  STY $00FB
0469 8DFC00  STA $00FC
046C 6CDC00  JMP ($00DC)
046F A923    LDA #$23
0471 8DF900  STA $00F9
0474 A924    LDA #$24
0476 D002    BNE $047A
0478 A941    LDA #$41
047A D01F    BNE $049B
047C A929    LDA #$29
047E 8DFD00  STA $00FD
0481 A92C    LDA #$2C
0483 8DFE00  STA $00FE
0486 A959    LDA #$59
0488 D00C    BNE $0496
048A A92C    LDA #$2C
048C 8DFD00  STA $00FD
048F A958    LDA #$58
0491 8DFE00  STA $00FE
0494 A929    LDA #$29
0496 8DFF00  STA $00FF

0499 A928    LDA #$28
049B 8DFA00  STA $00FA
049E D017    BNE $04B7
04A0 A958    LDA #$58
04A2 D002    BNE $04A6
04A4 A959    LDA #$59
04A6 8DFE00  STA $00FE
04A9 A92C    LDA #$2C
04AB D007    BNE $04B4
04AD A928    LDA #$28
04AF 8DF800  STA $00F8
04B2 A929    LDA #$29
04B4 8DFD00  STA $00FD
```

```
04B7 A000   LDY #$00
04B9 B1E0   LDA ($E0),Y
04BB A220   LDX #$20
04BD DDE705 CMP $05E7,X
04C0 F009   BEQ $04CB
04C2 CA     DEX
04C3 D0F8   BNE $04BD
04C5 4C8205 JMP $0582
04C8 4C5405 JMP $0554
04CB BD0706 LDA $0607,X

04CE 8DF500 STA $00F5
04D1 BD2706 LDA $0627,X
04D4 8DF600 STA $00F6
04D7 BD4706 LDA $0647,X
04DA 8DF700 STA $00F7
04DD A5DE   LDA $DE
04DF 85DA   STA $DA
04E1 A000   LDY #$00
04E3 B1E0   LDA ($E0),Y
04E5 20AB05 JSR $05AB
04E8 8CEA00 STY $00EA
04EB 8DEB00 STA $00EB
04EE C6DA   DEC $DA
04F0 F01E   BEQ $0510
04F2 A001   LDY #$01
04F4 B1E0   LDA ($E0),Y
04F6 20AB05 JSR $05AB
04F9 8CED00 STY $00ED
04FC 8DEE00 STA $00EE
04FF C6DA   DEC $DA
0501 F00D   BEQ $0510
0503 A002   LDY #$02
0505 B1E0   LDA ($E0),Y

0507 20AB05 JSR $05AB
050A 8CF000 STY $00F0
050D 8DF100 STA $00F1
0510 206CA8 JSR $A86C
0513 A200   LDX #$00
0515 BDE300 LDA $00E3,X
0518 2069FF JSR $FF69
051B E8     INX
051C E01D   CPX #$1D
051E D0F5   BNE $0515
0520 A5DB   LDA $DB
0522 3015   BMI $0539
0524 38     SEC
0525 E901   SBC #$01
0527 85DB   STA $DB
0529 D00E   BNE $0539
052B 2000FD JSR $FD00
052E C90D   CMP #$0D
0530 D003   BNE $0535
0532 6CFCFF JMP ($FFFC)

0535 A910   LDA #$10
0537 85DB   STA $DB
0539 E6E0   INC $E0

053B D002   BNE $053F
053D E6E1   INC $E1
053F A5E0   LDA $E0
0541 C5D7   CMP $D7
0543 D006   BNE $054B
0545 A5E1   LDA $E1
0547 C5D6   CMP $D6
0549 F0E7   BEQ $0532
054B C6DE   DEC $DE
054D D0EA   BNE $0539
054F 4C0104 JMP $0401
0552 EA     NOP
0553 EA     NOP
0554 A5E1   LDA $E1
0556 85DF   STA $DF
0558 A001   LDY #$01
055A B1E0   LDA ($E0),Y
055C 1002   BPL $0560
055E C6DF   DEC $DF
0560 18     CLC
0561 6902   ADC #$02
0563 9002   BCC $0567
0565 E6DF   INC $DF

0567 18     CLC
0568 65E0   ADC $E0
056A 9002   BCC $056E
056C E6DF   INC $DF
056E 20AB05 JSR $05AB
0571 8DFC00 STA $00FC
0574 8CFB00 STY $00FB
0577 A5DF   LDA $DF
0579 20AB05 JSR $05AB
057C 8CF900 STY $00F9
057F 4C9B04 JMP $049B
0582 A218   LDX #$18
0584 A000   LDY #$00
0586 B1E0   LDA ($E0),Y
0588 3DAF06 AND $06AF,X
058B 5DC706 EOR $06C7,X
058E F006   BEQ $0596
0590 CA     DEX
0591 D0F3   BNE $0586
0593 4CDC03 JMP $03DC
0596 BD6706 LDA $0667,X
0599 8DF500 STA $00F5
059C BD7F06 LDA $067F,X

059F 8DF600 STA $00F6
05A2 BD9706 LDA $0697,X
```

```
05A5 8DF700 STA $00F7
05A8 4CDD04 JMP $04DD
05AB AA     TAX
05AC 4A     LSR A
05AD 4A     LSR A
05AE 4A     LSR A
05AF 4A     LSR A
05B0 20B905 JSR $05B9
05B3 A8     TAY
05B4 8A     TXA
05B5 20B905 JSR $05B9
05B8 60     RTS
05B9 290F   AND #$0F
05BB C90A   CMP #$0A
05BD 18     CLC
05BE 3002   BMI $05C2
05C0 6907   ADC #$07
05C2 6930   ADC #$30
05C4 60     RTS
```

```
       E  F  0  1  2  3  4  5  6  7  8  9  A  B  C  D
035E  0D 0A 50 2F 56 3F 20 FF 0D 0A 45 41 3F 20 20 FF
036E  0D 0A 42 41 3F 20 20 FF 20 1B FA A2 10 20 B5 03
037E  20 C2 03 85 E1 20 C2 03 85 E0 A2 08 20 B5 03 20
038E  C2 03 85 D6 20 C2 03 85 D7 A2 00 20 B5 03 20 00
039E  FD C9 50 D0 0B A9 FF 85 DB A9 01 8D 05 02 D0 04
03AE  A9 10 85 DB 4C 01 04 BD 5E 03 C9 FF F0 15 20 69
03BE  FF E8 D0 F3 20 D2 03 0A 0A 0A 0A 85 E2 20 D2 03
03CE  18 65 E2 60 20 00 FD 20 69 FF 20 93 FE 60 A2 19
03DE  A9 20 9D E6 00 CA D0 F8 A9 3F 8D FF 00 A0 00 B1
03EE  E0 20 AB 05 8C EA 00 8D EB 00 A9 01 85 DE 85 DA
03FE  4C 10 05 A2 1D A9 20 9D E2 00 CA D0 F8 A5 E1 20
040E  AB 05 8C E3 00 8D E4 00 A5 E0 20 AB 05 8C E5 00
041E  8D E6 00 A2 00 A0 00 B1 E0 3D C8 05 5D D8 05 F0
042E  0D E8 E0 10 D0 EF A9 3F 8D FF 00 4C DC 03 BD E0
043E  06 85 DE A9 04 85 DD BD F0 06 85 DC A6 DE CA F0
044E  1D CA F0 0D A0 02 B1 E0 20 AB 05 8C F9 00 8D FA
045E  00 A0 01 B1 E0 20 AB 05 8C FB 00 8D FC 00 6C DC
046E  00 A9 23 8D F9 00 A9 24 D0 02 A9 41 D0 1F A9 29
047E  8D FD 00 A9 2C 8D FE 00 A9 59 D0 0C A9 2C 8D FD
048E  00 A9 58 8D FE 00 A9 29 8D FF 00 A9 28 8D FA 00
049E  D0 17 A9 58 D0 02 A9 59 8D FE 00 A9 2C D0 07 A9
04AE  28 8D F8 00 A9 29 8D FD 00 A0 00 B1 E0 A2 20 DD
04BE  E7 05 F0 09 CA D0 F8 4C 82 05 4C 54 05 BD 07 06
04CE  8D F5 00 BD 27 06 8D F6 00 BD 47 06 8D F7 00 A5
04DE  DE 85 DA A0 00 B1 E0 20 AB 05 8C EA 00 8D EB 00
04EE  C6 DA F0 1E A0 01 B1 E0 20 AB 05 8C ED 00 8D EE
04FE  00 C6 DA F0 0D A0 02 B1 E0 20 AB 05 8C F0 00 8D
050E  F1 00 20 6C A8 A2 00 BD E3 00 20 69 FF E8 E0 1D
051E  D0 F5 A5 DB 30 15 38 E9 01 85 DB D0 0E 20 00 FD
052E  C9 0D D0 03 6C FC FF A9 10 85 DB E6 E0 D0 02 E6
053E  E1 A5 E0 C5 D7 D0 06 A5 E1 C5 D6 F0 E7 C6 DE D0
054E  EA 4C 01 04 EA EA A5 E1 85 DF A0 01 B1 E0 10 02
```

```
055E  C6 DF 18 69 02 90 02 E6 DF 18 65 E0 90 02 E6 DF
056E  20 AB 05 8D FC 00 8C FB 00 A5 DF 20 AB 05 8C F9
057E  00 4C 9B 04 A2 18 A0 00 B1 E0 3D AF 06 5D C7 06
058E  F0 06 CA D0 F3 4C DC 03 BD 57 06 8D F5 00 BD 7F
059E  06 8D F6 00 BD 97 06 8D F7 00 4C DD 04 AA 4A 4A
05AE  4A 4A 20 B9 05 A8 8A 20 B9 05 60 29 0F C9 0A 18
05BE  30 02 69 07 69 30 60 FF FF D7 FF FF FF DF 9F 1F
05CE  8D 1F 1F 1C 1C 1C 1C 19 05 14 20 6C BE 96 0A 10
05DE  80 11 01 04 0C 14 1C 19 00 00 00 90 B0 F0 30 D0
05EE  10 50 70 18 D8 58 B8 CA 88 E8 C8 EA 48 08 68 28
05FE  40 60 AA A8 BA 8A 9A 98 38 F8 42 42 42 42 42 42
060E  42 42 42 43 43 43 43 44 44 49 49 4E 50 50 50 50
061E  52 52 54 54 54 54 54 54 53 53 52 43 43 45 4D 4E
062E  50 56 56 4C 4C 4C 4C 45 45 4E 4E 4F 48 48 4C 4C
063E  54 54 41 41 53 58 58 59 45 45 4B 43 53 51 49 45
064E  4C 43 53 43 44 49 56 58 59 58 59 50 41 50 41 50
065E  49 53 58 59 59 41 53 41 43 44 45 53 53 52 4F 4C
066E  4C 4C 4C 43 41 41 41 4A 53 53 52 49 44 43 43 42
067E  4A 53 4F 54 42 4F 52 53 44 44 44 4D 53 4E 44 4D
068E  54 54 4F 4E 45 50 50 49 53 45 52 41 43 4C 41 52
069E  59 58 41 50 4C 44 43 50 59 58 52 43 43 58 59 54
06AE  52 49 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 E3 DF
06BE  E7 E7 E3 E7 E7 F3 F3 F7 FF FF 41 81 E1 22 01 42
06CE  A0 A2 A1 C1 02 21 61 4C 84 86 62 E6 C6 E0 C0 24
06DE  20 78 03 03 03 02 01 02 02 02 02 02 03 02 03 03
06EE  01 02 B7 AD A4 A4 78 C8 6F 7C 8A B7 B7 A0 A0 A4
06FE  B7 6F
```

FORTH - HEXDUMP OP HET SCHERM.

```
SCR # 98
  0 (64 BYTES WORDEN OP HET SCHERM GESCHREVEN)
  1 HEX   VARIABLE CC1   VARIABLE CC2
  2 VARIABLE CC3  VARIABLE VIDEO  VARIABLE BYTE
  3 68 CC1 ! 8 CC2 ! D1D8 CC3 ! D1D8 VIDEO !
  4 : HAAL BYTE C@  DUP ;
  5 ( 1 BYTE WORDT OPGEHAALD EN 2-MAAL OP STACK GEZET)
  6 : HOOG 10 / DUP ;  ( 4 HOGE BITS EXTRA OP STACK)
  7   LAAG 10 * - ;        ( 4 LAGE BITS VAN BYTE IN TOS)
  8   KARAKTER DUP 9 > IF 7 + ENDIF 30 + ;
  9   ( VAN HEX-WAARDE NAAR ASCII-WAARDE)
 10 : HOGERVIDEO 1 VIDEO +! ;  ( HOOG SCHERMADRES OP)
 11 : HOGERBYTE 1 BYTE +! ;    ( HOOG GEHEUGENADRES OP)
 12 : WEERVIDEO CC3 VIDEO ! ;  ( HERSTEL SCHERMADRES)
 13 : SCHRIJF KARAKTER VIDEO C! HOGERVIDEO ;
 14 ( SCHRIJF ASCII-KARAKTER OP SCHERM, HOOG SCHERMADRES OP)
 15 -->
OK
```

```
SCR # 99
  0 ( FORTH-HEXDUMP VERVOLG)
  1 : VIDEOBIJ CC1 VIDEO ! ;  ( SLA REGEL OVER BEGIN VOORAAN)
  2 : SPATIE 20 VIDEO C! HOGERVIDEO ;
  3 ( SCHRIJF SPATIE OP SCHERM EN HOOG SCHERMADRES OP)
  4 : EENBYTE HAAL HOOG SCHRIJF LAAG SCHRIJF SPATIE HOGERBYTE
  5 ( SCHRIJF BYTE PLUS SPATIE OP HET SCHERM)
  6 : EENREGEL CC2 @ DO EENBYTE LOOP VIDEOBIJ ;
  7 ( SCHRIJF REGEL EN SLA REGEL OVER)
  8 : EENPAGINA WEERVIDEO 8 0 DO EENREGEL LOOP ;
  9 ( SCHRIJF 8 REGELS OP HET SCHERM)
 10 : 1P 1A EMIT BYTE .  EENPAGINA ;
 11 ( MAAK SCHERM SCHOON - SCRIJF ADRES 1E BYTE EN 64 BYTES)
 12 : HEDU BYTE ! BEGIN 1P KEY 20 = IF UNTIL 1A EMIT ;
 13 ( HEDU VERWACHT ADRES EERSTE BYTE OP STACK)
 14 ( TOETS GEEFT NIEUWE PAGINA - SPATIEBALK GEEFT EINDE)
 15 ;S
OK
```

HEDU IS GEEN GEWELDIG PROGRAMMA, WANT HET GEBRUIKT LANG
NIET ALLE MOGELIJKHEDEN VAN FORTH. MAAR HET WERKT.
OM HET TE LATEN LOPEN MOET U INTIKKEN:
HHLL HEDU <RETURN>
HHLL IS HET ADRES VANAF WAAR DE DUMPING GESCHIEDT.

```
63000 REM  *** OSI MINISPACER ***
63010 REM
63020 REM  AUTEUR: HANS DE JONG; 14-4-'82
63030 REM
63040 REM  DIT PROGRAMMA MAAKT EEN LISTING VAN
63050 REM  EEN BASIC-PROGRAMMA DAT ZICH IN HET
63060 REM  GEHEUGEN BEVINDT EN DRUKT DAARBIJ
63070 REM  OP DE GEWENSTE PLAATSEN SPATIES AF.
63080 REM
63090 REM  VEREIST: EEN BASIC-ROM 3 WAARIN HET
63100 REM  GARBAGE-COLLECTING PROBLEEM IS OP-
63110 REM  GELOST.
63120 REM
63500 GOSUB 63760
63550 O$= STR$( FNA(A0+2))+" "
63560 S1= 1: S2= 1
63570 FOR CP= A0+4 TO FNA(A0)-2
63580 CH= PEEK(CP)
63590 IF CH= DA OR CH= RE THEN S1= ZE
63600 IF CH>LB THEN CC$= K$(CH-HB): GOTO 63660
63610 CC$= CHR$(CH)
63620 IF S1= ZE THEN 63680
63630 IF CH= QU THEN S2= EN-S2
63640 IF S2= ZE THEN 63680
63650 IF CC$= ":" OR CC$= "," OR CC$= ";" THEN CC$= CC$+I$
63660 IF RIGHT$(O$, EN)<>" " THEN 63680
63670 IF LEFT$(CC$, EN)= " " THEN CC$= MID$(CC$, 2)
63680 O$= O$+CC$
63690 NEXT CP
63700 CT= LEN(O$)
63710 IF CT<= RL THEN 63730
63720 PRINT CHR$(7)"REGEL TE LANG:"CT"CH.:": PRINT O$: STOP
63730 PRINT O$
63740 A0= FNA(A0)
63745 IF FNA(A0)<>0 AND FNA(A0+2)<>63000 THEN 63550
63750 POKE 515, 0: POKE 517, 0: END
63760 DA= 131: RE= 142: ZE= 0: QU= 34: HB= 128: LB= 127: EN= 1
63770 DEF FNA(A0)= PEEK(A0)+256* PEEK(A0+1)
63780 A0= 769: REM  START BASICPROGRAMMA
63790 DIM K$(67)
63800 FOR I= 0 TO 67
63810 READ K$(I)
63820 NEXT I
63830 DATA " END "," FOR "," NEXT "," DATA "," INPUT "
63840 DATA " DIM "," READ "," LET "," GOTO "," RUN "
63850 DATA " IF "," RESTORE "," GOSUB "," RETURN "," REM "
63860 DATA " STOP "," ON "," NULL "," WAIT "," LOAD "
63870 DATA " SAVE "," DEF "," POKE "," PRINT "," CONT "
63880 DATA." LIST "," CLEAR "," NEW "," TAB("," TO "
63890 DATA " FN"," SPC("," THEN "," NOT "," STEP "
63900 DATA +,-,*,/,^
63910 DATA " AND "," OR ",>,"= ",<
63920 DATA " SGN"," INT"," ABS"," USR"," FRE"
63930 DATA " POS"," SQR"," RND"," LOG"," EXP"
63940 DATA " COS"," SIN"," TAN"," ATN"," PEEK"
63950 DATA " LEN"," STR$"," VAL"," ASC"," CHR$"
63960 DATA " LEFT$"," RIGHT$"," MID$"
63970 RL= 71: REM  MAXIMUM REGELLENGTE VOOR BASIC VERWERKBAAR
63980 I$= " "
63990 RETURN
```

TITLE "TEXT - SEARCHER

AUTHOR : "A. MATHLENER

MET BEHULP VAN DIT PROGRAMMA KAN MEN IN HET GEHELE GEHEUGENBEREIK
ZOEKEN NAAR EEN STUK TEKST.
INDIEN HET PROGRAMMA STOPT ZONDER 'OK' AF TE DRUKKEN KAN MEN HET
PROGRAMMA LATEN STOPPEN DOOR <REPEAT> IN TE DRUKKEN OF HET PROGRAMMA
VERVOLGEN DOOR <ESC> IN TE DRUKKEN.

VERKLARING PROGRAMMA:
```
 10 - 100   INITIALISERING PROGRAMMA
110 - 220   LEZEN EN AFDRUKKEN VAN GEHEUGENPLAATS
230 - 280   CONVERSIE ADRES VAN HEX NAAR DEC
290 - 310   KEUZE STOPPEN OF VERVOLGEN VAN 'T PROGRAMMA
320 - 370   CONVERSIE EN AFDRUKKEN VAN ADRES
380         EINDE PROGRAMMA
```

```
 10 FOR X = 1 TO 32 : PRINT   NEXT
 20 PRINT "  TEXT-SEARCHER"
 30 T = 15 : DIM C$(T)
 40 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
 50 FOR I = 0 TO T : READ C$(I)   NEXT
 60 PRINT
 70 INPUT"SPACE YES=1 NO=2 "; JL
 80 PRINT
 90 INPUT "STARTADDRESS HEX "; AD$
100 PRINT
110 GOSUB 230
120 A = AD : TEL = A - 1
125 PRINT
130 MN = MN + 1 : HG = 20 : IF JL = 1 THEN HG = 10
140 U = MN / HG : IF INT(U) = U THEN 290
150 TEL = TEL + 1 : IF JL = 1 THEN PRINT
160 GOSUB320
170 FOR X = 0 TO T
175 IF TEL + X > 65535 THEN 380
180 E = PEEK(TEL + X)
190 PRINT CHR$(E);
200 NEXT
210 TEL = TEL + X - 1
220 PRINT "" : GOTO 130
230 XI = 4096 : AD = 0
240 FOR L = 1 TO 4   FOR L1 = 0 TO T
250 IF MID$(AD$ , L , 1) = C$(L1) THEN 270
260 NEXT L1
270 AD = AD + L1 * XI   XI = XI / 16
280 NEXT L : RETURN
290 IF PEEK(57100) = 222 THEN 125
300 IF PEEK(57100) = 126 THEN 380
310 GOTO 290
320 J = TEL
330 J1 = INT(J/4096) : J2 = J - (J1*4096)
340 J3 = INT(J2/256) : J4 = J2 - (J3*256)
350 J5 = INT(J4/16)  : J6 = J4 - (J5*16)
360 PRINT C$(J1); C$(J3) ; C$(J5) ; C$(6) ; CHR$(32) ;
370 RETURN
380 PRINT : END
```

TITLE "HEX - SEARCHER

AUTHOR "A. MATHLENER

MET BEHULP VAN DIT PROGRAMMA KAN MEN IN HET GEHELE GEHEUGENBEREIK GEHEUGENCELLEN UITLEZEN.
INDIEN HET PROGRAMMA STOPT ZONDER 'OK' AF TE DRUKKEN KAN MEN HET PROGRAMMA LATEN STOPPEN DOOR <ESC> IN TE DRUKKEN OF HET PROGRAMMA VERVOLGEN DOOR <REPEAT> IN TE DRUKKEN.

VERKLARING PROGRAMMA:

```
 10 - 110   INITIALISERING PROGRAMMA
120 - 160   INDELING BEELDSCHERM; AFDRUKKEN ADRES
170 - 260   AFDRUKKEN INHOUD VAN GEHEUGENCELLEN
270 - 320   CONVERSIE HEX NAAR DEC
330 - 350   KEUZE STOPPEN OF VERVOLGEN VAN 'T PROGRAMMA
360 - 400   ADRESCONVERSIE VAN DEC NAAR HEX
410 - 430   CONVERSIE GEHEUGENCEL VAN DEC NAAR HEX
440         END
```

```
 10 FOR X = 1 TO 32 : PRINT   NEXT
 20 PRINT" HEX - SEARCHER"
 30 T = 15 : DIM C$(T)
 40 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
 50 FORI = 0 TO T : READ C$(I) : NEXT
 60 PRINT
 70 INPUT"EXPAND YES = 1  NO = 2"; JL
 80 PRINT
 90 INPUT "START ADDRESS HEX "; AD$
 100 PRINT
 110 GOSUB270
 120 A = AD : TEL = A - 1
 130 MN = MN + 1 : HG = 20 : IF JL = 1 THEN HG = 10
 140 U = MN / HG : IF INT(U) = U THEN 330
 150 TEL = TEL + 1 : IF JL = 1 THEN PRINT
 160 GOSUB360 : GOSUB410
 170 A = 8
 180 IF JL = 1 THEN A = 5
 190 FOR X = 0 TO A
 210 E = PEEK(TEL + X) : GOSUB 430
 220 PRINT C$(J5) ; C$(J6) ; CHR$(32) ;
 230 IF JL = 1 THEN PRINT CHR$(32) ;
 240 NEXT
 250 TEL = TEL + X - 1
 260 PRINT"" : GOTO 130
 270 XI = 4096 : AD = 0
 280 FOR L = 1 TO 4 : FOR L1 = 0 TO T
 290 IF MID$(AD$ , L , 1) = C$(L1) THEN 310
 300 NEXT L1
 310 AD = AD + L1 * XI   XI = XI / 16
 320 NEXT L : RETURN
 330 IF PEEK(57100) = 222 THEN 440
 340 IF PEEK(57100) = 126 THEN 120
 350 GOTO 330
 360 J = TEL
 370 J1 = INT(J / 4096) : J2 = J - (J1 * 4096)
 380 J3 = INT(J2 / 256) : J4 = J2 - (J3 * 256)
 390 J5 = INT(J4 / 16)  : J6 = J4 - (J5 * 16)
 400 RETURN
 410 PRINT C$(J1) ; C$(J3) ; C$(J5) ; C$(J6) ; CHR$(32) ;
 420 RETURN
 430 J4 = E : GOTO 390
 440 END
```

NOVEMBER 1982

```
SCR # 1
  0 ( ASSEMBLER)
  1 HEX
  2 VOCABULARY ASSEMBLER IMMEDIATE
  3 ASSEMBLER DEFINITIONS
  4 ( REGISTER ASSIGNMENT)
  5 B5 CONSTANT XSAVE    B1 CONSTANT W
  6 B3 CONSTANT UP       AE CONSTANT IP
  7 A6 CONSTANT N
  8 ( NUCLEUS LOCATIONS)
  9 ' (DO) 0E + CONSTANT POP
 10 ' (DO) 0C + CONSTANT POPTWO
 11 ' LIT  13 + CONSTANT PUT
 12 ' LIT  11 + CONSTANT PUSH
 13 ' LIT  18 + CONSTANT NEXT
 14 ' EXECUTE NFA 11 - CONSTANT SETUP
 15 -->

SCR # 2
  0 ( ASSEMBLER CONTINUED)
  1 0 VARIABLE INDEX -2 ALLOT
  2 0909 , 1505 , 0115 , 8011 , 8009 , 1D0D ,
  3 8019 , 8080 , 0080 , 1404 , 8014 , 8080 ,
  4 8080 , 1C0C , 801C , 2C80 ,
  5 2 VARIABLE MODE
  6 : .A 0 MODE ! ;    : # 1 MODE ! ;
  7 : MEM 2 MODE ! ;   : ,X 3 MODE ! ;
  8 : ,Y 4 MODE ! ;    : X) 5 MODE ;
  9 : )Y 6 MODE ! ;    : ) F MODE ! ;
 10   BOT ,X 0 ;  ( BOTTOM OF STACK)
 11   SEC ,X 2 ;  ( SECOND STACK ITEM)
 12   RP) ,X 101 ;  ( BOTTOM RETURN STACK)
 13
 14 -->
 15

SCR # 3
  0 ( ASSEMBLER CONTINUED)
  1
  2   UPMODE IF MODE @ 8 AND 0= IF 8 MODE +!
  3   THEN THEN 1 MODE @ 0F AND -DUP IF 0 DO DUP
  4   + LOOP THEN OVER 1+ @ AND 0= ;
  5
  6 : CPU <BUILDS C, DOES> C@ C, MEM ;
  7 00 CPU BRK, 18 CPU CLC, D8 CPU CLD,
  8 58 CPU CLI, B8 CPU CLV, CA CPU DEX,
  9 88 CPU DEY, E8 CPU INX, C8 CPU INY,
 10 EA CPU NOP, 48 CPU PHA, 08 CPU PHP,
 11 68 CPU PLA, 28 CPU PLP, 40 CPU RTI,
 12 60 CPU RTS, 38 CPU SEC, F8 CPU SED,
 13 78 CPU SEI, AA CPU TAX, A8 CPU TAY,
 14 BA CPU TSX, 8A CPU TXA, 9A CPU TXS,
 15 98 CPU TYA,    -->
```

```
SCR # 4
  0 ( ASSEMBLER CONTINUED)
  1
  2 : M/CPU <BUILDS C, , DOES>
  3 DUP 1+ @ 80 AND IF 10 MODE +! THEN OVER
  4 FF00 AND UPMODE UPMODE IF MEM CR LATEST ID.
  5 3 ERROR THEN C@ MODE C@ INDEX +
  6 C@ + C, MODE C@ 7 AND IF MODE C@
  7 0F AND 7 < IF C, ELSE , THEN THEN MEM ;
  8
  9 -->
 10
 11
 12
 13
 14
 15

SCR # 5
  0 ( ASSEMBLER CONTINUED)
  1
  2 1C6E 60 M/CPU ADC,   1C6E 20 M/CPU AND,
  3 1C6E C0 M/CPU CMP,   1C6E 40 M/CPU EOR,
  4 1C6E A0 M/CPU LDA,   1C6E 00 M/CPU ORA,
  5 1C6E E0 M/CPU SBC,   1C6E 80 M/CPU STA,
  6 0D0D 01 M/CPU ASL,   0C0C C1 M/CPU DEC,
  7 0C0C E1 M/CPU INC,   0D0D 41 M/CPU LSR,
  8 0D0D 21 M/CPU ROL,   0D0D 61 M/CPU ROR,
  9 0414 81 M/CPU STX,   0486 E0 M/CPU CPX,
 10 0486 C0 M/CPU CPY,   1496 A2 M/CPU LDX,
 11 0C8E A0 M/CPU LDY,   048C 80 M/CPU STY,
 12 0480 14 M/CPU JSR,   8480 40 M/CPU JMP,
 13 0484 20 M/CPU BIT,
 14
 15 : REPEAT, ?EXEC 1 ?PAIRS 4C C, , ; -->

SCR # 6
  0 ( ASSEMBLER CONTINUED)
  1 : BEGIN, HERE 1 ; IMMEDIATE
  2 : UNTIL, ?EXEC >R 1 ?PAIRS R> C, HERE
  3   1+ - C, ; IMMEDIATE
  4 : IF,     C, HERE 0 C, 2 ; IMMEDIATE
  5 : THEN,   ?EXEC 2 ?PAIRS HERE OVER C@ IF SWAP
  6   ! ELSE OVER 1+ - SWAP C! THEN ; IMMEDIATE
  7 : ELSE, 2 ?PAIRS HERE 1+ 1 JMP, SWAP HERE
  8   OVER 1+ - SWAP C! 2 ; IMMEDIATE
  9 : NOT 20 + ;
 10 90 CONSTANT CS
 11 D0 CONSTANT 0=
 12 10 CONSTANT 0<
 13 90 CONSTANT >=
 14 50 CONSTANT VS
 15 -->
```

```
SCR # 7
  0 ( ASSEMBLER CONTINUED)
  1
  2   END-CODE CURRENT @ CONTEXT !
  3   ?EXEC ?CSP SMUDGE ; IMMEDIATE
  4
  5 FORTH DEFINITIONS DECIMAL
  6 : CODE
  7   ?EXEC CREATE [COMPILE] ASSEMBLER
  8   ASSEMBLER MEM !CSP ; IMMEDIATE
  9
 10 ;S
 11
 12
 13
 14
 15


SCR # 6
  0 ( SCREEN EN PRINTER HANDLER  HANS COENEN)
  1 DECIMAL  : HOME 3 EMIT ;
  2 : PRON 1 517 C! ;                ( ZET PRINTER AAN)
  3   PROFF 0 517 C! ;               ( ZET PRINTER UIT
  4 : SMALLCHAR PRON 29 EMIT         ( KLEINE LETTERS)
  5 : BIGCHAR PRON 31 EMIT ;         ( BREDE LETTERS)
  6 : NORMCHAR PRON 30 EMIT ;        ( NORMALE LETTERS)
  7 : BREDE PRON 27 EMIT 66 EMIT ;   ( BREDE KANTLIJN)
  8 : SMALLE PRON 27 EMIT 65 EMIT ;  ( SMALLE KANTLIJN)
  9 : CLS 53248 2048 BLANKS ;        ( SNELLE SCREEN CLEAR)
 10 ;S
 11
 12
 13
 14
 15
OK


SCR # 31
  0 ( RANDOM NUMBER GENERATOR  HANS COENEN)
  1 VARIABLE RND  HERE RND !
  2 . RANDOM RND @ 31421 * 6927+ DUP RND ! ;
  3 : CHOOSE ( U1 --- U2) RANDOM U* SWAP DROP ;
  4 ;S
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
OK
```

PROGRAMMA'S MET DE INSTRUCTIE: WAIT

DOOR TON HELWIG

WAGENINGEN

```
 10 PRINT"   DRUK EENS OP EEN FUNCTIETOETS"
 20 PRINT"   OF OP CTRL+X,B,V,N,M OF DE KOMMA !"
 30 PRINT"   DAN MELD IK, WELKE TOETS(COMBINATIE) HET IS."
 40 POKE548,10:I=57088
 50 READJ,K,A$:PRINTCHR$(2);A$
 60 IFK=247THENRESTORE
 70 WAITI,J,K
 80 GOTO50
 90 REM    J   K           TOETS(EN)
100 REM  ---,---,-------------------------
110 DATA  32,222," CTRL+V OF ESC ( 1)
120 DATA 128,126," CTRL+X OF REPEAT ( 3)
130 DATA   2,252," CTRL+KOMMA OF SHIFT(R) ( 7)
140 DATA  16,239," CTRL+B ( 9)
150 DATA   4,250," SHIFT(L) OF CTRL+M ( 5)
160 DATA   2,253," CTRL+KOMMA OF SHIFT(R) ( 8)
170 DATA   4,251," SHIFT(L) OF CTRL+M ( 6)
180 DATA 128,127," CTRL+X OF REPEAT ( 4)
190 DATA  32,223," CTRL+V OF ESC ( 2)
200 DATA   8,247," CTRL+N (10)
1000 PRINT" DIT GESCHITTER STOPT, ALS"
1010 PRINT" CTRL WORDT INGEDRUKT."
1020 POKE548,6
1030 PRINT" HET PROGRAMMA GAAT VERDER, ALS U";
1040 PRINT" DRUKT OP DE "
1050 I=57088:READJ,K,A$:PRINTA$
1060 IFJ=3THENRESTORE
1070 WAITI,J,K
1080 PRINTCHR$(3):GOTO1030
1090 REM   J , K ,     TOETS
1100 REM  ---,---,-----------------------
1110 DATA127,127,C/V/B/N/M OF DE KOMMA
1120 DATA 15,239,N/M OF DE KOMMA
1130 DATA 31,223,B/N/M OF DE KOMMA
1140 DATA  7,247,M   OF DE KOMMA
1150 DATA  3,251,KOMMA
2000 K=254:I=57088
2010 READJ,A$
2020 PRINT"WAIT"I","J","K" HET PROGRAMMA GAAT VERDER,"
2030 PRINT"ALS "A$" WORDT INGEDRUKT."
2040 WAITI,J,K
2050 IFJ=2ANDK=255THENEND
2060 IFJ=2THENRESTORE:K=255
2070 PRINTCHR$(3):GOTO2010
2080 REM   J     TOETS
2090 REM  ---,----------
2100 DATA  32,ESC
2110 DATA 128,REPEAT
2120 DATA   4,SHIFT(L)
2130 DATA   2,SHIFT(R)
```

```
SCR # 1
  0 ( 1 PAGE PRT STP TEXT LINE      WFR-79MAY01)
  1 DECIMAL
  2 : PAGE 26 EMIT ;
  3
  4 : PRT 1 517 C! ;
  5 : STP CR 0 517 C! ;
  6 HEX
  7 FORTH  DEFINITIONS   HEX
  8 : TEXT          ( ACCEPT FOLLOWING TEXT TO PAD)
  9       HERE  C/L  1+   BLANKS  WORD  HERE  PAD  C/L  1+  CMOVE
 10
 11   LINE        .  ( RELATIVE TO SCR, LEAVE ADDRESS OF LINE)
 12       DUP  FFF0  AND  17  ?ERROR   ( KEEP ON THIS SCREEN)
 13       SCR  @  (LINE)  DROP  ;
 14 HEX
 15 -->

SCR # 2
  0 ( 2  LINE EDITOR                          WFR-79MAY03)
  1  VOCABULARY  EDITOR  IMMEDIATE   HEX
  2   WHERE                ( PRINT SCREEN # AND IMAGE OF ERROR)
  3     DUP  B/SCR  /  DUP  SCR  !  ." SCR # "  DECIMAL  .
  4     SWAP  C/L  /MOD  C/L  *  ROT  BLOCK  +  CR  C/L  TYPE
  5     CR  HERE  C@  -  SPACES  5E  EMIT  [COMPILE] EDITOR  QUIT
  6
  7 EDITOR  DEFINITIONS
  8 : #LOCATE            ( LEAVE CURSOR OFFSET-2, LINE-1)
  9         R#  @  C/L  /MOD  ;
 10   #LEAD                ( LINE ADDRESS-2, OFFSET-1 TO CURSOR)
 11         #LOCATE  LINE   SWAP  ;
 12   #LAG                 ( CURSOR ADDRESS-2, COUNT-1 AFTER CURSOI
 13         #LEAD  DUP  >R  +  C/L  R>  -  ;
 14   -MOVE    ( MOVE IN BLOCK BUFFER ADDR FROM-2, LINE TO-1)
 15         LINE  C/L  CMOVE  UPDATE  ;  -->

SCR # 3
  0 ( 3  LINE EDITING COMMANDS              WRF-79MAY03)
  1 : H               ( HOLD NUMBERED LINE AT PAD)
  2       LINE  PAD  1+  C/L  DUP  PAD  C!  CMOVE  ;
  3
  4   E               ( ERASE LINE-1 WITH BLANKS)
  5       LINE  C/L  BLANKS  UPDATE  ;
  6
  7   S               ( SPREAD MAKING LINE # BLANK)
  8       DUP  1  -  ( LIMIT )  0E  ( FIRST TO MOVE )
  9       DO  I  LINE  I  1+  -MOVE  -1  +LOOP  E  ;
 10
 11   D                ( DELETE LINE-1, BUT HOLD IN PAD)
 12       DUP  H  0F  DUP  ROT
 13       DO  I  1+  LINE  I  -MOVE  LOOP  E
 14
 15 -->
```

```
SCR # 4
  0 ( 4  LINE EDITING COMMANDS            WFR-79MAY03)
  1 : M        ( MOVE CURSOR BY SIGNED AMOUNT-1, PRINT ITS LINE)
  2      R#  +!  CR  SPACE  #LEAD  TYPE  5E  EMIT
  3                         #LAG  TYPE  #LOCATE     DROP
  4
  5
  6 : T          ( TYPE LINE BY #-1, SAVE ALSO IN PAD)
  7      DUP  C/L  *  R#  !  DUP  H  0  M  ;
  8
  9 : LIST  PAGE  LIST ;
 10
 11 : L             ( RE-LIST SCREEN)
 12                 SCR  @  LIST  0  M
 13 -->
 14
 15

SCR # 5
  0 ( 5 LINE EDITING COMMANDS             WFR-790105)
  1 : R                    ( REPLACE ON LINE #-1, FROM PAD)
  2       PAD  1+  SWAP  -MOVE  ;
  3
  4 : P                    ( PUT FOLLOWING TEXT ON LINE-1)
  5       1  TEXT  R  ;
  6
  7 : I                    ( INSERT TEXT FROM PAD ONTO LINE #)
  8       DUP  S  R  ;
  9
 10 : TOP                   ( HOME CURSOR TO TOP LEFT OF SCREEN)
 11       0 R# ! ;
 12 -->
 13
 14
 15 ;S

SCR # 6
  0 ( 6 SCREEN EDITING COMMANDS             WFR-79APR27)
  1 : CLEAR                ( CLEAR SCREEN BY NUMBER-1)
  2       SCR  !  10  0  DO  FORTH  I  EDITOR  E  LOOP ;
  3
  4 : FLUSH                ( WRITE ALL UPDATED BLOCKS TO DISC)
  5     [  LIMIT  FIRST  -  B/BUF  4  +  /  ]  ( NUMBER OF BUFFERS)
  6     LITERAL  0  DO  7FFF  BUFFER  DROP  LOOP  ;
  7
  8 : COPY                 ( DUPLICATE SCREEN-2, ONT SCREEN-1)
  9    B/SCR  *  OFFSET  @  +  SWAP  B/SCR  *  B/SCR  OVER  +  SWAP
 10     DO  DUP  FORTH  I  BLOCK  2  -  !  1+  UPDATE  LOOP
 11     DROP  FLUSH  ;
 12 -->
 13
 14
 15 ;S
```

```
SCR # 7
  0 ( 7 DOUBLE NUMBER SUPPORT                WFR-80APR24)
  1 ( OPERATES ON 32 BIT DOUBLE NUMBERS OR TWO 16-BIT INTEGERS)
  2 FORTH DEFINITIONS
  3
  4   2DROP   DROP   DROP  ;  ( DROP DOUBLE NUMBER)
  5
  6 : 2DUP   OVER   OVER     ( DUPLICATE A DOUBLE NUMBER)
  7
  8   2SWAP   ROT    >R    ROT   R>  ;
  9                    ( BRING SECOND DOUBLE TO TOP OF STACK)
 10 EDITOR DEFINITIONS  -->
 11
 12
 13
 14
 15 ;S

SCR # 8
  0 ( 8 STRING MATCH FOR EDITOR          PM-WFR-80APR25)
  1 : -TEXT              ( ADDRESS-3, COUNT-2, ADDRESS-1 ---)
  2  SWAP   -DUP  IF  ( LEAVE BOOLEAN MATCHED=NON-ZERO, NOPE=ZERO)
  3                OVER + SWAP   ( NEITHER ADDRESS MAY BE ZERO!)
  4          DO  DUP  C@  FORTH  I  C@  -
  5              IF 0= LEAVE ELSE 1+ THEN LOOP
  6          ELSE  DROP  0=  THEN  ;
  7   MATCH  (  CURSOR ADDRESS-4, BYTES LEFT-3, STRING ADDRESS-2, )
  8          (  STRING COUNT-1, --- BOOLEAN-2, CURSOR MOVEMENT-1)
  9   >R  >R  2DUP  R>  R>  2SWAP  OVER  +  SWAP
 10   ( CADDR-6, BLEFT-5, $ADDR-4, $LEN-3, CADDR+BLEFT-2, CADDR-1)
 11   DO  2DUP  FORTH   I   -TEXT
 12     IF  >R  2DROP  R>  -  I  SWAP  -  0  SWAP  0  0  LEAVE
 13         (  CADDR BLEFT $ADDR $LEN OR ELSE 0 OFFSET 0 0 )
 14       THEN  LOOP 2DROP  ( CADDR-2, BLEFT-1, OR 0-2, OFFSET-1)
 15     SWAP  0=  SWAP  ;  -->

SCR # 9
  0 ( 9 STRING EDITING COMMANDS          WFR-79MAR24)
  1 : 1LINE          ( SCAN LINE WITH CURSOR FOR MATCH TO PAD TEXT, )
  2                    ( UPDATE CURSOR, RETURN BOOLEAN)
  3        #LAG  PAD  COUNT  MATCH  R#  +!  ;
  4
  5 : FIND  ( STRING AT PAD OVER FULL SCREEN RANGE, ELSE ERROR)
  6     BEGIN  3FF  R#  @  <
  7         IF  TOP  PAD  HERE  C/L  1+  CMOVE  0  ERROR  ENDIF
  8         1LINE  UNTIL  ;
  9
 10   DELETE            ( BACKWARDS AT CURSOR BY COUNT-1)
 11     >R  #LAG  +  FORTH  R  -  ( SAVE BLANK FILL LOCATION)
 12     #LAG  R  MINUS  R#  +!    ( BACKUP CURSOR)
 13     #LEAD  +  SWAP  CMOVE
 14     R>  BLANKS  UPDATE  ;  ( FILL FROM END OF TEXT)
 15 -->
```

```
SCR # 10
  0 ( 10 STRING EDITOR COMMANDS              WFR-79MAR24)
  1 : N     ( FIND NEXT OCCURANCE OF PREVIOUS TEXT)
  2     FIND  0  M  ;
  3
  4 : F      ( FIND OCCURANCE OF FOLLOWING TEXT)
  5 -     1  TEXT  N  ;
  6
  7 : B      ( BACKUP CURSOR BY TEXT IN PAD)
  8       PAD C@ MINUS M ;
  9 : X     (  DELETE FOLLOWING TEXT)
 10
 11      1 TEXT FIND PAD C@ DELETE 0 M ;
 12
 13 : TILL  ( DELETE ON CURSOR LINE, FROM CORSOR TO TEXT END)
 14      #LEAD + 1 TEXT 1LINE 0= 0 ?ERROR
 15      #LEAD + SWAP - DELETE 0 M ;   -->

