

# OSI Gebruikersgroep

## deel 11

NOV. 1984

INHOUDSOPGAVE UIT PEEK-65  
SOLDEREN  
REALTIME KLOK/KALENDER VOOR 6502 V1.0  
WIJZIGING IN 'ZERO' EPROM PROGRAMMER  
AANPASSINGEN IN DE HENK-WEVERS-MONITOR  
BASIC SCHAAK VOOR SUPERBOARD  
ROSI HIGH RESOLUTION GRAPHICS  
6502 TRACER  
11 FORTH SCREENS  
SABR. VOOR SUPERBOARD  
'TOETS' EN 'WINDOW' ONDER OS65-D V3.3  
LISTER VAN BASIC-PROGRAMMA VARIABELEN

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

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

Copyright HCC.

## INHOUDSOPGAVE VAN PEEK-65

\*\*\*\*\* oktober 1983 \*\*\*\*\*

TEC65 review; bespreking van TECO, een commercieel tekstverwerkingsprogramma.  
Azimuth-reading; basic-programma om plaatsen in de sterrenhemel te berekenen.  
OS-65U smart-terminal program; voor disc-gebruikers, maar niet zo smart als in de OSI-POEL.  
Systems-generator v83.1 review; bespreking van een disc-databasemanager pakket (12 discs !!)  
Software for OSI; zeer uitgebreid overzicht van commerciële software voor allerlei OSI-machines  
Optimising OSI basic programs II; programmeertruuks.  
Cleaning up the clock; verbeter de phi-2 klok.  
Heeft mij zeer geholpen.

\*\*\*\*\* november 1983 \*\*\*\*\*

EPR0M-programmer for SB-II; hardware zonder programma voor de 2708 en 2716 EPROM's.  
Relocating WF-6502, part 1; gebruik van een tekstverwerkingsprogramma met nieuwe monitor-EPROM  
Printed report check; basic-disc programma ter hulp bij het verwerken van veel tekst.  
Machine language directory; disc-utility voor een snellere en efficiënte weergave directory.  
Software for OSI; zie oktober 1983.  
Programming tips for KEYBASIC; OSI's KEYBASIC toegelicht.

\*\*\*\*\* december 1983 \*\*\*\*\*

Wordprocessor for OSI; 8K basicprogramma (disk+tape). voor tekstverwerking.  
The 2716; Uitleg over de hardware v/d 2716 EPROM.  
Relocating WF-6502; zie november 1983.  
Hooks into BASIC V 1.8; machinetaaluitbreiding met extra commando's voor V1.8 disc-basic.  
Time & task-planner, a review; bespreking van een zogenaamd 'agenda-programma'.  
INHOUDSOPGAVE 1983.

\*\*\*\*\* januari 1984 \*\*\*\*\*

Relocating WF-6502; deel 3, zie november 1983  
DBPack review; bespreking van een database-programma dat vergelijkbaar is met DBASE II.  
Hooks into Basic V 1.8; vervolg op december 1983.  
Expansion for 6502 computers; uitbreiding met een moederbord met 4 stekers, hardware.  
Expanding the C1F; vervolg op augustus 1983, met 8K RAM uitbreiding.  
OSI ROM routines; beschrijving van de floppy-disk boot code (\$FF00-FFFF) v/d SYNMON ROM.

\*\*\*\*\* februari 1984 \*\*\*\*\*

What else for OSI-basic ?; toevoegen van een :ELSE  
commando (\$BEF5-BF2B) in basic-4.Frima,  
als nieuwe standaard-routine. WAIT is wel niet  
meer te gebruiken. (als wait).

XREF: Basic program cross reference generator; een  
disk-machinetaal programma om alle verwijzingen  
en variabelen uitgelist te krijgen.

Parallele printer interface for SB II; hardware met  
output+input logica en een 2708 EPROM om ev.  
een printerprogr. in op te slaan.

32->64 karakter video ombouw, snij-instructies en  
een schema.

Extended directory listing; disk-basic programma met  
een directory met datum/tijd/pagina etc.

\*\*\*\*\* maart 1984 \*\*\*\*\*

Old for OSI Basic in Rom; machinetaal routine om na  
NEW toch het oude progr. terug te krijgen.  
Ligt in Basic-4 van \$BEA4-BEF4.

Installing a non-OSI single-sided 40 track disk drive  
on the C1P; hardware voor Siemens EDD 100-5B.

Expanding the C1P; vervolg hardware voorgaande maanden  
XREF-cross-reference; vervolg februari 1984.

met vervolg v/h machinetaalprogramma.

32->64 karakter-ombouw; software voor de hardware  
ombouw van februari 1984.

New challenger personal computer; de ideale structuur  
voor de volgende OSI-machine.??!!

\*\*\*\*\* april 1984 \*\*\*\*\*

Tax preparation; disk-basic progr.voor US-fiscus.

Expanding the C1P; vervolg met wederom een moederbord

Diskdoubler; software-review voor dubbel-densitydisk

Screen to printer dump; machinetaalprogramma, dat  
ook eenvoudig in basic werkt.

Delete-all; disk utility voor OS65D in basic.

Memory-map for HEXDOS 4.0; van Steve Hendrix

Adapted Modem routine; verbetering op de Modem-rou-  
tine voorstellen van OSI. Machinetaal+basic.

SOLDEREN..... DAT IS MET EEN SOLDEERBOUT EVEN EEN DRUPJE SOLDEER OP EEN DRAADJE LEGGEN.....

NOU VERGEET DAT MAAR, VOORAL IN ONZE HOBBY IS SOLDEREN EEN HEEL PRECIJS KLUSJE. ER KOMEN NOGAL WAT FACTOREN OM DE HOEK KIJKEN, DIE DE KWALITEIT VAN EEN SOLDEERVERBINDING BEINVLOEDEN.

WE HEBBEN ZE HIER VOOR U MAAR EENS OP EEN RIJTJE GEZET.  
SOLDEERTIN

De samenstelling van het soldeer bepaalt de kwaliteit. Hoofdbestanddelen zijn tin en lood, verder zitten er nog wat restelementen in. Tin is verreweg het duurste element in de alliage, vooral omdat het z.g. maagdelijk tin moet zijn. Tin dat door recycling d.i. door afsmelting van oude materialen is verkregen, bevat teveel verontreiniging van andere metalen die, als we het zouden gebruiken, een broze structuur aan de verbinding zouden geven.

Er is veel soldeer waarop vermerkt staat (als het er al op staat) 40/60. Dit betekent 40% tin en 60% lood en het heeft een smelttemperatuur van 234°C, veel te hoog voor onze gevoelige componenten. Dit soldeer vindt U in de hobby en doe het zelf zaken, op een kaartje gewikkeld.

Het is prima geschikt om met een pittige vuurbout wat blikwerk in elkaar te solderen, maar voor onze doeleinden totaal ongeschikt.

Goed soldeer is b.v:

Ersin multicore - rood (60/40) 1.2 mm dik

Fluitin Sn 60 1mm dik

Het zijn harskernsoldeer soorten d.w.z. in de kern zit een harsachtige massa die bij verhitting een reinigende (z.g. reducerende) werking heeft. De smelttemperatuur van dit soldeer is 188°C. Het is te verkrijgen in de componenten handel en een rol van 500gr kost U de bodem uit de portemonnaie maar je hebt er lang plezier van.

SOLDEERBOUT

Bij de keuze van een soldeerbout zullen we in dit artikel ervan uitgaan dat we vrij regelmatig aan printen zullen solderen. Voor een enkele soldeerplaats kan het gewone 25W 220V boutje wel dienst doen. Als echter b.v. op een print, tientallen soldeerverbindingen moeten worden gemaakt, stellen we wel hogere eisen.

Een goede bout moet warmte toevoeren zoveel als nodig is voor een goede doorgevoerde verbinding, maar mag niet te heet worden als hij tussentijds even niet gebruikt wordt.

Een temperatuurgeregelde bout is een noodzaak om goed werk te maken. Een eenvoudig Weller stationnetje (Magnastat) met verwisselbare stiften voldoet al aan de eisen. Niet alleen kunnen wij de vorm van de stift aanpassen aan het soort werk, ook de temperatuur is door gebruik

van verschillend genummerde stiften, te regelen.

Een lang smal stiftje is fijn om op moeilijk bereikbare plaatsen te komen, een korte dikke stift brengt op zwaardere verbindingen de warmte wat direkter over. Een stift nr.7 is voor het normale werk, terwijl een nr.8 b.v. bij het uitsolderen goede diensten bewijst.

#### WERKVOORBEREIDEN

##### A- Printjes

Een van de leukste verrassingen die met printjes ondervonden wordt is de onderbroken verbinding. Dit euvel treedt in allerlei vormen op: haarscheurtje in een geleider - niet goed doorgemetalliseerd gat bij een tweezijdige print - allerlei aardige maniertjes om een mens grijze haren te bezorgen. Als we dan voortvarend de print al bestukt en gesoldeerd hebben, wordt het nog moeilijker om zo'n fout te vinden. Daarom, voor er iets met de print gedaan wordt: maak een kopie van òf de print layout òf van het schema, neem de universeel meter en meet met de Ohmmeter alle banen door, onderwijl met een viltstift elke gecontroleerde verbinding op de kopie aftekenend.

Een rotbaantje, ja, misschien zelfs volkomen onnodig, maar als er .... Voor de hobbyist zit er bovendien aan zo'n karwei toch wel een pluspuntje. De layout volgend zie je hoe de kaart in elkaar zit en krijg je tevens inzicht in de kronkelende geest van de printontwerper. Eventuele regelpunten zoals instelpots worden ook wat vertrouwd.

##### B- Componenten

Op printen te bevestigen onderdelen zijn meestal op een raster van 0.1 inch geplaatst. Gaten voor  $\frac{1}{4}$  watt weerstanden liggen veelal 0.4 in. uitelkaar en die voor ontkoppelcondensatoren op 0.3 in. Door nu een buigblokje te gebruiken, dat voor een luttel bedrag in de electronica handel te verkrijgen is, kunnen we vooral weerstanden en dioden op de juiste afstand buigen.

Zorg ervoor dat de waardeaanduiding die op sommige onderdelen gedrukt is, naderhand te zien zal zijn.

Bij het bestukken van de print nemen we eerst de laagste onderdelen, dit zullen meestal de weerstanden en dioden zijn. We steken deze door de print, leggen op de onderdelen een stukje schuimrubber, draaien het hele zwikje om en kunnen nu, zonder dat de onderdelen eruit zakken, de lange doorstekende draadeinden gaan behandelen.

We buigen deze draadjes nu in de richting van de van het gat weglappende geleider (zie fig.1) en laten ze schuin omhoog staan. Hierdoor zal een eventuele vervanging minder kracht kosten bij het verwijderen hetgeen de print tegen beschadiging spaart.

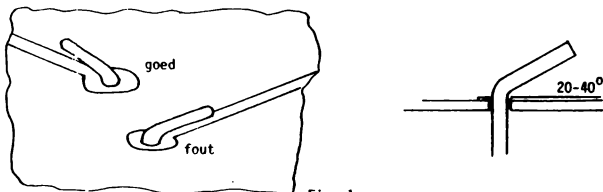


Fig. 1

Het inkorten van de draadeinden doen we met een scherpe tang. Er mag nooit gewrikt worden om een draad tot breken te brengen. De componenten zijn tegen dergelijke mechanische krachten niet bestand.

De stand van onze kniptang is ook nog kritisch. Het sprong-moment dat bij het knippen ontstaat, moet naar het af te knippen draadeind gericht zijn, niet naar het component. (zie fig.2)

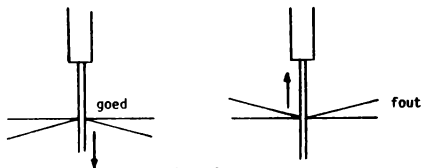


Fig. 2

#### HET SOLDEERWERK

Met het solderen brengen we een verbinding tot stand tussen twee geleiders. Aan die verbinding moten we eisen stellen, n.l. dat:

- A er een maximum aan geleiding ontstaat
- B de mechanische sterkte zo groot mogelijk is
- C er geen aantasting plaatsvindt door chemicalien

De beide eerste eisen kunnen we handwerkelijk beïnvloeden, de laatste (C) kan alleen door keuze van het soldeertin worden beïnvloed en aan dit materiaal hebben we reeds voldoende aandacht besteed.

We weten dat een dunne draad meer weerstand biedt aan de elektrische stroom dan een dikke. Zo ook bij het solderen, karig en slecht gevloeid soldeer kan voor een grotere overgangsweerstand zorgen en bovendien is zo'n verbinding niet sterk.

Het vloeien van het soldeer - en daardoor het zich innig verbinden van soldeer en koper - kan alleen maar plaats vinden als het koper de juiste temperatuur bereikt heeft. Bij het leggen van een verbinding moeten dan ook beide geleiders die temperatuur bereiken hebben.

De meeste fouten bij het solderen ontstaan doordat met de soldeerstift het uit de print omhoogstekende draadeinde of IC pootje wordt verhit, daar een druppel soldeer aan wordt gehangen en die druppel maar zelf moet zien hoe hij het rooit met die onderliggende printgeleider. Het

gevolg is dat het kleine beetje vloeimiddel dat op de print is terecht gekomen, niet die temperatuur bereikt waarbij het zijn reducerende werk kan doen. Het dunne koperoxyd laagje dat op de geleider altijd aanwezig is (en dat een veel hogere weerstand heeft als koper), wordt niet door het vloeimiddel "weggewerkt" en dus zakt het soldeertin op dit oxydlaagje neer, soms ook plakt het enkel maar met wat hard geworden vloeimiddel vast.

Resultaat: grote narigheid.

De juiste manier is dan ook: eerst warmtetoevoer aan de dikste geleider, en dat is tien tegen een de printbaan. Het helpt natuurlijk als we die warmteoverdracht versnellen door wat tin aan te brengen tussen soldeerstift en geleider. Als nu door het uitvloeien van dit tin over de oppervlakte rond de soldeerplaats (het "oogje") blijkt dat hier de soldeertemperatuur is bereikt, laten we nog even de stift tegen het draadje of pootje aanleunen, voegen zo nodig nog iets tin bij, en reeds "springt" het tin ook tegen deze geleider op en vormt een mooi hol staand poeltje tin.

Bij een enkelzijdige print is nu de kous af, door het hol staande poeltje kunnen we zien dat de soldeerplaats gevloeid is en er een goede elektrische verbinding is gemaakt.

Tweezijdige prints hebben echter doorgemetalliseerde gaten die ook moeten volvloeien met tin. U zult zien dat in dat geval na het eerste vloeien van het tin tussen beide geleiders, na nog even verder verhitten, het tin plotseling wegzakt en wij dus nog een weinig tin moeten bijvoegen om weer aan het ideale "holle" vloeibeeld te komen.

Zo met dit betoog is het onderwerp solderen zeker niet uitputtend behandeld. Het doel van dit stukje was dan ook om wat inzicht te verschaffen in materialen en technieken die bij goed solderen een rol spelen, met als gevolg: minder fouten bij het opbouwen van een print schakeling

Rein Heesterman



\*\*\*\* REALTIME KLOK/KALENDER VOOR 6502 V1.0 \*\*\*\*  
 \*\*\*\* Kees Hoogeland 1 Sept.1984 \*\*\*\*

Ondanks de eenvoudige schakeling van de klokkaart zijn toch alle mogelijkheden aanwezig die men zich wensen kan n.l.

- :Altijd standby door batterijvoeding en X-tal oscillator
- :Resisters voor tienden van seconden tot maanden
- :Automatische schrikkeljaar correctie
- :Onafhankelijke interrupt mogelijkheden van 0,5 tot 60 seconden
- :Leesbeveiliging gedurende het veranderen van de data
- :Eigen adresgebied inclusief eeprom en bufferram
- :Buffers voor adres- en datalijnen.

De zendamateurs zullen een klok wel waarderen voor het elektronisch logboek en de treinbaan-bestuurders kunnen de vertrektijden nu automatisch bewaken. Het operating-system "Newdos" van Ringersma zal dankbaar gebruik kunnen maken van deze klok om de datum en tijdsgevens toe te voegen in de directory, bij de aanmaak en wijzigingen van files. In de eeprom is voldoende ruimte om dit te realiseren en het ram buffer wordt al gevuld met TIME\$ en DATE\$.

