

# BULLETIN

SINCLAIR  
GEBRUIKERSGROEP  
GRONINGEN/ASSEN



Vrolijk pasen!



8e jaargang nr8  
april '91



# BULLETIN SGG

## COLOFON



VOORZITTER/  
PENNINGMEESTER/  
VERHUUR:  
Jan Arends  
Heiligelaan 66  
9636 CP Zuidbroek  
tel: 05985-2247  
Giro 5965342 t.n.v.  
rekening SGG.

SECRETARIS:  
Martin den Hollander  
Numero Dertien 8  
9644 TV Veendam  
tel: 05978-45474

VICE VOORZITTER/  
VICE SECRETARIS:  
Roelof Koning  
Selwerderstraat 26  
9717 GK Groningen  
tel: 050-124298

REDAKTIE CONTACT/  
VICE PENNINGMEESTER/  
MATERIAAL COMMISARIS:  
Coen Ballintijn  
B. Boermalaan 7  
9765 AP Paterswolde  
tel: 05907-1482

---

Redaktie: Mevr. F. Elstrodt, Rudy Blesma, Tonnie Stap en Johan Koning.  
Correspondentie adres: Coen Ballintijn, adres: zie boven.

---

*Kopij en vragen graag aan de redaktie contactpersoon!*

*Het SGG-Bulletin is een uitgave van de Sinclair gebruikersgroep Groningen/Assen. Het bulletin verschijnt 10 keer per jaar in de maanden september tot en met juni. Artikelen, listings, illustraties en andere inzendingen zijn voor verantwoordelijkheid van de inzender. Gepubliceerde programma's zijn getest maar niet gegarandeerd zonder fouten.*

*De sluitingsdatum voor kopij wordt in elk bulletin op pagina drie vermeld. Overname van artikelen, illustraties en andere publicaties zijn uitsluitend toegestaan met toestemming van de redaktie.*

*Het lidmaatschap van onze gebruikersgroep bedraagt per kalenderjaar f 20,00 voor personen tot en met 17 jaar en f 30,00 voor oudere personen. Bij deze prijs is het abonnement op het bulletin inbegrepen.*

## U kunt lid worden van de SGG

*Door u op te geven bij de penningmeester.*



## VAN DE REDAKTIE



Zo dat zit er bijna op, alleen nog dit stukje schrijven, paginanummers er op en het bulletin kan naar de drukker. Dat moest erg snel deze keer want tussen de kopijsluiting (= de gebruikers bijeenkomst) en het naar de drukker brengen zat deze keer maar vijf dagen. Dit omdat we het bulletin minstens een week voor de volgende bijeenkomst bij u in de bus wilt hebben. Verder is de drukker dicht op Goede vrijdag en met Pasen dus er was echt haast bij.

Over Pasen gesproken, leuke voorpagina hè, alleen jammer dat het al een week geleden is, met de 1 april moppen waren we gelukkig wel op tijd zoals u kunt lezen op pagina acht. Voor het volgende bulletin verwacht ik trouwens wel wat meer kopij want deze keer was het echt om te treuren (toch nog 24 pagina's geworden). Vooral artikelen over printers e.t.c. zijn zeer welkom. Even iets over de vorm van de kopij, we hebben die het liefst op tape, OPUS of DISCiPLE disk. De artikelen als TW2 of TW3 files, en eventuele BASIC en plaatjes daarbij op. In de Tasword files liefst geen controle codes voor allerlei vreemde print-opties want die zijn vaak verschillend. Komen er in de TW3 files geaccentueerde tekens voor, stuur dan even een proefuitdraai mee omdat die bij ons wel eens onder een andere toets kunnen zitten. Denk er verder aan dat onze eerste pagina v.e. artikel uit 51 regels bestaat en de volgende(n) uit 56, als u hier bij het schrijven rekening mee houdt is het niet nodig alinea's te onderbreken. Trouwens, enkele regels minder mag ook, vooral meer is een probleem. Helemaal mooi zou het zijn als het artikel ook nog precies een geheel aantal pagina's lang is, maar we kunnen natuurlijk altijd opvullen met plaatjes en SGG-tjes.

