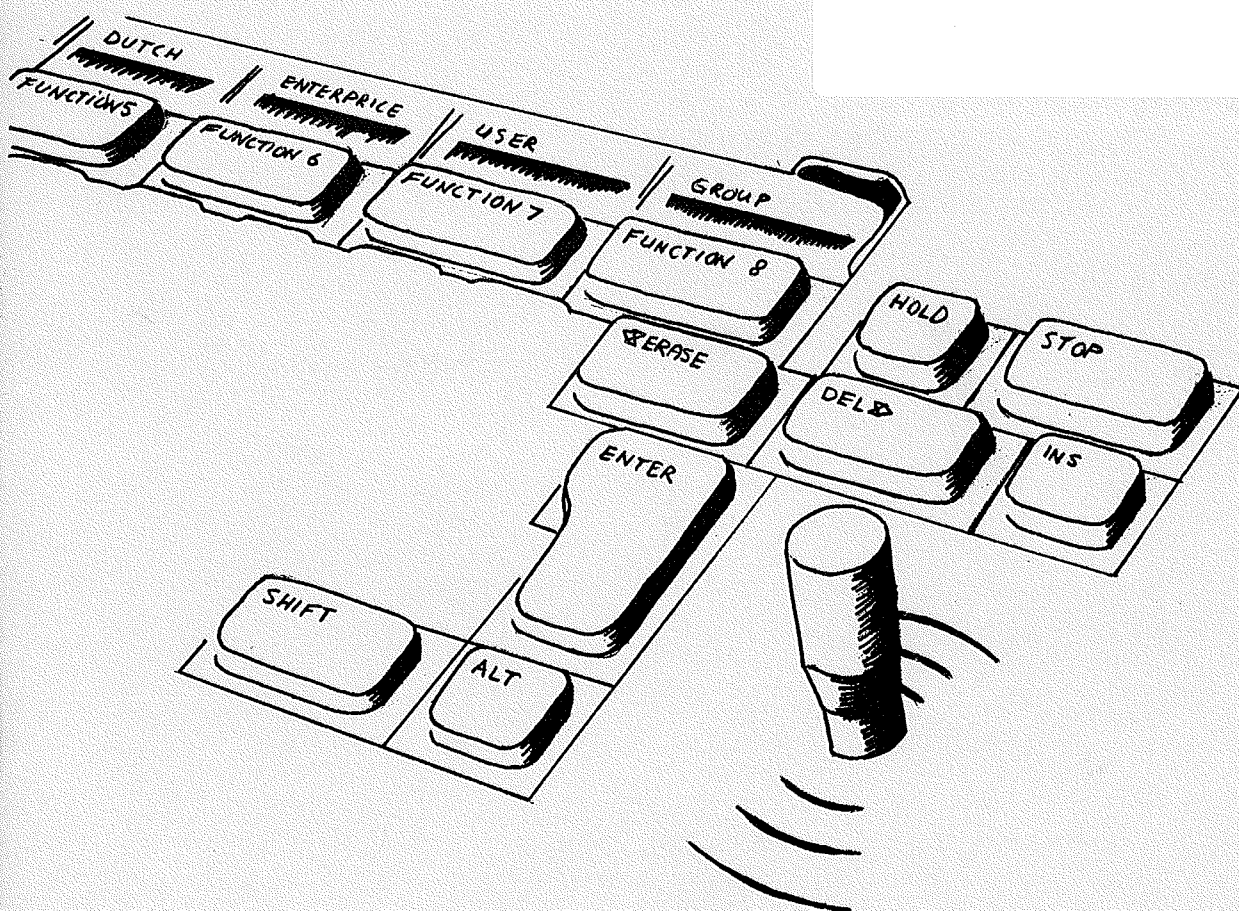


THE

INTERFACE

Port betaald
Port payé
Enschede

L. van Veen
Veldspaatstraat 278
9743 XS GRONINGEN



SEPT. '86

REDAKTIONEEL

Allereerst wil ik diegenen danken die ons aan de kopij hebben geholpen, die we voor deze editie hebben gebruikt. Maar de kopij is alweer bijna op, dus als u de volgende editie weer over twee maanden wilt ontvangen (of zelfs eerder), stuur dan ook eens een stukje. Wilt u daarbij op het volgende letten:

- * Gebruik geen ALT-characters, want daar kan de printer niet tegen.
- * Zet bij Word-Processor-files de LEFT-MARGIN op de 7e positie van links en de RIGHT-MARGIN op de laatste TAB-positie.
- * Stuur niet alleen programmas, maar schrijf er ook een stukje bij over de werking, de trucks e.d. bij, zodat een en ander wat leesbaarder wordt en tevens leerzamer.

Denk nooit dat uw programma's niet de moeite waard zijn om door anderen te worden gelezen, want er zijn nog een heleboel anderen die ook nog maar net bezig zijn met computeren.

Het adres is Menno Bins
Dollardstraat 92
7523 GS Enschede

GEBRUIKERSDAG OP 20 SEPTEMBER 1986

Hierbij nodigen wij u uit voor een bijeenkomst op zaterdag 20 september (en dus niet op 6 september zoals tijdens de jaarvergadering is gezegd) in het dorpshuis te De Bilt. De bijeenkomst duurt van 10:00 tot 16:00 en wordt gehouden in de "Over de Veght"-zaal. U wordt verzocht de opkomst te vergroten.

De Bilt is gemakkelijk te bereiken via o.a. de E-8 en de A-27. Er is bij het dorpshuis ruime parkeergelegenheid aanwezig.

Wilt u met het openbaar vervoer komen, doe dit dan als volgt:

- Met de trein naar Utrecht
- Met bus 57 van Centraal Nederland (rijdt elke 10 minuten)
- Uitstappen bij halte Nobellaan op de Blauwkapelseweg
- Oversteken naar Nobellaan
- 3e weg links

Het adres is: H.F. Witte Dorpshuis
Henri Dunantplein 4
De Bilt

INGEZONDEN BRIEF

Het was fijn om weer eens iets van ENTERPRISE-gebruikers te horen.

Alleen WAAROM MOET HET ALTIJD OP HET LAASTE NIPPERTJE?

Alle uitnodigingen voor gebruikersdagen ontving ik DE DAG ERVOOR OF DE DAG ZELF en dan is het natuurlijk niet voor iedereen mogelijk om te komen. Ikzelf werk meestal in het weekeind en moet dan wel een of twee weken vantevoren even vrij regelen!

Ik hoop dat de nieuwe redactie er voor zorgt dat alles wat meer op tijd onder de mensen is.

Een goed idee b.v. is een VASTE DAG een maal per kwartaal af te spreken om een gebruikersdag te houden, dan is die datum voor een aantal jaren al bekend. Ook voor de verschijningsdatum van het ENTERPRISE-blaadje is een (2)-maandelijkse datum af te spreken. Het blaadje kan desnoods bij gebrek aan kopij alleen maar uit een voorpagina bestaan met enkele redactionele aansporingen voor kopij.

Er stonden leuke dingen in het ENTERPRISE-blaadje en ik hoop dat we in Nederland ook zoveel interessante dingen te vertellen hebben.

met sterregroeten van Mister Spock

OPSLAG CHARACTERSET IN HET GEHEUGEN

Een character is opgebouwd uit een patroon 8×9 bits. 8 horizontale bits worden samengevoegd tot 1 byte (lijn). Een character is dus opgebouwd uit 9 lijnen van 8 bits. Er zijn 128 characters gedefinieerd, dus de hele characterset neemt $128 \times 9 = 1152$ bytes in beslag. Deze informatie ligt opgeslagen in RAM geheugen en wel in segment 255. Het adres van het eerste byte van de characterset zal ik Basis-Character-Set noemen, kortweg BCS. De waarde van BCS kan verschillen per computer.

Als er een character op het scherm geprint moet worden, wordt deze lijn voor lijn opgehaald uit de characterset en op de juiste plaats in schermgeheugen gezet. (Behalve bij text 40, waar een character direct in zijn geheel wordt afgebeeld.)

Characters kunnen veranderd worden mbv. SET CHARACTER en weer in hun originele staat gebracht mbv. CLEAR FONT. Deze commando's beïnvloeden dus de characterset in RAM.

De bytes van de verschillende characters zijn in de characterset opgeslagen op een manier die eigenlijk niet meteen voor de hand liggend is:

A. De opeenvolgende bytes van een bepaalde character zitten niet op opeenvolgende adressen, maar liggen 128 geheugenplaatsen uit elkaar.

