

sinclair

gids

NUMMER 12/JULI-AUGUSTUS 1988

PRIJS fl 6,50 / 130 BF

ONAFHANKELIJK BLAD VOOR SINCLAIR GEBRUIKERS

Basis van BASIC
DEEL 4

+

TIPS & TRUCS VOOR
VERBETERINGEN

LISTINGS SPECTRUM
VALUTA-LIJST
MATRIX-POKER

DRIE M/C UTILITIES
voor Spectrum:

VAR 58000
FILL-routine
CHAN 59000



ZX-81

SNEL &
PRECIES
REKENEN
DEEL 8b

EIGEN
KARAKTERSET
ONTWERPEN
MET:
"KARAKTERS"

QL Exceptions !!

128 K
geheugenorganisatie

MACHINETAAL OP DE SPECTRUM Deel 8

Maandblad (8 x per jaar) verschijnt niet in januari, april, juli, oktober

2e JAARGANG NUMMER 4

LUXE INBOUWKAST VOOR 5 1/4" DRIVES

Systeemkast voor de inbouw van 2 stuks 5 1/4" Disk drives, compleet met voeding en inbouw materiaal.

NIEUW!



179,-

249,-



EXPANSION SYSTEM

Compleet opslagsysteem met Interface 1, microdrive, softwarepakket met Tasword 2, Masterfile, Games Designer, Ant Attack, Introduction (op cartridge) en een lege cartridge. Direct aansluitbaar op de Spectrum 48k/+120k/+2.

DATA RECORDER

voor CBM 64, Atari, MSX en Sinclair Spectrum.

59,-



* Externe voeding f 9.95

CHEETA MK 5 MIDI KEYBOARD

Sluit nu uw Cheetah mini-keyboard aan op uw home-computer met midi uitgang. Enige bijzondere mogelijkheden van dit keyboard zijn:

- 128 programma's vanaf keyboard oproepbaar
- 61 toetsen (5 octaven)
- Octaven shift functies
- Pitch bend Wheel
- Hold functie
- Led display
- Midi out
- volledig polyfonisch
- 16 midi kanalen toekenbaar
- direct aansluitbaar op uw Atari ST



Nu 199,-

Midi Software voor de Atari ST, "Music Studio" f 99,-

Midi kabel voor de Atari ST f 9.95

Midi interface voor Amiga 500 f 179,-

Mini/Midi interface voor Spectrum 48k/+128/+2 incl. software f 79,-

KEMPSTON MUIS SYSTEEM VOOR DE SPECTRUM 48/128k

Dit muissysteem bestaat uit:

- Prof. Tekkenprogramma Art Studio
- Gratis muismat
- Muisinterface
- Muis

249,-



3,5" DISK DRIVE 1 MEGABYTE

399,-

3,5" Diskdrive met standaard Shugart aansluiting, compleet met kast en voedingsunit.

CARTRIDGES

10 cartridges, inclusief opbergbox	79,00
10 cartridges	75,00
4 cartridges, inclusief opbergmap	30,00
Opbergbox	14,95
Microdrive	99,00
Back to back connector, voor aansluiting 2e microdrive	14,50
Kempston joystick interface	39,00

NIEUWE SPECTRUM SOFTWARE

Garfield	37,50	Arkanoïd	34,50
California		Road wars	39,50
Games	34,50	Freddy	
Gauntlet II	34,50	Hardest	34,50
Rampage	39,50	Super	
Magnificent		hang on	39,50
seven	39,50	Fire fly	32,50
Games set and		Air combat emu-	
match	59,50	lator II	39,50
Platoon	39,50	Artist II	
Outrun	34,50	(128k v)	79,50
Star wars	39,50	OCP - Full Screen	
Solid Gold	39,50	ed/ass	19,95



SPECTRUM 128/+2 399,-

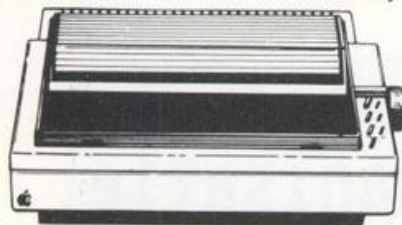
Een van de meest verkochte computers ter wereld compleet met ingebouwde cassetterecorder, rs.232/midi uitgang, joysticks aansluiting en geschikt voor TV en monitor.

Ned. Handleiding f 32.90
+ 2 Joystick f 39.50

APPLE SCRIBE PRINTER

Dot-matrix A4 printer met letterkwaliteit geschikt voor thermisch en normaal papier.
• Friction- en pinfeed • diverse lettertypes
• RS-232 • draft mode 80 cps
• printerstandaard.

199,-



3 CARBONLINTEN

29,-

APPLE MONOCHROOM MONITOR

Compacte monochroom monitor compleet met metalen monitor-voet.

159,-



VTX-5000

COMPLEET VIDITEL SYSTEEM VOOR DE SPECTRUM 48k/+

Direct aansluitbaar op uw Spectrum en telefoonleiding, dus geen modem en interface meer nodig. Software in rom. Compleet met user to user software.

149,-



**PRIJS-ARTIKEL
WIJZIGINGEN VOORBEHOUDEN**

COMPUTER SHOP

Zwartjanstraat 51
3035 AL Rotterdam
☎ 010 - 467 06 77

era

Colofon

De Sinclair gids is een uitgave van Terminal Software Publicaties, Postbus 111, 5110 AC Baarle Nassau. Hij verschijnt acht maal per jaar.

Hoofredactie : P. Pauwels
Tel.: 04257-9161

Vaste medewerkers :

H. van Abbe (ZX81/Z80)
F. Brands (QL)
R.-J. Donkers (BASIC, Nieuws)
W. Dijkgraaf (QL, 68000)
M. Feenstra (QL, Spectrum)
M. Hellema (Z80)
W. Lageman (Spectrum, C, Z80)
A. Plomp (QL, 68000)
A. Versluis (Nieuws, Test)
A.v.d. Wijdeven (QL, Spectrum)
E. Zwart (Spectrum, Z80)

ABONNEMENTEN :

NEDERLAND : f44,00 per jaar (8 nrs).
Abonnementsgeld te voldoen door f44,00
over te maken op giro 5109074 t.n.v.
De Sinclair Gids, Baarle Nassau.

BELGIË : 860 F per jaar (8 nrs). te
voldoen door 860 F over te schrijven
op rekening 000-1592677-34 t.n.v. De
Sinclair Gids, Baarle Nassau

Abonnementen lopen tot wederopzegging.
Opzegging kan uitsluitend per brief
geschieden en wel tot 1 maand voordat
het huidige abonnement afloopt. Nadien
vindt automatisch verlenging plaats
voor 1 jaar. Bij automatisch verlengen
dient u te betalen door middel van het
u toegezonden betaaldocument.

ADRESWIJZIGINGEN 3 weken van te voren
opgeven o.v.m. oude en nieuwe adres.

LOSSE NUMMERS : f6,50 of 130 F.
Bestellen losse nummers uitsluitend
per vooruitbetaling op giro (NL) of
postrekening (B).

INGEZONDEN PROGRAMMA'S

De inzender van programma's voor publicatie verklaart dat de programma's eigen werk zijn en vrij van rechten. Hij/zij vrijwaart de Sinclair gids en uitgeverij Terminal Software Publicaties voor enige aanspraak van derden op het copyright van desbetreffende programma's. Door inzending van een programma verklaart de inzender dat hij deze clause onderschrijft.

COPYRIGHT

Het is niet toegestaan (delen van) artikelen over te nemen zonder de voorafgaande schriftelijke toestemming van de uitgever. Programma's mogen uitsluitend voor eigen gebruik worden overgenomen.

DISTRIBUTIE NEDERLAND :
Betapress, Bgm. Krollaan 14, Gilze.
DISTRIBUTIE BELGIË :
AMP/PVD NV, Kl. Eilandstr. 1, Brussel

Edit

In dit nummer publiceren we een paar artikelen waar velen op hebben zitten wachten, te oordelen naar de vele telefoontjes die we ontvingen. Robert-Jan Donkers heeft de tijd gevonden om een extra lange aflevering van "de basis van BASIC" te schrijven, die weer heel wat nuttige tips voor beginnende programmeurs bevat.

René Uittenbogaard is onze redactie komen vervoegen als half-vaste medewerker; hij zal zorgen voor de 128K-gebruikers onder u. In dit nummer vindt u naast zijn eerste bijdrage in die reeks, ook nog drie Spectrum-utilities van zijn hand, die bewijzen dat we aan hem best een waardevolle medewerker zullen hebben. Welkom in de ploeg, René.

Érg mooi in dit nummer, vinden wij wel het QL-artikel van Wim Dijkgraaf over exceptions. Serieuze programmeurs hebben er een kluit aan. Anderen kunnen er wellicht zin door krijgen om serieus met een QL aan de slag te gaan.

Op de eerste bladzijden staan een aantal tips, die ons door lezers werden doorgegeven. Een zeer lovenswaardig initiatief. Het komt vaak voor, dat iets wat voor de een zó vanzelfsprekend is, voor de ander een flinke tijd puzzelen kost. Hebt u zelf iets handigs ontdekt, of weet u iets dat in geen boek of tijdschrift te vinden is, laat het ons dan weten. Alvast bedankt.

de redactie

Inhoud

Nieuws	2
Tips en hints allerlei	2, 18
Oeps ! Blunders en miskleunen	2
Valuta-lijst voor Spectrum	5
Matrix-poker, een origineel spel	8
Gemakkelijk een nieuwe tekenset ontwerpen	12
Opus utility	14
Drie utilities voor de Spectrum	16
Snel en precies rekenen op de ZX81, deel 8b ...	19
Cursus machinetaal op de Spectrum, deel 8	21
128K : organisatie van het geheugen	25
Software Service	27
De basis van BASIC, deel 4	29
QL : uitzonderlijke toestanden met exceptions .	34
In/Out : gezocht en aangeboden	37

Nieuws - Hints -

Tips - Oeps !

128K + Cheetah Spectrum

H. de Roo uit Huizen brengt troost in bange dagen voor de eigenaars van een **Cheetah Spectrum** die overgestapt zijn op een **128 K-Spectrum**. Die combinatie zou namelijk niet meer werken. Mis poes : die combinatie werkt wél ! Goed, alleen in 48K-mode, maar het werkt. De methode is even simpel als doeltreffend :

- leen een gewone 48K (of gebruik je eigen oude trouwe) en laad de software
- save de software met een (al dan niet geleende) Multiface 1

en zie : de kopie werkt op de 128 K, in 48K-mode !
Misschien lukt die truc ook wel bij andere software ?
Bedankt, Henc de Roo.

128K geluid via monitor

J. Bömer uit Terborg heeft een goeie tip voor **128K**-bezitters die **geluid** uit hun **monitor** willen halen. Plus 2-bezitters kunnen het geluid gewoon van de **SOUND**-uitgang aftakken, maar de "oude" 128 K heeft die niet. Maar die heeft wél de gewone **MIC** en **EAR**-stekkers. Om geluid naar de monitor te sturen, is het voldoende om een kabeltje te maken met aan de ene kant een 3 mm-plugje dat in de **MIC**-ingang past, en aan de andere kant een stekker die in de **AUDIO**-ingang van de monitor past. Verbinden, en klaar is kees. Het signaal is voldoende om de monitor aan te sturen. Bedankt, heer Bömer !

Shadecopy : extra info

In nummer 10 publiceerden we het mooie **Shadecopy**. Voor degenen die het programma met hun printer niet aan de praat krijgen, kan de volgende informatie veel ergernis voorkomen :

de printer moet Epson-codes kennen voor de grafische print-manier, en wel : **ESC "L"** (dubbele dichtheid). **Seikosha's** en **Apples** wijken daar over het algemeen van af.

Ook dient de printer het commando **ESC "A"** n te kennen, dat wordt gestuurd om de regelafstand op **n/72 inch** in te stellen.

Sprites op de QL

In nummer 11, op bladzijde 28, staat een typfout in de programma-regel onderaan : er staat

```
PUT A,0,0,1
```

en dat moet uiteraard zijn :

```
PUTSPRITE A,0,0,1
```

EXIT Komin

Tot onze spijt en die van vele van onze lezers, te merken aan de hoeveelheid telefoontjes die we daarover ontvingen, is de firma **Komin** failliet verklaard. Velen ergerden zich aan het feit, dat ze voor noppes naar **Eindhoven** gingen, of bléven bellen, dit naar aanleiding van de advertentie in nummer 11 van de gids.

Begrijpelijke ergernis, maar het was niet te voorkomen. De gids kwam net van de pers, toen we het nieuws over **Komin** ontvingen. Omdat we al over tijd waren met de gids, wilden we hem zo snel mogelijk de deur uit, zodat we ook beslisten om géén apart velletje met de mededeling in te leggen. Vervelend voor de mensen die voor niets naar **Komin** reisden. Nog veel vervelender voor **Komin** zelf.

Geleen gebruikersgroep

In **Geleen** is er een gebruikersgroep opgericht. Voor de geïnteresseerden onder de lezers : neem contact op met **E. J. W. Bauwens** op nummer **04490-76529** of met heer **Kuhles** op nummer **04490-52306**.

Succes toegewenst !

Kortere programma's

Bij bepaalde programma's (bv. databases) is het de kunst om zoveel mogelijk geheugen voor de variabelen te reserveren, en dus het programma zo kort mogelijk te schrijven.

Getallen in een programma nemen veel geheugen in beslag: vijf bytes per getal, plus een "marker" (**CHR\$ 14**), plus de tekens die het getal vormen. Het getal 255 kost dus $3+1+5=9$ bytes. Een manier om dit korter te schrijven is : **VAL"255"** : 6 bytes. Wanneer we er van uitgaan dat een beetje programma toch al gauw enkele tientallen cijfers kan bevatten, kan het de moeite lonen om alle getallen te "VALlen".

Wanneer in het programma vaak dezelfde getallen worden gebruikt, loont het soms de moeite om een variabele te gebruiken, die dat getal dan vervangt in de listing. Bij de start van het programma moeten die variabelen dan wel met de waarden worden ingevuld. Het vergt dus wel wat rekenwerk, om na te gaan of het écht de moeite waard is: het programma wordt enerzijds korter, doordat in plaats van de minimaal 7 bytes van het getal, de twee bytes van de naam van de variabele komen (een naam met 1 letter biedt maar 26 mogelijkheden), maar anderzijds wordt het programma ook langer door de regel(s) waarop de variabelen met hun waarde worden gevuld. Plus, de variabelen nemen uiteraard ook geheugenruimte in beslag.

Voor de getallen 32 tot 255 is er een alternatief voor de VAL-methode, die het voordeel heeft dat ze nog sneller werkt ook: gebruik CODE. In plaats van bijvoorbeeld:
LET a=VAL "65"(8 bytes)
schrijven we:
LET a=CODE "A" (7 bytes,
en nog sneller ook).
Dit werkt voor alle tekens van de Spectrum, vanaf de spatie tot en met het keyword COPY. Deze methode lukt niet op een Spectrum 128, omdat de keywords letter voor letter getypt moeten worden. Tenzij het programma in 48K-mode wordt geschreven, en dan in 128K-mode ingeladen.

Software ruilbeurs

E. Dullaard uit Zevenhuizen wil graag een ruilbeurs voor software op touw zetten. Hij ziet het zo: stel, dat iemand een stuk software heeft dat hij niet meer gebruikt. Die persoon stuurt het programma naar het adres onderaan dit stukje, samen met f4,- (voor de porto van het terug te sturen programma).

De ruil gebeurt op een objectieve basis: voor software uit 1982, krijgt de inzender software uit diezelfde periode terug.

De ruilbeurs staat alleen open voor originele software. Kopieën worden niet aanvaard.

Voor meer informatie, stuur een aan jezelf geadresseerde envelop naar:

SSSH, Groeneweg 11
2761 DD Zevenhuizen

Diezelfde E. Dullaard heeft een tip voor 128K-gebruikers met een Opus. Hij raadt die aan om een verlengstrook te kopen voor de verbinding tussen Opus en computer, omdat anders de TV-kabel tussen de drive en de computer kapot gedrukt kan worden. Wat hij heeft ervaren.

Mcode efficiënter

Rob Udo uit Eindhoven heeft een tip voor beginnende (en naar ons idee ook gevorderde) machinetaal-programmeurs. In nummer 6, stond het spel "Shootin' in code", met daarbij als commentaar, dat het bedoeld was om een aantal tips te geven over efficiënt gebruik van code. Hij plaatst twee kanttekeningen. Het is nóg efficiënter, zegt hij, en dat zijn we helemaal met hem eens, om alle 'CP 0' te vervangen door 'AND A': dat scheelt per keer 1 byte. Bovendien kunnen alle 'LD A,0' vervangen worden door 'XOR A', wat eveneens 1 byte korter is. Bedankt, Rob Udo.

QL real real time

De QL-gebruikersgroep Hasselt biedt een uitbreiding voor de QL aan, die menig gebruiker interessant zal vinden: een echte klok, d.w.z. eentje die niet de tijd verliest wanneer de QL wordt uitgeschakeld. Vandaar de kop stukje: een échte 'real time' klok. Het dingetje wordt als bouw pakket geleverd, samen met een (why English?) handleiding. Volgens de handleiding is de klok in te bouwen op een half uurtje, zelfs door iemand met twee linkerhanden.

Geïnteresseerden hoeven alleen een cheque van 930 F of f45,- te sturen naar:
QL-gg Limburg
Groenstraat 43
B-3511 Hasselt
en de klok wordt per omgaande toegestuurd.

Diezelfde gebruikersgroep meldde ons het bestaan van een **databank** voor QL-gebruikers: Microcon, tel. 011/254736 (B), 24 u op 24, 1200/75 baud viewdata 300/300 baud BBS

QL-tip

Nu we het toch over QL hebben, een tip voor QLers die geen Toolkit II bezitten. De QL overschrijft niet automatisch een programma op microdrive, wanneer we een gewijzigde versie ervan onder dezelfde naam willen wegschrijven. Met Toolkit II kan dat wel, omdat die netjes meldt dat dit programma reeds op de cartridge staat, en vraagt of het overschreven mag worden.

Om te vermijden dat we telkens de hele rimram moeten typen:
'delete mdv1_programmanaam'
gevolgd door
'save mdv1_programmanaam'
bedachten we het volgende.

Maak een procedure, op regels 32700 en hoger, die dat karwei van u overneemt, bv.:

```
32750 DEF PROCEDURE WSAVE (x$)
32751 DELETE 'mdv1_&x$
32752 SAVE 'mdv1_&x$
32753 END DEFine
```

en zet die op cartridge onder de naam 'boot'. Bij het opstarten van de QL wordt de procedure automatisch in het geheugen geladen, zodat we nu een gewijzigde versie van een programma op de cartridge kunnen zetten met het commando WSAVE 'naam_van_programma'. De allereerste keer moet het te bewerken programma geladen worden met MERGE, omdat anders de procedure wordt gewist. De volgende keren kan dat gewoon met LOAD, omdat de procedure dan al in het programma zelf is opgenomen.

Wafadrive cartridges

In ons kleine Sinclairland is het voor een aantal gebruikers een hard bestaan. Neem bijvoorbeeld de Wafadrive-bezitters. Hoe komen zij aan voldoende wafers etc. In Nederland is helaas niets verkrijgbaar. Gooi Uw wafadrive

niet weg. In Engeland zijn nog spullen te koop:

Wafers 32K £ 3.49
Wafers 16K £ 2.99
Wafadrive Printerkaart: £ 9.95
Wafadrive Toolkit : £ 9.95

Besteladres:

Capri Marketing Ltd.
24A White Pit Lane
Flackwell Heath nr. High Wycombe
Bucks. HP10 9HR
United Kingdom

Wij hopen dat U hier zult slagen.

Patience - OEPS !

In nummer 11 publiceerden we de listings van het spel "Patience", door W. Lageman.

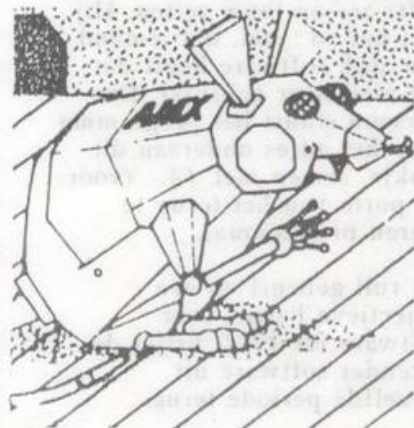
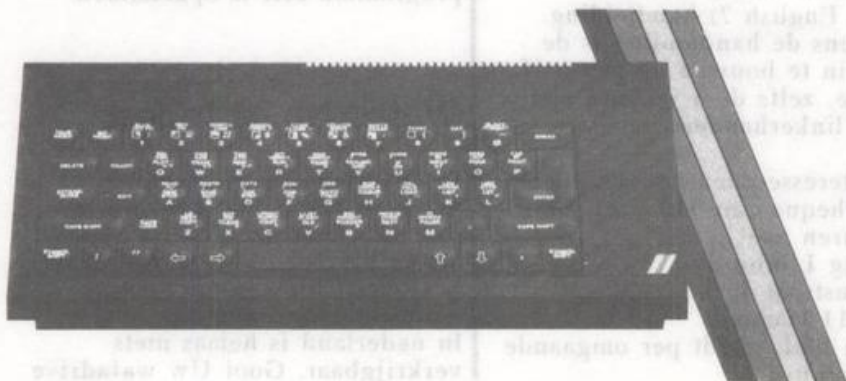
Onze montage-man vond het na een tijd blijkbaar goed geweest, en heeft een deel weggelaten: listing nummer 6. Die volgt hieronder. De montage-man belooft beterschap.

In de gepubliceerde listings van Patience zaten overigens ook nog een paar slordigheden. Op regels 3150, 3160, 3290, 3300, 3310 en 3320 moeten de vraagtekens tussen de aanhangstekens worden vervangen door (in dezelfde volgorde) de UDG's A, M, N, O, P en Q. Dan zal het programma werken zoals het hoort.

Let overigens ook goed op, dat op regels 1260 en volgende, het correcte aantal spaties in de string worden getypt, zoniet gaat het programma de mist in.

```
1000 DATA "pat.chr",60000,768
1010 DATA "0000000000000000:F5"
1020 DATA "1818181818001800:91"
1030 DATA "3333660000000000:D6"
1040 DATA "00367F36367F3600:EC"
1050 DATA "183E603C06067C18:B2"
1060 DATA "00666C1830664600:F0"
1070 DATA "38646070E4C6CE78:96"
1080 DATA "1818300000000000:9C"
1090 DATA "0C18181818181800:EE"
1100 DATA "3018181818181830:42"
1110 DATA "000024187E182400:52"
1120 DATA "000018187E181800:44"
1130 DATA "0000000000181830:CE"
1140 DATA "0000007E00000000:F6"
1150 DATA "0000000000181800:82"
1160 DATA "00040C1830604000:86"
1170 DATA "003C666E76663C00:C2"
1180 DATA "0018381818183C00:76"
1190 DATA "003C660C30627E00:6C"
1200 DATA "007E4C180C663C00:48"
1210 DATA "000C1C3C6C7E0C00:1C"
1220 DATA "007E627C06663C00:D0"
1230 DATA "003C607C66663C00:F6"
1240 DATA "007E460C18303000:28"
1250 DATA "003C663C66663C00:D0"
1260 DATA "003C663E060C3800:1E"
1270 DATA "0018180000181800:5C"
1280 DATA "0018180000181830:95"
1290 DATA "0000060C180C0600:4B"
1300 DATA "0000007C007C0000:13"
1310 DATA "0000603018306000:5D"
1320 DATA "003C660C18001800:0D"
1330 DATA "003C463676663C00:0B"
1340 DATA "00183C66667E6600:49"
1350 DATA "007C363C36367C00:25"
1360 DATA "003E666060663C00:5F"
1370 DATA "00FC66666666FC00:F5"
1380 DATA "00FE62786062FE00:09"
1390 DATA "00FE62786060F000:03"
1400 DATA "007CC4C0DEC7C00:AB"
1410 DATA "0066667E66666600:09"
1420 DATA "007E181818187E00:EF"
1430 DATA "001E0C0C0C0C7800:25"
1440 DATA "00F66C78786CF600:61"
1450 DATA "00F060606062FE00:27"
1460 DATA "00C6EEFED6C6C600:D7"
1470 DATA "00E7767E6E66E700:61"
1480 DATA "003C666666663C00:E1"
1490 DATA "00FC66667C60F000:73"
1500 DATA "003C666666683600:F1"
```

```
1510 DATA "00FC66667C6CF600:99"
1520 DATA "003E603C06067C00:5B"
1530 DATA "007E5A1818183C00:5F"
1540 DATA "0066666666663C00:48"
1550 DATA "00666666663C1800:D8"
1560 DATA "00C6C606FEEEC600:3C"
1570 DATA "00E7663C3C66E700:40"
1580 DATA "00C3623418183C00:F9"
1590 DATA "00FE8C183062FE00:74"
1600 DATA "003C303030303C00:80"
1610 DATA "00406030180C0400:4A"
1620 DATA "003C0C0C0C0C3C00:04"
1630 DATA "00183C7E18181800:80"
1640 DATA "00000000000000FF:6F"
1650 DATA "003C62F0607AEE00:D4"
1660 DATA "000078CCCCC7600:DA"
1670 DATA "E0607C6666667C00:FC"
1680 DATA "00003E6660663C00:40"
1690 DATA "0E0C7CCCCC7600:18"
1700 DATA "00003C667E603E00:6C"
1710 DATA "0E1818183E181818:92"
1720 DATA "0C386C78207C663C:2A"
1730 DATA "E0607C6666667C00:B5"
1740 DATA "1800381818183C00:A8"
1750 DATA "06000E060606663C:A6"
1760 DATA "E0606C78786CF600:EC"
1770 DATA "3818181818183C00:DE"
1780 DATA "0000FEDBDBDB800:6E"
1790 DATA "0000DC6666667C00:11"
1800 DATA "00003C6666663C00:BB"
1810 DATA "0000FC6666667C00:B3"
1820 DATA "00007ECCCC7C0C0E:D3"
1830 DATA "00006E3330307800:A8"
1840 DATA "00003E603C067C00:95"
1850 DATA "18187E1818180E00:47"
1860 DATA "0000EF6666663B00:AA"
1870 DATA "0000E76274381000:5E"
1880 DATA "0000C6D6D67C6C00:BF"
1890 DATA "0000663C183C6600:C7"
1900 DATA "00006666663E067C:69"
1910 DATA "00007E4C18327E00:13"
1920 DATA "0C18183018180C00:31"
1930 DATA "0018181818181800:23"
1940 DATA "3018180C18183000:69"
1950 DATA "324C000000000000:25"
1960 DATA "3C42DBD3D3DB423C:11"
1970 DATA "":B9"
```



Valuta-lijst

A. Goossen

Voor mensen die wel eens in het buitenland verblijven is het omrekenen van de prijzen in de winkels aldaar ter vergelijking met de prijzen in Nederland wel eens lastig.

Al enkele malen heb ik het gemak van een omrekentabel ondervonden, waarop snel is af te lezen hoe groot een bepaald bedrag omgerekend in Nederlands geld zou zijn. Daar ben je als Nederlander in het buitenland toch wel eens nieuwsgierig naar, vooral als je prijsbewust wilt kopen.

Bijgaand programma (dat gebruik maakt van de bekende 42/51 kolomsroutine van Herman Ament) vormt een of meer omrekentabellen, die automatisch op de printer (ZX of Seikosha 50) worden afgedrukt. Om een afdruk op een grote printer te maken, moet u het commando COPY vervangen door de aanroep van een of andere screendump-routine, die dan bij in het geheugen moet geladen worden. Het programma begint met een kleine toelichting.