In dit nummer:

	<u>auteur</u>	<u>blz.</u>
- Colofon	: bestuur	2
- Bijeenkomsten	: redactie	4
- Muziek voor de 128K Spectrum	: Frans Postma	4
- Van de voorzitter	: Jan Arends	5
- In's en Out's	: Roelof en Coen	6
- SODT in april	: Johan Koning	8
- Machine Code stap voor stap (8)	: J. van Alteren	10
- Zichtbaar sorteren	: Roelof Koning	13
- Esgeeggetjes	: Redactie	13
- Interface 1 hookcodes op DISCiPLE	: Tonnie Stap	14
- Skipload	: Tonnie Stap	16
- Diskdrive op track - 1	: Rudy Biesma	22
- Beste maatjes met SODT	: Johan Koning	23

Sluitingsdatum kopij: 16 april

## BIJeenKOMSTEN



In het: DENKSPORTCENTRUM  
OLIEMULDERSWEG 43  
GRONINGEN

Telefoon: 050-126937

### DATA GRONINGEN:

16 april dinsdag van 19.30 - 22.30  
25 mei zaterdag van 14.00 - 17.30 Printerspecial.  
18 juni dinsdag van 19.30 - 22.30 Ledenvergadering.

De bijeenkomsten van de HCC in Houten zijn dit jaar op  
11 mei, 22 juni, 24 augustus en 26 oktober.  
(Onder voorbehoud van wijzigingen)

## Muziek voor de 128K Spectrum

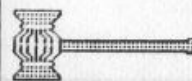
```

250 LET f$="T200M14UX2000W000N3c08BBO5f00c08BO0c08BBBO5f08BBB)"
251 LET g$="T200M14UX6000W000N5c05B&&"
252 LET d$=f$( TO 41)+g$
253 LET e$=d$+f$( TO 49)+d$+f$( TO 49)+g$+f$
254 LET a$="T200N3#f#fd04b&b05&e&e&e#g#gabaaae&d&#f&#f&#fee#fe3
#f#fd04b&b05&e&e&e#g#gabaaae&d&#f&#f&#f5e3&bH"
255 LET b$="03b&3&b&e7&5E&a&&3&D7&5#C&03b&3&b&e7&5E&a&&3&D7&5E&
"
256 LET i$="6D5D3#C5_9b3#C5#C#C3a5&3&5#F3#F5#FE6D3DD5#C8_3b3&e5
#C3D5#C3ba5ab3#C5ba&DD3D3_9DH"
257 LET j$="04N3#f#fd$c7&5Ee3#G#GABO3N5a&&3&a5D&#C&3bb5b7&04N5E
e3EE#F#GO3N5aaaaDD#C#C3#F#FDb7&04N5Ee3EE#F#G"
258 LET k$="&3aaaaaaa6#g3#g6#f04N9a#GA05N6e#f5e9aE#F6E6#F5E9#C#
GA5&3B5##B3B7AO6N8_9E9a9#g05N9_9#G9_9G9_5#C5e7#g9_5g#f7#C3#C6_7_
7D7A3A6_7_9bH"
259 LET l$="03N8#f5e9d03N3aa5a3&a&a5#g#g3#g#g5#f#f3&#f&#f03N5
d8&5aa3&a&a5#g#g3#g#g5#f#f3&#f&#f5d8&03N3aa5a3&a&a5#g#g3#g#g5
5#f#f3&#f&#f03N5d&e&aa3&a&a5#g#g3#g#g5dd3&d&dee5e3eeee03N3#C&&
#C&&#C#C#C#C#C&#C&#C#Cg&&g&&gggg&g&&gg03N3#C&&#C&&#C#C5#C8&3g&&g&
&gg5g8&03N5b&&3bbbb8&5e&&3eeee8&"
260 LET g$="T200M14UX2000W000N3c"
261 LET m$="05N3#f#fd04bV14&b05&e&V13e&e#gV12#gabaV11aae&V10d&#
f&V9#f&#feeV8#fe#f#fV7d04b&bV605&e&eV5#g#gabV4aaaeV3&d&#fV2&#f&V
1#fee#f"
262 PLAY f$( TO 49): PLAY f$( TO 49): PLAY f$( TO 49)
263 PLAY f$,a$,b$
264 PLAY e$,i$,j$
265 PLAY f$,k$,l$
266 PLAY g$,m$

```



## VAN DE VOORZITTER



Zojuist werd ik gebeld door Tonnie. Hij wilde graag mijn artikel voor het bulletin. Kun je er niet een keer zonder? Nee, liever niet. Er is al te weinig copy. Oke, morgen heb je het thuis.