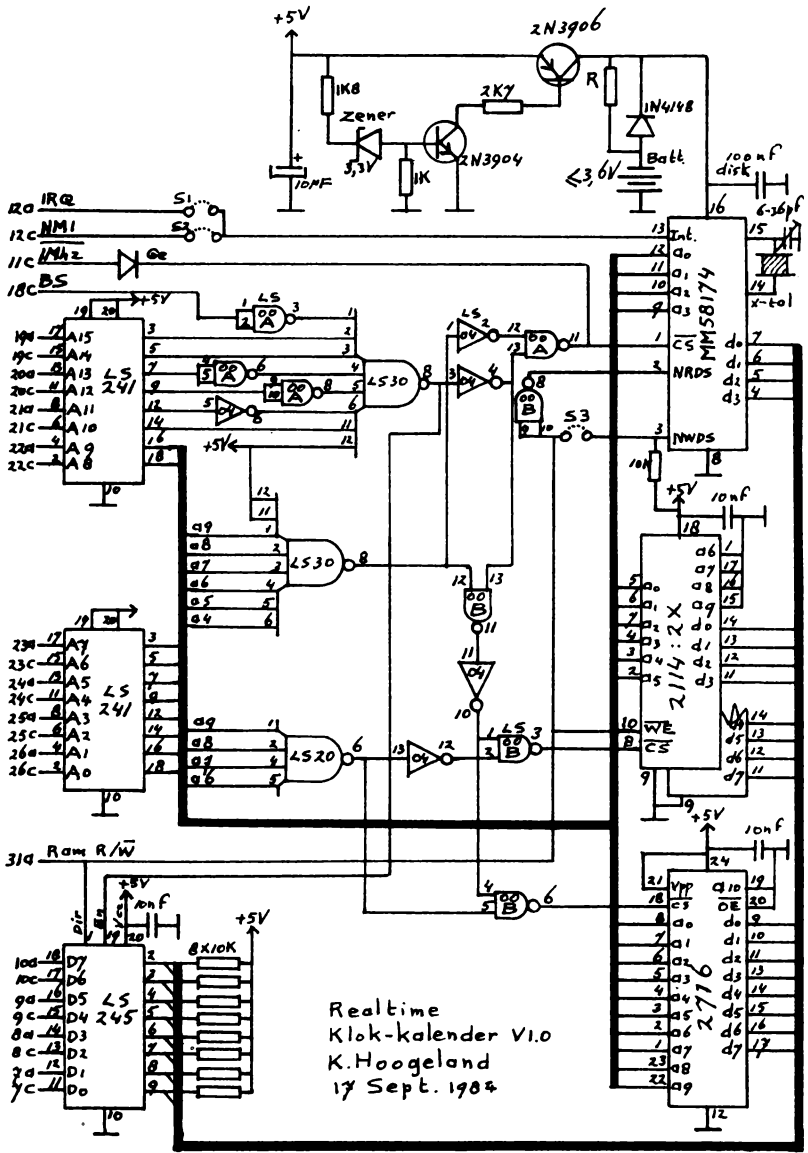
Het schema

De bovenste 74LS30 selecteert het adresgebied voor het gehele klok-gebeuren. Dit loopt van \$C400 TOT \$C800. Met de overige poorten (en inverters) wordt dit gebied als volgt uitgesplitst:

- 2716 Eprom van \$C400 t/m \$C7BF
- 2114 buffergeheugen van \$C7C0 t/m \$C7EF
- MM58174 van \$C7F0 t/m \$C7FF

De schakeling rondom de batterij verzorgt het automatisch overschakelen van net- naar batterijbedrijf. Als voedingsbron wordt een lithium-batterij gebruikt die 5 jaar meesaat. Ook kan een nicad-batterij worden gebruikt. In dat geval moet de weerstand R worden gemonteerd. De batterijspanning niet hoger kiezen als 3,6 volt.! Wordt "Bank-select" niet gebruikt dan pin 18c aan 0 volt. Draait de processor altijd op 1 Mc. dan kan de germanium diode vervallen. Is het signaal "Ram R/W" niet aanwezig, dan kan dit met een NAND worden samengesteld m.b.v. de signalen 02 en R/W-geinvertteerd. Het is aan te bevelen om na het gelijkzetten van de klok deze met S3 in de stand "No-write" te zetten zodat niet meer per abuis naar de klok kan worden geschreven.

Klok-IC : MM58174 fa. Rodelco  
 X-tal : 32768Hz (parallelresonantie)  
 b.v. Statek CX-1V fa. Tekelec  
 of ETA CX-LV fa. Heynen



## De programma's

Er worden alleen subroutines uit de monitor gebruikt. Daarom zijn de programma's makkelijk aan te passen aan andere computers. Het programma is zoveel mogelijk van commentaar voorzien en spreekt dus bijna geheel voor zichzelf. Alleen bij het opstarten en het gelijkzetten van de klok (via \$C406 of \$C409) worden wat handelingen verricht die voorgescreven zijn door de fabrikant om de chip te initialiseren. Dit gebeurt in "PWRUP", in regel 1010 e.v. In de subroutine "UPDATE" regel 1130 is te zien dat de registers van de klokchip worden overgebracht naar register "TIME" in de 2114's. De tienden van seconden worden, zowel aan het begin als aan het einde van de klok-lees routine gelezen en op "data-changed" sechecked. Dit is allemaal nodig om te voorkomen dat er een tijd wordt verwerkt tijdens het veranderen van de klok.

De subroutines vanaf regel 1800 zijn vooral bedoeld als voorbeeld en naar eigen wensen te veranderen.

```

10          ; ***** KLOK MM 58174 *****
11          ; ***** KEES HOOGELAND *****
13          ; ***** (C) 11 SEPT '84 ****
14          ;
15          TIME   =%C200      PIA-ram, voor 2114ram+%C7C0
16          RESREG=%F885      RESTORE A,X,Y
17          MDN    =%FF00
18          OUTCH  =%FF69      ALGEMENE OUTPUTROUTINE
19          KARIN  =%FD00      KEYBOARD
20          HEXASC=%FE93      TEST HEX & BINARIZE (0-9)
21          ROW    =%0229      CURSOR X
22          LINE   =%RW+1      CURSOR Y
23          DISPL  =%D7D0      ONDERSTE REGEL SCHERM
24          D1     =DISPL/256*256
25          POS    =DISPL-D1-192
26          RDFLAG=%C2D5      PIA-ADRESSEN
27          CLK1KT=%C2F7      VOOR DEMO VERTRAGING
180         ;
190         ;***** TIME-LOCATIES *****
200         ;
210         C200          *           =TIME
220         C201          SEK.1      **++1
230         C202          SEK        **++1
240         C203          SEK10      **++1
250         C204          MIN        **++1
260         C205          MIN10     **++1
270         C206          UUR        **++1
280         C207          UUR10     **++1
290         C208          DAG        **++1
300         C209          DAG10     **++1
310         C20A          DAGNR     **++1
320         C20B          MND        **++1
330         C215          MND10     **++10
340         C21F          TIME$     **++10
350         DATE$
360         ;

```

```

370                                     ;***** KLOK-REGISTERS *****
380                                     ;
390 C7F0                                *      =%C7F0
400 C7F1                                TESTM  ****+1
410 C7F4                                SK. 1  ****+3
420 C7FD                                MN     ****+9
430 C7FE                                JAARNR ****+1
440 C7FF                                STOPST ****+1
450                                     INTSTA
460                                     ;
470                                     ;
480 C400                                *      =%C400
490                                     ;
500 C400 4C21C7                         JMP DEMO
510 C403 4C1AC7                         JMP DATTYD
520 C406 4C3BC7                         JMP ALLIN
530 C409 4C12C4                         JMP SETKL
540 C40C 4CF4C4                         JMP PWRUP
550 C40F 4CFEC6                         JMP NMIVEC
560                                     ;
570                                     ;***** GELIJKZETTEN KLOK *****
580                                     ;
590 C412 20F4C4                         SETKL JSR PWRUP          INITIALISEER CHIP
600 C415 A900                            LDA #0
610 C417 8DFEC7                         STA STOPST          STOP KLOK
620 C41A 8D2902                         STA ROW            PRINT0(9,0)
630 C41D A909                            LDA #9
640 C41F 8D2A02                         STA LINE
650                                     ;
660 C422 A0FF                            LDY #FF           KARAKTERTELLER
670 C424 CB                               SK1  INY
680 C425 B949C4                         LDA TEKST, Y
690 C428 F006                            BEQ VULREG         0 DELIMITER GEVONDEN
700 C42A 2069FF                         JSR OUTCH
710 C42D 4C24C4                         JMP SK1
720                                     ;
730 C430 A209                            VULREG LDX #9      (ACC=0 DOOR DELIMITER)
740 C432 2000FD                         GETSET JSR KARIN    HAAL 10 DIGITS VAN KEYB.
750 C435 2069FF                         JSR OUTCH         DRUK AF
760 C438 2093FE                         JSR HEXASC        TEST HEX & BINARIZE ( 0-9
770 C43B 30F5                            BMI GETSET        GELDIG ?
780 C43D 9DF4C7                         STA MN, X         BRENG IN KLOKREGISTERS
790 C440 CA                               DEX
800 C441 10EF                             BPL GETSET
810 C443 A901                            LDA #1
820 C445 8DFEC7                         STA STOPST        START KLOK
830 C448 60                               RTS
840                                     ;
850 C449 0A                               TEKST .BYTE %0A, %0D, %0A, %0D
860 C44A 0D
870 C44B 0A
880 .BYTE 'VOOR SCHRIKKELJAAR RESP. +1, +2, +3, '
890 .BYTE %0A, %0D
900 .BYTE ' JAARNR. INVULLEN: 8 RESP. 1, 2, 4, '
910 .BYTE %0A, %0D
920 .BYTE 'VOOR DAGNR. GELDT: ZONDAG=1'
930 .BYTE %0A, %0D
940 .BYTE 'JAARNR/MND/DAGNR/'
950 .BYTE 'DATUM/UUR/MIN', %0A, %0D
960 .BYTE %0A, %0D, 'FORMAT: J/MA/D/DA/UU/MI'
970 .BYTE %0A, %0D
980 .BYTE ' ', 0
990 ;

```

```

990 ;***** POWER-UP/INITIALIZE *****
1000 ;
1010 C4F4 A900 PWRUP LDA #0 POWER-UP
1020 C4F6 8DF0C7 STA TESTM DISABLE TESTMODE
1030 C4F9 8DFFC7 STA INTSTA DISABLE INTERRUPT
1040 C4FC ADFFC7 LDA INTSTA PIN 13 OHMOOG
1050 C4FF ADFFC7 LDA INTSTA
1060 C502 ADFFC7 LDA INTSTA
1070 C505 A901 LDA #1
1080 C507 8DFEC7 STA STDPST START KLOK
1090 C50A A909 LDA #09
1100 C50C 8DFFC7 STA INTSTA SSEC.TIMER+REPT.INTERRUPT
1110 C50F 60 RTS
1120 ;
1130 ;***** UPDATE TIME-REGISTER *****
1140 ;
1150 C510 ADF1C7 UPDATE LDA SK.1
1160 C513 ADF1C7 LDA SK.1
1170 C516 ADF1C7 LDA SK.1
1180 C519 209DC5 JSR CHECK CHECK PLAUSIBILITY
1190 C51C A20C LDX #12 MOVE 12 LOCATIES
1200 C51E 8DF0C7 TT1 LDA SK.1-1,X VAN TUSSENGEHEUGEN
1210 C521 209DC5 JSR CHECK
1220 C524 290F AND #0F CONVERTEER NAAR ASCII
1230 C526 0930 ORA #30
1240 C528 9DFFC1 STA TIME-1,X NAAR TIMEGEHEUGEN
1250 C52B CA DEX VERLAAG 12 TELLER
1260 C52C D0F0 BNE TT1 TELLER OP 0 ?
1270 C52E ADF1C7 LDA SK.1 VOOR RECHECK
1280 C531 209DC5 JSR CHECK
1290 C534 2038C5 JSR STRING TIME- EN DATESTRING VORMEN
1300 C537 60 RTS
1310 ;
1320 ;***** MAAK TYD- EN DATUMSTRING *****
1330 ;
1340 C538 8E1DC2 STRING STX TIME#+8 X=0! DELIMITER VOOR BV.STROUT
1350 C53B 8E27C2 STX DATE#+8 IDEM
1360 C53E A9D0 LDA #D0
1370 C540 8D1EC2 STA TIME#+9 DELIMITER VOOR BV.SCR
1380 C543 8D28C2 STA DATE#+9
1390 C546 A93A LDA #'
1400 C548 8D17C2 STA TIME#+2
1410 C54B 8D1AC2 STA TIME#+5
1420 C54E A92F LDA #' /
1430 C550 8D21C2 STA DATE#+2
1440 C553 8D24C2 STA DATE#+5
1450 C556 AD06C2 LDA UUR10
1460 C559 8D15C2 STA TIME$
1470 C55C AD05C2 LDA UUR
1480 C55F 8D16C2 STA TIME#+1
1490 C562 AD04C2 LDA MIN10
1500 C565 8D18C2 STA TIME#+3
1510 C568 AD03C2 LDA MIN
1520 C56B 8D19C2 STA TIME#+4
1530 C56E AD02C2 LDA SEK10
1540 C571 8D1BC2 STA TIME#+6
1550 C574 AD01C2 LDA SEK
1560 C577 8D1CC2 STA TIME#+7
1570 C57A AD08C2 LDA DAG10
1580 C57D 8D1FC2 STA DATE$
1590 C580 AD07C2 LDA DAG
1600 C583 8D20C2 STA DATE#+1
1610 C586 AD0BC2 LDA MND10
1620 C589 8D22C2 STA DATE#+3
1630 C58C AD0AC2 LDA MND
1640 C58F 8D23C2 STA DATE#+4
X 1650 C592 A908 LDA #8 → moet zijn A938 LDA #38
1660 C594 8D25C2 STA DATE#+6
X 1670 C597 A904 LDA #4 → A934 LDA #34
1680 C599 8D26C2 STA DATE#+7 IN 1985 AANPASSEN IN BV.BEXEC*
1690 C59C 60 RTS
1700 ;

```

```

1710                                     ;***** CHECK FOR DATA-, CHANGED *****
1720                                     ;
1730 C59D C9FF CHECK CMP #FF BIJ DATA-CHANGED:FF OP DATABUS
1740 C59F F001 BEQ OUT DUS OPNIEUW UPDATEN
1750 C5A1 60 RTS
1760 C5A2 68 OUT PLA POP RETURNADRES
1770 C5A3 68 PLA
1780 C5A4 4C10C5 JMP UPDATE
1790                                     ;
1800                                     ;***** TYD NAAR SCHEM *****
1810                                     ;
1820 C5A7 2010C5 TISCR JSR UPDATE EERST BIJWERKEN
1830 C5AA A200 LDX #0 POSITIE TELLER
1840 C5AC A005 LDY #5 REGISTER TELLER
1850                                     ;
1860 C5AE B901C2 HAAL LDA SEK,Y 1 REGISTER NAAR SCHEM
1870 C5B1 9DE9D7 STA DISPL+19,X
1880                                     ;
1890 C5B4 E8 INX VERHOOG POSITIE TELLER
1900 C5B5 88 DEY REGISTERTELLER 1 LAGER
1910 C5B6 B901C2 LDA SEK,Y 1 REGISTER NAAR SCHEM
1920 C5B9 9DE9D7 STA DISPL+19,X
1930 C5BC E8 INX VERHOOG POSITIETELLER
1940                                     ;
1950 C5BD A93A LDA #1 :> NAAR SCHEM
1960 C5BF 9DE9D7 STA DISPL+19,X
1970 C5C2 E8 INX VERHOOG POSITIETELLER
1980 C5C3 88 DEY REGISTERTELLER 1 LAGER
1990 C5C4 10E8 BPL HAAL NOG MEER?, DAN HAAL
2000 C5C6 60 RTS
2010                                     ;
2020                                     ;***** DATUM NAAR SCHEM *****
2030                                     ;
2040 C5C7 AD2902 DASCR LDA ROW SAVE CURSORPOSITIE
2050 C5CA 48 PHA
2060 C5CB AD2A02 LDA LINE
2070 C5CE 48 PHA
2080                                     ;
2090 C5CF 2010C5 JSR UPDATE EERST UPDATEN
2100                                     ;
2110 C5D2 A212 LDX #POS+2 CURSOR OP DAG# POSITIE
2120 C5D4 BE2902 STX ROW
2130 C5D7 A21F LDX #31
2140 C5D9 BE2A02 STX LINE
2150                                     ;
2160 C5DC AD09C2 LDA DAGNR DAG#NR. BEPALEN EN
2170 C5DF 290F AND #0F
2180 C5E1 201FC6 JSR SCR VAN TABEL NAAR SCHEM
2190                                     ;
2200 C5E4 AD08C2 LDA DAG10 DATUM OPHALEN
2210 C5E7 BDDCD7 STA DISPL+12 NAAR SCHEM
2220 C5EA AD07C2 LDA DAG IDEM
2230 C5ED BDDDD7 STA DISPL+13
2240                                     ;
2250 C5F0 A920 LDA #20 SPATIE NAAR SCHEM
2260 C5F2 BDDDED7 STA DISPL+14
2270                                     ;
2280 C5F5 A21F LDX #POS+15 CURSOR OP MND# POSITIE
2290 C5F7 BE2902 STX ROW
2300                                     ;
2310 C5FA AD0BC2 LDA MND10 (< OF 1?)
2320 C5FD 290F AND #0F
2330 C5FF F00A BEQ MND1 MND10=0?: DAN NAAR MND1
2340 C601 AD0AC2 LDA MND
2350 C604 290F AND #0F
2360 C606 18 CLC
2370 C607 690A ADC #10 MND10 WAS IMMERS 1!
2380 C609 D006 BNE MND2 BRANCH ALWAYS
2390                                     ;