De tweede listing bevat de data om de code 42/51 te maken. De eerste regels zetten alle cijfers in het geheugen, en verwittigen wanneer er een typfout op een regel staat.

```
10 REM OMREKENTABEL
20 REM VAN VREEMDE VALUTA
30 REM NAAR GULDEN
40 REM
50 REM UITVOER NAAR SCHERM
60 REM EN PRINTER
70 REM
140 REM 42/51 kolomsroutine
150 REM van Herman Ament
160
170 BORDER 0: PAPER 0: CLS : IN
K 9
```

```
180 PRINT "          O M R E
K E N T A B E L "
190 PRINT AT 10,8;"T O E L I C
H T I N G ? ( j / n )"
200 LET i$=INKEY$: IF i$="" THE
N GO TO 200
210 IF i$="j" THEN CLS : GO TO
240
220 IF i$="n" THEN CLS : GO TO
360
230 GO TO 200
240 PRINT "          O M R E
K E N T A B E L "
250 PRINT "'Dit programma reke
nt een door uzelf te bepalen bui
-''tenlandse munteenheid om in
Nederlands geld. Het"
260 PRINT "'vormt een of meer t
abellen in stappen van een of"
270 PRINT "'tien. Geldbedragen
zijn in het buitenland dikwijls"
280 PRINT "'moeilijk snel om te
rekenen en het bij de hand heb-
"
290 PRINT "ben van een omreke
ntabel kan dan erg handig zijn."
300 PRINT "'U moet invoeren: a.
stappen van 1 of 10"
310 PRINT "          b.
afkorting munteenheid"
320 PRINT "          c.
de huidige koerswaarde"
330 PRINT "'Als de printer is i
ngeschakeld worden de tabellen"
340 PRINT "'automatisch uitgepr
int. Druk op een toets....."
350 PAUSE 0
360 REM ** STEP 1 of STEP 10 **
370 CLS
380 PRINT AT 4,8;"VAN VREEMDE V
ALUTA NAAR GULDEN"
390 PRINT AT 8,8;"Kies :";AT 10
,8;"1. In stappen van 1";AT 12,8
;"2. In stappen van 10"
400 PRINT AT 16,8;"Toets 1 of 2
"
410 LET k$=INKEY$: IF k$="" THE
N GO TO 410
420 IF k$="1" THEN LET la=59: G
O TO 450
430 IF k$="2" THEN LET la=590:
GO TO 450
440 GO TO 410
450
460 REM ** OPBOUW SCHERM **
470
480 CLS : LET ve=0: LET ee=0
490 PRINT AT 0,9;
```

```
500 IF k$="1" THEN FOR i=ee TO
la-9 STEP 10: PRINT i;" ";
NEXT i
510 IF k$="2" THEN FOR i=ee TO
la-90 STEP 100: PRINT i;" ";
: NEXT i
520 PLOT 0,164: DRAW 255,0
530 PLOT 15,164: DRAW 0,-160
540 PRINT AT 1,0;
550 IF k$="1" THEN FOR i=0 TO 9
: PRINT 'TAB 2-LEN STR$ i;: NEX
T i
560 IF k$="2" THEN FOR i=0 TO 9
0 STEP 10: PRINT 'TAB 2-LEN STR$
i;: NEXT i
570 NEXT i
580
590 REM ** HOOFDPROGRAMMA **
600
610 LET h=8: LET v=0: LET k=1
620 IF ve=1 THEN GO TO 710
630 INPUT " Munteenheid : ";
LINE m$: IF m$="" OR LEN m$>5 TH
EN GO TO 630
640 FOR j=1 TO LEN m$
650 IF m$(j)<"." OR m$(j)>". " A
ND m$(j)<"A" OR m$(j)>"z" THEN G
O TO 630
660 IF CODE m$(1)>90 THEN LET m
$(1)=CHR$ (CODE m$(1)-32)
670 NEXT j
680 PRINT AT 0,0;m$
690 INPUT (" 1 ";m$;" = f ");
LINE n$: IF n$<"." OR n$>". " AN
D n$<"0" OR n$>"9" THEN GO TO 69
0
700 LET w=VAL n$
710 IF k$="1" THEN FOR i=ee TO
la
720 IF k$="2" THEN FOR i=ee TO
la STEP 10
730 LET s$=STR$ (i*w)
740 LET v=v+2: GO SUB 960: PRIN
T AT v,h-4;w$ AND VAL w$<10000:
IF LEN w$=8 THEN STOP
750 IF v=20 AND h=k*8 THEN LET
k=k+1: LET v=0: LET h=k*8
760 NEXT i
770
780 REM * UITVOER NAAR PRINTER
*
790
800 IF ve=0 THEN LPRINT "OMREKE
NTABEL ( ";m$;" naar Gld.)": LPR
INT
810 COPY : LPRINT
820
830 REM ** TABEL VERDER ? **
840
```



```

850 PRINT #0;AT 1,3;"Tabel verd
er ? (j/n)"
860 LET i$=INKEY$: IF i$="" THE
N GO TO 860
870 IF i$="j" THEN LET ve=1: GO
TO 900
880 IF i$="n" THEN STOP
890 GO TO 860
900
910 REM ** VERDER **
920
930 IF k$="1" THEN LET ee=ee+60
: LET la=la+60: CLS : PRINT AT 0
,0;m$:TAB 6;(" " AND i<120);("
" AND i>=120 AND i<1020):: FOR i
=ee TO la-9 STEP 10: PRINT i;"
"+(" " AND i<90)+(" " AND (i>
=90 AND i<990)): NEXT i: GO TO
520
940 IF k$="2" THEN LET ee=ee+60
0: LET la=la+600: CLS : PRINT AT
0,0;m$:TAB 6;(" " AND i<1200)::
FOR i=ee TO la-90 STEP 100: PRI
NT i;" "+(" " AND i<900):: NE
XT i: GO TO 520
950 STOP
960
970 REM ** 2 CIJFERS ACHTER DE
KOMMA EN AFRONDEN **
980
990 LET t$=STR$ (INT VAL s$)
1000 FOR j=1 TO (4-LEN t$): LET
t$=" "+t$: NEXT j
1010 LET g$=STR$ (INT ((VAL s$-V
AL t$)*100+0.5)): LET t$=t$+"."
1020 IF VAL g$=0 THEN GO TO 1060
1030 FOR j=1 TO LEN g$: IF g$(j
TO j)=". " THEN GO TO 1050
1040 NEXT j
1050 LET g$=g$(1 TO (j-1))
1060 IF VAL g$<10 THEN LET g$="0
"+g$
1070 LET w$=t$+g$
1080 RETURN
1090
1100 REM *****
1110
1120 CLEAR 59999: LOAD "code 42/
51"CODE
1130 RANDOMIZE 2: RANDOMIZE USR
6e4: PRINT CHR$ (2)
1140 GO TO 100
9000
9010 REM *****
9020
9030 SAVE "Valuta" LINE 1120: PO
KE PEEK 23631+256*PEEK 23632+2,1
81: SAVE "code 42/51"CODE 60000,
1601
9040 STOP

```

```

10 CLEAR 59999: LET a=60000: F
OR r=100 TO 2100
20 LET som=0: FOR b=0 TO 7: RE
AD d: LET som=som+d: POKE a,d: L
ET a=a+1: NEXT b
30 PRINT AT 0,0;"Regel ";r;

```

```

40 READ x: IF x<>som THEN PRIN
T " bevat een fout": STOP
50 NEXT r
60 SAVE "code 42/51"CODE 60000
,1601
99 STOP
100 DATA 237,91,83,92,1,28,0,42
,574
110 DATA 79,92,9,237,82,56,10,2
35,800
120 DATA 43,1,5,0,205,85,22,24,
385
130 DATA 231,42,79,92,1,25,0,9,
479
140 DATA 17,174,238,6,2,115,35,
114,701
150 DATA 35,17,196,21,16,247,54
,83,669
160 DATA 58,118,92,230,15,135,1
98,22,868
170 DATA 111,38,92,54,26,35,54,
0,410
180 DATA 205,107,13,201,51,6,19
2,50,825
190 DATA 21,1,0,0,0,0,0,0,22
200 DATA 0,0,0,16,16,16,16,0,64
210 DATA 16,0,0,40,40,0,0,0,96
220 DATA 0,0,0,40,124,40,40,124
,368
230 DATA 40,0,0,24,60,40,56,20,
240
240 DATA 60,24,0,36,36,8,16,36,
216
250 DATA 36,0,0,16,40,16,52,40,
200
260 DATA 52,0,0,16,32,0,0,0,100
270 DATA 0,0,0,16,32,32,32,32,1
44
280 DATA 16,0,0,32,16,16,16,16,
112
290 DATA 32,0,0,0,20,8,60,8,128
300 DATA 20,0,0,0,8,8,28,8,72
310 DATA 8,0,0,0,0,0,0,8,16
320 DATA 8,16,0,0,0,0,60,0,84
330 DATA 0,0,0,0,0,0,0,48,48
340 DATA 48,0,0,0,4,8,16,32,108
350 DATA 0,0,0,24,44,44,52,52,2
16
360 DATA 24,0,0,8,24,8,8,8,80
370 DATA 28,0,0,24,36,4,24,32,1
48
380 DATA 60,0,0,24,36,8,4,36,16
8
390 DATA 24,0,0,8,24,40,40,60,1
96
400 DATA 8,0,0,60,32,56,4,36,19
6
410 DATA 24,0,0,24,32,56,36,36,
208
420 DATA 24,0,0,60,4,4,8,16,116
430 DATA 16,0,0,24,36,24,36,36,
172
440 DATA 24,0,0,24,36,36,28,4,1
52
450 DATA 24,0,0,0,0,16,0,0,40
460 DATA 16,0,0,0,16,0,0,16,48
470 DATA 16,32,0,0,8,16,32,16,1
20
480 DATA 8,0,0,0,0,60,0,60,128
490 DATA 0,0,0,0,32,16,8,16,72
500 DATA 32,0,0,24,36,8,8,0,108
510 DATA 8,0,0,24,36,44,44,32,1

```

```

88
520 DATA 28,0,0,24,36,36,60,36,
220
530 DATA 36,0,0,56,36,56,36,36,
256
540 DATA 56,0,0,24,36,32,32,36,
216
550 DATA 24,0,0,48,40,36,36,40,
224
560 DATA 48,0,0,60,32,56,32,32,
260
570 DATA 60,0,0,60,32,56,32,32,
272
580 DATA 32,0,0,24,36,32,44,36,
204
590 DATA 24,0,0,36,36,60,36,36,
228
600 DATA 36,0,0,28,8,8,8,8,96
610 DATA 28,0,0,4,4,4,36,36,112
620 DATA 24,0,0,36,40,48,48,40,
236
630 DATA 36,0,0,32,32,32,32,32,
196
640 DATA 60,0,0,36,60,60,36,36,
288
650 DATA 36,0,0,36,52,60,44,36,
264
660 DATA 36,0,0,24,36,36,36,36,
204
670 DATA 24,0,0,56,36,36,56,32,
240
680 DATA 32,0,0,24,36,36,52,44,
224
690 DATA 28,0,0,56,36,36,56,40,
252
700 DATA 36,0,0,24,32,24,4,36,1
56
710 DATA 24,0,0,28,8,8,8,8,84
720 DATA 8,0,0,36,36,36,36,36,1
88
730 DATA 24,0,0,36,36,36,36,24,
192
740 DATA 24,0,0,36,36,36,60,60,
252
750 DATA 36,0,0,36,20,24,24,40,
180
760 DATA 36,0,0,34,34,20,8,8,14
0
770 DATA 8,0,0,60,4,8,16,32,128
780 DATA 60,0,0,56,32,32,32,32,
244
790 DATA 56,0,0,0,32,16,8,4,116
800 DATA 0,0,0,56,8,8,8,8,88
810 DATA 56,0,0,8,28,28,8,8,136
820 DATA 8,0,0,0,0,0,0,0,8
830 DATA 0,126,0,12,20,56,16,16
,246
840 DATA 60,0,0,0,24,4,28,36,15
2
850 DATA 28,0,0,32,32,56,36,36,
220
860 DATA 56,0,0,0,28,32,32,32,1
80
870 DATA 28,0,0,4,4,28,36,36,13
6
880 DATA 28,0,0,0,24,36,56,32,1
76
890 DATA 28,0,0,12,16,24,16,16,
112
900 DATA 16,0,0,0,28,36,36,28,1
44
910 DATA 4,24,0,32,32,56,36,36,

```


220
 920 DATA 36,0,0,8,0,24,8,8,84
 930 DATA 28,0,0,4,0,4,4,4,44
 940 DATA 36,24,0,32,40,48,48,40
 ,268
 950 DATA 36,0,0,16,16,16,16,16,
 116
 960 DATA 12,0,0,0,36,60,36,36,1
 80
 970 DATA 36,0,0,0,56,36,36,36,2
 00
 980 DATA 36,0,0,0,24,36,36,36,1
 68
 990 DATA 24,0,0,0,56,36,36,56,2
 08
 1000 DATA 32,32,0,0,28,36,36,28,
 192
 1010 DATA 4,6,0,0,28,16,16,16,86
 1020 DATA 16,0,0,0,24,32,24,4,10
 0
 1030 DATA 56,0,0,16,56,16,16,16,
 176
 1040 DATA 12,0,0,0,36,36,36,36,1
 56
 1050 DATA 24,0,0,0,36,36,36,24,1
 56
 1060 DATA 8,0,0,0,36,36,36,60,17
 6
 1070 DATA 36,0,0,0,36,20,24,40,1
 56
 1080 DATA 36,0,0,0,36,36,36,28,1
 72
 1090 DATA 4,24,0,0,60,4,8,16,116
 1100 DATA 60,0,0,24,16,32,16,16,
 164
 1110 DATA 24,0,0,16,16,16,16,16,
 104
 1120 DATA 16,0,0,48,16,8,16,16,1
 20
 1130 DATA 48,0,0,20,40,0,0,0,108
 1140 DATA 0,0,0,24,36,60,52,60,2
 32
 1150 DATA 36,24,0,0,0,0,0,0,60
 1160 DATA 0,0,0,0,0,0,0,0,0
 1170 DATA 0,0,0,0,0,0,0,0,0
 1180 DATA 0,0,0,0,0,0,0,0,0
 1190 DATA 0,0,0,0,0,0,0,0,0
 1200 DATA 0,0,0,0,0,0,0,0,0
 1210 DATA 0,0,0,0,0,0,0,0,0
 1220 DATA 0,0,42,125,92,124,181,
 192,756
 1230 DATA 33,0,0,34,164,234,44,3
 4,543
 1240 DATA 125,92,58,169,234,230,
 128,246,1282
 1250 DATA 1,50,169,234,201,122,2
 54,24,1055
 1260 DATA 56,2,207,10,50,168,234
 ,205,932
 1270 DATA 232,237,201,17,174,238
 ,205,128,1432
 1280 DATA 10,42,14,92,87,125,167
 ,40,577
 1290 DATA 228,254,22,218,17,34,3
 2,23,828
 1300 DATA 58,168,234,188,56,15,5
 8,167,944
 1310 DATA 234,186,56,9,106,34,16
 4,234,1023
 1320 DATA 253,203,67,198,201,207
 ,4,108,1241

1330 DATA 98,229,205,226,237,225
 ,253,203,1676
 1340 DATA 1,198,237,91,167,234,2
 2,0,950
 1350 DATA 28,167,237,82,48,251,2
 5,69,907
 1360 DATA 58,167,234,60,79,58,16
 4,234,1054
 1370 DATA 185,56,2,62,0,184,200,
 197,886
 1380 DATA 62,32,217,215,217,193,
 24,237,1197
 1390 DATA 17,11,238,50,14,92,205
 ,128,755
 1400 DATA 10,201,17,11,238,50,15
 ,92,634
 1410 DATA 24,244,17,114,238,24,2
 36,205,1102
 1420 DATA 226,237,58,166,234,254
 ,129,6,1310
 1430 DATA 14,40,2,6,17,42,164,23
 4,519
 1440 DATA 175,128,189,56,252,111
 ,38,0,949
 1450 DATA 24,164,62,63,217,215,2
 17,201,1163
 1460 DATA 33,129,41,34,166,234,1
 95,232,1064
 1470 DATA 237,33,192,50,24,245,2
 45,205,1231
 1480 DATA 84,31,56,2,207,20,241,
 254,895
 1490 DATA 32,210,37,240,254,1,56
 ,168,998
 1500 DATA 40,222,254,2,40,227,25
 4,6,1045
 1510 DATA 56,208,40,179,254,8,56
 ,202,1003
 1520 DATA 254,12,218,32,239,40,1
 95,254,1244
 1530 DATA 13,40,25,254,16,56,187
 ,254,845
 1540 DATA 22,56,133,254,24,56,14
 7,254,946
 1550 DATA 30,56,175,33,169,234,2
 30,1,928
 1560 DATA 15,60,119,201,205,226,
 237,205,1268
 1570 DATA 177,239,58,167,234,60,
 50,164,1149
 1580 DATA 234,201,125,167,32,6,1
 24,167,1056
 1590 DATA 200,37,104,44,45,34,16
 4,234,862
 1600 DATA 201,125,184,56,6,121,3
 6,188,917
 1610 DATA 216,46,255,44,34,164,2
 34,201,1194
 1620 DATA 245,58,167,234,71,58,1
 68,234,1235
 1630 DATA 79,241,42,164,234,254,
 9,56,1079
 1640 DATA 209,40,222,254,11,56,6
 ,37,835
 1650 DATA 248,34,164,234,201,121
 ,36,188,1226
 1660 DATA 216,34,164,234,201,33,
 169,234,1285
 1670 DATA 203,126,192,53,192,58,
 168,234,1226
 1680 DATA 60,119,55,205,110,239,

253,203,1244
 1690 DATA 1,174,253,203,1,110,40
 ,250,1032
 1700 DATA 58,8,92,254,32,40,4,25
 4,742
 1710 DATA 226,32,2,207,12,167,62
 ,7,715
 1720 DATA 1,249,12,17,170,237,33
 ,224,943
 1730 DATA 80,245,229,197,205,156
 ,239,62,1413
 1740 DATA 8,245,56,1,235,126,18,
 10,699
 1750 DATA 119,56,1,235,36,19,3,2
 41,710
 1760 DATA 61,32,238,193,225,3,44
 ,241,1037
 1770 DATA 61,32,222,201,245,229,
 10,230,1230
 1780 DATA 127,237,75,54,92,111,3
 8,0,734
 1790 DATA 41,41,41,9,68,77,225,2
 41,743
 1800 DATA 201,42,164,234,58,167,
 234,189,1289
 1810 DATA 208,58,168,234,46,0,36
 ,188,938
 1820 DATA 48,9,103,229,205,69,23
 9,205,1107
 1830 DATA 254,13,225,34,164,234,
 201,42,1167
 1840 DATA 164,234,58,166,234,71,
 125,135,1187
 1850 DATA 135,133,203,112,32,1,1
 33,79,828
 1860 DATA 15,15,15,230,31,111,12
 4,68,609
 1870 DATA 15,15,15,230,224,181,1
 11,120,911
 1880 DATA 230,24,198,64,254,88,5
 6,2,916
 1890 DATA 62,64,103,121,230,7,60
 ,71,718
 1900 DATA 201,17,170,233,254,128
 ,48,9,1060
 1910 DATA 111,38,0,41,41,41,25,2
 35,532
 1920 DATA 201,254,144,56,8,214,1
 44,237,1258
 1930 DATA 91,123,92,24,235,71,20
 5,56,897
 1940 DATA 11,17,146,92,201,214,1
 65,210,1056
 1950 DATA 16,12,198,165,245,205,
 226,237,1304
 1960 DATA 205,177,239,241,205,1,
 240,205,1513
 1970 DATA 207,239,58,166,234,47,
 79,62,1092
 1980 DATA 8,245,213,26,86,44,94,
 197,913
 1990 DATA 5,40,10,203,2,203,10,2
 03,676
 2000 DATA 19,203,18,24,243,193,2
 53,203,1156
 2010 DATA 87,86,40,1,47,197,71,1
 22,651
 2020 DATA 253,203,87,70,32,1,175
 ,168,989
 2030 DATA 193,161,197,71,121,47,
 162,176,1128

2040 DATA 193,87,120,61,40,10,20
3,11,725
2050 DATA 203,3,203,26,203,27,24
,243,932

2060 DATA 115,45,114,209,241,19,
36,61,840
2070 DATA 32,183,37,229,197,205,
219,11,1113

2080 DATA 193,225,44,62,2,184,21
2,219,1141
2090 DATA 11,42,164,234,44,34,16
4,234,927
2100 DATA 201,0,0,0,0,0,0,201

MATRIX-POKER

W. Denissen

Matrix-poker is een variant op het pokerspel met 5 dobbelstenen. Als u een combinatie scoort met deze stenen, krijgt u als extra een worp met twee stenen op de matrix, die op zich ook weer extra bonus bevat.

Als het spel begint, beschikt u over 25 speelpunten (credits) en elk spel dat u speelt, kost u 1 credit.

Het spel is globaal verdeeld in 4 stukken. Helemaal onderaan het scherm, regel 23 en 24 worden gebruikt om u te laten zien welke mogelijkheden u heeft op een willekeurig moment en met welke toetsen u deze mogelijkheden kunt bereiken.

Daarboven ziet u de matrix met de getallen 1 tot en met 12 en links en rechts ernaast de 2 dobbelstenen voor deze worp.

Weer daarboven ziet u de pokersdobbelstenen en er omheen zijn elf vakken met de nummers 2 tot en met 12 en de betreffende bonus die u met de matrix-worp kunt verdienen. Niet elke bonus is zo leuk, 5 en 9 pakken alle winst die u op dat ogenblik heeft, weer af.

Als u het spel speelt, worden eerst de 5 pokerstenen gerold. Als u hiermee scoort, wordt de winnende combinatie zichtbaar bij M-POKER, bij WINST (rechts onder) zal uw winst worden bijgeteld.

Als u "NIETS" scoorde, is de beurt voorbij en wordt u de mogelijkheid geboden verder te spelen.

Heeft u wel een combinatie gescoord, dan gaan de twee matrix-stenen rollen. De waarden van beide dobbelstenen en hun som gaan nu op de matrix knippen en indien u een volledig knipperende rij scoort zal de betreffende matrix-waarde ervan worden uitbetaald. De waarde vindt u links (horizontaal) en rechts (vertikaal). Omdat matrix-bonus 7 de huidige waarde van de matrix verdubbelt, kan deze per beurt nogal uiteenlopen.

De som van beide dobbelstenen geeft recht op de matrix-bonus. Is dat 3 of 10 dan mag u opnieuw met de matrix-stenen rollen en krijgt u volledig de daarmee gewonnen bonus. Dit in tegenstelling tot bonus 6 of 8, waarbij ALLEEN het extra cijfer op de matrix gaat knippen.

Winstpunten kunt u pas incasseren als u er 100 of meer heeft, pas dan krijgt u steeds VOOR een nieuwe beurt de mogelijkheid om ze om te zetten in spelpunten. Per 100 winstpunten ontvangt u 1 spelpunt (credit).

Ik heb gemerkt dat de beste tactiek is, om niet meteen elke 100 winstpunten te incasseren, maar ze te bewaren. Dit omdat u als bonus al uw winstpunten zou kunnen verdubbelen, verdrievoudigen en zelfs verviervoudigen. Dat tikt natuurlijk lekkerder door bij bv. 300 punten dan bij 25 of 30!

```
10 BORDER 1: PAPER 1: INK 7: C
LS : RESTORE 1910: FOR x=0 TO 16
7: READ a: POKE USR "a"+x,a: NEX
T x: GO SUB 1540: GO TO 1330
20 PRINT BRIGHT 1: PAPER 0: IN
K 2:AT 21,0;"CREDIT:";k;AT 21,17
;"WINST:";w;" ": RETURN
30 LET p$=STR$ h: LET q$=STR$
40 IF LEN p$<4 THEN LET p$=" "
+p$: GO TO 40
50 IF LEN q$<4 THEN LET q$=" "
+q$: GO TO 50
60 PRINT PAPER 5: INK 0:AT 18,
0;"HOR=";AT 19,1;p$;AT 19,18;q$;
AT 18,17;"VER=": RETURN
70 LET u=INT (RND*6)+1: LET a=
13: LET b=1: IF u=VAL b$(1) THEN
GO TO 70
80 LET b$(1)=STR$ u: GO SUB 34
0
90 LET u=INT (RND*6)+1: LET a=
13: LET b=18: IF u=VAL b$(2) THE
N GO TO 90
100 LET b$(2)=STR$ u: GO TO 340
110 LET u=INT (RND*6)+1: LET a=
5: LET b=2: IF u=VAL a$(1) THEN
GO TO 110
120 LET a$(1)=STR$ u: GO TO 340
130 LET u=INT (RND*6)+1: LET b=
6: IF u=VAL a$(2) THEN GO TO 130
140 LET a$(2)=STR$ u: GO TO 340
150 LET u=INT (RND*6)+1: LET b=
10: IF u=VAL a$(3) THEN GO TO 15
0
160 LET a$(3)=STR$ u: GO TO 340
170 LET u=INT (RND*6)+1: LET b=
14: IF u=VAL a$(4) THEN GO TO 17
0
180 LET a$(4)=STR$ u: GO TO 340
190 LET u=INT (RND*6)+1: LET b=
18: IF u=VAL a$(5) THEN GO TO 19
0
200 LET a$(5)=STR$ u: GO TO 340
210 FOR x=18 TO 20: PRINT OVER
1: FLASH f: PAPER 2: INK 7:AT x,
22;" " " : NEXT x: RETURN
220 FOR x=15 TO 17: PRINT OVER
1: FLASH f: PAPER 3: INK 7:AT x,
22;" " " : NEXT x: RETURN
230 FOR x=12 TO 14: PRINT OVER
1: FLASH f: PAPER 4: INK 0:AT x,
22;" " " : NEXT x: RETURN
```



```

240 FOR x=9 TO 11: PRINT OVER 1
; FLASH f; PAPER 6; INK 0; AT x,2
2; "
NEXT x: RETURN
250 FOR x=9 TO 11: PRINT OVER 1
; FLASH f; PAPER 4; INK 0; AT x,1
2; "
NEXT x: RETURN
260 FOR x=9 TO 11: PRINT OVER 1
; FLASH f; PAPER 2; INK 7; AT x,2
; "
NEXT x: RETURN
270 FOR x=6 TO 8: PRINT FLASH f
; OVER 1; PAPER 2; INK 7; AT x,22
; "
NEXT x: RETURN
280 FOR x=3 TO 5: PRINT PAPER 5
; INK 0; FLASH f; OVER 1; AT x,22
; "
NEXT x: RETURN
290 FOR x=0 TO 2: PRINT PAPER 6
; INK 0; FLASH f; OVER 1; AT x,22
; "
NEXT x: RETURN
300 FOR x=0 TO 2: PRINT FLASH f
; OVER 1; PAPER 3; INK 7; AT x,12
; "
NEXT x: RETURN
310 FOR x=0 TO 2: PRINT FLASH f
; OVER 1; PAPER 5; INK 0; AT x,2;
; "
NEXT x: RETURN
320 PRINT #1; PAPER 1; AT 1,0; "
; RETURN
330 PRINT #1; PAPER 1; AT 0,0; "
; RETURN
340 LET q$=STR$ u: PRINT AT a,b
; "
; AT a+1,b; "
; AT a+2,b; "
; IF q$="1" OR q$="3" OR q$="
5" THEN PRINT AT a+1,b; "
; CHR$
159; "
350 IF q$="2" OR q$="3" OR q$="
5" THEN PRINT AT a,b; CHR$ 159; "
; AT a+2,b; "
; CHR$ 159
360 IF q$="5" THEN PRINT AT a,b
+2; CHR$ 159; AT a+2,b; CHR$ 159
370 IF q$="4" OR q$="6" THEN PR
INT AT a,b; CHR$ 159; "
; CHR$ 159
; AT a+2,b; CHR$ 159; "
; CHR$ 159
380 IF q$="6" THEN PRINT AT a+1
,b; CHR$ 159; "
; CHR$ 159
390 BEEP 0.01,INT (RND*40): RET
URN
400 GO SUB 20: LET e=0.1: PRINT
#1; PAPER 1; INK 6; FLASH 1; AT
1,0; "
SPELEN="; INVERSE
1;"ENTER"; INVERSE 0; "
"
410 BEEP e,INT -30+(RND*70): LE
T q=CODE INKEY$: IF q<>13 AND q<
>73 THEN LET e=e-0.005
420 IF q=73 THEN GO SUB 1540: G
O TO 1340
430 IF q=13 THEN GO SUB 320: GO
TO 460
440 IF e<0.0025 THEN LET e=0.1
450 GO TO 410
460 LET k=k-1: IF k<0 THEN PRIN
T #1; AT 1,0;"GELD IS OP. WILT U
OPNIEUW="; INVERSE 1;"TOETS": PA
USE 0: GO TO 1330
470 GO SUB 20: PRINT PAPER 6; I
NK 0; AT 3,9; "
480 PAPER 6: LET a$="00000": LE
T g=INT (RND*6)+6: FOR x=1 TO g:
GO SUB 110: NEXT x

```

```

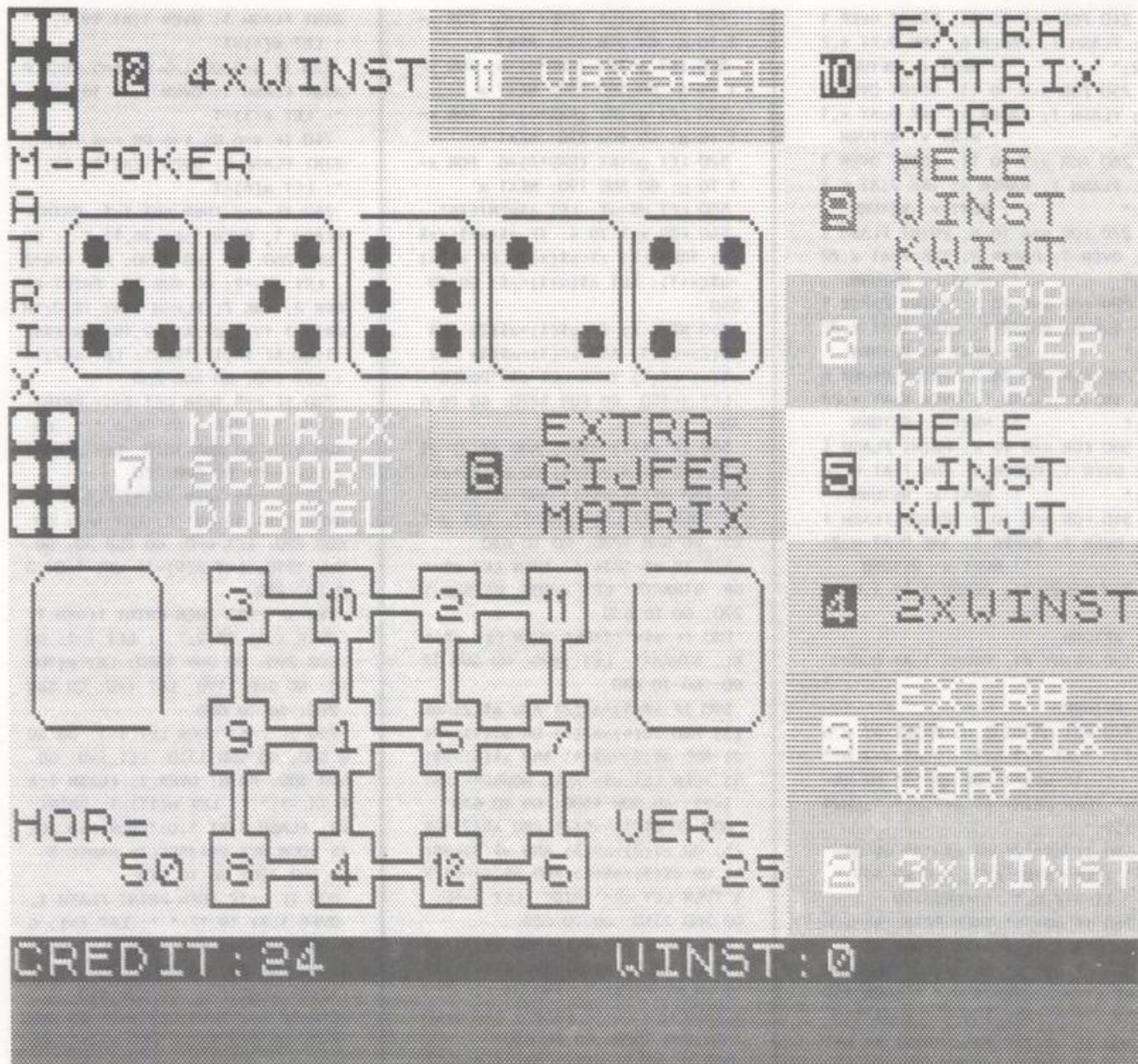
490 LET g=INT (RND*6)+6: FOR x=
1 TO g: GO SUB 130: NEXT x
500 LET g=INT (RND*6)+6: FOR x=
1 TO g: GO SUB 150: NEXT x
510 LET g=INT (RND*6)+6: FOR x=
1 TO g: GO SUB 170: NEXT x
520 LET g=INT (RND*6)+6: FOR x=
1 TO g: GO SUB 190: NEXT x
530 LET d$=a$: LET c$="NIETS"
540 FOR x=1 TO 4: IF a$(x+1)<a$
(x) THEN LET z$=a$(x): LET a$(x)
=a$(x+1): LET a$(x+1)=z$: GO TO
540
550 NEXT x: IF a$(1)=a$(2) AND
a$(2)=a$(3) AND a$(3)=a$(4) AND
a$(4)=a$(5) THEN LET c$="POKER":
LET j=150: GO SUB 1270: GO TO 6
30
560 IF a$(1)=a$(2) AND a$(2)=a$
(3) AND a$(3)=a$(4) OR a$(2)=a$(
3) AND a$(3)=a$(4) AND a$(4)=a$(
5) THEN LET c$="CARREE": LET j=1
25: GO SUB 1280: GO TO 630
570 IF a$="23456" THEN LET c$="
GR. STRAAT": LET j=100: GO SUB 1
290: GO TO 630
580 IF a$="12345" THEN LET c$="
KL. STRAAT": LET j=90: GO SUB 12
90: GO TO 630
590 IF a$(1)=a$(2) AND a$(2)=a$
(3) AND a$(4)=a$(5) OR a$(1)=a$(
2) AND a$(3)=a$(4) AND a$(4)=a$(
5) THEN LET c$="FULL HOUSE": LET
j=75: GO SUB 1300: GO TO 630
600 IF a$(1)=a$(2) AND a$(2)=a$
(3) OR a$(2)=a$(3) AND a$(3)=a$(
4) OR a$(3)=a$(4) AND a$(4)=a$(5
) THEN LET c$="TRIO": LET j=50:
GO SUB 1310: GO TO 630
610 IF a$(1)=a$(2) AND a$(3)=a$
(4) OR a$(1)=a$(2) AND a$(4)=a$(
5) OR a$(2)=a$(3) AND a$(4)=a$(5
) THEN LET c$="2 PAAR": LET j=40
: GO SUB 1320: GO TO 630
620 IF a$(1)=a$(2) OR a$(2)=a$(
3) OR a$(3)=a$(4) OR a$(4)=a$(5)
THEN LET c$="1 PAAR": LET j=20
630 IF c$="NIETS" THEN PRINT PA
PER 6; INK 0; AT 3,9; "
; c$: GO
SUB 1230: GO TO 400
640 LET w=w+j: PRINT PAPER 6; I
NK 0; AT 3,9; "
; c$
650 PRINT BRIGHT 1; PAPER 0; IN
K 2; AT 21,7;k; AT 21,23;w
660 PAPER 5: LET b$="00": LET g
=INT (RND*10)+10: FOR x=1 TO g:
GO SUB 70: NEXT x
670 LET o=VAL b$: IF o=l THEN G
O SUB 90
680 LET s=VAL b$(1): LET t=VAL
b$(2): LET r=s+t
690 IF s=1 OR t=1 THEN PRINT FL
ASH 1; OVER 1; AT 16,9; "
; LET m
(1)=1
700 IF s=2 OR t=2 OR r=2 THEN P
RINT FLASH 1; OVER 1; AT 13,12; "
; LET m(2)=1
710 IF s=3 OR t=3 OR r=3 THEN P
RINT FLASH 1; OVER 1; AT 13,6; "
; LET m(3)=1
720 IF s=4 OR t=4 OR r=4 THEN P