B. De afdruckbare characters lopen in BASIC van 32 t/m 159. De character set echter zet de characters 128 t/m 159 voorop. De volgorde wordt dus:

CHARACTERSET: [0, 1, 2...31] [32,33...127]

BASIC: [128,129,130...159] [32,33...127]

ADRES	CHARACTER:	0	1	2	...	A	B	C	...	~
adres 1e lijn=	BCS+0*128 + 0	1	2	...	65	66	67	...	126	127
adres 2e lijn=	BCS+1*128 + 0	1	2	...	65	66	67	...	126	127
adres 3e lijn=	BCS+2*128 + 0	1	2	...	65	66	67	...	126	127
////////////////////////////////////										
adres 8e lijn=	BCS+7*128 + 0	1	2	...	65	66	67	...	126	127
adres 9e lijn=	BCS+8*128 + 0	1	2	...	65	66	67	...	126	127

FORMULE:

ADRES_CHARACTER_LIJN = BCS + MOD(ORD("C"),128) + LIJN * 128 - 128

Voorwaarden: -LIJN is 1 t/m 9 (boven naar onder)

-BCS is basis character set

-C is de betreffende character

DE WAARDE VAN BCS:

De waarde van BCS kan gevonden worden door van het Line Parameter Table adres 1152 af te trekken.

SUBROUTINES

Ik heb een 4 tal subroutines in machine taal geschreven die de character set in een fractie van een seconde verandert, zodat de characters op een andere manier gebruikt kunnen worden.

1. Six-Pixel characters voor grafische afbeeldingen
2. Inverteren van character set
3. Alle characters een slag naar rechts draaien
4. Characters met dubbele hoogte in alle mode's (behalve text 40)

BELANGRIJK:

De programma's zijn geschreven op een ENTERPRISE 128 met BASIC versie 2.1, er is geen rekening gehouden met eventuele ALLOCATE fouten. Ik weet niet zeker of de berekening van BCS in alle gevallen de juiste waarde oplevert.

Een typfout in de HEX-code van de MC-Subroutines kan rampzalige gevolgen hebben, dus save het programma voor je het runt.

LOAD/SAVE CHARACTERSET

Met behulp van de characterdefiner kun je nieuwe characters maken. Je zou dit voor alle characters kunnen doen, zodat er een heel nieuw lettertype ontstaat, en dit vastleggen op band of disk. Ik heb een programma geschreven dat razendsnel de hele characteraset via een file vastlegt en zo'n file kan natuurlijk ook weer ingelezen worden. Je zou dit programma toe kunnen voegen aan de character definer.

VRAGEN:

1. De CUMANA diskdrive's die genoemd werden in ENTERFACE komen die ook naar Nederland?
2. Is het ook mogelijk hardware te bestellen bij de D.E.U.G? B.v. Disk Controller, stekkertjes.
3. Wat zijn de ervaringen met CP/M? Wat voor een soort programma's zijn er en hoe kom je eraan?

```
100 PROGRAM "Chr_subrout-BAS"
110 SET FKEY 8 "clear font"&CHR$(13)
120 ALLOCATE 1000
130 LET LPT=SPEEK(255,16372)+256*SPEEK(255,16373)
140 LET BCS=LPT-1152
150 GOSUB 1000 ! Def MC routines
160 CALL P1 ! six-pixel
170 LOOK #105:D
180 CALL P2 ! invert
190 LOOK #105:D
200 CALL P3 ! turn right
210 LOOK #105:D
220 CALL P4 ! dubbele hoogte
230 END
240 !
1000 REM *** MC-routines
1004 CODE INVERT=HEX$("0E,09,1E,FF,06,80,7E,AB,77,23,10,FA,0D,20,F5,C9")
1008 CODE TURNR=HEX$("11,7F,00,E5,D5,19,1E,01,0E,08,06,08,E5,7E,A3,C6,FF,CB,1A,
7D,C6,80,6F,30,01,24,10,F1,E1,D5,CB,23,0D,20,E7,06,08,11,80,00,F1,77,19,10,FB,36,
00,d1,E1,2D,20,Cf,C9")
1012 CODE SIXPIX=HEX$("11,80,00,06,40,e5,48,c5,06,03,3e,00,cb,39,30,02,f6,0f,cb,
39,30,02,f6,f0,77,19,77,19,77,19,10,ea,c1,e1,23,78,fe,21,20,02,ed,52,10,d9,c9")
1016 LET INIT$=HEX$("dd,e5,fd,e5,fd,21,00,00,eb,fd,19")
1020 LET UP$=HEX$("01,00,04,fd,09,21,80,ff")
1024 LET LO$=HEX$("01,00,00,fd,09,21,80,00")
1028 LET PROG$=HEX$("dd,21,00,00,dd,19,01,00,02,dd,09,eb,0e,80,dd,e5,fd,e5,dd,7
e,00,fd,77,00,dd,19,fd,19,06,04,dd,7e,00,dd,19,fd,77,00,fd,19,fd,77,00,fd,19,10,
ef,fd,e1,dd,e1,fd,23,dd,23,0d,20,d4,fd,e1,dd,e1,c9")
1032 CODE DUBUP=INIT$&UP$&PROG$
1036 CODE DUBLO=INIT$&LO$&PROG$
1040 RETURN
```

```

1100 DEF P1 ! Sixpix
1104 CALL USR(SIXPIX,BCS+96)
1108 LET W1$="aamtv"ambam95<#q"
1112 LET W2$="a'am↑pvam'a'amπs@fv"

1116 GRAPHICS ATTRIBUTE
1120 PRINT "64 SIX-PIXEL CHARACTERS VOOR GRAFISCHE AFBEELDINGEN IN TEXT MODE"
1124 FOR T=1 TO 6
1128 SET INK T:SET PAPER T+2
1132 PRINT #101:TAB(4+T) W1$
1136 PRINT #101:TAB(4+T) W2$:PRINT #101:
1140 NEXT
1144 LOOK #105:D
1148 SET INK 1:SET PAPER 0:CLEAR #101
1152 LET V=0
1156 FOR T=96 TO 159
1160 SET PAPER 4
1164 PRINT #101:TAB(1+V) STR$(T);
1168 SET PAPER 7
1172 PRINT #101:TAB(4+V) CHR$(T)
1176 PRINT #101:
1180 IF MOD(T-95,10)=0 THEN LET V=V+5
1184 NEXT
1188 END DEF
1200 DEF P2 ! Inverteren character set
1204 CLEAR FONT
1208 GRAPHICS HIRES 16
1212 PRINT #101,AT 1,1:"";
1216 PRINT "ALLE CHARACTERS WORDEN GEINVERTEERD"
1220 FOR T=1 TO 2
1224 FOR I=1 TO 5
1228 SET INK I
1232 PRINT #101:" Normaal of Inverse"
1236 NEXT
1240 CALL USR(INVERT,BCS)
1244 NEXT
1248 END DEF
1300 DEF P3 ! Characters naar rechts draaien
1304 SET BIAS 73
1308 LET X=4:LET Y=1
1312 GRAPHICS HIRES 16
1316 PRINT "Alle characters worden een slag naar rechts gedraaid"
1320 FOR T=1 TO 31
1324 SET INK RND(15)+1
1328 IF MOD(T,8)=0 THEN CALL USR(TURNR,BCS)
1332 IF T<9 THEN
1336 LET X=X+1
1340 ELSE IF T<17 THEN
1344 LET Y=Y+1
1348 ELSE IF T<25 THEN
1352 LET X=X-1
1356 ELSE
1360 LET Y=Y-1
1364 END IF
1368 PRINT #101,AT Y,X:"NORMAAL RECHTS OP DE KOP LINKS "(T:T)
1372 NEXT
1376 END DEF

```

```

1400 DEF P4 ! Dubbel
1404 CLEAR FONT
1408 TEXT 80:GRAPHICS ATTRIBUTE
1412 CALL PRDUB(0,1,1,"Characters met dubbele hoogte voor alle mode's (behalv
e text 40)")
1416 SET INK 7:SET PAPER 1
1420 CALL PRDUB(101,6,4," DEMONSTRATIE Subroutines ")
1424 SET INK 1:SET PAPER 2
1428 CALL PRDUB(101,8,4," ' Dubbele HOOGTE ' ")
1432 SET INK 2:SET PAPER 7
1436 CALL PRDUB(101,10,4," ENTERPRISE computers ")
1440 SET INK 1:SET PAPER 6
1444 PRINT #101,AT 12,4:" April 1986 "
1448 END DEF
5000 DEF PRDUB(CH,YY,XX,WW$)
5004 CALL USR(DUBUP,BCS)
5008 PRINT #CH,AT YY,XX:WW$;
5012 CLEAR FONT
5016 CALL USR(DUBLO,BCS)
5020 PRINT #CH,AT YY+1,XX:WW$;
5024 CLEAR FONT
5028 END DEF
9990 ! Wilfried Hesseling
9991 ! Enterprise 128 BASIC 2.1
9992 ! April 1986

```

```

100 PROGRAM "L-S-chrset-BAS"
110 ALLOCATE 100
120 LET LPT=SPEEK(255,16372)+256*SPEEK(255,16373)
130 LET BCS=LPT-1152 ! Basis CharacterSet
140 CODE LOAD_FILE=HEX$("3e,a,eb,01,80,04,f7,6,c9")
150 CODE SAVE_FILE=HEX$("3e,a,eb,01,80,04,f7,8,c9")
160 PRINT "<L>oad of <S>ave character set file?"
170 DO
180 LET I$=INKEY$
190 LOOP UNTIL I$="l" OR I$="s"
200 IF I$="l" THEN
210 PRINT "*** LOAD"
220 OPEN #10:"charfile" ACCESS INPUT
230 CALL USR(LOAD_FILE,BCS)
240 ELSE
250 PRINT "*** SAVE"
260 OPEN #10:"charfile" ACCESS OUTPUT
270 CALL USR(SAVE_FILE,BCS)
280 END IF
290 CLOSE #10
300 END

```

Files op de Enterprise.