```

```

2400 C50B A00AC2 MND1 LDA MND
2410 C60E 290F AND #50F
2420 C610 18 CLC
2430 ;
2440 C611 6907 MND2 ADC #507          MAAKT $NR. VAN MND.NR.
2450 C613 201FC6 JSR SCR
2460 ;
2470 C616 68 PLA          RESTORE CURSOR-POSITIE
2480 C617 8D2A02 STA LINE
2490 C61A 68 PLA
2500 C61B 8D2902 STA ROW
2510 ;
2520 C61E 60 RTS
2530 ;
2540 ;***** PRINT MSG. :NR. IN ACC *****
2550 ;
2560 C61F AA SCR TAX
2570 C620 CA DEX          STRINGNR. MOET 1 LAGER
2580 C621 A0FF LDY #5FF          SET Y VOOR DATA MSG'S
2590 C623 CA DK DEX          STRINGNR. AFTELLEN
2600 C624 3008 HMI DM          NEG. DAN GOEDE MSG
2610 C626 C8 DL INY          ZOEK
2620 C627 B940C6 LDA TABEL, Y
2630 C62A 10FA HPL DL
2640 C62C 30F5 BMI DK
2650 C62E C8 DM INY          GEVONDEN
2660 C62F B940C6 LDA TABEL, Y
2670 C632 3006 RMI DN          NEEM LETTER
2680 C634 2069FF JSR OUTCH          CHECK BIT 7 VOOR EINDE WOORD
2690 C637 4C2EC6 JMP DM          ADRUKKEN
2700 C63A 297F DN AND #57F          LAATSTE ASCII MAKEN
2710 C63C 2069FF JSR OUTCH          AFDrukKEN
2720 C63F 60 RTS
2730 ;
2740 ;***** MESSAGE-TABLE *****
2750 ;
2760 C640 20 TABEL .BYTE ' ZONDAG', $A0, ' MAANDAG', $A0
2770 TABEL .BYTE ' DINSdag', $A0, ' WOENSDAG', $A0
2780 .BYTE ' DONDERDAG', $A0, ' VRIJDAG', $A0
2790 .BYTE ' ZATERDAG', $A0
2800 ;
2810 .BYTE ' JANUARI ', $A0, ' FEBRUARI ', $A0
2820 .BYTE ' MAART ', $A0, ' APRIL ', $A0
2830 .BYTE ' MEI ', $A0, ' JUNI ', $A0
2840 .BYTE ' JULI ', $A0, ' AUGUSTUS ', $A0
2850 .BYTE ' SEPTEMBER', $A0, ' OKTOBER ', $A0
2860 .BYTE ' NOVEMBER ', $A0, ' DECEMBER ', $A0
2870 ;
2880 ;
2890 ;***** SET NMI-VECTOR *****
2900 ;
2910 C6FE A94C NMIVED LDA #54C
2920 C700= INT1=KLKINT/256*256
2930 C700 A20E LDX #KLKINT-INT1
2940 C702 A0C7 LDY #KLKINT/256
2950 C704 8D3001 STA $0130
2960 C707 8E3101 STX $0131
2970 C70A 8C3201 STY $0132
2980 C70D 60 RTS
2990 ;
3000 ;***** DEMO1 Progr. D. M. V. NMI *****
3010 ;
3020 C70E 48 KLKINT PHA          SAVE REGISTERS
3030 C70F 8A TXA
3040 C710 48 PHA
3050 C711 98 TYA
3060 C712 48 PHA
3070 C713 201AC7 JSR DATTYD
3080 C716 2085F8 JSR RESREG          HERSTEL REGISTERS
3090 C719 40 RTI
3100 ;

```

```

3110 ;***** DATUM+TYD NAAR SCHEM *****
3120 ;
3130 C71A 20C7C5 DATTYD JSR DASC R
3140 C71D 20A7C5 JSR TISCR
3150 C720 60 RTS
3160 ;
3170 ;
3180 ;***** DEMO2-PROGR. D. M. V. PIA-TIMER *****
3190 ;
3200 C721 201AC7 DEMO JSR DATTYD
3210 ;
3220 C72A A20F LDX #002 TELLER 1
3230 C72E A00F LDY #00F TELLER 2
3240 C72B A92F DELAY LDA #02F
3250 C72A 8DF7C2 STA CLK1KT PRESET 6532-TIMER
3260 C72D 2CD5C2 RET2 BIT RDFLAG FLAG-POLLING
3270 C730 10FB BPL RET2
3280 C732 8B DEY
3290 C733 10F3 BPL DELAY
3300 C735 CA DEX
3310 C736 10F0 BPL DELAY
3320 C738 4C21C7 JMP DEMO VOLGENDE RONDE
3330 ;
3340 ;***** DISPLAY NA GELIJKZETTEN *****
3350 ;
3360 C73B 2012C4 ALLIN JSR SETKL
3370 C73E 4C21C7 JMP DEMO
3380 ;
3390 ;
END
    
```

MM58174A

**Interrupt System**  
 The interrupt output and its frequency of operation is enabled by writing to address 15 (see Table 2a). To ensure correct operation, the interrupt should be serviced within 100 ns.

The interrupt is initialized by writing "0" to address 15. The interrupt is then enabled by writing "1" to address 15. Initialization must be performed at power on and also if the interrupt is not serviced correctly within 100 ns.

**Service of the Interrupt**  
 In a typical system the open drain interrupt output is wired to the processor interrupt system. Hence, when the interrupt timer times out, the interrupt output is pulled low and the processor is interrupted.

The processor may then reset the interrupt by utilizing the output pin for 15 three times.  
 This resets the interrupt output and restarts the interrupt timer when in the repeat mode.

It is recommended that the interrupt output is connected to a unique processor port.

**Crystal Parameters**  
 The following is a typical representation of the crystal used with some typical values. The 32,768 kHz crystal is an NT CUT (tuning fork type) or XY BAR for use in a parallel resonant Pierce oscillator.

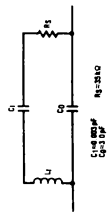


Figure 4. Typical Crystal Parameters

**Data Changed FIF**  
 This is any five-bit value of each Data byte to indicate the new value to be changed since the last read operation. It is reset by any clock read command. The flip-flop sets all data bus bits to a "1" during MPOS time indicating that a register has been updated. This transient condition may occur at the end of the Read Data strobe. Hence, invalid data may still be read from the clock, if the strobe width was less than 100 ns.

The possibility may be overcome by implementing a four-bit strobe width of every series of reads (starting with a read at the tenth or second's register) and checking for unchanged data

**Table 1. Address Decoding for Internal Registers**

Selected Counter	Address Bits	AD <sub>7</sub>	AD <sub>6</sub>	AD <sub>5</sub>	AD <sub>4</sub>	AD <sub>3</sub>	Mode
0	Test Only	0	0	0	0	0	Write Only
1	Units of days	0	0	0	0	0	Read Only
2	Units of weeks	0	0	0	0	1	Read Only
3	Units of secs.	0	0	0	1	0	Read Only
4	Units of mins.	0	0	1	0	0	Read or Write
5	Units of hours	0	1	0	0	0	Read or Write
6	Units of days	0	1	0	1	0	Read or Write
7	Units of weeks	0	1	0	1	1	Read or Write
8	Units of days	1	0	0	0	0	Read or Write
9	Units of weeks	1	0	0	0	1	Read or Write
10	Units of mins.	1	0	1	0	0	Read or Write
11	Units of months	1	0	1	1	0	Read or Write
12	Units of months	1	0	1	1	1	Read or Write
13	Years	1	1	0	0	0	Write Only
14	Years	1	1	0	0	1	Write Only
15	Interrupt	1	1	1	1	1	Read or Write

**Table 2a. Interrupt Selection Data**

Mode	Address 15, Write Mode	DB <sub>7</sub>	DB <sub>6</sub>	DB <sub>5</sub>	DB <sub>4</sub>	DB <sub>3</sub>	DB <sub>2</sub>	DB <sub>1</sub>	DB <sub>0</sub>
NO interrupt		X	0	0	0	0	0	0	0
Int. at 60 sec. intervals*		0	1	0	0	0	0	0	0
Int. at 5.0 sec. intervals*		0	1	0	1	0	0	0	0
Int. at 0.5 sec. intervals*		0	1	0	1	0	1	0	0
* - 10.0 MHz		0	1	0	1	0	1	1	0

DB<sub>0</sub> = 0, single interrupt DB<sub>0</sub> = 1, repeated interrupt

**Table 2b. Interrupt Read Back (Status)**

Mode	Address 15, Read Mode	DB <sub>7</sub>	DB <sub>6</sub>	DB <sub>5</sub>	DB <sub>4</sub>	DB <sub>3</sub>	DB <sub>2</sub>	DB <sub>1</sub>	DB <sub>0</sub>
Reset		X	0	0	0	0	0	0	0
60 sec. signal		X	1	0	0	0	0	0	0
5.0 sec. signal		X	0	1	0	0	0	0	0
0.5 sec. signal		X	0	0	1	0	0	0	0
X = don't care state		X	0	0	0	1	0	0	0

**Table 3. Years Status Register**

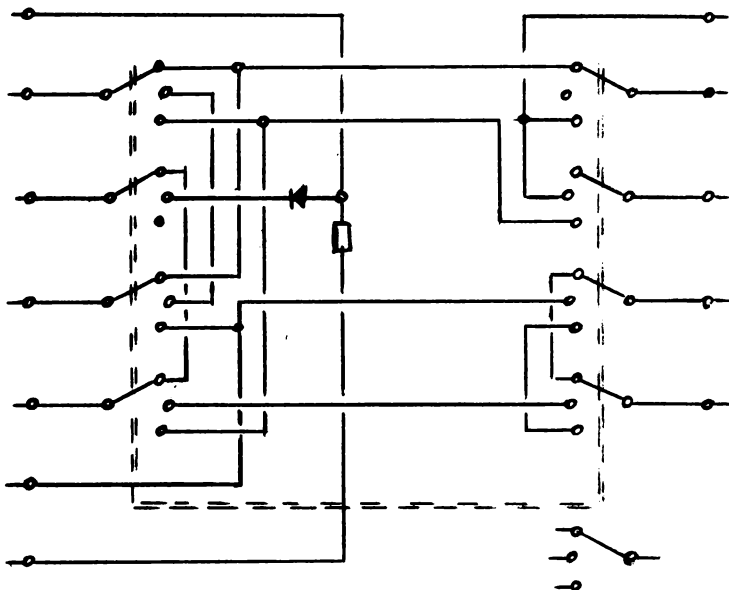
Mode	Address 15, Write Mode	DB <sub>7</sub>	DB <sub>6</sub>	DB <sub>5</sub>	DB <sub>4</sub>	DB <sub>3</sub>	DB <sub>2</sub>	DB <sub>1</sub>	DB <sub>0</sub>
Leap year		1	0	0	0	0	0	0	0
Leap year + 1		0	1	0	0	0	0	0	0
Leap year + 2		0	0	1	0	0	0	0	0
Leap year + 3		0	0	0	1	0	0	0	0



EM V2.3

&gt;DC400, C750

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
C400	4C	21	C7	4C	1A	C7	4C	3B	C7	4C	12	C4	4C	F4	C4	4C	
C410	FE	C6	20	F4	C4	A9	00	8D	FE	C7	8D	29	02	A9	09	8D	
C420	2A	02	A0	FF	C8	B9	49	C4	F0	06	20	69	FF	4C	24	C4	
C430	A2	09	20	00	FD	20	69	FF	20	93	FE	30	F5	9D	F4	C7	
C440	CA	10	EF	A9	01	8D	FE	C7	60	0A	0D	0A	0D	56	4F	4F	
C450	52	20	53	43	48	52	49	4B	48	45	4C	0A	41	41	52	20	
C460	52	45	53	50	2E	2B	31	2C	2B	32	2C	2B	33	2C	0A	0D	
C470	20	20	4A	41	41	52	4E	52	2E	20	49	4E	56	55	4C	4C	
C480	45	4E	3A	20	38	20	52	45	53	50	2E	31	2C	32	2C	34	
C490	2C	0A	0D	56	4F	4F	52	20	44	41	47	4E	52	2E	20	47	
C4A0	45	4C	44	54	3A	20	5A	4F	4E	44	41	47	3D	31	0A	0D	
C4B0	4A	41	41	52	4E	52	2F	4D	4E	44	2F	44	41	47	4E	52	
C4C0	2F	44	41	54	55	4D	2F	55	55	52	2F	4D	49	4E	0A	0D	
C4D0	0A	0D	46	4F	52	4D	41	54	3A	20	4A	2F	4D	41	2F	44	
C4E0	2F	44	41	2F	55	55	2F	4D	49	0A	0D	20	20	20	20	20	
C4F0	20	20	20	00	A9	00	8D	F0	C7	8D	FF	C7	AD	FF	C7	AD	
C500	FF	C7	AD	FF	C7	A9	01	8D	FE	C7	A9	09	8D	FF	C7	60	
C510	AD	F1	C7	AD	F1	C7	AD	F1	C7	20	9D	05	8D	02	0C	8D	F0
C520	C7	20	9D	C5	29	0F	09	30	9D	FF	C1	CA	D0	F0	AD	F1	
C530	C7	20	9D	C5	20	38	C5	60	8E	1D	C2	8E	27	C2	A9	D0	
C540	8D	1E	C2	8D	2B	C2	A9	3A	8D	17	C2	8D	1A	C2	A9	2F	
C550	8D	21	C2	8D	24	C2	AD	06	C2	8D	15	C2	AD	05	C2	8D	
C560	16	C2	AD	04	C2	8D	18	C2	AD	03	C2	8D	19	C2	AD	02	
C570	C2	8D	1B	C2	AD	01	C2	8D	1C	C2	AD	0B	C2	8D	1F	C2	
C580	AD	07	C2	8D	20	C2	AD	0B	C2	8D	22	C2	AD	0A	C2	8D	
X C590	23	C2	A9	08	8D	25	C2	A9	04	8D	26	C2	60	C9	FF	F0	
C5A0	01	60	68	68	4C	10	C5	20	10	C5	A2	00	A0	05	B9	01	
C5B0	C2	9D	E9	F7	E8	88	B9	01	C2	9D	E9	F7	E8	A9	3A	0D	
C5C0	E9	D7	E8	88	10	E8	60	AD	29	02	48	AD	2A	02	48	20	
C5D0	10	C5	A2	12	8E	29	02	A2	1F	8E	2A	02	AD	09	C2	29	
C5E0	0F	20	1F	C6	AD	08	C2	8D	DC	D7	AD	07	C2	8D	DD	D7	
C5F0	A9	20	8D	DE	D7	A2	1F	8E	29	02	AD	0B	C2	29	0F	F0	
C600	0A	AD	0A	C2	29	0F	18	69	0A	D0	06	AD	0A	C2	29	0F	
C610	18	69	07	20	1F	C6	68	8D	2A	02	68	8D	29	02	60	AA	
C620	CA	A0	FF	CA	30	08	C8	B9	40	C6	10	FA	30	F5	C8	B9	
C630	40	C6	30	06	20	69	FF	4C	2E	C6	29	7F	20	69	FF	60	
C640	20	20	20	5A	4F	4E	44	41	47	A0	20	20	4D	41	41	4E	
C650	44	41	47	A0	20	20	44	49	4E	53	44	41	47	A0	20	57	
C660	4F	45	4E	53	44	41	47	A0	44	4F	4E	44	45	52	44	41	
C670	47	A0	20	20	56	52	49	4A	44	41	47	A0	20	5A	41	54	
C680	45	52	44	41	47	A0	4A	41	4E	55	41	52	49	20	20	A0	
C690	46	45	42	52	55	41	52	49	20	A0	4D	41	41	52	54	20	
C6A0	20	20	20	A0	41	50	52	49	4C	20	20	20	20	A0	4D	45	
C6B0	49	20	20	20	20	20	20	A0	4A	55	4E	49	20	20	20	20	
C6C0	20	A0	4A	55	4C	49	20	20	20	20	A0	41	55	47	55		
C6D0	53	54	55	53	20	A0	53	45	50	54	45	4D	42	45	52	A0	
C6E0	4F	4B	54	4F	42	45	52	20	20	A0	4E	4F	56	45	4D	42	
C6F0	45	52	20	A0	44	45	43	45	4D	42	45	52	20	A0	A9	4C	
C700	A2	0E	A0	C7	8D	30	01	8E	31	01	8C	32	01	60	48	8A	
C710	48	98	40	20	1A	C7	20	85	F8	40	20	C7	C5	20	A7	C5	
C720	60	20	1A	C7	A2	02	A0	0F	A9	2F	8D	F7	C2	2C	D5	C2	
C730	10	FB	88	10	F3	CA	10	F0	4C	21	C7	20	12	C4	4C	21	
C740	C7	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	



### WIJZIGING IN 'ZERO' EPROM PROGRAMMER

Vervanging 'SELECTIE' DIL voetje door een schakelaar.

Bent U ook in het bezit van een 'ZERO' EPROM PROGRAMMER ? Persoonlijk vind ik het een voortreffelijk stukje techniek met zeer uitgeknipte software, met maar een groot praktisch bezwaar: de methode van selecteren van het te behandelen EPROM

Biggaand schema vervangt het 14-polige dil voetje door een 8-voudige 3-standen draaischakelaar. (de stand 'WRONG' heb ik niet geïmplementeerd, omdat alleen bij foutief plaatsen van de socket deze melding optreedt).