```

```

RINT FLASH 1; OVER 1; AT 19,9; "
; LET m(4)=1
730 IF s=5 OR t=5 OR r=5 THEN P
RINT FLASH 1; OVER 1; AT 16,12; "
; LET m(5)=1
740 IF s=6 OR t=6 OR r=6 THEN P
RINT FLASH 1; OVER 1; AT 19,15; "
; LET m(6)=1
750 IF r=7 THEN LET f=1: PRINT
FLASH 1; OVER 1; AT 16,15; "
; GO
SUB 260: GO SUB 1290: LET h=h+2
: LET v=v+2: GO SUB 30: PRINT PA
PER 2; INK 7; FLASH 1; AT 18,0;"H
OR"; AT 19,1;p$; AT 19,18; INVERSE
1;q$; AT 18,17;"VER": LET m(7)=1
: LET f=0: GO SUB 260
760 IF r=8 THEN LET f=1: PRINT
FLASH 1; OVER 1; AT 19,6; "
; GO
SUB 270: GO SUB 1070: LET m(8)=1
: LET f=0: GO SUB 270
770 IF r=9 THEN PRINT FLASH 1;
OVER 1; AT 16,6; "
; LET f=1: GO
SUB 280: LET w=0: GO SUB 20: GO
SUB 1210: LET m(9)=1: LET f=0: G
O SUB 280
780 IF r=10 THEN PRINT FLASH 1;
OVER 1; AT 13,9; "
; LET f=1: GO
SUB 290: GO SUB 1320: LET m(10)
=1: GO SUB 1190: LET f=0: GO SUB
290: GO TO 680
790 IF r=11 THEN LET f=1: GO SU
B 300: GO SUB 1310: LET f=0: GO
SUB 300: PRINT OVER 1; FLASH 1; A
T 13,15; "
; LET m(11)=1: PRINT
#1; FLASH 1; AT 1,0;"DRUK EEN TOE
TS VOOR HET VRIJSPEL": PAUSE 0:
GO SUB 320: GO TO 470
800 IF r=12 THEN PRINT FLASH 1;
OVER 1; AT 19,12; "
; LET f=1: G
O SUB 310: GO SUB 1270: LET w=w+
4: GO SUB 20: GO SUB 1270: LET m
(12)=1: LET f=0: GO SUB 310
810 IF r=2 THEN LET f=1: GO SUB
210: GO SUB 1280: LET w=w+3: GO
SUB 20: LET f=0: GO SUB 210
820 IF r=3 THEN LET f=1: GO SUB
220: GO SUB 1320: GO SUB 1190:
LET f=0: GO SUB 220: GO TO 680
830 IF r=4 THEN LET f=1: PRINT
FLASH 1; OVER 1; AT 16,15; "
; GO
SUB 230: GO SUB 1300: LET w=w+2
: GO SUB 20: LET m(4)=1: LET f=0
: GO SUB 230
840 IF r=5 THEN LET f=1: GO SUB
240: GO SUB 1210: LET w=0: GO S
UB 20: GO SUB 1210: LET f=0: GO
SUB 240
850 IF r=6 THEN LET f=1: GO SUB
250: GO SUB 1070: LET f=0: GO S
UB 250
860 LET n=0: IF m(2)=1 AND m(10
)=1 AND m(2)=1 AND m(11)=1 THEN
LET n=n+h
870 IF m(9)=1 AND m(1)=1 AND m(
5)=1 AND m(7)=1 THEN LET n=n+h
880 IF m(8)=1 AND m(4)=1 AND m(
12)=1 AND m(6)=1 THEN LET n=n+h
890 IF m(3)=1 AND m(9)=1 AND m(
8)=1 THEN LET n=n+v
900 IF m(10)=1 AND m(1)=1 AND m
(4)=1 THEN LET n=n+v

```

```

910 IF m(2)=1 AND m(5)=1 AND m(
12)=1 THEN LET n=n+v
920 IF m(11)=1 AND m(7)=1 AND m
(6)=1 THEN LET n=n+v
930 IF n=0 THEN GO TO 990
940 LET h=50: LET v=25
950 PRINT PAPER 0: INK 2; BRIGH
T 1; AT 21,17; "WINST: "; w: PRINT #
1; PAPER 1; INK 6; FLASH 1; AT 1,
0; " NEEM MATRIX WINST="; INVER
SE 1; "SPACE"; INVERSE 0; "
: PAUSE 0: IF CODE INKEY$ <> 32 TH
EN GO TO 950
960 DIM m(12): GO SUB 320: FOR
x=1 TO n: LET w=w+1: LET n=n-1:
BEEP 0.02, -30: PRINT BRIGHT 1; P
APER 0: INK 2; AT 21,23; w: NEXT x
970 GO SUB 30
980 FOR x=12 TO 20: PRINT PAPER
5; INK 0; OVER 1; AT x,0; "
: NEXT x
990 GO SUB 20: IF w<100 THEN GO
TO 400

```

```

1000 PRINT #1; PAPER 1; INK 6; F
LASH 1; AT 0,0; "NEEM WINST="; INV
ERSE 1; "SPACE"; INVERSE 0; FLASH
0; " 100 WINST = "; FLASH 1;
INVERSE 1; "DOORSPLEN"; INVERSE
0; "ENTER"; FLASH 0; " 1 CRED
IT "
1010 LET q=CODE INKEY$: IF q<32
AND q<13 THEN GO TO 1010
1020 IF q=73 OR q=104 THEN GO SU
B 1540: GO TO 1340
1030 GO SUB 330: GO SUB 320: IF
q=13 THEN GO TO 470
1040 FOR x=1 TO w: LET k=k+1: LE
T w=w-100: PRINT BRIGHT 1; PAPER
0; INK 2; AT 21,7; k; AT 21,23; w; "
: GO SUB 1260: IF w>=100 THEN
NEXT x
1050 LET h=50: LET v=25: DIM m(1
2): GO SUB 320: GO SUB 30
1060 FOR x=12 TO 20: PRINT PAPER
5; INK 0; OVER 1; AT x,0; "
: NEXT x: GO TO
400

```

```

1070 PRINT #1; PAPER 1; INK 6; F
LASH 1; AT 1,0; " STOP=";
INVERSE 1; "SPACE"; INVERSE 0; "
"
1080 BEEP 0.01, -30+INT (RND*70):
IF CODE INKEY$ <> 32 THEN GO TO 1
080
1090 BEEP 0.01, -30+INT (RND*70):
LET i=INT (RND*12)+1: IF m(i)=1
THEN GO TO 1090
1100 GO SUB 320: LET m(i)=1: GO
TO 1110
1110 IF i=3 OR i=9 OR i=8 THEN L
ET b=6
1120 IF i=10 OR i=1 OR i=4 THEN
LET b=9
1130 IF i=2 OR i=5 OR i=12 THEN
LET b=12
1140 IF i=11 OR i=7 OR i=6 THEN
LET b=15
1150 IF i=3 OR i=10 OR i=2 OR i=
11 THEN LET a=13
1160 IF i=9 OR i=1 OR i=5 OR i=7
THEN LET a=16

```



```

1170 IF i=8 OR i=4 OR i=12 OR i=
6 THEN LET a=19
1180 PRINT PAPER 0; AT 21,17; "
    ": PRINT FLASH 1; OVER 1
    ; AT a,b; " ": RETURN
1190 PAPER 5: PRINT #1; FLASH 1;
AT 1,0; " DRUK EEN TOETS VOOR EXT
RA WORP ": PAUSE 0: GO SUB 320:
LET b$="00": LET g=INT (RND*10)+
10: FOR x=1 TO g: GO SUB 70: NEX
T x: RETURN
1200 FOR x=1 TO 25: BEEP 0.005,x
*2: NEXT x: RETURN
1210 FOR x=40 TO -30 STEP -1: BE
EP 0.005,x: NEXT x: RETURN
1220 FOR x=1 TO 25: BEEP 0.005,x
*2: NEXT x: RETURN
1230 FOR x=40 TO -30 STEP -1: BE
EP 0.005,x: NEXT x: RETURN
1240 FOR x=25 TO 1 STEP -1: BEEP
0.01,x: NEXT x: FOR x=1 TO 25:
BEEP 0.01,x: NEXT x: RETURN
1250 BEEP 0.2,0: BEEP 0.4,20: FO
R x=1 TO 3: BEEP 0.2,10: BEEP 0.
2,22: PAUSE 2: NEXT x: BEEP 0.4,
27: RETURN
1260 FOR y=12 TO 25: BEEP 0.003,
y+20: NEXT y: RETURN
1270 RESTORE 1680: FOR x=1 TO 14
: READ a,b: BEEP a/2,b: NEXT x:
RETURN
1280 RESTORE 1690: FOR x=1 TO 36
: READ a,b: BEEP a/2,b: NEXT x:
RETURN
1290 RESTORE 1720: FOR x=1 TO 29
: READ b,a: BEEP a/30,b: NEXT x:
RETURN
1300 RESTORE 1730: FOR x=1 TO 8:
READ a: BEEP .05,a: BEEP .05,a-
12: BEEP .05,a: BEEP .05,a-12: N
EXT x: BEEP .05,5: BEEP .05,-7:
BEEP .1,-19: RETURN
1310 RESTORE 1740: FOR x=1 TO 13
: READ a,b: BEEP a/1.5,b: NEXT x
: FOR y=1 TO 15: NEXT y: BEEP .5
,19: RETURN
1320 RESTORE 1750: FOR x=1 TO 9:
READ a,b: BEEP a/1.5,b: NEXT x:
RETURN
1330 LET c$="": LET d$="00000":
LET b$="00": DIM m(12): LET l=10
: LET k=25: LET w=0: LET h=50: L
ET v=25
1340 CLS : PAPER 6: INK 0: RESTO
RE 1850: FOR x=3 TO 8: FOR y=0 T
O 21: READ a: PRINT AT x,y; CHR$
a: NEXT y: NEXT x
1350 PAPER 5: INK 0: RESTORE 176
0: FOR x=12 TO 20: FOR y=0 TO 21
: READ a: PRINT AT x,y; CHR$ a: N
EXT y: NEXT x
1360 GO SUB 30: PRINT PAPER 6; I
NK 0; AT 3,9; " ": c$
1370 FOR x=0 TO 2: PRINT PAPER 0
; INK 6; AT x,0; CHR$ 159; CHR$ 159
; AT x+9,0; CHR$ 159; CHR$ 159: NEX
T x
1380 LET f=0: PRINT BRIGHT 1; PA
PER 0; INK 2; AT 21,0; "
    ": GO SUB

```

20

```

1390 GO SUB 210: PRINT PAPER 2;
INK 7; AT 19,23; INVERSE 1; "2";
INVERSE 0; " 3xWINST"
1400 GO SUB 220: PRINT PAPER 3;
INK 7; AT 15,25; "EXTRA"; AT 16,23;
INVERSE 1; "3"; INVERSE 0; " MATR
IX"; AT 17,25; "WOP"
1410 GO SUB 230: PRINT PAPER 4;
INK 0; AT 13,23; INVERSE 1; "4"; I
NVERSE 0; " 2xWINST"
1420 GO SUB 240: PRINT PAPER 6;
INK 0; AT 9,25; "HELE"; AT 10,23; I
NVERSE 1; "5"; INVERSE 0; " WINST"
; AT 11,25; "KWIJT"
1430 GO SUB 250: PRINT PAPER 4;
INK 0; AT 9,15; "EXTRA"; AT 10,13;
INVERSE 1; "6"; INVERSE 0; " CIJFE
R"; AT 11,15; "MATRIX"
1440 GO SUB 260: PRINT PAPER 2;
INK 7; AT 9,5; "MATRIX"; AT 10,3; I
NVERSE 1; "7"; INVERSE 0; " SCOORT
"; AT 11,5; "DUBBEL"
1450 GO SUB 270: PRINT PAPER 2;
INK 7; AT 6,25; "EXTRA"; AT 7,23; I
NVERSE 1; "8"; INVERSE 0; " CIJFER
"; AT 8,25; "MATRIX"
1460 GO SUB 280: PRINT PAPER 5;
INK 1; AT 3,25; "HELE"; AT 4,23; IN
VERSE 1; "9"; INVERSE 0; " WINST";
AT 5,25; "KWIJT"
1470 GO SUB 290: PRINT PAPER 6;
INK 0; AT 0,25; "EXTRA"; AT 1,23; I
NVERSE 1; CHR$ 144; INVERSE 0; " M
ATRIX"; AT 2,25; "WOP"
1480 GO SUB 300: PRINT PAPER 3;
INK 7; AT 1,13; INVERSE 1; CHR$ 14
5; INVERSE 0; " VRYSPEL"
1490 GO SUB 310: PRINT PAPER 5;
INK 0; AT 1,3; INVERSE 1; CHR$ 146
; INVERSE 0; " 4xWINST"
1500 PAPER 6: INK 0: LET a=5: LE
T b=2: FOR x=1 TO 5: LET u=VAL d
$(x): GO SUB 340: LET b=b+4: NEX
T x
1510 PAPER 5: LET a=13: LET b=1:
FOR x=1 TO 2: LET u=VAL b$(x):
GO SUB 340: LET b=b+17: NEXT x
1520 PAPER 5: FOR x=1 TO 12: IF
m(x)=1 THEN LET i=x: GO SUB 1110
1530 NEXT x: GO TO 400
1540 CLS : PRINT "SPELREGELS."
    "Matrix-poker is een pokerspel
    met dobbelstenen."
    "De spelreg
    els zijn zeer eenvoudi-
    g en eraan het scherm wordt steeds we-
    r aangegeven wat op ieder will-
    ekeurig moment de mo-
    gelijkheden zijn en welke toets U in moet
    drukken om er gebruik van te mak-
    en."
1550 PRINT "Winstpunten kunt U
    pas opnemen indien U er meer da-
    n 100 heeft. Per 100 winstpunten
    ontvangt U 1 spelpunt (Credit)"
    "
1560 GO SUB 1640: PRINT "Indien
    U met de vijf pokerschijven tre-
    ffer scoort krijgt U een worp vo-
    or de MATRIX erbij. De punt
    en van deze twee dobbel-
    stenen en het totaal ervan wor-
    den dan

```

```

    op de matrix genoteerd en indi-
    en U horizontaal of ver-
    tikaal een volle rij scoort op de matr-
    ix wordt deze uitbetaald."
1570 PRINT "Tevens geeft het to-
    taal van de matrix-worp u een e-
    xtra bonus. Deze bonus kan het
    vermenigvuldigen van Uw winst
    met 2 zijn, een vrijspel of zel-
    fs 4 keer Uw winst, maar er is o-
    ok de kans dat U al uw winst v-
    erliest!"
    "U kunt zich niet ont-
    rekken aan dit bonusspel!"
1580 GO SUB 1640: PRINT "SCORE B
    IJ HET POKERSPEL:"
    "
    SCORE" INVERSE 1;
    "1 PAAR 20
    "; INVERSE 0; "(b.v. 2 keer de 4
    )"
    INVERSE 1; "2 PAAR
    40 "; INVERSE 0; "(b.
    v. 2 keer de 1 en 2 keer 6)"
1590 PRINT INVERSE 1; "TRIO
    40 "; INVERSE
    0; "(b.v. 3 keer de 5)"
    INVERSE 1; "FULL HOUSE
    75 "; INVERSE 0; "(trio en 1 paa-
    r)"
1600 PRINT INVERSE 1; "KLEINE STR-
    AAT 90 "; INVERSE
    0; "(1,2,3,4 en 5)"
    INVERSE 1; "
    GROTE STRAAT 100
    "; INVERSE 0; "(2,3,4,5 en 6)"
1610 PRINT INVERSE 1; "CARREE
    125 "; INVERSE
    0; "(b.v. 4 keer de 2)"
    INVERSE 1; "POKER
    1
    50 "; INVERSE 0; "(alle stenen h-
    etzelfde)"
1620 GO SUB 1640: PRINT "Indien
    U tijdens het spelen de spelreg-
    els nog eens door wilt le-
    zen kan dat door op 'I' te druk-
    ken aan het begin van een nieuwe-
    beurt (als onder aan het scherm
    SPELEN="
    "; INVERSE 1; "ENTER"; INVERSE 0;
    " staat."
1630 GO SUB 1640: GO SUB 1640: R
    ETURN
1640 PRINT #1; PAPER 2; INK 7; F
    LASH 1; AT 1,0; "C = COPY
    DOORGAAN = D": PAUSE 0
1650 LET q=CODE INKEY$: IF q<67
    AND q<99 AND q<68 AND q<100
    THEN GO TO 1640
1660 IF q=67 OR q=99 THEN COPY
1670 CLS : RETURN
1680 DATA .5,4,.25,5,.25,7,.36,7
    ,.12,5,.25,4,.25,2,.25,0,.25,0,.
    25,2,.25,4,.36,2,.12,0,.5,0
1690 DATA .25,0,.25,2,.125,3,.12
    5,2,.25,0,.25,0,.25,2,.125,3,.12
    5,2,.25,0
1700 DATA .25,3,.25,5,.5,7,.25,3
    ,.25,5,.5,7,.1875,7,.0625,8,.125
    ,7,.125,5,.125,3,.125,2,.25,0
1710 DATA .1875,7,.0625,8,.125,7
    ,.125,5,.125,3,.125,2,.25,0,.25,
    0,.25,-5,.5,0,.5,0,.5,-5,1,0
1720 DATA 0,3,2,3,4,3,5,5,0,6,0,
    0,0,0,5,3,4,3,5,3,7,5,2,6,0,0,0,
    0,5,3,4,3,5,3,9,3,5,7,1,5,7,4,5,
    3,0,0,0,0,5,3,4,3,2,4,4,4,5,5

```


1730 DATA 0,12,2,11,4,9,5,7
 1740 DATA .2,11,.2,7,.2,11,.2,7,
 .22,11,.22,11,.2,11,.3,2,.2,2,.2
 .2,.4,4,.3,6,.6,7
 1750 DATA .4,10,.1,10,.1,10,.4,1
 7,.1,10,.1,17,.1,20,.5,19,.6,17
 1760 DATA 160,155,155,155,161,14
 7,155,149,147,155,149,147,155,14
 9,147,155,149,32,155,155,155,161
 1770 DATA 158,32,32,32,157,158,5
 1,154,153,144,154,153,50,154,153
 ,145,157,158,32,32,32,157
 1780 DATA 158,32,32,32,157,150,1
 52,148,150,152,148,150,152,148,1
 50,152,148,158,32,32,32,157
 1790 DATA 158,32,32,32,157,147,1
 51,149,147,151,149,147,151,149,1
 47,151,149,158,32,32,32,157
 1800 DATA 163,156,156,156,32,158
 ,57,154,153,49,154,153,53,154,15
 3,55,157,163,156,156,156,162
 1810 DATA 32,32,32,32,32,150,152
 ,148,150,152,148,150,152,148,150
 ,152,148,32,32,32,32,32
 1820 DATA 32,32,32,32,147,151

,149,147,151,149,147,151,149,147
 ,151,149,32,32,32,32,32
 1830 DATA 32,32,32,32,32,158,56,
 154,153,52,154,153,146,154,153,5
 4,157,32,32,32,32,32
 1840 DATA 32,32,32,32,32,150,156
 ,148,150,156,148,150,156,148,150
 ,156,148,32,32,32,32,32
 1850 DATA 77,45,80,79,75,69,82,3
 2,32,32,32,32,32,32,32,32,32,
 32,32,32,32
 1860 DATA 65,160,155,155,155,161
 ,155,155,155,161,155,155,155,161
 ,155,155,155,161,155,155,155,161
 1870 DATA 84,158,32,32,32,164,32
 ,32,32,164,32,32,32,164,32,32,32
 ,164,32,32,32,157
 1880 DATA 82,158,32,32,32,164,32
 ,32,32,164,32,32,32,164,32,32,32
 ,164,32,32,32,157
 1890 DATA 73,158,32,32,32,164,32
 ,32,32,164,32,32,32,164,32,32,32
 ,164,32,32,32,157
 1900 DATA 88,163,156,156,156,163
 ,156,156,156,163,156,156,156,163

,156,156,156,163,156,156,156,162
 1910 DATA 0,76,210,82,82,82,76,0
 1920 DATA 0,36,108,36,36,36,36,0
 1930 DATA 0,76,210,68,72,80,94,0
 1940 DATA 0,0,0,0,0,7,4,4
 1950 DATA 32,32,224,0,0,0,0,0
 1960 DATA 0,0,0,0,0,224,32,32
 1970 DATA 4,4,7,0,0,0,0,0
 1980 DATA 36,36,36,36,36,231,0,0
 1990 DATA 0,0,231,36,36,36,36,36
 2000 DATA 4,4,252,0,0,252,4,4
 2010 DATA 32,32,63,0,0,63,32,32
 2020 DATA 0,0,0,0,0,255,0,0
 2030 DATA 0,0,255,0,0,0,0,0
 2040 DATA 32,32,32,32,32,32,32,3
 2
 2050 DATA 4,4,4,4,4,4,4,4
 2060 DATA 0,60,126,126,126,126,6
 0,0
 2070 DATA 0,0,0,0,0,0,1,2
 2080 DATA 0,0,0,0,0,0,128,64
 2090 DATA 64,128,0,0,0,0,0,0
 2100 DATA 2,1,0,0,0,0,0,0
 2110 DATA 36,36,36,36,36,36,36,3
 6

Karakters

H. & R. Kats

Dit programma is een hulp bij het zelf ontwerpen van een nieuwe tekenset en/of UDG's voor de Spectrum. Het werkt simpel en duidelijk.

Een paar stukjes machinetaal zorgen voor een aardige presentatie. Het oog wil tenslotte ook wat!

Vergeet niet om op de regels waarop staat: "REM grafische letters", de letters tussen de aanhalingstekens in grafische mode te typen. Dus eerst caps shift en 9, en dan de letter.

Met de toetsen 5 tot 8 (de pijltjes, **zonder caps shift**) wordt het teken "gemaakt"; met dezelfde toetsen + symbol shift kunt u vakjes wissen of door het raster heen "lopen". Verder wijst het programma zichzelf uit. Het start gewoon met RUN.

Als u de zelf ontworpen tekenset in een programma wilt gebruiken, moet de systeem-variabele CHARS (23606/07) van het adres van de nieuwe tekenset voorzien worden. Let er op dat het eigenlijke programma niet hoger in het geheugen komt dan adres 64600, anders kan de tekenset er niet meer bij. Maar dat zal bij de meeste zelf geschreven programma's nauwelijks voorkomen. In het programma moet alleen 1 regel opgenomen worden die CHARS POKet en de tekenset in het geheugen laadt:

```
1 CLEAR 64599: POKE 23606, 88
: POKE 23607,251: LOAD "naam" CO
```

en wanneer het om UDG's gaat:

```
1 LOAD "naam" CODE
```

of je kunt een klein laadprogrammaatje vóór alles zetten.

```
10 CLEAR 34999
20 LET machinetaal=1340
30 LET menu =740
40 LET vergroot=660
50 LET schermopbouw=540
70 GO SUB machinetaal
80 GO SUB schermopbouw
90 GO SUB menu
100
110 REM *** ONTWERP ***
120
130 RANDOMIZE USR 35012: PRINT
  AT 18,7;" ";AT 18,23;" "
140 IF poort=0 THEN PRINT AT
  1,0: RANDOMIZE USR 35068:
  GO TO 160
150 PRINT AT 3,5;"A B C D E F G
H I J K ";PRINT AT 6,6;"L M N
O P Q R S T U ";RANDOMIZE USR
35132: REM letters in graphic-
mode intypen!!!!
160 INPUT BRIGHT 1;"INPUT> ";
LINE BS
170 IF BS=CHR$ 226 THEN
RANDOMIZE USR 35109:
RANDOMIZE USR 35012: PRINT
AT 18,7;" ";AT 18,23;" "
GO TO 90
180 IF BS="" OR CODE BS<min OR
CODE BS>max THEN GO TO 160
190 IF poort=0 THEN POKE 35046
,CODE BS
```



```

200 IF poort=1 THEN LET a=(CODE
    B$+63): POKE 35046,a: LET
    B$=CHR$ (a+16)
210 LET A$=B$ ( TO 1)
220 GO SUB 660
230 LET reg=9: LET kol=27
240 DEF FN a()=(22528+(32*reg)+
kol)
250 LET s=FN a(): POKE s,213
260 LET C$=INKEY$
270 IF C$>="5" AND C$<="8" THEN
GO TO 370
280 IF C$>="Z" AND C$<="(" THEN
GO TO 330
290 IF C$=CHR$ 226 THEN
RANDOMIZE USR 35012: PRINT
AT 18,7;" ";AT 18,23;" ":
GO SUB 1290: GO TO 90
300 IF C$=CHR$ 7 THEN GO TO 110
310 IF C$=CHR$ 13 THEN
RANDOMIZE USR 35040:
GO TO 110
320 GO TO 260
330 REM * HOKJE ONGEKEURD *
340 LET d=CODE C$-32
350 LET paper=120
360 GO TO 400
370 REM * HOKJE ZWART *
380 LET d=VAL C$
390 LET paper=8
400 LET X=164+kol: LET Y=40-reg
410 IF paper=8 THEN PLOT X,Y
420 IF paper=120 AND POINT (X,Y
)=1 THEN PLOT OVER 1;X,Y
430 IF d=5 THEN LET kol=kol-1
440 IF d=8 THEN LET kol=kol+1
450 IF d=6 THEN LET reg=reg+1
460 IF d=7 THEN LET reg=reg-1
470 IF kol<20 THEN LET kol=20
480 IF kol>27 THEN LET kol=27
490 IF reg<9 THEN LET reg=9
500 IF reg>16 THEN LET reg=16
510 POKE FN a(),213
520 POKE s,paper
530 GO TO 250
540 REM * SCHERMOPBOUW *
550 FOR X=1 TO 72 STEP 8
560 PLOT 31,X+39: DRAW 64,0
570 PLOT 160,X+39: DRAW 64,0
580 PLOT X+30,40: DRAW 0,64
590 PLOT X+159,40: DRAW 0,64
600 NEXT X
610 PLOT 53,34: DRAW 13,0: DRAW
0,-13: DRAW -13,0: DRAW 0,13: P
LOT 181,34: DRAW 13,0: DRAW 0,-1
3: DRAW -13,0: DRAW 0,13
620 PLOT 70,7: DRAW 106,0: DRAW
0,25: DRAW -106,0: DRAW 0,-25
630 PRINT BRIGHT 1;AT 18,9;"STO
P =MENU " ;AT 19,9;"EDI
T =CLEAR " ;AT 20,9;"ENT
ER =DUMP "
640 RANDOMIZE USR 35012
650 RETURN
660 REM * VERGROOT ROUTINE *
670 PRINT AT 18,7;A$
680 FOR A=56 TO 63
690 FOR B=24 TO 31
700 IF POINT (A,B)=1 THEN PRINT
PAPER 5; OVER 1;AT 40-B,A-52;"

