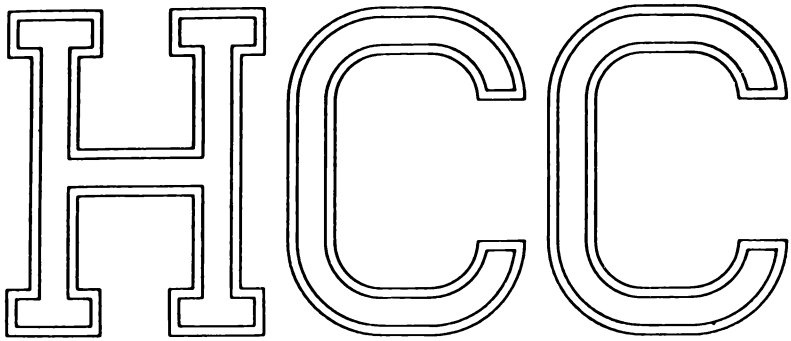

OSI Gebruikersgroep

1e OSI-UG 50 P-Boek

INHOUD

OSI - UG 50 P-boek
een uitgave van Hobby Computer Club Nederland
Postbus 149 - 2250 AC Voorschoten

1e druk - februari 1981 - oplage 200 ex.

V O O R W O O R D .

Dit vijftig-pagina boek is een idee van het voorlopgig bestuur van de OSI-gebruikersgroep.
Het werd verwezenlijkt door de vriendelijke medewerking van het hoofdbestuur van de Hobby Computer Club.
Het werd uitgegeven op de OSI-dag in Utrecht op 14 februari 1981.
De eigenlijke opzet van een vijftig-pagina boek zou kunnen zijn om kopij, die geen plaats vond in een HCC-Nieuwsbrief, verzameld uit te geven. Dat is bij deze uitgave niet het geval.
De samenstelling is ad hoc geschied van beschikbaar en nog niet aan de HCC-Nieuwsbrief of HCC-Software service aangeboden artikelen en programma's. Het gebruikersrapport van Henk Wevers is zelfs al gepubliceerd in een Nieuwsbrief.
De bedoeling van het boek is de beginnende Superboard II-gebruikers wat op weg te helpen met software. Voor de doorgewinterde OSI-gebruiker bevat het weinig nieuws en hardware ontbreekt geheel.
Maar de QUICK SAVE is iets aparts, waar zelfs de grote jongens met respect gebruik van zullen maken.
Er zijn een paar BASIC-programma's opgenomen, die, hoewel voor vermaak bedoeld, U toch enig hoofdbrekens zullen bezorgen, mits U het als een uitdaging ziet eens even niet te programmeren, knutselen of vergaderen, maar de opgaven tot een goed einde te brengen.
Het voorlopige bestuur wenst iedere deelnemer aan de OSI-dag een gezellige en leerzame dag toe.

14 februari 1981
Het voorlopig bestuur:

R. Heesterman	voorzitter
H.J. Kooy	secretaris
J.Burghouts	penningm./software
H. van Coenen	commissaris
J.M.A. Hermans	correspondent

Hans de Jong verzorgde de printings van de listings.

De HCC-Nieuwsbrief.

Artikelen in de HCCN, die van interesse zijn voor OSI-gebruikers.

15: P.Broers: Opzoeken van programma's (11)
16: H.de Jong: Heathkit H-14 aan Challenger 2p (10-12)
16: J.Burghouts: Demonstratie karakterset (12)
16: J.Burghouts: Poke-adressen (13)
16: J.Burghouts: Fruitautomaat (20-23)
16: H.de Jong: OSI-8KBASIC manual (27)
17: P.Broers: Machinetaal op cassette (14)
17: R.Jansen: Random vullen van het scherm; Lissajous; Cirkel (15)
17: H.Wevers: Meer tekens op het scherm (15)
17: J.de Knijff: HARRY (23-26)
18: C.Romijn: Verbeterde grafieken op het beeldscherm (13-14)
18: H.van Coenen: Output-poort voor het O.S.I.-Superboard (15)
20: T.v.d.B.: Output-poort (2) (13)
18: H.Wevers: OHIO Challenger C4PMF (18-22)
19: P.Broers: Superboard Toetsenbord-perikelen (12)
19: K.Romijn: Teletype 33 aan Superboard (13)
20: N.Awater: Weer een tipje van de OSI BASIC sluier... (13)
20: J.Burghouts: Spiralen tekenen (13-14)
21: P.S.Hinderks: Hardware-wijzigingen Superboard en 2P (12-13)
21: J.Burghouts: Functie teken programma (13-14)
21: H.C.de Wal: Toonladders met Superboard (14)
22: J.Burghouts: Cassette-kijker (13-14)
22: A.Hilgersom: Machinetaal op cassette, maar anders (14)
22: H.C.de Wal: Help-Help (14-15)
23: N.Awater: Superboard auto reset (10-11)
23: P.D.H.Broers: Singlist - single step lister (11)

De artikelen-reeks over de BASIC-INTERPRETER in de HHCN-nummers 17, etc.

WAT KAN IK MET MACHINETAAL?

(een vergelijking tussen BASIC en machinetaal)

Vanuit de BASIC is het mogelijk om naar een machine-taal - programma of - subroutine te springen, met behulp van de opdracht USR(). Een uitgebreide uitleg hoe dat werkt, is te vinden in de HCCN (20) van september 1980 in het artikel van Nard Awater.

Het is ook mogelijk zonder tussenkomst van de BASIC rechtstreeks machinetaalprogramma's in het geheugen te typen, te veranderen, van een bandje te laden en te laten lopen. In het nu volgende verhaal komen 3 programma's aan bod, die alle 3 hetzelfde doen,maar steeds op een andere manier; met deze programmaatjes kan de komputer worden gebruikt als eenvoudige typemachine (zonder de hinderlijke SN ERROR ertussen), waarbij de OSI bezitters met een nieuwe monitor-PROM ook de speciale 'CTRL' funkties kunnen gebruiken !!

Het eerste BASIC programma doet het een en ander met behulp van een (1) machinetaal-subroutine, en verder normale BASIC opdrachten.

Het 2e BASIC programma zet een machinetaalprogramma in een vrij stuk geheugen (met POKE opdrachten), en springt dan met een A=USR(A) opdracht naar dat programma toe.

Het derde programma moet met behulp van de monitor rechtstreeks in het geheugen worden getypt maar bevat hetzelfde machinetaalprogrammaatje als het tweede BASIC programma.

Programma 1:

```
10 REM TYPE PROGRAMMA IN BASIC
20 POKE 11,0 : POKE 12,253
30 A=USR(A)
40 PRINT CHR$(PEEK(531));
50 IF PEEK(57100)<>222 THEN 30
60 PRINT"EINDE TYPE-PROGRAMMA"
70 END
```

Wat geheurt er ?

- In regel 20 wordt het startadres van een monitor-subroutine (op $FD00) op de plaats gezet, waar de BASIC USR(A) opdracht naartoe moet gaan met dus op 11 de waarde 0 (de laatste twee getallen van het adres), en op 12 de waarde FD hexadecimaal, ofwel 253 decimaal (de eerste twee getallen van het adres).

De subroutine op $FD00 wacht net zolang totdat een toets wordt ingedrukt, en zet dan de waarde van die toets in de accumulator (=een register/geheugenplaats in de microprocessor) en op geheugenplaats 531 (decimaal).
- In regel 30 springt het programma naar die subroutine.
- In regel 40 wordt de aangeslagen toets op het beeldscherm geprint.
- De kontrole of het programma moet worden beeindigd staat in regel 50, waar wordt gekeken of de 'ESC' toets is aangeslagen. Zoniet dan begint het programma weer van voren af aan, zowel dan stopt het programma met een afmeldingsboodschap.

Programma 2:

```
10 REM TYPE-PROGRAMMA 2
20 FOR A=560 TO 575:READ B:POKE A,B:NEXT
30 POKE 11,48 : POKE 12,2 : A=USR(A)
40 DATA 32,0,253,32,238,255,173,12
50 DATA 223,201,222,208,243,76,0,255
```

Dit programma schrijft een machinetaal programma dat in de DATA statements staat naar het geheugen, beginnend bij geheugenplaats 540,(te berekenen door 2*256+48,zie regel 30 !!).
Het machinetaalprogramma ziet er als volgt uit: (eerst wordt het adres weergegeven, dan de inhoud ervan en van eventueel volgende adressen, en daarna de 'assembler' opdracht).

```
Adres   Inhoud     Opdracht
$0230   20 00 FD   JSR $FD00  START
$0233   20 EE FF   JSR $FFEE
$0236   AD 0C DF   LDA $FD0C
$0239   C9 DF      CMP #$DF
$023B   D0 F3      BNE START
$023D   4C 00 FF   JMP $FF00
```

- Op de eerste regel staat een subroutinesprong (=$20) naar de routine die begint op adres $FD00 dat weer te vinden is in de monitor (P)ROM, de zelfde als in regel 30 van het eerste programma wordt gebruikt.(JSR = Jump to SubRoutine)
- In regel 2 staat weer een monitor-subroutine die de inhoud van de accumulator naar het beeldscherm schrijft, en daarbij let op de regellengte, de 'RETURN' etc.
- In regel 3 wordt de accumulator 'geladen' met de inhoud van adres $DF0C, een geheugenplaats waar de waarde 223 wordt weggezet, als de 'ESC' toets wordt ingedrukt. (Zie voor een prima beschrijving van dit adres de HCCN 19 van augustus 1980, het artikel van Peter Broers op blz.12). (LDA = LoaD Accumulator)

- In regel 4 wordt de inhoud van die accumulator vergeleken met de waarde $DE ofwel 222 decimaal. (CMP # = CoMPare direct)
- In regel 5 wordt terug naar START gesprongen (de eerste regel) als de accumulator niet gelijk is aan $DE. Als dat wel zo is, dan gaat het programma verder met regel 6.
(BNE = Branch if Not Equal)
- Regel 6 tenslotte springt naar het 'BREAK' adres, waar ook naartoe gesprongen wordt als de 'BREAK' toets wordt ingedrukt.(JMP = JuMP)

Bovenstaand progr. kan worden uitgevoerd door simpel 'RUN' en 'RETURN' te typen, maar het is ook mogelijk naar de monitor te gaan (door eerst 'BREAK' en dan 'M'), dan naar adres $0230 (door '.230' te typen) en het progr. te starten door 'G' in te toetsen.

Programma 3:

Is een machinetaalprogramma exakt hetzelfde als dat in progr. 2 , alleen nu rechtstreeks in het geheugen gezet.
Type dan 'BREAK','M','.0230' en '/'; door de '/' wordt de monitor in de DATA-MODE gezet,waarin de inhoud van adressen kan worden bekeken en veranderd. Dat veranderen geheurt door de nieuwe inhoud in 16tallige notatie in te typen, gevolgd door 'RETURN'. Op het beeldscherm verschijnt dan het volgende adres waarvan de inhoud weer veranderd kan worden.
Om progr. 3 in het geheugen te zetten moet dus worden getypt:20 'RETURN' 00 'RETURN' FD 'RETURN' etc. totdat op adres $023F FF staat. ('RETURN' is hier r). Samenvattend ;
type; 'BREAK' M .0230 20 r 00 r FD r 20 r EE r FF r AD r 0C r DF r C9 r DE r D0 r F3 r 4C r 00 r FF r

Kontrole of geen type-fouten zijn gemaakt kan door in de ADRES-MODE te gaan door '.' te typen waarna adressen kunnen worden veranderd (hier dus '.0230'), dan naar de DATA-MODE met een '/' en daarna door telkens 'RETURN' de inhoud te inspekteren en indien nodig te veranderen.
Het programma kan daarna gestart worden door in de ADRES-MODE met het startadres op het beeld scherm een 'G' in te toetsen.(dus '.0230G)

Met een kleine verandering van de laatste 2 programma's is het mogelijk naar de kassette - inhoud te kijken en om alles wat daar binnenkomt alleen naar het beeldscherm te schrijven.
De twee listings zullen voor zichzelf spreken;

Programma 4:

```
10 REM BASIC KASSETTE-KIJKER
20 FOR A=560 TO 575:READ B:POKE A,B:NEXT
30 POKE 11,48 : POKE 12,2 : A=USR(A)
40 DATA 32,128,254,32,238,255,173,12
50 DATA 223,201,222,208,243,76,0,255
```

Programma 5:

```
Adres    Inhoud      Opdracht
$A230    20 80 FE    JSR $FE80   START
$A233    20 EE FF    JSR $FFEE
$A236    AD 0C DF    LDA $DF0C
$A239    C9 DF       CMP #$DF
$A23B    D0 F3       BNE START
$A23D    4C 00 FF    JMP $FF00
```

Jos Ruijhoute
Montessorilaan 112
2625 PR Delft
(tel. 015-565291)

Q U I C K S A V E

De hexdump van de 24 kar.versie volgt hier.
De 48 kar.versie is niet afgedrukt, en komt op band en fotokopie beschikbaar.
Geheel intypen, starten op D1ØØ en saven (met zichzelf).