De schakelaar is met de twee extra componentjes, via een 16-aderige bandkabel met de dilconnector met het programma-printje verbonden.

De dil connector wordt gestoken op de plaats van het selector-voetje.

De referentienummers in de tekening zijn de pinnummers van de 16-polige selectievoet.

De aanduidingen tussen haakjes verwijzen naar het schema van ZERO.

Kees van Luypen  
079-213245

[ SWITCH ] 12  
[ CMD1 ] 7  
[ CMD2 ] 11  
[ EPROM#18 ] 10  
[ EPROM#20 ] 6  
[ CTRL ] 9  
[ +5VOLT ] 13

GEDIODE  
1K

3 [ 0VOLT ]  
14 [ CDATA5 ]  
2 [ CDATA6 ]  
8 [ EPROM#21 ]  
5 [ 5/25VSWITCH ]

NOVEMBER 1984

[2714]  
[2732]  
[2532]

## AANPASSINGEN IN DE HENK-WEVERS-MONITOR

Dhr Frans Liekens uit Genk in België heeft een aantal verbeteringen gemaakt voor de Henk Wevers monitor (in EPROM).

Hij maakt echter gebruik van een oudere versie van de H.W. V4 monitor, zodat soms moet worden gekozen, en de door hem voorgestelde toetsenbordroutine niet kan worden gebruikt zonder erg grote aanpassingen. Zeker voor de Disk-gebruikers zijn die toetsenbordveranderingen niet door te voeren.

Voor geïnteresseerden is natuurlijk een kopie beschikbaar. De routine is zeker de moeite waard omdat het daarmee mogelijk is 64 toetsen te decoderen !!

EEN VERBETERDE DECODEERROUTINE VOOR DE CONTROLE-TOETSEN. (CTRL-A t/m CTRL-Z en de CHR\$(1..26))  
Het verschil met de bestaande routine is dat er een tabel wordt geraadpleegd, die voor een gedeelte in RAM-geheugen ligt, zodat alle (=alle) 26 CTRL toetsen kunnen worden uitgedecodeerd. Bovendien kan de gebruiker zelfs binnen een programma, BASIC of Machinetaal, naar een voorgeprogrammeerde routine springen.

## \*\*\*\*\* ADRESSEN - TABEL \*\*\*\*\*

CHR#	CTRL	H-W V4	LIEKENS	ALTERN.	ADRES
1	A	F954	F954	----	F846
2	B	F993	F993	----	F848
3	C	FA1B	FA1B	----	F84A
4	D	FA7C	FA7C	----	F84C
5	E	F9C5	F9C5	----	F84E
6	F	----	FD00	----	F850
7	G	FBD0	FBD0	----	F852
8	H	----	02E1	0131	F854
9	I	FA44	FA44	----	F856
10	J	F8D5	F8D5	----	F858
11	K	----	02FC	014C	F85A
12	L	FA21	FA21	----	F85C
13	M	F9B9	F9B9	----	F85E
14	N	----	02F0	0140	F860
15	O	----	02F3	0143	F862
16	P	----	02F6	0146	F864
17	Q	----	02F9	0149	F866
18	R	FA38	FA38	----	F868
19	S	----	02E4	0134	F86A
20	T	----	02E7	0137	F86C
21	U	FA27	FA27	----	F86E
22	V	----	02EA	013A	F870
23	W	----	02ED	013D	F872
24	X	FF47	FF47	----	F874
25	Y	FF58	FF58	----	F876
26	Z	FF52	FF52	----	F878

De niet ingevulde adressen onder H-W V4 worden niet gedecodeerd. In het programmeel van Frans wordt in dat geval gesprongen naar het deel vrije RAM in pagina 2. Deze oplossing heeft wat mij betreft de voorkeur. Als alternatief kan echter ook het gedeelte in pagina 1, waar de NMI vektor normaal staat, worden gebruikt. Vooral als de Extended Monitor wordt gebruikt, kunnen er conflicten ontstaan in de pagina 1. De werking is nu als volgt; stel de gebruiker wil met CTRL-S de save-routine in de toolkit rechtstreeks aansturen. Zet dan op 02E4/02E6 (of op 0134/0136) 4C 00 97 (of 00 E7 bij 'n andere TK) Deze adressen zijn in de tabel te vinden achter 19 S etc. Ook kan natuurlijk dit sprongadres in de monitor-EPROM worden vast-gebakken (op de adressen FB6A/6B). De ruimte waar de sprongadressen worden weggezet in pagina 2 of 1, moet ook nog worden volgeschreven met \$60 (=RTS) zodat er geen problemen komen als er geen gebruik wordt gemaakt van de extra CTRL toetsen. Zie de listing hieronder

ADRES	CODE	OPDRACHT	
F800	8D0202	STA \$0202	BEWAAR ACCUINHOUDE
F803	48	PHA	ZET ACCU, EN
F804	8A	TXA	X-REGISTER
F805	48	PHA	EN
F806	98	TYA	Y-REGISTER
F807	48	PHA	OP DE STACK
F808	A000	LDY £\$00	ZET STANDAARDTELLER OP 0
F80A	AD0202	LDA \$0202	HAAL ACCU OP
F80D	F01C	BEQ \$F82B	SPRING BIJ 'N NULL
F80F	C91B	CMP £\$1B	ACCU >= DAN \$1B ?
F811	B018	BCS \$F82B	ZOJA OVER CTRL-STUK SPRINGEN
F813	48	PHA	EVEN BEWAREN
F815	AA	TAX	EN ALS INDEX GEBRUIKEN
F816	BD44F8	LDA \$F844,X	HAAL ADRES IN DE TABEL
F819	8D1102	STA \$0211	EN ZET IN TIJDELIJK VRIJ
F81C	BD45F8	LDA \$F845,X	DEEL VAN HET RAM
F81F	8D1002	STA \$0210	
F822	A94C	LDA £\$4C	INKLUSIEF DE SPRONG
F824	8D0F02	STA \$020F	OPDRACHT (=JMP)
F827	68	PLA	ACCU VOOR EV.HERGEBRUIK
F828	200F02	JSR \$020F	VOER DE OPDRACHT UIT
F82B	C95F	CMP £\$5F	EEN BACKSPACE ?
F82D	D003	BNE \$F832	ZONIET, VERDER
F82F	20E0F9	JSR \$F9E0	DOE DE BACKSPACE ROUTINE
F832	C97F	CMP £\$7F	DELETE-TOETS?
F834	D003	BNE \$F839	ZONIET, VERDER
F836	207CFA	JSR \$FA7C	GA NAAR DELETERROUTINE
F839	C91A	CMP £\$1A	IS DE ACCU<\$1A?
F83B	9003	BCC \$F840	ZOJA, VERDER
F83D	203EFA	JSR \$FA3E	SCHRIJF KARAKTER WEG

F840	68	PLA	HERSTEL	REGISTERS
F841	A8	TAY		
F842	68	PLA		
F843	AA	TAX		
F844	68	PLA		
F845	60	RTS		

Op de adressen \$F846 t/m \$F879 staat de tabel met de sprongadressen. Let wel, in de tabel staan de adressen, die zelf in te vullen zijn in pagina 2 naar wens moeten die volgens de eerste tabel worden aangepast.

	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5
F846	F9	54	F9	93	FA	1B	FA	7C	F9	C5	FD	00	FB	D0	02	E1
F856	FA	44	F8	D5	02	FC	FA	21	G9	B9	02	F0	02	F3	02	F6
F866	02	F9	FA	38	02	E4	02	E7	FA	27	02	EA	02	ED	FF	47
F876	FF	58	FF	52												

In de H-W. monitor V4 blijven nu nog 17 bytes over totdat 'BERCUR' begint op \$F88B. Op die plaats kan precies de RTS (=60) initialisatie-routine staan, waar al eerder sprake van was. Zorg ervoor de STA indirect,Y op \$F87F/80 aan te passen voor de goede adressen (tabel 1.).

F87A	A960	LDA £#60	ZET HET RTS-TEKEN IN ACCU
F87C	A020	LDY £#20	ZET DAT 32 MAAL WEG
F87E	99DF02	STA \$02DF,Y	IN DE TABEL (OF IN 012F,Y)
F882	88	DEY	VOLGENDE PLEK
F883	DOFA	BNE \$F87E	SPRING ZOLANG NIET EINDE TABEL
F88u	4C22BD	JMP \$BD22	NAAR DE KOUDE START/BASIC

Blijft alleen nog over het sprongadres naar deze routine in de RESET routine op \$FF3E.  
Daar stond 22 BD dat wordt 7A FB

Als toegift is ook nog een soort automatische start-routine voor de Toolkyt door hem bedacht. Bij de warme-Basicstart wordt in plaats van een sprong naar \$0000, waar JMP \$A274 staat, gesprongen naar een soort 'tussenroutine', die controleert of er een Toolkit aanwezig is. Als dat zo is, wordt de toolkit geïnitieerd. Jammer genoeg wordt de ruimte in de monitor in Versie 4 door een andere routine ingenomen, zodat deze 16 bytes ev. op een andere plek (in de toolkit zelf, in BASIC4 of in de diskplaats in pagina \$FC voor niet-disk gebruikers) moeten worden weggepropt, als door Frans was bedacht (dat was \$F886).

```
$???? A2FC   LDX £$FC   ZET STACK GOED
$???? 9A     TXS       ZONDER O* ERROR MELDING
$???? AD0090 LDA $9000 HAAL EERSTE ADRES OP, MITS
                        T.K. OP 9000 STAAT
                        ANDERS AANPASSEN.
$???? C9A9   CMP £$A9   EERSTE ADRESINHHOUD.
$???? C9A9   CMP £$A9   EERSTE ADRESINHHOUD.
$???? F003   BEQ VERDER ZOJA T.K. STARTEN
$???? 4C0000 JMP $0000  ZONEE NORMALE WARMte START
VERF. 4C0090 JMP $9000  START EVENTUEEL OP ANDER
                        ADRES.
```

De verandering in het 'RESET' gedeelte van de monitor vanaf \$FF00 is dan op \$FF2F, waar in de plaats van 4C0000 (=JMP \$0000) 4C???? moet komen. Een andere minder ingrijpende oplossing is natuurlijk direkt op die plaats het Toolkit-startadres te zetten.

Jos Burghouts

```

1000 REM *****
1001 REM * BASIC SCHAAK *
1002 REM *****
1003 REM
1004 DIM AZ(1),Z(1,200):REM de 1 is MAX.PLY-1
1005 DIM B(122),S(5,5)
1009 REM--vul bord met rand--
1010 FOR T=0 TO 122:B(T)=-2:NEXT T
1019 REM--vul bord met beginopstelling--
1020 RESTORE:FOR T=22 TO 29
1030 READ B(T):B(T+70)=-B(T):B(T+60)=-1:B(T+10)=1
1040 B(T+20)=0:B(T+30)=0:B(T+40)=0:B(T+50)=0
1050 NEXT T:TZ=1:REM--beurtenteller--
1059 REM--vul array met beweging koersen--
1060 FOR I=0 TO 5:FOR J=0 TO 5
1070 READ S(I,J)
1080 NEXT J,I
1089 REM--data voor bord opstelling--
1090 DATA 5,3,4,9,100,4,3,5
1094 REM--zet printer routine neer--
1095 POKE 617,141:POKE 618,1:POKE 619,242
1096 POKE 620,76:POKE 621,105:POKE 622,255
1097 POKE 539,2:SAVE:PRINT CHR$(30)
1100 INPUT"Beginnt U met zwart of met wit (W/Z) ";A$:PRINTCHR$(26);
1102 IF A$="W" THEN KL= 1:GOSUB 2500:GOTO 3115
1103 IF A$="Z" THEN KL=-1:GOSUB 2500:GOTO 3000
1104 PRINTCHR$(26):GOTO 1100
1105 REM--scan bord voor zettengenerator--
1110 T=22:QT=0
1120 S=B(T):IF S=0 THEN 1150
1130 IF S=-2 OR ABS(S)/S<>KL THEN 1150
1140 GOSUB 1500
1150 T=T+1:IF T<101 THEN 1120
1160 AZ(PLY)=QT-3:QX=QX+QT-3:RETURN
1500 REM--stuk gevonden,bepaal koers tabel--
1510 BW=0:FOR I=0 TO 5
1520 IF ABS(S)=S(I,0) THEN 1600
1530 NEXT I
1540 REM--data voor richting tabel--
1550 DATA 1,10,9,11,0,0
1560 DATA 3,8,12,19,21,0
1570 DATA 4,11,9,0,0,0
1575 DATA 5,1,10,0,0,0
1580 DATA 9,1,9,10,11,0
1590 DATA 100,1,9,10,11,0
1599 REM--genereer alle requelementaire zetten voor gevonden stuk--
1600 WZ=0:BW=BW+1:A=S(I,BW)*KL:IF A=0 THEN RETURN
1601 REM--tot regel 1640 alle pos.koersen--
1602 REM--van regel 1640-1710 alle neg.koersen--
1605 B=A
1609 REM--rand bord--
1610 IF B(T+B)=-2 THEN 1639
1611 REM--leeg veld--
1612 IF B(T+B)=0 THEN 1625
1614 REM--eigen stuk--
1615 IF ABS(B(T+B))/B(T+B)=KL THEN 1639
1619 REM--stuk tegenstander--
1620 IF ABS(B(T+B))/B(T+B)=-KL THEN WZ=ABS(B(T+B))
1621 REM--pion mag niet zetten als er een stuk voor staat--
1622 IFABS(S)=1 AND ABS(A)=10 AND WZ<>0 THEN 1600

```

```
1625 REM--pion mag niet slaan naar leeg veld--
1626 IF ABS(S)=1 AND ABS(A)<>10 AND WZ=0 THEN 1600
1628 GOSUB 2000:REM--bewaars zet--
1629 REM--is er iets geslagen dan beweging afkappen--
1630 IF WZ<>0 THEN WZ=0:GOTO 1639
1631 REM--koning en paard mogen maar 1 keer in bepaalde koers--
1632 IF ABS(S)=3 OR ABS(S)=100 THEN 1640
1633 REM--pion ook maar 1 keer en bovendien alleen vooruit--
1634 IF ABS(S)=1 THEN 1600
1635 REM--blijf op deze koers verdergaan--
1636 B=B+A:GOTO 1610
1638 REM--pion alleen pos. richting--
1639 IF ABS(S)=1 THEN 1600
1640 B=A
1645 IF B(T-B)=-2 THEN 1600:REM--rand--
1650 IF B(T-B)=0 THEN 1680:REM--leeg--
1660 IF ABS(B(T-B))/B(T-B)=KL THEN 1600
1670 IF ABS(B(T-B))/B(T-B)=-KL THEN WZ=ABS(B(T-B))
1680 GOSUB 2100:REM--bewaars zet--
1685 IF WZ<>0 THEN 1600
1690 REM--koning of paard ?--
1700 IF ABS(S)=3 OR ABS(S)=100 THEN 1600
1710 B=B+A:GOTO 1645
2000 REM--bewaars zet in Z-array--
2010 Z(PLY,QT)=T:Z(PLY,QT+1)=T+B:Z(PLY,QT+2)=WZ
2020 QT=QT+3:RETURN
2100 REM--bewaars zet in Z-array--
2110 Z(PLY,QT)=T:Z(PLY,QT+1)=T-B:Z(PLY,QT+2)=WZ
2120 QT=QT+3:RETURN
2199 REM--stop zet in Z$, print Z$ als kl.=zwart--
2200 CJ=INT(T/10)
2210 LE=T-CJ*10
2215 IF KL=1 THEN GOSUB 2600:TZ=TZ+1:GOTO 2220
2216 FOR KN=1 TO 20-LEN(Z$):Z$=Z$+" ":NEXT
2220 Z$=Z$+CHR$(63+LE)+CHR$(47+CJ)
2225 IF B(NT)<>0 THEN Z$=Z$+"x":GOTO 2230
2226 Z$=Z$+"-"
2230 CJ=INT(NT/10)
2240 LE=NT-CJ*10
2250 Z$=Z$+CHR$(63+LE)+CHR$(47+CJ)+" "+STR$(ABS(WZ))
2255 IF KL=-1 THEN POKE 539,2:PRINT Z$:" -->":QX/3:POKE 539,255:Z$=""
2260 RETURN
2300 REM--zet string om in zet--
2305 T=0:NT=0:ER=0
2310 FOR I=1 TO 2
2320 AS=ASC(MID$(Q$,I,1))
2330 IF AS>64 AND AS<73 THEN T=T+(AS-65)
2340 IF AS>48 AND AS<57 THEN T=T+10*(AS-47)+2
2350 NEXT I
2360 FOR I=3 TO 4
2370 AS=ASC(MID$(Q$,I,1))
2380 IF AS>64 AND AS<73 THEN NT=NT+(AS-65)
2390 IF AS>48 AND AS<57 THEN NT=NT+10*(AS-47)+2
2400 NEXT I:REM--test op fouten--
2410 IF T<22 OR T>100 OR NT<22 OR NT>100 THEN ER=1:RETURN
2420 IF B(T)=0 OR B(T)=-2 OR B(NT)=-2 THEN ER=1:RETURN
2430 IF ABS(B(T))/B(T)<>KL THEN ER=1:RETURN
2440 IF B(NT)=0 THEN 2460
2450 IF ABS(B(NT))/B(NT)=KL THEN ER=1:RETURN
2460 ER=0:RETURN
```



```

2499 REM--print titel--
2500 PRINTTAB(5)"WIT"TAB(20)"ZWART"
2505 IF KL= 1 THEN PRINT TAB(5)"SPELER"TAB(20)"COMPIE"
2510 IF KL=-1 THEN PRINT TAB(5)"COMPIE"TAB(20)"SPELER"
2520 PRINT:POKE 539,255:RETURN
2600 REM--zet beurtenteller in Z#--
2610 Z#=STR$(TZ):FOR KN=1 TO 5-LEN(Z#)
2620 Z#=Z#+ " ":NEXT:RETURN
2999 REM--boom opzetten--
3000 PLY=0:KL=-KL:OX=0:GOSUB 1105:M1=-200:KC=KL
3010 FOR EP=0 TO AZ(0) STEP 3
3020 OP=Z(0,EP):NP=Z(0,EP+1):WA=Z(0,EP+2)
3025 REM--doe zet--
3030 OW=B(NP):B(NP)=B(OP):B(OP)=0
3035 REM verander van kleur en ga ply hoger--
3040 PLY=1:KL=-KC:GOSUB1105:M2=200
3050 FOR ER=0 TO AZ(1) STEP 3
3055 WQ=Z(1,ER+2)
3059 REM--minimaliseer verlies--
3060 IF WA-WQ<M2 THEN M2=WA-WQ
3070 NEXT ER
3074 REM--maximaliseer winst--
3075 IF M2>M1 THEN M1=M2:ZET=EP
3079 REM--terug zet--
3080 B(OP)=B(NP):B(NP)=OW
3090 NEXT EP:KL=KC
3094 REM--print beste zet--
3095 IF M1<-80 THEN 3200:REM--mat--
3100 T=Z(0,ZET):NT=Z(0,ZET+1):WZ=Z(0,ZET+2):GOSUB 2200
3109 REM--doe beste zet en verander van kleur--
3110 B(NT)=B(T):B(T)=0:KL=-KL
3111 REM--verander tekstwindow en bewaar cursor rij--
3115 C=PEEK(554):POKE 548,30:POKE 549,30
3118 POKE 539,255:REM--printer uit--
3119 PRINT CHR$(2):IF KL=-1 THEN PRINT Z#;" ";
3120 INPUT"Uw zet ";Q#:GOSUB 2300
3121 IF ER THEN PRINT TAB(20)"Foute invoer":GOTO 3119
3122 REM--reset tekstwindow en cursor rij--
3125 POKE 548,3:POKE 554,C:PRINT CHR$(13);
3126 REM--print zet + voer uit--
3130 WZ=B(NT):GOSUB 2200:B(NT)=B(T):B(T)=0:GOTO 3000
3200 POKE539,2:PRINTZ#;" Ik verlies":POKE539,255:POKE517,0:END