```

```

710 NEXT B
720 NEXT A
730 RETURN
740 REM *** MENU ***
750 PRINT AT 0,0:: RANDOMIZE US
R 35098
760 PRINT BRIGHT 1; PAPER 3; IN
K 7;AT 0,0;" SIMPLE SOFTW
ARE
770 PRINT BRIGHT 1;
    AT 10,13;"-MENU-"
    ;AT 11,13;"1.Save"
    ;AT 12,13;"2.Load"
    ;AT 13,13;"3.Kar."
    ;AT 14,13;"4.UDG "
780 PLOT 102,55: DRAW 50,0: DRA
W 0,41: DRAW -50,0: DRAW 0,-40
790 PRINT FLASH 1;AT 16,13;" KE
US ": PAUSE 0: LET K$=INKEY$
800 IF K$="" OR K$<"1" OR K$>"4
" THEN PRINT AT 16,18;"*": GO TO
790
810 GO SUB 1290
820 IF K$="1" THEN GO TO 1030
830 IF K$="2" THEN GO TO 1210
840 IF K$="4" THEN GO TO 920
850
860 REM *** karakterset ***
870
880 PRINT BRIGHT 1;AT 0,0;"
    DE ZX-KARAKTERSSET
890 LET poort=0
900 LET min=32: LET max=127
910 GO TO 1330
920
930 REM *** UDG's ***
940
950 PRINT AT 0,0:: RANDOMIZE US
R 35098
960 PRINT BRIGHT 1;AT 0,4;CHR$
138;"USER DEFINED GRAPHICS";CHR$
133
970 LET poort=1
980 LET min=65: LET max=85
990 PLOT 24,112: DRAW 207,0: DR
AW 0,55: DRAW -207,0: DRAW 0,-55
1000 PRINT AT 2,5;"A B C D E F G
H I J K ": REM grafische tekens
1010 PRINT AT 5,6;"L M N O P Q R
S T U ": REM grafische tekens
1020 GO TO 1330
1030 PRINT BRIGHT 1;AT 0,0;"
    SAVEN
1040 PRINT AT 3,3;"1-CASSETTE "
;AT 4,3;"2-DRIVE"
1050 PAUSE 0
1060 LET Z$=INKEY$: IF Z$="" OR
Z$<"1" OR Z$>"2" THEN GO TO 106
1070 PRINT BRIGHT 1; OVER 1;AT (
VAL Z$)+2,3;"
1080 PAUSE 30: GO SUB 1120
1090 IF Z$="1" THEN SAVE N$CODE
A,B
1100 IF Z$="2" THEN SAVE *M*;1;
N$CODE A,B
1110 GO TO 90
1120 PRINT AT 3,15;"1-ALLES ";AT
4,15;"2-KARAKTERS";AT 5,15;"3-U
DG'S"
1130 LET G$=INKEY$: IF G$="" OR
G$<"1" OR G$>"3" THEN GO TO 1130

```

```

1140 IF G$="1" THEN LET A=64600:
LET B=936
1150 IF G$="2" THEN LET A=64600:
LET B=768
1160 IF G$="3" THEN LET A=65368:
LET B=168
1170 PRINT OVER 1; BRIGHT 1;AT (
VAL G$)+2,15;"
1180 INPUT "FILE NAAM ";N$:
    IF LEN N$=0 OR LEN N$>10 THEN
GO TO 1180
1190 PRINT AT 7,0;"SAVE ";****;N
$;****;" CODE ";A;" ";B
1200 RETURN
1210 REM ** LADEN **
1220 RANDOMIZE USR 35109: PRINT
AT 2,9;"LADEN";AT 4,9;"1-CASSETT
E";AT 5,9;"2-DRIVE "
1230 PAUSE 0
1240 LET Z$=INKEY$: IF Z$="" OR
Z$<"1" OR Z$>"2" THEN GO TO 1240
1250 INPUT "FILE NAAM";N$
1260 IF Z$="1" THEN LOAD N$CODE
1270 IF Z$="2" THEN LOAD *M*;1;
N$CODE
1280 GO TO 90
1290 FOR C=9 TO 16
1300 PRINT AT C,12;"
1310 NEXT C
1320 RETURN
1330 RETURN
1340 REM *** MACHINE TAAL ***
1350 LET TEL=0
1360 RESTORE 1450
1370 FOR D=35000 TO 35152
1380 READ data: POKE D,data
1390 NEXT D
1400 RANDOMIZE USR 35000
1410 RETURN
1420
1430 REM data ROM naar RAM
(35000-35011)
1440
1450 DATA 33,0,61,17,88,252
1460 DATA 1,0,3,237,176,201
1470
1480 REM 2*BRIGHT VIERKANT
(35012-35039)
1490
1500 DATA 33,36,89,14,8,6,8,54
1510 DATA 120,229,17,16,0,25,54
1520 DATA 120,225,35,16,243,17
1530 DATA 24,0,25,13,32,234,201
1540
1550 REM DUMP-POINTER
(35040-35067)
1560 DATA 6,8,33,88,251,62,0,0
1570 DATA 95,22,0,25,16,253,6,8
1580 DATA 17,87,80,26,119,20
1590 DATA 35,16,250,68,77,201
1600
1610 REM NIEUWE KAR.SET PRINTER
(35068-35097)
1620
1630 DATA 62,2,205,1,22,14,32,6
1640 DATA 96,33,88,251,34,54,92
1650 DATA 121,215,33,0,60,34,54
1660 DATA 92,62,32,215,12,16
1670 DATA 236,201
1680

```


1690 REM 7 REGELS SCHOON
(35098-35108)

1700
1710 DATA 0,0,0,6,0,62,32
1720 DATA 215,16,250,201
1730
1740

1750 REM WEGFADEN TEKST
(35109-35031)

1760
1770 DATA 62,56,22,8,33,0,88
1780 DATA 6,0,119,35,14,40,13
1790 DATA 32,253,16,247,60,21
1800 DATA 32,238,201
1810

1820 REM BRIGHT UDG's
(35132-35152)

1830
1840 DATA 33,35,88,14,7,6,26,54
1850 DATA 120,35,16,251,6,6,35
1860 DATA 16,253,13,32,241,201
1870 RETURN

OPUS DISK UTILITY

J. Koster

Dit vrij uitgebreide basic-programma is bedoeld voor Opus Discovery-gebruikers, die niet zo vertrouwd zijn met machine-taal en toch wat meer met hun Opus willen doen, dan alleen maar laden en save van programmaatjes. Het is een soort van "onderhoudsprogramma", dat toelaat om diverse dingen met de schijf te doen, die zich op dat ogenblik in de drive bevindt: gewoon een programma laden, een uitgebreide catalogue opvragen (met lengte van de file, soort van file, aantal sectoren die de file beslaat), een file wissen, een nieuwe schijf formatteren, de catalogus op papier printen, de lege sectoren tussen de diverse files uithalen, naar een andere schijf kopiëren, de schijf een andere naam geven.

LET OP:

Het programma werkt enkel met de TASWIDE-code in het geheugen. Dit is nodig, om 64 tekens per regel te kunnen afdrucken. Misschien vindt iemand er wel iets op, om dit codeblok te vervangen door de 64-tekens-routine van W. Lageman in gids nummer 7?

Het programma bevat verder geen moeilijke dingen. Het werkt gewoon recht-toe recht-aan. Het wordt gestart met RUN.

Tenslotte nog dit, wellicht ten overvloede: op regel 300 wordt het keyword 'CAT' bedoeld, en niet de letters C A en T.

```
1 CLEAR 63222: LOAD *1: *TASWI
DE*CODE
2 LET cls=9990: LET ink=9995
3 LET cat=0
4 LET fi=0:
100 LET fi=fi-1: LET pa=1: GO S
UB cls
101 IF cat=0 THEN GO TO 195
102 LET fff=1: LET pa=1: LET pa
g=INT (fi/16)+1: LET ff=0: PRINT
AT 0,0: " Naam Soort
Start Lengte Startbl. Eind
bl. rest "
103 IF fi<=16 THEN LET f1=1: LE
T f2=fi: GO TO 107
104 IF fi>16 THEN LET f1=1: LET
f2=16: GO TO 107
107 PRINT AT 0,0: FOR z=f1 TO f
2: PRINT TAB 2-LEN STR$ z;z;TAB
3;m$(z);TAB 14;o$(z);TAB 29-LEN
STR$ e(z);e(z);TAB 40-LEN STR$ f
(z);f(z);TAB 49-LEN STR$ b(z);b(
z);TAB 55-LEN STR$ c(z);c(z);TAB
63-LEN STR$ a(z);a(z)
108 NEXT z: LET cu=1: PRINT FLA
SH 1;AT cu,63;*"
109 PRINT INVERSE 1;AT 17,0;"Ge
bruikte bytes ";fr;TAB 33;"over
op schijf ";178000-fr;TAB 64;AT
18,0;"Karacters in header: ";p$(
(pa-1)*16+cu);TAB 64
110 IF gap>5 THEN PRINT FLASH 1
;AT 19,0;gap;" gaten tussen de b
lokken, adviseer REFORM DISK";TA
B 64;
111 IF fi=0 THEN PRINT AT 10,0;
"SCHIJFNAAM ";m$(1);" "; FL
ASH 1;" SCHIJF IS LEEG "
195 PRINT AT 20,0: PAPER 2; INK
3;CHR$ 2;"Load Cat Erase Format
Print Next-page Reform-disk Mov
e Diskname ";#1;AT 0,0;"Transfer
files"
200 GO SUB ink
201 LET a$=INKEY$: IF a$="c" OR
a$="C" THEN GO TO 300
202 IF a$="n" OR a$="N" THEN GO
TO 400
```

```
203 IF a$="r" OR a$="R" THEN GO
TO 500
204 IF a$="m" OR a$="M" THEN GO
TO 510
205 IF a$=CHR$ 10 OR a$=CHR$ 11
THEN LET aa=CODE a$-9: GO TO 52
0
206 IF a$="l" OR a$="L" THEN GO
TO 550
207 IF a$="e" OR a$="E" THEN GO
TO 600
208 IF a$="f" OR a$="F" THEN GO
TO 700
209 IF a$="d" OR a$="D" THEN GO
TO 800
210 IF a$="p" OR a$="P" THEN GO
TO 900
211 IF a$="t" OR a$="T" THEN GO
TO 2000
299 GO TO 200
300 GO SUB cls: PRINT AT 10,20;
FLASH 1;" LEZEN KATALOGUS ": LE
T gap=0: CLOSE #4: OPEN #4;" CAT
";1: DIM f$(111,17):
301 FOR z=1 TO 110
302 FOR x=1 TO 16: LET f$(z)(x)
=INKEY$#4: NEXT x: NEXT z
304 FOR z=1 TO 110: IF CODE f$(
z)(5)+256*CODE f$(z)(6)=65535 TH
EN LET fi=z: GO TO 310
305 NEXT z: LET fi=z
310 DIM a(fi): DIM b(fi): DIM c
(fi): DIM n$(fi,10): DIM m$(fi,1
0): FOR z=1 TO fi: LET n$(z)=f$(
z)(7 TO 16): LET a(z)=CODE f$(z)
(1)
311 LET b(z)=CODE f$(z)(3)+256*
CODE f$(z)(4): LET c(z)=CODE f$(
z)(5)+256*CODE f$(z)(6)
312 FOR x=1 TO 10
313 IF n$(z)(x)>=" " AND n$(z)(
x)<=CHR$ 127 THEN LET m$(z)(x)=n
$(z)(x)
314 NEXT x
315 IF c(z)<b(z) THEN LET tt=b(
z): LET b(z)=c(z): LET c(z)=tt
316 NEXT z
317 DIM p$(fi+1,40): DIM o$(fi,
10): DIM e(fi): DIM f(fi)
```



```

318 LET fr=0: CLOSE #5: CLOSE #
4
319 IF fi=1 THEN GO TO 100
320 FOR z=1 TO fi-1: CLOSE #5:
OPEN #5;1;n$(z): LET a$=INKEY$#5
: IF a$="" THEN LET o$(z)="LEEG
FILE"
321 LET t=CODE a$
322 LET i=CODE INKEY$#5: LET j=
CODE INKEY$#5: LET f(z)=i+256*j:
LET fr=fr+f(z)
323 IF t=0 THEN LET i=CODE INKE
Y$#5: LET j=CODE INKEY$#5: LET e
(z)=i+256*j: LET o$(z)="Basic"
324 IF t=1 THEN LET i=CODE INKE
Y$#5: LET j=(CODE INKEY$#5)-32:
LET o$(z)="DATA "+CHR$ j+"()"
325 IF t=2 THEN LET i=CODE INKE
Y$#5: LET j=(CODE INKEY$#5)-96:
LET o$(z)="DATA "+CHR$ j+"$()"
326 IF t=3 THEN LET i=CODE INKE
Y$#5: LET j=CODE INKEY$#5: LET e
(z)=i+256*j: LET o$(z)="Code"
327 NEXT z
328 FOR z=1 TO fi: FOR x=1 TO 1
0
329 LET p$(z)(x*4-3 TO x*4)=STR
$ CODE n$(z)(x): NEXT x: NEXT z
330 LET cat=1
331 FOR z=1 TO fi-1: IF c(z)+1<
>b(z+1) THEN LET gap=gap+(b(z+1)
-(c(z)+1))
332 NEXT z: LET gap=gap-(b(fi)-
c(fi-1)): GO TO 100
400 RANDOMIZE USR 64300: PRINT
CHR$ 3,: FOR z=1 TO 19: PRINT AT
z,0;TAB 63;" ": NEXT z
401 IF pa=pag THEN LET pa=1: GO
TO 410
402 IF pa<pag THEN LET pa=pa+1
410 LET f1=(pa-1)*16+1: LET f2=
(pa)*16
411 IF f2>fi THEN LET f2=fi: GO
TO 107
412 GO TO 107
500 CLEAR #: LET dr=1
501 GO SUB cls: PRINT AT 3,0;"T
IJDENS DEZE ROUTINE MOET U UITKI
JKEN VOOR SPANNINGVERSCHILLEN"
"EN NIET SPELEN MET DE SCHIJVEN.
"GEBRUIK NIET DE "BREAK-TOE
TS", DIT KOST U UW SCHIJF": MOV
E "D";1 TO "D";dr: GO TO 300
502 IF o$(fff)(1)="B" THEN LOAD
*1;n$(fff)
510 CLEAR #: GO SUB cls: PRINT
AT 3,0;"1 = 1 drive""2 = 2 dr
ive"
511 GO SUB ink
512 LET a$=INKEY$: IF a$="1" TH
EN LET dr=3: GO TO 515
513 IF a$="2" THEN LET dr=2: GO
TO 515
514 GO TO 101
515 GO TO 501
520 RANDOMIZE USR 64300: PRINT
CHR$ 3;AT 19,0,: IF aa=1 AND cu<
16 THEN IF fff<fi THEN PRINT AT
cu,63;" ": LET cu=cu+1: GO TO 53
0
521 IF aa=2 AND cu>1 THEN PRINT

```

```

AT cu,63;" ": LET cu=cu-1: GO T
O 530
530 PRINT AT cu,63; FLASH 1;"*
: LET fff=(pa-1)*16+cu
531 GO TO 109
550 GO SUB cls
551 CLEAR #
552 IF o$(fff)(1 TO 5)="Basic"
THEN LOAD *1;n$(fff)
553 IF o$(fff)(1 TO 4)="Code" T
HEN LOAD *1;n$(fff)CODE
554 IF o$(fff)(1)="D" AND o$(ff
f)(7)="" THEN LET b$o$(fff)(5)
: LOAD n$(fff) DATA b$()
555 IF o$(fff)(1)="D" AND o$(ff
f)(7)<>"$" THEN LET q=VAL o$(fff
)(5): LOAD n$(fff) DATA q()
556 GO TO 200
600 GO SUB cls: PRINT "ERASEN
(1=ja) (ENTER=nee)": BEEP 1,30
601 GO SUB ink: LET a$=INKEY$:
IF a$="1" THEN ERASE "m";1;n$(ff
f): LET a(fff)=0: LET b(fff)=0:
LET c(fff)=0: LET e(fff)=0: LET
f(fff)=0: LET n$(fff)="" : LET m$
(fff)="" : LET o$(fff)="" : LET p$
(fff)="" : GO TO 101
602 IF a$<>"1" THEN GO TO 101
603 GO TO 601
700 GO SUB cls
701 PRINT "TOETS DE NAAM IN"
702 LET j$=""
703 LET l=LEN j$
704 IF l=10 THEN GO TO 720
705 PRINT AT 10,30-l;j$; FLASH
1;"_": FLASH 0;TAB 45;" "
706 GO SUB ink: LET a$=INKEY$
707 IF a$>=" " AND a$<=CHR$ 127
THEN LET j$=j$a$: GO TO 703
708 IF a$=CHR$ 13 THEN GO TO 72
0
709 GO TO 702
720 PRINT AT 10,30-l; FLASH 1;j
$; FLASH 0;TAB 40;" "
721 PRINT AT 12,20;"Korrekt 1=
ja": GO SUB ink
722 LET a$=INKEY$: IF a$<>"1" T
HEN GO TO 700
723 PRINT AT 14,20; FLASH 1;"FO
RMATEREN ";j$
724 LET cat=0: FORMAT 1;j$: GO
TO 100
800 DIM i$(16): GO SUB cls
801 CLOSE #4: OPEN #4;" CAT ";1
RND16
802 FOR i=1 TO 16: LET i$(i)=IN
KEY$#4: NEXT i
803 PRINT AT 10,0;"Nieuwe discn
aam ?"
804 LET j$=""
805 LET l=LEN j$: IF l=10 THEN
GO TO 820
806 PRINT AT 10,30-l;j$; FLASH
1;"_": FLASH 0;" "
807 GO SUB ink
808 LET b$=INKEY$: IF b$>=" " A
ND b$<=CHR$ 127 THEN LET j$=j$b
$: GO TO 805
809 IF b$=CHR$ 13 THEN GO TO 82
0
810 GO TO 804

```

```

820 PRINT AT 10,30-l;j$;"
*
821 POINT #4;1
822 FOR i=1 TO 6: LET i$=INKEY$
#4: NEXT i
823 FOR i=1 TO 10
824 IF i<=LEN j$ THEN PRINT #4;
j$(i);
825 IF i>LEN j$ THEN PRINT #4;"
";
826 NEXT i
827 CLOSE #4: GO TO 101
900 OPEN #14,"b"
901 GO SUB cls: PRINT AT 3,0;"1
= Label versie";AT 7,0;"2 = Tot
ale versie";AT 11,0;" 3 = return
"
902 GO SUB ink
903 LET a$=INKEY$: IF a$="3" TH
EN CLOSE #14: GO SUB cls: GO TO
101
904 IF a$="1" THEN GO TO 1000
905 IF a$="2" THEN GO TO 910
906 GO TO 902
910 PRINT #14;CHR$ 27;CHR$ 68;C
HR$ 6;CHR$ 15;CHR$ 30;CHR$ 40;CH
R$ 48;CHR$ 56;CHR$ 64;CHR$ 72;CH
R$ 0
911 PRINT #14;CHR$ 15
912 PRINT #14;CHR$ 27;CHR$ 77
913 PRINT #14;CHR$ 27;CHR$ 45;C
HR$ 1;"NAAM DISK ";m$(fi);"
VRIJE RUIMTE ";178000-fr;CH
R$ 27;CHR$ 45;CHR$ 0;CHR$ 10;CHR
$ 10;CHR$ 10
914 IF gap>5 THEN PRINT #14;"VR
IJE BLOKJES TUSSEN PROGRAMMA's:
";gap;CHR$ 10;CHR$ 10
915 PRINT #14;CHR$ 9;"Nummer";C
HR$ 9;"Naam";CHR$ 9;"Soort";CHR$
9;"Start";CHR$ 9;"Lengte";CHR$
9;"Startbl";CHR$ 9;"Eindbl.";CHR
$ 9;"Rest in eindblok";CHR$ 10;C
HR$ 10
916 FOR z=1 TO fi
917 PRINT #14;CHR$ 9;z;CHR$ 9;m
$(z);CHR$ 9;o$(z)(1 TO 7);CHR$ 9
;e(z);CHR$ 9;f(z);CHR$ 9;b(z);CH
R$ 9;c(z);CHR$ 9;a(z);CHR$ 10
918 NEXT z
919 PRINT #14;CHR$ 10;CHR$ 10;C
HR$ 10;CHR$ 10;CHR$ 10: CLOSE #1
4: GO TO 101
1000 PRINT #14;CHR$ 15: PRINT #1
4;CHR$ 27;CHR$ 77: PRINT #14;CHR
$ 27;CHR$ 68;CHR$ 1;CHR$ 5;CHR$
17;CHR$ 26;CHR$ 33;CHR$ 0
1001 PRINT #14;CHR$ 9;"Nr.";CHR$
9;"Naam";CHR$ 9;"Soort";CHR$ 9;
"Start";CHR$ 9;"Lengte";CHR$ 10;
CHR$ 10
1002 FOR z=1 TO fi
1003 PRINT #14;CHR$ 9;z;CHR$ 9;m
$(z);CHR$ 9;o$(z)(1 TO 7);CHR$ 9
;e(z);CHR$ 9;f(z);CHR$ 10
1004 NEXT z: PRINT #14;CHR$ 10
1005 PRINT #14;"DISKNAAM ";m$(
fi);CHR$ 10
1006 PRINT #14;"VRIJE RUIMTE ";1
78000-free;CHR$ 10
1007 IF gap>5 THEN PRINT #14;"Vr

```



```

ije blokjes tussen programma's "
:gap;CHR$ 10
1008 PRINT #14;CHR$ 10;CHR$ 10;C
HR$ 10;CHR$ 10: CLOSE #14: GO TO
101
2000 GO SUB cls
2001 FOR z=1 TO fi STEP 4
2002 FOR x=1 TO 4: PRINT TAB x*1
6-16;z+x-1;; " ";m$(z+x-1);: NEXT
x
2003 NEXT z
2004 PRINT AT 20,0;"TOETS IN 1=
tranfer via 1 drive / 2=transfer
via twee drives"
2005 GO SUB ink
2006 LET a$=INKEY$: IF a$="1" TH
EN LET dr=3: GO TO 2010

```

```

2007 IF a$="2" THEN LET dr=2: GO
TO 2010
2008 GO TO 2005
2010 PRINT AT 20,0;TAB 63;" "
2011 BEEP .1,10: INPUT "Number ?
(ENTER=stoppen) "; LINE q$
2012 IF q$="" THEN GO TO 101
2013 LET l=LEN q$
2014 FOR z=1 TO l: IF q$(l)>="0"
AND q$(l)<="9" THEN GO TO 2016
2015 GO TO 2011
2016 NEXT z: LET fil=VAL q$: IF
fil>fi THEN GO TO 2011
2017 BEEP .1,10: INPUT "ANDERE N
AAM ? (ENTER=NEE) "; LINE
q$
2018 IF q$="" THEN LET q$=m$(fil
)

```

```

2020 CLEAR #: MOVE "m";1;m$(fil)
TO "m";dr;q$
2030 GO TO 2000
9989 STOP
9990 PAPER 7: BORDER 0: BRIGHT 0
: INK 0: CLS : RANDOMIZE USR 643
00: PRINT CHR$ 3;: RETURN
9995 BEEP .05,20
9996 IF INKEY$<>" " THEN GO TO 99
96
9997 IF INKEY$="" THEN GO TO 999
7: RETURN
9998 RETURN
9999 SAVE *1;"B Dutility" LINE 0
: SAVE *1;"TASWIDE"CODE 63223,14
92

```

DATE UTILITIES

R. Uittenbogaard

VAR-58000 :

een variabelen-utility

Heeft u ook wel eens een programma, waarvan u denkt: welke variabelen heb ik ook alweer gebruikt? Toen ik die ervaring voor de 64e keer had, dacht ik bij mezelf: waarom geen programma maken dat de lijst variabelen op het scherm zet? En dit is het resultaat: VAR-58000.

Het programma bestaat voor 100% uit machinecode en ik heb er een uitdraai van gemaakt voor de HEXENMAKER uit Sinclair Gids 5 (dat is het programma VAR-LIST). De HEXENMAKER kan dit programma omzetten in een blok bytes.

Het programma werkt als volgt: laad de CODE en typ in:

```
PRINT STR$ USR 58000 AND 0 .
```

Dit zet de titel op het scherm: VARIABLE DUMP en daaronder verschijnt de lijst variabelen. Definieer nu zelf een paar variabelen (zoals DIM a\$(10,20,2): FOR g=1 TO 80: LET hiscore=0) en dan weer

```
PRINT STR$ USR 58000 AND 0 .
```

Zowel de naam van de variabele als het soort (array, string, enz.) verschijnt op het scherm.

De reden van de STR\$... AND 0 is dat USR een getal oplevert. Om dit getal niet op het scherm te krijgen, wordt het in een string gezet, die vervolgens niet gePRINT wordt (denk aan de SG serie over logica).

Maar waarom gebruiken we dan niet het zo vaak toegepaste RANDOMIZE USR 58000 ? Hierom: VAR-58000 kiest niet het scherm uit om op te printen. Dat kan de gebruiker zelf doen. Dus als u in plaats van PRINT een LPRINT gebruikt, komt de lijst op de printer. (Voor gevorderden: PRINT #5; enz. stuurt de lijst natuurlijk naar #5.)

Het programma VAR-DEMO geeft een demonstratie van de routine.

```

1 REM VAR-DEMO
2 REM
5 CLEAR 57999: LOAD *1;"VAR-5
8000"CODE
10 LET freeze=8: DIM r$(40): L
ET t=10: LET t$="Hello": DIM y(1
0,5): FOR f=1 TO 10: NEXT f: DIM
q$(2,30,2): LET goodbye=23
15 PRINT STR$ USR 58000 AND 0

```

```

995 REM VAR-LIST
996 REM
1000 DATA "1:VAR-58000",58000,364
1010 DATA "2184E3CDC1E32A4B:6D"
1020 DATA "5C7EFE80C832BEE3:FC"
1030 DATA "E61FC66032BDE33A:49"
1040 DATA "BEE3E6E032BEE3FE:58"
1050 DATA "60CACBE2FEA0CA09:70"
1060 DATA "E3FE80CA3FE3FEE0:5F"
1070 DATA "CA29E3FE40CAE7E2:E3"
1080 DATA "C360E3E5219CE3CD:9E"
1090 DATA "C1E3E13E3AD73E06:66"
1100 DATA "D73ABDE3D73E0DD7:04"
1110 DATA "01060009C399E2E5:93"
1120 DATA "21A2E3CDC1E3E13E:A4"
1130 DATA "3AD73E06D73ABDE3:7C"

```



```

1140 DATA *D73E24D73E0DD723:D3*
1150 DATA *4E23462309C399E2:A9*
1160 DATA *00E5219CE3CDC1E3:8C*
1170 DATA *E13E3AD73E06D73A:23*
1180 DATA *BDE3D723CDC1E33E:F3*
1190 DATA *0DD701060009C399:FE*
1200 DATA *E2E521B0E3CDC1E3:AC*
1210 DATA *E13ABDE3D73E0DD7:7C*
1220 DATA *01130009C399E2E5:10*
1230 DATA *219CE3CDC1E321A8:B6*
1240 DATA *E3CDC1E3E13ABDE3:F7*
1250 DATA *D73E28D7CDCBE33E:BD*
1260 DATA *29D73E0DD7C399E2:5A*
1270 DATA *E521A2E3CDC1E321:23*
1280 DATA *A8E3CDC1E3E13ABD:E3*
1290 DATA *E3D73E24D73E28D7:47*
1300 DATA *CDCBE33E29D73E0D:25*
1310 DATA *D7C399E217080014:71*
1320 DATA *0120564152494142:07*
1330 DATA *4C452044554D5020:42*
1340 DATA *14000D8D4E756D62:85*
1350 DATA *65F2537472696EE7:A1*
1360 DATA *2061727261793A86:5A*
1370 DATA *436F6E74726F6C20:66*
1380 DATA *7661723A86000000:76*
1390 DATA *007ECBFFD7CB7EC0:65*
1400 DATA *2318F6E511BF323:71*
1410 DATA *EDA0EDA046C5234E:27*
1420 DATA *2346E5C5EFA03823:96*
1430 DATA *23C1712370CDE32D:68*
1440 DATA *E1C13E2CD710E6E1:69*
1450 DATA *232323ED4BBFE309:03*
1460 DATA *3E08D7C9:A3*

```

FILL :

een grafische utility

Ik heb verschillende FILL routines in actie gezien, en ze waren allemaal enkel in staat eenvoudige figuren op te vullen zoals cirkels en rechthoeken. Ik heb zelf geprobeerd om een FILL routine te maken die IEDER figuur kon inkleuren. Het resultaat is is misschien niet zo snel, maar wel volledig.

Ook van dit programma heb ik een HEXENMAKER-listing gemaakt (het programma FILL_LIST). Natuurlijk kan deze CODE ook ingetypt worden zonder de HEXENMAKER; met een goede HEX-LOADER moet het lukken.

Het gebruik van dit programma is heel eenvoudig. Een voorbeeld:

```

CIRCLE 128,87,87: CIRCLE 128,87,
77: POKE 60001,128: POKE 60002,5
: RANDOMIZE USR 60000

```

Dus:

```

POKE 60001,(x-coordinaat):
POKE 60002,(y-coordinaat):
RANDOMIZE USR 60000

```

Soms ziet het inkleuren van een figuur er vreemd uit. Af en toe lijkt het of de routine een stuk overslaat, of dat hij helemaal niets doet. In dat geval zoekt hij juist naar een nog niet ingekleurd stuk. Na terugkeer naar BASIC is de figuur in ieder geval helemaal ingekleurd.

Wat betreft de kleuren: de routine reageert hetzelfde als het PLOT-commando, dat wil zeggen: INK, OVER en INVERSE worden bepaald, en PAPER, BRIGHT en FLASH staan op 8.

Maar LET OP!! Als de computer in OVER 1 of INVERSE 1 staat, komt de routine nooit klaar met inkleuren: hij blijft hangen. Zorg er dus altijd voor eerst in OVER 0 en INVERSE 0 te schakelen.

Ook van dit programma heb ik een demo gemaakt: FILL_DEMO

Ik hoop dat u deze routines in uw eigen programma's kunt gebruiken.

```

1 REM FILL_DEMO
2 REM
5 CLEAR 59999: LOAD *1:*FILL_
CODE*CODE
10 PLOT 50,50: DRAW 15,0: DRAW
0,30,PI: DRAW 0,10,-PI: DRAW 15
,0: DRAW 0,10: DRAW -15,0: DRAW
0,-30,PI: DRAW 0,-10,-PI: DRAW -
15,0: DRAW 0,-10
15 PLOT 120,50: DRAW 7,0: DRAW
0,20: DRAW -12,0: DRAW 0,-7: DR
AW 5,0: DRAW -5,15,-1.25*PI: DRA
W 10,0: DRAW 0,10: DRAW -10,0: D
RAW 5,-35,1.125*PI: DRAW 0,-3
20 PRINT AT 16,5:"Sinclair Gid
s"
25 INK 2: POKE 60001,52: POKE
60002,52: LET mc=USR 60000
30 INK 1: POKE 60001,122: POKE
60002,52: LET mc=USR 60000

```

```

995 REM FILL_LIST
996 REM
1000 DATA *1:FILL_CODE*,60000,82

```

```

1010 DATA *0180572100003922:4D*
1020 DATA *AFEAC5C5CDE522C1:C3*
1030 DATA *0CCDA6EA0DB72003:60*
1040 DATA *0C18EF0DCDA6EA0C:A3*
1050 DATA *B720030D18E404CD:D6*
1060 DATA *A6EA05B720030418:B7*
1070 DATA *D905CDA6EA04B720:50*
1080 DATA *030518CE2AAFEAA7:9A*
1090 DATA *ED72C8C118C5C5CD:A7*
1100 DATA *CE22CDD52DC1C948:E9*
1110 DATA *EA00:46*

```

CHAN-59000 :

een kanaal-utility

Als u vaak met "channels" en "streams" werkt (daarvoor worden de commando's OPEN # en CLOSE # gebruikt) dan zult u op hetzelfde probleem gestoten zijn als ik: als je een paar kanalen achter elkaar opent, moet je er wel je hoofd bij houden, anders vergeet je welk kanaal aan welke stream verbonden is. Vandaar deze utility: deze machinecode-routine drukt een overzicht van alle streams op het scherm af, en de kanalen die daarmee verbonden zijn.