Het is normaal niet mogelijk om diverse file-types door elkaar te verwerken. Bij het laden van bijvoorbeeld een BASIC-file in de tekstverwerker ontstaat een foutmelding (Invalid file format). Het is via de in Enterface 1.1 door Bas de Kruijff genoemde methode echter wel mogelijk. Wat hierbij gebeurt is dat de file in ASCII vorm, dus compleet met alle spaties, wordt weggeschreven in plaats van in gekodeerde vorm.

Hierin kun je zelfs nog veel verder gaan. Het is namelijk vanuit alle mij bekende systeem-software mogelijk om files in ASCII-vorm op te slaan. Ik heb het geprobeerd met IS-BASIC, GEN en QUASAR PASCAL. Files in ASCII-vorm kunnen vervolgens weer door de verschillende talen worden geladen. Als bij IS-BASIC echter een tekst wordt geladen die geen basic is volgt een foutmelding. Bij QUASAR kan elke ASCII-file worden behandeld als een normale pascal textfile.

Het leuke hiervan is, dat bijvoorbeeld een pascal-programma in de wordprocessor kan worden geschreven, weggezet en vervolgens ingelezen en gecompileerd in QUASAR.

Dit laatste heeft, gezien de nogal beperkte editor-mogelijkheden van QUASAR, in bepaalde gevallen voordelen. De maximale omvang van een file in de wordprocessor is echter kleiner dan die in QUASAR.

Een bijzonderheid van het laden van ASCII-files in de wordprocessor is, dat de bestaande tekst niet wordt gewist. De nieuw geladen tekst wordt over de oude heengeschreven of er tussen gezet, afhankelijk van de mode van de cursor. Begonnen wordt op de plaats waar de cursor staat.

Hierdoor is het mogelijk om in een bestaande tekst een stuk tussen te voegen dat reeds op band of schijf staat. Dit kan een programma-listing zijn. Als bij aangesloten EXDOS een DIR-commando wordt gegeven (toets FUNCTION 8, dan GEEN <enter> maar DIR <enter>) wordt de directory van de schijf in de tekst gezet.

Het is wel van belang de maximale lengte van de tekst in de wordprocessor in de gaten te houden, aangezien deze geen waarschuwing geeft als ze vol raakt. De tekst aan het begin wordt dan gewoon weggegooid. Voorzichtigheid geboden!

J.F. Dubbeldam
v Poelgeestlaan 59
2352 TB Leiderdorp
tel: 071-892368

ARRAY'S

De ENTERPRISE komputer kent drie statements om variabelen of array's te definiëren: STRING, NUMERIC en DIM.

Nu is het, b.v. voor een database programma handig om te weten hoeveel geheugenruimte er in beslag genomen wordt. Je kan dan niet teveel geheugen ongebruikt laten.

De dimensionering is als volgt:

```

STRING naam$(E)*S ==> G=2*E+E*S+S+69+L
STRING naam$*S ==> G=S+19+L
NUMERIC naam(A,B) ==> G=6*(A+A*B+B)+53+L
DIM is hetzelfde als NUMERIC

```

G is het benodigde geheugen,
 L de lengte van de naam van het array,
 E het aantal elementen in het array
 S de lengte van de string
 en A en B spreken voor zich

Uit deze formules kun je dus bij een gegeven een ander berekenen.

Nu wil het geval dat bij een deklaratie van een STRING er geen variabele kan staan op de plaats van S, alleen een getal is mogelijk.

Dit is nog iets voor machine-software-cracks om te verbeteren.

Een variabele of een array mag normaal gesproken maar een keer gedeclareerd worden.

Wanneer je nu een declareerd binnen een definitie gaat het wel goed.

```

100 DO !
110 STRING SPOCK$(S)*7 ! dit gaat dus niet
120 LOOP !

100 DO !
110 CALL SPOCKDEF ! dit gaat wel!
120 LOOP !
130 DEF SPOCKDEF !
140 STRING SPOCK(S)*7 !
150 END DEF !

```

In het programma "MAATWERK" is een werkend voorbeeld te zien.

Ervaringen met QUASAR PASCAL.

In het eerste nummer van dit jaar werd in de Enterface een berichtje opgenomen over QUASAR PASCAL. De zin om weer eens wat anders te doen met de Enterprise en het feit dat QUASAR in ROM wordt geleverd deden mij besluiten het te bestellen.

Na er enige tijd mee geëxperimenteerd te hebben heb ik een aantal voor- en nadelen op een rij gezet. In het dagelijks (beroeps-) leven werk ik veelvuldig op een Apple IIe computer met UCSD-pascal. Aangezien de Apple een computer is met "slechts" 64K geheugen en diskdrives met een capaciteit van 180K (Enterprise 713K geformatteerd op een DS/DD drive!) waren mijn verwachtingen wellicht wat hoog gespannen.

Een vergelijking tussen beide PASCAL-dialekten valt al zeer snel in het voordeel van de UCSD uit. Alleen op het gebied van snelheid wint QUASAR het. Het lijkt mij trouwens zinvoller om een vergelijking met HISOFT PASCAL te maken maar dat is nergens te krijgen.

Een aantal punten (wellicht heb ik op een aantal geen gelijk en zie ik iets over het hoofd maar... reacties graag!!):

- het is niet mogelijk een code-file apart op schijf of tape te bewaren. Een programma moet altijd eerst gecompileerd worden na te zijn geladen.

- er wordt nauwelijks gebruik gemaakt van de editor-mogelijkheden van EXOS (geen overwrite, wordwrap, cursorbesturing in combinatie met SHIFT, CTRL en ALT toetsen enz.).

- beperkte geheugenruimte beschikbaar (25K) voor programma's. De zeer uitgebreide geheugenfaciliteiten van de Enterprise zijn alleen beschikbaar voor channels (tekst, grafisch enz.).

- een groot aantal standaard PASCAL-kommando's is niet beschikbaar.

- QUASAR is nog lang niet vrij van bug's. Zo werkt bij mijn exemplaar de EOF-variabele niet goed, deze is altijd TRUE.

Uiteraard zijn er ook positieve kanten:

- programma's werken snel, zeker vergeleken met BASIC.

- het feit dat QUASAR in ROM zit maakt het zeer snel toegankelijk.

- find-kommando is zeer praktisch

- de compiler is behoorlijk snel

Het klinkt misschien over het algemeen wat zwart maar het is echt niet mijn bedoeling QUASAR af te kraken. Het lijkt in eerste instantie vooral te zijn ontworpen voor grafische toepassingen, een argument dat velen deed besluiten tot het aanschaffen van de Enterprise zelf!

Echter voor wat zakelijker gebruik lijkt het niet de beste keuze.

Als laatste nog een bug, die ontstaat in samenspel met EXDOS. Vanuit QUASAR kunnen op de gebruikelijke manier, dus dubbele punt gevolgd door EXDOS kommando, opdrachten worden gegeven. Als echter zo'n opdracht wordt gevolgd door een vraag vanuit EXDOS (bijvoorbeeld na :TIME) reageert de computer niet meer op het toetsenbord en moet worden ge-RESET. Dit komt omdat QUASAR een afwijkend kanaal voor het toetsenbord gebruikt.

J.F. Dubbeldam
v Poelgeestlaan 59
2352 TB Leiderdorp
tel: 071-892368

Aansluitingen van de ENTERPRISE.

Veel mensen zijn niet of nauwelijks bekend met de aansluitgegevens van de verschillende card-edge aansluitingen op de Enterprise. De importeur heeft ooit eens een blaadje uitgegeven maar niet iedereen heeft dit. Bovendien wordt daarin de expansion-connector niet genoemd. De aansluitingen zijn als volgt:

Aanzicht printkaart:

B1 B2 enz.

----- - -

A1 A2 enz.

Een / voor een signaal betekent inverse logika: 0=+5V, 1=0V.

Control 1/control 2:

A1: keyboard J (common)

A2: keyboard L

A4: KB4 (7) (right)

A5: KB2 (7) (down)

A6: KB0 (5) (fire)

B1: 0V

B4: +5V

B5: KB3 (8) (left)

B6: KB1 (6) (up)

De nummers tussen haakjes zijn control 2. Alle nivo's zijn TTL.