SCR # 11
  0 (  11 STRING EDITOR COMMANDS                  WFR-79MAR24)
  1 : C       ( SPREAD AT CURSOR AND COPY IN THE FOLLOWING TEXT)
  2      1 TEXT PAD COUNT
  3      #LAG ROT OVER MIN >R
  4      FORTH R R# +!   ( BUMP CURSOR)
  5      R - >R          ( CHARS TO SAVE)
  6      DUP HERE R CMOVE  ( FROM OLD CURSOR TO HERE)
  7      HERE #LEAD + R> CMOVE    ( HERE TO CURSOR LOCATION)
  8      R> CMOVE UPDATE    ( PAD TO OLD CURSOR)
  9      0 M    ( LOOK AT NEW LINE)  ;
 10 FORTH DEFINITIONS DECIMAL
 11 LATEST    12 +ORIGIN !  ( TOP NFA)
 12 HERE      28 +ORIGIN !  ( FENCE)
 13 HERE      30 +ORIGIN !  ( DP)
 14 ' EDITOR  6 + 32 +ORIGIN !  ( VOC-LINK)
 15 HERE FENCE !  ;S
```

'LINE EDITOR' UIT 'INSTALLATION MANUAL' VAN FIG.

ALS DE EDITOR GEIMPLEMENTEERD IS ROEPT U HEM OP MET
EDITOR <RETURN> EN U VELAAT HEM WEER MET FORTH <RETURN>
!!! MAAR ALTIJD BEGINNEN MET EMPTY-BUFFERS <RETURN>

* LINE EDITOR COMMANDS

N H ZET REGEL N IN 'PAD'
N D WIS REGEL N MAAR HOUDT HEM IN PAD
REGEL 15 WORDT BLANK TERWIJL REGELS N+1 TO 15
EEN REGEL OMHOOG GAAN
N T TYPE REGEL N EN SAVE HEM IN PAD
N R VERVANG REGEL N MET DE REGEL IN PAD
N I VOEG DE TEKST IN PAD TUSSEN OP REGEL N
DE OUDE REGEL N EN DE VOLGENDE REGELS GAAN NAAR
BENEDEN, REGEL 15 GAAT VERLOREN
N E WIS REGEL N EN VUL HEM MET BLANKS
N S SPREID BIJ REGEL N. N EN DE VOLGENDE REGELS GAAN
1 REGEL NAAR BENEDEN. REGEL N WORDT GEVULD MET
BLANKS. REGEL 15 GAAT VERLOREN.

* CURSOR POSITION COMMANDS

TOP PLAATS DE CURSOR OP DE START-POSITIE VAN HET SCREEN
N M VERPLAATS DE CURSOR OVER N POSITIES (POSITIEF OF
NEGATIEF) EN PRINT DE REGEL, WAARIN DE CURSOR ZICH
BEVINDT. DE POSITIE VAN DE CURSOR WORDT AANGEGEVEN
MET ^

* STRING EDITING COMMANDS

F 'TEKST' ZOEK VANAF DE GELDENDE CURSOR-POSITIE TOTDAT
DE STRING 'TEKST' GEVONDEN IS.
DE CURSOR WORDT GEPLAATST ACHTER DE LAATSTE POSITIE
VAN DE 'TEKST', EN DE REGEL MET CURSOR WORDT OP
SCHERM GESCHREVEN. ALS DE STRING NIET GEVONDEN WORDT,
WORDT EEN FOUTMELDING GEGEVEN EN WORDT DE CURSOR
AAN HET BEGIN VAN HET SCREEN GEPLAATST.
B GEBRUIKT NA F PLAATST DE CURSOR VOOR DE 'TEKST'
N GEEFT HET VOLGENDE OPTREDEN VAN DE 'TEKST'
X 'TEKST' ZOEKT EN WIST DE STRING 'TEKST'
C 'TEKST' COPIEER 'TEKST' VANAF DE PLAATS WAAR DE CURSOR
ZICH BEVINDT.
TILL 'TEKST' WIST IN DE CURSOR-REGEL VANAF DE POSITIE
VAN DE CURSOR TOT EN MET DE STRING 'TEKST'.

N. B. ALS U C INTIKT ZONDER 'TEKST' WORDT OP DE POSITIE
VAN DE CURSOR EEN 0 GEPLAATST. DIT ZAL DE COMPILATIE
VAN HET SCREEN ABRUPT DOEN EINDIGEN !!
DEZE FOUT WORDT OPGEHEVEN MET 'TOP X <RETURN>'

* SCREEN EDITING COMMANDS

N LIST	LIST SCREEN N EN SELECTEER DIT SCREEN VOOR EDITING.
N CLEAR	VUL SCREEN N MET BLANKS EN SELECTEER HET VOOR EDITING.
N1 N2 COPY	COPIEER SCREEN N1 NAAR SCREEN N2
L	LIST HET ONDER HANDEN ZIJNDE SCREEN. DE CURSOR-REGEL WORDT ONDER HET SCREEN NOG EENS GELIST, OM DE CURSOR-POSITIE TE TONEN.
FLUSH	GEBRUIKT AAN HET EINDE VAN HET EDIT-EN VAN EEN SCREEN OM HET GEWIJZIGDE SCREEN NAAR NAAR DISK (OF HOGE RAM) TE SCHRIJVEN.

```
5 ::::::::::::::::::::::::::::::::::::::::::::::::::RUN10
10 REM X-Y RECORDER STURING MET D/A CONVERTERS
20 REM
30 REM COPYRIGHT JOHN GLASER 033-751472
40 REM WERKT MET VIA OP $C400
50 REM INITIALISATIE
60 VIA=50176:PEN=VIA:U=0:D=1:DAC=VIA+1:SEL=VIA+12
70 POKEVIA+2,1:POKEVIA+3,255
80 A=202:B=234:REM SELECT DAC A OF B
85 POKEPEN,U:GOSUB13000:GOSUB21000
90 PRINTCHR$(26):INPUT"RECORDER KLAAR";Z$
100 GOTO500
110 X0=100:X1=200:Y0=100:Y1=200
120 GOSUB1000:PRINT"KLAAR"
130 GOSUB13000:END
200 FORZ=0TO255STEP5
210 X0=127-Z:X1=128+Z:Y0=127-Z:Y1=128+Z
220 GOSUB1000
230 NEXTZ
240 GOSUB14000:END
500 REM DRAW
510 INPUT"COORDINATEN STARTPUNT X0,Y0";X0,Y0:PRINT
520 INPUT"COORDINATEN EINDPUNT X1,Y1";X1,Y1:PRINT
530 POKEPEN,U:GOSUB21000:POKESEL,A:POKEDAC,X0:POKESEL,B
535 POKEDAC,Y0
540 GOSUB22000:POKEPEN,D:GOSUB21000
545 XV=SQR((X1-X0)^2)
550 IFY1-Y0<>0THENCT=XV/(Y1-Y0)
560 FORX2=0 TO XV
565 IF (Y1-Y0)=0THENY2=0:GOTO575
567 IFXV=0GOTO500
570 Y2=INT(X2/CT)
575 Y3=Y0+Y2:IFX1<X0THENX3=X0-X2:GOTO580
577 X3=X0+X2
580 POKESEL,A:POKEDAC,X3:POKESEL,B:POKEDAC,Y3
590 NEXT X2:POKEPEN,U:GOTO500
1000 POKEPEN,U
1010 POKESEL,B:POKEDAC,Y0
1020 POKESEL,A:POKEDAC,X0:GOSUB22000:GOSUB10000
1030 FORX=X0TOX1:GOSUB20000:POKEDAC,X:NEXTX
1040 POKESEL,B:GOSUB21000
1050 FORY=Y0TOY1:GOSUB20000:POKEDAC,Y:NEXTY
1060 POKESEL,A:GOSUB21000
1070 FORX=X1TOX0STEP-1:GOSUB20000:POKEDAC,X:NEXTX
1080 POKESEL,B:GOSUB21000
1090 FORY=Y1TOY0STEP-1:GOSUB20000:POKEDAC,Y:NEXTY
1095 POKEDAC,Y
1096 GOSUB21000
1100 POKEPEN,U:GOSUB22000
1200 RETURN
10000 POKEPEN,D:RETURN:REM PEN DOWN
11000 POKEPEN,U:POKESEL,A:RETURN
12000 POKEPEN,U:POKESEL,B:RETURN
13000 GOSUB11000:POKEDAC,0:GOSUB12000:POKEDAC,0:RETURN
14000 GOSUB11000:POKEDAC,255:GOSUB12000:POKEDAC,255:END
20000 FORT=1TO2:NEXTT:RETURN
21000 FORT=1TO100:NEXTT:RETURN
22000 FORT=1TO1000:NEXTT:RETURN
23000 FORT=1TO10000:NEXTT:RETURN
```

ZIE OOK PLOTTERS CF 1

Minidatabase voor Superboard

Om het Superboard te kunnen gebruiken voor het opslaan van bijv. adressen, is het van belang dat de adressen, of andere data, efficiënt worden opgeslagen. Men moet bepaalde data dus terug kunnen vinden volgens verschillende kenmerken zoals, om bij het voorbeeld van de adressen te blijven, naam, straat, postcode of woonplaats. We kunnen de adressen heel goed als Basic-regels opslaan, iets in de trant van:
10 HCC, Postbus 149, Voorschoten.
Met de Find-routine uit de toolkit kunnen we dan gemakkelijk bepaalde adressen opvragen:
≠F:HCC zoekt alle regels op waarin HCC staat en list die dan. Mooier zou het zijn als we, om bijv. de adressen uit te printen op een envelop, alles mooi onder elkaar konden krijgen:

```
10 HCC,
Postbus 149,
Voorschoten.
```

Door de output-vector te verleggen naar een routine die één of ander ongebruikt karakter vervangt door een Carriage Return, dus mooi onder elkaar, kunnen we dit oplossen. In het eerste stuk van het programma wordt de \ (shift L) vervangen door een CR. De edit-flag en de save-flag worden vooraf getest om geen moeilijkheden met de interpreter te krijgen.
10 HCC, Postbus 149, Voorschoten wordt dan door het voorbeeld hierboven vervangen.
De rest van de database wordt gevormd door een verbeterde versie van de Find-routine uit de toolkit. Er hoeft dan niet meer ≠F:... ingetypt te worden maar kan men volstaan met:
]: (dat is dus de shift M)
Verder wordt er door de nieuwe routine niet gelet op het verschil tussen hoofdletters en kleine letters en worden de regelnummers bij het listen weggelaten omdat die in de meeste gevallen toch niet nodig zijn. Ook wordt vóór het listen nog even het scherm schoongemaakt. Door gebrek aan geheugenruimte blijft een toolkit desondanks toch noodzakelijk!

Het gebruik:
Zoals eerder gezegd wordt het] -karakter gebruikt om de routine in te schakelen.

```
]:          list alle regels
]: WOORD    list alle regels waarin WOORD voorkomt
```

\ is een CR zonder dat een regel wordt afgebroken zoals bij de Return-toets het geval is. (\ = shift L)
Het \ -teken is echter niet zichtbaar, tenzij de computer zich in de save-,of editmode bevindt.

Opstarten van de routines:
De routine die de \ vervangt door een CR wordt opgestart door:
POKE 538,48 : POKE 539,2

De □ -Find wordt opgestart door:

Break M
02BB G
en de computer bevindt zich weer in Basic via een warme start, die door de initialisatie-routine wordt gegeven.

Na een koude start moeten beide routines weer opnieuw opgestart worden, alle vectoren zijn dan immers weer in hun oude stand gezet. De toolkit blijft bereikbaar, hij is namelijk nodig voor het programma. Wil men zonder toolkit het programma gebruiken, dan moet men het programma op een andere plaats in het geheugen zetten om de aanpassingen een plaats te kunnen geven.

De volgende aanpassingen kunnen naar believen gemaakt worden:
- Bij Save na een \ wel CR (bv. voor een printer)

```
0237  D007    BNE 0240     Save, dan ook CR doen
```

- Wel onderscheid tussen hoofd- en kleine letters

```
0273  EA      NOP          Doe niets
0274  EA      NOP
en
027B  EA      NOP          Doe ook niets
027C  EA      NOP
```

- Wel een regelnummer tijdens listen

```
02A5  205EB9  JSR B95E     Druk regelnummer af
02A8  20E0A8  JSR A8E0     met een spatie
```

- Maak scherm niet schoon voor listen

```
02E5  205802  JSR 0258     Naar Find-routine, zonder CLW
```

Een versie van dit programma voor mensen zonder toolkit zal ik hier niet beschrijven. Er is op een van de volgende OSI-dagen vast wel zo'n versie te krijgen.

Henk Schär, (01727-6536)
Reijerskoop 55,
2771 BE Boskoop.

Minidatabase voor Superboard.

```
0230      C95C    CMP i 5C        Is het een \?
0232      D013    BNE 0247        Nee, dan weg
0234      AD0502  LDA 0205        Haal Save-flag
0237      D011    BNE 024A        Save, dan weg
0239      AD2702  LDA 0227        Haal Edit-flag
023C      C9FF    CMP i FF        Edit?
023E      D00A    BNE 024A        Ja, dan weg
0240      A90A    LDA i 0A        Haal linefeed
0242      2000F8  JSR F800        en druk die af
0245      A90D    LDA i 0D        Haal carriage ret.
0247      4C69FF  JMP FF69        en druk die ook af
024A      A95C    LDA i 5C        Herstel Accu.
024C      4C69FF  JMP FF69        en druk die af
024F      EA      NOP
0250      EA      NOP
0251      EA      NOP
0252      EA      NOP
-------------------------
    FIND-ROUTINE
-------------------------
0253      A903    LDA i 03        Haal CLW
0255      2000F8  JSR F800        en maak scherm schoon
0258      20A6A3  JSR A3A6        Interpreteer bufferinhoud
025B      A579    LDA 79          Begin vooraan in basic text
025D      A67A    LDX 7A
025F      85E0    STA E0          Regel pointers in E0E1
0261      86E1    STX E1
0263      A001    LDY i 01
0265      B1E0    LDA (E0),Y      Test einde van de text bereikt
0267      F072    BEQ 02DB        Ja, dan afzetten
0269      A003    LDY i 03        Zet Y voor de text van de rege
026B      A2FF    LDX i FF        Zet X voor de buffer
026D      E8      INX
026E      C8      INY
026F      B515    LDA 15,X        Wat staat er in de buffer
0271      F010    BEQ 0283        Eind-nul, dan gevonden
0273      29DF    AND i DF        Kleine letter wordt groot
0275      85E5    STA E5          Bewaar letter even
0277      B1E0    LDA (E0),Y      Wat staat er in basic text
0279      F012    BEQ 028D        Eind-nul, dan geen treffer
027B      29DF    AND i DF        Klein wordt weer groot
027D      C5E5    CMP E5          Vergelijk text met buffer
027F      F0EC    BEQ 026D        Gelijk, volgende proberen
0281      D0E8    BNE 026B        Niet gelijk, vooraan in buffer
0283      209802  JSR 0298        GEVONDEN: list deze regel
0286      2000FD  JSR FD00          wacht op toets
0289      C91B    CMP i 1B          is het escape?
028B      F04E    BEQ 02DB          Ja, dan afzetten
028D      A001    LDY i 01          Nee, dan pointers volgende
028F      B1E0    LDA (E0),Y        regel overnemen in A
0291      AA      TAX               en X
0292      88      DEY
0293      B1E0    LDA (E0),Y
0295      4C5F02  JMP 025F          en volgende regel doen
```

```
0298    206CA8  JSR A86C        LIST: doe CR+LF
029B    A002    LDY i 02          neem regelnr. in A en X
029D    B1E0    LDA (E0),Y
029F    AA      TAX
02A0    C8      INY
02A1    B1E0    LDA (E0),Y
02A3    84E5    STY E5            bewaar Y even
02A5    20E0A8  JSR A8E0          druk een spatie af
02A8    206CA8  JSR A86C          en een extra CR+LF
02AB    A4E5    LDY E5          LOOP: pak Y weer op
02AD    C8      INY               stap verder
02AE    B1E0    LDA (E0),Y        wat staat er in de text
02B0    F025    BEQ 02D8          eind-nul, einde regel
02B2    84E5    STY E5            berg Y weer op
02B4    20BA02  JSR 02BA          list dit ene byte
02B7    4CAB02  JMP 02AB          en loop door
02BA    101A    BPL 02D6        LIST1: ascii, print en weg
02BC    AA      TAX               token over naar X
02BD    A0FF    LDY i FF          zet Y voor basic woordenlijst
02BF    CA      DEX             TELAF: verlaag X
02C0    1008    BPL 02CA          tot we onder 128 komen
02C2    C8      INY             ZOEK: stap door woordenlijst
02C3    B984A0  LDA A084,Y
02C6    10FA    BPL 02C2          geen woordeinde, print
02C8    30F5    BMI 02BF          woordeinde, tel X af
02CA    C8      INY             GEVONDEN: stap verder
02CB    B984A0  LDA A084,Y        neem letter uit woordenlijst
02CE    3006    BMI 02D6          einde woord, print en weg
02D0    20D602  JSR 02D6          geen woordeinde, print
02D3    4CCA02  JMP 02CA          ga verder bij gevonden
02D6    297F    AND i 7F        PRINT: clear bit 7
02D8    4CE5A8  JMP A8E5          en druk af
02DB    4CEB9F  JMP 9FEB        AFZET: doe warme start(tk)
-------------------------
    VERDEELROUTINE
-------------------------
02DE    C95C    CMP i 5         Is het een ] ?
02E0    F003    BEQ 02E5        Ja, dan routine ingaan
02E2    4C4890  JMP 9048        Nee, verder naar toolkit (tk)
02E5    205302  JSR 0253        Naar Find-routine
02E8    4CBC00  JMP 00BC        en terug naar Basic
-------------------------
    INITIALISATIE
-------------------------
02EB    A94C    LDA i 4C        Breng omweg naar 02DE
02ED    A2DE    LDX i DE
02EF    A002    LDY i 02
02F1    85C5    STA C5          aan op C5C7
02F3    86C6    STX C6
02F5    84C7    STY C7
02F7    4C2CFF  JMP FF2C        en doe warme start
02FA    EA      NOP
-------------------------
```

BASICODE-2

HIERØNDER STAAT EEN UITLEGPRØGRAMMA DAT DØØR DE NØS IS UITGEZØNDEN, ØVER DE VERSCHILLENDE SUBRØUTINES VØØR DE BAASICØDE-2. BØVENDIEN STAAT TELKENS DE VØØR DE ØSI GESCHIKTE SUBRØUTINE (GESCHREVEN DØØR ØNS ALLER HENK WEVERS) ERACHTER GEPLAKT. BIJ DE NØS IS BØVENDIEN VØØR FL. 25,= EEN BASICØDE-2 BØEK PLUS CASSETTE TE KØØP WAAR ØØK DE ØSI-SUBRØUTINES ØP STAAN EN HET VØØRBEELDPRØGRAMMA VAN HENK WEVERS.

HALLØ, DAAR ZIJN WE DAN MET HET DERDE EN LAATSTE DEEL VAN HET PRØGRAMMA ØVER HET NIEUWE BASICØDE-2 PRØTØCØL. DEZE KEER HEBBEN WE HET ØVER DE VARIABELEN EN ØVER DE STANDAARD SUBRØUTINE'S. (DUS TØT REGEL 1000).
NØGMAALS VØØR DE DUIDELIJKHEID: DEZE RØUTINE'S ZIJN VØØR ELKE CØMPUTER APART GEMAAKT. DE SUBRØUTINE'S WØRDEN U KANT EN KLAAR DØØR HET BASICØDE-2 VERTAALPRØGRAMMA VØØR UW CØMPUTER GELEVERD.

WE BEGINNEN MET HET BESPREKEN VAN DE WERKING VAN DE STANDAARD SUBRØUTINE'S.

```
5 PRINT CHR$(26);:Ø0=PEEK(546):Ø1=PEEK(548)
7 Ø3=PEEK(547):Ø2=PEEK(549):Ø3=Ø3-Ø0:Ø2=Ø2-Ø1
8 PØKE530,1
10 GØTØ 1000
20 GØTØ 1010
```

GØSUB **100**

MET DEZE ØPDRACHT KUNT U VØØRTAAN HET SCHERM LATEN SCHØØNWISSEN EN DE CURSØR LINKS BØVEN IN DE HØEK PLAATSEN.

ØP REGEL 100 ZAL VØØR UW CØMPUTER DUS STAAN:
-HØME ØF
-CLS ØF
-PRINT CHR$(12) ØF
-PRINT CHR$(147) ØF

NØU JA, GA ZØ MAAR DØØR.

ØP DEZELFDE REGEL NA EEN : ØF ØP DE VØLGENDE REGEL STAAT DUS EEN RETURN

```
100 PRINT CHR$(26);:RETURN
```

GØSUB 110

DEZE ØPDRACHT KUNT U GEBRUIKEN ØM DE CURSØR ØP EEN GEWENSTE PLAATS VAN HET SCHERM TE ZETTEN. DAAR HØØRT DUS NØG WAT BIJ:

DE BØVENSTE SCHERMREGEL HEET VØØRTAAN REGEL 0, DE VØLGENDE IS REGEL 1, ENZ.
DE MEEST LINKSE PLAATS ØP ELKE REGEL IS PLAATS 0, EN EVENZØ ØPLØPEND TØT 39 BIJ EEN SCHERM MET 40 KARAKTERS PER REGEL.

DE BEDØELING IS NU, DAT U IN DE VARIABELE HØ AANGEEFT ØP WELKE PLAATS IN DE HØRIZØNTALE RICHTING DE CURSØR MØET KØMEN EN IN DE VARIABELE VE ØP WELKE REGEL IN VERTICALE RICHTING.

EEN VØØRBEELD:

```
VE=2 : HØ=4 : GØSUB 110
```

HIERMEE KØMT DE CURSØR ØP REGEL 2 (DUS DE DERDE REGEL VAN BØVEN) EN ØP DIE REGEL ØP PLAATS 4, DUS ØP HET VIJFDE KARAKTER VAN DIE REGEL.
HIERNA ZAL HET EERSTE KARAKTER VAN EEN VØLGEND PRINT - STATEMENT ØP DIE AANGEGEVEN PLAATS TERECHT MØETEN KØMEN.