Het programma CHAN_LIST is ervoor gemaakt om door het programma HEXENMAKER (uit Sinclair Gids 5) omgezet te kunnen worden in een blok bytes. Zonder dat programma lukt het, met een goede HEX-LOADER, natuurlijk ook. Vergeet in dat geval de checksum (achter de dubbele punt). De CODE is 285 bytes lang en begint op adres 59000.

Het programma werkt als volgt: typ RANDOMIZE USR 59000 en op het scherm verschijnt in de linker kolom de lijst van streams en in de rechter kolom de lijst kanalen met specificaties. Voorbeeld: typ

```
OPEN #8,"p"
```

```
PRINT #8 schrijft nu dus
naar de printer, omdat #8 aan
kanaal "p" gekoppeld is. Maar
dat is verder onzichtbaar.
Typ nu:
```

```
RANDOMIZE USR 59000
```


Nu is het wel zichtbaar: op het scherm verschijnt achter #8 het kanaal "p". Het programma is ook in staat Interface 1 en Discovery kanalen te herkennen.

Het programma CHAN_DEMO geeft een demonstratie van de werking. Het programma is bedoeld voor een Discovery disc drive, dus als u die niet heeft laat u gewoon de commando's weg die een syntax error opleveren.

Bovenaan het scherm verschijnt iets wat voor sommige mensen misschien nieuw is: de streams FD, FE en FF. Deze zijn alleen via machinecode te bereiken. (De ZX Spectrum gebruikt deze intern). Daar hoeft u zich verder geen zorgen over te maken. Het kan misschien wel handig zijn als u in machine-code werkt.

Terwijl ik dit programma maakte, ontdekte ik bijzondere kanalen (voor de Discovery). Bijvoorbeeld: CHR\$ 0 als naam van een kanaal verwijst naar kanaal "m".

Voorbeeld:

```
OPEN #5,"m",1,"naam"
```

Maar dit:

```
OPEN #5,CHR$ 0,1,"naam"
```

is hetzelfde. CHR\$ 3 verwijst naar een andere stream.

Voorbeeld:

```
OPEN #6,#10
```

Dit is hetzelfde als

```
OPEN #6,"#";10
```

volgens de handleiding, en nu dus ook:

```
OPEN #6,CHR$ 3,10
```

CHR\$ 143 (grafisch symbool op toets 8) verwijst naar het "CODE" kanaal. Dus:

```
OPEN #9,CHR$ 143
```

Het ziet er waarschijnlijk ingewikkelder uit dan het is. Een ander leuk grapje is het volgende:

```
OPEN #3,CHR$ 3,3:PRINT #3
```

Eerst wordt #3 aan #3 gekoppeld en dan wordt er informatie naar #3 gestuurd. Dit zou betekenen dat het systeem blijft hangen. Maar dat doet het niet: probeer het maar eens uit.

Ik hoop dat u dit programma nog eens kunt gebruiken, als u de kluts kwijtgeraakt bent.

```
5 REM CHAN_DEMO
6:
10 CLEAR 58999: CLEAR #: LOAD
*1: "CHAN-59000" CODE
15 OPEN #9,"b": OPEN #14,#6: 0
```

```
PEN #4,"t";2: OPEN #7,"CODE":0
PEN #6,"CAT":11N: OPEN #12,"j
";1: OPEN #15,"d";11N
20 RANDOMIZE USR 59000
```

```
995 REM CHAN_LIST
996:
1000 DATA "1:CHAN-59000",59000,2
85
1010 DATA "CD6B0D3E02CD0116:64"
1020 DATA "1181E7AFCD0A0CDD:EF"
1030 DATA "2A4F5C01105C3E20:AC"
1040 DATA "D73E20D73E23D760:BE"
1050 DATA "69114B8B197ED723:05"
1060 DATA "7ED73E17D73E0AD7:CE"
1070 DATA "D70A6F030A6703B5:B2"
1080 DATA "2836DDE5D1191103:60"
1090 DATA "0019E5D13E22D71A:6C"
1100 DATA "D73E22D71AFE8F28:35"
1110 DATA "2BFE032835FE4D28:5C"
1120 DATA "5FA7285CFE54284C:BA"
1130 DATA "FE4A2848FE442844:DA"
1140 DATA "FECF2840FE23283C:3A"
1150 DATA "3E0DD721365CA7ED:F1"
1160 DATA "42C8189A3E08D73E:A9"
1170 DATA "08D73EAFD73E22D7:78"
1180 DATA "18E63E08D73E08D7:DE"
1190 DATA "3E08D73E23D71313:2B"
1200 DATA "131ACB273D2162E7:80"
1210 DATA "856F3001247ED723:85"
1220 DATA "7ED718C43E3BD713:64"
1230 DATA "13131AC630D718B8:B5"
1240 DATA "3E08D73E08D73E4D:A7"
1250 DATA "D73E22D73E3BD721:6D"
1260 DATA "0300197EC630D73E:9B"
1270 DATA "3BD73E22D7C5060A:20"
1280 DATA "237ED710FBC13E22:AF"
1290 DATA "D7188D4644464546:EA"
1300 DATA "4630303031303230:B4"
1310 DATA "3330343035303630:B7"
1320 DATA "3730383039313031:C9"
1330 DATA "3131323133313431:C7"
1340 DATA "358053545245414D:C6"
1350 DATA "170A004348414E4E:D6"
1360 DATA "454C0D8D00:82"
```

128K

BESCHERMEN VAN JE EIGEN PROGRAMMA'S

Velen die hun eigen programma's schrijven, zien niet graag dat die programma's door jan-en-alleman maar bekeken worden en veranderd worden. Hieronder een aantal 'POKE's die een programma alvast gedeeltelijk 'op slot' zetten:

Hints &

De BREAK-toets uitschakelen doe je met: POKE 23613,100

De EDIT-toets uitschakelen met: POKE 23743,200

POKE 23756,0 maakt van programmalijn 10 programmalijn 0. (Dit geldt alleen voor een Spectrum zonder Interface 1!)

Tips

Je programma zgn. MERGE-proof maken doe je als volgt:

```
POKE(PEEK 23637+ 256*
PEEK 23638),255:SAVE"naam"
LINE 0.
```

Dit MERGE-proof maken kan van nut zijn om sommige kopieer-programma's de das om te doen.

Wie heeft er nog meer goede ideeën?

Snel en precies rekenen op de ZX81 (8b)

H. van Abbe

OPMERKING

Zoals toegelicht bij de bespreking van **Factor**, gedraagt **SQR N** zich "raar" voor $N > 4294967280$. In de **CALCULATOR** is dat gedrag echter anders dan op grond van figuur 7 zou worden verwacht: **INT SQR N** is dan namelijk 8000 hex en niet 0000.

Eenzijds betekent dit dat een korrektieterm, vergelijkbaar met ($K=256$) in de regels 45 en 50 van figuur 6A niet nodig is. Anderzijds zou de ontbinding tussen $N=4294967280$ en $N=2^{**}32$ mis kunnen gaan. Dat gebeurt echter niet, omdat alle betreffende getallen ook delers < 32768 ($=8000$ hex) hebben, behalve 4294967291 . Dit is evenwel een **PRIEM**getal en het gevolg is alleen dat de **RTN** delers > 32768 voor dit bewuste getal niet onderzoekt. Fundamenteel fout, maar zonder kwalijke gevolgen.

In mijn vorige artikel heb ik gesteld dat in regel 110 **IF D=1** nodig is bij het onderzoek van getallen $> 65521^{**}2$. Die

test, nu met **INC D** in de **NEXT-DIVISOR-RTN**, heeft geen betekenis voor het zoëven genoemde **PRIEM**getal, aangezien gestopt wordt < 32768 . Voor **priem**getallen < 4294967280 is **INT SQR N** echter 65535 en is de "dreiging" **D=1** wel aanwezig, vandaar **TEST-END** in de juist genoemde **RTN**. Het "gevaarlijkste" **PRIEM**getal is dan ook 4294967279 .

TIMING

In figuur 9 staan 19 voorbeelden van ontbindingen "genummerd" A t/m S, met de tijden voor:

Factor het verkorte **BASIC-PG**
Factor "Publ." als besproken de vorige keer
Factor "Snlr." hetzelfde **PG** met de snelste **DIVISION-LOOP**
Factor het nu behandelde **PG**

Twee opmerkingen vooraf:

- het verkorte **BASIC-PG** is sneller dan het oorspronkelijk juni 1985 door mij gepubliceerde dat herhaald is in deel (6); daarom zijn de eerder genoemde maximale rekestijden van ruim 7 minuten gereduceerd tot 6 min 41 sec

- een verdere versnelling van ca. 3% is mogelijk door in het **PG** van figuur 3 van regel 50, regel 5 te maken; daardoor komt de het vaakst gebruikte variabele **D** voortaan in **VARS** te staan en dat brengt deze tijdswinst.

In figuur 9 valt in de eerste plaats op, dat de voorbeelden C, F, K en S veruit het snelst worden ontbonden in **BASIC**! De betreffende 4 getallen bevatten steeds een groot aantal kleine factoren. En zo'n kleine factor kost in het **BASIC-PG** 1/7 sec per factor, in het snelste **MC-PG** 0.23 sec per factor.

Dit verschil in benodigde tijd is voornamelijk te danken aan de test **IF D>N** in regel 80 van figuur 3. Ik heb deze regel ook functioneel geïntegreerd in de **MC-RTN** van **Factor** met tot 50% tijdswinst voor de voorbeelden met vele factoren, echter met 20% tijdverlies voor het - veel langer durende - onderzoek van **PRIEM**getallen en getallen met grote factoren. Daarom is deze test **NIET** ingevoerd.

De tijdswinst van de **MC-RTNS** blijkt 't duidelijkst bij grote factoren: daar is **Factor** tot 22 maal zo snel als het snelste **BASIC-PG**. Voorbeelden: H, I, J, L, M, O, P en Q.

Zoals hiervoor verklaard, wordt voorbeeld Q extra snel berekend door de beperking van de deler tot $< 2^{**}16$. Anders was de benodigde tijd bijna tweemaal zo groot.

Factor gebruikt met de nu ingevoerde snellere **DIVISION-LOOP** voor een **PRIEM** en bij grote factoren tot 11% minder tijd, zie de tabel van figuur 9.

FIG. 9 ONTBINDING IN PRIEMFACTOREN

T I M I N G in seconden

N#	FACTOREN	FACTOR Verkort	FACTOR Publ.	FACTOR Snlr.
A	12345	1.9	1.6	1.6
B	12347	1.1	1.1	1.1
C	12346	1.1	1.1	1.1
D	12346	1.1	1.1	1.1
E	12346	1.1	1.1	1.1
F	12346	1.1	1.1	1.1
G	12346	1.1	1.1	1.1
H	12346	1.1	1.1	1.1
I	12346	1.1	1.1	1.1
J	12346	1.1	1.1	1.1
K	12346	1.1	1.1	1.1
L	12346	1.1	1.1	1.1
M	12346	1.1	1.1	1.1
N	12346	1.1	1.1	1.1
O	12346	1.1	1.1	1.1
P	12346	1.1	1.1	1.1
Q	12346	1.1	1.1	1.1
R	12346	1.1	1.1	1.1
S	12346	1.1	1.1	1.1

Tenslotte : **FACTOR** brengt ten opzichte van **FACTOR-Snlr.** nog tot bijna 50% tijdswinst bij getallen met vele factoren, echter begrijpelijkerwijs slechts enkele procenten bij getallen met grote delers of **PRIEM**getallen.

DELING-PROGRAMMA TOT 2**32

Vorige keer heb ik geïndiceerd hoe ik met de "volledige" **DIVISION-RTN** en het "afmaken" van het quotient een exact **DELING-PROGRAMMA** voor getallen tot 2**32 kan worden gemaakt. Omdat ik me realiseer dat dit voor velen nog hoofdbreken kost, als "toegift" in figuur A het programma.

Na de eerdere bespreking van het eenvoudige **DELING-PG** van figuur 5 en het in dit artikel behandelde, zullen de bij-schriften in figuur A voldoende de toelichting vormen om het **PG** en de **RTN** te begrijpen.

LET OP :

Na adres 40CD **EXX** worden alle verdere instructies uitgevoerd met de 'exchange registers'.

Na **RST 08** moet een **DEFB** volgen die er voor zorgt dat na uitvoering van regel 6 doorgegaan wordt met regel 7; **ED** is zo'n **CODE**.

Regel 6 en 7 mogen niet worden gecombineerd.

De volgende keer gebruiken we de verworven kennis om **priem**getallen te genereren.

WIE KAN MIJ HELPEN

Zoals blijkt uit mijn in dit periodiek gepubliceerde **MC-routines** maak ik nuttig gebruik van de met de **CALCULATOR** mogelijke 'operations'. Maar de aandachtige lezer zal hebben bemerkt dat ik de 'COMPARISON' operations : **09 <= 0A >= 0B < 0C > 0D <** NIET heb gebruikt. Terwijl dat op een aantal plaatsen toch tot eenvoudiger routines zou hebben geleid. En wie mij kent, weet dat ik zo'n mogelijkheid altijd benut.

De reden van het niet benutten is dat deze operations bij mij niet goed werken. In plaats van een "antwoord" 'true' (=1) of 'false' (=0), afhankelijk van de twee 'values' die vergeleken worden, krijg ik altijd 'true' bij gelijke waarden en 'false' bij ongelijke waarden, onafhankelijk van de grootte van die waarden.

Een eenvoudige testroutine is bijvoorbeeld :

EF/Ax/Ay/zz/34/C3DB15

met **Ax** en **Ay** één van de 5 **STK-constant CODEs** en **zz** één van de 5 bovengenoemde operations **CODEs**.

Wie, o wie, kan mij vertellen of ik een denkfout maak en zo ja, welke. Of zijn mijn twee **ZX81's**, waarvan 1 een **TIMEX** is, niet in orde ?

Gaarne reacties, waarvan ik verslag zal doen in dit periodiek.

FIG. A N/D=0 R= N(2**32 D<2**16

```
1 REM E(RNDY?M74 UNPLOT )
INPUT N$
LET N=VAL N$
INPUT O
RRND O
PRINT N$;" / ";D;" = ";USR 1651
7 PRINT " R=";USR 1653
```

HEXDUMP

```
4082--2A10 403E 73BE 2320--22C
408A--FC11 7640 0105 00E0--22B6
4092--B0EF E530 0041 002E--323
409A--34CD 3A15 C5CD 3A15--3D1
40A2--C5D9 E1C1 78D9 ED58--450
40AA--3240 2100 0006 20D9--47A
40B2--ED6A CB11 1709 ED6A--47A
40BA--3808 ED52 3007 19A7--276
40C2--1804 A7ED E523 10E7--330
40CA--2250 40D9 ED6A CB11--3BE
40D2--1747 E5CD 2015 C1CD--3D3
40DA--2015 EF01 3000 4100--196
40E2--040F C530 3748 2E2D--1E2
40EA--3040 4474 2403 3200--181
40F2--1501 C5A4 0332 2C00--1E0
40FA--08A0 0134 C00B 15EF--389
4102--34CD DB15 EF55 34C0--4C6
410A--DB15 CFED 4850 40C9--450
```

```
16514-2A10 403E 73BE 2320--556
16522--FC11 7640 0105 00E0--55A
16530--B0EF E530 0041 002E--303
16538--34CD 3A15 C5CD 3A15--377
16546--C5D9 E1C1 78D9 ED58--1497
16554--3240 2100 0006 20D9--402
16562--ED6A CB11 1709 ED6A--1146
16570--3808 ED52 3007 19A7--630
16578--1804 A7ED E523 10E7--816
16586--2250 40D9 ED6A CB11--958
16594--1747 E5CD 2015 C1CD--979
16602--2015 EF01 3000 4100--406
16610--040F C530 3748 2E2D--482
16618--3040 4474 2403 3200--385
16626--1501 C5A4 0332 2C00--480
16634--08A0 0134 C00B 15EF--905
16642--34CD DB15 EF55 34C0--1222
16650--DB15 CFED 4850 40C9--1104
```

```
MC-ROUTINE
16514 4082--2A10 LD HL,(4010) YARS
SFRK 4083--B0EF LD A,73 DIVIDEND "N"
4084--34CD LD HL,(HL)
4085--C5D9 LD HL,NZ,4087 SEEK
4086--3240 LD DE,4076 DEH-S
4087--ED6A LD BC,0000 TRFR
4088--3808 RST 20;FPR:
4089--1804 CALL 150A N=MOD-65536
408A--2250 LD HL,0000 HI-N -BC
408B--040F LD HL,150A FI-T0-BC
408C--3040 LD HL,N LO-N
408D--1501 LD HL,0000 LO-N in N11
408E--08A0 LD HL,0000 HI-N in A, C1
408F--34CD EXX
4090--DB15 LD DE,(4032) DIVISOR "D"
4091--CFED LD HL,0000 RES REMAINDER
4092--B0EF LD HL,200
LOOP 40B1--09 EXX HL,HL
40B2--ED6A LD HL,C
40B3--FC11 LD HL,C
40B4--34CD LD HL,HL
40B5--C5D9 LD HL,HL
40B6--3240 LD HL,HL
40B7--ED6A LD HL,HL
40B8--3808 LD HL,HL
40B9--1804 LD HL,HL
40BA--2250 LD HL,HL
40BB--040F LD HL,HL
40BC--3040 LD HL,HL
40BD--1501 LD HL,HL
40BE--08A0 LD HL,HL
40BF--34CD LD HL,HL
40C0--DB15 LD HL,HL
40C1--CFED LD HL,HL
40C2--B0EF LD HL,HL
40C3--34CD LD HL,HL
40C4--C5D9 LD HL,HL
40C5--3240 LD HL,HL
40C6--ED6A LD HL,HL
40C7--3808 LD HL,HL
40C8--1804 LD HL,HL
40C9--2250 LD HL,HL
40CA--040F LD HL,HL
40CB--3040 LD HL,HL
40CC--1501 LD HL,HL
40CD--08A0 LD HL,HL
40CE--34CD LD HL,HL
40CF--DB15 LD HL,HL
40D0--CFED LD HL,HL
40D1--B0EF LD HL,HL
40D2--34CD LD HL,HL
40D3--C5D9 LD HL,HL
40D4--3240 LD HL,HL
40D5--ED6A LD HL,HL
40D6--3808 LD HL,HL
40D7--1804 LD HL,HL
40D8--2250 LD HL,HL
40D9--040F LD HL,HL
40DA--3040 LD HL,HL
40DB--1501 LD HL,HL
40DC--08A0 LD HL,HL
40DD--34CD LD HL,HL
40DE--DB15 LD HL,HL
40DF--CFED LD HL,HL
40E0--B0EF LD HL,HL
40E1--34CD LD HL,HL
40E2--C5D9 LD HL,HL
40E3--3240 LD HL,HL
40E4--ED6A LD HL,HL
40E5--3808 LD HL,HL
40E6--1804 LD HL,HL
40E7--2250 LD HL,HL
40E8--040F LD HL,HL
40E9--3040 LD HL,HL
40EA--1501 LD HL,HL
40EB--08A0 LD HL,HL
40EC--34CD LD HL,HL
40ED--DB15 LD HL,HL
40EE--CFED LD HL,HL
40EF--B0EF LD HL,HL
40F0--34CD LD HL,HL
40F1--C5D9 LD HL,HL
40F2--3240 LD HL,HL
40F3--ED6A LD HL,HL
40F4--3808 LD HL,HL
40F5--1804 LD HL,HL
40F6--2250 LD HL,HL
40F7--040F LD HL,HL
40F8--3040 LD HL,HL
40F9--1501 LD HL,HL
40FA--08A0 LD HL,HL
40FB--34CD LD HL,HL
40FC--DB15 LD HL,HL
40FD--CFED LD HL,HL
40FE--B0EF LD HL,HL
40FF--34CD LD HL,HL
4100--ED4B5040 LD BC,(4050) REMAINDER
4101--C9 RET
```

Ir. H.H. van Abbe
Van der Doeslaan 1
2242 PP Wassenaar
Tel. 01751-14216

Cursus machinetaal op de ZX Spectrum - deel 8

M. Dijkema

Wat ik nu ga behandelen is het beeldscherm. Probeer eens het volgende programma uit en observeer nauwkeurig wat er gebeurt.

```
10 FOR N=16384 TO 23296: PRINT
AT 0,0:N: POKE N,170: NEXT N
```

Als je goed hebt gekeken, zullen je wel enkele merkwaardigheden zijn opgevallen. Deze vat ik even samen.

Het scherm wordt verdeeld in twee duidelijk verschillende delen. Een deel waarin allerlei lijntjes op het scherm worden gezet. Ieder lijntje bestaat uit een bepaald aantal stipjes. Deze stipjes worden pixels genoemd. Ook is er een deel waar de kleuren erin worden gepoked. Dit heet het attribute-scherm. We hebben het scherm dus nu in twee delen opgedeeld, het pixel-deel en het attribute-deel.

Probeer eens:

```
FOR N=16384 TO 16384+31:POKE N,
255:NEXT N.
```

Daar is uit te zien dat de breedte van het scherm 32 is. Dit geldt ook voor het attribute-gedeelte. Ieder lijntje van het pixelgedeelte is 8 pixels (2 tot de macht 8 = 256) breed, dus op de breedte van het scherm kunnen 256 pixels staan. Dit komt precies overeen met wat we van het commando PLOT afweten. De hoogte van het scherm is 192 lijntjes of 24 attributes. Dat laatste komt precies overeen van wat we van PLOT en AT weten als we in ogenschouw nemen dat we op de onderste 2 regels niet kunnen plotten.

Wat nog meer opvalt, is dat het pixel-gedeelte zowel als het attribute-gedeelte van links naar rechts en van boven naar beneden wordt ingevuld. Bovendien is het pixelgedeelte in 3 gelijke delen verdeeld. Deze drie delen worden op dezelfde manier ingedeeld. Eerst de eerste lijn, dan de achtste, dan de zestiende enz. tot en met de 56ste lijn. Dan wordt de 2de lijn ingevuld, de 9de enz. tot en met de 57ste. Dit gaat zo door totdat uiteindelijk de 63ste lijn is getekend. Daarna komt een volgend pixelblok aan de beurt. Als we even aan het rekenen slaan, kunnen we vlug uitrekenen hoe groot bepaalde delen van het scherm zijn: 32 in de breedte*192 in de hoogte = 6144 bytes. 32 in de breedte*24 in de hoogte = 768 bytes voor het attribute-scherm. $6144/3=2048$ per pixelgedeelte. In HEX-code wordt de werking van het scherm nog veel vlugger duidelijk.

```
#4000..#401F 1e pixelregel
#4100..#411F 2e pixelregel
#4200..#421F enzovoorts
```

```
....
#4020..#403F 9e regel
#4120..#413F 10e regel
...(#4040,#4060,#4080,$40A0,
#40C0,#40E0)
#4800..#481F 64e regel
#5000..#501F 128e regel
#5800..#581F 1e attr-regel
#5820..#583F 2e attr-regel
enz. tot
#5900..#5AFF 32e attr-regel
```

Als ik in een pixelregel een POKE doe, vul ik meteen acht pixels in. Vaak is het handig om het pixelscherm te kunnen gebruiken als een X- en een Y-as zodat iedere pixel zijn eigen X- en Y-coördinaat heeft. Daarvoor moeten we eerst een punt definiëren. Hier heb ik X=0, Y=0 gedefinieerd als de uiterste hoek linksonder. (Dus twee schermregels lager dan bij PLOT).

Vraag :

Wat is het verschil tussen schermregels, pixel-regels en attribute-regels ?

Antwoord :

Een pixelregel is slechts 1 pixel hoog. Een schermregel daarentegen acht pixels. Op een schermregel kan een teken staan, op een pixelregel niet. Een attribute-regel bevat de kleur van het teken, een attribute-regel heeft een hoger adres dan de bijbehorende schermregel. Bij een attribute horen 8*8 pixels. Zo'n blokje van 8*8 pixels noemen we een karaktercel of ook een cel.

Nu gaan we een pixeladres uit elkaar rafelen. Bijvoorbeeld:

```
#4731 =
20100 0111 0011 0001
|  |  |  |  |----kolom 0..31
|  |  |  |  |--- regel 0..7
|  |  |  |  |---regels 8,16,24,...
|--pixel deel #4000,#4800,#5000
```

en een attribute adres:

```
#5A22 =
20101 1010 0010 0010
|  |  |  |  |----kolom 0..31
|  |  |  |  |---attr regel 0..23
opgepast dat dit regel-
nummer niet groter wordt
dan 23, anders kom je
boven #5800 terecht en dat
is teveel. In 128K mode
kan zelfs een reset
volgen.
```

We hadden echter X=0, Y=0 onderin het scherm gedefinieerd (in les 10 zal blijken waarom) dus krijgen we het volgende programma om van de pixelcoördinaten naar het pixeladres te rekenen.


```

pixadr PUSH BC
LD B,H
LD C,L
LD A,191
SUB B
JR C,pixart ;Y coördinaat is
;maximaal 191.
r1 LD B,A ;b=191-Y coörd.
AND A
RRA ;A=X0b7b6b5b4b3b2b1
SCF
RRA ;A=X10b7b6b5b4b3b2
AND A
RRA ;A=X010b7b6b5b4b3
XOR B
AND #F8 ;A=X010b7b6b2b1b0
XOR B
LD H,A ;H=highbyte pixeladr.
LD A,C ;A=Xc7c6c5c4c3c2c1c0
RLCA
RLCA ;A=Xc2c1c0c7c6c5c4c3
XOR B
AND #C7
XOR B ;A=Xc2c1b5b4b3c5c4c3
RLCA
RLCA ;A=Xb5b4b3c7c6c5c4c3
LD L,A ;L=low byte pixeladr.
LD A,C
r2 AND #07 ;A=0..7 pixelbit
pixart POP BC
RET

```

Het stuk r1 ... r2 staat ook in het ROM; we kunnen net zo goed schrijven : CALL #22B0.

Het gebruik van de routine is als volgt : H = X-coörd. L = Y-coörd. Bij uitgang van de routine : HL=pixeladres. A is pixelbit (er waren immers 8 pixels op 1 pixellijntje). Nu is A echter 0..7 maar we moeten met A het juiste bit aangeven, het 0de tot en met het 7de bit. Dat kan met de volgende routine heel vlug worden gedaan.

```

mplex AND X00000111
RLA ;isoleer 0..7
RLA
RLA ;LET A=A*8
ADD A,#C7 ;maak 2e deel SET
;instructie
LD (plex),A
XOR A
DEFB #CB
plex DEFB #00
RET

```

N.B. #CB C7=SET 0,A
#CB CF=SET 1,A
#CB D7=SET 1,A enz..

Pas wel op als je bovenstaande techniek zelf gebruikt, je moet wel erg zeker van je zaak zijn wil je hier geen fouten maken.

Een dergelijke routine voor de attributes met X=0 en Y=0 (breedte X : 0..31, hoogte Y : 0..23) is de volgende:

```

atadr PUSH BC ;H=Y coörd
PUSH DE ;L=X-coörd
LD B,A
LD A,L
CP 32
CCF ;X>=32 is fout
JR C,atadre
LD A,H
CP 24
CCF ;Y>=24 is fout
JR C,atadre
LD A,23 ;X=0,Y=0 is linksonder
SUB H
LD D,H ;LET D=Y
LD E,L ;LET E=X
LD HL,22528 ;22528=start attr
RLC E
RLC E
RLC E
RR D
RR E
RR D
RR E
RR D ;DE klaar gemaakt
RR E ;bij 22528 op tellen
ADD HL,DE
LD A,B
POP DE
POP BC ;uitkomst Carry=1 als
RET ;een fout is gemaakt
;anders HL=attr adres.