```

ready

## Opmerkingen:

1. voer zetten in ZONDER streepje, dus E2E4 i.p.v. E2-E4
2. pijltje geeft aantal overwogen zetmogelijkheden aan.

## Voorbeeld:

```

Begin U met zwart of met wit (W/Z) ? W
      WIT          ZWART
      SPELER      COMPIE

```

Uw zet ? E2E4

```

1  E2-E4, 0      B7-B6  0  --> 275

```

Uw zet ?



## ROSI HIGH RESOLUTION GRAPHICS \* HRG

Het videobeeld, dat wordt samengesteld met de karaktergenerator van het OSI-superboard, bestaat uit karakters van 8x8 bits. Per regel worden (na modificatie) 64 karakters geschreven. Dit zijn  $64 \times 8 = 512$  punten naast elkaar. Er worden 32 regels geschreven van elk 8 punten boven elkaar ->  $8 \times 32 = 256$  lijnen onder elkaar.

Deze resolutie is ook aangehouden voor het HRG-beeld, dat dus  $256 \times 512 = 131072$  punten omvat. Elk punt komt overeen met 1 bit van

512 HRG-RAM, zodat dit  $131072/8$  bytes groot moet zijn d.i. 16 KRAM.  
Gekozen is voor 8 RAM-chips 6116, elk  $2K \times 8$  groot.

Uiteraard moet elke byte een eigen adres in het systeem hebben. De adressen lopen van \$4000 (=16384 decimaal) tot en met \$7FFF (32767 decimaal).

256 Deze geheugens worden bijna continu uitgelezen en op het scherm geschreven. Het uitlezen wordt alleen onderbroken, als de geheugeninhoud wordt gewijzigd.

De volgorde van de geheugens is zo gekozen, dat het eerste (\$4000) links boven geschreven wordt en \$4001 rechts ernaast. Het laatste van de bovenste lijn is \$403F, waarna op de tweede lijn begonnen wordt met \$4040.

Elk geheugen omvat 8 bits, waarvan op het scherm het rechtse het minst significant is, de waarde 1 vertegenwoordigt.

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

Wordt in geheugenadres \$4000 (=16384 dec.) het getal \$A3 (=163) geschreven, dan licht het geheugen links boven op het scherm op, waar een 1 in het geheugen staat.

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

A

3

Om dit erin te krijgen:

in ASSEMBLY:

in BASIC:

```
LDA #$A3  A9A3
STA $4000
```

```
POKE 16384,163
```

Voor meer info bel 02510-26520

Bob Rijkeboer  
Velsen-Noord

ROSI versus MOSI (32 lijnen van 64 karakters)

MOSI

A	15 14	13 12 11 10 9	8 7 6	5 4 3 2 1 0	
C		17 16 15 14 13	12 11 10	<del>X</del> 8 7 6 5 4 3	2 1 0
	VA	1 scherm = 32 regels	lijn char.gen	1 byte = 8 bits	

Als AB-7-6 zouden worden gebruikt om de HRG-RAM's aan te sturen, komt in de RAM's geen aansluitende (seriele) informatie te staan. Op het scherm wordt de volgorde:

4000	....	403F
4700	....	473F
5000	....	503F

Dit is bezwaarlijk, als een deel (b.v. de bovenste helft) moet worden gebruikt.

7700	....	773F
------	------	------

4040	....	407F
4740	....	477F
	enz.	

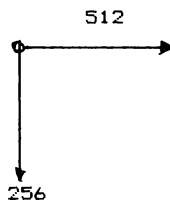
Om dit te ondervangen kan de schakeling worden:

ROSI

A	15 14	13 12 11	10 9 8 7 6	5 4 3 2 1 0	
C		17 16 15	14 13 12 11 10	<del>X</del> 8 7 6 5 4 3	2 1 0
	VA	8x2 kRAM	32 regels	64 bytes	8 bits

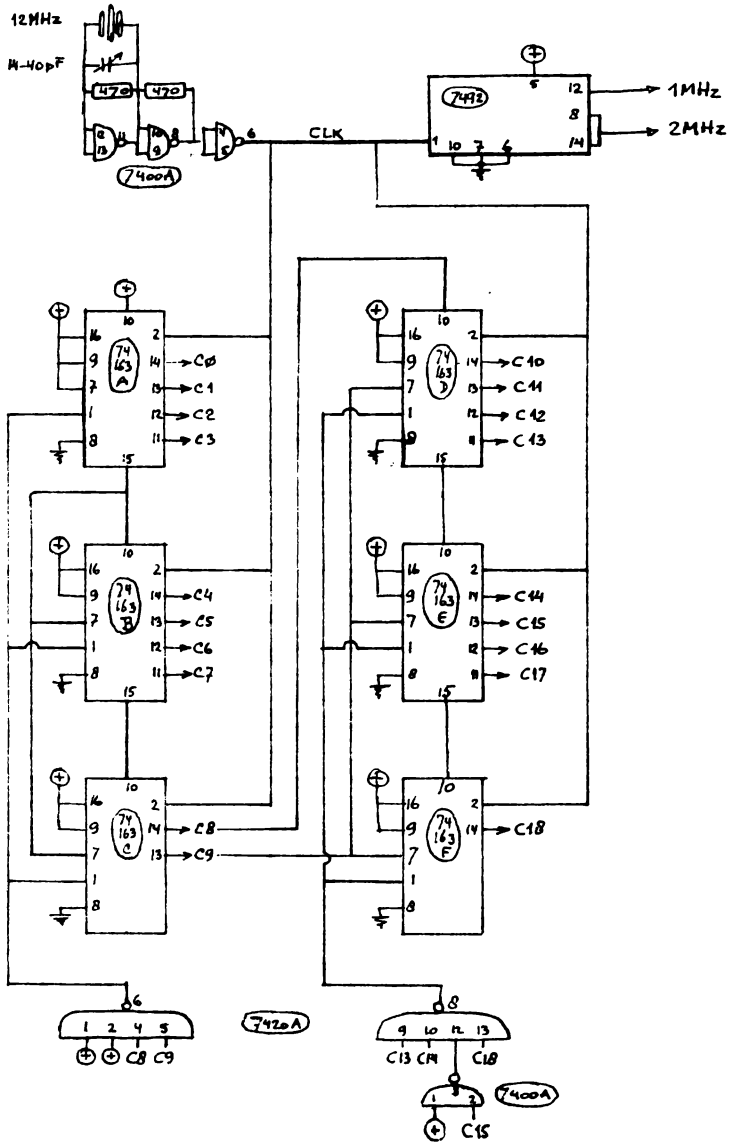
Nu wordt de volgorde in de RAM's gelijk aan de schermvolgorde, doorlopend vanaf links boven \$4000 tot rechts onder \$7FFF. Het adres van elk paar onder elkaar liggende geheugens verschilt \$0040 = 64 decimaal.

4000	....	403F
4040	....	407F
4080	....	40BF
40C0	....	40FF
4100	....	413F
	enz.	
7F00	....	7FFF



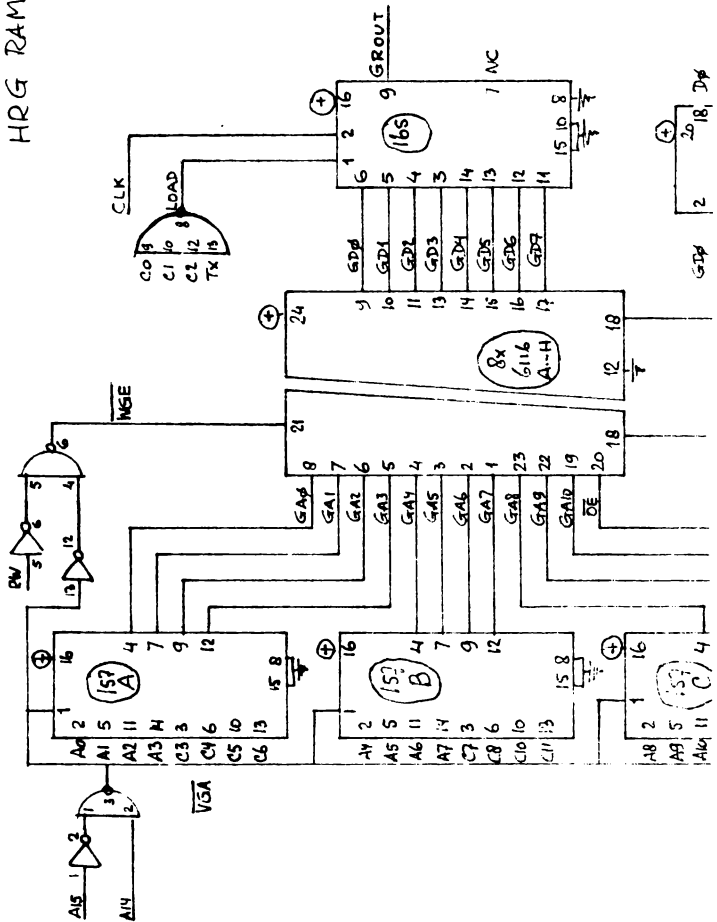
16 K HGR-RAM (vol scherm) = 131072 bits.