Printer (parallel Centronics)

A1: 0V

A2: /STROBE

A3: data 3

A5: data 2

A6: data 1

A7: data 0

B1: 0V

B2: /READY

B3: data 4

B5: data 5

B6: data 6

B7: data 7

Serial/networking

A1: reference

A3: RTS

A4: CTS

B1: 0V

B3: data out

B4: data in

relatieve signaalnivo's:

t.o.v. 0V-lijn: 0=0V, 1=12V

t.o.v. ref-lijn: 0=-5V, 1=+7V

Bij netwerk: verbind RTS met CTS en DATA IN met DATA OUT

Monitor:

A1: groen

A2: 0V

A3: monochrome video

A4: /HSYNC

A5: /VSYNC

A7: audio links

B2: 0V

B3: blauw

B4: rood

B5: /CSYNC

B6: mode switch

B7: audio rechts

De SYNC's zijn TTL-nivo's. De kleur-signalen zijn lineaire nivo's 0 tot 4 volt.

Expansion-connector:

A1 : rechts audio in	B1 : links audio in
A2 : /RFSH	B2 : /WR
A3 : /RD	B3 : /IORQ
A4 : NC	B4 : NC
A5 : /MREQ	B5 : /NMI
A6 : A8	B6 : A9
A7 : A10	B7 : A11
A8 : A12	B8 : A13
A9 : A14	B9 : A15
A10: A0	B10: A1
A11: A2	B11: A3
A12: A4	B12: A5
A13: A6	B13: A7
A14: D0	B14: D1
A15: D2	B15: D3
A16: D4	B16: D5
A17: D6	B17: D7
A18: /RESET	B18: /INT
A19: /WAIT	B19: 0V
A20: /MI	B20: 0V
A21: 1 MHz klok	B21: 0V
A22: /EXP	B22: 0V
A23: 8 MHz klok	B23: 0V
A24: EC0	B24: EC1
A25: EC2	B25: EC3
A26: /EXTC	B26: A16
A27: A17	B27: A18
A28: A19	B28: B20
A29: A21	B29: B22
A30: 0V	B30: VSYNC
A31: 0V	B31: NC
A32: HSYNC	B32: 0V
A33: +9V	B33: +9V

De functies van de verschillende aansluitingen op de expansion-connector zijn mij niet alle bekend. Wel duidelijk zijn de 22 adreslijnen. De eerste 16 komen direkt van de Z80, de hoogste 6 uit de DAVE-chip. De 8 datalijnen en 10 controllijnen zijn ook terug te vinden.

De signalen EC0 tot EC3 en /EXTC zijn ingangssignalen die direkt naar NICK gaan, de functie is mij onbekend. HSYNC en VSYNC zijn uitgangen, ze zijn verbonden met de gelijknamige uitgangen op de monitor-aansluiting.

Er zijn twee audio-ingangen voor het bijmengen van externe geluidsbronnen. De overige signalen zijn nog enkele klok- en voedingslijnen.

De signalen op de expansion-connector zijn, in tegenstelling tot die op de overige aansluitingen, niet gebufferd. Een groot aantal is zelfs direkt op de Z80 aangesloten. Bij uitbreiding moet hiermee terdege rekening worden gehouden.

J.F. Dubbeldam
v Poelgeestlaan 59
2352 TB Leiderdorp
tel: 071-892368

System variables

N=Nummer van system variable DV=Default Value

N	DV	Functie	N	DV	Functie
0	20	zie *0	20	255	NET CHANNEL
1	0	STOP KEY	21	255	NET MACHINE
2	32	INTERRUPT CODE	22	2	VIDEO MODE
3	0		23	0	VIDEO COLOR
4	0	DEFAULT CHANNEL	24	40	VIDEO X
5	0	TIMER	25	24	VIDEO Y
6	0	lock; zie *6	26	255	STATUS
7	0	KEY CLICK	27	0	BORDER
8	0	INTERRUPT STOP	28	0	BIAS
9	255	INTERRUPT KEY	29	102	EDITOR VIDEO
10	3	KEY RATE	30	105	EDITOR KEY
11	30	KEY DELAY	31	8	EDITOR BUFFER
12	0	TAPE SOUND	32	20	
13	0		33	0	FAST SAVE
14	0	SPEAKER	34	0	
15	20	SOUND BUFFER	35	0	TAPE LEVEL
16	15	SERIAL BAUD	36	255	REM1
17	0	SERIAL FORMAT	37	255	REM2
18	0	NET NUMBER	38	0	
19	0	INTERRUPT NET	39	0	TELLER zie *39

*0		bit 0: als 1 dan processor 6 keer vertraagd			
		bit 2: als 0 dan time\$ stopt			
		bit 4: als 0 dan toetsenbord uitgeschakeld			
*6		N=1: CAPS lock			
		N=2: SHIFT lock			
		N=8: ALT lock			
*39		Telt met snelheid van 1/50 seconde			

gebruik:		SET N,V			
		TOGGLE N			
		ASK N V			
adres:		segment 255,adres +/- 16325			

(LISP op de ENTERPRISE)

(Al geruime tijd ben ik bezig LISP onder de knie te krijgen en het lukt aardig) (Alleen heb ik een probleem met de hoeveelheid geheugen die er ter beschikking is)

(Wanneer ik namelijk een oude situatie laad met

(LOAD "FILENAAM.LSP") dan kan ik geen (RECLAIM) doen om te zien hoeveel geheugen er is gebruikt) (Ik krijg dan allerlei tekens en daarna:

*** interrupt being handled

*** interrupt being handled

en dit gaan door totdat je op de RESET knop drukt)

(Ik heb op dat moment alleen deze funties in het systeem:

Volume in drive B: is LISP.1

Directory of B:\

START	DEF	304	om de datum in te vullen
LISP	DEF r	414	om het systeem te saven
HELP	DEF	714	voor lijstje van zelfgemaakte funkties
TEKST	DEF r	86	om tekst modes te wisselen
ISDOS	DEF	328	om naar ISDOS te gaan
INKEY	DEF r	75	net als in basic
HAAL	DEF r	372	om een funktie .DEF van disk te halen
2K in 7 files		282K free)

(Dit vind ik niet echt veel) (Wie helpt mij?)

(Ik heb zoals je ziet de extension .DEF gebruikt voor LISP funkties die met (DEFUN) gemaakt zijn)

(Toen ik de funties maakte had ik nog niet (EXT <string>) ontdekt die niet in de handleiding staat maar wel in de (OBLIST))

(Deze werkt prima) (Je kunt naar ISDOS door (EXT "ISDOS") en een inhoudsopgave van de schijf krijg je door (EXT "DIR")) (Ik ben dit stuk aan het schrijven door (EXT "WP"))

(Dus enkele van mijn funties zijn dan ook achterhaald (hier niet READ-ONLY) en ik stuur daarom de "r" files op)

(De funtie (HAAL) is bijvoorbeeld erg handig)

(Ik stuur ook "23-AP-5.LSP" en "LISP.002", twee van mijn situaties die het niet goed doen (de beste))

(In ENTERFACE 1.1 staat een stukje van G.E uit E. die (m/v?) schrijft over LISP alsof het om een wiskundige taal gaat) (Dit is niet echt de bedoeling van de LISP uitvinders geweest) (LISP is een taal om te b.v. met TAAL te werken en om INTELLIGENTE programmaas te maken) (Er zijn ook diverse eenvoudiger (met minder of geen haakjes) talen geschreven in LISP) (METEOR voor taalkunde en PROLOG voor EXPERT-systemen)

(Boeken over LISP zijn er wel, maar moeilijk te vinden en duur) (Een leuk boek is DE TEKSTMACHINE van Academic Service)

(Ik zou het leuk vinden als er onder de weinige LISP-ENTERPRISER mensen waren die wat regelmatig met elkaar in kontakt komen om ervaringen uit te wisselen en van elkaar te leren) (Ikzelf ben nog geen EXPERT maar ik zou er wel een willen worden) (reakties wacht ik af)

(met de groeten van SPOCK)

(robin ketelaars, geerweg 2, 2611 VN DELFT, 015*126422)

KEYWORD VERVANGING

De vorige keer stond er een stukje in de INTERFACE over het plaatsen van basic keywords LOAD of SAVE etc. in een programma. Nu is dit wel makkelijk, maar de methode waarop is nog niet erg gebruikers-vriendelijk.

MR. SPOCK maakt daar voor je een eind aan door middel van het programma SUPERKEY.

Het gaat als volgt:

```
tik in      EDIT 1
           ok
tik in      LOAD "superkey.bas"
           ok
tik in      EDIT Ø
           ok
```

Nu is MR_SPOCK'S SUPERKEY gereed voor gebruik

```
maak dit programma:  1 END
                     100 PROGRAM "spock_voorbeeld"
                     110 PRINT 100-
                     120 PRINT
                     130 PRINT
                     140 PRINT
                     150 PRINT
```

```
tik nu in:  CHAIN "superkey"(110,"list")
```

Je krijgt dan LIST op regel 110

Speel met de joystick en verander 110 in 120 en LIST in INFO

Doe dit voor de andere regels ook tot je hebt:

```
1 END
100 PROGRAM "spock_voorbeeld"
110 LIST 100-
120 INFO
130 EDIT 1
140 LIST
150 EDIT Ø
```

Hernummer 1 in 160 en druk op START

In regel 1 is END noodzakelijk anders start het programma vanzelf en dat wil je waarschijnlijk niet.