```

Nu geef ik nog een aantal routines die zeer zeker handig kunnen zijn als je zelf gaat programmeren.

```

lasty PUSH DE ;bereken vorige adres in
PUSH BC ;Y-richting.
LD B,A
CALL lastym
LD A,B
POP BC
POP DE
RET
lastym LD A,H
AND X00000111
INC A
BIT 3,A
JR Z,lstlin
LD A,H
AND X11111000
LD H,A
LD A,L
AND X11100000
ADD A,X00100000
JR NC,lstblk
LD A,L
AND X00011111
LD L,A
LD A,H
AND X11111000
ADD A,X00001000
LD H,A
XOR A
checkm CALL INSCR
RET
lstlin INC H
JR checkm

```

```

lstblk ADD A,L
ADD A,X00100000
LD L,A
JR checkm
inscr PUSH HL ;is HL wel in attr deel?
PUSH DE
LD DE,16384
SBC HL,DE
POP DE
POP HL
RET C ;carry als het niet zo is.
PUSH HL
PUSH DE
LD DE,16384+2048+2048+2048
SBC HL,DE
CCF
POP DE
POP HL
RET

```

Over deze en volgende routines het volgende : X=Y=0 wordt weer linksonder gekozen. De routines zijn gegarandeerd niet de kortste, omdat ze ook nog controleren of je buiten het scherm bent gekomen. Dit is echter wel veel zekerder. Als je 0.0 anders kiest zullen de routines er ook anders uit gaan zien. Er is zo'n grote verscheidenheid aan routines omdat je vooral bij het programmeren van spelen je vlug vorige of volgende adressen moet kunnen bereken zonder iedere keer opnieuw van coördinaat naar adres te moeten rekenen. Dit laatste zou een zeer traag en sterk knipperend spel geven, heb ik uit ervaring geleerd.

```

nexty PUSH DE
PUSH BC
LD B,A
CALL nextym
LD A,B
POP BC
POP DE
RET
nextym LD A,H
AND X00000111
DEC A
BIT 3,A
JR Z,nxtlin
LD A,H
AND X00000111
LD H,A
LD A,L
AND X11100000
SUB X00100000
JR NC,nxtblk
LD A,L
OR X11100000
LD L,A
LD A,H
SUB X00001000
LD H,A
XOR A
checkn CALL inscr
;Voor inscr zie lasty
RET
nxtlin DEC H
JR checkn

```



```

nextblk LD A,L
SUB X0100000
LD L,A
JR checkn

```

```

lstys PUSH BC ;nextys gelijk,
LD B,A behalve het
XOR A onderstaande
RR H
RRA
RR H
RRA
RR H
RRA
RL L
RL L
RL L
RRA
RR L
RRA
RR L
RRA
RR L
RRA
RR H
RRA
RR H
RRA
ADD A,B SUB B
LD B,A
LD A,H
ADC A,O SBC A,O
LD H,A
LD A,B
RLA
RL H
RLA
RL H
RL L
RLA
RL L ;zowel nextys als lstys
RLA ;geldt het volgende voor
RL L ;HL=pixeladres
RLA ;A=aantal op te schuiven
RR L ;pixellijnen voor nextys
RR L ;omhoog en voor lstys
RR L ;omlaag
RLA ;BV:HL=16384 A=2
RL H ;CALL nextys geeft
RLA ;HL=16896
RL H
RLA
RL H
CALL inscr
POP BC
RET

```

```

nextx RLCA lastx RRCA
RET NC
PUSH AF
LD A,L
AND X00011111
INC A INC A
BIT 5,A
JR NZ,nextx2
POP AF
SCF
CCF
INC L DEC L
RET
nextx2 LD A,L
AND X1110000
LD L,A
POP AF
SCF
RET

```

Berekent uit HL=pixeladres en A als uit mplex komend, het volgende adres in HL en de volgende pixel (in A) in positieve X-richting.

```

nextxs PUSH DE lstxs is hetzelfde
PUSH BC behalve het
LD C,A onderstaande
LD A,B
AND X11111000
RRCA
RRCA
RRCA
LD D,A
LD A,B
AND X00000111
XOR A
nextxs2 RLC C RRC C
ADC A,O
DJNZ nextxs2 SUB D
ADD A,D
LD D,A
LD A,L
AND X00011111
ADD A,D SUB D
CP X00100000
CCF
PUSH AF
LD B,A
LD A,L
AND X11100000
OR B
LD L,A
POP AF
LD A,C
POP BC
POP DE
RET

```

HL=pixeladres A=pixel masker
B=aantal op te schuiven
pixels. Bij nextxs in positieve
X-richting en bij lstxs in
negatieve X-richting.

```

inatisc PUSH BC
LD B,A
LD A,H
CP #58
CCF
JR C,inats2
CP #58
inats2 LD A,B
POP BC
RET

```

Ingang : HL=attribute-adres.
Uitgang: carry=1 als HL niet
in het attributegedeelte valt.

```

nextx PUSH BC lstx is hetzelfde
LD B,A behalve het
ADD A,L onderstaande
AND X00011111
INC A DEC A
LD C,A
LD A,L
AND X11100000
LD L,A
LD A,C
AND X00011111
OR L
LD L,A
LD A,C
CP X00100000
CCF
LD A,B
POP BC
RET

```

Vindt het adres van de volgende (of vorige voor lstx) attribute. HL bevat het adres bij het ingaan en uitkomen van de routine. Carry=1 als HL links of rechts buiten het scherm komt.

```

inscp PUSH BC
LD B,A
LD A,H
CP #58
CCF
JR C,inscp2
CP #40
inscp2 LD A,B
POP BC
RET

```

Bepaalt of HL in het pixel-
scherm ligt. Zoniet dan volgt
Carry=1.

```

nextxs PUSH BC lstxs is hetzelfde
PUSH DE behalve het
LD D,A onderstaande
LD A,L
AND X00011111
ADD A,B SUB B
RR E
RLA
RL E
RRA
LD C,A
LD A,L
AND X11100000
LD L,A
LD A,C
AND X00011111
OR L
LD L,A
LD A,C
CP X00100000
CCF
LD A,D
POP DE
POP BC
RET

```

HL=attr-adres, B=aantal
attributes verderop. Uitgang :
HL=adres van bedoelde
attribute. De werking is als
nextxs alleen dan op attributes
werkend.

```

nexty PUSH BC lsty is hetzelfde
LD B,A behalve :
LD A,L
SUB X00100000 ADD A,X00100000
LD L,A
LD A,H
SBC A,O
LD H,A
LD A,B
POP BC
CALL inatisc
RET

```

Dit zijn volgende en vorige
attribute in Y-richting.


```

acpa PUSH BC
LD B,A
LD A,23
SUB H
LD H,A
LD A,L
RLCA
RLCA
RLCA
LD L,A
RR H
RRA
RR H
RRA
RR H
RRA
LD L,A
LD A,H
RLCA
RLCA
RLCA
AND #18
OR #40
LD H,A
LD A,B
POP BC
RET
pixeladres van
bovenste pixel-
lijn wordt
aangewezen.

```

```

aaac PUSH BC
LD B,A
LD A,L
AND #1F
LD C,A
LD A,H
AND #03
RLC L
RLA
RLC L
RLA
RLC L
RLA
LD H,A
LD A,23
SUB H
LD H,A
LD L,C
LD A,B
POP BC
RET

```

```

aapc2 PUSH BC
LD B,A
LD A,L
AND #1F
LD C,A
LD A,H
AND #03
RLC L

```

```

RLA
RLC L
RLA
RLC L
RLA
LD H,A
LD A,23
SUB H
RLCA
RLCA
RLCA
OR #07

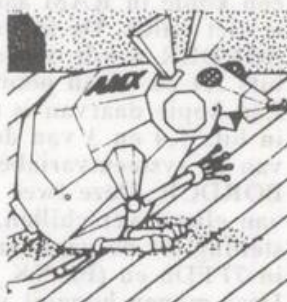
```

```

LD H,A
LD A,C
RLCA
RLCA
RLCA
LD L,A
LD A,B
POP BC
RET

```

als aapc alleen
linkerbovenhoek
wordt nu aangewezen



```

pms1 LD B,8
pms11 LD A,(HL)
LD (DE),A
INC HL
INC D
DJNZ pms11
RET

```

plaatst 8 opeenvolgende
bytes uit het geheugen
aangewezen door HL op
het scherm. De scherm
positie wordt aangewezen
door DE en is de
bovenste regel van
een karakter cel.

```

aapa PUSH BC
LD B,A
LD A,H
AND #03
RLCA
RLCA
RLCA
OR #40
LD H,A
LD A,B
POP BC
RET

```

HL wordt pixeladres
van de bovenste pixel-
lijn van de bijbeho-
rende aangewezen attr.

128 K

georganiseerd geheugen

R. Uittenbogaard

Zoals bekend, heeft de ZX Spectrum 128, 128 K RAM en 32 K ROM. De Z80 microprocessor kan echter maar tot 16 bits adresseren, dit is 64 K. Hoe heeft sir Clive het dan toch voor elkaar gekregen die extra K's in de computer te krijgen?

Sir Clive heeft een slimme techniek toegepast: het geheugen is onderverdeeld in stukjes van ieder 16 K. Deze stukjes noemen we PAGINA's. Er zijn dus $32/16 =$ twee ROM pagina's en $128/16 =$ acht RAM pagina's. De ROM pagina's hebben nummers 0 en 1, de RAM pagina's zijn genummerd van 0 t/m 7. In figuur 1 staan vertikaal de adressen 0 t/m FFFFh, horizontaal de

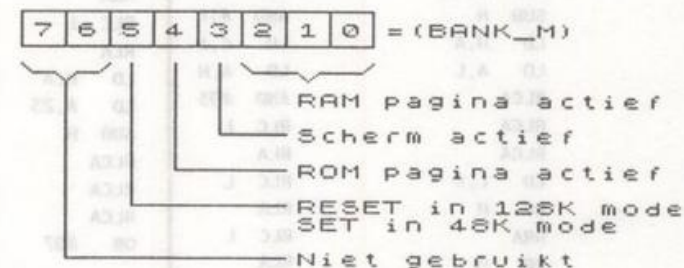
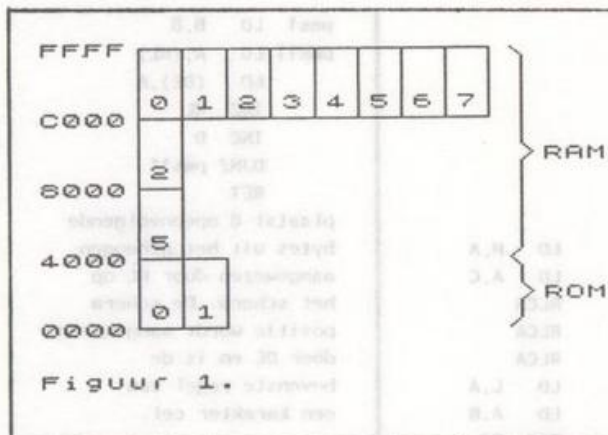
verschillende pagina's. Ieder van die pagina's kan al dan niet te bereiken zijn, dat hangt er van af of de pagina ACTIEF is of niet. Dat een pagina actief is, wil zeggen dat de gebruiker de pagina kan bereiken (kan adresseren).

De twee ROM pagina's bevinden zich op adressen 0 t/m 3FFFh. Maar hoe refereren we naar deze bytes? Stel dat we de CHAN_OPEN routine in de oude ROM bedoelen, dan volstaat 1601h niet. Want 1601h is dubbelzinnig: is dit pagina nul of pagina een? Dit lossen we op door het pagina-nummer voor het adres te zetten. Voor CHAN_OPEN levert dat 11601h (ROM 1 is de "oude" ROM, ROM 0 de "nieuwe"). De twee ROMs hebben dus adresbereik 00000 t/m 03FFF en 10000 t/m 13FFF.

Er zit dus een groot gat tussen, maar daar trekken we ons niets van aan. De eerste digit betekent immers alleen het pagina-nummer.

De acht RAM pagina's hebben adressen van C000h t/m FFFFh. Hier passen we dezelfde techniek toe: als we adres DB00 in pagina zeven bedoelen schrijven we 7DB00h. De RAM beslaat dan adressen 0C000 t/m 0FFFF, 1C000 t/m 1FFFF, ..., 7C000 t/m 7FFFF.

Maar wat bevindt zich dan op adressen 4000h t/m BFFFh? Als PEEK 23610 toch een betekenis moet hebben, dan moet daar ook iets aanwezig zijn. Het antwoord is te vinden in figuur 1: vanaf 4000 t/m 7FFF staat pagina 5, en van 8000 t/m BFFF staat pagina 2. D.w.z. dat



4000 precies dezelfde byte is als 5C000, 5000 is 5D000, 6000 is 5E000, 8000 is 2C000, 9000 is 2D000, enz.

Maar hoe weet je nu welke pagina actief is, of beter: hoe kun je een andere pagina actief maken? Het antwoord is te zoeken in een input/output poort. Poort 7FFDh bevat alle informatie (zie figuur 2). Helaas kun je niets via deze poort invoeren. Maar dat is geen ramp, want de Spectrum heeft een nieuwe systeemvariabele die precies dezelfde informatie bevat. De nieuwe systeemvariabele heet BANK_M en heeft adres 5B5Ch (23388d). Van deze byte bepalen bits 0,1 en 2 het paginanummer van de RAM pagina die actief is en bit 4 die van de actieve ROM pagina. Bit 5 moet altijd geRESET blijven in 128K mode, anders krijg je een complete crash die de 48 K mode inschakelt.

Dit systeem van informatie in een poort waaruit niet gelezen kan worden, maar waarvan wel een kopie in RAM aanwezig is, is niet uniek. De kleur van de BORDER wordt opgeslagen in bits 2,1 en 0 van poort FEh. Een kopie daarvan is te vinden in bits 5,4 en 3 van de inhoud van de systeemvariabele BORDER. Deze twee mogen wel van elkaar verschillen, in tegenstelling tot de paginanummers in 7FFDh en (BANK_M). Deze mogen hooguit verschillen zolang interrupts zijn uitgeschakeld. Als ze verschillen, dan is altijd in beide gevallen de waarde in de I/O poort bepalend. De BORDER kleur is te veranderen m.b.v. OUT 254,kleurnummer in BASIC. Probeer dit echter niet in 7FFDh. Dat lukt niet.

Nu het echte pagina-wisselen. Stel ik wil pagina 7 actief maken. Dan is het niet zomaar een byte uitvoeren naar poort 7FFD, want als je niet uitkijkt loopt het mis. Er zijn een paar dingen waar je rekening mee moet houden.

1) De CPU mag niet bezig zijn met een machinetaalroutine boven adres BFFFh, want die adressen worden na de pagina-wissel verondersteld naar een andere pagina te verwijzen. De CPU zou dan naar het midden van een onbekende routine springen: 99% kans op crash.

2) Er mag voor en na de pagina-wissel niet met de stack gemanipuleerd worden (bijvoorbeeld corresponderende PUSH en POP instructies, maar ook CALL en RET), of de stack moet beneden adres C000 staan.

3) De byte uit BANK_M kan het beste eerst worden opgehaald en veranderd, en daarna twee keer worden teruggezet in zowel BANK_M als poort 7FFD. Doe dat terugzetten in die volgorde. Het Spectrum-systeem is er namelijk niet blij mee als die twee van elkaar verschillen. Als je ze in de verkeerde volgorde uitvoert en er vindt een interrupt plaats tussen de instructies in, dan volgt een crash.

```

10      ORG ????      ;zelf invullen; wel lager dan C000h
20 PAGE_A PUSH BC    ;BC registerpaar op stack bewaren
30      LD C,A        ;paginanummer in C
40      LD A,(BANK_M) ;huidige pagina
50      AND #F8
60      OR C           ;verander pagina-informatie
70      LD BC,#7FFD
80      LD (BANK_M),A ;sla veranderde informatie op
90      OUT (C),A      ;de echte pagina-wissel
100     POP BC         ;haal BC registerpaar terug
110     RET            ;einde
120 BANK_M EQU #5B5C  ;BANK_M op 5B5Ch

```

Figuur 3. Een andere RAM-pagina kan actief worden gemaakt, door het A-register te laden met het pagina-nummer en dan PAGE_A aan te roepen.

Alles bij elkaar leidt dit tot een universele machinecode-routine: PAGE_A. Zie figuur 3. Deze kan eigenlijk in geen enkel voor de Spectrum 128 gemaakt machinecode-programma ontbreken. Er wordt verondersteld dat de stack en de routine beneden C000h liggen. De routine is van commentaar voorzien.

De volgende keer zal ik o.a. bit 3 van (BANK_M) behandelen. U kunt vast thuis de twee commando's POKE 23388,24 en POKE 23388,16 proberen.

SOFTWARE-SERVICE

VERZAMEL CASSETTES

Prijzen voor Cassettes SG-1 t/m SG-6 en SG-8:

Abonnees: f 12,50/250 BF
Andere Lezers: f 17,50/250 BF

SG-01 met programma's uit Sinclair Gids 1+2. met Specblaster/ Breien/Ziektekosten/Opus Ext Catalogue/Explorer ext cat voor mdv/ Grafic.

SG-02 met de algemene programma's uit Sinclair Gids 3+4. met: Darts/ Schaatsen/Laatste 16/48K-Luxe Lister/16K-Luxe Lister/ Lotto- kans/UDG-maker/ Attributen/ One Liners 1 t/m 5/HEX-DEC.

SG-03 met educatieve programma's uit Sinclair Gids 4; Functies/ Tal- stelsels/Rekentest/ 20-veld/ sommen/ klok1/ klok2/ verkeer/ klokkijken.

SG-04 met programma's uit de Sinclair Gids 5+6 met Mdv-cat/ Relatieve vochtigheid/ Grafieken/ Key-in utility/ Shootin' in code/ Beta Basic programma's/ 3D-Solitaire Masterfile 128/ Tasword 128.

SG-05 met programma's uit de Sinclair Gids 7+8+9: Fill-Routine/ Print 64, Specbrot/ Shadescreeen/ One- Liner Creator/ 42-karakters/ NI-List/ Interrupt-routines/ schaatsen/ Draw-Poker.

SG-06 met programma's uit Sinclair Gids 10+11: Reformat/ Tasword- II aanpassingen/ Shadecopy/ Memory-test/ Yatzee/ Patience/ Random-Access File (OPUS).

SG-07 Dubbelpak (2 cassettes) met A+B programma's uit de zomerspecial maar liefst 21 programma's ! Dit Dubbelpak kost:
abonnee's f 25,00/500 BF
andere lezers: f 35,00/700 BF

SG-08 met programma's uit Sinclair Gids 12:
Valuta lijst, Matrix_poker, Karakters, OPUS Disk Utility, VAR58000, FILL, CHAN59000, SERPENT(niet gepubliceerd!)

OPUS SOFTWARE

DEVPAC 4: 3.5" Disk..... f 79,00
Abonnee: f 70,00
Belgische Lezers..... 1580 BF
Belgische Abonnees..... 1400 BF

PASCAL 1.7: 3.5" Disk.... f 125,00
Abonnee: f 115,00
Belgische Lezers..... 2500 BF
Belgische Abonnees..... 2300 BF

VERZAMEL DISKETTES

LET OP !!

OPUS Diskettes Standaard 178 K
Disciple Diskettes 3.5" 80 tracks/DS.

Verzameldiskettes:

OPUS-1 Verzameldiskette 3.5" met de programma's als SG-01

OPUS-2 Verzameldiskette 3.5" met de programma's als SG-02

OPUS-3 Verzameldiskette 3.5" met de programma's als SG-03

OPUS-4 Verzameldiskette 3.5" met de programma's als SG-04

OPUS-5 Verzameldiskette 3.5" met de programma's als SG-05

OPUS-6 Verzameldiskette 3.5" met de programma's als SG-06

OPUS-8 Verzameldiskette 3.5" met de programma's als SG-08

De verzameldiskettes OPUS-1 t/m 6 & 8
Abonnees: f 12,50/250 BF
Andere Lezers: f 17,50/350 BF

OPUS-7 Dubbelpak 2 diskettes 3.5" Verzameldiskettes met de programma's als SG-07 A+B.
Dit dubbelpak kost:
Abonnees: f 25,00/500 BF
Andere Lezers: f 35,00/700 BF

DISCIPLE SOFTWARE

DEVPAC 4.1: 3.5" (80 Tracks/DS)
Normaal: f 79,00
Abonnee: f 70,00
Belgische Lezers..... 1580 BF
Belgische Abonnees..... 1400 BF

PASCAL 1.7: 3.5" (80 Tracks/DS)
Normaal: f 125,00
Abonnee: f 115,00
Belgische Lezers..... 2500 BF
Belgische Abonnees..... 2300 BF

DISCIPLE-1
Verzameldiskette met programma's als SG-01. Minus Opus programma's

DISCIPLE-2
Verzameldiskette met programma's als SG-02. Minus Opus programma's

DISCIPLE-3
Verzameldiskette met programma's als SG-03. Minus Opus programma's

DISCIPLE-4
Verzameldiskette met programma's als SG-04. Minus Opus programma's

DISCIPLE-5
Verzameldiskette met programma's als SG-05. Minus Opus programma's

DISCIPLE-6
Verzameldiskette met programma's als SG-06. Minus Opus programma's

DISCIPLE-8
Verzameldiskette met programma's als SG-08 minus OPUS programma's

De verzameldiskettes Disciple 1 t/m 6 en Disciple 8:
Abonnees: f 12,50/250 BF
Andere Lezers: f 17,50/350 BF

DISCIPLE-7
Verzameldiskette met de programma's als SG-07 A+B.
Abonnees: f 25,00/500 BF
Andere Lezers: f 35,00/700 BF

QL-SOFTWARE

QS-01 Hisoft DEVPAC QL
voor..... f 125,00/2500 BF
> Abonnee: f 110,00/2200 BF

QS-02 Metacomco QL-C
voor..... f 379,00/7580 BF
> Abonnee: f 340,00/6800 BF

QS-03 Metacomco QL Pascal
voor..... f 349,00/6980 BF
> Abonnee: f 315,00/6300 BF

QL-Disk-01

Verzameldiskette 3.5" DSDD 80 track met QL-programma's uit de diverse Sinclair Gidsen. Met o.a.: Qlogo, QBlast, Sprite Designer, Sprite Toolkit, Print_Spooler, Vadertje Tijd, Q_Erny, Space Simulator.

Abonnee (3.5" Flp) f 17,50/350 BF
Andere Lezers: (Flp) f 25,00/500 BF

SPECIALS

SS-01 Special 1: TT-S Toolkit 48K voor programmeurs incl. Ned. Handleiding. De GAMMA-TOOLKIT met 10 extra commando's: TRACE, FIND, RENUMBER, MOVE, etc. verder een UDG-ontwerper, een RAH-test-programma, SCREEN tekenprogramma en TAPE-INVESTIGATOR. 5 Utilities voor:
.....fl.27,50/550 BF
> Abonnee: fl.22,50/450 BF

SS-03 Special 3: Utilities 1 48K COPY-A4, COPY RS232. Screencopy voor Centronics en RS232.
4 utilit. voor fl.15,00/300 BF
> Abonnee: fl.10,00/200 BF

SS-04 Hisoft DEVPAC 4. cassette Voor 48/128/128+2
voor.....fl.59,00/1180BF
> Abonnee: fl.53,00/1160BF

SS-05 Hisoft BASIC Compiler De meest complete compiler voor.....fl.95,00/1900BF
> Abonnee: fl.85,00/1700BF

SS-06 Hisoft PASCAL 1.7
voor.....fl.95,00/1900BF
> Abonnee: fl.85,00/1700BF

SS-07 Gilsoft PAMS Prof. Adventure Writing System. TOPPER !!
voor.....fl.89,00/1780BF
> Abonnee: fl.80,00/1600BF

SS-08 Hisoft 'C' voor Spectrum 48K
voor.....fl.95,00/1900BF
> Abonnee: fl.85,00/1700BF

BOEKEN QL

The Sinclair QL serie van Hutchinson. Originele prijs fl. 39,00 per stuk

QL-B1: Introduction to Superbasic on the Sinclair QL nu: fl. 15,00/300 BF

TASWORD TWEE

Nog steeds DE tekstverwerker voor wie niet met microdrive of disk-drive werkt. Geeft 64 karakters per regel, ook op het scherm. Tal van opties van duurdere computers behoren met Tasword-2 ook op de Spectrum tot de mogelijkheden: woord zoeken en vervangen, alinea's verplaatsen, automatische woordomschlag, invoegen en uitlijnen, etc., etc. Met Nederlandse handleiding:
Prijs: Cassette f 59,00
Belgische Lezers.....1180 BF

TASWORD +3

Programma gelijk aan de TASWORD 128, maar nu aangepast aan de specificaties van de Spectrum +3. Programma wordt geleverd op 3" Disk.
Prijs: (Engels) 3" Disk..... f 89,00
Belgische Lezers.....1780 BF

TASWORD DRIE

De definitieve tekstverwerker voor Spectrum in combinatie met een of meerdere microdrives of de OPUS Discovery. De geheel Nederlandse versie heeft een duidelijke handleiding van 69 pagina's A4.
Prijs: Microdrive f 89,00
Belgische Lezers..... 1780 BF

TASWORD 128

Is feitelijk Tasword-3 voor de Spectrum 128K. Met alle eigenschappen die dat programma zo gewaardeerd maken, met als extra de mogelijkheid om te werken met tekstbestanden die tot 64K groot kunnen zijn.
Prijs: Cassette Engels..... f 69,00
Belgische Lezers.....1380 BF

TASWORD +2

Programma gelijk aan de TASWORD 128, maar nu voor de Spectrum 128 +2.
Prijs: Cassette (Engels)..... f 69,00
Belgische Lezers.....1380 BF

TASPRINT

Geeft extra (grote en bijzondere) letters aan elke Dot-Matrix-printer (ook: Smith Corona Fastext 80). Programma is afzonderlijk, maar ook in Tasword 2 te gebruiken.
Prijs: (Engels) Cassette..... f 45,00
Belgische Lezers..... 900 BF

TASCOPY

Geeft een afdruk van het beeldscherm op een 'grote' printer. Grootte van de afdruk: A4 (met grijs tinten) of zelfs tot Poster-formaat. Erg mooi. Geschikt voor een hele gamma van printers.
Prijs: (Engels) Cassette..... f 45,00
Belgische Lezers..... 900 BF

TASDIARY voor 48K Spectrum

Deze elektronische agenda houdt al Uw afspraken bij met alle voordelen die met computerverwerking van gegevens zijn verbonden. U kunt Tasdiary ook prima gebruiken om een dagboek in bij te houden.
Prijs: (Engels) Cassette..... f 45,00
Belgische Lezers..... 900 BF

OPUS SOFTWARE

Tasword 128
3.5" Disk.....(Engels) f 79,00
Belgische Lezers..... 1580 BF

Tasword 3
3.5" Disk.....(Nederlands) f 79,00
Belgische Lezers..... 1580 BF

Alle prijzen incl. BTW en verzendkosten voor alle abonnees. Bent U geen abonnee, dan betaalt U per bestelling fl. 2,50 of 50 BF administratiekosten extra !!!

AANBIEDINGEN gelden zolang de voorraad strekt.

Bestellen door overmaking (of betaalcheque/girobetaalkaart) op giro 5109074 t.n.v. de SINCLAIR GIDS, Baarle Nassau, België. Postrek: 000-1592677-34 tnv. Sinclair Gids, Baarle Nassau.

Betalen bij ontvangst kan ook, maar dan worden de rembourskosten (fl.8,75/165 BF) extra in rekening gebracht.

VERHELD BIJ ALLE BESTELLINGEN DE NUMMERS VAN DE CASSETTES WAAR HET OM GAAT !!

Verzending van bestelde producten geschiedt vanaf heden één keer per week.

De basis van BASIC - deel 4

R.-J. Donkers

Zoals ik in deel 3 reeds voor-
zichtig aanroerde, was het mij
door drukke werkzaamheden toen
niet mogelijk verder aan het
programma te bouwen. Dat het
echter zo druk zou zijn dat ik
zelfs een keer moest overslaan,
d ik helaas ook niet voor-
zien. Mijn excuses aan allen
die met smart uitkeken naar het
vervolg.

Een geluk (of moet ik spreken
van ongeluk?) was dat de Sin-
clair Gids enige vertraging
opliep, zodat ik deze keer wat
ruimer in de tijd kwam te zit-
ten. Dat geeft mij de gelegen-
heid om wat extra te schrijven.

In deel 3 van deze reeks had ik
het al genoemd: zelf sleutelen
aan programma's. Dat is het
mooiste wat er is en juist de
beginnende programmeur zal dat
nog niet helemaal inzien, maar
dra ook hij of zij beginner
af is en de faalangst overwon-
nen heeft, zal hij/zij ook zelf
gaan sleutelen aan programma's.
Je kunt op vele manieren sleu-
telen aan je eigen programma's;
zoals het zo kort mogelijk of
de werking sneller maken etc.,
etc. Ook bestaande commerciële
programma's aanpassen aan je
eigen ideeën en wensen is een
hele uitdaging. Uiteraard moet
je dan al de nodige ervaring
hebben en bovendien het te
bewerken programma goed kennen.
Een mooi voorbeeld van zo'n
aanpassing stond in nummer 10,
waarin Drs. A.J. Schoneveld een
hele reeks aanpassingen presen-
teerde voor de tekstverwerker
Tasword II. Hij gebruikte hier-
voor niet het Spectrum BASIC,

waar wij ons nu mee bezig hou-
den, maar het alom geprezen
Beta BASIC. Deze BASIC is een
sterke uitbreiding op het
Spectrum BASIC. Later als we de
beginfase achter ons hebben,
zullen velen zeer zeker ook
Beta BASIC gaan gebruiken,
omdat de mogelijkheden van de
Spectrum daarmee aanzienlijk
kunnen worden uitgebreid.

Maar nu eerst even terugkomen
op het sleutelen. Ook aan ons
programma in wording kan al
gesleuteld worden. Daarom zal
ik aan het eind van dit deel
een aantal voorbeelden geven,
die met dit 'sleutelen' te
maken hebben.

VERDER INVULLEN VAN PROGRAMMA

Waar waren we ook alweer blij-
ven steken? We hebben het ruwe
raamwerk opgezet en hebben de
invoer-routine en de opslag- en
inlees-routine ingevuld, zodat
we al een beetje met het pro-
gramma konden werken. Ik hoop
ook dat iedereen dat ook gedaan
heeft en inmiddels een adres-
sen-bestandje heeft opgebouwd.
Daarmee kunnen we dan aan het
werk en de nieuwe routines
uitproberen.

HET OVERZICHT.

Als we willen weten wie er al-
lemaal in ons bestand voorko-
men, dan is het handig dat we
de complete lijst kunnen bekij-
ken of kunnen afdrukken op
papier.

Deze routine start op regel
4000 en eindigt op regel 4999.

4000 REM overzicht
4005 PAPER 5: INK 1: BORDER 5: C
LS

Regel 4005 zet het scherm en
border in een bepaalde kleur en
gebruikt donkerblauw voor de
tekst (INK) en maakt het scherm
schoon voor onze routine.

Allereerst gaan we controleren
of er een bestand in het geheu-
gen aanwezig is. Is dat niet 't
geval dan moeten we een waar-
schuwing krijgen en wel zodanig
dat we ook te zien krijgen wat
we dan eerst moeten doen. We
moeten namelijk eerst een of
ander bestand inlezen van tape
of er een aanmaken. Daarom
sprak ik in het voorafgaande de
hoop uit dat een ieder die deze
serie volgt inmiddels aan het
werk is geweest met dit pro-
gramma en al een bestand heeft
aangemaakt. Als je dat nou wel
gedaan hebt, dan pluk je nu de
vruchten van je arbeid.

Om te controleren of er al een
bestand in het geheugen aanwe-
zig is, kunnen we eenvoudig
zien aan de waarde van de
'controle'. Is controle=0 dan
zit er geen enkel adres in het
geheugen. Is controle groter
dan 0 (>0) dan zit er zeker een
bestand van minimaal 1 adres in
het geheugen.

In BASIC vertaald ziet boven-
staand verhaal er als volgt
uit:

```
4010 IF controle>0 THEN GO TO 40  
50  
4020 PRINT AT 5,1;"Geen bestand  
aanwezig!"; PRINT AT 7,1;"Eerst  
bestand aanmaken of inlezen": P  
RINT AT 18,1;"TOETS [ENTER] VOOR  
HOOFDMENU"
```



```

4030 LET QS=INKEY$: IF QS="" THE
N GO TO 4030
4040 IF CODE QS<>13 THEN GO TO 4
030
4045 RETURN

```

Even in 't kort deze regels verklaren mocht het niet direct duidelijk zijn.

In regel 4010 controleren we of 'controle' groter dan 0 is. Als dat zo is, dan gaat het programma verder op regel 4050. Zouden we namelijk controleren of controle=0 dan krijgen we problemen met de boodschap die we in ons programma willen opnemen voor de gebruiker. We raken dan in een wirwar van IF.....THEN..... opdrachten verzeild.