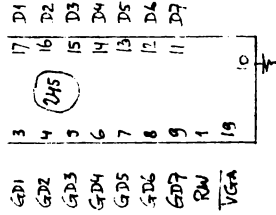
# OSCILLATOR EN DELETREIN



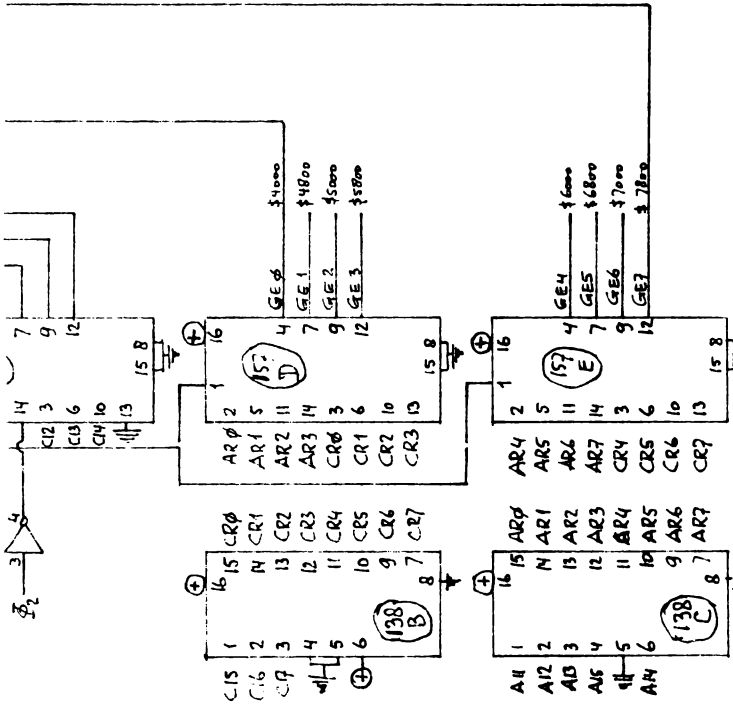
Posi  
maart 84

16K  
HRG RAM

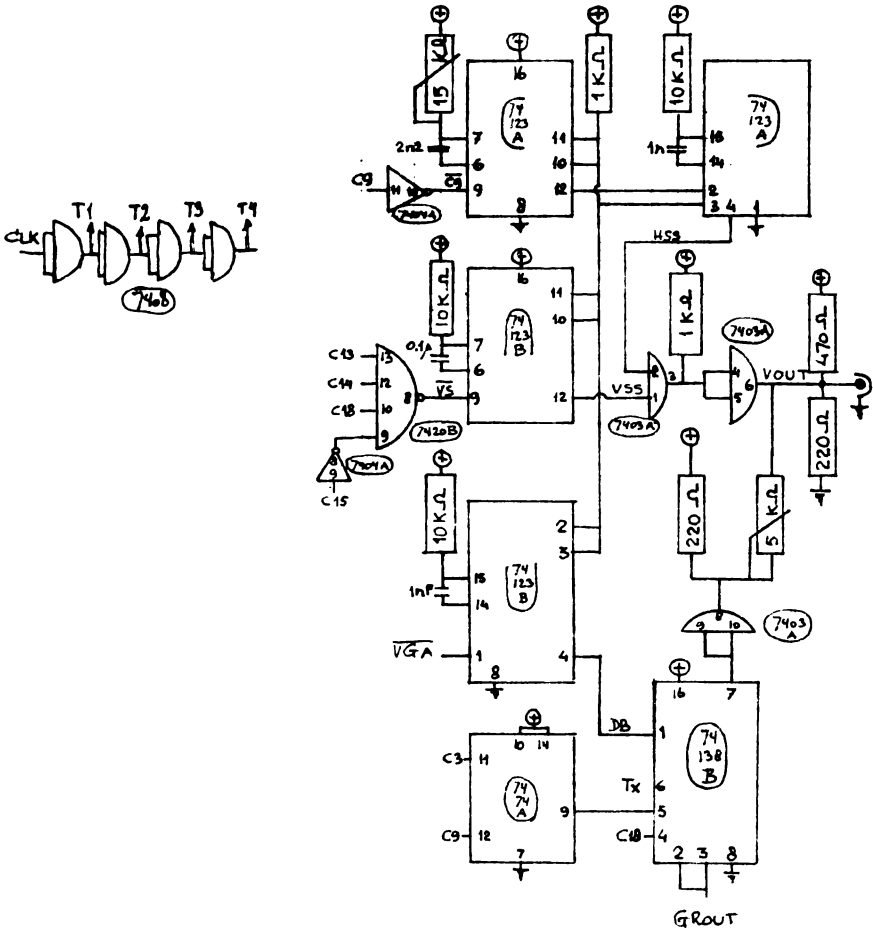




ROSL  
MILVANT 84

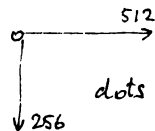


# SAMENSTELLING VIDEO SIGNAL



Posi  
maart 84

High Resolution Graphics





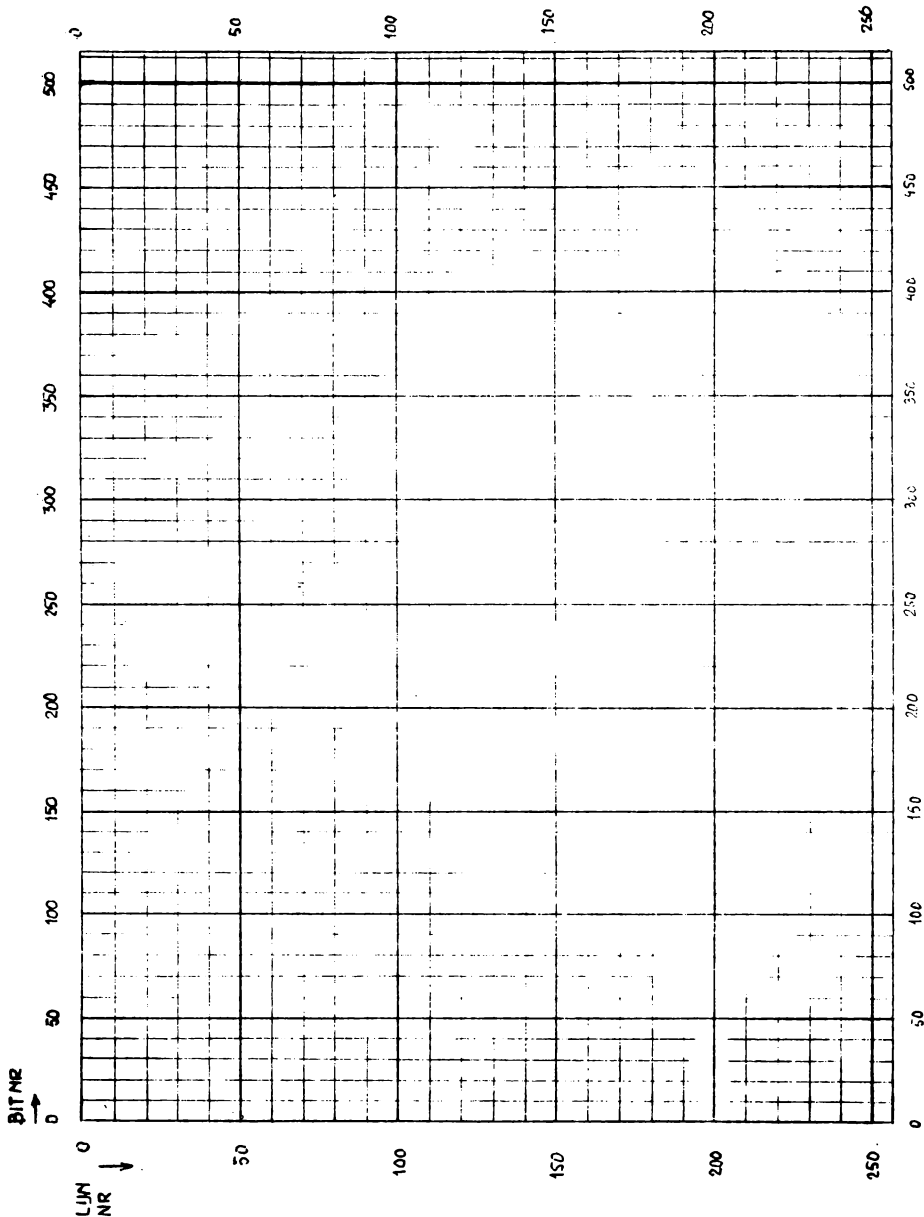
## Demonstratieprogramma HIGH RESOLUTION GRAPHICS.

Trek lijn tussen punt P en punt Q.

```

100 PRINTCHR$(26)
190 READ BP : READ LP : REM BITNR - LIJNNR P
200 READ BR : READ LR : REM BITNR - LIJNNR R
210 IF BR=.1 THEN 190
220 IF BP=BR THEN 400
230 IF BR=.2 THEN END
240 SP=1 : IF LP=LR THEN SP=2
244 IF BR<BP THEN SP=-SP
250 RC=(LP-LR)/(BP-BR)
260 FOR BB=BP TO BR STEP SP
270 LQ=INT(LP+RC*(BB-BP)+.5)
280 AD=16384+64*LQ+INT(BB/B)
290 WA=INT(128/(2^(BB-INT(BB/B)*8))+.5)
300 S=PEEK(AD) OR WA : POKE AD,S
310 NEXT BB
380 BP=BR : LP=LR : GOTO 200
400 AP=16384+LP*64+INT(BP/B)
410 AR=16384+LR*64+INT(BR/B)
420 WA=INT(128/(2^(BP-INT(BP/B)*8))+.5)
430 SP=64 : IF AR<AP THEN SP=-64
440 FOR BB=AP TO AR STEP SP
450 S=PEEK(BB) OR WA : POKEBB,S
460 NEXT BB
580 BP=BR : LP=LR : GOTO200
600 DATA130,130,130,70,160,70,170,80,170,100
610 DATA160,110,140,110,140,130,130,130,.1,.1
620 DATA140,100,140,80,150,80,160,90,150,100,140,100,.1,.1
630 DATA180,130,180,90,200,70,220,90,220,130,210,130,210,110
640 DATA190,110,190,130,180,130,.1,.1,190,100,190,95,200,85
650 DATA210,95,210,100,190,100,.1,.1,240,90,250,90,260,80
660 DATA230,80,230,70,270,70,270,80,260,90,270,100,270,120
670 DATA260,130,240,130,230,120,230,110,240,110,240,120
680 DATA250,120,260,110,250,100,240,100,240,90,.1,.1
690 DATA280,130,280,90,300,70,320,90,320,130,310,130,310,110
700 DATA290,110,290,130,280,130,.1,.1,290,100,290,95,300,85
710 DATA310,95,310,100,290,100,.1,.1,345,130,345,80,330,80
720 DATA330,70,370,70,370,80,355,80,355,130,345,130,.1,.1
730 DATA380,130,380,70,390,70,390,130,380,130,.1,.1
740 DATA130,160,140,160,140,185,145,190,150,185,150,160
750 DATA160,160,160,190,145,210,130,190,.1,.1,165,210,165,160
760 DATA195,160,195,170,175,170,175,180,195,180,195,190
770 DATA175,190,175,200,195,200,195,210,165,210,.1,.1
780 DATA210,160,200,160,200,210,230,210,230,200,210,200
790 DATA210,160,.1,.1,235,205,240,210,260,210,265,205,265,185
800 DATA260,180,245,180,245,170,255,170,260,175,265,170,265,165
810 DATA260,160,240,160,235,165,235,185,240,190,255,190,255,200
820 DATA250,200,245,195,235,200,235,205,.1,.1,130,190,130,160
830 DATA.1,.1,300,210,27,210,270,160,300,160
835 DATA300,170,280,170,280,180
840 DATA300,180,300,190,280,190,280,200,300,200,300,210,.1,.1
850 DATA305,210,305,160,315,160,325,170,325,160,335,160
860 DATA335,210,325,210,325,185,315,175,315,210,305,210,.1,.1
870 DATA340,190,340,180,355,180,355,190,340,190,.1,.1,360,210
880 DATA360,160,370,160,380,170,380,160,390,160,390,210,380,210
890 DATA380,185,370,175,370,210,360,210,.2,.2

```



NOVEMBER 1984

Posi

6502 TRACER

Onder deze titel verscheen in Elektuur van februari 1984 een artikel van J. Ruppert. Dit artikel zou de helft van het antwoord kunnen zijn op een opmerking die laatst iemand over mijn schouder heen maakte, toen ik op een van onze OSI-bijeenkomsten achter mijn systeem zat. Hij zei: "Jammer dat we nog geen "step by step utility" hebben. Dat zou het programmeren in machinetaal een stuk makkelijker maken !" De andere helft van het antwoord geeft de "6502-tracer" niet, omdat alles met grote snelheid over het beeldscherm vliegt. Dat is overigens een beetje te ondervangen door de computer in de SAVE-mode te zetten op een zo laag mogelijke BAUD-rate (\$0205-FF, \$F000-02 en hardware 300 BAUD). Uitprinten is nog handiger !

De "6502 tracer" was geschreven voor de Junior computer. Maar door aanpassing van de sprongen naar de "PRBYTE" en de "PRCHA" routines op resp. de adressen \$06A0 en \$06A5 zou dit programma op elk 6502-systeem moeten kunnen draaien.

De auteur vermeldde overigens niet dat dan ook de IRQ-vector op de adressen \$051C en \$0521 aangepast dient te worden. Deze vector is bij een 6502-systeem in eerste instantie altijd op de twee laatste adressen te vinden: \$FFFE en \$FFFF. In de OSI-1P-systeem ROM staat daar resp. \$C0 en \$01. Controleer dit overigens even ! In de monitor ROM's die ik heb is op adres \$FFFF-24 (blijven) staan en daarom moet vaak de "step stone" \$24C0 4C C0 01 worden gelegd.

De 6502 tracer begint op \$0500 en zet - als op \$00ED en \$00EE resp. de lage en de hoge byte van het startadres van het te testen programma zijn gezet - achtereenvolgens adres voor adres de belangrijkste processorgegevens naast elkaar op een regel.

Ton Helwig

## Stukje voorbeeld:

(wie wat wil verbeteren kan beginnen met het voorkomen van de overbodige line feeds).

```
:G0500
6502 - TRACER
ADR. -INSTR.- :A :Y :X NV--DIZO STACK
C800
4C 80 C8 00 00 00 ..... FD-1E08
C880
20 AE C8 00 00 00 ..... FB-C883
C8AE 48      00 00 00 ..... FA-00C8
C8AF A9 00   00 00 00 .....1. FA-00C8
C8B1 85 F2   00 00 00 .....1. FA-00C8
C8B3 A9 18   18 00 00 ..... FA-00C8
C8B5 85 F3   18 00 00 ..... FA-00C8
C8B7 68      00 00 00 .....1. FB-C883
```

HEX-dump van de OSI-1P versie van de 6502 tracer van J. Ruppert.

:00500,072F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0500	58	20	95	06	A9	00	A0	0F	99	13	07	88	D0	FA	B9	0C
0510	06	20	A5	06	C8	C0	36	D0	F5	A9	26	8D	C1	01	A9	05
0520	8D	C2	01	4C	A2	05	8D	1B	07	68	8D	20	07	68	68	8C
0530	1C	07	8E	1D	07	BA	8E	14	07	D8	58	A0	03	B9	15	07
0540	20	A0	06	20	A3	06	C8	C0	06	B0	11	A0	16	07	D0	09
0550	20	A3	06	20	A3	06	4C	43	05	CE	16	07	C0	09	D0	D0
0560	AD	20	07	29	CF	8D	13	07	A2	08	0E	13	07	90	04	A9
0570	31	D0	02	A9	2E	20	A5	06	CA	D0	EF	20	A3	06	AD	14
0580	07	20	A0	06	A9	2D	20	A5	06	BA	E0	FF	B0	14	68	8D
0590	16	07	20	A0	06	E0	FE	B0	05	68	48	20	A0	06	AD	16
05A0	07	48	A0	00	20	95	06	A5	EE	20	A0	06	A5	ED	20	A0
05B0	06	20	A3	06	B1	ED	8C	1A	06	8C	1B	06	8C	1A	07	8C
05C0	19	07	20	A8	06	8C	1E	07	98	8D	16	07	CE	16	07	88
05D0	B1	ED	99	19	06	99	18	07	98	D0	F4	E6	ED	D0	02	E6
05E0	EE	CE	1E	07	D0	F5	AD	18	07	29	0F	D0	13	AD	18	07
05F0	C9	20	F0	29	C9	40	F0	2E	C9	60	F0	2E	29	10	D0	62
0600	AD	18	07	C9	4C	F0	2C	C9	6C	F0	3D	AE	1D	07	AC	1C
0610	07	AD	20	07	48	AD	18	07	28	8D	2C	02	00	A5	ED	48
0620	A5	EE	48	4C	33	06	68	8D	20	07	68	85	EE	68	85	ED
0630	4C	3D	06	AD	1A	06	85	ED	AD	18	06	85	EE	A9	00	8D
0640	19	06	20	9A	06	4C	08	06	AD	1A	06	85	ED	AD	18	06
0650	85	EE	A0	00	B1	ED	AA	C8	B1	ED	85	EE	8A	85	ED	4C
0660	3D	06	AD	20	07	48	AD	18	07	8D	6D	06	28	F0	03	4C
0670	82	06	58	D8	AD	1A	06	30	11	18	65	ED	85	ED	90	02
0680	E6	EE	A9	00	8D	1A	06	4C	00	06	18	65	ED	85	ED	B0
0690	F1	C6	EE	90	ED	A9	0D	20	A5	06	A9	0A	20	A5	06	60
06A0	4C	AC	9A	A9	20	4C	EE	FF	A0	01	C9	00	F0	1A	C9	40
06B0	F0	16	C9	60	F0	12	A0	03	C9	20	F0	0C	29	1F	C9	19
06C0	F0	06	29	0F	AA	BC	03	07	8C	21	07	60	36	35	30	32
06D0	20	2D	20	54	52	41	43	45	52	0D	0A	41	44	52	2E	20
06E0	2D	49	4E	53	54	52	2E	2D	20	3A	41	20	3A	59	20	3A
06F0	58	20	4E	56	2D	2D	44	49	5A	43	20	53	54	41	43	4E
0700	20	0D	0A	02	02	02	01	02	02	02	01	01	02	01	01	03
0710	03	03	03	00	EC	00	F8	00	8D	2C	02	66	00	00	00	00
0720	30	03	00	24	24	24	24	24	24	24	24	24	24	24	24	24

NOVEMBER 1984

SCR# 1

```

0 ( LADEN EN LOSSEN ZIE FD 19 E.V. OSI-POEL)
1 HEX 1 VARIABLE FOUT : WACHT 5000 0 DO LOOP ;
2 : PRT 1 205 C! : : STP 0 205 C! : ( EE39)
3 : GOED!! 1 FOUT ! : ( WEET ZEKER: FOUT=1)
4 : CIJFER? KEY DUP EMIT CR 30 - ;
5 : FOUT? IF ELSE 0 FOUT ! THEN ;
6 ( MET MIJN 32 KRAM HEB IK 16 SCHERMEN FD 9)
7 : I/O-FOUT? DUP 0 > FOUT? 2DUP + 10 < FOUT?
8   OVER OF < FOUT? OVER 0 > FOUT?
9   FOUT @ IF CR ELSE 24 ERROR THEN ;
10 : VRAAG GOED!! ." AANTAL SCREENS ? " CIJFER?
11   ." VANAF SCREENNUMMER ? " CIJFER? I/O-FOUT? ;
12 : BEWAAR VRAAG PRT WACHT ." 6" WACHT
13   400 * 4000 + SWAP 400 * TYPE WACHT STP ;
14 : LAAD VRAAG 400 * 4000 + SWAP 4 * T>
15   ." GELADEN " CR : ;S

```

SCR# 2

```

0 ( HOE VOEG IK DEFINITIES AAN FORTH TOE)
1 ( VERKLAAR ZE EERST ALS ZIJNDE FORTH-LID)
2 FORTH DEFINITIONS DECIMAL
3 ( HAAL DE SCREEN BINNEN)
4 ( BIJVOORBEELD 1 LOAD )
5 FORTH DEFINITIONS DECIMAL
6 ( GA DE SIESTEEM-POINTERS AANPASSEN)
7 LATEST 12 +ORIGIN ! ( TOP NFA)
8 HERE 28 +ORIGIN ! ( FENCE)
9 HERE 30 +ORIGIN ! ( DP)
10 ? FORTH 6 + 32 +ORIGIN ! ( VOC-LINK)
11 HERE FENCE !
12 ( HET EINDE VAN FORTH IS TE VINDEN DOOR)
13 ( HEX HERE . WAARNA MET EEN CHECKSUM-SAVER HET HELE)
14 ( GEHEUGENGEBIED VANAF $0280 TOT HET EINDE GESAVED KAN WORDEN)
15 :S

```

SCR# 3

```

0 ( SMART VLIST VIJGEBLAD #1, CHOOSE)
1 DECIMAL
2 : FIND.LAST.NFA CONTEXT @ @ ;
3 : PRINT.NAME DUP . DUP ID. ;
4 : NEXT.NFA PFA LFA @ ;
5 : READY? DUP 0= ?TERMINAL OR ;
6 : HTAB? 553 C@ ;
7 : CR? HTAB? 32 < IF 32 HTAB? SPACES
8   ELSE CR ENDIF ; ( 2 KOLOMMEN OP SCHERM)
9 : VLIST HEX FIND.LAST.NFA CR
10   BEGIN PRINT.NAME CR? NEXT.NFA READY?
11   UNTIL DROP DECIMAL ;
12 0 VARIABLE RND HERE RND ! ( ZIE EE37 OSI-POEL)
13 : RANDOM RND @ 31421 * 6927 + DUP RND ! ;
14 : CHOOSE RANDOM U* SWAP DROP ;
15 :S