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
D000  A9 00 85 FB A9 20 20 60 D0 A9 FD 8D 00 DF AD 00
 010  DF C9 EF D0 03 4C 43 FE A2 00 A9 B9 8D 00 F0 A0
 020  00 AD 00 F0 4A 90 FA 2C 00 F0 50 10 A9 74 20 60
 030  D0 A9 B1 8D 00 F0 20 D0 FB 4C 43 FE AD 01 F0 91
 040  F0 A5 F0 C5 F2 D0 11 A5 F1 C5 F3 D0 0B A9 B1 8D
 050  00 F0 20 D0 FB 6C F4 00 E6 F0 D0 C5 E6 F1 D0 C1
 060  A2 00 9D 05 D1 CA D0 FA 60
D100  A0 00 20 DD D1 A0 0D A2 04 20 BE D1 E8 88 20 BE
 110  D1 CA CA CA 88 88 10 F1 A0 40 20 DD D1 A9 E8 8D
 120  D7 D2 A2 0A A9 00 20 B1 FC CA D0 FA A2 03 20 3E
 130  D2 A9 2F 20 B1 FC A2 18 A0 40 B9 05 D3 20 1E D2
 140  C8 CA D0 F6 A2 01 20 3E D2 A9 2F 20 B1 FC A2 69
 150  A0 00 B9 00 D0 20 1E D2 C8 CA D0 F6 A2 05 20 3E
 160  D2 A9 2F 20 B1 FC A9 54 20 B1 FC A0 06 A2 00 B5
 170  F0 20 1E D2 E8 88 D0 F7 A2 01 20 3E D2 A9 47 20
 180  B1 FC A9 B9 8D 00 F0 20 13 D2 B1 F0 20 B1 FC A9
 190  01 4D D7 D2 8D D7 D2 A5 F0 C5 F2 D0 19 A5 F1 C5
 1A0  F3 D0 13 20 13 D2 A9 B1 8D 00 F0 A9 20 8D D7 D2
 1B0  20 D0 FB 4C 00 D1 E6 F0 D0 D0 E6 F1 D0 CC B9 05
 1C0  D3 20 93 FE 30 14 95 F0 88 B9 05 D3 20 93 FE 30
 1D0  09 0A 0A 0A 0A 15 F0 95 F0 60 4C 43 FE A9 B7 99
 1E0  25 D3 A2 00 20 00 FD C9 0D D0 01 60 48 A9 20 99
 1F0  25 D3 68 C9 3C F0 0F E0 18 F0 11 C9 3E F0 03 99
 200  05 D3 E8 C8 D0 06 E0 00 F0 02 CA 88 A9 B7 99 25
 210  D3 D0 D1 A2 00 A0 00 88 D0 FD CA D0 F8 60 48 4A
 220  4A 4A 4A 20 32 D2 68 29 0F 20 32 D2 A9 0D 20 B1
 230  FC 60 09 30 C9 3A 90 02 69 06 20 B1 FC 60 A9 2E
 240  20 B1 FC BD 51 D2 20 1E D2 CA BD 51 D2 20 1E D2
 250  60 00 D0 45 D3 F0 00
```

LOAD-GEDEELTE

SAVE-GEDEELTE
QUICKSAVE

QUICKSAVE

Quicksave kan een stuk geheugen saven. Het saven en loaden met die programma gaat drie maal sneller dan met een hexdump. Dat komt doordat de hex dump voor het saven van één byte geheugen drie bytes op de band zet, bijv 'ØØ' wordt op de band '3Ø 3Ø ØD' (de ascii kodes voor nul en 'RETURN'). Quicksave zet de acht bits van het byte in één keer naar de band. Het is ook ongeveer drie maal sneller dan de checksum-save. Daar ontbreken wel de 'RETURN's, maar er komen ook extra bytes bij.

Quicksave gebruikt de parity-check mogelijkheden van de ACIA. Dat houdt in dan de ACIA met ieder byte een extra bit op de band zet, dat '1' is als het byte een even aantal enen (binair) bevat, anders is dat oneven. (Andersom kan ook.) Bij het loaden kontroleert de ACIA of bit nog wel klopt met het bijbehorende byte. Het programma kan dan reageren met een foutmelding. Er ís een zwakke plek: als er twee fouten in één byte optreden volgt er geen parity-error.

U krijgt de gelegenheid om met het programma een identifikatie mee te saven. Bij het loaden krijgt u deze weer te zien en kunt u voorkomen dat de verkeerde data geload worden.

Quicsave staat in het VIDEORAM en gebruikt daarom alleen zes zeropage-adressen van het "gewone" geheugen. Ieder programma, waar het ook staat, kan dus gesaved en geload worden.

Het programma is voorlopig alleen goed voor de Challenger 1P en superboard. Challengers 2P (en andere waarbij de ACIA niet op adres FØØØ zit) zijn voorlopig geïsoleerd. Er zijn twee versies: de 24 kar. versie en de 48 kar. versie. Saven moet op de bijbehorende machine, loaden kan onderling verwisseld gebeuren.

Gebruik

Loaden: typ de L in, terwijl de komputer in de monitor-mode staat (dus na 'BREAK' M). Als u nu verder niets doet, wordt alles geladen. Eerst verschijnt de identifikatie in beeld. Is het ongewenst, dan kunt u de SPATIEBALK ingedrukt houden, terwijl de komputer verder gaat (met het loaden van het loadprogramma). Op het moment dat de data geload gaan worden (ca 12 sec) springt het programma naar de (warme) start van de monitor. Dus met L begint u weer opnieuw.

Doet u niets, dan begint het "echte" loaden. Het beeld verdwijnt nagenoeg. Oorzaak is een wachtroutine in het videoRAM, waardoor de processor het videogedeelte geen kans geeft dit RAM uit te lezen.

Mocht er een parity-error optreden, dan stopt het loaden direkt, het loadprogramma zet een stel grafkruisjes in het beeld en springt weer naar de monitor. Dus evt. opnieuw proberen door terugspoelen en L typen. Gaat het herhaaldelijk fout dan is er iets mis met de kassette of de audioapparatuur (bv. kop vuil, verkeerde bandsnelheid, enz, tijdens weergave óf opname).

Treedt er geen fout op dan springt het programma naar het bij het saven opgegeven zelfstart-adres.

Saven: -load quicksave, het start zichzelf.
- u moet nu twee regels invullen (48 kar.: staan naast elkaar).
Hierbij hebt u een kursorbesturing: '<' en '>' voor links en rechts.
De eerste regel: hier moeten drie adressen staan:
BBBB EEEE SSSS .
Hierin is BBBB het begin- (eerst te saven) adres,
EEEE het eind- (laatst te saven) adres,
SSSS het zelfstartadres voor de processor wanneer die klaar is met loaden.

Op de plaats van de spaties mag alles staan, als het aantal karakters maar klopt. Als de eerst regel er goed staat, typ dan 'RETURN'. De plaats van de kursor op dat moment doet er niet toe.
- De tweede regel: hier moet (kan) de identifikatie komen. De 24 karakters op deze regel worden bij het loaden weer zichtbaar.
- Start de band en typ 'RETURN' als het saven mag beginnen.
Het saven begint onmiddellijk: het busy-tekentje verschijnt. Terwijl het stil staat worden identifikatie en autoload-(=load-) programma gesaved op de monitor-manier ('hexdump'). Daarna, terwijl het flikkert, worden de 'echte' data gesaved op de quicksave-manier, dus met parity enz.

Als het busy-tekentje verdwijnt is het saven klaar. Het quicksave is dan opnieuw gestart, zodat u wéér kunt gaan saven. Omdat de informatie in het videoRAM onveranderd is gebleven, heeft twee maal 'RETURN' herhaling tot gevolg (handig voor meerdere keren saven van hetzelfde programma).

Om uit deze cirkel te komen kunt u EX 'RETURN' intypen in de eerste regel. De komputer doet dan (u raadt het nooit) de warme monitor. (In feite gebeurt dit altijd als het programma op de plaats van een adres een karakter tegenkomt dat geen hexcijfer is, in dit geval de X.) ('BREAK'en kan natuurlijk ook.)

Het programma zal onherroepelijk verdwalen als beginadres kleiner dan eindadres genomen wordt.

BIJ DE SOURCE-LISTING

Deze is voor de OSI Assembler/Editor. Bijzonderheidjes:'*' staat voor het meelopend adres. Een branch * + iets is 2 minder ver dan verwacht (zie de machinekode). Bij TAB1 wordt een tabel aangelegd van de drie adressen, gebruikt door SAVADR. Het X- register bepaalt welk. Bij veranderingen oppassen: X- en Y-register vaak bepaald. Er zijn branches bij die altijd genomen worden. (bv.die op DØ5E). De listing is te groot om in 8K ineens te worden geassembleerd. Save en load scheiden gaat goed. De routine BELL doet in de originele monitor-versie (zo goed als) niets.
De geprinte versie is voor de 24 kar. versie. De geschrevenwijzigingen maken de 48 kar. versie.

Waarschuwing: het assembleren is niet zo makkelijk als je denkt:

Problemen bij het zetten in het videoRAM!
Grote problemen? Even bellen.

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

SOURCELISTING

QUICKSAVE

```
                ;QUICKSAVE V.24.3
                WARMON=$FE43
                HEXBIN=$FE93
                GETKEY=$FD00
                SAVBYT=$FCB1
                BELL=$FBD0
                CONTRO=$F000
                STATUS=CONTRO
                DATAPO=CONTRO+1
                BASE=$F0
                BAL=BASE
                BAH=BASE+1
                EAL=BASE+2
                EAH=BASE+3
                SAL=BASE+4
                SAH=BASE+5
                START=$D100     48: $D500
                LAUTO=$D000
                REGEL=$D305     48: $D308
                INTER=$40       48: $ 10
                REGLEN=$20      48: $ 40
                CURSOR=REGEL+REGLEN
                LENID=$18
                ERRNEL=$D105    48: $ D308
                BUSY=$D2D7      48: $ D1AB
                LENAUT=$69
                *=START
A000    START   LDY #$00
20DDD1  JSR GETLIN
A00D    LDY #$0D
A204    LDX #$04
20BED1  B1  JSR MOVHEX
E8      INX
88      DEY
20BED1  JSR MOVHEX
CA      DEX
CA      DEX
CA      DEX
88      DEY
88      DEY
10F1    BPL B1
A040    LDY #INTER
20DDD1  JSR GETLIN
A9E8    LDA #$E8
8DD7D2  STA BUSY
A20A    LDX #$0A
A900    LDA #$00
20B1FC  NULL    JSR SAVBYT
CA      DEX
D0FA    BNE NULL
A203    LDX #$03
203ED2  JSR SAVADR
A92F    LDA #'/
20B1FC  JSR SAVBYT
A218    LDX #LENID
A040    LDY #INTER
B905D3  SAVID   LDA REGEL,Y
201ED2  JSR SAVASC
C8      INY
CA      DEX
D0F6    BNE SAVID
D144 A201       SAVAUT  LDX #$01
D146 203ED2     JSR SAVADR
D149 A92F       LDA #'/
D14B 20B1FC     JSR SAVBYT
D14E A269       LDX #LENAUT
D150 A000       LDY #$00
D152 B900D0     LA.1    LDA LAUTO,Y
D155 201ED2     JSR SAVASC
D158 C8         INY
D159 CA         DEX
D15A D0F6       BNE LA.1
D15C A205       LDX #$05
D15F 203ED2     JSR SAVADR
D161 A92F       LDA #'/
D163 20B1FC     JSR SAVBYT
D166 A954       LDA #'T
D168 20B1FC     JSR SAVBYT
D16B A006       LDY #$06
D16D A200       LDX #$00
D16F B5F0       LA.2    LDA BASE,X
D171 201ED2     JSR SAVASC
D174 E8         INX
D175 88         DEY
D176 D0F7       BNE LA.2
D178 A201       LDX #$01
D17A 203ED2     JSR SAVADR
D17D A947       LDA #'G
D17F 20B1FC     JSR SAVBYT
D182 A9B9       LDA #$B9
D184 8D00F0     STA CONTRO
D187 2013D2     JSR WAIT
D18A B1F0       SAVDAT  LDA (BAL),Y
D18C 20B1FC     JSR SAVBYT
D18F A901       LDA #$01
D191 4DD7D2     EOR BUSY
D194 8DD7D2     STA BUSY
D197 A5F0       LDA BAL
D199 C5F2       CMP EAL
D19B D019       BNE LA.4
D19D A5F1       LDA BAH
D19F C5F3       CMP EAH
D1A1 D013       BNE LA.4
D1A3 2013D2     JSR WAIT
D1A6 A9B1       LDA #$B1
D1A8 8D00F0     STA CONTRO
D1AB A920       LDA #$20
D1AD 8DD7D2     STA BUSY
D1B0 20D0FB     JSR BELL
D1B3 4C00D1     JMP START
D1B6 E6F0       LA.4    INC BAL
D1B8 D0D0       BNE SAVDAT
D1BA E6F1       INC BAH
D1BC D0CC       BNE SAVDAT
D1BE B905D3     MOVHEX  LDA REGEL,Y
D1C1 2093FE     JSR HEXBIN
D1C4 3014       BMI LA.7
D1C6 95F0       STA BASE,X
D1C8 88         DEY
D1C9 B905D3     LDA REGEL,Y
D1CC 2093FE     JSR HEXBIN
D1CF 3009       BMI LA.7
D1D1 0A         ASL A
D1D2 0A         ASL A
D1D3 0A         ASL A
D1D4 0A         ASL A
D1D5 15F0       ORA BASE,X
```

```
D1D7 95F0     STA BASE,X
D1D9 60       RTS
D1DA 4C43FE   LA.7 JMP WARMON
D1DD A9B7     GETLIN LDA #$B7
D1DF 9925D3   STA CURSOR,Y
D1E2 A200     LDX #$00
D1E4 2000FD   L1 JSR GETKEY
D1E7 C90D     CMP #$0D
D1E9 D001     BNE *+3
D1EB 60       RTS
D1EC 48       PHA
D1ED A920     LDA #$20
D1EF 9925D3   STA CURSOR,Y
D1F2 68       PLA
D1F3 C93C     CMP #'<
D1F5 F00F     BEQ L2
D1F7 E018     CPX #LENID
D1F9 F011     BEQ L3
D1FB C93E     CMP #'>
D1FD F003     BEQ *+5
D1FF 9905D3   STA REGEL,Y
D202 E8       INX
D203 C8       INY
D204 D006     BNE L3
D206 E000     L2 CPX #$00
D208 F002     BEQ L3
D20A CA       DEX
D20B 88       DEY
D20C A9B7     L3 LDA #$B7
D20E 9925D3   STA CURSOR,Y
D211 D0D1     BNE L1
D213 A200     WAIT LDX #$00
D215 A000     W1 LDY #$00
D217 88       W2 DEY
D218 D0FD     BNE W2
D21A CA       DEX
D21B D0F8     BNE W1
D21D 60       RTS
D21E 48       SAVASC PHA
D21F 4A       LSR A
D220 4A       LSR A
D221 4A       LSR A
D222 4A       LSR A
D223 2032D2   JSR ASCI
D226 68       PLA
D227 290F     AND #$0F
D229 2032D2   JSR ASCI
D22C A90D     LDA #$0D
D22E 20B1FC   JSR SAVBYT
D231 60       RTS
D232 0930     ASCI ORA #$30
D234 C93A     CMP #$3A
D236 9002     BCC *+4
D238 6906     ADC #$06
D23A 20B1FC   JSR SAVBYT
D23D 60       RTS
D23E A92E     SAVADR LDA #'.
D240 20B1FC   JSR SAVBYT
D243 BD51D2   LDA TAB1,X
D246 201ED2   JSR SAVASC
D249 CA       DEX
D24A BD51D2   LDA TAB1,X
D24D 201ED2   JSR SAVASC
D250 60       RTS
D251 00D0     TAB1 .WORD LAUTO,REGEL+INTER,BASE
D253 45D3
D255 F000
```

```
D000          *=LAUTO
D000 A900     LAUTO LDA #$00
D002 85FB     STA $FB
D004 A920     LDA #$20
D006 2060D0   JSR SCREEN
D009 A9FD     LDA #$FD
D00B 8D00DF   STA $DF00
D00E AD00DF   LDA $DF00
D011 C9EF     CMP #$EF
D013 D003     BNE *+5
D015 4C43FE   JMP WARMON
D018 A200     LDX #$00
D01A A9B9     LDA #$B9
D01C 8D00F0   STA CONTRO
D01F A000     LDY #$00
D021 AD00F0   LOADAT LDA STATUS
D024 4A       LSR A
D025 90FA     BCC LOADAT
D027 2C00F0   BIT STATUS
D02A 5010     BVC NOTERR
D02C A974     LDA #'t
D02E 2060D0   JSR SCREEN
D031 A9B1     LDA #$B1
D033 8D00F0   STA CONTRO
D036 20D0FB   JSR BELL
D039 4C43FE   JMP WARMON
D03C AD01F0   NOTERR LDA DATAPO
D03F 91F0     STA (BAL),Y
D041 A5F0     LDA BAL
D043 C5F2     CMP EAL
D045 D011     BNE LA.5
D047 A5F1     LDA BAH
D049 C5F3     CMP EAH
D04B D00B     BNE LA.5
D04D A9B1     LDA #$B1
D04F 8D00F0   STA CONTRO
D052 20D0FB   JSR BELL
D055 6CF400   JMP (SAL)
D058 E6F0     LA.5 INC BAL
D05A D0C5     BNE LOADAT
D05C E6F1     INC BAH
D05E D0C1     BNE LOADAT
D060 A200     SCREEN LDX #$00
D062 9D05D1   STA ERRMEL,X
D065 CA       DEX
D066 D0FA     BNE *-4
D068 60       RTS
D069          END
```

Over microcumputers. (vooral het Superboard II)

Inleiding.

Sinds enkele jaren zijn er betaalbare (gelukkig) microcomputers op de markt. Je kunt daar veel leuke dingen mee doen, als hobby. Er zijn hobbyisten, die niet veel verder komen dan copieëren van programma's, vooral spelletjes. Maar de meesten willen en doen ook meer.

Een hobby bedrijf je vaak niet erg systematisch. Zo weet je soms veel van een detail en heb je maar weinig zicht op het geheel.

Een microcomputer is eigenlijk een erg ingewikkeld stuk speelgoed. En het kost veel tijd om er goed in thuis te raken. Nieuwe kennis moet soms echt bezinken voor je er een praktische toepassing voor ziet.

De bedoeling van dit verhaal is wat te vertellen over het wat en waar van software en het is eigenlijk alleen geschreven voor de nieuwelingen in de hobby. En eigenlijk voor de mensen, die een superboard II (of Challengerlp) hebben. Maar misschien kunnen de oude rotten er ook nog wat in vinden.

De microcomputer.

Het fundament van de computer is de bit: een binair digit, en tevens een klein stukje hardware, dat aan min ligt en dan waarde Ø heeft, of aan plus ligt en dan waarde 1 heeft.

Een computer heeft heel veel bits tot zijn beschikking: een grote computer vele miljarden.

Bits zijn samengevoegd tot bytes. Bij onze microcomputer bestaat een byte uit acht bits. Deze acht bits staan altijd in dezelfde volgorde in de byte: van bit Ø, dat in figuren altijd rechts staat en ook LSB (least significant bit = minst belangrijke bit) genoemd wordt, tot bit 7, in figuren altijd links en ook wel MSB (most significant bit = meest belangrijke bit) genoemd. In een 8-bits-systeem wordt altijd gewerkt met de 8-bits-byte, en past één byte steeds in een geheugenplaats. Elke bit van een byte wordt steeds op de plaats van zijn nummer gezet, ongeacht zijn waarde (Ø of 1).

De microprocessor of chip is de dictator en tevens werkezel van de microcomputer. In ons systeem is dat 6502. Hij heeft naast andere werktuigen vier 8-bits registers (accumulator, index Y, index X, statusregister), een 9-bits register (stack pointer, waarvan bit 8 altijd de waarde 1 heeft, en dus soft ware-matig niet aanwezig is) en een 16-bits programmateller.

Een microprocessor heeft een instructieset meegekregen: bepaalde combinaties van nullen en enen in een byte verzameld dwingen de chip zijn flip-flops al of niet om te gooien: de chip voert een opdracht uit; b.v. $A9 = laad de accumulator met het getal, dat in de volgende byte staat.

In het microprocessor-systeem heeft elke geheugenplaats een eigen nummer,

een adres. Die bytes kunnen deel uit maken van een ROM (read only memory), waarin een binaire code is ingebakken, of van een RAM (random access memory), de inhoud van de byte kan door de chip gewijzigd worden. Bij het uitschakelen van de microcomputer gaat de inhoud van de RAM's verloren, maar niet die van de ROM's.

Bij het inschakelen zal de microprocessor op een willekeurig adres beginnen en mogelijk allerlei instructies uitvoeren, die voor ons geen enkel nut hebben. De break-toets (ook wel reset genoemd) roept hem tot de orde.

Wat doet de chip: hij voert rekenkundige en logische bewerkingen uit aan bytes;
hij transporteert data (= waarde van een byte) tussen zijn registers en geheugenplaatsen en tussen geheugenplaatsen onderling;

Hoe doet hij dat: hij haalt een instructie op in een byte, waarvan het adres staat in de programmateller;
hij verhoogt de waarde van de programmateller met 1;
hij voert de opgehaalde instructie uit;
hij haalt een instructie op .....

De ROM- en RAM-bytes hebben adressen, die eigen zijn aan het systeem.
Voor het Superboard II: zie geheugen-atlas.
De microprocessor 6502, die gebruikt wordt in de Challengers, heeft, net als de andere microprocessoren (6800, 8080, Z80, SC/MP, etc) een 2 bytes-P.C. (programmateller), en kan dus 2^16 (= 65536) adressen bereiken, het laagste adres is $0000 (= 0) en het hoogste adres is $FFFF (= 65535). Het $-teken wordt gebruikt om aan te geven, dat het een getal is in het hexadecimale stelsel (zestientallig).
Bepaalde adressen of bepaalde blokken van het geheugen zijn gereserveerd voor perifere apparatuur (rand-apparaten): toetsenbord, videoscherm, cassette-recorder-uitgang, printer, floppy disc ("disc operating system"), etc.
Voor het gebruik van zo'n rand-apparaat moet meestal het adres van het betreffende apparaat actief worden gemaakt ("enabled") en na het gebruik weer passief worden gemaakt ("disabled"). Zo kan b.v. het toetsenbord worden gebruikt om de uitvoering van een programma te beinvloeden: Bij het draaien ("running") van een BASIC-programma, wordt alleen ctrl-C afgetast. In een BASIC-programma kan het toetsenbord worden geactiveerd met de instructie (Superboard II): POKE 530,1. Met POKE 57088,X wordt rij X van het toetsenbord geactiveerd en met PEEK(57088) wordt vervolgens gekeken of een bepaalde kolom een toets is ingedrukt.
in

Voorbeeld: (voor Superboard II)