```
110 PØKE 14,1
111 Ø4=HØ:IF HØ>Ø3 THEN Ø4=Ø3
112 Ø5=VE:IF VE>Ø2 THEN Ø5=Ø2
113 PØKE 553,Ø4+Ø0:PØKE 554,Ø5+Ø1-1
114 PRINT CHR$(10);
115 RETURN
```

GØSUB 120

DEZE SUBRØUTINE DØET JUIST HET ØMGEKEERDE VAN DE VØRIGE:
NADAT UW PRØGRAMMA DEZE SUBRØUTINE HEEFT AANGERØEPEN STAAT IN VE EN HØ DE JUISTE PØSITIE VAN DE CURSØR ØP DAT MØMENT.

DØØR DAN BIJ VE EN/ØF HØ IETS ØP TE TELLEN ØF AF TE TREKKEN EN DAARNA GØSUB 110 TE GEVEN PLAATST U DE CURSØR TEN ØPZICHTE VAN DE ØUDE PLAATS.

```
120 HØ=PEEK(553):HØ=HØ-Ø0:VE=PEEK(554):VE=VE-Ø1
121 RETURN
```

GØSUB **200**

'DIT IS EEN SUBRØUTINE DIE VØØR U KIJKT ØF ER EEN TØETS VAN HET TØETSENBØRD WAS INGEDRUKT.
ALS ER EENTJE WAS INGEDRUKT DAN WØRDT IN DE VARIABELE IN$ AANGEGEVEN WELKE TØETS, ALS ER GEEN WAS INGEDRUKT DAN ZAL IN$ EEN LEGE STRING WØRDEN.

```
200 04=PEEK(11):05=PEEK(12):P0KE57088,1:P0KE11,0:P0KE12,253
201 06=PEEK(57088):IF06=255THEN204
202 06=USR(1):06=PEEK(531):IN$=CHR$(06):P0KE533,0:G0T0207
203 06=USR(1):06=PEEK(531):IN$=CHR$(06):P0KE533,0:G0T0207
204 P0KE57088,254:06=PEEK(57088)
205 IF06AND32=0THENIN$=CHR$(27):P0KE533,0:G0T0207
206 IN$=""
207 P0KE11,04:P0KE12,05:RETURN
```

GØSUB **210**

DEZE SUBRØUTINE LIJKT WAT ØP DE VØRIGE. NU WØRDT ECHTER DØØR DE SUBRØUTINE GEWACHT TØTDAT ER EEN TØETS IS INGEDRUKT EN PAS DAN WØRDT DE INGEDRUKTE TØETS ALS IN$ WEER AAN UW PRØGRAMMA AFGEGEVEN.

```
210 GØSUB 200:IF IN$="" THEN 210
211 RETURN
```

GØSUB **250**

MET DEZE AANRØEP KAN EEN BASICØDE - PRØGRAMMA VØØRTAAN EEN PIEPJE GEVEN.
BIJ CØMPUTERS WAAR GEEN LUIDSPREKER IS INGEBØUWD DØET DEZE SUBRØUTINE NATUURLIJK NIETS.

MET DEZE SUBRØUTINE KAN GEEN MUZIEK WØRDEN GEMAAKT. DE TØØNHØØGTE EN DUUR VAN HET PIEPJE IS NIET VASTGESTELD.
DE SUBRØUTINE IS ALLEEN BEDØELD, DE MØGELIJKHEID TØT EEN HØØRBAAR SIGNAAL TE BIEDEN.

```
250 PRINT CHR$(7);:RETURN
```

GØSUB **260**

HET AANRØEPEN VAN DEZE SUBRØUTINE LEIDT TØT EEN ANTWØØRD IN DE VARIABELE RV

IN RV WØRDT NAMELIJK EEN RANDØM GETAL AFGELEVERD. HET KLEINSTE GETAL DAT DE RØUTINE KAN AFLEVEREN IS 0, ALLE AFGELEVERDE GETALLEN ZIJN KLEINER DAN 1.

DEZE SUBRØUTINE IS ØPGENØMEN ØMDAT DE RND FUNCTIE IN DIVERSE MICRØ'S NØGAL VERSCHILLEND BLIJKT TE WERKEN.

```
260 RV=RND(1):RETURN
```

GØSUB **270**

MET DEZE AANRØEP WØRDEN AFGEDANKTE STRINGS ØPGERUIMD EN WØRDT IN DE VARIABELE FR ALS ANTWØØRD GEGEVEN HØEVEEL BYTES ER NØG ØNGEBRUIKT ZIJN.
TEKSTPRØGRAMMA'S KUNNEN HIERMEE TIJDIG ZIEN AANKØMEN ØF HET GEHEUGEN VØL DREIGT TE RAKEN EN EVENTUEEL EEN PASSENDE MELDING PRØDUCEREN.

```
270 FR=FRE(2):RETURN
```

```
300 SR$=STR$(SR)
301 Ø7=LEN(SR$):IFØ7=0 THEN RETURN
302 IF RIGHT$(SR$,1)<>" " THEN 304
303 SR$=LEFT$(SR$,Ø7-1):GØTØ 301
304 IF LEFT$(SR$,1)<>" " THEN RETURN
305 SR$=RIGHT$(SR$,Ø7-1):GØTØ 301
```

GØSUB **300**

ALVØRENS DEZE SUBRØUTINE AAN TE RØEPEN MØET U DE VARIABELE MET DE NAAM SR EEN (GETALS-)WAARDE HEBBEN GEGEVEN.
ALS ANTWØØRD GEEFT DEZE SUBRØUTINE U DE STR$(SR).

DIT LIJKT WAT ØVERBØDIG, MAAR ER IS NØGAL WAT VARIATIE IN WAT DE STR$ FUNCTIE ØP DIVERSE MACHINES DØET: DE EEN GEEFT WEL EEN SPATIE NA HET LAATSTE CIJFER, DE ANDER NIET.
SØMMIGEN GEVEN BIJ PØSITIEVE GETALLEN EEN SPATIE ØP DE PLAATS VAN HET TEKEN, ANDERE BEGINNEN MET HET EERSTE CIJFER.

DEZE NIEUWE BASICØDE SUBRØUTINE GEEFT GEEN SPATIES, NIET ERVØØR EN NIET ERNA.

GØSUB **310**

DIT IS WEL EEN HELE MØØIE, ALTHANS VØØR EEN AANTAL CØMPUTERS DIE DIT NIET VAN HUIS UIT KENNEN:
HET GETAL DAT U IN SR MEEGEEFT WØRDT ØMGEZET IN EEN SR$ DØØR DE SUBRØUTINE.
DE LENGTE VAN SR$ IS NA AFLØØP GELIJK AAN WAT U TEVØREN IN CT HAD ØPGEGEVEN EN ALS U IN CN EEN GETAL GRØTER DAN 0 HAD VERMELD DAN BEVAT SR$ EEN DECIMALE PUNT MET DAARNA NØG CN CIJFERS.
EEN VØØRBEELDJE:

```
SR=123.4567 : CT=7 : CN=2 : GØSUB 310
```

DIT LEVERT ALS ANTWØØRD SR$=' 123.46'

```
SR=-1E-3 : CT=7 : CN=3 : GØSUB 310
```

LEVERT ALS RESULTAAT SR$=' -0.001'

```
SR=98765 : CT=5 : CN=1 : GØSUB 310
```

GEEFT ALS REAKTIE SR$='*****'

```
SR=98765 : CT=5 : CN=0 : GØSUB 310
```

GEEFT ALS REAKTIE SR$='98765'

U ZIET: CT IS STEEDS HET TØTALE AANTAL CIJFERS IN HET RESULTAAT EN NA DE PUNT STAAN NØG CN CIJFERS. ALS HET NIET MØGELIJK IS KØMEN ER CT STERRETJES EN HET ANTWØØRD IS STEEDS IN CIJFERNØTATIE EN NETJES AFGERØND.

DIT IS NATUURLIJK BEDØELD ØM MØØIE TABELLEN TE KUNNEN PRØDUCEREN.

```
310 Ø4=SR:IF CN<>0 THEN 316
312 SR=INT(SR+.5):GØSUB300:GØTØ330
316 Ø5=SGN(SR):SR=ABS(SR):Ø8=INT(SR):Ø9=SR-Ø8
318 FØR Ø6=1 TØ CN:Ø9=Ø9*10:NEXT Ø6
320 Ø9=INT(Ø9+.5):SR=Ø9:GØSUB 300
322 Ø9$=RIGHT$("0000000000000000000"+SR$,CN)
324 IF Ø8=0 AND Ø9=0 THEN Ø5=1
326 SR=Ø8:GØSUB300:IF Ø5=-1 THEN SR$="-"+SR$
328 SR$=SR$+"."+Ø9$
330 IF LEN(SR$)<=CT THEN 334
332 SR$=LEFT$("********************",CT):GØTØ 340
334 SR$=RIGHT$("                    "+SR$,CT)
340 SR=Ø4:RETURN
350 PØKE 517,1:PRINT SR$;:PØKE517,0:RETURN
```

GØSUB **350**

ZØRGT ERVØØR DAT SR$ WØRDT GEPRINT ØP DE PRINTER.
SR$ KAN ZIJN BEPAALD DØØR DE SUBRØUTINE ØP REGEL 300 ØF 310 MAAR NATUURLIJK ØØK DØØR UW EIGEN PRØGRAMMA.
NA HET PRINTEN VAN SR$ WØRDT NIET AUTØMATISCH EEN LINEFEED NAAR DE PRINTER GESTUURD.

DAARTØE DIENT

GØSUB 360

DIE DUS NIETS ANDERS DØET DAN DE 'CARRIAGE RETURN' EN DE 'LINEFEED' NAAR DE PRINTER GEVEN.

```
360 PØKE 517,1:PRINT      :PØKE517,0:RETURN
READY
```

DAT WAREN DE SUBRØUTINES.
DE VRIJE REGELNUMMERS ZIJN VØØR EVENTUELE LATERE UITBREIDINGEN.

VARIABELEN

ØM PRØBLEMEN BIJ UITWISSELEN VAN PRØGRAMMA'S TE VØØRKØMEN SPREKEN WE HET VØLGENDE AF:

A. NUMERIEKE VARIABELEN ZIJN SINGLE PRECISIØN. REKEN IN DE PRAKTIJK NIET ØP MEER DAN 6 CIJFERS.

B. NAMEN VAN VARIABELEN ZIJN MAXIMAAL TWEE KARAKTERS LANG, WAARVAN HET EERSTE EEN LETTER MØET ZIJN. ALS ER NØG EEN TWEEDE KARAKTER VØLGT, MAG DAT EEN LETTER ØF EEN CIJFER ZIJN. VØØR STRINGVARIABELEN WØRDT DE NAAM GEVØLGD DØØR $. KLEINE LETTERS ZIJN IN NIET TØEGESTAAN.

C. STRINGVARIABELEN MØGEN MAXIMAAL 255 KARAKTERS LANG ZIJN.

D. NAMEN VAN DE VARIABELEN MØGEN NIET MET DE LETTER Ø BEGINNEN.

E. UITGESLØTEN ZIJN TEVENS DE VARIABELEN:
AS AT FN GR IF PI ST TI TI$ TØ.

F. VØØR CØMMUNICATIE MET DE STANDAARD SUBRØUTINE'S WØRDEN GEBRUIKT DE VARIABELEN:
HØ VE FR SR CN CT RV IN$ SR$.

ZØ, DAT WAS HET DAN VØØR VAN DE WEEK. MISSCHIEN IS HET U ALLEMAAL NØG NIET EVEN DUIDELIJK, MAAR DAT KØMT IN DE TØEKØMST VANZELF ALS U EEN AANTAL PRØGRAMMA'S IN BASICØDE-2 HEEFT GEZIEN.
HET ZAL U DAN ØPVALLEN HØEVEEL MØGELIJKHEDEN BASICØDE-2 EXTRA BIEDT!

STUURT U ØØK EENS EEN PRØGRAMMA IN BASICØDE-2 NAAR HØBBYSCØØP?!

HØBBYSCØØP
PØSTBUS 1200
HILVERSUM

Titel, LOAD-off, SAVE-off en auto-RUN op band

Als je een programma op de standaard manier op een cassette zet (SAVE'd) dan is het makkelijk als er eerst even een programma-titel op de band komt. Maar dan mag dat bij het laden geen "syntax error" geven ! Het SAVE'n zou verder automatisch moeten ophouden nadat eerst een LOAD-off (POKE 515,0) op de band is gezet.
Helemaal makkelijk wordt het als een programma dat je van de band haalt (LOAD) zichzelf opstart (RUN). Zeker als bijv. daarna via een loop met een INPUT er in eerst nog een heleboel gegevens binnengehaald moeten worden die iets verder op dezelfde band staan (bijv. bij een aangepaste versie van "Trefwoorden" pag. EC 1).
In het laatste geval kan je na LOAD:LOAD en return dus rustig achter de computer weglopen. Als je een keer terugkomt is het programma geladen, zijn de nodige variabele gegevens binnengehaald en staat de eerste vraag op het beeldscherm te wachten !
Het onderstaande programma'tje dat deze vier zaken regelt, moet eerst onder het te SAVE'n programma worden gezet.
De OSI-BASIC kent wel geen CHAIN- of MERGE-commando, maar LOAD doet in feite hetzelfde (pas op: #L van de Toolkit niet !).
Via RUN 63000 kom je daarna waar je wezen wil, als tenminste in een van de eerste regels van het te SAVE'n programma POKE 515,0 is opgenomen.

Het is praktisch dit programma aan de BASICODE-2 subroutines van H. Wevers toe te voegen. Dan start het zondagavond's binnengekomen programma na het toevoegen van de subroutines zichzelf en een RUN 63000 zet het daarna netjes op de band.
Het recept daarvoor is: eerst dit programma intikken en (tijdelijk) op de band zetten, dan renumberen (bijv. #R2000,10) en LIST-62999 vervangen door LIST-999:LIST63000-. Vervolgens de BASICODE-2 subroutines en het onderstaande programma laden, de computer op 300 BAUD zetten en RUN 2000 geven.

Ton Helwig 5/83

```
63000 POKE515,0
63010 PRINT "AL POKE515,0 OP EERSTE REGEL GEZET ?"
63020 INPUT"PROGRAMMANAAM ";A$:A$="REM "+A$
63030 NULL8:SAVE:PRINTA$
63040 POKE4,194:POKE5,165:LIST-62999
63050 POKE4,174:POKE5,251
63060 PRINT:PRINT"POKE560,32:POKE561,119"
63070 PRINT"POKE562,164:POKE563,76:POKE564,194"
63080 PRINT"POKE565,165:POKE11,48:POKE12,2"
63090 PRINT"X=USR(X)"
63100 POKE517,0
```

```
10 REM
20 REM INDIRECT FILE UTILITY UNDER OS-65D V3.0
25 REM **** BY H.A.BARTEN 6/1/82 ****
30 REM
35 FORI=1TO30:PRINT:NEXTI
40 PRINT"INDIRECT FILE UTILITY":PRINT:PRINT
50 PRINT"WITH THIS UTILITY YOU CAN SET UP":PRINT
60 PRINT"A WORKSPACE FOR INDIRECT FILES IN BASIC":PRINT
70 PRINT:PRINT"THE TERMINAL WIDTH IS AUTOMATIC SET TO 132":PRINT
80 MP=PEEK(8960):MS=12926:WD=132
85 IFPEEK(8999)=58THENMS=14974
90 NB=(MP+1)*256-MS
100 PRINT"YOU HAVE ";NB;" BYTES FREE":PRINT
105 PRINT"IN THE BASIC WORKSPACE":PRINT
110 PRINT"HOW MANY PAGES DO YOU WANT FOR THE":PRINT
120 INPUT"INDIRECT FILE PART OF THE WORKSPACE ";A:PRINT
130 IFA<10RA>INT(NB/256)THEN110
140 U=(MP-A+1)*256:HB=INT(U/256)
150 PRINT"YOU HAVE LEFT ";U-MS;" BYTES FREE":PRINT
155 PRINT"IN THE NORMAL BASIC WORKSPACE":PRINT
160 INPUT"IS THIS ALRIGHT ";A$
170 IFMID$(A$,1,1)<>"Y"THENRUN
180 REM ---- TERM WIDTH ----------
190 POKE23,WD:POKE24,INT(WD/14)*14
200 REM ---- END OF WORKSPACE ----------
205 POKE132,U-HB*256:POKE133,HB
210 REM ---- IND FILE OUT- AND IN-ADDRESSES ------
220 POKE9554,HB:POKE9368,HB
230 FORI=1TO10:PRINT:NEXT
235 PRINT"USE OF THE INDIRECT FILE WORKSPACE:":PRINT
237 PRINT:PRINT
240 PRINT:PRINT"   -LOAD SOURCE FILE 1 WITH LOAD COMMAND"
250 PRINT"   -TYPE LIST";CHR$(91);" (";CHR$(91);"= SHIFT K)<CR>"
260 PRINT"   -AFTER COMPLETION OF THE LISTING"
270 PRINT"      TYPE A ";CHR$(93);" (";CHR$(93);" = SHIFT M";
275 PRINT" ECHOED AS A ";CHR$(93);CHR$(93)
280 PRINT"   -LOAD SOURCE FILE 2 WITH LOAD COMMAND",
290 PRINT"   -TYPE A CONTROL-X TO MERGE THE TWO FILES"
295 FORI=1TO10:PRINT:NEXTI
300 END
```

OS-65D ALLERLEI
==========

1. WEGSCHRIJVEN NAAR CASSETTE-RECORDER.

DE GEBRUIKELIJKE NULLEN NA EEN CR ONTBREKEN.
WERKEN MET POKE 21,X VOOR 300 BAUD X = 10
VOOR 600 BAUD X = 18
VERDER 2 MOGELIJKHEDEN: LIST#1 OF DISK!"IO 03".
BIJ LIST#1 KOMT ER GEEN 'OK' OP DE BAND AAN HET SLOT.
NADEEL: BIJ WEER INLEZEN BLIJFT DE COMPUTER IN DE INLEES-
TOESTAND EN KOMT DAAR NIET MEER UIT !
DUS BETER DISK!"IO 03" GEBRUIKEN, GEVOLGD DOOR LIST.
INLEZEN MET DISK!"IO 01".

2. GRAFISCHE KARAKTERS

MET 'POKE 9819,N : PRINT CHR$(16)' KAN ELK TEKEN WORDEN
AFGEDRUKT, WAARBIJ N = ASCII-WAARDE VAN HET TEKEN.

3. X = USR(Y)

ALS I.P.V. GEHEUGENLOCATIES 574 ($033E) EN 575 ($033F)
GEBRUIK WORDT GEMAAKT VAN 8955 ($32FB) EN 8996 ($32FC)
OM HET ADRES VAN DE USR-SUBROUTINE IN OP TE BERGEN, KAN MEN OOK
DE ZERO-PAGE ADRESSEN BENUTTEN. ALLE GEGEVENS UIT ZERO-PAGE
WORDEN DAN NAMELIJK NAAR DE PAGE 0/1 SWAP-BUFFER OVERGEBRACHT
EN BIJ TERUGSPRINGEN WEER GEPLAATST !

4. GET KEY

DEZE ROUTINE START OP $2528.
DE ROUTINE LAAT DE ASCII-WAARDE VAN HET INGETYPTE TEKEN
ACHTER IN $2363.
DUS: POKE 574,40 POKE 575,37 X = USR(1) X = PEEK(9059)

5. 'LIST' IN EEN PROGRAMMA

POKE 4,180 : POKE 5,7 : LIST : POKE 4,204 : POKE 5,10
DE 'OK'-MELDING BLIJFT NU ACHTERWEGE, OMDAT SUBROUTINE $0ACC
TIJDELIJK VERVANGEN WORDT DOOR SUBROUTINE $07B4.

H.A. BARTEN
JAN VAN GOYENSTRAAT 39
3351 JM PAPENDRECHT