```

SCR# 4

```

0 ( PRINT DE STACK EN HDUMP)
1 HEX : DEPTH 9E SP@ - 1 - 2 / ;
2 : .STACK DEPTH ." >> " IF SP@ 2 - 9E
3   DO I @ . -2 +LOOP ELSE ." STACK LEEG !"
4   ENDIF CR ;
5 ( UIT FORTH-DIMENSIONS)
6 : DUMP 0 DO DUP I + C@ 3 .R LOOP DROP ;
7 : ?NON-PRINTING DUP 20 < SWAP 7E > OR ;
8 : &TYPE 0 DO DUP I + C@ 7F AND DUP ?NON-PRINTING
9   IF DROP 2E THEN EMIT LOOP DROP ;
10 : A-LINE CR DUP 0 6 D.R SPACE 8 OVER OVER
11   DUMP 2 SPACES &TYPE ;
12 : HDUMP 0 DO DUP I + A-LINE 8 +LOOP DROP ;
13 ( GEBRUIK: STARTADRES AANTAL-BYTES HDUMP)
14
15 ;S

```

SCR# 5 .

```

0 ( FORTH DECOMPILER 6502 VIJGEBLAD #2)
1 -FIND : DROP DROP CFA @ CONSTANT COLAD
2 -FIND ;S DROP DROP CFA CONSTANT SEMIAD
3 -FIND CLIT DROP DROP CFA CONSTANT CLITAD
4 -FIND LIT DROP DROP CFA CONSTANT LITAD
5 -FIND OBRANCH DROP DROP CFA CONSTANT OBRANAD
6 -FIND BRANCH DROP DROP CFA CONSTANT BRANAD
7 -FIND (LOOP) DROP DROP CFA CONSTANT FLOAD
8 -FIND (." ) DROP DROP CFA CONSTANT DOTGAD
9 -FIND VARIABLE DROP DROP 4 + CONSTANT VARAD
10 -FIND CONSTANT DROP DROP 8 + CONSTANT CONAD
11 -FIND USER DROP DROP 4 + CONSTANT USAD
12 HEX 0 VARIABLE CURAD 0 VARIABLE ALIAD
13 : ALI+ 2 ALIAD +' 40 CURAD ! ;
14 : ALI- -2 ALIAD +' 40 CURAD ! ;
15 1 . -->

```

SCR# 6

```

0 ( VERVOLG DECOMPILER)
1 : NAME 2+ NFA DUP C@ 7F AND 1+ CURAD +'
2   CURAD @ 39 < IF ID.
3   ELSE CR ALIAD @ DUP CURAD ! SPACES ID. THEN ;
4 : DECOMPILER BEGIN DUP @ DUP SEMIAD = IF NAME CR 0
5   ELSE DUP CLITAD = IF DROP 2+ DUP C@ . 1 -
6   ELSE DUP LITAD = IF DROP 2+ DUP @ .
7   ELSE DUP OBRANAD = IF ALI+ NAME 2+
8   ELSE DUP BRANAD = IF ALI- NAME 2+
9   ELSE DUP FLOAD = IF NAME 2+ DUP @ .
10  ELSE DUP DOTGAD = IF CR NAME 2+ DUP C@
11   7F AND OVER 1+ OVER TYPE 1 - +
12   ELSE NAME
13   THEN THEN THEN THEN THEN THEN 2 + DUP
14   . THEN 0= UNTIL ;
15 2 . -->

```

SCR# 7

```

0 ( DECOMPILER EINDE WERKING: DECOMPIL NAAM )
1 : DECOMPIL -FIND CR
2   IF DROP DUP CFA @ COLAD =
3     IF ." : " DUP NFA ID. CR & ALIAD ! & CURAD !
4     CURAD @ SPACES DECOMPILER
5     ELSE CR DUP CFA @ VARAD =
6       IF DUP ." 0 VARIABLE " NFA ID.
7       ELSE DUP CFA @ CONAD =
8         IF DUP ." 0 CONSTANT " NFA ID.
9         ELSE DUP CFA @ USAD =
10        IF ." 0 USER " DUP NFA ID.
11        ELSE DUP DUP CFA @ =
12        IF ." CODE " DUP NFA ID.
13        THEN THEN THEN THEN THEN
14        ELSE ." ONBEKEND ! " THEN DROP CR ;
15  DECIMAL 3 . ;S

```

SCR# 8

```

0 ( CASE OF ENDOF ENDCASE )
1 : CASE ?COMP CSP @ !CSP 4 ; IMMEDIATE
2 : OF 4 ?PAIRS COMPILE OVER COMPILE = COMPILE
3   OBRANCHE HERE 0 , COMPILE DROP 5 ; IMMEDIATE
4 : ENDOF 5 ?PAIRS COMPILE BRANCH HERE 0 ,
5   SWAP 2 [COMPILE] ENDIF 4 ; IMMEDIATE
6 : ENDCASE 4 ?PAIRS COMPILE DROP BEGIN SF@
7   CSP @ = 0= WHILE 2 [COMPILE] ENDIF REPEAT
8   CSP ! ; IMMEDIATE
9 ( VOORBEELDJE: CASE .. OF <TRUE> ENDOF )
10 ( .. OF <TRUE> ENDOF ETC. ENDCASE )
11 : WELKE? ." NUMMER ?" KEY CASE
12   65 OF ." A" ENDOF 66 OF ." B" ENDOF
13   67 OF ." C" ENDOF 68 OF ." D" ENDOF
14   ." GEEN " ENDCASE CR ;
15   :S

```

SCR# 9

```

0 ( TRACER UIT VYGEBLAD 11 PAG 23)
1 0 VARIABLE ?TRACE 0 VARIABLE #INSFRING
2 : #IN #INSFRING @ SPACES ; HEX
3 ( 1 TRACE ZET DE TRACER AAN EN 0 TRACE UIT)
4 : TRACE DUP ?TRACE ! 0= IF 0 #INSFRING !
5   ." UIT !" ELSE ." AAN !" ENDIF CR ;
6 : ARG.ADDR R> R> DUP >R SWAP >R ;
7 : (:) ?TRACE @ IF CR #IN .STACK ARG.ADDR 2 -
8   NFA ID. ." : ==> ; " 1 #INSFRING +!
9   ?TERMINAL DROP ENDIF ;
10 : (:) ?TRACE @ IF -1 #INSFRING +! .STACK ENDIF ;
11 : W [COMPILE] : COMPILE (:) ; IMMEDIATE
12 : 0 COMPILE (:) [COMPILE] ; IMMEDIATE
13 3A 80 + ? W NFA 1+ C! ( STOP : IN X NAAMVELD)
14 3B 80 + ? 0 NFA 1+ C! ( DOEK ; BIJ Y)
15 :S

```

SCR # 58

```
0 ( SCREEN PRINTER; print een of een aantal screens naar)
1 ( de printer; in combinatie met de SCREEN EDITOR te )
2 ( gebruiken om brieven te schrijven b.v.)
3 0 VARIABLE SCHERM 0 VARIABLE PAGINA
4 : SCR. 26 EMIT ." SCREEN PRINTER" CR CR CR
5   ." NAAR PRINTER VANAF SCREEN "
6   QUERY INTERPRET SCHERM ! CR CR
7   ." AANTAL SCREENS = " QUERY INTERPRET CR
8   1 517 C! 0 PAGINA !
9   0 DO I SCHERM @ + BLOCK
10      DUP 1024 + SWAP
11      DO CR 10 SPACES I 64 -TRAILING TYPE
12      PAGINA @ 1 + DUP PAGINA !
13      64 = IF 0 PAGINA ! 12 EMIT THEN
14      64 +LOOP
15      LOOP 0 517 C! ; ;S
```

SCR # 59

```
0 ( SCREEN SHOW; om teksten op het video-scherm te tonen)
1 ( als een screen gelezen is, moet U op een toets drukken)
2 ( Tekst via SCREEN EDITOR intikken met 'regelnr F tekst)
3 0 VARIABLE SCH
4 : SHOW 26 EMIT ." SCREEN SHOW" CR CR CR
5   ." NAAR VIDEO-SCHERM VANAF SCREEN "
6   QUERY INTERPRET SCH ! CR CR
7   ." AANTAL SCREENS = " QUERY INTERPRET CR
8   26 EMIT
9   0 DO I SCH @ +
10      BLOCK DUP 1024 + SWAP
11      DO I 64 -TRAILING TYPE CR
12      64 +LOOP KEY DROP
13      LOOP ;
14 ;S
15
```



```

1 REM SABR. UIT INTERFACE AGE,MA/ 1979
2 REM DEEL 1
3 PRINTCHR$(26)
10 DIM D(25,4),F9(25),B(25)
11 P$="*****"
14 PRINT:PRINT:PRINTTAB(10)"### SABR ###":PRINT:PRINT:PRINT
15 PRINTTAB(5)"Komen de gegevens van tape (=T)"
16 PRINTTAB(8)"of worden ze via het":PRINTTAB(8)"toetsenbord ";
17 PRINT"ingevoerd (≠K)";:INPUTA$;IFLEFT$(A$,1)="T"THEN900
18 IFLEFT$(A$,1)<>"Y"THEN14
20 PRINT!(28):PRINT:PRINTTAB(15)"Invoer via het toetsenbord":PRINT
25 INPUT"Wat is de naam van gegevensbestand/file";T$
30 PRINT:INPUT"Hoeveel variabelen/reeksen zijn er ";VB:PRINT
33 IFVB<10RVB>4THENPRINT"Let op de dimensionering !!":GOTO30
35 INPUT"Hoeveel waarnemingen per reeks";CS:PRINTT
40 FORI=1TOVB:PRINT
45 FORJ=1TOCS:PRINT"Waarn."J;:INPUTD(J,I)
50 NEXTJ:PRINT:NEXTI
60 GOSUB3000
70 INPUT"Wilt U nog veranderen";N$
80 IFLEFT$(N$,1)="N"ANDA<>12THEN122
81 IFLEFT$(N$,1)="N"ANDA=12THENRETURN
90 PRINT"Geef reeksnr. en waarn.nr.":INPUTV1,C1
100 INPUT"Nieuwe waarde";D(C1,V1)
110 GOTO60
120 PRINT!(28):PRINT
122 PRINT"Welke bewerking wilt u uitgevoerd hebben":PRINT
130 PRINT"1. DESC. Beschrijvende statistiek"
140 PRINT"2. DIST. Frekwentie verdelingen"
150 PRINT"3. TRAN. Variabelen bewerken"
155 PRINT"4. SAVE. Het gegevensbestand op de band zetten"
160 PRINT"5. STOP. De gehele run beëindigen"
170 PRINT"6. WRIT. Het gegevensbestand weergeven"PRINT
175 INPUT"Welk getal;";A
180 IFA<10RA>6THEN120
183 PRINT!(28)
185 DNAGOSUB300,500,1000,800,9999,3000
190 GOTO120
300 REM BEREKENING VAN MIN,MAX,GEM,SIGMA,SIGMA^2,SD,SD^2,COV.
305 S1=0:S2=0:GOSUB3000
310 INPUT"Welke variabele (nummer)";V
315 IFV<10RV>VBTHENGOSUB3200:GOTO310
320 FORI=1TOCS:S1=S1+D(I,V)*D(I,V):S2=S2+D(I,V):NEXTI:M=S2/CS
350 VP=(S1-(S2*S2/CS))/CS:SP=SQR(VP):VS=(S1-(S2*SS/CS))/(CS-1)
360 SD=SQR(VS):CV=SD/M:N=CS-1:MI=D(1,V):MX=D(1,V)
400 FORI=1TON:IFMI>D((I+1),V)THENMI=D((I+1),V)
410 IFMX<D((I+1),V)THENMX=D((I+1),V)
420 NEXTI
430 PRINT:PRINT"Minimum"MI;TAB(24)"Maximum"MX
440 PRINT"Gemiddelde"M:PRINT"Variatie coeff."CV
450 PRINT"Variantie v/d populatie ";VP
455 PRINT"Standaard deviatie v/d populatie"SD
460 PRINT"Variantie v/d steekproef"VS
465 PRINT"Standaard deviatie v/d steekproef"SD
470 PRINT:PRINT"druk op spatiebalk om verder te gaan";
475 POKE530,1:POKE57088,253
480 IFPEEK(57088)=239THENPOKE530,0:RETURN
485 POKE2073,173:GOTO475

```

```

500 REM FREKWENTIE VERDELINGEN
510 FORI=1TOVBSTEP2:PRINTI;L$(I),I+1;L$(I+1):NEXTI
520 INPUT"Welke variabele (nummer)";V
530 IFV<1ORV>VBTHENGOSUB3200:GOTO520
535 FORI=1TOVB:F9(I)=0:NEXTI:INPUT"Hoeveel klassen";N
550 FORI=1TON:PRINT"Bovengrens van klasse";I::INPUTB(I):NEXTI
560 INPUT"Nog veranderingen";A$:IFLEFT$(A$,1)="N"THEN580
570 INPUT"Welk getal";I:INPUT"Waarde";B(I):GOTO560
580 FORI=1TOCS:IFD(I,V)<B(1)THENF9(I)=F9(I)+1
585 FORJ=2TON:IFNOT(D(I,V)<B(J))THEN599
595 IFD(I,V)<B(J-1)THEN599
597 F9(J)=F9(J)+1
599 NEXTJ
600 NEXTI:PRINT::PRINT::PRINT"#### FREKWENTIE VERDELING ####":PRINT
610 PRINT"Bovengrens      Aantal      Aandeel":PRINT
620 FORI=1TON:A=F9(I)/CS:PRINTB(I),F9(I),A:NEXTI
630 PRINT:PRINT"Wilt u:":PRINTSPC(10);"1. stoppen"
633 PRINTSPC(10);"2. histogram"
635 PRINTSPC(10);"3. meer":PRINT:PRINTSPC(15);"1/2/3 ";
640 INPUT:IFA<1ORA>3THENGOSUB3200:GOTO630
645 ONAGOTO647,646,530
646 GOSUB700
647 RETURN
700 PRINTCHR$(26):REM HISTOGRAM SUBR
705 FORI=1TO48:PRINT"#":NEXT:PRINT:PRINT:PRINT
710 PRINTTAB(10)"*** H I S T O G R A M ***":PRINT:PRINT
720 J=1:FORI=1TON:IFF9(I)>JTHENJ=F9(I)
730 NEXTI
760 PRINT"Bovengrens":PRINT:FORI=1TON:N1=F9(I)/J*10:IFN1<1THENN1=1
775 PRINTB(I);TAB(7)+"LEFT$(P$,N1)TAB(42)F9(I):NEXTI
780 GOTO470
800 REM SUBR. OM FILES TE SAVEN
801 REM FORMAAT=#V,#C,IND.LABELS,V1C1,V1C2,ETC
810 PRINT:PRINT:PRINTTAB(10)"Het bestand wordt gesaved.":PRINT:PRINT
812 INPUT"Onder welke naam";T$
815 PRINT:PRINT"Start de recorder, het saveen begint zodra u"
816 PRINT"een willekeurige toets aanslaat":PRINT
817 POKE11,0:POKE12,253:I=USR(0):POKE517,1:PRINTT$:PRINTVB:PRINTCS
825 FORI=1TOVB:PRINTL$(I):FORJ=1TOCS:PRINTD(J,I):NEXTJ,I
835 FORI=1TO3000:NEXTI:POKE517,0
840 PRINT"Het bestand is weggeschreven":FORI=1TO3300:NEXTI:RETURN
900 REM SUBR. OM HET BESTAND IN TE LADEN
905 PRINT:PRINTTAB(10)"Het bestand wordt van band gelezen"
910 PRINT"zodra U de 'RETURN' toets induwt":PRINTT
915 POKE11,0:POKE12,253:I=USR(0):POKE515,1
925 FORI=1TOVB:INPUTL$(I):FORJ=1TOCS:INPUTD(J,I):NEXTJ,I
930 POKE515,0:PRINT:PRINT"Het bestand is ingeladen":PRINT:PRINT
935 PRINT"Bestands-naam ;"T$
940 PRINT"Aantal reeksen ="VB:PRINT"Aantal waarnemingen="CS
950 GOSUB3003
955 PRINT"Type 'ESC' om verder te gaan":WAIT57088,32,254:GOTO70
1000 PRINTCHR$(26):REM BEWERKINGEN SUBR.
1010 PRINT:PRINT:PRINTTAB(10)"## R E E K S B E W E R K I N G E N ##"
1020 PRINT:PRINT:GOSUB2500
1025 INPUT"Welke bewerking";A:PRINT:IFA<1ORA>12THEN1025
1026 IFA=12THEN60