```
10 POKE 530,1 : POKE 57088,127 : REM BOVENSTE (RIJ) MATRIX TOETSENBORD
20 Z = PEEK ( 57088 ) : REM IS TOETS INGEDRUKT ?
30 IF Z = 255 THEN POKE 530,0 : GOTO 10 : REM NEE, GA TERUG
40 Z = 8 - LOG ( 255 - Z ) / LOG ( 2 ) : REM WELKE TOETS ?
50 POKE 530,0
60 ON Z GOSUB 1000 , 2000 , 3000 , 4000 , 5000 , 6000 , 7000
```

Dit stukje programma laat de microcomputer wachten op het indrukken van een van de toetsen 1 t/m 7. De ingedrukte toets dirigeert de uitvoering naar het aangegeven adres, alwaar een subroutine wordt uitgevoerd. Deze subroutine kan een heel programma zijn.

B A S I C.

Beginners All purpose Symbolic Instruction Code is omstreeks 1960 ontwikkeld door John Kemeny en Thomas Kurtz . Zij werkten bij het Dartmout College (USA). Hun opzet was een eenvoudige, begrijpbare, probleemgerichte computertaal ter beschikking te hebben, waarmee ook mensen kunnen werken, voor wie het geen dagelijkse bezigheid is met een computer te werken (zij hebben blijkbaar niet gedacht aan ons).

Programmeurs van beroep vinden BASIC geen goede taal. De gebruikte termen zijn niet goed (genoeg) gedefinieerd.

Door de opkomst van de microcomputer kreeg BASIC een kans. De "interpreter" (=vertalerprogramma) hoeft niet groot te zijn, omdat de eisen, die de niet-professionele programmeur aan een computertaal stelt niet zo hoog zijn. Daardoor kan met BASIC een goedkope computer gebouwd worden (weinig geheugen) voor een zeer grote markt.

Misschien wel daardoor, zijn de ontwikkelingen om BASIC wat chaotisch. Diverse software-firma's hebben hun eigen BASIC-versie op de markt gebracht en meestal zijn ze niet "compatible" (d.w.z. een programma, dat ontwikkeld is op het ene merk machine loopt niet op het andere), ook als dezelfde microprocessor wordt gebruikt in de verschillende apparaten (6502 in de Apple, Challenger en PET). De cassette- en diskette-programma's kunnen ook niet geladen worden; zeker niet, als ze niet in de juiste 'baud rate' (= met dezelfde bit-snelheid) op de tape of diskette gezet zijn. Soms zijn zelfs twee machines van dezelfde fabrikant verschillend, (b.v. Challenger1p (= Superboard II) en Challenger2p) al zijn die verschillen vaak erg gering.

De verschillen hebben nog twee andere zeer belangrijke oorzaken:

1. de gebruikte 'tokens' (zie later) verschillen;
2. de adressen van de rand-apparaten verschillen.

Dan zijn er ook nog verschillende soorten BASIC, die op dezelfde machine gebruikt kunnen worden, b.v. integerbasic (geen breuken) en floating point basic (vloeiende komma en dus wel breuken).

Instructies en commando's.

Commando's zijn instructies, die in de "immediate mode" (onmiddelijke verwerking) dus zonder regelnummer worden gebruikt.
Instructies worden zo genoemd, als ze in "deferred mode" (uitgestelde verwerking) dus met regelnummer (in een programma) worden gebruikt.
Voorbeelden: commando: RUN of PRINT 2 + 2 beide gevolgd door "RETURN"
instructie: 10 A = LOG (6)
20 GOTO 40

BASIC-programma's in het geheugen.

Bij het intikken van een programma, of bij het laden van een programma van tape of diskette wordt het programma in het Superboard geladen vanaf $301. Het adres $300 bevat een nul. Elke programmaregel begint in RAM met het adres, waar de volgende regel begint (byte 1 en 2), gevolgd door het regelnummer in hexadecimale notatie (byte 3 en 4). Steeds is het eerste van twee adresbytes het lage 16-tal en het tweede het hoge 16-tal. Regelnummer 1000 wordt dus: C8 03.
Daarna volgende de BASIC-instructies als tokens, afgewisseld met de ASCII-waarden van de gebruikte letters, cijfers, leestekens, etc. (ASCII = American Standard Code for Information Interchange). De regel wordt afgesloten met een nul.
Als na de afsluitende 0 de bytes 1 en 2 allebei een 0 bevatten, is het programma eëindigd. In BASIC van Challenger wordt de uitvoering van het programma dan afgesloten. In sommige BASIC-versies moet de laatste opdracht 'END' zijn. Dat geldt ook voor het einde van een programma, als dat optreedt voor het laatste regelnummer. B.v. als er nog subroutines volgen.
Het volgende korte BASIC-programma geeft een hex-dump van zichzelf.

```
10 P=768:REM $300 = 768
20 RN=PEEK(P+3)+256*PEEK(P+4)
30 PRINTRN;" ";:P=P+5
40 T=PEEK(P):IFT=0THEN80
50 T1=INT(T/16):T2=T-16*T1:IFT1>9THENT1=T1+7
60 IFT2>9THENT2=T2+7
70 PRINTCHR$(T1+48)CHR$(T2+48);" ";:P=P+1:GOTO40
80 IFPEEK(P+1)=0ANDPEEK(P+2)=0THENEND
90 I=I+1:IFI>5THEN110
100 GOTO20
```

```
110 POKE530,1:POKE57088,253
120 IFPEEK(57088)=239THENPOKE530,0:GOTO20
130 POKE530,0:GOTO110
```

De eerste programmaregel ziet er met dit programma als volgt uit:

```
10  50 AB 37 36 38 8E 20 24 33 30 30 20 3D 20 37 36 38
10  P  =  7  6  8  REM  $  3  0  0     =     7  6  8
```

Tokens.

De BASIC-instructies worden dus in het geheugen opgeslagen als tokens.
Zo is het token voor REM 8E en het token voor de bewerking = AB.
Let wel: in een REM-opdracht wordt = 3D.
Met het gegeven programma zijn de tokens gemakkelijk te achterhalen.
Overigens kunt U een lijst van de tokens voor het Superboard vinden in:
H. Wevers: Shorthand Commands for Superboard II and Challenger ClP BASIC's;
Micro The 6502 Journal; mei 1980 pagina 26.
Bij de uitvoering van een BASIC-programma zoekt de BASIC-INTERPRETER aan de hand van de waarde van het token eerst in een tabel het adres van de subroutine op, die bij een bepaald token behoort. Vervolgens voert de INTERPRETER die subroutine uit.

Gereserveerde woorden.

Bij het intikken van een regel in 'immediate mode' wordt die regel opgeborgen in een buffer. Die buffer heeft een lengte, die van machine tot machine veschilt. Bij het Superboard is die lengte 72 bytes. Is de buffer vol, dan neemt de computer het niet meer op in een programmaregel.
Na het indrukken van de returntoets wordt deze regel ge'token'iseerd en achter het programma in het geheugen bijgeschreven (of ertussen geplaats, als het regelnummer lager is dan een regelnummer, dat al in het programma voorkomt) of, in 'direct mode',die regel wordt onmiddelijk uitgevoerd.
Bij het 'token'iseren wordt gelet op bepaalde lettergroepen, zoals: GO TO ON etc.Gebruik daarom nooit variabelen, die met letters van gereserveerde groepen beginnen. De microcomputer let trouwens alleen op de eerste twee letters, behalve na gereserveerde groepen. Zo zal ONNO worden gezien als ON (niet de variabele, maar de instructie).

Lijsten van variabelen.

De waarden van numerieke variabelen worden opgeborgen in bytes, die direct volgen achter het BASIC-programma, in de volgorde, waarin ze in het programma worden verwerkt.
String-variabelen worden geplaatst vanaf de hoogste beschikbare RAM-byte naar beneden, ook weer in de volgorde , waarin ze door het programma worden

aangetroffen.
Array's (getalreeksèn) en stringarray's (karakterreeksen) worden in aparte lijsten opgeborgen.
Voor een ASCII-karakter zijn maar 7 bits nodig. Voor elke variable-naam worden 2 bytes vrijgehouden. Een variabele kan dus een of twee karakters bevatten in de lijst. En er zijn twee bits beschikbaar om het type van de variabele aan te geven. Het MSB van het tweede byte wordt een '1' voor een string. Het MSB van het eerste byte wordt een '1' voor een functie (b.v. DEF FN AB$: A èn B hebben een MSB van '1')
Het startadres van de lijst van enkelvoudige numerieke variabelen is voor het Superboard te vinden op adres $7B en $7C (= 123 en 124) in pagina 0.
(Het hoge byte van een adres is tevens het paginanummer van het adres.
De adressen $0000 tot en met $00FF bevinden zich in pagina 0 (page 0).)
Elke enkelvoudige numerieke variabele wordt opgeborgen in een blok van 6 bytes: 2 bytes voor de naam en 4 bytes voor de waarde.
Als de INTERPRETER een variabele zoekt in de lijst, begint hij op het adres dat hij vindt in pagina 0 (zie boven). Hij vergelijkt de naam van de variabele die hij zoekt, met die in de eerste 2 bytes en springt naar de volgende naam als het niet dezelfde is. Komt hij bij het adres, dat staat in $7F en $80, en heeft hij de naam niet gevonden, dan reserveert hij 6 bytes voor de variabele en verhoogt het adres in $7F en $80.
De 'floating point'-waarde van een numerieke variabele wordt in 4 bytes opgeborgen volgens genormaliseerde, binaire, wetenschappelijke notatie. Dat zit zo:
Noemen we de bytes voor een variabele B0 tot en met B5.
B0 en B1 bevatten de naam van de variable.
B2: MSB is het teken van de macht van 2, waarmee het getal in de bytes B3, B4 en B5 moet worden vermenigvuldigd. MSB is 1 voor een positieve macht. De andere bits geven de waarde van de macht.
B3: MSB geeft het teken van het getal. 0 is positief en 1 is negatief. Er wordt aangenomen, dat de numerieke waarde van MSB altijd 1 is,
Tussen B2 en B3 wordt altijd een punt (of komma) gedacht, die dus staat vóór de binaire waarde van het getal. Om de juiste waarde van het getal te krijgen moet het getal achter de komma worden vermenigvuldigd met de macht van 2 voor de komma.
B4 en B5 vormen samen met B3 een binair getal van 24 cijfers (0 of 1).
Voorbeeld: 10 A = 3.415192 staat in de variabelen-lijst als

0100001 00000000 100000010 01001001 00001111 11010111

A = $2^{2} \times .490FD7_{16}$

Aan de hand van dit voorbeeld kan de lezer zelf wel nagaan wat de hoogste getalswaarde is, die de microcomputer met deze variabelen-lijst kan verwerken.

Hoe arrays en stringvariabelen zijn opgeborgen zou de superboardhobbyist nu zelf moeten kunnen uitzoeken. Het startadres van de arrays staat in $7D en $7E en dat van de strings in $81 en $82.

Disk Operating System.

Een DOS bestaat uit een disk drive, een interface (bij het Superboard bevindt dit zich op het 601-board) en diskettes.
De disk drive wordt via de microcomputer gevoed (voedingsspanning).
Hij bestaat uit een aandrijfmechanisme en een lees/schrijf-mechanisme.
Omdat de lees/schrijfkop erg dicht bij de diskette komt (fractie van een millimeter) is het zaak de diskettes met zeer veel zorg te behandelen.
Stof en smeer zijn erg gauw fataal.
De diskettes gaan zo'n 40 bedrijfsuren mee. Ze worden bij het initialiseren (dat moet, anders kan de computer er niets mee doen) ingedeeld in tracks, en deze weer in sectoren. Meestal wordt een bufferomvang gebruikt van 256 bytes (1/4 Kbyte), zodat dan de omvang van een sector ook 256 bytes is.
Een deel van de opslagruimte van een diskette wordt gebruikt voor geheugenorganisatie (bibliotheek en programmaregisters).
Bij het wegschrijven naar diskette worden 256 bytes opgeslagen in de buffer en van daaruit op de diskette op een vrije sector gezet. Dit gaat zo door, totdat het gehele programma is weggeschreven. Ondertussen wordt door het systeem een track/sector-lijst bijgehouden, waarin wordt opgenomen, welke sectoren in welke volgorde gebruikt zijn. Deze lijst komt ook op de diskette te staan. Tot besluit van wegschrijven worden op een speciaal stuk van een track (de 'directory'-track) weggeschreven: file-naam (=programmanaam), type, lengte en plaats op de diskette.
Bij het teruglezen wordt uiteraard in omgekeerde volgorde gewerkt.
Bij het werken met files is de organisatie iets anders. (Files zijn reeksen van karakters).

'Sequential text-files'.

Bij Stf's, die op diskette worden geschreven, is de lengte niet gedefinieerd (vastgesteld en opgeborgen). Een 'entry' (tot 32767 karakters lang) eindigt met een cariage-return-karakter (13 of $D) en heet 'field'.
Zonder speciale opdracht leest DOS een Stf altijd vanaf het eerste karakter.

'Random access files'.

Bij Raf's wordt wel een lengte-parameter meegegeven. Deze parameter specificeert het aantal bytes (= karakters), die een 'record' maximaal zal bevatten. Een 'record' is een 'field' of een groep 'fields', die door DOS als eenheid wordt (worden) behandeld. Omdat de lengte gespecificeerd is, kan een 'field' worden verandert, zonder de andere 'fields' te veranderen of op te schuiven, hetgeen niet kan met Sft's.

De printer.

De printer is via een interface verbonden met de computer. Na een print-opdracht voor de printer vult de computer een buffer en print vervolgens de inhoud van die buffer uit via de printer.

Rekentijd.

De verschillende 8-bits microcomputers zijn allemaal ongeveer even snel. Er zijn 'bench-mark'-programma's, waarmee de snelheid van een computer kan worden bepaald.

Voorbeeld:

```
300 PRINT"S"
400 FOR K=1 TO 1000
500 NEXT K
600 PRINT"E"
```

```
300 PRINT"S"
400 K = 0
500 K = K + 1
600 IF K  1000 THEN 500
700 PRINT"E"
```

```
300 PRINT"S"
400 K = 0
500 K = K + 1
550 A = K ^ 2
600 B = LOG (K)
650 C = SIN (K)
700 IF K  100 THEN 500
750 PRINT"E"
```

Dit zijn de 'bench-mark'-programma's BM1, BM2 en BM7, die een jaar of drie geleden in een nummer van KILOBAUD hebben gestaan.
Door dit soort programma's kan worden berekend, hoeveel tijd elk van de beschikbare BASIC-instructies vraagt. Een vergelijking met dezelfde instructies van een andere BASIC geeft dan een idee met welke machine of met welke BASIC een bepaald programma het snelst zal lopen. Ook heeft het wel nut te weten wat de tijdsduur van verschillende BASIC-instructies is, b.v. voor 'real time'-simulaties.

ATLAS VAN HET SUPERBOARD II.