```
.A

 10 0000        ;*****************************
 20 0000        ;* HALTTOETS VOOR ASSEMBLER *
 30 0000        ;* VOOR A A1 A2 P KOMMANDOS *
 40 0000        ;* VAN CASSETTE VERSIE      *
 50 0000        ;*                          *
 60 0000        ;*    MARTIEN SCHOT         *
 70 0000        ;*     HOFSTEDE 21          *
 80 0000        ;* 3902 CG VEENENDAAL       *
 90 0000        ;*  TEL 08395 13219 `       *
100 0000        ;*****************************
110 0000        ;
120 0000        ;DE JMP WELKE NAAR FFEE SPRINGT UIT DE ASSEMBLER
130 0000        ;MOET WORDEN OMGELEID VIA DIT PROGRAMMATJE
140 0000        ;
150 0000        ;OP 1333 STAAT NU 4C EE FF DIT WORDT NU 4C 91 13
160 0000        ;
170 0000        ;DE POINTER WELKE HET BEGIN VAN DE SOURCE FILE
180 0000        ;AANWIJST MOET WORDEN VERHOOGD ,WAS 1391 WORDT
190 0000        ;13A2 DUS INHOUD VAN 12C9 WAS 91 WORDT A2
200 0000        ;
210 0000                KEY=$DF00
220 1391                *=$1391 ;BEGIN VAN HULPPROGRAMMATJE
230 1391 48             PHA     ;ACCU NAAR STACK
240 1392 A9DF    LUSJE  LDA #%11011111 ;ROW5
250 1394 8D00DF         STA KEY
260 1397 A910           LDA #%00010000 ;COL 4
270 1399 2C00DF         BIT KEY
280 139C F0F4           BEQ LUSJE ;LINE FEED INGEDRUKT DAN
290 139E                          ;NAAR LUSJE TERUG EN OPNIEUW
300 139E 68             PLA       ;NIET INGEDRUKT DAN ACCU VAN
310 139F                          ;DE STACK
320 139F 4CEEFF         JMP $FFEE ;EN GA NAAR OUTPUTR
```

S T R I N G S E A R C H

'DISK FIND' VOOR OS-65D

DOELSTELLING: ZOEK ALLE PLAATSEN IN DATA-FILE WAAR OPGEGEVEN STRING VOORKOMT

OPZET MACHINE CODE ROUTINE, DIE DEEL VAN RAM AFZOEKT OP VOORKOMEN VAN OPGEGEVEN STRING EN DE STRINGADRESSEN NOTEERT IN TABEL.
RAMDEEL, DAT ONDERZOCHT WORDT, IS GEHEEL AANTAL PAGINA'S

GEBRUIK MET DOS: ZOEKEN IN FILE-BUFFER
MC-ROUTINE WORDT GEPLAATST TUSSEN GERESERVEERDE FILE-RUIMTE EN BASIC-WERKRUIMTE
(MET TAPE-FORTH: ZOEKEN IN SCREENS IN HOGE RAM)

TOEPASSING : PROGRAMMA'S, WAARIN ZOEKEN NAAR ASCII-STRINGS VEELVULDIG OPTREEDT.

WERKING
1. FILE WORDT (IS) IN RAM GELADEN
2. STRING WORDT IN BUFFER GELADEN
   STRING-LENGTE IN RAM GEPOKED
3. STRINGBUFFER WORDT BYTE VOOR BYTE VERGELEKEN MET TE ONDERZOEKEN RAM
4. ALS VOLLEDIG IDENTIEKE STRING IN RAM WORDT GEVONDEN, WORDT HET ADRES VAN DE RAMBYTE IN TABEL GESCHREVEN EN DE STRINGTELLER OPGEHOOGD.
5. ALS OPGEGEVEN RAMDEEL VOLLEDIG OP HET VOORKOMEN VAN DE STRING IS ONDERZOCHT, WORDT TERUGGESPRONGEN NAAR HOOFDPROGRAMMA
6. ALS ADRESTABEL VOL IS, WORDT TERUGGESPRONGEN NAAR HET HOOFDPROGRAMMA
   (NOOT: ER IS (NOG) NIET VOORZIEN IN VERDER ZOEKEN IN DEZELFDE BUFFER, MAAR DAT IS IN HET HOOFDPROGRAMMA OP TE VANGEN DOOR HET LAATSTE STRINGADRES + STRINGLENGTE TE GEBRUIKEN ALS NIEUW RAM-START-ADRES)

OPZET :
Y = TELLER VOOR DE OVEREENKOMSTIGE STRING-KARAKTERS
X = INDEX VOOR ADRES-TABEL
LENGTH = LENGTE VAN TE ZOEKEN STRING (<13 BYTES)
COUNT = TELLER VOOR DE GEVONDEN OVEREENKOMSTIGE STRINGS
HBLB = POINTER VAN TE ONDERZOEKEN RAM
EERSTE WAARDE IS STARTADRES
LIMIT1 = HOOGSTE WAARDE VOOR COUNT
LIMIT2 = HOOGSTE WAARDE VOOR HB
LIMIT3 = LB VAN HOOGSTE WAARDE VOOR HBLB
TABLE = ADRES VAN EERSTE BYTE IN ADRESTABEL

'STRING SEARCH'
==========

VANUIT HET HOOFDPROGRAMMA:
STRING WORDT GEPOKED VANAF 'STRINGSTART'
LENGTH WORDT GEPOKED

```
            TABEL       = START + #$70
            STRINGSTART = START - #$10
            COUNT       = START - #$02
            LENGTH      = START - #$01
START:                          EVENTUEEL OPBERGEN VAN
                                REGISTERS A, X EN Y
            LDX #$00            TABLE-INDEX = 0
            LDA #$00
            STA COUNT           COUNT = 0
            LDA LB              (LB KAN HIER WORDEN GEPOKED)
            STA BRAVO + 1
            LDA HB              (HB KAN HIER WORDEN GEPOKED)
            STA BRAVO + 2
ALPHA:      LDY #$0             KARAKTER-TELLER = 0
BRAVO:      LDA HBLB,Y          LAAD KARAKTER
            CMP STRINGSTART,Y   VERGELIJK MET BYTE IN RAM
            BNE CHARLY          NIET HETZELFDE? SPRING
            INY                 VERHOOG KARAKTER-TELLER
            TYA
            CMP LENGTH          VERGELIJK Y MET LENGTH
            BNE BRAVO           NIET GELIJK? NOG EENS
            LDA BRAVO + 1       SCHRIJF LB VAN ADRES
            STA TABLE,X         IN TABEL
            INX
            LDA BRAVO + 2       SCHRIJF HB VAN ADRES
            STA TABLE,X         IN TABEL
            INX
            INC COUNT           VERHOOG COUNT
            LDA COUNT           IS ADRES-TABEL AL VOL?
            CMP LIMIT1          DAN TERUG NAAR HOOFDPROGRAMMA
            BNE CHARLY          ANDERS VERDER
                                EVENTUEEL OPHALEN VAN
                                REGISTERS A, X EN Y.
            RTS
CHARLY:     INC BRAVO + 1       VERHOOG LB
            BNE DELTA           ALS LB <> 0 DAN SPRING
            INC BRAVO + 2       VERHOOG HB
DELTA       LDA BRAVO + 1       IS LB NIET GELIJK AAN
            CMP LIMIT3          GRENSWAARDE DAN SPRING TERUG
            BNE ALPHA           VOOR NIEUWE ZOEKACTIE
            LDA BRAVO + 2       IS HB NIET GELIJK AAN
            CMP LIMIT2          GRENSWAARDE DAN SPRING TERUG
            BNE ALPHA           VOOR NIEUWE ZOEKACTIE
                                EVENTUEEL OPHALEN VAN
                                REGISTERS A, X EN Y
            RTS
```

HET HOOFDPROGRAMMA KIJKT NAAR COUNT EN VERWERKT VERVOLGENS
DE INFORMATIE, DIE VERVAT IS IN DE ADRESTABEL:
  BEREKENT HET NUMMER VAN DE RECORD, WAARIN DE STRING OPTREEDT
  EN GEEFT DE RECORD WEER OP HET SCHERM OF VIA DE PRINTER.

' S T R I N G S E A R C H '

'DISK FIND' VOOR OS-65D

GEBRUIK VAN 'STRING SEARCH'

DE ROUTINE WORDT GEPLAATST TUSSEN FILE BUFFER #6 EN HET
BASIC-PROGRAMMA, DAT DE ROUTINE GEBRUIKT.
RESERVEER M.B.V. 'CHANGE' OF OP ANDERE WIJZE TENMINSTE
170 BYTES VOOR DE ROUTINE.
DE ROUTINE BEGINT OP $4290 EN WORDT (EENMAAL) GEPOKED MET
DE REGELS 500-620.
DE FILES WORDEN PER TRACK VAN DISK GEHAALD.
HET PROGRAMMA MOET DUS WETEN WELKE FILE OPGEHAALD MOET WORDEN.
REGELS 200-250 STELLEN DE TRACK-NUMMERS VAST.
MET 3300-3490 WORDT DE FILE OP HET OPTREDEN VAN 'STRING'
ONDERZOCHT.
IN HET PROGRAMMA, WAAR DEZE REGELS IN VOORKOMEN ('ADRES'),
HEBBEN DE RECORDS EEN LENGTE VAN 32 BYTES EN VORMEN 4 RECORDS
EEN FILE-ITEM (ADRES).
HET NUMMER VAN DE RECORD MET 'STRING' WORDT BEREKEND
MET 3700-3890.
6000-6160 TENSLOTTE LEEST HET FILE-ITEM VAN DISK.

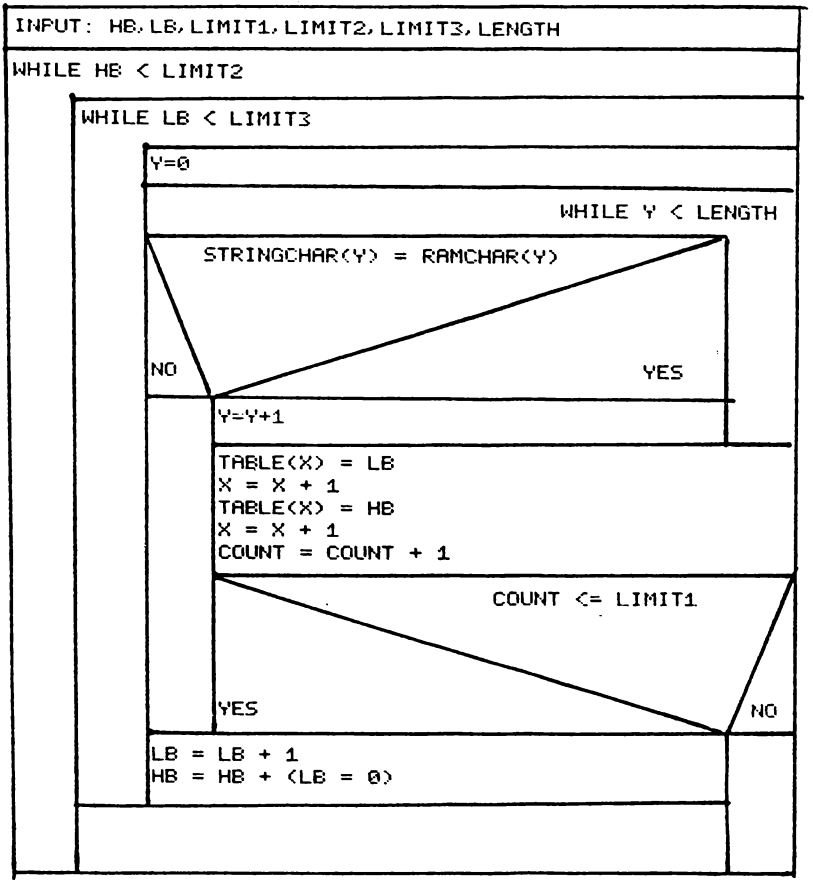
```
200 DISK OPEN,6,NF$
205 REM BEPAAL EERSTE EN LAATSTE TRACKNUMMER VAN DATAFILE NF$
210 X=PEEK(9002):Y=PEEK(9003)
220 X1=INT(X/16):X2=X-X1*16:X1=X1*10+X2:REM EERSTE TRACKNUMMER
230 Y1=INT(Y/16):Y2=Y-Y1*16:Y1=Y1*10+Y2:REM LAATSTE TRACKNUMMER
240 DISK CLOSE,6
250 RETURN
500 REM FORPP=17040TO17121:READQQ:POKEPP,QQ:NEXTPP:RETURN
505 REM MACHINE CODE ROUTINE 'STRING SEARCH'
540 REM DATA162,0,169,0,141,142,66,169,126,141
550 REM DATA164,66,169,58,141,165,66,160,0,185
560 REM DATA126,58,217,128,66,208,32,200,152,205
570 REM DATA143,66,208,241,173,164,66,157,0,67
580 REM DATA232,173,165,66,157,0,67,232,238,142
590 REM DATA66,173,142,66,201,40,208,1,96,238
600 REM DATA164,66,208,3,238,165,66,173,164,66
610 REM DATA201,126,208,199,173,165,66,201,66
620 REM DATA208,192,96
```

```
300 REM ZOEK ADRESSEN WAARIN OPGEGEVEN STRING STAAT
305 LM=0:PQ=17024:REM PQ=$4280 HIER WORDT 'STRING' GEPOKED
310 INPUT"GEEF STRING: ";A$:PRINT
320 L=LEN(A$):IFL>12THENL=12
340 FORI=1TOL: POKEPQ+I-1,ASC(MID$(A$,I,1)):NEXTI
360 POKEPQ+14,0: POKEPQ+15,L:REM COUNT EN LENGTH
370 POKE574,144:POKE575,66:REM ADRES VAN 'STRING SEARCH'
390 T=X1:KK=Y1-X1
400 FORK=0TOKK:T1=T+K:T$=RIGHT$(STR$(T1),2)
440 DISK!"CA 3A7E="+T$+",1":REM LAAD TRACK IN BUFFER #6
450 X=USR(X):LE=PEEK(PQ+14):REM HAAL COUNT OP
460 IFLE<>0THENGOSUB3700
465 IFLL=1ANDLE<>0THENLL=0:K=4:NEXTK:RETURN:REM LL UIT 1500
470 IFLL=1THEN3480
475 PRINTA$;" NIET OP TRACK ";T$:PRINT
480 NEXTK
490 GOSUB15000
500 RETURN



3700 IFLL=0THEN3800
3705 REM BEREKEN RECORD-NR
3710 RP=INT((PEEK(17152)+256*PEEK(17153)-14974)/128)*4
3720 R=RP+K*64:RETURN
3795 REM HAAL ADRESSEN OP
3800 GOSUB10000:PRINT"OP TRACK ";T$;" STAAT";LE;"-MAAL ";A$:PRINT
3805 IFLE=0THENRETURN
3810 FORY=1TOLE
3820 LO=PEEK(17152+2*(Y-1)):HI=PEEK(17152+2*(Y-1)+1)
3830 RP=INT((LO+256*HI-14974)/128)*4
3840 R=RP+K*64:REM R = RECORDNUMMER
3850 PRINT"RECORD ";R/4:PRINT
3870 GOSUB6000:NEXTY
3890 LM=1:RETURN


6000 REM LEZEN VAN ADRES
6020 DISK OPEN,6,NF$
6030 POKE12042,64:POKE12076,5
6060 FORI=0TO3:RR=R+I
6070 DISK GET,RR
6080 INPUT#6,N$(I)
6090 NEXTI
6110 DISK CLOSE,6
6130 FORJ=0TO3: PRINTSPC(10);N$(J):PRINT:NEXTJ:PRINT
6160 RETURN


15000 REM ONDERBREKER
15010 RETURN

JOHN M.A. HERMANS
SCHIPLUIDEN
APRIL 1983
```

'STRING SEARCH'

'DISK FIND' FOR OS-65D

PROGRAM STRUCTURE DIAGRAM

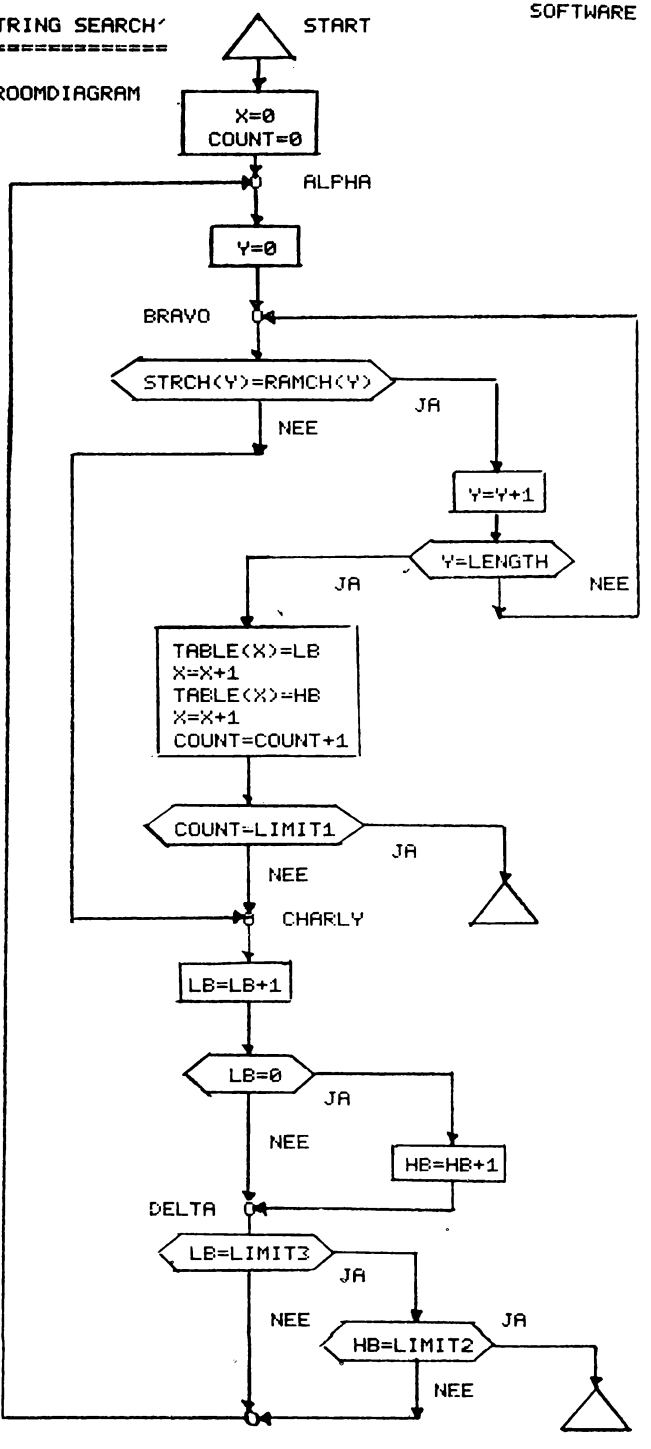


JOHN M. A. HERMANS
SCHIPLUIDEN
APRIL 1983

HB = HIGH BYTE OF ADDRESS IN RAM
LB = LOW BYTE OF ADDRESS IN RAM
LIMIT1 = MAXIMAL NUMBER OF RAM STRING ADDRESSES
LIMIT2 = HIGHEST VALUE OF HB
LIMIT3 = VALUE OF LB BELONGING TO LIMIT2
LENGTH = LENGTH OF STRING TO SEARCH FOR (<= 12)
COUNT = NUMBER OF MATCHING STRINGS

'STRING SEARCH'
==========

STROOMDIAGRAM



F TALEN

FA Assembler

FB Basic

FC Pascal

FD Forth

FE Lisp

FORTH

FORTH is een nieuwe, interaktive computertaal. Een heel andere taal dan BASIC.
Er is literatuur over Forth, maar daar is momenteel in Nederland moeilijk aan te komen. Misschien komt er op korte termijn meer, nu ook hobbyisten deze taal ontdekt hebben.

Een aantal goede artikelen over Forth zijn:
1- Arie Kattenberg: FORTH, HCCN 11(JUli 79) pag 12-14.
2- Fred Rabouw: Introduktie tot FORTH HCCN 14(Jan 80) pag 12-17
3- Gregg Williams: Threads of a FORTH tapestry, BYTE Aug 80, pag 6,128,130,132.
4- Charles H Moore: The Evolution of FORTH, an unusual language, BYTE Aug 80, pag 76-92
5- John S James: What is FORTH? A tutorial introduction BYTE Aug 80, pag 100-126.(literatuur lijst)
6- A.R. en J.Miller: Breakforth into FORTH, BYTE Aug 80 pag 150-163.
7- Kim Harris: FORTH Extensibility or How to write a compiler in 25 words or less. BYTE Aug 80 pag 164-184.

Dit verhaal is bedoeld als smaakmaker.
De schrijver is een amateur op het gebied van FORTH, maar wil toch een poging doen U er iets over te vertellen.

Forth werd omstreeks 1970 uitgedacht door Charles H Moore Zijn bedoeling was: een hogere programmeerproduktie te halen dan met de bestaande talen mogelijk was. Hij vond dat het maken en testen van een programma te veel tijd kostte. Hij beweert dat hij in FORTH een programma schrijft en test in een tiende van de tijd die hij nodig heeft in een andere taal.
Forth is een vijfletterafkorting van het Engelse "fourth" (vierde). Moore bedoelt ermee, dat het een vierde generatie computertaal is. In de tijd waarin deze naam gegeven werd, was de derde generatie computers er net.

FORTH is volgens zijn uitvinder een elitaire programmeer taal. Hij bedoelt daarmee, dat voor goed programmeren in FORTH geldt, hetgeen Godfried Bomans in zijn boek "Erik, of het klein insectenboek" de wespen laat zeggen: "Je kunt het of je kunt het niet".
Dat mag natuurlijk geen reden zijn nu al af te haken. Beter is het vanaf het begin de raad van Charles Moore op te volgen:"Maak geen lange definities". Je werkt dan trouwens vanzelf overzichtelijker en het is gemakkelijker een gestruktureerd programma te maken.

De ontwikkeling van FORTH is sterk beinvloed door de struktuur van LISP (zie artikelen van Frits vd Wateren over Lisp in vroegere HCCN).

In FORTH is de spatie het afscheidingsteken tussen twee woorden, twee getallen of een woord en een getal. Dit is een zeer kenmerkend verschil tussen FORTH en BASIC.

FORTH is opgebouwd als een woordenboek van gedefinieerde 'woorden'. Elk woord is gedefinieerd in eerder gedfinieerde, eenvoudigere'woorden'. De basis van FORTH is gedefinieerd in machinetaal subroutines.

Bij het verwerken van een ingetikte regel in FORTH wordt gebruik gemaakt van de interpreter die in elk FORTH systeem aanwezig is (evenals een compiler).
De interpreter onderzoekt of de woorden die in de tekst van de regel voorkomen, ook in de dictionnaire voorkomen, zodat ze ook uitgevoerd kunnen worden.
Na het intikken van de definitie van een neiuw woord, wordt na het indrukken van 'return' de definitie meteen geinterpreteerd en gecompileerd.

COMPILE TIME
Het nieuw gedfinieerde woord wordt in de dictionnaire bijgeschreven gevolgd door de adressen van de woorden alsmede de getallen of strings, die bij de definitie gebruikt zijn (een getal of string wordt ook woord genoemd als het in een definitie voorkomt of op stack staat).
Komt in de definitie iets voor dat (nog) niet bekend is, dan wordt onmiddelijk de compilatie gestaakt en een foutmelding gegeven. Het woord wordt niet aan de dictionnaire toegevoegd.
Er zijn voorzieningen in FORTH om dat te omzeilen.
Een programma wordt gemaakt, door woorden te definieren met eerder gedefinieerde woorden en met 'basiswoorden'.
De programmering wordt afgesloten met definiering van het laatste woord.
Door dit laatste woord in te tikken, gevolgd door 'return', wordt het programma gestart (gerund). Het programma is dan al helemaal gecompileerd (in tegenstelling tot een programma in BASIC) en daardoor veel sneller, ergens tussen vijfmaal en twintigmaal zo snel.

FORTH maakt het eenvoudig, snel te werken met virtueel geheugen (voorla disk, maar ook tape).
FORTH is zeer geschikt voor een microcomputer met een of meer diskdrives. Echter met een cassetterecorder moet het ook kunnen. In elk geval is het een uitdaging, dat eerst eens te proberen alvorens een disk drive aan te schaffen (inclusief de daarvoor benodigde software).
FORTH staat meestal op disk of tape en wordt in de computer geladen (QuickSave is daarvoor uitstekend geschikt! zie OSI 50 pag boek No 1). FORTH kan echter ook in ROM gezet worden. Het werkt dan verder zoals BASIC, de programmas worden ingetikt of geladen vanuit disk of tape en werken al of niet met disk of tape. Als zonder EDITOR gewerkt wordt lijkt het ons het gemakkelijkst een FORTH programma op tape of disk te zetten inclusief het hele FORTH systeemprogram.

KENMERKEN VAN FORTH

FORTH heeft een aantal zeer typerende eigenschappen, die het duidelijk onderscheiden van andere computertalen, in het bijzonder van BASIC:
- FORTH is zeer compact. Met 8K byte Forth doe je veel meer dan met 8K byte BASIC.
- FORTH is snel dankzij zijn typische, piramidale, zeer gestruktureerde opbouw in de vorm van instrukties die woorden worden genoemd.
- FORTH is uitbreidbaar. Als je in FORTH programmeert, ben je bezig FORTH uit te breiden. Je zou dit kunnen noemen aktief programmeren, tegenover passief programmeren met b.v. BASIC. Je kunt namelijk nieuwe elementen aan FORTH toevoegen zoals: nieuwe typen data of nieuwe operatoren. Je kunt b.v. de BASIC functies voor matrixbewerking, die in BASIC gedefinieerd zijn, maar niet in de BASIC van het Superboard voorkomen, zelf in FORTH definieren. Deze nieuwe functies worden dan met dezelfde voorrang en snelheid uitgevoerd als de andere FORTH woorden.
- FORTH is zeer transportabel. Een programma in FORTH kun je laten lopen op elke machine met gelijke microprocessor, monitorprogramma en dezelfde FORTH start-dictionnaire (geordende verzameling van FORTH woorden).

Forth code is moeilijk te lezen, zeker voor de beginneling. Als een FORTH programma alleen maar in gecompileerde vorm beschikbaar is, is het zelfs onleesbar.

HOE WERKT FORTH

Forth werkt met twee stapelgeheugens (stacks) van het type LIFO (lastin-firstout): een getallenstack, meestal kortweg stack genoemd (2bytes per getal of adres) en een return-stack (2 bytes per adres)
De getallenstack dient om getallen of adressen in op te slaan,waarmee resp. op welks inhoud een bewerking wordt uitgevoerd volgens de'reversed polish notation'methode (RPN), ook genoemd'post fix notation'(zoals gebruikt in de HP rekenmachientjes, er is alleen geen ENTER instruktie).
In Forth zijn diverse stack operaties mogelijk, b.v. DUP kopieert de waarde van het bovenste getal en plaatst het bovenin de stack, dus:

1 2 3 DUP heeft als resultaat dat van boven naar beneden in de stack staan:

3 3 2 1
Op de returnstack staan in RUN TIME (daarnaast kennen we COMPILE TIME), adressen voor subroutines. Je kunt met de returnstack stoeien, maar weet wel wat je doet!!
In een BASIC programmaregel mag je alles achter elkaar tikken zonder spaties, maar in FORTH kan dat niet.

EDITOR en ASSEMBLER

Het werken aan een FORTH programma wordt aanzienlijk vergemakkelijkt als de FORTH versie voorzien is van een EDITOR. Met deze EDITOR wordt een programma in RAM geschreven en gewijzigd en daarna ongecompileerd in virtueel geheugen (of afgeschermde RAM) opgeborgen.
Als het programma daarna met het commando LOAD uit het virtuele geheugen in RAM wordt overgebracht, wordt het tijdens het laden geinterpreteerd en gecompileerd, maar blijft in ongecompileerde vorm in het virtuele geheugen aanwezig.
Werkt het programma niet naar wens, dan kan het met het commando EDIT in ongecompileerde vorm in RAM geladen worden en met de EDITOR gewijzigd en/of aangevuld worden.
Zonder EDITOR moet je erg veel meer werken met papier en potlood. Voor de hobbyist met veel tijd en geduld zal dat echter geen probleem vormen.
Een ASSEMBLER helpt bij het maken van machinetaal routines, die onder een naam (als'woord') in de dictionnaire worden bijgeschreven.

BASIC - FORTH

Twee voorbeelden die het verschil aangeven tussen BASIC en FORTH:

1 WAT IS HET RESULTAAT VAN 2*3+4
A - DIRECT MODE BASIC ?2*3+4
B - FORTH 2 3 * 4 + .
HIERIN IS . DE INSTUCTIE OF HET WOORD:
PRINT HET BOVENSTE GETAL VAN DE STACK EN
VERWIJDER HET UIT DE STACK.

ALS PROGRAMMA:
A - BASIC: 10 P=2*3+4:?P

B - FORTH: 0 VARIABLE P
: 2*3+4. 2 3 * 4 + TO P P . ;

: KONDIGT DE DEFINITIE AAN VAN HET NIEUWE WOORD
2*3+4.
; BEEINDIGT DE DEFINITIE.
HET WOORD 2*3+4. DAT GEEN SPATIES HEEFT WORDT AAN
DE FORTH DICTIONNAIRE TOEGEVOEGD.
ELKE KEER ALS WE HIERNA INTIKKEN 2*3+4. ZONDER
SPATIES WORDT HET RESULTAAT OP HET SCHERM GEPRINT.
DIT KAN IN ELK GETALSTELSEL DAT WE DEFINIEREN B.V:
7 BASE 2*3+4.
, GEEFT ALS RESULTAAT 13

2 ALS WE OP HET SCHERM OP PLAATS D11F EEN WIT BLOKJE WILLEN POKEN, DOEN WE DAT:
A - BASIC: POKE 53535,161

B - FORTH: DECIMAL 161 53535 C!
OF: HEX A1 D11F C!
C! BETEKENT: ZET OP HET ADRES IN DE TOP VAN DE

STACK HET CHARACTER MET ASCII- OF BYTEWAARDE DAT OP DE TWEEDE PLAATS IN DE STACK STAAT EN VERWIJDER BEIDE GETALLEN UIT DE STACK.

EEN NIEUW TYPE DEFINITIE

In FORTH bestaat de mogelijkheid een type woord te creeren dat FORTH nog niet kent. Dat gebeurt met de woorden BUIDS en DOES
Om duidelijk te maken hoe dat werkt,een voorbeeld.
De volgende nieuwe definitie geeft de mogelijkheid tonen te genereren van verschillende toonhoogte, verschillend timbre en verschillende duur.