De SUPERKEY veranderd alleen PRINT en achter PRINT moet je meteen zetten wat er evt. achter het nieuwe keyword moet.

De verkrijgbare keywords zijn: AUTO,DELETE,EDIT,LIST,LOAD,MERGE,NEW,OK,SAVE,START,INFO en VERIFY

VEEL PLEZIER MET MR_SPOCK'S SUPERKEY

```

100 PROGRAM "SUPERKEY"(REGLNUMMER,NEW_KEYWORD$)
110 !--- LOAD dit programma in PROGRAM 1 -- ga terug naar PROGRAM 0
120 !--- alleen PRINT wordt vervangen door dit programma
130 !--- gebruik: b.v. CHAIN "superkey"(320,"edit")
140 !--- zet aan het begin van je programma ---> 1 END anders start het
150 LET ADRES=4809;LET MEER=1
160 LET KEYWORD$="":LET NEW_KEYWORD$=UCASE$(NEW_KEYWORD$)
170 DO
180   LET JUMP=PEEK(ADRES)
190   IF REGLNUMMER=PEEK(ADRES+1)+PEEK(ADRES+2)*256 THEN
200     FOR J=ADRES TO ADRES+JUMP
210       IF PEEK(J)=96 AND PEEK(J+1)=56 THEN
220         LET NEW_KEY=VERVANGING
230         IF NEW_KEY THEN
240           POKE J+1,NEW_KEY
250           PRINT CHR$(25);PRINT CHR$(25);"poke "&STR$(J+1)&","&STR$(NEW_KEY
)
260           PRINT CHR$(25);"dus op regel";REGLNUMMER;"is PRINT gewijzigd in
";KEYWORD$
270           END IF
280           EXIT FOR
290           END IF
300           NEXT
310           END IF
320           LET ADRES=ADRES+JUMP
330 LOOP WHILE JUMP
340 CHAIN 0
350 END
360 DATA 02=AUTO,15=DELETE,20=EDIT,42=LIST,43=LOAD,45=MERGE
370 DATA 46=NEW,51=OK,67=SAVE,71=START,73=INFO,79=VERIFY,00=NIET
380 DEF VERVANGING
390   RESTORE 360
400   DO
410     READ KEY$
420     LET KEYWORD$=KEY$(4:)
430     LOOP UNTIL KEYWORD$=NEW_KEYWORD$ OR KEYWORD$="NIET"
440     LET VERVANGING=VAL(KEY$)
450 END DEF
460 !----- spock data '86 -----for ENTERPRISE 128-----

```


ZELF BASIC-KEYWORDS MAKEN

Na enig spitwerk is het me gelukt de opslagwijze van BASIC-keywords te vinden en zelfs ook zelf keywords te definiëren. Omdat hierna vraag is van meerdere mensen, heb ik wat meer haast gemaakt om dit in de Enteface te zetten. Er wordt gebruikt gemaakt van veel pointers dus heb ik er maar een schema van gemaakt om het een beetje overzichtelijk te houden. Hierna volgt ook een programma dat het BASIC-keyword REM omzet in CLS als voorbeeld. Als je geen zin hebt om het helemaal over te typen, dan kun je nog altijd een beroep doen op de softwarebibliotheek waar dat programma (CLS.BAS) in zou moeten staan.

BASIC-FUNCTIES

Na de keywordtabel staat er in het eerste RAM-segment ook nog een BASIC-functie-tabel. Het is me nog niet gelukt om daar bruikbare veranderingen in aan te brengen: zo gauw je iets aan een naam veranderd is hij de functie (het lijkt wel een soort variabele) vergeten. Ik heb ook geen pointer gevonden naar die tabel.

BASIC-VARIABLEN EN DEF'S

De opslag van BASIC-variabelen (de namen van def's horen daar ook bij) is wat moeilijk weer te geven in een schema of zo iets. Ik heb wel een programma gemaakt waarin je kunt zien hoe dat gebeurt, die je kunt overtikken of verkrijgen bij de softwarebibliotheek. (echt nuttig voor gebruik is het programma trouwens niet) Je zou de variabelentabel kunnen gebruiken om gegevens vanuit machinetaal door te geven aan BASIC. Het is alleen wel lastig om een bepaalde variabele terug te vinden in de lijst, maar ze worden wel opgeslagen in volgorde van initiatie dus je kan bijvoorbeeld die variabele helemaal vooraan kunnen zetten. Het variabelen-geheugen komt na het BASIC-programma in het geheugen en wordt dus telkens verplaatst. Net als bij BASIC-functies is het me niet gelukt variabele-namen te veranderen.

Fabian Moonen
Krijterslaan 18
1689 EM Zwaag
tel. 02290-37122

```

100 PROGRAM "CLS.BAS"
110 ' version 16-6-'86
120 '
130 ' made by Fabian Moonen
140 '       Krijterslaan 18
150 '       1689 EM Zwaag
160 '       tel. 02290-37122
170 '
180 LET PL_NAAM=1500 'address naamtabel
190 LET PL_CHECK=1550 'address checkroutine
200 LET PL_FUNC=1600 'address functionroutine
210 '
220 ' Naamtabel (gegevens over naam, syntax-flags en plaats check- en functionro-
utines)
230 POKE PL_NAAM+0,MOD(PL_FUNC,256)
240 POKE PL_NAAM+1,INT(PL_FUNC/256)
250 POKE PL_NAAM+2,MOD(PL_CHECK,256)
260 POKE PL_NAAM+3,INT(PL_CHECK/256)
270 FOR ADDRESS=PL_NAAM+4 TO PL_NAAM+1000
280   READ CODE
290   IF CODE=500 THEN EXIT FOR
300   POKE ADDRESS,CODE
310 NEXT
320 DATA 83      'flags keyword
330 DATA 3       'lengte keywordnaam
340 DATA 67,76,83'keywordnaam (CLS)
350 DATA 500     'eind data
360 '
370 'Checkroutine (bij invoer van het keyword kijkt hij naar syntax-fouten. De
ze checkroutine keurt alles goed.):
380 FOR ADDRESS=PL_CHECK TO PL_CHECK+1000
390   READ CODE
400   IF CODE=500 THEN EXIT FOR
410   POKE ADDRESS,CODE
420 NEXT
430 DATA 221,54,6,2 'LD (IX+6),2
440 DATA 201        'RET
450 DATA 500        'eind data
460 '
470 'Funcieroutine (het uitvoeren van het keyword)
480 FOR ADDRESS=PL_FUNC TO PL_FUNC+1000
490   READ CODE
495   IF CODE=500 THEN EXIT FOR
500   POKE ADDRESS,CODE
510 NEXT
520 DATA 62,0      'LD A,0
530 DATA 6,26     'LD B,26
540 DATA 247      'RST 30h
550 DATA 7        'DEFB 7
560 DATA 62,0     'LD A,0
570 DATA 6,185    'LD B,185
580 DATA 247      'RST 30h
590 DATA 7        'DEFB 7
600 DATA 201      'REI
610 DATA 500      'eind data
620 '
625 ' hier wordt het keyword in plaats van een andere keyword (REM) geplaatst:
630 LET PBKT=PEEK(562)+256*PEEK(563)
640 LET KEYW_CODE=61 'code van REM
650 LET PKNG=PBKT+3+3*KEYW_CODE
660 POKE PKNG+0,MOD(PL_NAAM,256)
670 POKE PKNG+1,INT(PL_NAAM/256)
680 POKE PKNG+2,IN(176) 'segmentnummer naamtabel

```

LOOPJES

Het is op de ENTERPRISE mogelijk om een DO LOOP te gebruiken. Ik doe dit dan ook vaak, zoniet altijd, omdat het een gestructureerde manier van programmeren is. En het programma blijft gemakkelijk leesbaar. Een voorbeeld:

```
100 PRINT "SPOCK"          100 DO
110 GOTO 100                110 PRINT "SPOCK"
                           120 LOOP
```

Hoewel links regel minder is, is recht mooi symmetries. Het maakt nu nog niet zoveel uit, maar let op:

```
  90 PRINT "SPOCK";
100 INPUT DATA              100 DO
110 IF DATA=1986 THEN 130   110 INPUT PROMPT "SPOCK":DATA
120 GOTO 100                120 LOOP UNTIL DATA=1986
```

Gelijkspel, maar rechts wint kwa leesbaarheid. Nog verder:

```
100 SPOCK=0                 100 DO
110 GOSUB 150               110 LOOP UNTIL SPOCK
120 IF SPOCK=1 THEN 140     120 !
130 GOTO 100               130 DEF SPOCK
140 END                   140 SPOCK=0
150 REM netjes hoor       150 INPUT PROMPT "SPOCK":DATA
160 PRINT "SPOCK";        160 IF DATA=1986 THEN SPOCK=1
170 INPUT DATA           170 END DEF
180 IF DATA=1986 THEN 160
190 SPOCK=1
200 RETURN
```

Wel meer regels voor links, en is het nog te volgen? Rechts is PRACHTIG toch? En warempel nog korter ook. Een mooie 2-regelige routine! Hier is dezelfde, maar dan nog ingewikkelder voor GOTO-ers

```
100 DO
110 LOOP WHILE SPOCK("SPOCK DATA")
120 !
130 DEF SPOCK$(DATA$)
140 SPOCK=1
150 INPUT PROMPT DATA$:DATA
160 IF DATA=1986 THEN SPOCK=0
170 END DEF
```

Afijn in mijn programmaas is er altijd wel wat van te zien! Ik hoop dat nu echt iedereen GOTO's vaarwel zegt.