```
FF00-FFFF KOUDE START EN BASIC I/O PROM
FE00-FEFF MONITOR PROM
FD00-FDFF POLLED KEYBOARD PROM
F800-FCFF FLOPPY DISC BOOTSTRAP
F000-     ACIA
DF00      TOETSENBORD
D000-D700 VIDEO MEMORY (voor 1 Kbyte video: D000-D300)
A000-BFFF BASIC IN ROM
0222-A000 MOGELIJKE RAM (bij nieuwe MONITOR EPROM: 0230-A000)
0300      BASIC START (kan worden verplaats via vector $79 en $7A;121 en 122)
0000-01FF GERESERVEERDE (WERK-)RUIMTE VAN DE 6502 EN DE BASIC.
```

GEBRUIKERSRAPPORT

O.S.I.

Superboard II en
Challenger IP

door Henk Wevers

Sinds eind 1978 is in de meeste hobbycomputerbladen regelmatig een advertentie van Ohio Scientific Instruments te vinden waarin een erg goedkope computer wordt aangeboden. Deze computer wordt in twee uitvoeringen aangeboden en wel als een grote printplaat, het Superboard II en een uitvoering waarbij dezelfde printplaat samen met een voeding in een kast is opgenomen; hij heet dan Challenger IP. Volgens de advertenties is de machine ontworpen als direkte konkurrent voor de PET 2ØØ1. Nieuwsgierig geworden vroegen we een kennis een Superboard uit Amerika mee te nemen. Door de geweldige belangstelling bleek er bij geen enkele dealer in de staat Californie een Superboard te krijgen. Pas na tussenkomst van de fabriek kon na 14 dagen op het laatste nippertje nog een Challenger IP worden gekocht.
Inclusief invoerrechten kwam deze op ca. f 9ØØ,- zodat we voor een prikje een computer rijker waren. Er werd natuurlijk wel op gegokt dat de zaak zou werken; voor garantie terug naar Amerika sturen zou wat duur worden.
Over de verkrijgbaarheid in Nederland en de prijs later meer. We bespreken nu eerst de ervaringen met deze computer.

Algemene Beschrijving Hardware

Het Superboard heeft een gestabiliseerde voeding nodig van 5 Volt, 3 Ampere. Op de print zijn reeds een zekering en een diode aangebracht om verkeerd aasluiten van deze spanning te beveiligen. Een sporenplan voor de gestabiliseerde voeding is reeds op de print aanwezig, alleen de regel IC moet op een andere plaats (koelplaatje) gemonteerd worden. We zullen in het verdere verslag geen onderscheid meer maken tussen het Superboard en de Challenger. Zoals reeds vermeld is het enige verschil dat de Challenger IP is voorzien van een kast en een voeding.

De Challenger is opgebouwd rondom de 65Ø2 microprocessor, evenals PET, Apple, enz. Ohio was de eerste fabrikant van BASIC in ROM computers die deze keuze maakte. De computer bevat in de geteste uitvoering 4K RAM voor de programma's en een extra 1K RAM voor het videodisplay. Voor de computer is deze laatste 1K RAM direkt toegankelijk, zodat snelle animatie mogelijk is.

Het videodisplay heeft de beschikking over hoofd- en kleine letters, leestekens en daarnaast nog een groot aantal grafische tekens. In totaal zijn er 256 tekens mogelijk welke zijn opgebouwd met een 8x8 puntenraster. De tekens sluiten horizontaal en vertikaal op elkaar aan, zodat een continu beeld van totaal 256 bij 256 punten mogelijk is. Tengevolge van de gebruikelijke overscan bij een TV is de alfanumerieke display begrensd (door middel van software) tot 24x24 tekens. Door wijziging van de TV is het mogelijk totaal 32x32 tekens op het scherm te krijgen.

Om het systeem werkend te krijgen is naast de voeding een video-monitor met hoge impedantie nodig of een HF modulator met TV. De in Europa verkrijgbare uitvoeringen hebben een 22ØV/5ØHz voeding (Challenger) en een Europese video uitgang (5Ø Hz vertikaal). Deze wijzigingen worden door de importeur aangebracht. Een aansluiting voor een cassetterecorder voor opslag van programma's en (eenvoudige) bestanden is aanwezig. Voor wat betreft de door ons aangeschafte uitvoering (Amerikaanse norm);we hebben geen enkel probleem ondervonden om de aanpassing zelf uit te voeren. Een (Philips)TV had geen enkele moeite met de videosignalen, na

instellen van de vertikale synchronisatie (een potmeter achterop) hadden we een uiterst stabiel en plezierig beeld. De 220-110Volt omzetting deden we met een oude voedingstransformator uit het buizentijdperk. De zaak werkte met een UHF modulator van Elektuur van 20 gulden na 10 minuten feilloos.

De grote print ziet er netjes uit, de meeste IC's (vooral MOS) zitten in voetjes en er is ruimte voor extra IC's voor experimenten. Er is een 40 pins IC voet waarop de gehele bus is aangesloten; hierop wordt een eventuele uitbreiding aangesloten.
De geheugen IC's zijn van het type 2114 (low power 450 ns) en er kan zonder meer een extra 4K op de kaart worden geprikt. De voetjes zijn hiervoor aanwezig.
Op de print (bij de Challenger niet op de kast) zijn pluggen aanwezig voor vreugdestokjes (joy-sticks) oftewel stuurknupples voor spelletjes en een aansluiting voor een LF versterker om wat lawaai te maken (anderen noemen dit soms computermuziek). Ook is het mogelijk een printer of externe TTY aan te sluiten, de print is er al op voorbereid.

De print bevat een kompleet toetsenbord van het QWERTY type, dat software-matig wordt uitgecodeerd. Het toetsenbord is prima en wekt de afgunst van menige PET bezitter. Elke toets is voorzien van auto-repeat, d.w.z. als een toets langer dan een halve seconde wordt ingedrukt, er net zolang tekens verschijnen tot de toets weer wordt losgelaten.

Er is een BREAK-toets waarmee de computer **ALTIJD** in een herstartbare situatie komt. Deze toets is verbonden met de Reset-ingang van de microprocessor. Dit is een groot voordeel t.o.v. een aantal andere computers, zoals de PET, waarbij nog wel eens de steker uit het stopkontakt moet worden getrokken als de computer op hol slaat. Dat gebeurt natuurlijk net na het intikken van een lang programma. Bij de Challenger is het mogelijk het programma in zo'n geval te behouden.

De cassette-interface heeft de 300 baud Kansas City standaard en werkt vlekkeloos. Zelfs met de goedkoopste recorders werd geen probleem ondervonden. Alle door ons geprobeerde recorders hadden automatische sterkteregeling bij opname. In tegenstelling tot wat algemeen wordt aangenomen werkte ook dit uitstekend. Vooral de Philips N2210 welke regelmatig voor rond 80 gulden wordt aangeboden, werkt erg handig.

BASIC is aanwezig in vier ROM's van 2K, hoewel de print reeds is voorbereid voor 8K ROM's. Naast BASIC is er nog een vijfde 2K ROM aanwezig voor een machine-taal monitor, een floppy disk bootstrap en besturing, en verdere systeemprogramma's zoals de software voor het toetsenbord.

Software

Na het inschakelen moet eerst op de BREAK toets worden gedrukt. De kosten voor een automatische power up reset (2 weerstanden en een condensator) konden er kennelijk niet af. enfin, echt lastig is het natuurlijk niet. Na het drukken op de BREAK toets verschijnt **altijd** op het scherm de boodschap:

C/C/W/M ?

Door het indrukken van toets D starten we de floppy disk bootstrap. Voor ons systeem heeft dat geen zin, we hebben immers (nog) geen floppy.
We kunnen BASIC starten met C (cold start) of met W (warm start), in het laatste geval blijft het programma in het geheugen ongeschonden.
De machinecode monitor wordt gestart met M.

De laatste twee mogelijkheden (BASIC en monitor) zullen we nu nader bekijken

Monitor

De monitor is erg eenvoudig en beslaat slechts 256 bytes van de 2K system ROM. De volgende mogelijkheden zijn aanwezig:
- het uitlezen van de inhoud van een geheugenplaats

- het wijzigen van de inhoud van een geheugenplaats
- het starten van een machine-taal programma op een gegeven adres
- het laden van een machinetaal programma van de cassetterecorder.

Merkwaardigerwijs is er geen mogelijkheid een machinetaal programma op de cassetterecorder te zetten. Een programma voor dat doel is echter snel geschreven.

Een uitgebreide machine-monitor is op cassette verkrijgbaar voor een 8K systeem. Debugging van programma's is hiermee beter mogelijk. Hoewel de aanwezige monitor eenvoudig is, geeft hij vooral de beginner goede mogelijkheden ervaringen met machinetaal op te doen.

BASIC

De in ROM aanwezige BASIC is een 8K extended versie van de bekende firma Microsoft, die ook de BASIC's voor de PET, TRS-80 level II, MITS, AIM, Apple (Applesoft) en vele andere computers heeft geschreven. Deze BASIC lijkt min of meer de standaard voor de microsystemen te worden.

De versie in de Ohio Scientific computers heeft een nauwkeurigheid van 6 a 7 significante cijfers en is daardoor sneller, maar minder nauwkeurig dan die van de PET. De programmaregel heeft maximaal 72 tekens. Er zijn strings (en LEFT$, MID$, RIGHT$), floating point notatie en gehele getallen mogelijk. De bekende reeks wetenschappelijke functies is aanwezig. De benaming voor de variabelen is wat handiger dan bij de meeste BASIC's, zo is het bijvoorbeeld mogelijk een variabele GEMIDDELDE te noemen. De gehele naam wordt in het programmageheugen opgenomen, hoewel alleen de eerste twee letters het onderscheid met andere variabelen uitmaken. De variabelen GELD en GE zijn voor de machine dus hetzelfde als GEMIDDELDE.
Grafische mogelijkheden zijn er via PEEK en POKE, terwijl ook een koppeling met machinetaal routines via USR mogelijk is.

Variabelen mogen meer-dimensionaal zijn, waarbij het wel uitkijken geblazen is met gedimensioneerde stringvariabelen. Een onhebbelijkheid van deze BASIC is dat als een gedimensioneerde string wordt gewijzigd en de lengte niet meer hetzelfde is als voordien, de interpreter nieuwe ruimte reserveert. Was A$(1) eerst bijvoorbeld "123" en wordt ze "1234", dan verliezen we de geheugenplaatsen voor "123". Op deze manier wil het beschikbare geheugen nog wel eens snel vollopen. Vreemd genoeg werkt na het gebruik van een dergelijke string de functie FRE(X), waarmee de nog vrije geheugenruimte kan worden opgevraagd, niet. De machine slaat dan op hol. Gelukkig is de Challenger te stoppen met BREAK en te herstarten met behoud van programma door W. In normale gevallen wordt een programma en een listing onderbroken met behulp van de controlC toets.

Een koude start van BASIC (C na BREAK) geeft de mogelijkheid een paar parameters te definieren. De machine vraagt eerst de geheugengrootte. Indien alleen de RETURN wordt ingedrukt, wordt alle beschikbare geheugen getest en voor BASIC gebruikt. Het aantal vrije bytes wordt opgegeven. Door het zelf intoetsen van de geheugengrootte is het mogelijk ruimte te reserveren voor machinetaal programma's. BASIC kan ze dan niet vernietigen.
Het is eveneens mogelijk de breedte van een aan te sluiten terminal op te geven. Bij overschrijding van de opgegeven breedte automatisch een CR en LF wordt gegenereerd. Indien de breedte niet wordt opgegeven, wordt deze op 72 gesteld.

Ervaringen bij het Gebruik

De ervaringen na een aantal maanden werken met de Cahllenger zijn zeer positief. Uiteraard zijn er altijd op- en aanmerkingen te maken, maar laten we voorop stellen dat deze Challenger IP resp. het Superboard II de computer is met de beste prijs/prestatieverhouding die we tot nu toe hebben gezien.

Het videodisplay is uitstekend leesbaar, maar de breedte-hoogte verhouding van de tekens doet vreemd aan. Dit komt doordat de tekens oorspronkelijk ontworpen zijn voor een display met 64 tekens per regel (Challenger IIP). Het formaat went snel, maar een schakelaartje om het formaat naar keuze om te zetten van 32x32 naar 16x64 zou erg handig zijn geweest. De benodigde software is hiervoor al in ROM aanwezig. De reden die de fabriek voor het gekozen formaat opgeeft is, dat dan met behulp van een goedkope TV nog een redelijk beeld mogelijk is.

Het toetsenbord werkt prettig, zeker voor wie dat van de PET gewend is, maar het is wel het meest kwetsbare deel van de computer. Als er een toets defect is, moet de hele machine ter reparatie worden aangeboden omdat de zaak op een print is ondergebracht. Laat vooral geen koffie of limonade op het toetsenbord vallen; de zaak is daar niet tegen beveiligd.

Met betrekking tot de software is reeds het een en ander vermeld. We hebben gemerkt dat de RND functie niet geheel juist werkt. Na een koude start moet de functie RND(X) met een andere X steeds een andere reeks toevalsgetallen genereren. Helaas is dat niet het geval. Er wordt steeds dezelfde reeks gegenereerd. Er is wel over dit probleem heen te komen door eerst RND(X) aan te roepen met een negatieve X. Het is echter niet volgens de specificatie. Of dit een Challenger fout is of principieel een Microsoft verschijnsel, weten we niet. Laten andere gebruikers van Microsoft BASIC deze zaak ook eens bekijken.

Documentatie

Bij de computer wordt geleverd:
- een BASIC in ROM manual
- een user manual
- een graphics manual
- een technical review
- een prijslijst voor alle O.S.I. hardware en software cassette's en floppy's).

De Amerikaanse documentatie is vrij volledig, Zo zijn alle elektrische schema's opgenomen. We moesten soms behoorlijk zoeken, maar uiteindelijk bleek de meeste informatie aanwezig. Alleen een afdruk van de machinetaal routines in ROM, vooral de I/O, missen we.
Naast een paar schoonheidsfoutjes, zoals het ontbreken van een mogelijke foutmelding, zat er een bijzonder storende fout in de documentatie. Bij het gebruik van de USR routine staat vermeld dat de startadressen van deze routine in $023E en $023F moeten worden gePOKEd. Dit moet echter $000B en $000C zijn. Zonder deze informatie is het USR commando niet te gebruiken.

Uitbreidingsmogelijkheden

Op de print is 4K extra RAM in te pluggen. Er is een uitbreidingskaart met:
- extra 16K RAM (in totaal dus 24K)
- interface voor twee mini floppy disk eenheden
- printer aansluiting
- verloopkabel naar O.S.I moederbord, waarop alle O.S.I. interface borden passen.

In Amerika bestaat een user-club voor O.S.I. produkten.

Verkrijgbaarheid en Prijzen

Importeur: Ingenieursbureau Koopmans, Papendrecht,telefoon: 078-156033
Prijzen (excl BTW):

	USA:	Nld:
Superboard II	$ 279	f 1060
Challenger IP	$ 349	f 1250
Uitbreiding 4K RAM	$ 69	f 200
Uitbr.kaart incl. 8K RAM	$ 299	f 932
Kompleet systeem Challenger IP, 12K RAM, met een mini floppy		$ 995
Nld: idem, echter met 16K RAM		f 3644

Puzzle-kubus. **(alleen voor 40/64 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 supportprom 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 POKE11,0:POKE12,254:A=USR(A)
1010 POKE540,12:POKE548,2
1020 REM PUZZLE-KUBUS
1030 REM J.HERMANS SCHIPLUIDEN
1040 REM DECEMBER 1980
1050 GOTO1070
1060 A=PEEK(VU(2,0,0)):B=PEEK(VU(2,2,0))
1070 POKEU(2,0,0),B:B=PEEK(VU(2,2,2))
1080 POKEU(2,2,0),B:B=PEEK(VU(2,0,2)):POKEU(2,2,2),B:POKEU(2,0,2),A
1090 A=PEEK(VU(2,0,1)):B=PEEK(VU(2,1,0))
1100 POKEU(2,0,1),B:B=PEEK(VU(2,2,1))
1110 POKEU(2,1,0),B:B=PEEK(VU(2,1,2)):POKEU(2,2,1),B:POKEU(2,1,2),A
1120 A=PEEK(VU(0,0,2)):B=PEEK(VU(3,0,0))
1130 POKEU(0,0,2),B:B=PEEK(VU(4,2,0))
1140 POKEU(3,0,0),B:B=PEEK(VU(1,2,2)):POKEU(4,2,0),B:POKEU(1,2,2),A
1150 A=PEEK(VU(0,1,2)):B=PEEK(VU(3,0,1))
1160 POKEU(0,1,2),B:B=PEEK(VU(4,1,0))
1170 POKEU(3,0,1),B:B=PEEK(VU(1,2,1)):POKEU(4,1,0),B:POKEU(1,2,1),A
1180 A=PEEK(VU(0,2,2)):B=PEEK(VU(3,0,2))
1190 POKEU(0,2,2),B:B=PEEK(VU(4,0,0))
1200 POKEU(3,0,2),B:B=PEEK(VU(1,2,0)):POKEU(4,0,0),B:POKEU(1,2,0),A
1210 IFC>0GOTO1240
1220 PRINT"1";:W=W+1:IFW=5THENW=0:PRINT" ";
1230 FORX=1TO100:NEXTX
1240 RETURN
1250 A=PEEK(VU(1,0,0)):B=PEEK(VU(1,2,0))
1260 POKEU(1,0,0),B:B=PEEK(VU(1,2,2))
1270 POKEU(1,2,0),B:B=PEEK(VU(1,0,2)):POKEU(1,2,2),B:POKEU(1,0,2),A
1280 A=PEEK(VU(1,0,1)):B=PEEK(VU(1,1,0))
1290 POKEU(1,0,1),B:B=PEEK(VU(1,2,1))
1300 POKEU(1,1,0),B:B=PEEK(VU(1,1,2)):POKEU(1,2,1),B:POKEU(1,1,2),A
1310 FORI=0TO2:FORJ=0TO2
1320 A=PEEK(VU(0,I,J)):B=PEEK(VU(2,I,J))
1330 POKEU(0,I,J),B:B=PEEK(VU(4,I,J))
1340 POKEU(2,I,J),B:B=PEEK(VU(5,I,J)):POKEU(4,I,J),B:POKEU(5,I,J),A
1350 NEXTJ:NEXTI
```

```
1360 A=PEEK(V(3,0,0)):B=PEEK(V(3,0,2))
1370 POKEV(3,0,0),B:B=PEEK(V(3,2,2))
1380 POKEV(3,0,2),B:B=PEEK(V(3,2,0)):POKEV(3,2,2),B:POKEV(3,2,0),A
1390 A=PEEK(V(3,0,1)):B=PEEK(V(3,1,2))
1400 POKEV(3,0,1),B:B=PEEK(V(3,2,1))
1410 POKEV(3,1,2),B:B=PEEK(V(3,1,0)):POKEV(3,2,1),B:POKEV(3,1,0),A
1420 IFC>0GOTO1240
1430 PRINT"2";:W=W+1:IFW=5THENW=0:PRINT" ";
1440 FORX=1TO1000:NEXTX
1450 RETURN
1460 A=PEEK(V(0,0,0)):B=PEEK(V(0,0,2))
1470 POKEV(0,0,0),B:B=PEEK(V(0,2,2))
1480 POKEV(0,0,2),B:B=PEEK(V(0,2,0)):POKEV(0,2,2),B:POKEV(0,2,0),A
1490 A=PEEK(V(0,0,1)):B=PEEK(V(0,1,2))
1500 POKEV(0,0,1),B:B=PEEK(V(0,2,1))
1510 POKEV(0,1,2),B:B=PEEK(V(0,1,0)):POKEV(0,2,1),B:POKEV(0,1,0),A
1520 FORJ=0TO2:FORJ=0TO2
1530 A=PEEK(V(3,I,J)):B=PEEK(V(5,2-I,2-J)):POKEV(3,I,J),B
1540 B=PEEK(V(1,I,J)):POKEV(5,2-I,2-J),B:B=PEEK(V(2,I,J))
1550 POKEV(1,I,J),B:POKEV(2,I,J),A
1560 NEXTJ:NEXTI
1570 A=PEEK(V(4,0,0)):B=PEEK(V(4,2,0))
1580 POKEV(4,0,0),B:B=PEEK(V(4,2,2))
1590 POKEV(4,2,0),B:B=PEEK(V(4,0,2)):POKEV(4,2,2),B:POKEV(4,0,2),A
1600 A=PEEK(V(4,0,1)):B=PEEK(V(4,1,0))
1610 POKEV(4,0,1),B:B=PEEK(V(4,2,1))
1620 POKEV(4,1,0),B:B=PEEK(V(4,1,2)):POKEV(4,2,1),B:POKEV(4,1,2),A
1630 IFC>0GOTO1240
1640 PRINT"3";:W=W+1:IFW=5THENW=0:PRINT" ";
1650 FORX=1TO1000:NEXTX
1660 RETURN
1670 PRINT:PRINT:PRINTTAB(11)"PUZZLE-KUBUS":PRINT:PRINT:PRINT
1680 POKE548,3:PRINT:INPUT"GEEF EEN RANDOM-GETAL (ONGELIJK 0):";C
1690 PRINTCHR$(3):P=53516
1700 FORJ=0TO18
1710 FORI=0TO24
1720 IF(J<6ORI>12)AND(J<60RJ>12)THEN1740
1730 Q=P+I+J*64:READR:POKEQ,R
1740 NEXTI
1750 NEXTJ
1760 POKE546,30:PRINT
1770 PRINT"1 = VOORVLAK DRAAIT"
1780 PRINT"    MET WIJZERS VAN":PRINT"    DE KLOK MEE":PRINT
1790 PRINT"2 = BLOK DRAAIT OM":PRINT"    VERTICALE AS":PRINT
1800 PRINT"3 = BLOK DRAAIT OM":PRINT"    HORIZONTALE AS"
1810 PRINT"    // BEELDSCHERM"
1820 PRINT:PRINT"4 = VERLATEN VAN":PRINT"    HET PROGRAMMA"
1830 PRINT:PRINT:PRINT
1840 POKE548,24:POKE549,31:PRINT
1850 DIMV(5,2,2),U(53):I=0:P=53581
```

```
1860 FORK=0TO3
1870 FORL=0TO2
1880 FORM=0TO2
1890 FORN=0TO2
1900 O=P+N*2+K*6+M*128+L*384
1910 IFK=0ANDL=0GOTO1980
1920 IFK=2ANDL=0GOTO1980
1930 IFK=3ANDL=0GOTO1980
1940 IFK=0ANDL=2GOTO1980
1950 IFK=2ANDL=2GOTO1980
1960 IFK=3ANDL=2GOTO1980
1970 U(I)=O:I=I+1
1980 NEXTN
1990 NEXTM
2000 NEXTL
2010 NEXTK
2020 L=0
2030 FORI=0TO5
2040 FORJ=0TO2
2050 FORK=0TO2
2060 U(I,J,K)=U(L):L=L+1
2070 ONI+1GOTO2080,2090,2100,2110,2120,2130
2080 G=32:GOTO2140
2090 G=161:GOTO2140
2100 G=188:GOTO2140
2110 G=219:GOTO2140
2120 G=232:GOTO2140
2130 G=226
2140 POKEU(I,J,K),G
2150 NEXTK
2160 NEXTJ
2170 NEXTI
2180 POKE546,12:PRINT:GOSUB2260
2190 POKE530,1:POKE57088,127
2200 Z=PEEK(57088):IFZ=255THENPOKE530,0:GOTO2190
2210 Z=8-LOG(255-Z)/LOG(2)
2220 IFZ>=5GOTO2200
2230 ONZGOSUB1060,1250,1460,2250
2240 GOTO2190
2250 POKE548,3:PRINT"4":FORX=1TO1000:NEXTX:END
2260 FORII=0TO50
2270 E1=E:E=INT(RND(C)*3+1):IFE1=0THENE1=1
2280 IFE=1AND(E1=2ORE1=3)THENGOSUB1060
2290 IFE=2AND(E1=1ORE1=3)THENGOSUB1250
2300 IFE=3AND(E1=1ORE1=2)THENGOSUB1460
2310 NEXTII
2320 PRINTCHR$(3):C=0
2330 RETURN
```

```
2340 DATA204,148,217,148,217,148,205
2350 DATA149,32,149,32,149,32,149
2360 DATA216,148,219,148,219,148,218
2370 DATA149,32,149,32,149,32,149
2380 DATA216,148,219,148,219,148,218
2390 DATA149,32,149,32,149,32,149
2400 DATA204,148,217,148,217,148
2410 DATA219,148,219,148,219,148,219
2420 DATA148,217,148,217,148,219,148,217,148,217,148,205
2430 DATA149,32,149,32,149,32,149,32,149,32,149,32,149,32,149,32
2440 DATA149,32,149,32,149,32,149,32,149
2450 DATA216,148,219,148,219,148
2460 DATA219,148,219,148,219,148,219,148
2470 DATA219,148,219,148,219,148,219,148,219,148,218
2480 DATA149,32,149,32,149,32,149,32,149,32,149,32,149,32,149,32
2490 DATA149,32,149,32,149,32,149,32,149
2500 DATA216,148,219,148,219,148
2510 DATA219,148,219,148,219,148,219,148
2520 DATA219,148,219,148,219,148,219,148,219,148,218
2530 DATA149,32,149,32,149,32,149,32,149,32,149,32,149,32,149,32
2540 DATA149,32,149,32,149,32,149,32,149
2550 DATA203,148,215,148,215,148
2560 DATA219,148,219,148,219,148,219
2570 DATA148,215,148,215,148,219,148,215,148,215,148,206
2580 DATA149,32,149,32,149,32,149
2590 DATA216,148,219,148,219,148,218
2600 DATA149,32,149,32,149,32,149
2610 DATA216,148,219,148,219,148,218
2620 DATA149,32,149,32,149,32,149
2630 DATA203,148,215,148,215,148,206
OK
```