```

```

1030 PRINT"Op welke reeksen (resp. nieuwe reeks waarin het"
1040 INPUT"resultaat wordt gezet en te bewerken reeks");V,V1
1042 IFV>4THENPRINT"Geheugenproblemen !!":RETURN
1044 IFV>VBTHENVB=V:FORI=1TOCS:D(I,V)=0:NEXTI
1048 INPUT"Naam v/d nieuwe reeks";L$(V)
1060 ONAGOSUB1200,1300,1400,1500,2100,2000,1900,2200,1600,1700,1800
1069 FORI=1TO2000:NEXTI
1070 RETURN
1200 PRINT"  Optelling":PRINT:PRINT
1210 GOSUB2320:IFA$="V"THEN1260
1230 GOSUB2300:FORI=1TOCS:D(I,V)=D(I,V1)+P:NEXTI:RETURN
1260 GOSUB2310:FORI=1TOCS:D(I,V)=D(I,V1)+D(I,V2):NEXTI:RETURN
1300 PRINT"  Aftrekking":PRINT:PRINT
1310 GOSUB2320:IFA$="V"THEN1360
1330 GOSUB2300:FORI=1TOCS:D(I,V)=D(I,V1)-P:NEXTI:RETURN
1360 GOSUB2310:FORI=1TOCS:D(I,V)=D(I,V1)-D(I,V2):NEXTI:RETURN
1400 PRINT"  Vermenigvuldiging":PRINT:PRINT
1410 GOSUB2320:IFA$="V"THEN1460
1430 GOSUB2300:FORI=1TOCS:D(I,V)=D(I,V1)*P:NEXTI:RETURN
1460 GOSUB2310:FORI=1TOCS:D(I,V)=D(I,V1)*D(I,V2):NEXTI:RETURN
1500 PRINT"  Deling":PRINT:PRINT
1510 GOSUB2320:IFA$="V"THEN1560
1530 GOSUB2300:FORI=1TOCS:D(I,V)=D(I,V1)/P:NEXTI:RETURN
1560 GOSUB2310:FORI=1TOCS:D(I,V)=D(I,V1)/D(I,V2):NEXTI:RETURN
1600 PRINT"  E-macht verheffing":PRINT:PRINT
1610 FORI=1TOCS:D(I,V)=EXP(D(I,V1)):NEXTI:RETURN
1700 PRINT"  Absolute waarde":PRINT:PRINT
1710 FORI=1TOCS:D(I,V)=ABS(D(I,V1)):NEXTI:RETURN
1800 PRINT"  Omzetting naar een geheel getal":PRINT:PRINT
1810 FORI=1TOCS:D(I,V)=INT(D(I,V1)):NEXTI:RETURN
1900 PRINT"  Worteltrekken (2e machts wortel)":PRINT:PRINT
1910 FORI=1TOCS:IFD(I,V1)>=0THEN D(I,V)=SQR(D(I,V1)):NEXTI:RETURN
1920 PRINT"Bij waarneming"I"wordt de worteltrekking niet"
1930 PRINT"uitgevoerd omdat de waarn. negatief is !!"
1940 D(I,V)=D(I,V1):NEXTI:RETURN
2000 PRINT"  Kwadrateren":PRINT:PRINT
2010 FORI=1TOCS:D(I,V)=D(I,V1)*D(I,V1):NEXTI:RETURN
2100 PRINT"  Machtsverheffing":PRINT:PRINT
2110 INPUT"Tot welke macht";P
2120 FORI=1TOCS:D(I,V)=D(I,V1)^P:NEXTI:RETURN
2200 PRINT"  Natuurlijke logaritme":PRINT:PRINT
2210 FORI=1TOCS:IFD(I,V1)>0THEN D(I,V)=LOG(D(I,V1)):NEXTI:RETURN
2220 PRINT"Bij waarn."I"kon geen nat.log. worden berekend"
2230 PRINT"omdat de waarn. <=0 was":D(I,V)=D(I,V1):NEXTI:RETURN
2300 INPUT"De waarde v/d konstante";P:RETURN
2310 INPUT"Welke variabele (reeksnr.):";V2:RETURN
2320 INPUT"Met een variabele (V), of een konstante (K)";A$:RETURN
2500 PRINT:PRINT"  De beschikbare bewerkingen zijn"
2510 PRINT:PRINT"Optellen",,"= 1":PRINT"Aftrekken",,"= 2"
2520 PRINT"Vermenigvuldigen",,"= 3":PRINT"Delen",,"= 4"
2530 PRINT"Machtsverheffen",,"= 5":PRINT"Kwadrateren",,"= 6"
2540 PRINT"Worteltrekken",,"= 7":PRINT"Logaritme",,"= 8"
2550 PRINT"E-macht",,"= 9":PRINT"Absolute waarde",,"=10"
2560 PRINT"Integer waarde",,"=11":PRINT"Waarneming veranderen",,"=12"
2570 RETURN

```

```
3000 PRINTCHR$(26)
3003 FORI=1TOVB:TB(I)=5+(I-1)*INT(42/VB):NEXTI
3010 FORI=1TOVB:PRINTTAB(TB(I));I;:NEXTI:PRINT""
3020 FORI=1TOVB:PRINTTAB(TB(I));L$(I);:NEXTI:PRINT:PRINT
3030 FORJ=1TOCS:PRINTJ":":FORI=1TOVB:PRINTTAB(TB(I));D(J,I);:NEXTI
3035 PRINT":":NEXTJ:FORI=1TO48:PRINT":":NEXTI:PRINT
3040 IFA=6THENPRINT"Type 'ESC' om verder te gaan":WAIT57088,32,254
3045 RETURN
3200 PRINT" #### Verkeerde opgave ####":RETURN
9999 END
```

```
10 REM TOETS-SCHRIJVER
20 REM JOHN HERMANS SCHIPLUIDEN
30 REM Dit programma maakt gebruik van de ROM MONITOR (superboard II)
35 REM vanuit een programma, dat werkt onder OS65-D.
40 REM Regel 105 bergt de inhoud op van $212 t/m $216
45 REM Regel 110 haalt de toetsindruk op en plaats hem in $217
50 REM Regel 115 herstelt de inhoud van $212 t/m 216
55 REM X=95 --> <RUBOUT>
60 REM X=1 --> <CTRL> A het programma wordt verlaten
65 REM X=13 --> <RETURN> de ingetikte string verschijnt bovenaan.
70 REM De ingetoetste tekst verschijnt onder op het scherm. Als U
75 REM regel 110 aanvult met ?X; ziet U de ASCII-waarde van de
77 REM ingedrukte toets op het scherm verschijnen.
80 P=55050:REM schermadres links-onder
85 POKE13033,24:REM MAAR 24 REGELS OP HET SCHERM
90 REM POKE13033,28 GEEFT WEER HET NORMALE SCHERM
95 PRINT!(28):GOSUB540
100 POKE2073,96:REM disable CTRL C
105 AA=PEEK(530):BB=PEEK(531):CC=PEEK(532):DD=PEEK(533):EE=PEEK(534)
110 DISK!"GO FD00":X=PEEK(531)
115 POKE530,AA:POKE531,BB:POKE532,CC:POKE533,DD:POKE534,EE
125 IFX=95THENGOSUB200:GOTO190
140 IFX=13THENK=I:GOSUB500:GOTO190
145 IFX=1THENPOKE2073,173:POKE13033,28:END
150 IFX<32THEN190
155 A$=A$+CHR$(X)
160 POKEP+I,X:I=I+1:IFI=490RI=113THENI=I+15
165 IFI=480RI=113THENM=16
170 POKEP+I+M,194
175 K=0
185 IFI=113THENI=128
190 POKE2073,173:REM restore CTRL C
195 GOTO100
200 IFI=0THENRETURN:REM <RUBOUT>-hulp
205 I=I-1:X=32:J=1
210 IFI=0THENA$="":GOTO230
215 IFI=630RI=127THENI=I-15:J=16
220 A$=LEFT$(A$,I)
230 POKEP+I,194:POKEP+I+J,32
235 RETURN
500 PRINTA$:A$=""
510 FORJ=0TOK+5
520 POKEP+J,32
530 NEXTJ
540 I=0:POKEP,194
550 RETURN
```

```
10 REM WINDOW, een programma voor Superboard II
15 REM onder OS65-D V 3.3
20 REM Dit programma vult het scherm in blokken
30 REM JUNI 1984
90 PRINT!(28):REM CLS EN ALLE WINDOW CONTROLS OP DEFAULT
100 DIMA$(4,5)
995 REM PRINT!(17,KOLOMNUMMER,RIJNUMMER)
1000 PRINT!(17,0,27);:INPUT"Y VAN BLOK (0-4): ";Y
1005 YY=Y*6:Y1=Y:IFY<0ORY>4THEN1000
1010 PRINT!(17,0,27);:INPUT"X VAN BLOK (0-5): ";X
1015 IFX<0ORX>5THEN1010
1020 XX=X*8:X1=X
1030 U=48-XX:V=28-YY
1040 GOSUB2000
1090 PRINT!(28)
1100 FORY=0TO4
1110 YY=Y*6
1120 FORX=0TO5
1130 XX=X*8
1135 U=48-XX:V=28-YY
1140 GOSUB2130
1150 NEXTX,Y
1160 IFA$(Y1,X1)="STOP"THENEND
1170 PRINT&(0,0):GOTO1000
1999 END
2000 PRINT&(XX,YY);:REM PLAATS CURSOR
2010 PRINT!(22,U,V);:REM BAKEN WINDOW AF
2015 A$(Y,X)=" "
2020 INPUTB$
2030 PRINT&(XX,YY);!(15);B$:REM PLAATS CURSOR EN MAAK REGEL SCHOON
2040 L=LEN(B$)
2050 PRINT&(XX,YY);
2060 FORI=1TOL
2070 PRINT!(33):INPUTC$:REM HAAL KARAKTER OP OP PLAATS CURSOR
2075 A$(Y,X)=A$(Y,X)+C$
2080 PRINT&(XX,YY);
2090 FORJ=1TO1
2100 PRINTCHR$(16);:REM VERPLAATS CURSOR NAAR RECHTS
2110 NEXTJ
2120 NEXTI
2130 PRINT&(XX,YY);
2140 PRINT!(22,U,V);
2150 PRINTA$(Y,X);
2160 RETURN
```

NOVEMBER 1984

LISTER VAN BASIC-PROGRAMMA VARIABELEN.

```

=====
10          ; Het volgende programma komt uit het
20          ; clubblad van de engelse OSI/UK101-
30          ; bezitters. Het geeft op eenvoudige
40          ; wijze alle variabelen en functies
50          ; weer, die Basic tijdens de laatste
55          ; RUN van het programma heeft gebruikt.
60          ; Wanneer een volgende keer uw programma
65          ; in de fout gaat, kunt U het met dit
70          ; machinetaalprogramma in RAM of EPROM
80          ; geheugen de variabelen met de laatste
90          ; waarde listen. Als U het in een EPROM
95          ; stopt, kunt U nog een van de overge-
100         ; bleven letters in Uw Toolkit naar
110         ; het begin van dit ml-programma laten
120         ; laten wijzen. Maar ook in pagina 2
130         ; werkt het prima.
135         ; In dat geval moet U <BREAK> geven
140         ; en .0235G
150         ; Na iedere 15 regels kunt U verder
160         ; door de spatiebalk in te drukken.
170         ; Aan het einde geeft het programma
180         ; een jump naar Basic.
185         ; Als U de routine via de TOOLKIT
190         ; aan wilt spreken, kunt het best
195         ; het ml-programma afsluiten met RTS,
200         ; in plaats van met JMP WARMST (zie
205         ; regel 700, $0269)
210         ; Voeg dan wel twee NOP's toe.
215         ;
220         ;
230         ; LIST BASIC VARIABLES AND
240         ; FUNCTIONS VAN DR. MIKE WHITTLE
250         ; OSI/UK USER GROUP NEWSLETTER
260         ; JUNE 1981
270         ;
280 0230=    START=$0230
290 007B=    VSTL=$7B      ; BEGINADRES VARIABELEN
300 007C=    VSTH=VSTL+1  ; TABEL
310 007D=    VENDL=$7D    ; EINDADRES VARIABELENTABEL
320 007E=    VENDH=VENDL+1
330 0050=    TEMP1=$50    ; HULPADRESSEN
340 0051=    TEMP2=TEMP1+1
350 0052=    TEMP3=TEMP1+2
360 0053=    TEMP4=TEMP1+3
370 00AA=    NAAM  =$AA

```

```

380 0100=          GETAL =#0100
390 FD00=          KEYB  =#FD00
400 FFEE=          PRINT =#FFEE
410 0000=          WARMST=#0000
420 B96E=          TOASC =#B96E
430 ABC3=          PRASC =#ABC3
440                ;
450 0230           *=START
460 0230 A57B      START LDA VSTL          ; BEWAAR DE ADRESSEN
470 0232 8550          STA TEMP1         ; TIJDELIJK
480 0234 A57C          LDA VSTH
490 0236 8551          STA TEMP2
500 0238 A9FD          LDA ##FD          ; SPATIEREGEL TOETSENBORD
510 023A 8D00FD       STA KEYB
520 023D A90F          LABA  LDA #15          ; 15 REGELS PER SCHERM
530 023F 8553          STA TEMP4
540 0241 A90A          LABB  LDA ##0A          ; HAAL LINEFEED
550 0243 20EEFF       JSR PRINT
560 0246 A90D          LDA ##0D          ; HAAL CARRIAGE RETURN
570 0248 20EEFF       JSR PRINT
580 024B C653          BNE LABE          ; VOLGENDE REGELTELLER
590 024D D009          BNE LABE
600 024F A9EF          LDA ##EF          ; SPATIEKOLOM TOETSENBORD
610 0251 CD00FD       LABD  CMP KEYB          ; SPATIE INTOETSEN OM
620 0254 D0FB          BNE LABD          ; VERDER TE GAAN
630 0256 F0E5          BEQ LABA          ; NAAR VOLGENDE SCHERM
640 0258 A550          LABE  LDA TEMP1
650 025A C57D          CMP VENDL
660 025C D009          BNE LABF
670 025E A551          LDA TEMP2
680 0260 C57E          CMP VENDH
690 0262 D003          BNE LABF
700 0264 4C0000       JMP WARMST          ; KLAAR
710 0267 A005          LABF  LDY #5          ; GA 6 BYTES OPHALEN
720 0269 B150          LABG  LDA (TEMP1),Y
730 026B 99AA00       STA NAAM,Y
740 026E 88          DEY
750 026F 10FB          BPL LABG
760 0271 8452          STY TEMP3          ; #FF ALS GETALVLAG
770 0273 A5AA          LDA NAAM
780 0275 100E          BPL LABH
790 0277 48          PHA
800 0278 A946          LDA #'F          ; TYPE 'FN'
810 027A 20EEFF       JSR PRINT
820 027D A94E          LDA #'N
830 027F 20EEFF       JSR PRINT

```



```

840 02B2 68          PLA
850 02B3 E652        INC TEMP3
860 02B5 297F LABH   AND  ##7F
870 02B7 20EEFF     JSR PRINT
880 02BA A5AB        LDA NAAM+1 ; 2E LETTER NAAM
890 02BC D002        BNE LABI
900 02BE A920        LDA #32
910 0290 4B LABI    PHA
920 0291 297F        AND  ##7F
930 0293 20EEFF     JSR PRINT
940 0296 68          PLA
950 0297 1018        BPL LABK
960 0299 A924        LDA #'$
970 029B 20EEFF     JSR PRINT
980 029E E652        INC TEMP3
990 02A0 A920        LDA #32
1000 02A2 20EEFF    JSR PRINT
1010 02A5 A000        LDY #0
1020 02A7 B1AD LABJ  LDA (NAAM+3),Y
1030 02A9 20EEFF    JSR PRINT
1040 02AC CB          INY
1050 02AD C4AC        CPY NAAM+2
1060 02AF 30F6        BMI LABJ
1070 02B1 2452 LABK  BIT TEMP3
1080 02B3 101F BPL LABM
1090 02B5 A920        LDA #32
1100 02B7 20EEFF    JSR PRINT
1110 02BA A5AD        LDA NAAM+3
1120 02BC 4B          PHA
1130 02BD 0980        ORA  ##80
1140 02BF 85AD        STA NAAM+3
1150 02C1 206EB9     JSR TOASC
1160 02C4 68          PLA
1170 02C5 1008        BPL LABL
1180 02C7 A92D        LDA #'-
1190 02C9 20EEFF    JSR PRINT
1200 02CC 8D0001     STA GETAL
1210 02CF A900 LABL  LDA #0
1220 02D1 20C3AB     JSR PRASC
1230 02D4 1B LABM   CLC
1240 02D5 A550        LDA TEMP1
1250 02D7 6906        ADC #6
1260 02D9 8550        STA TEMP1
1270 02DB 9002        BCC LABN
1280 02DD E651        INC TEMP2
1290 02DF 4C4102 LABN JMP LABB

```

OSI FOEL

NOVEMBER 1984

## **Ledenadministratie**

Hierbij geef ik mij op als lid van de HCC

Ik ben reeds lid  
code lid nr

le twee letters van uw achternaam

Wilt u de aangekruiste wijziging aanbrengen.

Naam en voornaam:

Straat en huisnummer:

Postcode

Woonplaats

Telefoonnummer

Ik vraag indeling bij afdeling / gebruikersgroep / werkgroep:

Ik bezit volgend systeem:

Ik wens niet voor te komen op mailinglijsten.

Handtekening:

In envelop zenden naar:

**Hobby Computer Club  
Leden Administratie  
Prinsenhof 11  
2641 RN Pijnacker**

Aanmelding Nieuw lid

Adreswijziging

Wijziging Afdeling

Wijziging GG/WG