Nu sturen we eenvoudig het programmaverloop naar regel 4050 als er inderdaad al een bestand aanwezig is. Is dat niet het geval dan gaat het programma verder op regel 4020 waar we dan onze boodschap kunnen neerzetten.

Regel 4020 zet de boodschap op het scherm en 4030 t/m 4045 zorgen ervoor dat U terugkeert naar het hoofdmenu als U de boodschap heeft gelezen en [ENTER] indrukt.

Op regel 4050 begint dan de 'echte' overzicht-routine.

```

4050 PRINT PAPER 1; INK 7; AT 0,0
; " OVERZICHT
4060 PRINT AT 5,1;"WILT U HET OV
ERZICHT:"," NAAR HET [S]CHERM OF
NAAR DE [P]RINTER"
4070 PRINT AT 18,1;"MAAK UW KEUZ
E [S] OF [P]"
4075 LET QS=INKEY$: IF QS="" THE
N GO TO 4075
4080 IF QS="S" OR QS="s" THEN GO
TO 4100
4090 IF QS="P" OR QS="p" THEN GO
TO 4500
4095 GO TO 4075
4100 CLS : PRINT PAPER 1; INK 7;
AT 0,0;" OVERZICHT
4110 PRINT AT 5,1;"NAAM : "
4115 PRINT AT 6,1;"ADRES : "
4120 PRINT AT 7,1;"POSTCODE : "

```

```

4125 PRINT AT 8,1;"PLAATS : "
4130 PRINT AT 10,1;"GEB.DATUM:"
4140 FOR N=1 TO CONTROLE
4150 PRINT AT 5,12;N$(N)
4151 PRINT AT 6,12;A$(N)
4152 PRINT AT 7,12;C$(N)
4153 PRINT AT 8,12;P$(N)
4154 PRINT AT 10,12;V$(N)
4160 IF N<CONTROLE THEN PRINT AT
18,1;"TOETS [ENTER] VOOR VOLGEN
DE"
4170 LET QS=INKEY$: IF QS="" THE
N GO TO 4170
4175 IF CODE QS<>13 THEN GO TO 4
170
4180 NEXT N
4190 PRINT AT 18,1;" EINDE VAN H
ET BESTAND : " AT 20,1;"TOET
S [ENTER] VOOR HOOFDMENU"
4195 LET QS=INKEY$: IF QS="" THE
N GO TO 4195
4200 IF CODE QS<>13 THEN GO TO 4
195
4210 RETURN

```

Eerst weer eens even bekijken wat al die programmaregels doen en waarom.

4050 spreekt voor zichzelf dacht ik zo. In regel 4060 vragen we de gebruiker of hij/zij het overzicht op het scherm of op papier wil hebben. De daarop volgende regels controleren dan weer welke keuze we hebben gemaakt. Het programma-verloop wordt dan naar regel 4100 of regel 4500 gestuurd.

In regel 4100-4130 drukken we dan een kopregel af en de hoofdtaksten.

De regels 4140-4210 omvatten de routine om achter de hoofdtaksten de daarbij horende DATA af te drukken. Het eigenlijke afdrukken vindt plaats op regels 4150-4154. Daarna zetten we op het scherm de boodschap: "druk op [enter] voor de volgende". Drukt U op ENTER dan wordt het volgende record uit het bestand afgedrukt net zo vaak als er records aanwezig zijn. Als het bestand in z'n geheel op het scherm afgedrukt is, dan krijgen we de boodschap: "Einde bestand !". Daarna komen we met een druk op [Enter] weer in het hoofdmenu. We hebben dan onze

keuze ("overzicht") afgewerkt en via het hoofdmenu kunnen we weer verder met ons programma.

Nu deze routine verder afmaken met het gedeelte dat alles naar de printer stuurt.

Nu is het voor mij moeilijk om te bepalen hoe we de gegevens naar de printer zullen sturen.

Sommigen hebben de beschikking over een 80-koloms printer anderen hebben een ZX Printer of een Seikosha GP50 of Timex 2040 printer. Ik geef daarom een print-routine die in ieder geval op alle printers werkt en die in de gevallen van een 80-koloms printer wat extra papier kost (helaas).

De printroutine is een kopie van de routine die alles op het scherm afdrukt met dit verschil dat er geen onderbreking plaats vindt met een druk op [enter]. Wel drukken we tussen elk adres twee lege regels af, die eventueel te vervangen zijn door 1 regel met streepjes (-).

```

4500 REM AFDrukKEN OP PAPIER
4510 BORDER 4: PAPER 4: INK 0: C
LS
4520 PRINT "ZET UW PRINTER GERE
D EN TOETS [ENTER]"
4525 LET QS=INKEY$: IF QS="" THE
N GO TO 4525
4530 IF CODE QS<>13 THEN GO TO 4
525
4540 FOR N=1 TO CONTROLE
4550 LPRINT N$(N)
4551 LPRINT A$(N)
4552 LPRINT C$(N); " ";P$(N)
4553 LPRINT
4554 LPRINT V$(N)
4555 LPRINT
4556 LPRINT "-----"
4557 LPRINT
4560 NEXT N
4570 RETURN

```

Nu we toch aan het printen zijn kunnen we eigenlijk ook best verdergaan met het afdrukken van labels.

Er zit een 'maar' aan deze routine: de kleine printers zoals GP50/Timex 2040 zijn niet

geschikt voor het verwerken van etiketten op kettingbanen. De oplossing voor die gebruikers is: alles afdrukken zoals bij het 'overzicht' op één lange strook en dan met de schaar de adressen losknippen. Daarna de adresstroken met de Pritt-stift op de enveloppen plakken.

Voor hen die de beschikking hebben over een printer die wel kettingformulieren kan verwerken zullen we de routine maken, hoewel de andere gebruikers er ook dankbaar gebruik van kunnen maken, omdat we in deze routine wat selectie-mogelijkheden zullen inbouwen. Zodat iedereen selectief kan afdrukken.

De routine 'labels' start op regel 7000 en eindigt op 7999.

```
7000 REM Labels
7010 BORDER 1: PAPER 7: INK 1: C
LS
7020 PRINT PAPER 1: INK 7: AT 0,0
: " LABELS AFDRUKKEN
"
7030 PRINT AT 5,10: "KEUZEMENU"
7031 PRINT AT 6,10: "=====
7050 PRINT AT 8,1: "[1]... ALLES
AFDRUKKEN"
7060 PRINT AT 9,1: "[2]... BEPAAL
DE NAMEN AFDRUKKEN"
7070 PRINT AT 11,1: "[3]... BEPAA
LDE PLAATS AFDRUKKEN"
7080 PRINT AT 13,1: "[4]... BEPAA
LDE DATUM AFDRUKKEN"
7090 PRINT AT 15,1: "[5]... TERUG
NAAR HOOFDMENU"
7100 PRINT AT 19,5: "BEPAA UW KE
UZE: [1]... [5]"
7105 LET QS=INKEY$: IF QS="" THE
N GO TO 7105
7110 IF QS<"1" OR QS>"5" THEN GO
TO 7105
7115 LET Q=VAL QS
7120 GO TO 7200*(Q=1)+7400*(Q=2)
+7600*(Q=3)+7800*(Q=4)+7999*(Q=5
)
7125 GO TO 7000
7200 REM ALLES AFDRUKKEN
```

Deze routine kunnen we letterlijk overnemen van onze overzicht-routine. Alleen moeten we wel even bepalen hoeveel regels er op ons etiket kunnen. Bij een standaard afdruk worden er zes regels

afgedrukt. Het is dus zaak het etiket op te meten en zelf voor het aantal te printen regels (incl. de lege regels) te zorgen. Bovendien moeten we ook nog rekening houden met een extra lege regel, om de ruimte tussen twee etiketten te overbruggen.

Een regel is gelijk aan $25.4/6 = 4.23$ mm. Deel nu de hoogtemaat van het etiket door 4.23 en U heeft het aantal regels, tel daarbij 1 op en U heeft het aantal 'print -regels' van onze routine. Stel dat we op totaal 9 regels uitkomen, dan kunnen we de printroutine afmaken.

```
7210 CLS : PRINT PAPER 1: INK 7:
AT 0,0: " ALLES AFDRUKKEN
"
7220 PRINT AT 5,1: "TOETS [ENTER]
ALS PRINTER GEREED IS."
7230 LET QS=INKEY$: IF QS="" THE
N GO TO 7230
7240 IF CODE QS<>13 THEN GO TO 7
230
7250 FOR N=1 TO CONTROLE
7260 LPRINT : LPRINT
7270 LPRINT NS(N)
7280 LPRINT
7290 LPRINT AS(N)
7300 LPRINT
7310 LPRINT CS(N): " "; PS(N)
7320 LPRINT : LPRINT
7330 NEXT N
7340 GO TO 7000
```

Bij het printen op kettingbanen er wel op letten dat de printkop van de printer gelijk staat met de eerste regel van het etiket.

Voorwaardelijk afdrukken.

In onze volgende onderdelen van deze label-routine, gaan we alleen bepaalde adressen afdrukken. Met andere woorden: voorwaardelijk afdrukken. In de eerste routine willen we dat alleen bepaalde namen worden afgedrukt.

```
7400 REM BEPAALE NAMEN AFDRUKKE
N
```

De andere routines drukken dan adressen af met als voorwaarde dat er een bepaalde plaats of een bepaalde verjaardag moet voldoen aan onze vraag.

```
7600 REM BEPAALE PLAATSEN AFDRU
KKEN
```

```
7800 REM BEPAALE VERJAARDAGEN
```

SLEUTELN

Zoals gezegd is sleutelen aan programma's een nuttig en aangenaam tijdverdrijf. In de eerste plaats kunnen we een aantal zaken vereenvoudigen en gebruikersvriendelijker maken.

Zo moeten we in ons programma een postcode en een plaatsnaam invoeren. Zo'n postcode bestaat in Nederland uit 4 cijfers een spatie en twee letters. Die letters nu moeten in hoofdletters geschreven worden.

Door nu handig gebruik te maken van systeemvariabele FLAGGS2 op adres 23658, kunnen we ervoor zorgen dat de invoer direct in hoofdletters wordt gezet.

Wat doet systeemvariabele FLAGGS2 nu precies?

Raadpleeg ook het handboek van Uw spectrum en dan vooral het hoofdstuk systeemvariabelen.

Om dit uit te leggen, moeten we ons eerst eens afvragen: Wat is een systeemvariabele?

Het woord zegt het eigenlijk zelf al. Het is een variabele bestaande uit een of meer bytes die in het geheugen van de computer worden gezet bij het aanzetten van de machine. De BASIC van onze spectrum (die in ROM zit) gebruikt die systeem-variabelen als een soort geheugensteuntjes. Omdat de waarde van die variabelen steeds kan veranderen zitten ze niet in ROM maar in RAM. Daarom heten ze ook variabelen.

En omdat de systeemsoftware (in ROM) van onze spectrum ze gebruikt, noemen we ze aldus systeemvariabelen.

Een aantal van die systeemvariabelen kunnen we manipuleren.

Als U het handboek ernaast hebt liggen dan ziet U voor de systeemvariabelen nog een kolom met een getal en soms voorafgegaan door een 'X' of een 'N'. Bij de 128K machines kan daar ook nog een 'R' staan.

Het getal duidt het aantal bytes aan dat de variabele omvat. De 'N' betekent dat als we de waarde veranderen met POKE, deze verandering slechts tijdelijk is omdat de systeemsoftware deze tijdens de werking van een programma voortdurend aanpast. Een 'X' betekent dat we de waarde(n) van de betreffende variabele beter niet kunnen veranderen, omdat bij onoordeelkundig POKEn ons speccie kan 'crashen'. Bij de 128K betekent een 'R' dat dit een machinetaal-routine is. Die wordt aangeroepen door het systeem. Machinetaalprogramma's zouden hierin wel het nodige kunnen veranderen, zoals bijv. systeem-routine POUT op adres 23348-23369. Maar beginnende programmeurs kunnen het beste die variabelen waar een 'X', 'N' of 'R' voor staat, laten voor wat ze zijn. Bij de meeste is een grondige kennis van de werking van de Spectrum nodig. Beperken we ons liever tot die systeemvariabelen waar we wel wat mee kunnen doen.

Zoals misschien al wel duidelijk is geworden uit het voorgaande, kunnen we de bytes behorende bij een systeemvariabele veranderen door er een andere waarde in te POKEn. Het is geen Nederlands, maar maakt in ieder geval duidelijk wat we bedoelen.

Zo zien we dat FLAGGS2 één byte omvat op adres 23658.

Deze 'Vlag' staat op '0' als we met kleine letters werken en heeft de waarde '8' als we met hoofdletters werken. Als we de waarde van FLAGGS2 nu zo

veranderen, dat we op het juiste moment hoofdletters krijgen, besparen we ons de toetsaanslag van 'CAPS SHIFT'. We doen dat door POKE 23658,8 in een programmaregel op te nemen voordat we de hoofdletters nodig hebben en daarna zetten we alles weer in normale toestand met POKE 23658,0.

Deze uitleg bevat in principe een fout, omdat deze variabele niet altijd nul hoeft te zijn om met kleine letters te kunnen werken. Voor de echte delvers onder ons de preciese uitleg. Om hoofdletters te krijgen moeten we eigenlijk gebruiken:

POKE 23658,(PEEK 23658+8)

We moeten bij die waarde van de variabele er eigenlijk 8 bijtellen. Maar omdat de waarde in de meeste gevallen 0 is, kunnen we ook best volstaan met de eerste methode. Bovendien kunnen we niet onbepaald er steeds 8 bijtellen omdat we dan snel het maximum van 255 zullen overschrijden. Als we systeemvariabele 23659 nergens gebruiken kan het geen kwaad om onze eerste methode toe te passen. Dat zullen we dan ook maar doen om het programma niet al te gecompliceerd te maken.

In ons geval willen we hebben dat de postcode en voor de duidelijkheid ook de plaatsnaam in hoofdletters worden ingetikt.

De invoerroutine van ons programma begint op regel 1000. In regel 1155 wordt de postcode gevraagd. Nu voegen we de volgende in vóór regel 1155:

1152 POKE 23658,8

We willen ook de plaatsnaam in hoofdletters, dus na de invoer van de plaatsnaam met bijbehorende controles zetten we de systeemvariabele weer in 'nor-

male' toestand.

1240 POKE 23658,0

Dat is dat, een lang verhaal om duidelijk te maken wat we met die twee tussengevoegde regels wilden waarmaken.

Behalve deze systeemvariabele zijn er nog meer die in programma's toepasbaar zijn. Voorbeelden:

PIP op adres 23609

Hiermee kunnen we de klik van het toetsenbord beïnvloeden, maken we die bijv. 25 ipv. 0 dan duurt een toetsklik langer. Maken we die erg kort, dan gaat de cursor zo snel, dat het niet echt meer bruikbaar is. Alleen als we een erg lange programmaregel willen 'editten', is het raadzaam die toetsklik zo kort mogelijk te houden omdat de cursor anders zo traag wordt, dat dit ook weer vervelend wordt.

BORDCR op adres 23624 bepaalt de kleur van de BORDER. Door een kleur te POKEn, kunnen we het uitzicht van de border veranderen. Wel de waarde van de kleur met 8 vermenigvuldigen. Dus rood wordt dus: $8 \times 2 = 16$:

POKE 23624,16.

Voor een volledig overzicht verwijs ik graag naar Sinclair Gidsen nummer 1,2,3 waarin de systeemvariabelen volledig met voorbeelden worden behandeld.

OPMERKINGEN

Verdere tips voor veranderingen in het programma. Nu ik de serie ook zelf nog eens heb nagelezen, ontdekte ik een paar 'stomme' oplossingen voor enkele controles in de invoerroutine. Zo zie je maar weer dat haastige spoed zelden goed is.

Van de andere kant gelukkig dat ik de 'fout' ontdekte, want nu kan ik uitleggen waarom de gekozen oplossing nou niet de

'slimste' is.

Kijken we even terug naar nummer 10 deel 3 van deze serie.

In regel 1065 controleren we of het ingevoerde niet langer is dan de toegestane lengte 'L'. Nu heb ik in regels 1065-1075 maar liefst 4 IF..THEN structuren toegepast, wat enorm vertragend werkt op de loop van het programma, omdat al die controles door de computer doorlopen moeten worden.

Al redenerend ging ik ervan uit dat we moesten controleren of de lengte L niet werd overschreden maar als we nu eens gaan kijken of het ingevoerde kleiner of gelijk is aan de lengte 'L', kunnen we voorkomen dat bij correcte invoer alle IF..THENs doorlopen moeten worden. Dit zal ons zeker een snelheidswinst opleveren.

1065 en volgende worden dan:

```
1065 IF lengte<=L THEN GO TO 1080
1070 PRINT #0;"TE LANG ! TOETS [ENTER]"
1072 LET QS=INKEY$: IF QS="" THEN GO TO 1072
1075 IF CODE QS<>13 THEN GO TO 1072
1077 PRINT #0;"
": GO TO 1050
```

't kort de werking zoals het nu is:

Voldoet de lengte van het ingevoerde dan wordt direct naar de regel 1080 gesprongen, waarin het ingevoerde wordt afgedrukt. Merk het verschil op met onze eerste versie uit deel 3 van deze serie.

Voor de andere invoerroutines geldt dit uiteraard ook. Probeer eerst zelf eens met bovenstaand voorbeeld voor ogen om de andere invoercontroles te veranderen. Kijk daarna pas naar de hierna volgende programma-regels.

```
1115 IF Lengte<=LA THEN GO TO 1130
1120 PRINT #0;"TE LANG ! TOETS [ENTER]"
1122 LET QS=INKEY$: IF QS="" THEN GO TO 1122
1125 IF CODE QS<>13 THEN GO TO 1122
1127 PRINT #0;"
": GO TO 1100
```

```
1165 IF Lengte<=LC THEN GO TO 1180
1170 PRINT #0;"TE LANG ! TOETS [ENTER]"
1172 LET QS=INKEY$: IF QS="" THEN GO TO 1172
1175 IF CODE QS<>13 THEN GO TO 1172
1177 PRINT #0;"
": GO TO 1150
```

```
1215 IF lengte<=LP THEN GO TO 1230
1220 PRINT #0;"TE LANG ! TOETS [ENTER]"
1222 LET QS=INKEY$: IF QS="" THEN GO TO 1222
```

```
1225 IF CODE QS<>13 THEN GO TO 1222
1227 PRINT #0;"
": GO TO 1200
```

```
1265 IF lengte>=LV THEN GO TO 1280
1270 PRINT #0;"TE LANG ! TOETS [ENTER]"
1272 LET QS=INKEY$: IF QS="" THEN GO TO 1272
1275 IF CODE QS<>13 THEN GO TO 1272
1277 PRINT #0;"
": GO TO 1250
```

Tot zover het gesleutel aan ons 'eigen' programma.

De volgende keer zal ik het programma verder bewerken, om de mogelijkheid te scheppen later meer adressen met het programma te kunnen verwerken.

INDIVIDUAL SOFTWARE

Das neue ISO-ROM für den SPECTRUM mit :

- *Reset ohne Programmverlust * Zeichensatz nach Wunsch * Verbessertes CAT für MD
- *Verkürzte Syntax für MD/Opus oder Beta-Disk * Erweiterter Editor mit Cursor rauf und runter in Programmzeilen * Variable Zeichenbreite: 32/36/42/51 oder 64 Zeichen/Zeile

Wählen Sie unter drei Ausführungen:

MONITOR-ROM: Eingebauter HEX-Monitor ermöglicht die Unterbrechung jedes Programms sowie Eingabe und Austesten eigener MC-Routinen **nur 70,-DM**

Backup-ROM: überträgt sämtliche Programme auf Ihr Speichermedium. Dabei werden nur die belegten Bytes abgesaved. Die übertragenen Programme laufen auch ohne ISO-ROM. Mit der **POKE-Option** können Sie jederzeit Ihr Spiel unterbrechen, einen POKE eingeben und weiterspielen. Backup-ROMs gibt es für Microdrive/OPUS-Disk, Beta-Disk und Cassette !! **nur 80,-DM**

Toolkit-ROM: Mit erweitertem BASIC. Unter anderem: RENUMBER (mit GOTO.....) AUTO-Zeilennummern, Zeilen-DELETE und ON ERROR GOTO **nur 70,-DM**

- **Bis zu 4 ISO-ROMs mit absturzsicherer Umschaltung in einem Eprom möglich **
- **Ausführliche Einbauanleitung ** Einbau durch uns nur 25,- DM **

SYS - Das vollständige MC-Entwicklungspaket für den Spectrum. Löst alle MC-Probleme: Superschneller **Macroassembler** mit extrem platzsparendem Textformat, **Disassembler**, **Debugger** mit Single-Step und Trace und **Re-assembler**! Übertragbar auf alle Speichermedien !!! **nur 55,-DM**

MTOS-Multitasking für den Spectrum. Es laufen bis zu 62 MC-Programme und ein Basicprogramm gleichzeitig. Beispielprogramme: Sprite-Controller, Uhr und Printer-Spooler. Für alle Speichermedien. Mit 2 Handbüchern. **nur 50,-DM**

AMADEUS - Das SPECTRUM-Musiksystem für alle 3-Kanal-Soundmodule. Noten-Teil für bequeme Noteneingabe und Korrektur. Synthesizer-Teil mit 7 Instrumenten, 8 Rhythmen, eingebauten Soundeffekten **45,-DM (mit Soundmodul 125,-)**

TRI-STEP: Schrittmotorsteuerung endlich einfach, kompakt und preiswert !!! Unser neues Steuerinterface für 3 Schrittmotoren und 1 Schaltausgang. Für Fahrtrisch-, Plotter-, Robotersteuerung usw. **Kein Extra-Netzteil** erforderlich. Anschlußfertig inkl. Software und drei hochwertigen Schrittmotoren **nur 150,-DM**

ISO-CNC-MASCHINE (Hitzedrahtschneidegerät und Plotter) Sehr stabile Maschine, die in Schulen eingesetzt wird, um CAD-CAM zu demonstrieren. Ansteuerbar über **TRI-STEP** oder andere Computer mit Stepperinterface **nur 390,-DM**

VISION - IBM hat es, Atari hat es, wir haben ein DESKTOP für den SPECTRUM mit Beta-Disk 3.0 + 4.xx. Ohne Laden, da Software im Controllereprom! **nur 50,-DM**
** Fragen Sie nach unserem Info für **Beta-Disk-User**

Alle Programme auch mit englischer Anleitung
Porto/ Verpackung pro Bestellung 5,00 DM

Gratis-Info und Bestellungen mit Scheck bei:
V.Marohn, Am Bellstück 30, 46 Dortmund 50
☎ 0231/715555 ☎ 0231/123109 v. 17"-20"

Uitzonderlijk ... Exceptions op de QL

W. Dijkgraaf

Eén van de redenen waarom het zo eenvoudig is, machinetaal op de QL te schrijven, is dat het operating system (QDOS) bijna alle routines bevat die een programmeur zich maar kan wensen. De TRAP-instructie, waarmee de aanroep van een QDOS-routine plaats vindt, is dan ook een geliefde instructie voor veel QL-programmeurs. Echter, hoe de TRAP-routine precies door de processor wordt uitgevoerd, is voor velen een raadsel en de prachtige mogelijkheid van de 68000 om zelf exception-routines te schrijven, wordt door weinig mensen benut, terwijl dit eigenlijk erg eenvoudig is.

Exceptions (= uitzonderingen) worden gebruikt om na de uitvoering van een instructie, het programma ergens anders verder te laten gaan en na afloop terug te keren naar de volgende instructie die uitgevoerd moest worden. We bekijken even de TRAP-functie. Deze functie gebruiken we als we tijdelijk een andere routine nodig hebben. Als deze routine is beëindigd, gaat het programma keurig na de TRAP-functie verder.

Onder exceptions verstaan we evenwel heel wat meer dan alleen de TRAP-instructie. Er zijn precies 10 exceptions te onderscheiden, die we weer in twee soorten kunnen verdelen:

1. externe exceptions

- interrupt
- busfout
- externe reset

2. interne exceptions

- ongeldige instructie
- niet geïmplementeerde instructies
- adresseringsfout
- supervisor instructie in user mode
- trace-functie
- deling door nul
- TRAP, TRAPV, CHK

De tweedeling wordt gemaakt op grond van de manier van aanroepen of ontstaan. De interne exceptions worden, zoals u in de tabel hierboven kunt zien, altijd door het programma gegenereerd, terwijl de externe exceptions altijd door de hardware worden gegenereerd.

Interrupts

Van de externe exceptions zijn de interrupts op dit moment het belangrijkste. Interrupts worden gegenereerd als een bepaald randapparaat informatie aan de processor moet geven, of als het randapparaat informatie van de processor nodig heeft. Zo'n interrupt-aanvraag vindt plaats door het prioriteitsniveau van de interrupt aan de processor kenbaar te maken.

De M68000 kent 8 interrupts, waaruit een keuze gemaakt wordt door middel van het prioriteitsniveau. Of de interrupt wordt uitgevoerd, hangt af van de prioriteitswaarde die het statusregister aangeeft. Als het statusregister bijvoorbeeld niveau 4 aangeeft, zal de processor alle interrupt-aanvragen met een prioriteitsniveau lager dan 4 negeren. Niveau 7 is evenwel een uitzondering. Deze

zal bij een aanvraag altijd gegenereerd worden; hij is dus eigenlijk de niet-maskeerbare interrupt of NMI van de acht bits-processoren.

68000 op de vlucht

Vijf van de interne exceptions zijn bedoeld als "vluchtroute" als de processor een programmeerfout tegenkomt. Als de processor een 'ongeldige instructie' of een 'niet geïmplementeerde instructie' vindt, zal hij een exception genereren.

1. Ongeldige instructies

Onder 'ongeldige instructie' verstaat men een WORD waaruit de processor geen instructie kan afleiden. Dit kan bv. voorkomen als de Program Counter in een data-gebied terecht komt.

2. Niet geïmplementeerde instructies

'Niet geïmplementeerde instructies' zijn instructies waarvan de meest significante vier bits (van het eerste woord) 1010 of 1111 zijn. Deze waren bij het ontwerpen wel voorzien, maar zijn om de een of andere reden geschrapt.

3. Deling door nul

Vele scholen gebruiken nog steeds het regeltje: 'Delen door nul is flauwekul!'. De mannen van Motorola hebben zich hier ook stevast aan gehouden, want zodra er een deling door nul plaatsvindt, is dat voor de processor een reden om een exception te genereren.

4. Adresseringsfout

Zodra de programmeur op een oneven adres een woord of een lang woord probeert te lezen of te schrijven, zal er zich ook een exception voordoen die men 'adresseringsfout' noemt. Het voorbeeldprogramma laat zien hoe zo'n exception opgevangen kan worden.

5. Geprivilegieerde instructies

Er zijn ook een paar instructies die alleen in de supervisor mode uitgevoerd kunnen worden. Zodra zo'n instructie in de user mode moet worden uitgevoerd, zal de processor meteen een exception genereren.

Op de voet gevolgd ...

Eén van de handigste exceptions is zeker de TRACE. Zodra men bit 15 van het statusregister set, zal TRACE in werking treden. Na elke instructie die is uitgevoerd, zal deze exception worden gegenereerd en men kan dus op deze manier zeer makkelijk een eigen programma debuggen. Een vereiste is dan wel dat dit programma in user mode moet draaien.

Hoe werkt het nu ?

Aan het voorgaande is duidelijk te zien dat de 68000 precies in de gaten houdt wat er allemaal van hem verlangd wordt en dat hij, zodra hij iets tegenkomt dat niet klopt, dat zelf opvangt. Het is alleen erg jammer dat QDOS deze 'fouten' niet verwerkt en keurig een foutmelding geeft. We zullen dus zelf naar een oplossing moeten gaan zoeken. Voordat we evenwel in QDOS duiken, moeten we eerst wat meer over de afhandeling van exceptions weten. Wat doet de processor nu precies bij het genereren van een exception, en hoe moet zo'n exception-routine er uitzien ?

Als de processor een exception dient te genereren bv. door een TRAP, dan zorgt hij er eerst voor dat A7 naar de supervisor stack wijst. Dit is meteen al een kenmerk van de exceptions : **ze draaien altijd in supervisor mode**. Nu A7 de supervisor stack is geworden, wordt de PC (die naar de volgende uit te voeren instructie wijst) op de stack gezet en daarna het status-register. Nu gaat de processor kijken wat voor type exception het is.

De 68000 microprocessor kent maar liefst 255 verschillende types, waarvan we de eerste 47 zullen bekijken. De types zijn genummerd van 0 tot 255, en elke exception heeft zijn eigen nummer. Natuurlijk moet elke exception een eigen start-adres hebben. Het adres van de offset van elke exception wordt door de processor berekend door het type met vier te vermenigvuldigen en dit bij nul op te tellen. Hierbij valt het al meteen op dat deze exception-vectoren in ROM staan en dat in dit gebied dus niets te veranderen is. Op dit probleem zal ik later terugkomen.

Zodra de processor het adres van de offset heeft berekend, zet hij de inhoud van de offset in de PC. Deze exception moet dan eindigen met de instructie RTE, die zowel de PC als het oude statusregister van de stack haalt. Sommige exceptions zetten evenwel meer informatie weg dan alleen de PC en het statusregister. Bij een busfout, adresfout of een reset, zet de processor namelijk ook nog het adres van de instructie die een exception genereerde (long word), de instructie zelf (word) en extra informatie op de stack. De volgorde is als volgt :

- oude PC (L.)
- oude statusregister (W.)
- instructie (W.)
- instructie-adres (L.)
- extra informatie (W.)

Bij deze exceptions worden er dus 8 bytes meer op de stack gezet. Om het return-adres terug te krijgen, moeten er dus 8 bytes bij de stack opgeteld worden alvorens met het commando RTE terug te keren.

Het laatste word bevat extra

informatie over de reden waarom de exception gegenereerd moest worden. Het derde en vierde bit is voor de programmeur het meest van belang. Bit 4 is 1 als de exception door een schrijf-poging werd opgeroepen. Was dit niet het geval, dan is bit 4 = 0. Het derde bit staat uit als de exception de processor moest onderbreken terwijl die een instructie of een exception aan het uitvoeren was. Normaal staat dit bit dus uit.

Zoals ik hierboven al vermeld heb, staan de vectoren van alle exceptions in ROM. Deze tabel staat volledig aan het begin van het voorbeeldprogramma. De kolom ADRES geeft de adressen aan van de JM-ROM-versie. Deze adressen kunnen dus bij een andere versie anders zijn, maar de exceptions die erbij horen, zijn gelijk.

Wat doet QDOS ?