Voor de Superboard II met 1 KRAM video (24/32 karakters per regel) zal het programma herschreven moeten worden.
Voor een systeem met 2 KRAM video (48/64 karakters per regel), maar zonder de supportprom versies 3.7 up, loopt dit programma na de volgende wijzigingen:
1000, 1010 en 1830 vervallen. 1770 en 1780 samenvoegen tot 1 regel print. 1790: één regel print. 1800, 1810 en 1820 samenvoegen tot 1 regel print.

```
1840 RETURN      1760 GOTO1850     2250 END     2320 C=0
2180 GOSUB 2260
1690 FORX=1TO30:PRINT:NEXTX:P=53516:GOSUB1770
```

1220, 1430 en 1640 laten vervallen of wijzigen in verwijzing naar nieuwe (door U te maken)POKE-subroutine
Voor 542 polled keyboard in 2200 Z=255 wijzigen in Z=0 en regel 2210 wijzigen in 2210 Z=8-LOG(Z)/LOG(2). Tevens in regel 2190 127 veranderen in 128.

```
2180 GOSUB2260
```

Bij het op band zetten van programma's:

type eerst SAVE (RETURN) dan een aantal ?: dan ?"Programmanaam"; CHR$(64):?:?:? en afsluitend :LIST

Start de band en druk de (RETURN) toets in. Allereerst komt netjes de naam van het programma op de band en alles wat je nog meer kwijt wilt, en bovendien wordt bij het inlezen , later, van de band geen SN ERROR gegeven.

Zet nu zodra het programma op de band is gelist nog POKE 515,0 op de band (intypen terwijl de band loopt) gevolgd door RETURN. De komputer stopt dan automaties met laden, zodra het programma is afgelopen. Sommige Superboard versies (n.l. die op 2 MHz lopen en bij het laden een NULL hebben gegeven), moeten eerst een RETURN en dan pas een POKE 515,0 geven.

Door nu de programmazoeker van Peter Broers te gebruiken(zie HCCN 15, 11) wordt alleen het gewenste programma ingeladen.

Instrukties voor Sporen

Twee spelers trekken een spoor door het speelveld, degene die het langst kan blijven doorrijden zonder een spoor (van zichzelf of tegenstander) of zijkant te raken wint.

- Het videobeeld wordt door 4 punten bepaald(regel 290 t/m 320).
- TL=meest linkse en hoogste zichtbare punt v/h videobeeld.
  VB=breedte van het speelveld.
  VH=hoogte van het speelveld.
  RL=standaardregellengte v/d machine (dus 32 of 64).
- Dit programma is de joystick-versie (sticks van Radio-Twenthe uitdekodering van Hans de Jong). Uitdekodering gebeurt in de regels 700 t/m 882 naar 8 richtingen (KB=keyboardplaats 57088). Mensen zonder joystick zullen links en rechts op het toetsenbord 2 maal 4 of 8 toetsen moeten uitdekoderen (zie het graphics manual) en dus de regels 710 t/m 865 moeten herschrijven.
- De snelheid is instelbaar in regel 881 (en 805) door vaker of minder vaak de X-loop te doorlopen.

SPOREN

```
10 REM TRAX,UIT:OHIO SMALL SYSTEMS JOURNAL,PAG.20
20 REM AANGEPAST VOOR HET SUPERBOARD DOOR JOS BURGHOUTS
40 GOSUB 890
50 PRINT" **  S P O R E N  * *  ":PRINT:PRINT
60 INPUT"WIL JE INSTRUKTIES:";IANS$
70 IF IANS$="NEE" THEN 280
80 IF IANS$="JA" THEN 100
90 PRINT:PRINT"TYPE SVP 'JA' OF 'NEE'  ":PRINT:GOTO60
100 PRINT:PRINT:PRINT"BY DIT SPEL IS HET DE BEDOELING ;"
110 PRINT"NERGENS TEGENAAN TE RYDEN, NIET TEGEN DE"
120 PRINT"ZYKANTEN, HET SPOOR VAN JE TEGENSTANDER OF JE"
130 PRINT"EIGEN SPOOR.":PRINT
140 PRINT"SPELER 1 BEGINT LINKS BOVEN IN HET BEELD.":PRINT
150 PRINT"SPELER 2 HELEMAAL RECHTS ONDERAAN MET RYDEN.":PRINT
170 PRINT"DE WAGENS ZYN TE BESTUREN  ;"
180 PRINT"MET TWEE JOYSTICKS , DIE IN DE 8 WINDRICHTINGEN"
190 PRINT"BESTUURD KUNNEN WORDEN.LET WEL OP DAT OOK HET  "
200 PRINT"TERUGKEREN OP JE EIGEN SPOOR TOT VERLROJH&EIDT!"
210 PRINT"ALS JULLIE GEREED ZYN, DRUK DAN DE 'ESC' TOETS "
220 PRINT"IN, DIREKT DAARNA START HET PROGRAMMA":PRINT
230 IF PEEK(57100)<>222THEN 230
280 D1=1:D2=-1
290 VB=48:REM VIDEOBREEDTE
300 BH=25:REM BEELDHOOGTE
310 RL=64:REM REGELLENGTE V/D MACHINE
320 TL=53449:REM STARTPUNT RAND UITERST LINKS BOVEN
330 P1=TL+RL+1:P2=P1+(BH-2)*RL+VB-2
340 S1=P1:S2=P2
360 GOSUB 890
370 KB=57088:DP=BH*RL:BC=161
380 FOR BD=TL TO TL+VB
390 POKE BD,BC:POKE BD+DP,BC:NEXT BD
400 FOR BD=TL TO TL+DP STEP RL
410 POKE BD,BC:POKE BD+VB,BC:NEXT BD
420 GOSUB 700
430 NP=P1+D1
440 IF PEEK(NP)<>32 THEN 510
450 POKE NP,187:P1=NP
460 GOSUB 700
470 NP=P2+D2
480 IF PEEK(NP)<>32 THEN 530
```

```
490 POKE NP,161:P2=NP
500 GOTO 420
510 FP=S2-5
520 GOTO 540
530 FP=S1
540 FOR FL=1TO10
550 POKEFP,73:POKEFP+1,75:POKEFP+2,32:POKEFP+3,87:POKEFP+4,73
560 POKEFP+5,78
570 FOR DM=1TO120
580 NEXT DM
590 FORFH=0TO5
600 POKEFP+FH,32
610 NEXT FH
620 FOR DM=1 TO 120
630 NEXT DM
640 NEXT FL
650 PRINT:PRINT:INPUT"NOG EEN SPELLETJE";IANS$
660 IFLEFT$(IANS$,1)="J" THEN 280
680 GOSUB 890
690 PRINT"EINDE PROGRAMMA":END
700 POKE 530,1
710 POKE KB,127
720 Q=255-PEEK(KB)
722 IFQ>63THENQ=Q-32
724 IFQ>31THENQ=Q-32
726 IFQ<=2THEN800
730 ONQ/2-2GOTO740,800,800,745,750,800,755,800,760,765,770,775
740 D1=-RL:GOTO800
745 D1=1:GOTO800
750 D1=-RL+1:GOTO800
755 D1=-1:GOTO800
760 D1=-RL-1:GOTO800
765 D1=RL:GOTO800
770 D1=RL-1:GOTO800
775 D1=RL+1
800 POKE KB,253
805 FORX=1TO10:NEXTX
810 Q=255-PEEK(KB)
812 IFQ>63THENQ=Q-32
814 IFQ>31THENQ=Q-32
816 IFQ<=2THEN880
820 ONQ/2-2GOTO830,880,880,835,840,880,845,800,850,855,860,865
830 D2=-RL:GOTO880
835 D2=1:GOTO880
840 D2=-RL+1:GOTO880
845 D2=-1:GOTO880
850 D2=-RL-1:GOTO880
855 D2=RL:GOTO880
860 D2=RL-1:GOTO880
865 D2=RL+1
880 POKE530,1
881 FORX=1TO50:NEXTX
882 RETURN
890 POKE11,6:POKE12,254:SC=USR(SC):RETURN
```