N.B. achter een EXT statement in BASIC moet er helaas een om niet in een mooie LOOP te geraken!

```
b.v.      1000 EXT "mkdir SPOCK" :goto 1010
          1010 PRINT "directory SPOCK is gemaakt"
```

GROETEN VAN MISTER SPOCK

```

100 PROGRAM "ACCOORDEN.BAS"
110 ! version 6-7-'86
120 !
130 ! made by Fabian Moonen
140 !      Krijterslaan 18
150 !      1689 EM Zwaag
160 !      tel. 02290-37122
170 !
172 ! Gebruiksaanwijzing:
174 ! Voer een accoord in en de computer speelt dat accoord dan.
176 ! > = kruis < =mol
178 ! Voorbeelden:
179 ! C Cm F> G7 B<6 D+
180 !
185 SET SPEAKER ON
190 LET BAS$="CDEFGAB"
200 NUMERIC HB(7)
205 ! HB=hoogte bas: PITCH-waarde van de noten
210 LET HB(1)=33:LET HB(2)=35:LET HB(3)=37:LET HB(4)=38:LET HB(5)=40:LET HB(6)
=42:LET HB(7)=44
220 DO
230   DO
240     INPUT I$
250     LET I$=LTRIM$(RTRIM$(UCASE$(I$)))
260     LOOP WHILE LEN(I$)=0 OR LEN(I$)>4 OR POS(BAS$,I$(1:1))=0
270     CALL PLAYACC(I$)
280   LOOP
290 DEF PLAYACC(A$)
300   LET TEKEN,MINEUR,PLUS,SEX,SEPT=0
310   IF A$(2:2)=">" THEN
320     LET TEKEN=1
330     LET A$=T$(A$)
340   ELSE IF A$(2:2)="<" THEN
350     LET TEKEN=-1
360     LET A$=T$(A$)
370   END IF
380   IF A$(2:2)="M" THEN LET MINEUR=1:LET A$=T$(A$)
390   IF A$(2:2)="6" THEN LET SEX=1:LET A$=T$(A$)
400   IF A$(2:2)="7" THEN LET SEPT=1:LET A$=T$(A$)
410   IF A$(2:2)="+" THEN LET PLUS=1:LET A$=T$(A$)
420   IF LEN(A$)>1 THEN PRINT "Dat accoord kan ik niet spelen":EXIT DEF
430   LET H1=HB(POS(BAS$,A$))+TEKEN+4
440   LET H2=H1+4+MINEUR
450   LET H3=H1+7+PLUS+2*SEX+3*SEPT
460   CLEAR SOUND
470   SOUND PITCH H1,SOURCE 0,DURATION 100
480   SOUND SOURCE 1,DURATION 10,ENVELOPE 60
490   SOUND SOURCE 2,DURATION 20,ENVELOPE 60
500   SOUND PITCH H2,SOURCE 1,LEFT 0,DURATION 90
510   SOUND PITCH H3,SOURCE 2,RIGHT 0,DURATION 80
520 END DEF
530 DEF T$(A$)=A$(:1)&A$(3:)

```

MODEM

Sinds 14 mei ben ik in het bezit van een goedgekeurde modem, de POLYPHEMUS van het blad ELEKTUUR. Het is een moden geschikt voor viditel met 1200/75, 75/1200 modes. Door een kleine aanpassing is het geschikt voor 1200/1200 en 75/75 toepassingen. Vooral de 1200/1200 mode lijkt mij interressant voor ENTERPRISE-bezitters. Het hcc FIDO-netwerk gebruikt deze ook.

Het wordt nu voor + 250,- en een soldeerbout mogelijk met elkaar via een modem in kontant te komen en zo een snellere soft-, hard-, en infoware uitwisseling tot stand te krijgen. Misschien via een eigen databank.

Ikzelf heb nog geen werkend viditel-programma of een ander terminal programma om enige leesbare informatie uit anderekomputers te halen of er naar te sturen. Ik ben nu net wel in het bezit gekomen van het viditel-protocol, en ga dus zelf aan de slag.

Wanneer er mensen zijn die ook geïnteresseerd zijn in modemen of mij aan een viditel- of ander terminal-programma kunnen helpen kunnen ze bellen naar 015*126422.

met de groeten van SPOCK
starbase: DELFT, code: geerweg 2/2611VN

Hier volgen een aantal kleine dingen,

die ik allemaal maar even bij elkaar zet, die vast gebruikt kan worden door een aantal mensen.

Er is nog een niet verbeterd foutje in IS-BASIC 2.1:
twee keer achter elkaar NOT werkt niet goed.
bv. IF NOT 1 AND NOT 1 THEN PRINT "that's TRUE"

Een aantal nuttige routines:

Geef de status een andere kleur:

```
10 LET S=SPEEK(255,16372)+256*(SPEEK(255,16373) BAND 63)
20 SPOKE 255,S+8,BLUE ! COL.0
30 SPOKE 255,S+9,WHITE ! COL.1
40 SPOKE 255,S+10,BLUE ! COL.2
50 SPOKE 255,S+11,RED ! COL.3
```

Maak status leeg:

```
10 LET S=SPEEK(255,16374)+256*(SPEEK(255,16375) BAND 63)
20 FOR C=0 TO 39
30 SPOKE 255,S+C,32
40 NEXT C
```

INKEY\$ in machinetaal:

```
RST 10h
DEFB 25h
DEFB 0h
```

De waarde van INKEY\$ komt in de accumulator terecht.

JOY in machinetaal:

```
LD B,9h
LD C,0h (welke joystick)
LD A,69h (keyboard channel)
RST 30h
DEFB 0Bh
```

De waarde van JOY komt in register C.

Fabian Moonen

```

100 PROGRAM "Memory-BAS"
110 NUMERIC V(0 TO 63),KRT(1 TO 2),POSK(1 TO 2),UITPAREN,ZET,SP
120 NUMERIC COLPIC(0 TO 31,0 TO 1),CX,CY,SCORE(1 TO 2),NTLSP
130 STRING NAAM$(2)
140 CLEAR FONT:RANDOMIZE
150 ! Hoofdprogramma
160 CALL P11 ! Naam prog
170 CALL P12 ! Kies spelmogelijkheid
180 CALL P1 ! Initiatie
190 CALL P2 ! Schud kaarten
200 CALL P3 ! Zet scherm op
210 DO
220 LET ZET=ZET+1
230 CALL P10 ! Druk af zet, stand, speler
240 CALL P4 ! Kies 2 kaarten
250 IF P5 THEN ! De kaarten zijn gelijk ?
260 CALL P6 ! Verwijder de kaarten en verhoog score
270 ELSE
280 CALL P7 ! Draai de 2 kaarten weer om
290 END IF
300 LOOP UNTIL P8 ! Alle kaarten zijn op ?
310 CALL P10 ! Druk af zet, stand, speler
320 CALL P9 ! Eind spel
330 DO
340 LOOP
350 END
360 !
1000 ! Procedures in orde
1100 DEF P1 ! Initiatie
1104 CALL Q11 ! Set char kaarten
1108 CALL Q12 ! Initiatie kleuren van kaarten
1112 CLEAR TEXT: CLEAR GRAPHICS: SET INK 1
1116 LET UITPAREN=0: LET CX,CY,ZET=0: LET SP=1
1120 LET SCORE(1),SCORE(2)=0
1124 REDIRECT FROM #0 TO #102
1128 ENVELOPE NUMBER 1;0,60,60,1;40,0,0,10;-40,0,0,20
1132 ENVELOPE NUMBER 2;0,60,60,1;-10,0,0,10;-5,0,0,15
1136 END DEF
1200 DEF P2 ! Schud kaarten
1204 LET T=0
1208 FOR VELD=0 TO 63
1212 LET V(VELD)=INT(T): LET T=T+.5
1216 NEXT
1220 FOR VELD=0 TO 63
1224 LET R=RND(64): LET W=V(R): LET V(R)=V(VELD): LET V(VELD)=W
1228 NEXT
1232 END DEF
1300 DEF P3 ! Zet scherm op
1304 PRINT #101,AT 1,1:" 1 2 3 4 5 6 7 8"
1308 FOR Y=2 TO 9
1312 PRINT #101,AT Y,1:CHR$(Y+63)&" [][][][][][][][]"&CHR$(Y+63)
1316 NEXT
1320 PRINT #101:" 1 2 3 4 5 6 7 8"
1324 SET LINE MODE 3
1328 PRINT AT 4,1:"+++++ ZET: +++++"
1332 PRINT AT 2,2:NAAM$(1);TAB(22) NAAM$(2)
1336 SET #102:INK 2
1340 END DEF