```
Ø  VARIABLE N1
Ø  VARIABLE N2
:  TOON  BUILDS , , , DOES
         DUP DUP
         4 +   TO N2
         2 +   TO N1
           DO 4Ø FØØØ C!
              N2 Ø DO LOOP
              Ø FØØØ C!
              N1 Ø DO LOOP
           LOOP
;
6 8 2ØØ TOON HOI
```

Als we hierna intikken HOI en 'return', horen we een toon. De eerste twee getallen bepalen toonhoogte en timbre, het derde getal bepaalt de duur. Tussen BUILDS en DOES staan drie kommas. Hiermee wordt ruimte voor drie getallen in het dataveld van de nieuwe definitie gereserveerd. Bij de definitie van een TOON b.v. HOI moeten dus drie getallen op de stack staan. Het deel van de definitie achter DOES wordt uitgevoerd bij het aanroepen van een gedefinieerde toon. Eerst wordt het adres van het dataveld van TOON, dat bovenin de stack staat, tweemaal gedupliceerd. Daarna wordt de data die op de derde plaats in het dataveld staat (dataveld-adres +4) opgehaald ( ) en naar N2 overgebracht (dit is de waarde 6) Daarna wordt de data die op de tweede plaats in het dataveld staat (dataveld+2) opgehaald en naar N1 overgebracht. Vervolgens wordt het eerste getal in het dataveld (dit is de waarde 2ØØ) opgehaald. Hiermee wordt de grotr LOOP gemaakt (DO LOOP is hetzelfde als een FOR NEXT loop in BASIC). Vervolgens wordt op adres FØØØ (de bel) het getal 4Ø gepoked, er wordt even gewacht en het getal Ø wordt naar de bel geschreven, weer even gewacht. Dit bellen wordt 200 maal uitgevoerd.
Elke keer als we HOI intikken zal die TOON klinken.

SLOTOPMERKING

Als U beslag kunt leggen op een FORTH systeem (tape of disk) vraag er dan wel dokumentatie bij. Die dokumentatie bestaat minimaal uit een verklaring van de werking van de woorden in de dictionnaire.

John Hermans

Forth Interest Group (F.I.G.) is een groep in de Verenigde Staten van Amerika van ca 1200 personen, die meehelpen FORTH verder te ontwikkelen en bekend te maken. F.I.G. werkt aan een standaard voor FORTH. De meest recente is '79-STANDARD. Iedereen kan lid worden van F.I.G. en is dan tevens geabonneerd op FORTH DIMENSIONS, die overigens ook los verkrijgbaar zijn voor $ 2.-.
FORTH INTEREST GROUP P.O. BOX 1105 SAN CARLOS, CA 94070 U.S.A.
In Nederland is er FORTH ORIENTED RESEARCH TEAM HOLLAND (F.O.R.T.H.), dat voor de APPLE II de FORTH version 0.1 heeft ontwikkeld.
Rijks Universiteit FORTH Users Society (R.U.F.U.S.) werkt eraan deze versie van FORTH met FORTH'79-STANDARD in overeenstemming te brengen.
En voor FORTH belangstelling te wekken in Nederland.
Sinds kort bestaat de Hobbie Computer Club Forth Interest Group in oprichting. Deze groep staat open voor iedereen, die belangstelling heeft voor FORTH en mee wil helpen aan de verdere ontwikkeling van FORTH. Zie de H.C.C.Nieuwsbrief.

Het zal duidelijk zijn, dat net als BASIC de basis van FORTH (de primitieven, dit zijn de FORTH-woorden, die in machinetaal geschreven zijn) geschreven wordt voor een specifieke microprocessor.
De primitieven, die de periferie besturen, zijn systeem-eigen (O.S.I., Apple, etc.)
Er bestaan al een aantal versies van FORTH, die alle min of meer (of helemaal) de FORTH-standaard volgen. RUFUS ziet het als zijn taak de ontwikkeling van FORTH in Nederland te coördineren.

Sinds kort is er een goed amerikaans boek over FORTH, dat ook geschikt is voor beginners:
Leo Brodie: STARTING FORTH
Geschreven in opdracht van FORTH INC. en uitgegeven door PRENTICE-HALL, INC. De prijs is ongeveer $ 25.-.

J.Hermans october 1981

DISK-LOZE FIG-FORTH

MET DISK-SIMULATIE EN EENVOUDIGE TAPE-I/O.

CEES VAN LEEUWEN VAN DE O. S. I. G. G. HEEFT DE ASSEMBLER SOURCE LISTING VAN FIG-FORTH OP HET SUPERBOARD II EN DE CHALLENGER 1P EN 2P GEIMPLEMENTEERD.
JOHN HERMANS HEEFT EEN ZEER EENVOUDIGE 'TAPE READ'-SUBROUTINE EN UIT HET 'INSTALLATION MANUAL' DE 'RAM DISC SIMULATION' IN GECOMPILEERDE VORM INGEBOUWD IN DE KERNEL. HIERBIJ IS ALLEEN HET WOORD R/W HERSCHREVEN. DE BYTES, DIE OVERBLEVEN, ZIJN GEDEELTELIJK GEBRUIKT OM HET WOORD T> (TAPE READ) TE IMPLEMENTEREN. HET ML-DEEL VAN HET WOORD T> (TAPE READ) STAAT IN PAGE 2 ($028E-$02C2).
JOS BURGHOUTS HEEFT DE I/O-ROUTINES VOOR HET SUPERBOARD II GESCHREVEN. ZE STAAN IN PAGE 2: $02C3-02FE. HIERIN WORDT DE MONITOR AANGEROEPEN.
DEZE SUBROUTINES ZIJN SPECIFIEK VOOR HET SUPERBOARD II !!!

DEZE FIG-FORTH-VERSIE DRAAGT HET NUMMER 1.0, MAAR ALS GEVOLG VAN DE GENOEMDE VERANDERINGEN WIJKT HIJ DAARVAN AF.
OP CASSETTE 5 VINDT U:
- FIG-FORTH (KAAL);
- 11 EDITOR SCREENS, DIE GELADEN KUNNEN WORDEN IN HOGE RAM MET BEHULP VAN GENOEMDE 'TAPE READ';
- FIG-FORTH MET GECOMPILEERDE EDITOR;
- 11 EDITOR SCREENS TE LADEN MET QUICK SAVE VANAF $4000.

DE WIJZE, WAAROP NAAR TAPE GESCHREVEN WORDT IS PRIMITIEF, MAAR BIJ 300 BAUD GAAT HET REDELIJK TOT GOED.
DIT HANGT AF VAN DE GEBRUIKTE CASSETTE- OF BAND-RECORDER EN DE GEBRUIKTE TAPE.
HET BELANGRIJKSTE PROBLEEM, DAT OP KAN TREDEN, IS RUIS IN DE HEADER VAN EEN SCREEN, DIE ALS ALS ASCII GEINTERPRETEERD WORDT. ALS DIT OPTREEDT, ZAL HET SCREEN MET CMOVE VERPLAATST MOETEN WORDEN. HET BESTE ZOU ZIJN, DAT T> OP EEN HERKENNINGSTEKEN (B. V. ASCII $47) WACHT, ALVORENS EEN SCREEN IN TE LEZEN.
PROGRAMMEREN MAAR !!!

DE TAPE-I/O WERKT, ALS VOLGT:

SCHRIJVEN NAAR TAPE:

```
  TIK IN
        HEX 1 205 C! A B CR TYPE CR 0 205 C!
  START DE RECORDER VOOR OPNAME
  DRUK OP TOETS <RETURN>
        HIERIN IS A = BEGIN-ADRES VAN HET EERSTE SCREEN IN HEX
                  B = AANTAL BYTES (= $400 * AANTAL SCREENS)
  B. V. ' HEX 1 205 C! 4400 1C00 CR TYPE CR 0 205 C! '  (7 SCREENS)
```

LEZEN VAN TAPE:

```
  TIK IN
        HEX A C T>
  START DE RECORDER EN DRUK OP TOETS <RETURN>
        HIERIN IS A = BEGIN-ADRES VAN HET EERSTE SCREEN (IN HEX)
                  C = AANTAL PAGES (IN HEX) DAT GELADEN
                      MOET WORDEN (= 4 * AANTAL SCREENS)
                  T> = DE INGEBOUWDE 'TAPE READ'
  B. V. ' HEX 5000 8 T> '  (2 SCREENS)
```

ZOWEL BIJ LEZEN ALS SCHRIJVEN ZIET U DE TEKST OP HET SCHERM
VERSCHIJNEN.

ALVORENS FORTH VAN TAPE TE LADEN MOET U EERST UW MICROCOMPUTER
EEN KOUDE BASIC-START GEVEN. ZO KRIJGEN ALLE POINTERS DE GOEDE
WAARDE, ZODAT DE MONITOR EN FORTH ERMEE KUNNEN WERKEN.

ALS U EEN SCREEN WILT 'LOAD'-EN DAN EERST TERUG NAAR FORTH EN DAARNA
INTIKKEN EMPTY-BUFFERS .

OM EEN 'VLIST' TE ONDERBREKEN MOET U OP <ESC> DRUKKEN.
LOSLATEN VAN <ESC> LAAT VLIST VERDER GAAN.
MET MON <RETURN> KOMT U IN DE MONITOR-MODE.
TERUG NAAR FORTH KOMT U MET
   0300G (KOUDE START)
   0304G (WARME START)
RAAKT U IN DE KNOOP EN WILT U EEN NIEUWE (KOUDE) START
DAN KAN DAT DUS VIA MON.

DE FORTH, DIE OP CASSETTE STAAT, WERKT MET HOGE RAM VAN $4000
TOT $7FFF (DUS 32K RAM).
HET EERSTE SCREEN ($4000-$43FF) IS SCREEN 0 EN KAN MET
DEZE VERSIE VAN R/W NIET GELADEN ('0 LOAD') WORDEN.
ALS UW SYSTEEM MINDER DAN 32K RAM HEEFT, MOET U DE VOLGENDE
BYTES WIJZIGEN.

	$0311	$09A3	$17B2	$17B6	AANTAL SCREENS
16K	$2F	$2F	$30	$3C	4
20K	$37	$37	$38	$4C	6
24K	OK	OK	OK	$5C	8
28K	OK	OK	OK	$6C	12

DE GEBRUIKTE EDITOR IS DIE VAN. FIG-FORTH INSTALLATION MANUAL
RELEASE 1 VAN WILLIAM F. RAGSDALE, FORTH INTEREST GROUP NOV 1980

BIJ HET GEBRUIK MOET U NOG OP HET VOLGENDE LETTEN.
- DE <RUB OUT>-TOETS WERKT WEL, MAAR DIT IS NIET OP HET SCHERM
  TE ZIEN;
  DAT IS DUS OUDERWETS TELLEN GEBLAZEN !!!
- DE MEEST UITGEBREIDE INSTRUCTIES OVER DEZE VERSIE VAN FORTH KUNT
  U VINDEN IN EERDER GENOEMD INSTALLATION MANUAL, MAAR OOK IN DE
  FIG-FORTH 6502 ASSEMBLY SOURCE LISTING, RELEASE 1.1
  VAN SEPTEMBER 1980;
- ALS UW RAM-GEHEUGEN NIET TOEREIKEND IS KUNT U DE SCREENS OOK VAN
  HET FORTH-BANDJE BINNENHALEN MET BEHULP VAN DE CROSSLOADER VAN
  JAN EN BAS EDIXHOVEN (ZIE OSI-POEL, EE15)
  BOVENDIEN ZIJN DE EDITOR-SCREENS VAN KANT A STUK VOOR STUK BINNEN
  TE HALEN, ZE HEBBEN EEN GENUMMERDE HEADER EN TUSSEN DE SCREENS
  IS STEEDS 10 SEC TUSSENRUIMTE GELATEN.

DISK-LOZE FORTH

OVERZICHT VAN DE PLAATSEN IN DE 'KERNEL' WAAR DE
I/O-ROUTINES, DIE IN PAGINA 2 STAAN, WORDEN AANGEROEPEN.

SUBROUTINE	PAG. 2 ADRES	KERNEL ADRES
CRLF	02C3	0371
		03C8
HEX2	02D2	037B
		0395
		039A
		03A2
		03AA
		03B0
LETTER	02FA	03CD
		03EE
XBLANK	02CD	037E
		039D
		03A5
		03B3
		03D0
		03F5

DE TABEL OP PAGINA FD 8 IS ONVOLLEDIG.
DE VOLLEDIGE TABEL IS:

ADRES	0311	0997	09A3	14C5	14D0	17B2	17B6	AANTAL SCREENS
16K	2F	2B	2F	2B	2B	30	3C	4
20K	37	33	37	33	33	38	4C	6
24K	OK	OK	OK	OK	OK	OK	5C	8
28K	OK	OK	OK	OK	OK	OK	6C	12
32K	OK	OK	OK	OK	OK	OK	OK	16
36K	OK	OK	OK	OK	OK	OK	8C	20
40K	OK	OK	OK	OK	OK	OK	9C	24

FEBRUARI 1983

RAM DISC SIMULATION AND TAPE READ FOR C1P

(SEE FIG-FORTH INSTALLATION MANUEL 11/81 PAGE 3)

```
1783                    .WORD RAM DISC SIMULATION
1783  83 52 2F D7       .BYTE $83, 'R/W'
1787  6A 17             .WORD BCD  LINK TO BCD
1789  D6 08             .WORD DOCOL
178B  3A 07             .WORD TOR
178D  AC 09             .WORD BBUF
178F  44 14             .WORD STAR
1791  AF 17             .WORD LO
1793  99 07             .WORD PLUS
1795  3A 08             .WORD DUP
1797  B3 17             .WORD HI
1799  2A 03             .WORD LIT,$0006
179D  06 00
179F  11 0C             .WORD QUERR
17A1  4C 07             .WORD RFROM
17A3  56 04             .WORD ZBRAN
17A5  04 00             .WORD 4
17A7  22 08             .WORD OVER
17A9  AC 09             .WORD BBUF
17AB  FE 05             .WORD CMOVE
17AD  0C 07             .WORD SEMIS
17AF  15 09             .WORD DOCON  'LO'
17B1  00 40             .WORD $4000
17B3  15 09             .WORD DOCON  'HI'
17B5  00 7C             .WORD $7C00
17B7                    .WORD TAPE TO RAM
17B7  82 54 BE          .BYTE T>
17BA  83 17             .WORD R/W  LINK TO R/W
17BC  D6 08             .WORD DOCOL
17BE  2A 03             .WORD LIT,$0299
17C0  99 02
17C2  B5 08             .WORD CSTORE
17C4  2A 03             .WORD LIT,$029D
17C6  02 9D
17C8  9D 08             .WORD STORE
17CA  F4 05             .WORD CR
17CC  85 02             .WORD T>
17CE  0C 07             .WORD SEMIS
```

DISK-LOZE FORTH

'T>' VERSIE 2