YAHTZEE

```
1060 DIMB(5),A(5)
1070 FORI=1TO5:B(I)=0:A(I)=0:NEXTI
1100 PRINTCHR$(3);"WILT U INSTRUKTIES (JA=0,NEE=1)";
1110 INPUTP:IFP=0THENGOSUB4980
1120 PRINT:INPUT"HOEVEEL SPELERS DOEN ER MEE";P
1125 IFP>6THENPRINT"MAXIMAAL 6":GOTO1120
1130 IFP<2THENPRINT"DE KOMPUTER HOUDT ALLEEN DE STAND BY":GOTO1120
1140 DIMN$(P),R(13,P),TG(P),H1(5)
1240 FORI=1TOP:FORJ=1TO13:R(J,I)=-5:NEXTJ:K(I)=0
1270 PRINT:PRINT"WAT IS DE NAAM VAN SPELER";I;:INPUTN$(I):NEXTI
1480 R=0
1490 R=R+1:IFR>13THEN6470
1500 POKE548,3:POKE546,9:PRINTCHR$(3)"RONDE NUMMER"R:POKE548,4
1505 POKE546,9
1540 FORI=1TOP
1542 POKE546,9:POKE547,27:PRINTCHR$(3);:GOSUB6170
1544 POKE546,28:POKE547,56:PRINTCHR$(3);
1546 PRINTN$(I)" IS AAN DE BEURT"
1550 PRINT"DRUK OP 'ESC' OM TE GOOIEN"
1555 PRINT"(OP 'REP' VOOR TOT.OVERZ.)"
1560 IFPEEK(57100)=222THEN1580
1570 IF PEEK(57100)<>126THEN1560
1575 GOSUB7000:GOTO1542
1580 FORL=1TO5:A(L)=INT(6*RND(9)+1):NEXTL
1590 IFPEEK(57100)=222THEN1580
1595 PRINTCHR$(21)CHR$(21)CHR$(5);
1600 GOSUB1610:GOTO1700
1610 G1=0
1620 G1=G1+1:IFG1=6THEN1670
1625 G=0:FORX=1TO5:IFA(X)>GTHENG=A(X):H=X
1630 NEXTX
1640 A(H)=0:H1(G1)=G:GOTO1620
1670 FORX=1TO5:A(X)=H1(X):NEXTX:RETURN
1700 PRINT"JY KRIJGT"A(1)A(2)A(3)A(4)A(5)
1720 INPUT"HOEVEEL STENEN WIL JE VERANDEREN";Z
1730 GOSUB2300
1780 IFZ=0THEN2620
1790 IFZ=5THEN1890
1800 FORS=1TOZ:INPUT"WELKE STEEN";B(S):NEXTS:GOTO1930
1890 PRINT"DRUK OP 'ESC' OM TE GOOIEN"
1900 IFPEEK(57100)<>222THEN1900
1910 FORL=1TO5:A(L)=INT(6*RND(9)+1):NEXTL:IFPEEK(57100)=222THEN1910
1920 GOTO2050
1930 FORL=1TO5:FORL1=1TO5:IFB(L)<>L1THEN1980
1970 A(L1)=0
1980 NEXTL1,L
2000 FORL=1TO5:IFA(L)<>0THEN2040
2030 A(L)=INT(6*RND(9)+1)
2040 NEXTL
2050 GOSUB1610
2140 PRINT"JE KRIJGT"A(1)A(2)A(3)A(4)A(5)
2160 INPUT"HOEVEEL STENEN WIL JE VERANDEREN";Z:GOSUB2300
2220 IF Z=0THEN2620
2230 IF Z=5THEN2360
2270 FORL=1TOZ:INPUT"WELKE STEEN";B(L):NEXTL:GOTO2400
2300 FORX=1TO5:B(X)=0:NEXTX:RETURN
2360 PRINT"DRUK OP 'ESC' OM TE GOOIEN"
2370 IFPEEK(57100)<>222THEN2370
```

```
2380 FORL=1TO5:A(L)=INT(6*RND(9)+1):NEXTL:IFPEEK(57100)=222THEN2380
2390 GOTO2520
2400 FORL=1TO5:FORL1=1TO5:IFB(L)<>L1THEN2450
2440 A(L1)=0
2450 NEXTL1,L
2470 FORX=1TO5:IFA(X)<>0THEN2510
2500 A(X)=INT(6*RND(9)+1)
2510 NEXTX
2520 GOSUB1610

2610 PRINT"JE KRIJGT"A(1)A(2)A(3)A(4)A(5)
2620 PRINT:INPUT"IN WELKE KATAGORIE";Z$
2700 IFZ$="ENEN"THEN2850
2710 IFZ$="TWEEEN"THEN2940
2720 IFZ$="DRIEEN"THEN3030
2730 IFZ$="VIEREN"THEN3120
2740 IFZ$="VYVEN"THEN3210
2750 IFZ$="ZESSEN"THEN3300
2760 IFZ$="DRIE DEZELFDE"THEN3390
2770 IFZ$="VIER DEZELFDE"THEN3440
2780 IFZ$="FULL HOUSE"THEN3520
2790 IFZ$="KLEINE STRAAT"THEN3620
2800 IFZ$="GROTE STRAAT"THEN3660
2810 IFZ$="YAHTZEE"THEN3730
2820 IFZ$="KANS"THEN3820
2830 IFZ$="NUL"THEN3890
2840 PRINTCHR$(21)CHR$(5)CHR$(21);:GOTO2620
2850 IFR(1,I)<>-5THEN3870
2860 R(1,I)=0:FORS=1TO5:IFA(S)<>1THEN2910
2900 R(1,I)=R(1,I)+1
2910 NEXTS:M=R(1,I):GOTO4690
2940 IFR(2,I)<>-5THEN3870
2950 R(2,I)=0:FORS=1TO5:IFA(S)<>2THEN3000
2990 R(2,I)=R(2,I)+2
3000 NEXTS:M=R(2,I):GOTO4690
3030 IFR(3,I)<>-5THEN3870
3040 R(3,I)=0:FORS=1TO5:IFA(S)<>3THEN3090
3080 R(3,I)=R(3,I)+3
3090 NEXTS:M=R(3,I):GOTO4690
3120 IFR(4,I)<>-5THEN3870
3130 R(4,I)=0:FORS=1TO5:IFA(S)<>4THEN3180
3170 R(4,I)=R(4,I)+4
3180 NEXTS:M=R(4,I):GOTO4690
3210 IFR(5,I)<>-5THEN3870
3220 R(5,I)=0:FORS=1TO5:IFA(S)<>5THEN3270
3260 R(5,I)=R(5,I)+5
3270 NEXTS:M=R(5,I):GOTO4690
3300 IFR(6,I)<>-5THEN3870
3310 R(6,I)=0:FORS=1TO5:IFA(S)<>6THEN3360
3350 R(6,I)=R(6,I)+6
3360 NEXTS:M=R(6,I):GOTO4690
3390 IFR(7,I)<>-5THEN3870
3410 R(7,I)=A(1)+A(2)+A(3)+A(4)+A(5):M=R(7,I):GOTO4690
3440 IFR(8,I)<>-5THEN3870
3450 IFA(2)<>A(4)THEN4960
3455 IFA(1)<>A(4)ANDA(2)<>A(5)THEN4960
```

```
3480 R(8,I)=A(1)+A(2)+A(3)+A(4)+A(5):M=R(8,I):GOTO4690
3520 IFR(9,I)<>-5THEN3870
3530 IFA(1)<>A(2)THEN4960
3540 IFA(4)<>A(5)THEN4960
3550 IFA(2)<>A(3)ANDA(3)<>A(4)THEN4960
3560 R(9,I)=25:M=25:GOTO4690
3620 IFR(10,I)<>-5THEN3870
3630 IFA(2)-A(4)<>2ANDA(3)-A(5)<>2ANDA(1)-A(3)<>2THEN4960
3640 R(10,I)=30:M=30:GOTO4690
3660 IFR(11,I)<>-5THEN3870
3670 IFA(1)<>A(5)+4THEN4960
3680 IFA(2)<>A(4)+2THEN4960
3690 IFA(3)<>A(5)+2THEN4960
3700 R(11,I)=40:M=40:GOTO4690
3730 IFR(12,I)<>-5THEN3870
3740 FOR0=1TO5:FOR01=1TO5:IFA(0)<>A(01)THEN 4960
3770 NEXT01,0:R(12,I)=50:M=50:GOTO4690
3820 IFR(13,I)<>-5THEN3870
3830 R(13,I)=A(1)+A(2)+A(3)+A(4)+A(5):M=R(13,I):GOTO4690
3870 GOSUB4650:GOTO2620
3890 INPUT"WAT WIL JE OP NUL ZETTEN";Z$
3970 IFZ$="ENEN"THEN4110
3980 IFZ$="TWEEEN"THEN4150
3990 IFZ$="DRIEEN"THEN4190
4000 IFZ$="VIEREN"THEN4230
4010 IFZ$="VYVEN"THEN4270
4020 IFZ$="ZESSEN"THEN4310
4030 IFZ$="DRIE DEZELFDE"THEN4350
4040 IFZ$="VIER DEZELFDE"THEN4390
4050 IFZ$="FULL HOUSE"THEN4430
4060 IFZ$="KLEINE STRAAT"THEN4470
4070 IFZ$="GROTE STRAAT"THEN4510
4080 IFZ$="YAHTZEE"THEN4550
4090 IFZ$="KANS"THEN4590
4100 GOTO3890
4110 IFR(1,I)<>-5THEN3870
4120 R(1,I)=0:M=0:GOTO4690
4150 IFR(2,I)<>-5THEN3870
4160 R(2,I)=0:M=0:GOTO4690
4190 IFR(3,I)<>-5THEN3870
4200 R(3,I)=0:M=0:GOTO4690
4230 IFR(4,I)<>-5THEN3870
4240 R(4,I)=0:M=0:GOTO4690
4270 IFR(5,I)<>-5THEN3870
4280 R(5,I)=0:M=0:GOTO4690
4310 IFR(6,I)<>-5THEN3870
4320 R(6,I)=0:M=0:GOTO4690
4350 IFR(7,I)<>-5THEN3870
4360 R(7,I)=0:M=0:GOTO4690
4390 IFR(8,I)<>-5THEN3870
4400 R(8,I)=0:M=0:GOTO4690
4430 IFR(9,I)<>-5THEN3870
4440 R(9,I)=0:M=0:GOTO4690
4470 IFR(10,I)<>-5THEN3870
4480 R(10,I)=0:M=0:GOTO4690
4510 IFR(11,I)<>-5THEN3870
4520 R(11,I)=0:M=0:GOTO4690
4550 IFR(12,I)<>-5THEN3870
4560 R(12,I)=0:M=0:GOTO4690
```

```
4590 IFR(13,I)<>-5THEN3870
4600 R(13,I)=0:M=0:GOTO4690
4650 PRINTZ$" HEB JE AL GEBRUIKT !!":RETURN
4690 PRINTN$(I)" DEZE RONDE LEVERT JE"M"OP"
4695 PRINT"OM VERDER TE GAAN 'ESC' INDRUKKEN"
4700 POKE546,9:POKE547,27:PRINTCHR$(2);
4710 GOSUB4740:GOSUB6170:POKE546,28:POKE547,56
4715 IF PEEK(57100)<>222THEN4715
4720 NEXT I
4730 GOTO1490
4740 FORJ=1TOP:IFK(J)<>0THEN4790
4750 FORL=1TO6:IFR(L,J)<>-5THENK(J)=K(J)+R(L,J)
4760 NEXTL
4770 IFK(J)<63THENK(J)=0:GOTO4790
4780 K(J)=35
4790 NEXTJ:RETURN
4960 PRINT"HET IS NIET TOEGESTAAN "Z$" DEZE RONDE TE GEBRUIKEN"
4970 GOTO2620
4980 PRINT"*** INSTRUKTIES VOOR ***":PRINT:PRINT
4990 PRINT"* * * *  Y A H T Z E E  * * * *":PRINT
5000 PRINT"ER MAG DOOR MAX.5 PERSONEN WORDEN GESPEELD."
5010 PRINT"VERDER ZYN DE SPELREGELS GELYK AAN HET BEKENDE"
5020 PRINT"DOBBELSPEL":PRINT"(DOOR GEHEUGENGEBREK HIER NIET VERMELD"
5030 RETURN
6170 PRINT"OVERZICHT "N$(I):PRINT
6180 PRINT"ENEN          ";:IFR(1,I)<0THENPRINT"---":GOTO6190
6185 PRINTR(1,I)
6190 PRINT"TWEEEN        ";:IFR(2,I)<0THENPRINT"---":GOTO6200
6195 PRINTR(2,I)
6200 PRINT"DRIEEN        ";:IFR(3,I)<0THENPRINT"---":GOTO6210
6205 PRINTR(3,I)
6210 PRINT"VIEREN        ";:IFR(4,I)<0THENPRINT"---":GOTO6220
6215 PRINTR(4,I)
6220 PRINT"VYVEN         ";:IFR(5,I)<0THENPRINT"---":GOTO6230
6225 PRINTR(5,I)
6230 PRINT"ZESSEN        ";:IFR(6,I)<0THENPRINT"---":GOTO6240
6235 PRINTR(6,I)
6240 PRINT"DRIE DEZELFDE";:IFR(7,I)<0THENPRINT"---":GOTO6250
6245 PRINTR(7,I)
6250 PRINT"VIER DEZELFDE";:IFR(8,I)<0THENPRINT"---":GOTO6260
6255 PRINTR(8,I)
6260 PRINT"FULL HOUSE    ";:IFR(9,I)<0THENPRINT"---":GOTO6270
6265 PRINTR(9,I)
6270 PRINT"KLEINE STRAAT";:IFR(10,I)<0THENPRINT"---":GOTO6280
6275 PRINTR(10,I)
6280 PRINT"GROTE STRAAT ";:IFR(11,I)<0THENPRINT"---":GOTO6290
6285 PRINTR(11,I)
6290 PRINT"YAHTZEE       ";:IFR(12,I)<0THENPRINT"---":GOTO6300
6295 PRINTR(12,I)
6300 PRINT"KANS          ";:IFR(13,I)<0THENPRINT"---":GOTO6310
6305 PRINTR(13,I)
6310 TG(I)=0:FORL=1TO13:IFR(L,I)>=0THENTG(I)=TG(I)+R(L,I)
6315 NEXTL
6320 PRINT"TOTAAL        ";TG(I)+K(I)
6330 PRINT"(BONUS)       ";K(I)
6400 RETURN
6470 GOTO7000
6475 POKE546,9:POKE548,24:PRINTCHR$(2);
```

```
 6480 PRINT"DIT IS HET EINDE VAN YAHTZEE"
 6490 G=0:FORX=1TOP:IFK(X)+TG(X)>GTHENG=K(X)+TG(X):H=X
 6500 NEXT X
 6510 PRINTN$(H)" HEEFT GEWONNEN MET"K(H)+TG(H)"PUNTEN"
 6540 FORX=1TOP:PRINTN$(X)" HEEFT EEN TOTAAL VAN"K(X)+TG(X):NEXTX
 6550 POKE548,4:END
 7000 POKE11,6:POKE12,254:X=USR(X):POKE546,9:POKE547,56
 7005 PRINTCHR$(2);
 7010 PRINT"TOTAAL OVERZICHT":PRINT:PRINT
 7020 PRINT"ENEN":PRINT"TWEEEN":PRINT"DRIEEN":PRINT"VIEREN"
 7030 PRINT"VYVEN":PRINT"ZESSEN":PRINT"DRIE DEZELFDE"
 7040 PRINT"VIER DEZELFDE":PRINT"FULL HOUSE":PRINT"KLEINE STRAAT"
 7050 PRINT"GROTE STRAAT":PRINT"YAHTZEE":PRINT"KANS":PRINT
 7060 PRINT"TOTAAL":PRINT"(BONUS)      ";:FORX=1TOP
 7090 L=PEEK(553):POKE546,L+6:PRINTCHR$(2):PRINTN$(X):PRINT
 7110 FORA=1TO13:IFR(A,X)<0THENPRINT"---":GOTO7130
 7120 PRINTR(A,X)
 7130 NEXTA:PRINT"----":PRINTTG(X)+K(X):PRINTK(X):NEXTX
 7140 IFR>13THEN6475
 7160 POKE546,9:POKE548,25:PRINTCHR$(2);:INPUT"VERDERGAAN";Z$
 7180 POKE548,4:POKE11,6:POKE12,254:X=USR(X):RETURN
READY
```

Instrukties voor Yahtzee

Dit is een dobbelspel voor maximaal 6 personen, waarbij de komputer alleen de stand bijhoudt en weergeeft (in een scrolling window m.b.v. de nieuwe monitor EPROM van Henk Wevers).
Elke beurt mag de speler 3 keer werpen (zolang de ESC toets is ingedrukt gooit de komputer).
De bedoeling is om stenen te verzamelen, die het beste in de nog overgebleven katagorieen passen, dus de enen in de katagorie ENEN etc. Een worp van 3 vieren een 1 en een 2 wordt dus het beste in de katagorie VIEREN weggezet (als die nog vrij is!), wat in totaal 12 oplevert.