```

```

1400 DEF P4 ! Kies 2 kaarten
1404 LET POSK(1)=-2:LET POSK(2)=-3:LET CX=RND(8):LET CY=RND(8)
1408 SET COLOR 6,0:SET COLOR 7,0
1412 FOR NUM=1 TO 2
1416 LET VELD=Q41 ! Kies kaart mbv pook
1420 IF NUM=2 AND POSK(1)=VELD THEN GOTO 1416
1424 IF VELD<0 OR VELD>63 OR V(VELD)=-1 THEN GOTO 1416
1428 LET K,KRT(NUM)=V(VELD):LET POSK(NUM)=VELD
1432 SET COLOR 2+2*NUM,COLPIC(K,0)
1436 SET COLOR 3+2*NUM,COLPIC(K,1)
1440 LET PC=MOD(V(VELD),16)*2:LET AFB$=CHR$(PC+126)&CHR$(PC+127)
1444 CALL PRKRT(VELD,AFB$,2+2*NUM,3+2*NUM)
1448 NEXT
1452 END DEF
1500 DEF P5 ! De kaarten zijn gelijk ?
1504 LET P5=(KRT(1)=KRT(2))
1508 END DEF
1600 DEF P6 ! Verwijder de kaarten en verhoog score
1604 CALL YEL
1608 LET UITPAREN=UITPAREN+1:LET SCORE(SP)=SCORE(SP)+1
1612 LET CL1=COLPIC(KRT(1),0):LET CL2=COLPIC(KRT(1),1)
1616 FOR T=0 TO 14
1620 SET COLOR 4,CL1:SET COLOR 5,CL2
1624 SET COLOR 6,CL1:SET COLOR 7,CL2
1628 LET D=CL1:LET CL1=CL2:LET CL2=D
1632 NEXT
1636 FOR NUM=1 TO 2
1640 CALL PRKRT(POSK(NUM)," ",0,0)
1644 LET V(POSK(NUM))=-1
1648 NEXT
1652 END DEF
1700 DEF P7 ! Draai de 2 kaarten weer om
1704 CALL DOWN
1708 WAIT 2
1712 CALL PRKRT(POSK(1),"[]",0,1)
1716 CALL PRKRT(POSK(2),"[]",0,1)
1720 IF NTLSP=2 THEN LET SP=3-SP
1724 END DEF
1800 DEF P8 ! Alle kaarten zijn op ?
1804 LET P8=(UITPAREN=32)
1808 END DEF
1900 DEF P9 ! Eind spel
1904 SET PALETTE 156,32,217:SET #102:PALETTE 32,156,32,217
1908 SET INK 2:PRINT #101,AT 5,4:"+L EIND SPEL +"
1912 PING:PING
1916 END DEF
2000 DEF P10 ! Druk af zet, stand, speler
2004 PRINT AT 2,16:SCORE(1)
2008 IF NTLSP=2 THEN PRINT AT 2,36:SCORE(2)
2012 PRINT AT 4,18:ZET;
2016 PRINT AT 3,SP*20-18:"[ ] [ ] [ ] [ ]"
2020 PRINT AT 3,-SP*20+42:" "
2024 END DEF
2100 DEF P11 ! Naam prog
2104 TEXT 40:GRAPHICS HIRES 16:SET BIAS 255:SET STATUS OFF
2108 SET BORDER 14:SET PALETTE 32,156,217
2112 SET #102:PALETTE 156,32,156,217
2116 LET W$=" * M E M O R Y * M E M O R Y *"
2120 SET LINE MODE 3
2124 FOR T=1 TO 8
2128 PRINT #101,AT T+1,2:W$(T*2-1:T*2+14)
2132 NEXT

```



```

2136   FOR T=0 TO 7
2140     SET INK T+8
2144     PRINT #101,AT 2,T*2+3:"++"
2148     IF T>0 THEN PRINT #101,AT T+2,3:"++"
2152   NEXT
2156 END DEF
2200 DEF P12 ! Kies spel mogelijkheid
2204   PRINT :PRINT "<<< SPEL >>>"&CHR$(241)
2208   PRINT "W. Hesseling - ENTERPRISE"&CHR$(241)&CHR$(246)
2212   PRINT "Maart 1986"&CHR$(241)&CHR$(254);
2216   DO
2220     LOOP WHILE INKEY$=""
2224     CLEAR TEXT
2228     PRINT " KIES:"
2232     PRINT "      1. Alleen spelen."&CHR$(246)
2236     PRINT "      2. Spelen met 2 personen."
2240     DO
2244       LET I$=INKEY$
2248       LOOP UNTIL POS(" 12",I$)>1
2252       LET NTLSP=VAL(I$)
2256       CLEAR TEXT:PRINT
2260       LET NAAM$(1),NAAM$(2)=""
2264       FOR T=1 TO NTLSP
2268         INPUT PROMPT " NAAM"&STR$(T)&": ";NAAM$(T)
2272         LET NAAM$(T)=UCASE$(NAAM$(T))(1:10)&": "
2276       NEXT
2280       CLEAR TEXT:PRINT "Kies kaart mbv. spelpook en spatiebalk!"
2284 END DEF
2999 !
3000 ! Procedures 2e orde
3100 DEF Q11 ! Set char kaarten
3102   SET CHARACTER 91,0,127,68,74,113,74,68,127,0
3104   SET CHARACTER 93,0,254,34,82,142,82,34,254,0
3106   SET CHARACTER 126,30,63,97,252,254,99,49,24,0
3108   SET CHARACTER 127,0,24,140,198,127,63,134,252,120
3110   SET CHARACTER 128,6,12,56,120,127,15,15,20,36
3112   SET CHARACTER 129,0,0,0,24,248,224,240,40,20
3114   SET CHARACTER 130,0,0,0,255,63,7,7,3,0
3116   SET CHARACTER 131,0,0,6,255,240,224,192,0,0
3118   SET CHARACTER 132,0,1,2,4,255,135,183,48,0
3120   SET CHARACTER 133,0,248,76,70,255,225,237,12,0
3122   SET CHARACTER 134,0,16,56,40,56,63,61,45,47
3124   SET CHARACTER 135,0,0,0,248,252,84,84,252
3126   SET CHARACTER 136,127,96,64,64,64,96,127,63
3128   SET CHARACTER 137,252,100,60,36,60,36,124,212,248
3130   SET CHARACTER 139,8,28,54,240,240,224,32,32,32
3132   SET CHARACTER 138,0,0,44,95,95,91,72,8,4
3134   SET CHARACTER 140,0,0,3,3,127,53,31,15,0
3136   SET CHARACTER 141,32,32,248,24,255,87,254,252,0
3138   SET CHARACTER 142,252,194,143,130,130,128,255,240,0
3140   SET CHARACTER 143,0,0,240,172,162,3,255,254,0
3142   SET CHARACTER 145,0,96,64,239,25,31,15,0,0
3144   SET CHARACTER 144,24,24,8,15,18,39,67,65,0
3146   SET CHARACTER 146,0,8,8,63,63,191,201,54,54
3148   SET CHARACTER 147,255,73,73,201,255,255,201,54,54
3150   SET CHARACTER 148,0,0,97,127,62,0,15,0,0
3152   SET CHARACTER 149,103,190,156,252,124,124,254,63,63
3154   SET CHARACTER 150,48,56,48,24,60,46,39,35,24
3156   SET CHARACTER 151,0,0,0,48,80,144,18,254,0