```
* 0285 8702        FORTH SPRINGT NAAR $0285 EN VANDAAR NAAR
* 0287 48      PHA DE MACHINE CODE SUBROUTINE VAN 'T>'.
* 0288 8A      TXA
* 0289 48      PHA
* 028A 98      TYA
* 028B 48      PHA
* 028C A9FF    LDA ##$FF
* 028E 8D0302  STA $0203 SET LOAD FLAG
* 0291 2080FE  JSR $FE80 'ACIA IN' HAAL KARAKTER VIA ACIA
* 0294 C947    CMP ##$47  IS HET EEN 'G' ?
* 0296 D0F9    BNE $0291  ZO NIET KIJK WEER NAAR ACIA
  0298 A004    LDY ##$04  TE LADEN AANTAL PAGES
                          WORDT DOOR FORTH VERANDERD
  029A A200    LDX ##$00  TELLER
  029C 2080FE  JSR $FE80 'ACIA IN'
  029F 9D0044  STA $4400,X SCHRIJF IN HOGE RAM SCREEN
                          HB WORDT DOOR FORTH VERANDERD
* 02A2 8D70D1  STA $D170 TOON KARAKTER OP HET SCHERM
  02A5 E8      INX        HOOG TELLER OP
  02A6 D0F4    BNE $029C TELLER <> 0 ? DAN TERUG
  02A8 88      DEY        PAGES = PAGES - 1
  02A9 F006    BEQ $02B1 PAGES = 0 ? DAN SPRING
  02AB EEA102  INC $02A1 VERHOOG HB HOGE RAM ADRES
  02AE 4C9C02  JMP $029C BEGIN AAN NIEUWE PAGE
  02B1 EAEAEA  NOP
  02B4 EA EA   NOP
  02B6 A900    LDA ##$00
  02B8 8D0203  STA $0203 CLEAR LOAD FLAG
  02BB 68      PLA
  02BC A8      TAY
  02BD 68      PLA
  02BE AA      TAX
  02BF 68      PLA
  02C0 4C4403  JMP $0344 SPRING NAAR 'NEXT'
```

DEZE VERSIE VAN 'T>' SCHRIJFT PAS KARAKTERS NAAR HOGE RAM
ALS EERST EEN 'G' IS AANGEBODEN.
DE SCREENS MOETEN DUS NAAR TAPE GESCHREVEN ZIJN ALS VOLGT

HEX 1 205 C! WACHT ." G" WACHT AAAA NNNN TYPE WACHT 0 205 C!

HIERIN IS 'WACHT' EEN WACHTLUS.
AAAA = START ADRES VAN DE HOGE RAM SCREENS
NNNN = AANTAL BYTES, DAT NAAR TAPE GESCHREVEN WORDT.

LADEN KAN OP DE WIJZE VAN DE OUDE 'T>'-VERSIE.
ALLEEN WORDT NU STEEDS MAAR 1 BYTE OP HET SCHERM GESCHREVEN.

IN DE KERNEL VAN FORTH MOET DE INHOUD VAN BYTE $17CC
GEWIJZIGD WORDEN IN ##$85.

* NIEUW OF VERPLAATST.

FEBRUARI 1983

OPZET VAN EEN FORTH T-APE O-PERATING S-YSTEEM

OM EFFECTIEF MET FORTH TE KUNNEN WERKEN MOET DE FORTH-PROGRAMMEUR OVER EEN EDITING SYSTEEM KUNNEN BESCHIKKEN. MET BEHULP VAN ZO'N EDITING PROGRAMMA KAN EEN PROGRAMMA MET FORTH KOMMANDO'S EN BIJBERHOREND KOMMENTAAR WORDEN OPGEBOUWD. DÈ TEKST VAN HET PROGRAMMA KAN DAN WORDEN BEWAARD (OP FLOPPY-DISC, KASSETTEBAND OF OP PAPIER), EN DAARNA AAN FORTH WORDEN AANGEBODEN, DIE DAT PROGRAMMA KOMPILEERT (IN VOOR FORTH BEGRIJPELIJKE OPDRACHTEN OMZET). OMDAT EEN EENMAAL GEKOMPILEERD PROGRAMMA MOEILIJK WEER TERUG OM TE ZETTEN IS IN FORTH-TEKST, IS HET HET BESTE OM DE BASIS-PROGRAMMATEKST (DE BRON, 'THE SOURCE') TE BEWAREN.

NA DEZE KORTE INLEIDING ZAL IEDEREEN BEGRIJPEN, DAT FORTH IN KOMBINATIE MET EEN DISK-DRIVE DE MEEST IDEALE KOMBINATIE VORMT. DAT IS DAN OOK DE MEEST GEBRUIK-TE BASISOPZET. MAAR, ZONDER DISK-DRIVE KAN HET OOK.... ER ZIJN DAARBIJ TWEE MOGELIJKHEDEN OM EEN DISK OPERATING SYSTEM (DOS) TE SIMULEREN:

1. GEBRUIK EEN KASSETTE-REKORDER IN PLAATS VAN DE DISK-DRIVE.
2. GEBRUIK BOVENDIEN RAM-PAGES (ACHTERAAN IN HET GEHEUGEN) VOOR TUSSENOPSLAG VAN SCREENS (SIMULATIE VAN OP DISK OPGESLAGEN SCREENS = FORTH-EDITOR PROGRAMMA-BLOKKEN).

AD 1. EEN TAPE OPERATING SYSTEEM IS UITERAARD SEQUENTIEEL, DAT WIL ZEGGEN, DAT DE GEGEVENS NA ELKAAR OP DE BAND STAAN. ZODOENDE KAN ER NIET ZONDER MEER NAAR HET GEWENSTE SCREEN WORDEN GESPRONGEN, ZOALS DAT BIJ EEN DISK GEBEURT, EN KOST HET OPZOEKEN VAN SCREENS VEEL TIJD BIJ GEBRUIK VAN EEN AUDIO-KASSETTE-REKORDER ALS VIRTUEEL GEHEUGEN.
BIJ HET WERKEN MET SNELHEDEN VAN 1200 BAUD KAN DIT ACCEPTABEL ZIJN, ZEKER ALS WORDT GEWERKT MET SCREEN-GROEPEN (LANGERE PROGRAMMA'S).

AD 2. HET SIMULEREN VAN DISK-SCREENS DOOR HOGE RAM-PAGES WERKT ZELFS SNELLER, ALS HET TELKENS LEZEN EN SCHRIJVEN VAN EN NAAR DISK.
MAAR ER MOET DAN MINSTENS 16K RAM BESCHIKBAAR ZIJN EN DE R/W-ROUTINE MOET AANGEPAST ZIJN (ZIE DAARVOOR HET FIG-FORTH INSTALLATION MANUEL OP PAGINA 3 <RAM-DISC-SIMULATION> OF OSI POEL FD 9/11).
NET ALS BIJ GEBRUIK VAN DISK KUNNEN DE PROGRAMMA'S MET BEHULP VAN DE EDITOR IN SCREENS WORDEN GESCHREVEN, DIE DAN ACHTER DE GEHEUGENRUIMTE VOOR FORTH IN RAM WORDEN WEGGEZET (MET 'FLUSH').
DEZE SCREENS KUNNEN OP EEN AANTAL MANIEREN OP TAPE WORDEN OPGESLAGEN:

A. MET EEN (EXTENDED) MONITOR PROGRAMMA (IN ROM OF IN RAM) OF EEN ANDER PROGRAMMA IN MACHINE-CODE, DAT GEDEELTEN VAN HET GEHEUGEN KAN SAVEN EN LADEN.
B. MET EEN SAVE-LOAD PROGRAMMA IN BASIC, ZOALS B.V. HET PROGRAMMA VAN PETER BROERS (VOOR B.V. OSI-SUPERBOARD) UIT HCCN 17, PAGINA 14.
C. MET EEN R/W-ROUTINE IN FORTH VOOR TAPE, DIE NU IN EEN EENVOUDIGE VORM VOOR OSI BESCHIKBAAR IS, EN WAAROP HIERONDER NOG WORDT TERUGGEKOMEN.

FEBRUARI 1983

NATUURLIJK KUNNEN OOK DISK-GEBRUIKERS MET EEN
NOG NIET OP FORTH AANGEPASTE DOS OF R/W ROUTINE
MET HUN EIGEN DOS DE SCREENS NAAR DISK VERPLAATSEN.

BIJ HET WERKEN MET DISK-SIMULATIE ZOU HET GEHEUGEN,
ALS VOLGT, INGEDEELD KUNNEN WORDEN:

GEHEUGEN ->	16K	20K	24K+
FORTH+USER	$0000-27FF	0000-2FFF	0000-37FF
BLOCKS	$2800-2FFF	3000-37FF	3800-3FFF
SCREENS	$3000-37FF	3800-4FFF	4000
AANTAL SCREENS	4	6	8+

VOOR DE AANPASSING VAN FORTH ZIE DE TABEL:
OSI POEL FD 9

WE KUNNEN NU MIN OF MEER OP EEN RIJTJE ZETTEN, HOE PUNT 2.C:
DE FORTH TAPE-VERSIE VOOR DE R/W-ROUTINE, UITGEWERKT MOET
WORDEN TOT EEN ECHTE TOS.

1. HET TOS MOET AFGEPASTE STUKKEN PROGRAMMATEKST KUNNEN
INLEZEN VAN BAND. HET VERDIENT AANBEVELING OM AFGEPASTE
STUKKEN TE GEBRUIKEN. NIET ALLEEN SLUIT DAT AAN BIJ
HET GEBRUIK VAN DISC-FORTH SYSTEMEN, OOK DE ADMINISTRA-
TIEVE VERWERKING VOOR HET TOS WORD' SIMPELER.
DE PROGRAMMA-TEKST KAN VERVOLGENS MET DE EDITOR WORDEN
BEWERKT, WAARBIJ KOMMANDO'S/REGELS WORDEN TOEGEVOEGD
OF WEGGEHAALD. DE GEWIJZIGDE TEKST MOET WEER OP TAPE
KUNNEN WORDEN WEGGESCHREVEN. TENSLOTTE KAN DE TEKST AAN
FORTH WORDEN AANGEBODEN. FORTH ZAL DIE TEKST OMZETTEN, ZODAT
DAARNA HET PROGRAMMA KAN WORDEN 'GERUND'.
IN OVEREENSTEMMING MET DE DISC-SYSTEMEN ZOU HET FORTH
LEES-KOMMANDO 'LIST' EN HET KOMMANDO VOOR KOMPILATIE
'LOAD' MOETEN HETEN.
BOVENDIEN MOET OPGEGEVEN KUNNEN WORDEN, HOE HET BLOK
HEET, DAT VAN BAND GELEZEN MOET WORDEN (B.V. BRIEF27
SCR#38).
DE MOGELIJKHEID MOET BESTAAN DIRECT VAN TAPE TE KOM-
PILEREN MET B.V. 'CLOAD' OF EERST NAAR HOGE RAM
IN TE LEZEN EN VAN DAAR OP KOMMANDO TE KOMPILEREN.
HET MOET DAAROM MOGELIJK ZIJN OP TE GEVEN IN WELK
DEEL VAN HET RAM-GEHEUGEN DE SCREENS MOETEN WORDEN
OPGESLAGEN.
GEVOLG VAN HET EEN EN ANDER IS DAT:
- HET TOS ZELF DE JUISTE PASSAGE IN HOGE RAM OPSLAAT
EN DE GEBRUIKER ALLEEN DE BAND EEN STUK(JE) VOOR DE
WAARSCHIJNLIJKE PLAATS MOET STARTEN.
- HET TOS (ZO MOGELIJK) DE KASSETTE-REKORDER KAN
STOPPEN, ALS HET STUK IS BINNENGELADEN EN KAN MELDEN
OF DAT GOED GEGAAN IS (DOOR B.V. EEN CHECKSUM LOADER
TE GEBRUIKEN).
- HET TOS EEN ADMINISTRATIE BIJHOUDT VAN WAT IN WELKE
DELEN VAN HOGE RAM STAAT, EN OF ER GEEN SCREENS IN
HOGE RAM WORDEN OVERSCHREVEN BIJ HET BIJLADEN VAN
DE NIEUWE SCREENS. DOOR BIJVOORBEELD BEPAALDE
RAM-SCREENS TE BESCHERMEN, ZULLEN ALLEEN ALS DAARTOE
UITDRUKKELIJK OPDRACHT GEGEVEN WORDT (DOOR EEN
'N1 N2 UNPROTECT' OPDRACHT) SCREENS WORDEN OVER-
SCHREVEN.
- HET TOS MOET DUS EEN KOMMANDO 'N1 N2 PROTECT' KENNEN,
DAT SCREENS BESCHERMT.

FEBRUARI 1983

2. HET TOS MOET DE (STUKKEN) PROGRAMMA-TEKST, DIE MET DE EDITOR ZIJN OPGEBOUWD KUNNEN WEGSCHRIJVEN NAAR DE KASSETTE-REKORDER. OM HET LADEN ONDER NAAM MOGELIJK TE MAKEN MOET OOK HET NAAR TAPE SCHRIJVEN ONDER NAAM GEBEUREN. BOVENDIEN IS HET GEMAKKELIJK (MAAR DAT IS AL EERDER VERMELD) OM EEN ZEKERE VORM VAN KONTROLE MEE TE STUREN (B.V. MET EEN CHECK SUM KONTROLE). DE NAAM VAN HET KOMMANDO KAN 'CFLUSH' ZIJN, ALS HET EEN SCREEN BETREFT, TERWIJL INFORMATIE BETREFFENDE HET NUMMER VAN HET (INTERNE) WEG TE SCHRIJVEN SCREEN OF ZIJN NAAM, WAARONDER HET OP TAPE MOET KOMEN TE STAAN OF AL STAAT, MOET WORDEN MEEGEGEVEN (B.V. 'ARTIKEL7 CFLUSH')

IN PRINCIPE IS HET TOS, ZOALS HET HIER BESCHREVEN IS, VOLDOENDE OM EENVOUDIG EN DOELTREFFEND MET KASSETTE-REKORDERS BINNEN FORTH TE KUNNEN WERKEN.
NAMELIJK:
== HET IS MOGELIJK OM PROGRAMMATEKST VAN BAND TE LADEN OM DIE, DESGEWENST, DAARNA MET EEN EDITOR TE BEWERKEN.
== MET EEN EDITOR OPGEBOUWDE OF BEWERKTE PROGRAMMATEKST KAN NAAR TAPE WORDEN GE-'CFLUSH'-ED.
== PROGRAMMATEKST KAN AAN FORTH WORDEN AANGEBODEN IN EEN ZODANIGE VORM, DAT FORTH ERMEE KAN WERKEN, HETZIJ DIRECT MET 'CLOAD', HETZIJ VIA HOGE RAM.

WAT DAN NOG AAN EEN VOLLEDIGE TOS ONTBREEKT ZIJN MOGELIJKHEDEN OM:
A. TE KIJKEN WAT ER IN HET GEHEUGEN IS GELADEN VAN TAPE
B. WELKE PROGRAMMA'S OP DE TAPE STAAN.
AD A. MET HET KOMMANDO 'N1 N2 INDEX' KAN EEN OVERZICHT WORDEN GEVRAAGD, WAARBIJ VAN EEN AANTAL (N1 T/M N2) BESCHIKBARE (INTERNE) SCREENS REGEL 0 OP HET SCHERM WORDT GEPRINT, ZODAT MEN KAN ZIEN:
** OF ZE GEVULD ZIJN
** MET WELKE NAAM ZE EVENTUEEL VAN TAPE ZIJN GEHAALD
DAARNAAST MOET INFORMATIE VERKRIJGBAAR ZIJN:
** OF DE SCREENS BESCHERMD OF NIET BESCHERMD ZIJN
** OF ZE AL AAN FORTH ZIJN AANGEBODEN GEDURENDE DE LOPENDE SESSIE.
AD B. MET DE OPDRACHT 'KIJK' MOET DE TAPE KUNNEN WORDEN AFGEZOCHT NAAR SCREEN-NAMEN. EVENTUEEL KAN HIERBIJ OOK WEER REGEL 0 WORDEN GEPRINT, WAAR VAAK TOELICHTING OP DE INHOUD VAN HET SCREEN WORDT VERMELD, B.V. MET HET KOMMANDO 'KIJK+'.

DE KOMMANDO'S ONDER B VERMELD, ZIJN NIET STRIKT NOODZAKELIJK MAAR ZULLEN AARDIG WAT HANDMATIGE ADMINISTRATIE UIT HANDEN VAN DE GEBRUIKER NEMEN.
DIT VERHAAL GEEFT ALLEEN EEN OPZET VAN EEN TOS, ZOALS OOK UIT DE TITEL TE LEZEN IS. OP DIT MOMENT (1-1983) ZIJN WIJ ZELF NOG NIET IN STAAT OM OOK WERKELIJK ZO'N TOS (IN FORTH !!) TE PROGRAMMEREN, MAAR EEN SOORT OVEREENSTEMMING OVER DE VORM VAN EEN TOS IS AL VAAK HET HALVE WERK. WIJ HOUDEN ONS IN IEDER GEVAL BIJZONDER AANBEVOLEN VOOR OPMERKINGEN OVER EN KRITIEK OP DEZE OPZET, EN NATUURLIJK OOK VOOR (GEHELE OF GEDEELTELIJKE) TOS'SEN, DIE AL GEMAAKT ZIJN EN NOG GAAN WORDEN.

JOS BURGHOUTS
FEBRUARI 1983 JOHN HERMANS

F Ø R T H INTRØDUKTIE

IN HET VØLGENDE VERHAAL STAAN MIJN EERSTE STAPPEN ØP HET FØRTH TERREIN. HET GEHEEL IS GESCHREVEN IN FØRTH ZELF MET BEHULP VAN DE EDITØR.
BESCHREVEN WØRDT WAT DIE EDITØR IS, HØE JE HEM KUNT GEBRUIKEN EN WAT DIE EDITØR MET FØRTH TE MAKEN HEEFT.

16K IS WEL DE MINIMAAL NØØDZAKELIJKE ØMVANG VAN HET RAM-GEHEUGEN.

IN VØLGENDE BØEKJES ZAL IK MIJN VØRDERINGEN MELDEN.

JØS BURGHØUTS

MET 'EMIT' KAN EEN BEPAALD KARAKTER NAAR HET BEELDSCHERM EN DUS ØØK PRINTER WØRDEN GESTUURD. DØØR DE ØPDRACHT : '26 EMIT' WØRDT HETZELFDE EFFEKT VERKREGEN ALS IN BASIC 'PRINT CHR$(26)'. MET 'CLS' ERVØØR EN INGEPAKT DØØR EEN ':' EN EEN ';' WØRDT DIT EEN SØØRT SUBRØUTINE DIE ØP TE RØEPEN IS DØØR CLS IN TE TYPEN

SAMEN MET 'T ."KØMMANDØ DAT ØVEREENKØMT MET DE 'PRINT'-ØPDRACHT IN BASIC IS UIT TE LEGGEN HØE DE TEKST DIE HIER AFGEDRUKT STAAT IS GEMAAKT. ALLEREERST MAAR EENS EEN LISTING ZØALS FØRTH DIE GEEFT

```
SCR # 1
  0 ( FØRTH INTRØDUKTIE EN BEGRIPPEN DEMØNSTRATIE)
  1 DECIMAL : CLS 26 EMIT : ( MAAK HET SCHERM SCHØØN )
  2 : TEKST1 ( MET DIT KØMMANDØ WØRDT TØELICHTING BIJ CLS GEGEVEN)
  3 ." MET 'EMIT' KAN EEN BEPAALD KARAKTER NAAR HET BEELDSCHERM" CR
  4 ." EN DUS ØØK PRINTER WØRDEN GESTUURD. DØØR DE ØPDRACHT :" CR
  5 ." '26 EMIT' WØRDT HETZELFDE EFFEKT VERKREGEN ALS IN" CR
  6 ." BASIC 'PRINT CHR$(26)'.  MET 'CLS' ERVØØR EN INGE-" CR
  7 ." PAKT DØØR EEN ':' EN EEN ';' WØRDT DIT EEN SØØRT SUBRØUTINE"
  8 CR ." DIE ØP TE RØEPEN IS DØØR CLS IN TE TYPEN" CR CR
  9 ." SAMEN MET 'T ." 34 EMIT ." KØMMANDØ DAT ØVEREENKØMT" CR
 10 ." MET DE 'PRINT'-ØPDRACHT IN BASIC IS UIT TE LEGGEN" CR
 11 ." HØE DE TEKST DIE HIER AFGEDRUKT STAAT IS GEMAAKT. " CR
 12 ." ALLEREERST MAAR EENS EEN LISTING ZØALS FØRTH DIE GEEFT" CR ;
 13 -->
 14
 15
```

'DECIMAL' HØUDT IN DAT DE CIJFERMATIGE IN- EN UITVØER IN
HET 10-TALLIGE STELSEL GEBEURT. VERDER WØRDT HET NIEUWE
KØMMANDØ 'CLS' GEDEFINIEERD EN TØEGELICHT.
IN REGEL 2 WØRDT HET KØMMANDØ 'TEKST1' ØPGEZET, EN PAS IN
REGEL 12 AFGESLØTEN MET EEN ';'.
DE REGELS 3 EN VØLGENDE STARTEN STEEDS MET EEN PUNT-AAN-
HALINGSTEKENS. DE TEKST TØT DE VØLGENDE SLUITTEKENS
WØRDT ØP HET SCHERM (ØF PRINTER) AFGEDRUKT.
DE 'CR' GEEFT AAN DAT ER EEN CARRIAGE-RETURN KØMT.
PAS IN SCR#6 WØRDT HET WEER INTERESSANT.
DAAR WØRDT NADAT ALLE TEKSTEN ZIJN GEDEFINIEERD IN
REGEL 11 HET KØMMANDØ 'INTRØ' GENØEMD, DAT BESTAAT UIT
DE CLS, TEKST1, DE LISTING VAN SCR#1 EN DE REST VAN DE
TEKSTEN. LET WEL DAT BIJ DE LISTING VAN DE RESTERENDE
SCHERMEN HET LAATST GENØEMDE GETAL 'T EERST WØRDT GEPAKT
ØP REGEL 13 WØRDT DE PRINTER AANGEZET, INTRØ UITGEVØERD,
EN DE PRINTER WEER UITGEZET.
DØØR DE ØPDRACHT 'INTRØ' TE GEVEN WØRDT HET GEHELE
 VERDER KLAARGEMAAKTE PRØGRAMMA UITGEVØERD.