- Full House is de kombinatie met drie dezelfde en twee dezelfde
- Kleine Straat is een kombinatie met 4 opeenvolgende nummers.
- Grote Straat met 5 opeenvolgende nummers.
- Yahtzee de kombinatie met 5 dezelfde stenen.
- Kans is een katagorie waarbij geen eisen worden gesteld aan de stenen, de opbrengst is de optelling van alle 5.

Als geen katagorie gevuld kan worden, (vaak tegen het einde van een spelletje, als kans al is gebruikt),kan de katagorie NUL worden gebruikt. Hiermee is een nog vrije katagorie op 0 te zetten.

Voor vervolg zie pag. 43.

Vluchtsimulator. (alleen voor 2 kRAM video)

Dit programma is gebaseerd op de formules, die ook gebruikt zijn door F.R.Ruckdeschel in zijn artikel:"Microcomputer Flight Simulation" in Byte Book Programming Techniques Volume 2: Simulation. Dat programma is een batch programma.

Vluchtsimulator is geschreven voor een systeem met een supportprom versies 3.7 up (48 karakters per regel). Voor een systeem met een andere monitorrom hebben de poke's op 546 en hoger geen uitwerking. PRINTCHR$(2) kan vervallen. PRINTCHR$(3) moet worden vervangen door een printlus of een andere clear screen..

De plaats waar de getallen worden gepoked, wordt bepaald in regel 3340.

De "decimaal"-streep wordt geschreven met subroutine 1280.

De tekst wordt geprint met de subroutine 2990.

Het programma is een bijna-"real time"-programma. De load-omvang is ca 6200 bytes en de run-omvang is ca 6800 bytes.

Het programma wordt elke 6 seconden helemaal doorgewerkt op een 2 MHz-Superboard.

Het programma simuleert een landing op een strip van 1 mijl lang, die loopt van west naar oost (en omgekeerd). Referentiepunt voor de positiebepaling is het westelijk uiteinde van de strip. De positie wordt gegeven door de verkeerstoren met RANGE (afstand) en POSITION OF RUNWAY (hoek in graden ten opzichte van het oosten gemeten over het noorden). De richting, waarin het vliegtuig zich beweegt, wordt gegeven achter HEADING OFF EAST, waarvoor eenzelfde hoekmeting wordt gebruikt.

De gedragingen van het vliegtuig kunnen worden beïnvloed met de toetsen, die op het scherm te lezen zijn. Ze veranderen: FLAPS (remklappen), TRIM, THRUST (ingeschakeld motorvermogen), BANK (rolhoek, tevens bocht), ATTACK ANGLE (elevatie of declinatie) en LANDING GEAR (landingsgestel). De vliegcyclus kan worden versneld (tot 1 seconden per cyclus) en vertraagd (tot 12 seconden per cyclus).

De andere data zijn: ALTITUDE (vlieghoogte), SPEED (vliegsnelheid t.o.v van de grond), STALL SPEED (minimaal vereiste snelheid), ENGINE TEMP. (motortemp.), FUEL (brandstofvoorraad), CLIMBE RATE (stijgsnelheid), WIND DIRECTION (windrichting) en WIND SPEED (windsnelheid).

Het is de bedoeling het vliegtuig veilig aan de grond te zetten. Daarvoor moet het op de baan landen (60 m breed) met een daalsnelheid, die niet groter is dan 2% van de vliegsnelheid, met het landingsgestel neer. Succes.

```
1000 REM VLUCHT-SIMULATOR
1010 REM
1020 REM
1030 REM
1040 REM BEWERKT DOOR J.M.A.HERNANS
1050 REM
1060 PRINTCHR$(3)
1070 IFQ=1GOTO1090
1080 DIME(19),K(12)
1090 P2=3.141595:P3=3.28:P4=57.28:PU=53673:PV=64:PE=139
1110 P5=1609:P6=.5148:G=9.8
1120 PRINTTAB(12);:PRINT"** VLUCHT-SIMULATOR **":FORI=1TO5:PRINT:NEXT
1160 E1=253:E2=127:E3=253:E4=251:E5=247:E6=239:E7=223:E8=191:EE=57088
1170 E9=127:EF=54436:T3=3:VI=0
1180 PRINT:PRINT:PRINT:PRINT:PRINTTAB(12);
1200 PRINT" * *    LANDING    * *"
1210 FORIU=1TO8:PRINT:NEXTIU:PRINT"VOOR VERVOLG DRUK OP SPATIEBALK"
1230 GOSUB1440 :GOSUB1480
1240 POKE530,1:POKEEE,253:IFPEEK(EE)=239THENPOKE530,0:GOTO1270
1260 POKE530,0:GOSUB2930:GOTO1240
1270 PRINTCHR$(3):GOTO1330
1280 FORI=0TO20:IFI=14GOTO1320
1290 IFI=12GOTO1320
1300 IFI=15GOTO1320
1310 POKEPU+I*PV,PE
1320 NEXTI
1330 GOSUB1420:GOSUB1560:GOSUB2330
1340 IFX3<60RF9<0GOTO3500
1350 GOSUB1620:GOSUB2300:GOSUB2380
1360 IFVI=0THENVI=1:GOSUB2990
1370 GOSUB2280:GOSUB1760:GOSUB2540:T4=T4+T3:GOSUB2720
1410 GOTO1280
1420 V2=V1*SIN(T2)/2/N1:V2=V2*(1+.3*ABS(SIN(B)))
1425 V2=V2*SQR(1.5/(1+F/2)):RETURN
1440 M=5:F9=.2:N1=.4:V1=200:T2=12
1460 M=M*907:V1=V1*P6:T2=T2/P4:F9=F9*907:M9=M:M=M9+F9:RETURN
1480 X5=SQR(2):X1=-8*P5:X2=0:X3=P5*2/3:S1=.65*V1/X5:S2=0
1530 V=.75*V1:S3=0:F=0:T1=0:N3=.55:B=0:T9=280:F1=0:R9=-1:N2=0:D3=0
1540 G1=0:RETURN
1550 FORJ=1TO6:L=L*V/V2:NEXTJ:RETURN
1560 C1=N1*M*G/V1:C2=C1*(V2↑3)/((M*G*COS(T2))↑2)
```

Door als extra regel in te lassen: 2935 RETURN, wordt de landing uitgevoerd bij windstilte.
Voor 542 polled keyboard dienen de volgende wijzigingen aangebracht te worden:
1160 E1=2:E2=128:E3=2:E4=4:E5=8:E6=16:E7=32:E8=64:EE=57088
1170 E9=128:EF=54436:T3=3:YI=0
1240 253 wordt 2 en 239 wordt 16
3610 239 wordt 16
3600 251 wordt 4

```
1570 C3=1.5*M*G*COS(T2)/((V2*V2*(1+80/P4))):RETURN
1580 P1=EXP(-X3/2000):IFV=0THENV=.0000001
1590 L=C3*(1+F/2)*(1+5*T1)*V*V*P1*(1-1/(1+(V-V2)*(V-V2)))
1600 IFV<V2THENGOSUB1110
1610 D=C2*L*L/P1/V/V+C1*V*(1+.6*G1)*P1:GOSUB1420 :RETURN
1620 T3=T3/4:I=0
1630 I=I+1:GOSUB1580 :GOSUB1700 :GOSUB1720 :Z=N1*M*G*COS(T1)*P1*N3
1640 Z=Z-M*G*SIN(D3)-D:Y=L*SIN(B)/COS(D3):W=T3/M/V:S3=V*SIN(D3)
1650 S1=S1+W*(Z*S1-Y*S2):S2=S2+W*(Z*S2+Y*S1):X1=X1+S1*T3:X2=X2+S2*T3
1660 V=SQR(S1*S1+S2*S2+S3*S3):X3=X3+S3*T3:IFX3<0THENI=4
1670 IFI<4GOTO1630
1680 T3=T3*4:X1=X1+S4*T3:X2=X2+S5*T3
1700 IFABS(T1)>16/P4THENGOSUB2580
1710 RETURN
1720 T7=N1*N3*G*SIN(T1)/V-G*COS(D3)/V+L*COS(B)/M/V:D3=D3+T3*T7
1730 IFD3>P2THEND3=D3-P2
1740 IFD3<-P2THEND3=D3+P2
1750 RETURN
1760 A=INT(X3*P3):J=0:GOSUB3220
1770 A=INT(V/P6):J=1:GOSUB3220
1780 A=INT(V2/P6):J=2:GOSUB3220
1790 A=INT(T9):J=3:GOSUB3220
1800 A=INT(2.2*F9):J=4:GOSUB3220
1810 POKE530,1
1820 POKEEE,E3:IFPEEK(EE)=E9THENF1=F1+.06:IFF1>.9THENF1=.9
1840 POKEEE,E9:IFPEEK(EE)=E9THENF1=F1-.06:IFF1<-.9THENF1=-.9
1850 F=(F1+3*R9)/75
1870 A=INT(5728*F1)/100:J=5:GOSUB3220
1880 POKEEE,E6:IFPEEK(EE)=E9THENR9=R9+1:IFR9>10THENR9=10
1900 POKEEE,E9:IFPEEK(EE)=E8THENR9=R9-1:IFR9<-10THENR9=-10
1910 F=(F1+3*R9)/75
1920 A=INT(R9*10)/10:J=6:GOSUB3220
1940 POKEEE,E6:IFPEEK(EE)=E8THENN3=N3+.05:IFN3>1THENN3=1
1960 POKEEE,E9:IFPEEK(EE)=E7THENN3=N3-.05:IFN3<0THENN3=0
1980 A=INT(100*N3+.5)/100:J=7:GOSUB3220
1990 POKEEE,E6:IFPEEK(EE)=E7THENB=B+.06:IFB>2THENB=2
2010 POKEEE,E9:IFPEEK(EE)=E6THENB=B-.06:IFB<-2THENB=-2
2030 A=INT(B*572.8)/10:J=8:GOSUB3220
2040 POKEEE,E6:IFPEEK(EE)=E6THENTT=TT+1
2050 POKEEE,E9:IFPEEK(EE)=E5THENTT=TT-1
2070 T1=(TT/P4)*U/V1
2080 A=INT(T1*P4*10)/10:J=9:GOSUB3220
2090 A=INT(10*(D3+T1)*P4)/10:J=10:GOSUB3220
2100 M1=S2:M2=S1:GOSUB2430 :AL=A:J=11:GOSUB3220
2120 POKEEE,E6:IFPEEK(EE)=E5THENG1=0
2140 POKEEE,E9:IFPEEK(EE)=E4THENG1=1
2160 GOSUB2390
2170 POKEEE,E6:IFPEEK(EE)=E4THENT3=T3+1:IFT3>12THENT3=12
2190 POKEEE,E9:IFPEEK(EE)=E3THENT3=T3-1:IFT3<1THENT3=1
2200 POKE530,0
2210 A=T4:J=13:GOSUB3220
2220 A=INT(SQR(X1*X1+X2*X2)/16.09)/100:J=16:GOSUB3220
2230 A=INT(S3*P3):J=17:GOSUB3220
2240 M1=X2:M2=X1:GOSUB2430 :J=18:GOSUB3220
```

```
2250 M1=-S5:M2=-S4:GOSUB2430 :J=19:GOSUB3220
2260 R=INT(SQR(S4*S4+S5*S5)):J=20:GOSUB3220
2270 RETURN
2280 IFV<1.4*V1THENRETURN
2290 GOTO3500
2300 F9=F9-T3*N3*N3*M9/50000:IFF9<0THENF9=0
2310 IFF9=0THENN3=0
2320 M=M9+F9:RETURN
2330 FORI=1TO8:K(I)=K(I+1):NEXTI:K(9)=N3:T9=0
2340 FORI=2TO9:T9=T9+K(I)*(I-1)*3.2:NEXTI:T9=T9+340:T9=T9*(1+P1)/2
2350 IFT9<75THENT9=75
2360 IFT9>450THENN3=0
2370 RETURN
2380 FORI=1TO4:L=L+16*T1/P4:NEXTI:RETURN
2390 IFG1=0THENGOSUB2480
2400 IFG1=0ANDX3<30THENGOSUB2500
2410 IFG1=1THENGOSUB2520
2420 RETURN
2430 IFM2=0THENM2=10↑-4
2440 A=ATN(M1/M2)*P4
2450 IFM2<0THENA=A+180
2460 IFM1<0ANDM2>0THENA=A+360
2470 RETURN
2480 POKEEF,32:POKEEF+2,85:POKEEF+3,80:POKEEF+4,32
2490 POKEEF+5,32:RETURN
2500 POKEEF,238:POKEEF+2,85:POKEEF+3,80:POKEEF+4,32
2510 POKEEF+5,32:RETURN
2520 POKEEF,32:POKEEF+2,68:POKEEF+3,79:POKEEF+4,87
2530 POKEEF+5,78:RETURN
2540 PX=PU+485
2550 AX=0
2560 IFX1>2000THENAX=-64
2570 IFX1<0THENAX=64
2580 IFX2>1000THENAX=AX-1
2590 IFX2<-1000THENAX=AX+1
2600 POKEPX,149
2610 IFS1=0THENS1=.0001
2620 BX=ATN(S2/S1)
2630 IFS1>0ANDS2<0THENBX=P2*2+BX
2640 IFS1<0THENBX=BX+P2
2650 BX=(2*P2-BX)/P2*4+.5
2660 POKEPX-65,32:POKEPX-64,32:POKEPX-63,32:POKEPX-1,32
2670 BX=16+INT(BX)
2680 IFBX=24THENBX=16
2690 POKEPX+1,32:POKEPX+63,32:POKEPX+64,32:POKEPX+65,32
2700 POKEPX+AX,BX
2710 RETURN
2720 IFX1>0ANDX1<2000GOTO2740
2730 RETURN
2740 IFX2<30ANDX2>-30GOTO2760
2750 RETURN
2760 IFX3>-1ANDX3<4GOTO2780
2770 RETURN
2775 SA=S1:IFSA=0THENSA=.0001
2780 SS=ABS(S2/SA)
2785 IFSS<.05GOTO2795
2790 RETURN
2795 IFD3+T1>-.05GOTO2805
```

```
2810 RETURN
2820 IFS3<=0ANDS3>-10GOTO2820
2830 RETURN
2840 IFC1=0GOTO2890
2850 PRINTCHR$(3)
2860 PRINTTAB(12);"U BENT VEILIG GELAND"
2870 FORI=0TO13:PRINT:NEXTI:FORI=1TO10000:NEXTI
2880 PRINTTAB(12);"SAVE LANDING":PRINT:PRINT:PRINT
2890 GOTO3510
2900 ZF=INT(S3*P3):IFZF>0ANDZF<2.5GOTO2920
2910 RETURN
2920 GOTO2830
2930 B9=ABS(V/V1/3):C9=ABS(V/V1/4):B9=B9*RND(5):C9=C9*RND(5)
2940 S4=2*(B9-.05)*V+S4
2950 IFABS(S4)>20THENS4=SGN(S4)*20
2960 S5=2*(C9-.05)*V+S5
2970 IFABS(S5)>20THENS5=SGN(S5)*20
2980 RETURN
2990 PRINTCHR$(3):POKE546,20:PRINTCHR$(2)TAB(26);"+     -":PRINT
3000 PRINT"ALTITUDE.......F"
3010 PRINT"SPEED........K/H"
3020 PRINT"STALL SPEED..K/H"
3030 PRINT"ENGINE TEMP....C"
3040 PRINT"FUEL...........L"
3050 PRINT"FLAPS..........D          Q    1"
3060 PRINT"TRIM............          W    2"
3070 PRINT"THRUST..........          E    3"
3080 PRINT"BANK...........D          R    4"
3090 PRINT"ATTACK ANGLE...D          T    5"
3100 PRINT"HORIZON........D"
3110 PRINT"HEAD.OFF EAST..D"
3120 PRINT"LANDING GEAR....          Y    6"
3130 PRINT"FLIGHT TIME....S          U    7"
3140 PRINT: PRINTTAB(4);"****";
3150 PRINT" CONTROL TOWER ";"****"
3160 PRINT"RANGE..........M"
3170 PRINT"CLIMBE RATE..F/S"
3180 PRINT"POS.OFF RUNWAY D"
3190 PRINT"WIND DIRECTION D"
3200 PRINT"WIND SPEED...K/H"
3210 POKE546,12:PRINT:RETURN
3220 IFA<0THENA=-A:KX=1
3230 E(0)=INT(A/10000)
3240 E(1)=INT((A-E(0)*10000)/1000)
3250 E(2)=INT((A-E(0)*10000-E(1)*1000)/100)
3260 E(3)=INT((A-E(0)*10000-E(1)*1000-E(2)*100)/10)
3270 E(4)=INT(A-E(0)*10000-E(1)*1000-E(2)*100-E(3)*10)
3280 E(5)=A-INT(E(0)*10000)-INT(E(1)*1000)-INT(E(2)*100)
3290 E(5)=E(5)-INT(E(3)*10)-INT(E(4))
3300 E(6)=INT(E(5)*10+.005)
3310 E(7)=INT(E(5)*100-(E(6)-.005)*10+.05)
3320 E(5)=-2
3330 LX=0
3340 PD=53668+64*J
3350 IFKX=1THENKX=0:E(0)=-3
3360 FORI=0TO7:E(I)=E(I)+48:NEXTI
```

```
3370 IFE(0)=48THENE(0)=32
3380 IFE(1)=48ANDE(0)=32THENE(1)=32
3390 IFE(1)=48ANDE(0)=45THENE(1)=32
3400 IFE(2)=48ANDE(1)=32THENE(2)=32
3410 IFE(2)=32ANDE(3)=48THENE(3)=32
3420 IFE(4)=48ANDE(3)=32THENE(4)=32
3430 IFE(7)>57THENE(7)=48
3440 IFE(6)=48ANDE(7)=48THENE(6)=32:E(7)=32
3450 IFE(6)=32ANDE(4)=32THENE(4)=48
3460 FORI=0TO7
3470 POKEPD+LX+I,E(I)
3480 NEXTI
3490 RETURN
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:POKEEE,239
3610 IFPEEK(57088)=247THEN0A=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
```