```

```

2150 SET CHARACTER 152,12,20,7,58,77,84,68,56,0
3150 SET CHARACTER 153,112,32,224,46,49,253,242,14,0
3162 SET CHARACTER 154,0,193,255,255,1,0,0,0,0
3164 SET CHARACTER 155,0,242,28,254,255,31,47,15,15
3166 SET CHARACTER 156,15,63,63,127,127,187,152,24,24
3168 SET CHARACTER 157,140,254,250,254,247,98,98,98,100
3170 END DEF
3200 DEF Q12 ! Initiatie kleuren van kaarten
3204 RESTORE 3208 ! Kleuren v kaarten
3208 DATA 5,75,162,63,162,255,240,73,167,49,255,0,130,255,180,12,2,91
3212 DATA 2,219,36,211,78,209,4,207,174,5,104,78,98,199
3216 DATA 130,219,251,67,134,145,148,22,134,145,139,88,6,24,180,107,48,209
3220 DATA 2,77,230,104,143,24,4,43,30,176,216,30,46,56
3224 FOR PL=0 TO 31
3228 READ ACHTER,VOOR
3232 LET COLPIC(PL,0)=ACHTER:LET COLPIC(PL,1)=VOOR
3236 NEXT
3240 END DEF
3300 DEF Q41 ! Kies kaart mbv pook
3304 SET LINE MODE 3:SET INK 2
3308 DO
3312 DO
3316 PRINT #101,AT CY+2,CX*2+3:"++"
3320 LET J=JOY(0)
3324 PRINT #101,AT CY+2,CX*2+3:"++"
3328 LOOP WHILE J=0
3332 SOUND PITCH 50,DURATION 1
3336 LET CX=MOD(CX+(J BAND 1)/1-(J BAND 2)/2,8)
3340 LET CY=MOD(CY+(J BAND 4)/4-(J BAND 8)/8,8)
3344 LOOP UNTIL J=16
3348 SET LINE MODE 0
3352 LET Q41=CX+CY*8
3356 END DEF
6999 !
7000 ! Procedures ALGEMEEN
7100 DEF PRKRT(VV,WW$,AK,VK)
7104 LET YY=INT(VV/8)+2:LET XX=MOD(VV,8)*2+3
7108 SET INK AK
7112 PRINT #101,AT YY,XX:"++"
7116 SET INK VK
7120 PRINT #101,AT YY,XX:WW$
7124 END DEF
7128 DEF YEL
7132 FOR T=1 TO 10
7136 SOUND ENVELOPE 1,DURATION 21,PITCH 20
7140 NEXT
7144 END DEF
7148 DEF DOWN
7152 SOUND ENVELOPE 2,DURATION 30,PITCH 30
7156 END DEF
7160 ! Wilfried Hesseling
7164 ! Enterprise 128 - EXOS 2.1
7168 ! Maart 86

```

*** ADVENTURES ***

code	filename	ext	bytes	language	description	received	required
A.001.01	ADVENT	DIR	194304	IS-DOS	Uitgrebreid 'Colossal Cave'	01/04/86	D/E
A.001.02	GEVAAR	BAS	22743	IS-BASIC	Nederlandstalig Adventure	13/04/86	

*** BUSINESS ***

code	filename	ext	bytes	language	description	received	required
B.001.01	SPELL	DIR	156002	IS-DOS	Spelling checker with English dictionary	04/04/86	D/E
B.001.02	KRTN_BAK	DIR	17093	IS-BASIC	Kaartenbak systeem	26/04/86	C
>N> B.001.03	3D_DRAW	DIR	9900	IS-BASIC	Teken program voor Interlace	24/06/86	E
>N> B.001.04	COMPOSER	DIR	13092	IS-BASIC	Muziek invoeren op notenblad	30/07/86	E?
>N> B.001.05	SPRSHEET	DIR	11800	IS-BASIC	Elektronisch rekenblad	30/07/86	E?

*** DEMONSTRATIONS ***

code	filename	ext	bytes	language	description	received	required
D.001.01	NETWORK	BAS	12672	IS-BASIC	Network demo	01/02/86	
D.001.02	PICTURES	DIR	115259	IS-BASIC	Hires pictures	01/04/86	D/E
D.001.03	ENVELOPE	BAS	5263	IS-BASIC	Sound demo	26/04/86	
>N> D.001.04	FIETS	BAS	2118	IS-BASIC	Tekening	24/04/86	
>N> D.001.05	RUN_MAN	BAS	3144	IS-BASIC	Running man	24/06/86	
>N> D.001.06	TIENTJE	BAS	26521	IS-BASIC	Tekening	28/06/86	

*** EDUCATION ***

code	filename	ext	bytes	language	description	received	required
>N> E.001.01	HOLLAND	BAS	21076	IS-BASIC	Ken je Nederland ?	26/04/86	

*** GAMES ***

code	filename	ext	bytes	language	description	received	required
G.001.01	Z80CHESS	DIR	13312	IS-DOS	Chess	01/04/86	D/E
G.001.02	OTHELLO	BAS	17201	IS-BASIC	Othello (reversi)	26/04/86	
G.001.03	WOOPER	BAS	16455	IS-BASIC	Spelletje met sprites	26/04/86	
G.001.04	HAPSLANG	BAS	3368	IS-BASIC	Eet de vliegen	26/04/86	
G.001.05	EXPERT	BAS	3162	IS-BASIC	De computer raad het !	26/04/86	C
>N> G.001.06	ROULETTE	BAS	5867	IS-BASIC	Roulette spel	24/06/86	
>N> G.001.07	AUTO	BAS	867	IS-BASIC	Rij rond circuit	24/06/86	
>N> G.001.08	TRON	BAS	4898	IS-BASIC	Spel met motoren en muren	24/06/86	

*** PROGRAMMING ***

code	filename	ext	bytes	language	description	received	required
>N> P.001.01	BASIC00	DIR	16665	IS-BASIC	Basiccode	26/04/86	
P.001.02	NPS_CBL	DIR	57728	IS-DOS	NPS Micro Cobol ver 2.1	01/04/86	D

*** ROUTINES ***

code	filename	ext	bytes	language	description	received	required
R.001.01	CONVERT	BAS	1780	IS-BASIC	Hex/Decimal/Binary conversion routines	01/02/86	
R.001.02	SORT	BAS	1559	IS-BASIC	Sort routine	01/02/86	
R.001.03	ALLOCATE	BAS	534	IS-BASIC	Allocate bug work-a-round	01/02/86	
R.001.04	XR_HEAD	GEN	1792	DEVPACK	Standard Header for system extensions	01/02/86	

*** TEXTFILES ***

code	filename	ext	bytes	language	description	received	required
T.001.01	MEM_MAP	WPB	9088	WP	Memory map 25-01-86	01/02/86	
T.001.02	BASIC_1	WPB	3712	WP	Hoe BASIC opgeslagen wordt	01/02/86	
T.001.03	CHAR_SET	WPB	4736	WP	Character set table	01/02/86	
>N> T.001.04	BASFOOT	WPB	4180	WP	Fouten in BASIC 2.0	24/06/86	

*** UTILITIES ***

code	filename	ext	bytes	language	description	received	required
U.001.01	COB00	DIR	4470	DEVPACK	Veranderd EDITOR naar 80 columns	01/02/86	E
U.001.02	SD	DIR	2176	IS-BASIC	Sorted directory naar FX-80	01/02/86	D/P
U.001.03	DATA_GEN	BAS	2196	IS-BASIC	Data statement generator	01/02/86	
U.001.04	LU	DIR	20824	IS-DOS	Library Utility	01/04/86	D/E
U.001.05	SQUEEZE	DIR	7040	IS-DOS	File squeezer	13/04/86	D/E
U.001.06	LOCK	DIR	25472	IS-DOS	File Lock/Unlock	13/04/86	D/E
U.001.07	VSLD	DIR	3120	IS-BASIC	Video Save/Load/Dump utilities	13/04/86	
U.001.08	BASIC20	DIR	17577	IS-BASIC	Memory map - IS-BASIC 2.0	26/04/86	
U.001.09	MONITOR	MBS	28890	IS-BASIC	Monitor	11/04/86	
U.001.10	DISSASM	BAS	11266	IS-BASIC	Disassembler	11/04/86	
U.001.11	FONTSY	DIR	125184	IS-DOS	Banner print utility	25/04/86	D/P/E
U.001.12	KBRD	DIR	4082	DEVPACK	Keyboard translator for Wordstar	26/04/86	D/E
U.001.13	SDUMP	DIR	8599	DEVPACK	Screen dump naar EPSON	26/04/86	P
U.001.14	HEXDUMP	BAS	11477	IS-BASIC	Hex dump	02/05/86	

STANDAARD FILE EXTENSIONS

Om files makkelijk te kunnen identificeren wordt er gebruik gemaakt van standaard EXDOS filenames (max 8 characters) met een van de volgende extensions.

<i>ext</i>	<i>beschrijving</i>
.4TH	Forth file
.APP	Application
.ASC	ASCII file
.ASM	Assembler source file
.BAS	Single BASIC file
.C	C source file
.COM	ISDOS command file
.DAT	Data file
.DOC	Document file
.GEN	DEVPAC source file
.H	C header file
.HLP	Help file
.MBS	Multiple BASIC program file
.MRN	Multiple BASIC program file (autostart)
.OVR	Overlay file
.PAS	PASCAL file
.PRT	Print file
.RUN	Autostart BASIC program
.TXT	Text file
.XA	Absolute system extension
.XR	Relocatable system extension
.WP4	40 column Word Processor file
.WP8	80 column Word Processor file
.DIR	Directory containing multiple files