Nu we precies weten hoe exceptions worden gegenereerd en zijn opgebouwd, kunnen we gaan kijken hoe QDOS ze verwerkt.

Uit de eerste twee gegevens van de tabel kunnen we afleiden dat bij een reset, de SSP op adres \$30000 wordt gezet, en dat de opstart-routine begint op adres \$168.

De routine voor een busfout start op adres \$5C, waar de volgende korte routine staat :

```
ADDQ.W #8,A7
RTE
```

Een busfout handelt QDOS dus zelf af.

We slaan even een paar exceptions over, en kijken naar de interrupts. We zien dat 5 van de 7 interrupts dezelfde vector hebben : \$5E. De routine op adres \$5E is alleen maar de instructie RTE, waaruit blijkt dat deze interrupts op de QL niet bruikbaar zijn. Interrupt-niveau 2 wordt daarentegen wel gebruikt : het is de 50/60 Hz interrupt, ook gekend als de 'polled interrupt'. Deze wordt door QDOS gebruikt voor o.a. het afhandelen van JOBS. Interrupt-niveau 7 is wel door

*Deze tabel bevat alle informatie omtrent de adressen van de exceptions

* VECTOR	Offset	ADRES:	ADRES IN	Exception naam:
* NUMMER: (hex)		VECTOR TAB.		
* 0	0	30000		Reset (Start adres van SSP)
* 1	4	168		Reset (Start van PC)
* 2	8	5C		Busfout
* 3	C	28	52	Adresseringsfout
* 4	10	2A	58	Ongeldige instructie
* 5	14	2C	5c	Deling door nul
* 6	18	2E	60	CHK instructie
* 7	1C	30	64	TRAPV instructie
* 8	20	32	68	Prioriteitsovertreding
* 9	24	34	6c	Trace
* 10	28	61266124		Niet geimpl. instr.
* 11	2C	61226120		Niet geimpl. instr.
* 12	30	}		
* 13	.	}		Gereserveerd voor toekomstig
* 23	5C	}		gebruik.
* 24	60	5E		Value interrupt
* 25	64	5E		Int. niveau 1
* 26	68	340		Int. niveau 2
* 27	6C	5E		Int. niveau 3
* 28	70	5E		Int. niveau 4
* 29	74	5E		Int. niveau 5
* 30	78	5E		Int. niveau 6
* 31	7C	36	70	Int. niveau 7
* 32	80	308		TRAP #0
* 33	84	30C		TRAP #1
* 34	88	312		TRAP #2
* 35	8C	318		TRAP #3
* 36	90	31E		TRAP #4
* 37	94	38	74	TRAP #5
* 38	98	3A	78	TRAP #6
* 39	9C	3C	7c	TRAP #7
* 40	A0	3E	80	TRAP #8
* 41	A4	40	84	TRAP #9
* 42	A8	42	88	TRAP #10
* 43	AC	44	8c	TRAP #11
* 44	B0	46	90	TRAP #12
* 45	B4	48	94	TRAP #13
* 46	B8	4A	98	TRAP #14
* 47	BC	4C	9c	TRAP #15

```

start lea redicttab,a1 Begin adres van TRAP REDIRECTION table
move.l a1,$28050 Zet in system variabele SV TRAPV
move.l $c,a2 Offset van Adres Fout Exception
adda.l #2,a2 Maak er een Return adres van
adda.l a2,a2 Bereken plaats in tabel
adda.l a1,a2 Bereken adres binnen tabel
lea adrfout,a3 Start adres van exception routine
move.l a3,(a2) Zet in tabel
move.w #1,(a1) Genereer een adres fout
moveq #0,d0 Geen errors
rts Terug naar BASIC

```

```

adrfout movem.l d0-d6/a0-a6,-(a7) Bewaar alle te gebruiken registers
move.w $c8,a2 Vector utility UT.SCR
lea scrblock,a1 Start adres def. block voor window
jsr (a2) Roep utility aan
lea channel,a1
move.l a0,(a1) Bewaar CHANNEL ID.
bsr info Print alle informatie
moveq #2,d0 IO.CLOSE
lea channel,a0
move.l (a0),a0 CHANNEL ID. in a0
trap #2 Sluit window
movem.l (a7)+,d0-d6/a0-a6 Haal oude waarden van regs. op
adda.l #8,a7 Extra informatie was 8 bytes lang
rts Keer terug naar programma

```

```

info lea ptekst,a1 Start adres van te printen tekst
bsr prttekst Print de tekst
move.l 70(a7),d5 Waarde van de PC in d5
bsr hexprint Print d5 in HEX.
lea srtekst,a1
bsr prttekst
clr.l d5
move.w 68(a7),d5 Oude Status Register
bsr hexprint

```

de gebruiker te bepalen en hoort dan ook bij een bepaalde groep, die ik verderop zal bespreken.

TRAP #0 t/m TRAP #4 zijn eveneens niet door de gebruiker te definiëren, omdat ze door QDOS gebruikt worden.

Alle andere vectoren (behalve van type 10 en 11) wijzen naar adressen tussen \$28 en \$4C. Gaan we deze adressen bekijken, dan zien we alleen maar de instructie BSR.S \$50 staan. Dat houdt dus in dat steeds bij die exceptions naar de routine op adres \$50 zal worden gesprongen. Op dat adres staat de volgende instructie: BRA \$13A

Al deze exceptions zullen dus naar de routine op \$13A springen; daar staat:

```

TST.L $28050
BEQ $54
MOVE.L A6,-(A7)
MOVEA.W $6(A7),A6
ADDA.W A6,A6
ADDA.L $28050,A6
MOVE.L (A6),$4(A7)
MOVEA.L (A7)+,A6
RTS

```

Het adres \$28050 ligt in het gebied van de systeemvariabelen. Het geeft het adres aan van de TRAP Redirection Table. In die tabel moeten blijkbaar de offsets van de TRAP-instructies komen, maar voor de structuur van de tabel moeten we kijken hoe het programma werkt. Met het commando TST wordt gekeken of er wel een tabel aanwezig is (we gaan er van uit dat er wel een tabel aanwezig is). Daarna wordt de waarde in A6 op de supervisor stack bewaard. Nu wordt er met de instructie MOVE.W het return-adres van het commando BSR.S \$50 van de stack gehaald. Dit mag in word-vorm gebeuren, omdat het adres toch niet hoger dan \$FFFF kan zijn. Met het commando ADDA.L wordt de waarde in A6 verdubbeld. A6 kan nu als pointer dienen, die binnen een tabel in stappen van 4 iets kan aanwijzen. Hierna wordt het tabel-adres in \$28050 bij A6 opgeteld, en voor de weggezette waarde van de oude A6 op de stack gezet.


```

lea instrkst,a1
bsr prttekst
move.w 66(a7),d5
bsr hexprint
lea adrtekst,a1
bsr prttekst
move.l 62(a7),d5
bsr hexprint
lea infotkst,a1
bsr prttekst
clr.l d5
move.w 60(a7),d5
bra hexprint

prttekst lea channel,a0      Adres van CHANNEL ID.
move.l (a0),a0             Zet CHANNEL ID. in a0
move.w (a1)+,d2            Lengte in d2
moveq #-1,d3              Niet aan tijd gebonden
moveq #7,d0               IO.SSTRG
trap #3
rts                         Keer terug

*hexprint print een long getal dat in d5 staat.
hexprint move.w #32/4-1,d4   Een lus van 7 maal
hexprll rol.l #4,d5         Zet nibble in voorste 4 bits
clr.l d6
move.b d5,d6              Gebruik d6 als werk register
andl.b #21111,d6          Laagste nibble overhouden
lea channel,a0
move.b 4(a0,d6.1),d1      Tabel met ascii tekens 0-F
move.l (a0),a0            Zet CHANNEL ID. in a0
move.w #-1,d3             Niet aan tijd gebonden
moveq #5,d0              IO.SBYTE
trap #3
dbra d4,hexprll          7 keer
rts

scriblock dc.b 7,1,0,7      BORDER 1,7:PAPER 0:INK 7
dc.w 448,85,32,16          SCR 448x85a32x16
pctekst dc.b 0,32,'ADRES-FOUT !!!',10
dc.b 10,'Program Counter:'
srtekst dc.b 0,17,10,'Status Register:',0
dc.b 0,12,10,'Instructie:'
adrtekst dc.b 0,18,10,'Instructie adres:'
infotkst dc.b 0,18,10,'Extra informatie:'

channel dc.l 0
hexconv dc.b '0123456789ABCDEF'

redicttab ds.b 0           Start adres van expection table

end                         Einde programma

```

A6 wordt van de stack gehaald en met het return-adres springt men naar het adres uit de TRAP Redirection Table. We kunnen nu voor elke zelf te definiëren exception het adres van diens offset binnen de tabel berekenen. De formule luidt :

$$(\text{vector-adres} + 2) * 2 + \text{tabel-adres}$$

Tabel-adres is hier natuurlijk het adres dat staat in de systeemvariabele op adres \$28050. Vector-adres is het adres dat hoort bij de bepaalde exception.

We nemen als voorbeeld de TRAP#10-exception. Zijn vector-adres is \$A8. Het adres binnen de tabel wordt dan : $2 * (\$A8 + 2) + \28050 . Let wel op dat hier met \$42 het getal op adres \$A8 bedoeld wordt, en met \$28050 de inhoud (.L) van adres \$28050.

Door deze methode te gebruiken, kunnen we nu zelf onze eigen exceptions gaan schrijven. Het voorbeeld-programma maakt een eigen exception om adresfouten op te vangen. In de opstart-routine wordt eerst de exception gedefinieerd en daarna een exception gegenereerd door op een oneven adres een word te schrijven. De exception-routine geeft dan alle informatie die op de stack is weggezet.

Veel plezier bij het programmeren en het gebruik !
(W.D.)

IN/OUT Ik bied aan

ADVERTENTIES VAN LEZERS

De Sinclair gids aanvaardt geen enkele aansprakelijkheid omtrent de inhoud van de aangeboden advertenties. De Sinclair gids biedt zijn lezers slechts de gelegenheid om artikelen aan te bieden of te vragen. De aansprakelijkheid over het gebodene berust ten volle bij de aanbieder of de aanvrager.

Ik bied aan: Spectrum 128+2 met Brother HR-5 printer, 2 joysticks, tientallen orig. software (o.a. Tasword 128, Masterfile) en div. boeken.
Tel: 010-4046135 (na 18.00 uur)

Ik bied aan: QL (JM-versie) met originele software en handleiding
Tel: 02230-34250 (18.00-20.00 uur)

Ik bied aan: DATEX Switchable joystick I/F eventueel ruilen met I/F-2
W.van de Velden, Tel: 073-423089

Ik bied aan: ZX printer Fl. 100,00. Masterfile in originele verpakking fl. 30,00. Tel: 045-422864

Ik bied aan: QL 128K + prof. software pakket + 62 cartr. + boeken + aansluitkabels. Samen fl. 500,00
Tel: 070-478755 na 18.00 uur.

Ik bied aan: Spectrum 128K + Opus + Printer GP50 + Lichtpen + PIO + Progr. Joyst. I/F + Boeken + Software voor fl. 900,00. Tel: 01880-10494

Ik bied aan: ZX Spectrum 48K + Seikosha GP50S Printer + Tijdschriften + Boeken + div. Software fl. 250,00.
H.van Leeuwen Tel: 05759-3033

Ik bied aan Sanyo Datarecorder in zeer goede staat, de prijs n.o.t.k.
F.Vrancken, Norbertijnenstraat 34, 6166 AK Geleen. Tel: 04490-44196

Ik bied aan: Uitstekende QL+Voeding + monitor kabel + stofhoes + CO pascal + Pstion Software + Chess + 39 cartr. (incl. Software) in 2 cartr-boxen + 5 boeken + documentatie fl. 200,00. P.J. Pijnacker, Prinsenhof 32, Pijnacker

Ik bied aan: ZX Spectrum 48K, Saga Keyboard, I/F-1, Idrive, multiface 1, Progr. Joyst. I/F, Div. software, boeken en tijdschriften. fl. 450,00. Tel: 070-675642

Ik bied aan: Multiface 1 48/128 fl. 100,00. Beta-Disk I/F 4.12 fl. 75,00. Seikosha GP-50S + 4 rol fl. 100,00. Progr. Joyst. I/F fl. 25,00. Sinclair/Kempston joyst. I/F fl. 25,00. Incl. eventuele handleidingen. Tel: 074-665840

Ik bied aan: ZX81 computer met 48K geheugenuitbreiding fl. 50,00. Tel: 020-971568

Ik bied aan: Sinclair ZX Spectrum+ met datarecorder, Idrive, I/F-1, software (Tasword 2+3 e.a.). Boeken. Verder een seikosha GP500AS printer prijs n.o.t.k. Tel: 020-418689

Ik bied aan: ZX Spectrum 48K en Spectrum+2 128K. Ook joyst. I/Fs, kabels, cassette-recorder, originele spellen, boeken, tijdschriften etc. Ook los te koop. Bel: Jeroen: 02521-15653

Ik bied aan: veel ORIGINELE software voor ZX81, o.a. Schaken, MCode, Toolkit, Database, Spelletjes etc. Bel: 05473-557/085-433194 vragen naar Freek.

Ik bied aan: Centronics Matrix printer A4 Brother HR-5 + externe voeding, zgan. fl. 150,00 Tel: 03200-21088

Ik bied aan: ZX Idrives (2x)+kabel, Tasword 2, Beta-Basic 3.0, Basic-compiler Handleidingen en boeken 'Werken met Idrive', 'Basicprog', 'ZX machinetaal' voor fl. 150,00. C. de Birk, tel: 04130-40974

Ik bied aan: Spectrum 48K, Aquarius recorder, DK Tronics Progr. joyst. I/F, Quickshot Joystick, Boeken, Software Vraagprijs: fl. 150,00. Tel: 05951-2644 vraag naar Michel !

Ik bied aan: Apple 2E 128K Ram, 80-koloms kaart, 2 drives, joystick, beeldscherm. Incl. Software en Documentatie fl. 1500,0 Tel: 03402-47692

Te KOOP: Spraakgenerator fl. 69,00. ZX Spectrum 16K fl. 50,00. Mirage M'Driver fl. 50,00. Spectrum 48K (defect) fl. 20,00. I/F-1 + Idrive + cartr. + kabels fl. 130,00. Ook veel software en boeken. Tel: 01802-1708

Ik bied aan: Alphacom 32 printer voor de Spectrum Prijs: 1995 BF/fl. 110,00 Rudi Breedenraedt, Kapellestraat 3, B-8591 Moen België Tel: 056-645369

Ik bied aan: Videoface Digitiser met BNC aansluitkabel. Nieuw in doos fl. 150,00. C.L. Snoek, Maconhof 46, Eindhoven

Ik bied aan: Spectrum 128K+2 met originele en Ned. Handleiding. Disciple, 3.5" Diskdrive, Seikosha SP 1000 Printer, Spectrum Joystick, Tasword 128, Spelletjes. Alles zgan. fl. 1000,00. Monochroom Monitor voor: fl. 250,00. Tel: 078-127575

Ik bied aan: Spectrum 48+, Monitor Z/W, Datarecorder, Kempston I/F, Seikosha Printer, veel software en boeken. fl. 400,00. Tel: 01880-40740

Voor de KNUSELAAR: Defecte Spectrum + defecte Interface 1, een in goede staat zijnde Idrive, 10 cartridges, 5 boeken. Roel Peeters, Doornappelstr. 10, 5552 RN Valkenswaard

Ik bied aan: Spectrum+, Idrive, I/F 1 Brother printer, recorder, amber monitor, cartridges. Prijs fl. 500,00 Tel: 05920-50147

Voor de KNUSELAARS: 2x QL met 512K; CST Diskinterface, Printer I/F, alle QL-IC's o.a. processoren, Sandy keyboard I/F, midi I/F, Sample I/F, Toolkit 2. Spotprijs ! Tel: 011/565930 (België)

Ik bied aan: QL-128K, 2x 3.5" drives, Zentith Amber-monitor, Disks, Boeken, Tijdschriften en software. fl. 1500,- Tel: 050-135656

Ik bied aan: QL, Philips Monitor, Printer I/F, programma's zoals Chess. Alles 1 jaar oud: Prijs fl. 450,00. Tel: 079-510326. Scarlattirode 10, Zoetermeer

Ik bied aan: Spectrum 128K met Disciple en 3.5" Drive en Zentih Monitor, Brother HR5 en veel programma's voor fl. 950,00. Eventueel ook afzonderlijk te koop. A. Meckel Tel: 02963-1846 (na 18.00uur)

Ik bied aan: ZX Spectrum+, Lo Prof. Keyboard, Idrive, Brother HR5, Timex 2040 printer, Joystick, 22 cartridges Prijs fl. 800,00. Tel: 02158-3973.

Ik bied aan: Seikosha GP500A Printer (serieel) voor Spectrum met originele Tasword 3 fl. 325,00. J. Goddijn, Drachten. Tel: 05120-12519

Te koop: ZX Spectrum 128K+ Kleuren-monitor, Recorder, Interface 1, Idrive, cartridges, lichtpen en veel software. Prijs 15000 F/ fl. 750,00. Te bevragen: Danny De Maeyer, Dodoensstr. 58, B-2800 Mechelen Tel: 015-419521 (na 18.00uur).

Ik bied aan: Astronomer fl. 25,00. DK-Tronics Lichtpen fl. 50,00. Sinclair Printer met 2 rol papier fl. 40 C. Snoek, Maconhof 46, Eindhoven.

Ik bied aan: Spectrum 80K in Lo Prof. Keyboard met resetknop, cassette-recorder en programma's. Tel: 02230-34250 18-19 u

Ik bied aan: Spectrum 48 (defect), I/F 1, Idrive, Kraak I/F, Joystick I/F, datarecorder, Boeken. Tel: 01802-1708. (Patrick)

Ik bied aan: Printer Brother M1109 ser+par. in goede staat. fl. 475,00. Tevens Sprocket feeder fl. 65,00. Tel: 01807-20015

Ik bied aan: Spectrum 128K, I/F 1, Multiface 1, Progr. Joystick I/F, Idrive, 10 cartr., 30 progs, boeken. Prijs fl. 600,00. L.J. Brouwer Tel: 020-199890

Ik bied aan: Spectrum+, I/F1, Joyst. I/F (3 poorten), Idrive, 10 cartr., 30 progs, boeken. Prijs fl. 500,00 L.J. Brouwer Tel: 020-199890

Ik bied aan: Timex 1000 computer (ZX81), 16K Rampack fl. 70,00. Bel: 080-442638 na 18.00 uur

Ik bied aan: ZX Interface 2 +ROM cartr. Planetoids. samen fl. 130,00. GP-50S printer +papier fl. 110,00. Bel: 080-442638 na 18.00 uur

Ik bied aan: Spectrum+ nog 1 mnd garantie ! + defecte Spectrum 48K+ fl. 50,00. RUILEN voor Spectrum 128K. J. Morales. Tel: 010-4731405

Ik bied aan: Spectrum 48K, Opus Discovery, Seikosha GP250X, Boeken en Tijdschriften voor fl. 750,00. J. van Leeuwen, Voorhout. Tel: 02522-12518

Ik bied aan: ZX Spectrum 48K, I/F1, I/F-2, Datarecorder, veelsoftware en boeken: fl. 200,00 Tel: 010-4356256 na 18.00 uur

Ik zoek

Ik zoek Masterfile-gebruikers met Disciple en diskdrive-ervaring. Luch Leffers: tel: 05150-21271

Hoe pas ik een een Citizen Commodore-compatible printer aan voor gebruik met Spectrum 128K+ en Disciple. Luch Leffers Tel: 05150-21271

Ik zoek een programma voor ZX Spectrum om signaal van DCF77 (Atoomklok) via EAR-Plug binnen te halen en te de coderen. Eric Augusteijn, Tel: 05195-1647

Ik zoek een programma om de Elra Telexconverter te gebruiken met de Spectrum. Tel: 03434-54528

Ik zoek Tijdschriften/Spellen van 1983 t/m 1986 voor de Spectrum 48/128 en de Sinclair Gids 1 t/m 3. W. van de Velden Tel: 073-423089 (na 18.00 uur). Gids 1 t/m 3 nog steeds leverbaar ! (Redactie).

12

Onder omslag
als drukwerk
verzenden.

sinclair gids

Postbus 111

5110 AC Baarle-Nassau

12

Onder omslag
als drukwerk
verzenden.

sinclair gids

Postbus 111

5110 AC Baarle-Nassau

Ik zoek contact met iemand die
ervaring heeft met de timex 2040
printer. H.Smits tel:010-465608

Ik zoek een defecte Interface 1 voor
onderdelen. Tel: 045-242272

Ik zoek een Interface 1 voor ongeveer
25-50 gulden. Bel ca. 18.00 uur:
02508-3326 (Jesse)

Ik zoek Educatieve Software voor
Spectrum 128K+2.
F.Nelson Tel: 03430-15271

Ik zoek alle mogelijke Multiface
Pokes zoveel mogelijk ! Bel me snel:
A.Adema Tel: 023-240551 (Haarlem)

WIE kan mij vertellen hoe ik mijn
viditel-modem op mijn 27 Mc
apparatuur kan aansluiten !!!
Björn Dinse, Kandelaar 37, Heiloo
Tel: 072-336343

Ik zoek defekte ZX81 met werkend
toetsenbord of toetsenbord alleen is
ook goed. H.C. van Aert, Buurtweg 67,
Halsteren Tel: 01641-3136

Ik zoek handboek en aansluitgegevens
Spectrum 128K. A. Kolkman, Eemmeer 10
Zaandam. Tel: 075-161757
Opm.Redactie: Al aan het Handboek 128
K+2 gedacht van onze uitgeverij ?

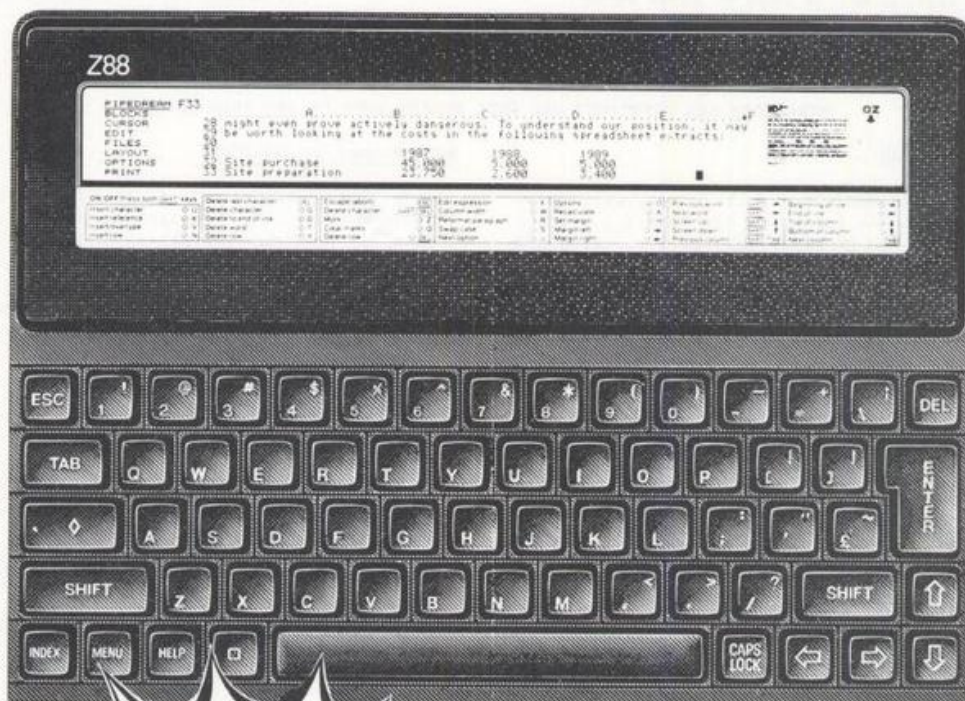
Ik zoek 'Guideboek van Tower of
Despair' tegen vergoeding.
L. Callaerts, Min. De Taeyelaan 1, B-
8500 Kortrijk (België) tel:056-200005

Ik zoek Tijdschriften, Interface 2/3
en Fruitmachine Spel voor Spectrum
16/48K.
W. van de Velden Tel:073-423089

Ik zoek Spectrum 128K fl.175,00.
Tel: 02230-34250 (van 18.00-20.00uur)

Computerclub Zutphen zoekt Sponsor.
2x per maand bijeenkomst.
Tel: 05750-13897

Z88



fl. 995,00
excl.BTW

Nu met GRATIS
32K RAM- of
EPROMpack

t.w.v. fl. 80,00 !

CAMBRIDGE COMPUTER

Z88, de enig echte portable computer

A4 formaat en weegt minder dan 1 kg.

Complete personal computer met ingebouwd display, RS-232-poort en div. software-pakketten in EPROM: o.a. Tekstverwerker, Database, Spreadsheet, BBC-Basic, Diary, Calendar, Calculator.

Eenvoudige gegevens-overdracht van en naar IBM-compatibles.

Vraag onze Z88-folder aan of kom langs voor demonstratie.

Data-Skip Gouda heeft ook alle Z88-randapparatuur op voorraad zoals: par.- en ser. printerkabels, RAM- en EPROM-uitbreidingen, Netvoeding en Z88-PC Link.

DATA-SKIP GOUDA
Oosthaven 58
2801 PE Gouda

Tel: 01820-20581
Open: di t/m za
van 10 tot 17 uur

computercollectief

Amstel 312 (t.o. Carré) / 1017 AP Amsterdam / Giro 4 475 158 / Bank NMB 69.79.15.646

* onze nieuwe ZOMER 1988 CATALOGUS is nu uit. *
* stuur ons een kaartje met je naam en adres en de *
* vermelding 'Sinclair Gids' en we sturen hem gratis toe. *

in BELGIE zijn al onze artikelen verkrijgbaar bij :
Het Computerwinkeltje pvba,
M Sabbestraat 39, B-2800 MECHELEN
telefoon (015) 206 645

HIERONDER EEN OVERZICHT VAN ONZE BOEKEN VOOR DE SPECTRUM EN QL

BOEKEN voor de SPECTRUM

Handboek voor ZX SPECTRUM 128 + 2	32,90
*Spectrum 128 Companion - Including 128 PLUS-2 ...	22,00
100 Programma's voor de ZX Spectrum	55,00
BASICODE-3 boek & cassette	27,50
BASIC met de ZX Spectrum	25,50
Machine Code met de ZX Spectrum	30,70

Speciale Aanbieding (OP=OP)

Leren Programmeren ZX Spectrum+ boek 2	10,00
Machinetaal voor de ZX Spectrum	10,00
ZX Spectrum Hardware Boek	10,00
ZX Spectrum Machinetaalroutines	10,00

BOEKEN voor de SPECTRUM

Praktijkprogramma's voor de ZX Spectrum deel 2 ..	19,15
QUESTO Meerkeuze toetsprogramma voor ZX Spectrum	18,90
Van BASIC naar Machinetaal op de ZX Spectrum	17,90
Financiële Programma's voor de ZX Spectrum	25,75
Programmatuur I voor de ZX Spectrum	23,25
Toolkits en Enkele Spelen voor de ZX Spectrum ...	25,75
Het ZX Spectrum (+) Software boek	34,90

BOEKEN voor de QL

Het Sinclair QL Handboek	20,00
Sinclair QL leren programmeren	24,75
QL Assembly Language Programming	59,00

ACTUELE EN NIEUW BINNENGEKOMEN SOFTWARE

SOFTWARE voor de SPECTRUM

3 Coin Op Classics	39
Breakthru, Kung Fu Master, Crystal Castles	39
3D Game Maker	39
3D Game Maker disk Spectrum+3 ...	39
6 PAK vol 2 -	39
Ace, Intern Karate, Light Force Batty, Shockaway Rider	36
*6 PAK vol 3	36
Paperboy, Enduro Racer, Living Daylights etc.	36
720 Degrees	36
ACE 2	39
Agent X II	10
*Arkanoid II	32
ATF -advanced tactical fighter ..	36
ATV Simulator	10
Barbarian - the Ultimate Warrior	39
*Black Lamp	32
*Brainstorm	10
Bobsleigh	39
California Games	36
Colossus Bridge 4.0 (Acol)	49
Colossus Chess 4.0	45
Combat School	32
*Cybernoid	32
Dark Sceptre -mark singleton! ...	32
Driller	59
*Earthlight	32
Elite Collection - 8 games	59
oa Paperboy,Commando,Bombjack Enduro Racer	39

SOFTWARE voor de SPECTRUM

*Firefly	32
Flash Gordon	15
Fruit Machine Simulation	10
Game, Set and Match	55
20 sportsimulaties	36
*Garfield - big, fat, hairy deal .	36
Gauntlet	36
Gauntlet II	29
Gnome Ranger (level9 adventure) .	39
Grand Prix Simulator	10
Gunship	35
High Frontier (SDI wargame)	39
*Ikari Warriors	36
International Karate +	39
Jet Bike Simulator	20
Knight Orc (3 level9 adventures)	59
Last Mission	36
Live Ammo	39
Army Moves, Rambo, Green Beret, Top Gun, Great Escape, the Living Daylights	39
*Magnetron	32
Magnificent 7	39
oa Wizball, Frankie, Great Escape	36
Matchday II	36
Mercenary	39
Outrun	36
PAW Professional Adventure Writer	89
opvolger van de Quill	39
*PHM Pegasus	39
Platoon	39
*Predator	39

SOFTWARE voor de SPECTRUM

Rampage	39
*Race Against Time	20
Road Runner	36
Sidewize	32
Silent Service (duikboot)	39
Silicon Dreams	59
Solid Gold	39
10th Frame,Dambusters,BeachHead II Gauntlet,Wintergames,Infiltrator Starglider	59
*Sorcerer Lord (Phantasy Wargame)	55
Summer Gold	39
*Target: Renegade	32
Thundercats	32
*Venom Strikes Back	32
Wizball	32
World Glass Leaderboard	36
World Games - 8 sporten	36
programmeertalen ed.:	
DEVPAK 4 editor/assembler/monitor	59
*DEVPAK 4 disciple version	79
*DEVPAK 4 opus version	79
Hisoft BASIC Compiler	95
Hisoft Pascal 1.7	95
*Hisoft Pascal opus version	125
*Hisoft Pascal disciple version .	125
Hisoft C Compiler	95
Tasword III	89
Mini Office	29
MICRODRIVE CARTRIDGES leeg, p. st .	10

winkel open van dinsdag t/m zaterdag tussen 10.00 en 17.00 (maandag gesloten) - alle prijzen inclusief BTW
verzendkosten f 6,- per bestelling - vraag onze nieuwe ZOMER 1988 CATALOGUS aan.

microcomputer tijdschriften boeken en software

dealer aanvragen welkom