Vervolg van instructies 'YAHTZEE'

Iedere beurt moet een van de nog overgebleven katagorieen worden gevuld desnoods met 0 (NUL).
Als het totaal van de eerste 6 katagorieen (de ENEN t/m de ZESSEN) 63 of hoger is, wordt een bonus van 35 punten uitgedeeld.
Geef bij een beurt na een worp eerst het aantal opnieuw te werpen stenen op (ev. 0 als je bent uitgegooid) en daarna telkens het nummer van de te veranderen steen, de meest linkse is 1, die daarnaast 2 etc.
Door op REPEAT te drukken voor een worp, wordt het totaal overzicht gegeven. Het programma sorteert iedere keer de stenen in dalende volgorde.

```
10 REM BREAKOUT,VAN D.E.TEWKSBARY,UIT 0.S.I SMALL SYSTEMS JOURNAL
20 REM BEWERKT VOOR SUPERBOARD DOOR JOS BURGHOUTS,MAART'80
30 REM ER MOETEN 5 STARTWAARDEN WORDEN OPGEGEVEN AFH. VAN DE GEBRUIKTE
40 REM VIDEO EN KEY-BOARD LEESROUTINE. HET KEY-BOARD WORDT RECHTSTREEKS
50 REM GELEZEN VIA PEEK EN POKE OP T1(VOOR SUPERBOARD 57088),
60 REM EN INDIREKT MET DE MACHINETAAL ROUTINE OP $FD00
70 REM HET VIDEOBEELD KAN WORDEN OPGEBOUWD MET DE VOLGENDE WAARDEN:
80 REM A=STARTPUNT,LINKSBOVEN OP HET SCHERM
90 REM RB=REGELBREEDTE, DE BREEDTE VAN HET SPEELVELD(EXKL. DE RAND!)
100 REM RL=REGELLENGTE, VOOR SUPERBOARD VAST 32, VOOR 2P 64 ETC.
110 REM VH=VIDEOHOOGTE, HET AANTAL REGELS VERTIKAAL VAN HET SPEELVELD
120 GOSUB 900
130 INPUT"WIL JE INSTRUKTIES";Y$:IF LEFT$(Y$,1)="N"THEN250
135 PRINTCHR$(2)
140 PRINT:PRINTTAB(22)"**** BREAKOUT ****":PRINT:PRINT
145 POKE15,0
150 PRINT"HET IS DE BEDOELING OM ZOVEEL BARIERES WEG TE"
160 PRINT"WERKEN, DAT DE BAL DE BOVENKANT VAN HET SCHERM"
170 PRINT"KAN BEREIKEN.JOUW SLAGHOUT (OP DE BODEM"
190 PRINT"VAN HET SCHERM) KUN JE MET EEN JOYSTICK NAAR"
200 PRINT"LINKS EN NAAR RECHTS BEWEGEN,EN DOOR DE KNIPPEL"
210 PRINT"IN TE DUWEN SERVEER JE DE VOLGENDE BAL.":PRINT
220 PRINT"JE KRYGT MAX. 6 BALLEN, EN JE KRYGT MEER PUNTEN"
230 PRINT"NAARMATE JE DIEPER DOORDRINGT IN DE BARIERE."
240 INPUT"BEN JE GEREED";Y$
250 GOSUB 900:POKE530,1
260 A=53450:T1=57088:RB=46:RL=64:VH=22
270 B=A+RB:C=B+RL+1:D=C+VH*RL:E=D+RL-1:F=E-RB:G=F-RL-1:H=G-VH*RL
280 T=A+(VH+4)*RL+5:P=G+RB/2:T2=T-RL-1:T3=T2+9
285 POKET2,66:POKET2+1,65:POKET2+2,76
290 POKET3,83:POKET3+1,67:POKET3+2,79:POKET3+3,82:POKET3+4,69
295 POKET3,83:POKET3+1,67:POKET3+2,79:POKET3+3,82:POKET3+4,69
300 FORI=CTODSTEPRL:POKEI,47:POKEI+1,47:NEXTI:POKEI,35
310 FORI=ETOFSTEP-1:POKEI,95:NEXTI:POKEI,35
320 FORI=GTOHSTEP-RL::POKEI,92:POKEI-1,92:NEXTI:POKEI,35
330 FORI=ATOB:POKEI,42:NEXTI:POKEI,35
340 FORI=A+RLTOB+RL:POKEI,51:NEXTI
350 FORI=A+2*RLTOB+2*RL:POKEI,50:NEXTI
360 FORI=A+3*RLTOB+3*RL:POKEI,49:NEXTI
370 M(1)=-RL-1:M(2)=-RL:M(3)=-RL+1:M(4)=RL-1:M(5)=RL:M(6)=RL+1
380 BC=79:BO=32:PC=61
```

```
390 P=G+12:T=A+(VH+4)*RL+5
400 POKET,48:POKET+10,43:POKET+11,48
410 POKEP,PC:POKEP+1,PC
420 X=0
430 IF X=6 THEN 730
440 POKET1,127:IF(255-PEEK(T1))<>64THEN440
450 X=X+1:POKET,48+X
460 BP=INT(RND(BC)*RB+A+5*RL+1):GOSUB710:L=12
470 FOR AS=1TOL:IFAS<>1THEN630
480 POKEBP,B0:Q=PEEK(BP+M(IB)):IFQ=B0THEN 620
490 IFQ=96THEN 620
500 L=L*.92:IFQ=92THENIB=IB+2:PRINTCHR$(7);:GOTO600
510 IFQ=47THENIB=IB-2:PRINTCHR$(7);:GOTO600
520 Q2=PEEK(BP+RL)
530 IFQ2=61THENIB=INT(RND(BC)*3)+1:PRINTCHR$(7);:GOTO600
540 IFQ=95THEN430
550 IFQ=42THEN730
560 IFQ=49THENZ=Z+1:PRINTCHR$(7);:GOSUB710:GOTO610
570 IFQ=50THENZ=Z+2:PRINTCHR$(7)CHR$(7);:GOSUB710:GOTO610
580 IFQ=51THENZ=Z+3:PRINTCHR$(7)CHR$(7)CHR$(7);:GOSUB710:GOTO610
590 IB=INT(RND(BC)*6)+1
600 Q2=PEEK(BP+M(IB)):IFQ2<>32ANDQ2<>96THEN590
610 POKET+10,INT(Z/10)+43:POKET+11,Z+48-INT(Z/10)*10
620 BP=BP+M(IB):POKEBP,BC
630 POKET1,127:C=255-PEEK(T1):IFC=12ANDP<D-1THEN660
640 IFC=13ANDP>G+1THEN680
650 GOTO700
660 P=P-1:IFP>GTHENPOKEP,B0
670 P=P+2:POKEP,PC:GOTO700
680 P=P+1:IFP<DTHENPOKEP,B0
690 P=P-2:POKEP,PC
700 NEXTAS:GOTO470
710 POKEBP+M(IB),B0:IB=INT(RND(BC)*3)+4:RETURN
720 POKE11,0:POKE12,253:TR=USR(TR):TR=PEEK(531):RETURN
730 POKE530,0:GOSUB900:POKE15,255
740 FORX=1TO20:PRINTCHR$(7);:NEXTX
750 PRINT"DIT MOET HET EINDE VAN DIT SPELLETJE ZYN.":PRINT
760 IF Q<>42 THEN 820
770 PRINT"GEFELICITEERD ! JE HEBT DE BARIERE GENOMEN !!!!!"
780 PRINT"DAARBY HEB JE";Z;"PUNTEN GEHAALD.":PRINT
790 PRINT"OM TE BEWYZEN DAT HET GEEN PUUR GELUK WAS ZOU "
800 PRINT"JE NOG EEN SPELLETJE MOETEN SPELEN !":PRINT
810 GOTO 850
820 PRINT"JAMMER MAAR JE HEBT HET NIET GEHAALD !!":PRINT
830 PRINT"WEL HEB JE";Z;"PUNTEN IN DE WACHT GESLEEPT."
840 PRINT:IFZ>20THENPRINT"LANG NIET ZO SLECHT !!! "
850 PRINT:INPUT"NOG EEN SPELLETJE";Y$
860 IF LEFT$(Y$,1)="J"THEN RUN 250
870 PRINT:PRINT"EINDE PROGRAMMA":END
900 POKE11,6:POKE12,254:SC=USR(SC):RETURN
READY
```

Instrukties voor Breakout

Dit is een soort televisie-tennis spel , waarbij de speler met een slaghout de bal moet terugspelen naar een wand/barriere. Telkens als de bal de barriere raakt, wordt die barriere verder afgebroken (en worden punten uitgedeeld).

- Het videobeeld kan worden opgebouwd met behulp van 4 gegevens (op regel 260);
  A=startpunt linksboven in het videobeeld
  RB=regelbreedte = breedte van het speelveld (exklusief de rand van 4!)
  RL=regellengte =de breedte ven de video, dus 32 of 64.
  VH=hoogte van het videobeeld/speelveld.
- In de regels 630 t/m 690 wordt de beweging van het slaghout bepaald.
- De balsnelheid neemt toe naarmate dezelfde bal langer in het spel is. De beginsnelheid is instelbaar met L (in regel 460), de snelheidsvergroting in regel 500 met L=L*.92. Bij 1=12 gaat de bal redelijk snel met een 2 MHz siesteem.

RENUMBER heeft geen extra toelichting nodig. Load-omvang ca 900 bytes. Run-omvang: ca 25 bytes extra voor elke regel van het te hernummeren basic-programma. PRINTCHR$(3) in regel 50410 is een clear screen.

```
50000 PRINTCHR$(3)
50001 REM 'RENUMBER' VOOR SUPERBOARD II
50002 REM J.M.A. HERMANS  SCHIPLUIDEN
50003 REM
50004 REM AUGUSTUS 1980
50005 REM HERNUMMERT PROGRAMMA'S MET REGELNUMMERS
50006 REM VAN ALLEEN 2 (OF 3 OF 4 OF 5) CIJFERS
50010 INPUT"HET GEWENSTE LAAGSTE REGELNR = ";W:PRINT
50015 INPUT"HET GEWENSTE REGELNR-VERSCHIL = ";V
50020 N=50000:P=773
50025 S=PEEK(P):IFS=0GOTO50035
50030 P=P+1:GOTO50025
```

```
50035 C=C+1:P=P+5:M=PEEK(P-2)+256*PEEK(P-1)
50038 PRINTC;M;" ";
50040 IFM=NTHENC=C-1:GOTO50050
50045 P=P+1:GOTO50025
50050 P=769:G=-1:DIMK(5,C)
50055 FORI=0TOC:K(0,I)=P+2
50060 AD=PEEK(P)+256*PEEK(P+1)
50065 RN=PEEK(P+2)+256*PEEK(P+3)
50070 K(1,I)=RN:K(2,I)=U*I+W
50080 0=P+4:P=AD
50090 R=PEEK(0):IFR=0GOTO50180
50100 IFR=136ORR=140GOTO50120
50105 IFR=160AND(PEEK(0+1)>48ANDPEEK(0+1)<58)GOTO50120
50110 0=0+1:GOTO50090
50120 U=0:G=G+1:S=0+1
50130 0=0+1:R=PEEK(0):IFR=0GOTO50180
50140 IFR=44GOTO50120
50145 IFR>47ANDR<58GOTO50160
50150 IFU=0THENG=G-1
50155 GOTO50110
50160 K(3,G)=S:U=U*10+R-48:K(4,G)=U
50170 GOTO50130
50180 NEXTI
50190 FORI=0TOG
50200 FORJ=0TOC
50210 IFK(1,J)=K(4,I)THENK(5,I)=K(2,J)
50220 NEXTJ
50230 NEXTI
50240 FORI=0TOC
50250 IFK(2,I)=0THENI=C:GOTO50280
50260 N=INT(K(2,I)/256):POKEK(0,I)+1,N
50270 M=K(2,I)-256*N:POKEK(0,I),M
50280 NEXTI
50290 FORI=0TOG
50300 IFK(3,I)=0THENI=G:GOTO50400
50310 E=K(5,I):E=K(5,I)
50320 A1=INT(E/10000):A2=INT((E-10000*A1)/1000)
50325 A3=INT((E-10000*A1-1000*A2)/100)
50330 A4=INT((E-10000*A1-1000*A2-100*A3)/10)
50335 E=INT(E-10000*A1-1000*A2-100*A3-10*A4)
50340 IFE=0THENA5=0:GOTO50350
50345 A5=INT(E*.1)
50350 IFA1=0THENA1=A2:A2=A3:A3=A4:A4=A5:GOTO50350
50360 ON LEN(STR$(K(5,I)))-1GOTO50390,50380,50375,50370,50365
50365 POKEF+4,A5+48
50370 POKEF+3,A4+48
50375 POKEF+2,A3+48
50380 POKEF+1,A2+48
50390 POKEF,A1+48
50400 NEXTI
50410 PRINTCHR$(3):LIST-49999: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 saven met zichzelf.
Voordeel: load-gedeelte wordt vanzelf vooraf geladen (de hexdump).
Je hoeft dan dus alleen het save-gedeelte te saven.
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 .
Moet zijn: $ D3Ø9.