```
SCR # 2
  0 ( VERVØLG FØRTH-INTRØ)
  1 : TEKST2 ." WE LØPEN DE LISTING REGEL VØØR REGEL DØØR." CR CR
  2 ." BØVENAAN STAAT SCR# 1." CR ." DAT IS DE PLEK WAAR IK BIJ HET"
  3 ." MAKEN VAN DEZE TEKST" CR ." DE LISTING HEB ØPGEBØRGEN." CR
  4 ." IN BASIC HØEF JE JE DAARØVER NIET TE BEKØMMEREN, ECHTER" CR
  5 ." IN FØRTH MØET JE GØED WETEN WAAR WAT STAAT IN 'T GEHEUGEN" CR
  6 ." EN BØVENAL HØE DAT ER STAAT !!" CR CR
  7 ." DE LISTING VAN DE SCR#1 (SCREEN NUMMER 1) HEEFT ALLES" CR
  8 ." TE MAKEN MET DE EDITØR." CR
  9 ." MET DIE EDITØR ( WAT EEN FØRTH-PRØGRAMMA IS), DIE IS" CR
 10 ." BESCHREVEN IN DE OSI-PØEL EE 30 T/M 42, BØEKJE 6" CR
 11 ." KUNNEN TEKSTEN IN HET GEHEUGEN WØRDEN GETYPT. " CR CR
 12 ." EERST DE WERKING VAN DE EDITØR." CR
 13 ." SCHAF EERST EEN FØRTH MET EDITØR AAN ( BANDJE 5)" CR
 14 ." DAARVØØR MØET HET SUPERBØARD MINIMAAL 16 K HEBBEN" CR ;
 15 -->

SCR # 3
  0 ( VERVØLG FØRTH INTRØDUKTIE)
  1 : TEKST3 ." NA HET INLADEN VAN DE FØRTH & EDITØR MELDT DE" CR
  2 ." KØMPUTER ZICH MET FIG-FØRTH ETC.  " CR CR
  3 ." TYPE IN 'EMPTY-BUFFERS' RETURN EN 'EDITØR' RETURN" CR
  4 ." NU STAAN ALLE KØMMANDØ'S VAN DE EDITØR GEREED (ZIE ØØK" CR
  5 ."  DE ØPSØMMING IN BØEKJE 7). " CR
  6 ." MAAK NU HET WERKGEBIED VØØR DE EDITØR SCHØØN MET 1 CLEAR" CR
  7 ." DAARMEE WØRDEN SPATIES IN SCREEN (=SCHERM) 1 GEZET. " CR
  8 ." DAT IS TE KØNTRØLEREN MET 1 LIST (REGELNRS EN SPATIES)" CR
  9 ." JE BENT NU IN SCHERM 1 WAARVAN DE REGELS IN TE VULLEN" CR
 10 ." ZIJN. DE PRØCEDURE IS ALS VØLGT :" CR
 11 ." TYPE EERST HET REGELNUMMER MET EEN SPATIE, DAARNA EEN" CR
 12 ." P <SPATIE> EN DE REGEL ZØALS VØØR FØRTH NØDIG IS" CR
 13 ." ALLE ANDERE KØMMANDØ'S VAN DE EDITØR KUNNEN WØRDEN GEBRUIKT"
 14 CR ." ØM REGELS ZØ AAN TE PASSEN ALS DAT NØDIG IS. " CR CR ;
 15 -->
```

```
SCR # 4
  0 ( VERVOLG FORTH INTRODUKTIE)
  1 : TEKST4 CR CR ." ZODRA JE HET IDEE HEBT DAT DE REGELS DIE" CR
  2 ." IN DE SCREEN STAAN JUIST ZIJN, SLUIT JE DE LAATSTE REGEL" CR
  3 ." AF MET ';S' EN KUN JE HET SCHERM GAAN VERTALEN (COMPILE-" CR
  4 ." REN). WANNEER ER FOUTEN WORDEN GEMELD, GA JE TERUG NAAR" CR
  5 ." DE EDITOR EN VERBETER JE DE TEKST. " CR
  6 ." LET WEL DAT BIJ HET KOMPILEREN NIEUWE KOMMANDO'S WORDEN" CR
  7 ." TOEGEVOEGD AAN DE FORTH-KERN (KONTROLEREN MET 'VLIST')" CR
  8 ." EVENTUELE OVERBODIGE FUNKTIES (SUBROUTINES) ZIJN TE VER-" CR
  9 ." WIJDEREN DOOR 'FORGET <FUNKTIE>'. " CR CR
 10 ." TERUG NAAR DE LISTING VAN SCR# 1. " CR
 11 ." HET IS EEN GOED GEBRUIK OM OP REGEL 0 DE TITEL VAN HET" CR
 12 ." SCHERM TE VERMELDEN.  MET HET KOMMANDO <EERSTE><LAATSTE>" CR
 13  ." 'INDEX' (B.V. 1 4 INDEX) KUNNEN NAMELIJK DE 0 REGELS" CR
 14 ." VAN DE SCREENS WORDEN BEKEKEN. " CR ;
 15 -->

SCR # 5
  0 ( VERVOLG FORTH INTRODUKTIE)
  1 : TEKST5 ." EEN HAAKJE OPENEN <SPATIE> GEEFT AAN DAT ER KOM-" CR
  2 ." MENTAAR VOLGT, HAAKJE SLUITEN = EINDE KOMMENTAAR. " CR
  3 ." LET ER WEL OP DE FORTH-OPDRACHTEN VAN ELKAAR TE SCHEIDEN" CR
  4 ." ANDERS KAN DE KOMPILER DE KOMMANDO'S,VARIABELENNAMEN ETC" CR
  5 ." NIET UIT ELKAAR HOUDEN. " CR
  6 ." 'DECIMAL' HOUDT IN DAT DE CIJFERMATIGE IN- EN UITVOER IN" CR
  7 ." HET 10-TALLIGE STELSEL GEBEURT.  VERDER WORDT HET NIEUWE" CR
  8 ." KOMMANDO 'CLS' GEDEFINIEERD EN TOEGELICHT. " CR
  9 ." IN REGEL 2 WORDT HET KOMMANDO 'TEKST1' OPGEZET, EN PAS IN" CR
 10  ." REGEL 12 AFGESLOTEN MET EEN ';'. " CR
 11 ." DE REGELS 3 EN VOLGENDE STARTEN STEEDS MET EEN PUNT-AAN-" CR
 12 ." HALINGSTEKENS. DE TEKST TOT DE VOLGENDE SLUITTEKENS " CR
 13 ." WORDT OP HET SCHERM (OF PRINTER) AFGEDRUKT. " CR
 14 ." DE 'CR' GEEFT AAN DAT ER EEN CARRIAGE-RETURN KOMT. " CR ;
 15 -->

SCR # 6
  0 ( EINDE FORTH INTRODUKTIE )
  1 : TEKST6 ." PAS IN SCR#6 WORDT HET WEER INTERESSANT. " CR
  2 ." DAAR WORDT NADAT ALLE TEKSTEN ZIJN GEDEFINIEERD IN " CR
  3 ." REGEL 11 HET KOMMANDO 'INTRO' GENOEMD, DAT BESTAAT UIT" CR
  4 ." DE CLS, TEKST1, DE LISTING VAN SCR#1 EN DE REST VAN DE" CR
  5 ." TEKSTEN. LET WEL DAT BIJ DE LISTING VAN DE RESTERENDE" CR
  6 ." SCHERMEN HET LAATST GENOEMDE GETAL 'T EERST WORDT GEPAKT" CR
  7 ." OP REGEL 13 WORDT DE PRINTER AANGEZET, INTRO UITGEVOERD," CR
  8 ." EN DE PRINTER WEER UITGEZET." CR
  9 ." DOOR DE OPDRACHT 'INTRO' TE GEVEN WORDT HET GEHELE" CR
 10 ."  EERDER KLAARGEMAAKTE PROGRAMMA UITGEVOERD. " CR CR CR  ;
 11 : INTRO CLS TEKST1 1 LIST CR TEKST2 TEKST3 TEKST4 TEKST5
 12    TEKST6 CR CR 6 5 4 3 2 LIST LIST LIST LIST LIST ;
 13 HEX 1 205 C!
 14 INTRO HEX 0 205 C!
 15 ;S
```

WE LØPEN DE LISTING REGEL VØØR REGEL DØØR.

BØVENAAN STAAT SCR# 1.
DAT IS DE PLEK WAAR IK BIJ HETMAKEN VAN DEZE TEKST
DE LISTING HEB ØPGEBØRGEN.
IN BASIC HØEF JE JE DAARØVER NIET TE BEKØMMEREN, ECHTER
IN FØRTH MØET JE GØED WETEN WAAR WAT STAAT IN 'T GEHEUGEN
EN BØVENAL HØE DAT ER STAAT !!

DE LISTING VAN DE SCR#1 (SCREEN NUMMER 1) HEEFT ALLES
TE MAKEN MET DE EDITØR.
MET DIE EDITØR (WAT EEN FØRTH-PRØGRAMMA IS), DIE IS
BESCHREVEN IN DE OSI-PØEL EE 39 T/M 42, BØEKJE 6
KUNNEN TEKSTEN IN HET GEHEUGEN WØRDEN GETYPT.

EERST DE WERKING VAN DE EDITØR.
SCHAF EERST EEN FØRTH MET EDITØR AAN (BANDJE 5)
DAARVØØR MØET HET SUPERBØARD MINIMAAL 16 K HEBBEN
NA HET INLADEN VAN DE FØRTH & EDITØR MELDT DE
KØMPUTER ZICH MET FIG-FØRTH ETC.

TYPE IN 'EMPTY-BUFFERS' RETURN EN 'EDITØR' RETURN
NU STAAN ALLE KØMMANDØ'S VAN DE EDITØR GEREED (ZIE ØØK
DE ØPSØMMING IN BØEKJE 7).
MAAK NU HET WERKGEBIED VØØR DE EDITØR SCHØØN MET 1 CLEAR
DAARMEE WØRDEN SPATIES IN SCREEN (=SCHERM) 1 GEZET.
DAT IS TE KØNTRØLEREN MET 1 LIST (REGELNRS EN SPATIES)
JE BENT NU IN SCHERM 1 WAARVAN DE REGELS IN TE VULLEN
ZIJN. DE PRØCEDURE IS ALS VØLGT :
TYPE EERST HET REGELNUMMER MET EEN SPATIE, DAARNA EEN
P <SPATIE> EN DE REGEL ZØALS VØØR FØRTH NØDIG IS
ALLE ANDERE KØMMANDØ'S VAN DE EDITØR KUNNEN WØRDEN GEBRUIKT
ØM REGELS ZØ AAN TE PASSEN ALS DAT NØDIG IS.

ZØDRA JE HET IDEE HEBT DAT DE REGELS DIE
IN DE SCREEN STAAN JUIST ZIJN, SLUIT JE DE LAATSTE REGEL
AF MET ';S' EN KUN JE HET SCHERM GAAN VERTALEN (CØMPILE-
REN). WANNEER ER FØUTEN WØRDEN GEMELD, GA JE TERUG NAAR
DE EDITØR EN VERBETER JE DE TEKST.
LET WEL DAT BIJ HET KØMPILEREN NIEUWE KØMMANDØ'S WØRDEN
TØEGEVØEGD AAN DE FØRTH-KERN (KØNTRØLEREN MET 'VLIST')
EVENTUELE ØVERBØDIGE FUNKTIES (SUBRØUTINES) ZIJN TE VER-
WIJDEREN DØØR 'FØRGET <FUNKTIE>'.

TERUG NAAR DE LISTING VAN SCR# 1.
HET IS EEN GØED GEBRUIK ØM ØP REGEL 0 DE TITEL VAN HET
SCHERM TE VERMELDEN. MET HET KØMMANDØ <EERSTE><LAATSTE>
'INDEX' (B.V. 1 4 INDEX) KUNNEN NAMELIJK DE 0 REGELS
VAN DE SCREENS WØRDEN BEKEKEN.
EEN HAAKJE ØPENEN <SPATIE> GEEFT AAN DAT ER KØM-
MENTAAR VØLGT, HAAKJE SLUITEN = EINDE KØMMENTAAR.
LET ER WEL ØP DE FØRTH-ØPDRACHTEN VAN ELKAAR TE SCHEIDEN
ANDERS KAN DE KØMPILER DE KØMMANDØ'S,VARIABELENNAMEN ETC
NIET UIT ELKAAR HØUDEN.

HET LADEN EN BEWAREN VAN FØRTH SCREENS

VØØR DE ØSI-BEZITTERS MET CASSETTE EN FØRTH IS EEN
EENVØUDIGE EN GØED WERKENDE METHØDE BESCHIKBAAR ØM
SCREENS, DIE MET DE EDITØR ZIJN GESCHREVEN VAN EN
NAAR BAND TE ZETTEN. ØMDAT IK AFVILDE VAN HET IEDERE
KEER WEER MØETEN ØPZØEKEN VAN HØE DAT PRECIES MØET
HEB IK EEN AANTAL FØRTH-KØMMANDØ'S GESCHREVEN, DIE
HET WERK VØØR MIJ DØEN.(ZIE ØØK FD7 T/M 11 ØSI-PØEL !)

HIER VØLGT NØG EEN KØRTE TØELICHTING ;
DE BELANGRIJKSTE ØPDRACHTEN ZIJN 'LAAD' EN 'BEWAAR'.
IN BEIDE WØRDT 'VRAAG' ØPGERØEPEN, WAARIN HET
AANTAL SCREENS EN DE STARTSCREEN WØRDT ØPGEVRAAGD.
'VRAAG' ZET DUS TWEE GETALLEN ØP DE STACK.
IEDER GETAL WØRDT MET CYFER? BINNENGEHAALD, DAT ØØK
DAT GETAL NØG WEERGEEFT ØP HET SCHERM.
ØMDAT DE FØRTH-ØPDRACHT 'KEY' (IN CYFER?)
DE ASCII-WAARDE VAN DE GEBRUKTE TØETS GEEFT, MØET
VØØRDAT WØRDT TERUGGESPRØNGEN DE ASCII-WAARDE VAN 0
WØRDEN AFGETRØKKEN (=30 HEX ØFWEL 48 DECIMAAL).
AAN HET EINDE VAN VRAAG WØRDT GEKØNTRØLEERD ØF DE
INGETØETSTE GETALLEN ZINNIG ZIJN . VØØR MIJN SIESTEEM
MET 24K RAM KAN IK MAXIMAAL 6 SCREENS HEBBEN, DUS TEST
IK ØF DE STARTSCREEN EN DE EINDSCREEN KLEINER ZIJN
DAN 7 EN GRØTER DAN 0.(IN I/Ø-FØUT?)
ØMDAT ER VIER TESTS ZIJN, WAARBIJ ALLEEN INTERESSANT
IS ØF ER EEN FØUT ØPTREEDT, HEB IK ØØK DAARVØØR EEN
ØPDRACHT 'FØUT?' GEMAAKT. BIJ EEN FØUT WØRDT DE WAARDE
VAN FØUT VERANDERD VAN EEN 1 IN EEN 0, WAARDØØR BIJ HET
EINDE VAN I/Ø-FØUT? GEEN CARRIAGE RETURN,MAAR EEN FØUT-
MELDING WØRDT GEGEVEN (NUMMER 24 HEX)
DE WERKING IS SIMPEL, NA HET VRAGEN VAN LAAD ØF
BEWAAR, WØRDEN DE TWEE VRAGEN GESTELD.
HET IS DAARBIJ ALLEEN MAAR ZAAK TE ZØRGEN DAT DE RECØRDER
LØØPT <VØØRDAT> DE TWEEDE VRAAG WØRDT BEANTWØØRD.
FØRTH BEGINT DAN (MITS ALLES GØED IS) ØNMIDDELIJK
TE LADEN ØF LØSSEN.

DE METHØDE ØM DE NU GEMAAKTE ØPDRACHTEN DEFINITIEF AAN
FØRTH TØE TE VØEGEN STAAT ØP PAG. EE 42 ØSI-PØEL, WAAR
DE EDITØR BESCHREVEN STAAT.
HØE DAT PRECIES GAAT STAAT ØP SCR# 6 HIERØNDER

BØVENDIEN STAAT EEN WERKFØRMULIER AFGEDRUKT, DAT IK IN
BYTE HEB GEVØNDEN EN SINDSDIEN VEEL GEBRUIK. DE METHØDE
SPREEKT VØØR ZICH, ADR=ADRES N EN M=GETAL C=ASCII KØDE

JØS BURGHØUTS
PUTHØF 35
5655 AM EINDHØVEN
TEL 040-528372

```
SCR # 1
  0 ( LADEN EN LØSSEN)
  1 HEX  1 VARIABLE FØUT
  2 : CYFER? KEY DUP EMIT CR 30 - ;
  3 : FØUT? IF ELSE 0 FØUT C! THEN ;
  4 : I/0-FØUT? DUP 0 > FØUT? 2DUP + 8 < FØUT?
  5   ØVER 7 < FØUT? ØVER 0 > FØUT?
  6   FØUT C@ IF CR ELSE 24 ERRØR THEN ;
  7 : VRAAG ." AANTAL SCREENS ? " CYFER?
  8   ." VANAF SCREENNUMMER ? " CYFER? I/0-FØUT? ;
  9 : BEWAAR VRAAG PRT WACHT ." G" WACHT
 10   400 * 4000 + SWAP 400 * TYPE WACHT STP ;
 11 : LAAD VRAAG 400 * 4000 + SWAP 4 * T> ." GELADEN" ;
 12
 13
 14
 15 ;S

SCR # 6
  0 ( HØE VØEG IK DEFINITIES AAN FØRTH TØE)
  1 ( VERKLAAR ZE EERST ALS ZIJNDE FØRTH-LID)
  2 FØRTH DEFINITIØNS DECIMAL
  3 ( HAAL DE SCREEN BINNEN)
  4 1 LØAD
  5 FØRTH DEFINITIØNS DECIMAL
  6 ( GA DE SIESTEEM-PØINTERS AANPASSEN)
  7 LATEST  12 +ØRIGIN !  ( TØP NFA)
  8 HERE    28 +ØRIGIN !  ( FENCE)
  9 HERE    30 +ØRIGIN !  ( DP)
 10 ' FØRTH 6 + 32 +ØRIGIN ! ( VØC-LINK)
 11 HERE FENCE !
 12 ( HET EINDE VAN FØRTH IS TE VINDEN DØØR)
 13 ( HEX HERE . WAARNA MET EEN CHECKSUM-SAVER HET HELE)
 14 ( GEHEUGENGEBIED VANAF $0280 TØT EINDE GESAVED KAN WØRDEN)
 15 ;S
```

AUG. 1983 (DATE) | (LOCATION) | CYFER? FOUT? I/O-FOUT? (WORD)

JOS (WRITER) | FORTH (VOCABULARY)

OSI POEL

TALEN FD 21

STACK						TOP	WORDS
Haal cijfer op en zet op scherm						-	:CYFER?
Wacht op toets indruk					-	C	KEY
				-	C	C	DUP
Geef weer					-	C	EMIT
Nieuwe regel					-	C	CR
Van ASCII naar cijfer				-	C	3Ø	3Ø
					-	n	-
							;
					-	F/T	:FOUT?
Indien true klaar						-	IF
Indien False						-	ELSE
een Ø toekennen					-	Ø	Ø
aan de variabele				-	Ø	ADR	FOUT
FOUT voor kontrole later						-	C!
						-	THEN
							;
n = aantal screens				-	n	m	:I/O-FOUT?
m = start screen			-	n	m	m	DUP
		-	n	m	m	Ø	Ø
Test of m > Ø			-	n	m	F/T	>
				-	n	m	FOUT?
		-	n	m	n	m	2DUP
Test of n + m < 8			-	n	m	SOM	+
		-	n	m	SOM	8	8
			-	n	m	F/T	<
				-	n	m	FOUT?
Test of n < 7			-	n	m	n	OVER
		-	n	m	n	7	7
			-	n	m	F/T	<
				-	n	m	FOUT?
Test of n > Ø			-	n	m	n	OVER
		-	n	m	n	Ø	Ø
			-	n	m	F/T	>
				-	n	m	FOUT?

AUG. 1983 (DATE) | (LOCATION) | verwolg I/O-FOUT? VRAAG BEWAAR (WORD)

Jos (WRITER) | FORTH (VOCABULARY)

OSI POEL | TALEN FD 22

STACK						TOP	WORDS
				–	n	m	ADR	FOUT
				–	n	m	1/ø	C @
Als FOUT 1 was					–	n	m	IF
CR en bekaar					–	n	m	CR
Als FOUT ø was					–	n	m	ELSE
ophouden met				–	n	m	24	24
onvoorwaardelijke	stop						–	ERROR
					–	n	m	THEN
								;
Haal 2 getallen op							–	:VRAAG
							–	." Aantal Scr "
						–	n	CYFER?
						–	n	." Vanaf... "
					–	n	m	CYFER?
Kontrole					–	n	m	I/O-FOUT?
								;
Zet naar kassette							–	: BEWAAR
					–	n	m	VRAAG
Zet save-vlag aan					–	n	m	PRT
Wacht 3 seconden					–	n	m	WACHT
Begin gegevensblok					–	n	m	." G "
					–	n	m	WACHT
				–	n	m	4øø	4øø
					–	n	m1	*
4øøø = mijn startadres				–	n	m1	4øøø	4øøø
SCR #ø	m2 = startadres			–	n	m2		+
gewenste scrreen					–	m2	n	SWAP
				–	m2	n	4øø	4øø
n1 = aantal te					–	m2	n1	*
saven bytes							–	TYPE
							–	WACHT
Zet save-vlag uit							–	STP
								;

OKTOBER 1983

OSI POEL

STACK							TOP	WORDS
Haal van kassette							–	:LAAD
					–	n	m	VRAAG
				–	n	m	4ØØ	4ØØ
					–	n	m2	*
				–	n	m1	4ØØØ	4ØØØ
m2 = startadres n/d					–	n	m2	+
te laden scrn					–	m2	n	SWAP
				–	m2	n	4	4
n1 = aantal te					–	m2	n1	*
laden pages.							–	T>
							–	." Geladen"
								;

DATE _____ LOCATION _____ WORD _____

WRITER _____ VOCABULARY _____

STACK TOP WORDS