Waarschijnlijk het probleem van iedere redacteur van welke uitgave dan ook. Het kan en mag toch niet zo zijn, dat het bestuur en de redactie van een vereniging iedere maand maar moeten zorgen dat er voldoende leesvoer is voor de leden. Het zou de zorg moeten zijn van een ieder die de Sinclair computers een warm hart toe draagt. Denk mee, doe mee en laat eens iets van je lezen.

Zaterdag 23 maart was er weer een clubmiddag in het Denksportcentrum te Groningen. We hadden een grotere zaal gehuurd dan normaal, afspraken gemaakt met de kantinebaas om voldoende koffie klaar te hebben, extra spullen meegenomen om te laten zien want er kwam een delegatie van een zustervereniging uit het westen van het land. Het werd 2 uur, half drie, drie uur maar geen bezoekers. Zou zelfs de telefoon het af hebben laten weten?

Noteer alvast in Uw agenda: Dinsdag 18 juni algemene ledenvergadering. Reserveer deze avond voor de vereniging.

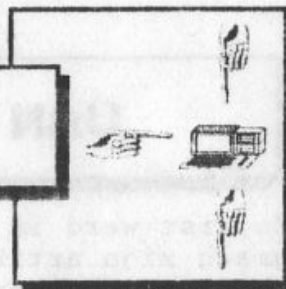
Tot ziens

J.H. Arends



## IN'S EN OUTS'S

De IN en OUT poorten van de SPECTRUM.  
Uitgelegd door C. Ballintijn en R. Koning.



Het (oude, engelstalige) SPECTRUM-handboek spreekt in Hst. 23 over INPUT- en OUTPUT-ports die op de SPECTRUM aanwezig zijn, en waarvoor je zelf een gebruik zou kunnen verzinnen. Ik wil proberen aan te geven wat hiermee bedoeld wordt en hoe dit werkt.

In deze eerste aflevering beginnen we met enige uitleg over de elektrische werking van de SPECTRUM.

Een microprocessor-chip van het type Z80-A vormt het hart van onze SPECTRUM, deze chip voert al het interne rekenwerk uit. Deze processor werkt uitsluitend met getallen tussen 0 en 255. Die getallen kunnen worden weergegeven in het binaire stelsel d.m.v. enen en nullen, de z.g.n. 'binary units' of 'bits', en hiervan zijn er acht nodig om tot 255 te kunnen tellen. Dec. 0 = BIN 00000000, en Dec. 255 = BIN 11111111.

Je zou binaire getallen ook kunnen uitbeelden met acht lampjes waarbij kan gelden 'aan'=1 en 'uit'=0, of (en zo gebeurt het binnen de SPECTRUM), m.b.v. acht draden waarop een elektrische spanning staat. '1' = 5 Volt en '0' = 0 Volt. Deze draden heten DATAlijnen, (in werkelijkheid zijn het koperbanen op een 'print') en worden tesamen DATA-BUS genoemd, omdat binnen een computer meestal meerdere chips op dit getallen-transporteersysteem zijn aangesloten. (Omnibus = voor allen). De getransporteerde getallen (spanningen dus) stellen, afhankelijk van het verloop van een programma, een programma-instructie, een letterteken of een werkelijk getal voor. (Door een programmeertruuk worden getallen boven 255 eerst in delen opgesplitst voor een bewerking plaatsvindt).

In de microprocessor zelf is praktisch geen geheugen aanwezig. Alle programma-onderdelen én gegevens én getallen (alle DATA), moeten daarom in aparte geheugenchips worden ondergebracht. Dat gebeurt dus m.b.v. acht bits via de DATA-BUS.

Omdat de microprocessor DATA uit het geheugen moet kunnen halen (lezen), en er ook weer DATA in moet kunnen opbergen (schrijven), heeft elke geheugenplaats een volgnummer, ADRES geheten. Er zijn 65535 geheugenplaatsen (adressen) in een 48K SPECTRUM. Deze kunnen in het binaire stelsel door 16 nullen of enen worden weergegeven. Om het transport van DATA tussen de chips te regelen zijn dus 16 draden nodig, de ADRESlijnen of ADRES-BUS. De Z80-A chip bezit deze 16 aansluitingen, en kan, door er een getal tussen 0 en 65535 'op te zetten', een bepaalde geheugenplaats (een adres) aanwijzen. Deze geheugenplaats kan dan via de DATA-BUS uitgelezen of beschreven worden. De ADRES-BUS en de DATA-BUS werken met z.g.n. positieve logica, d.w.z. dat een hoge spanning (ca. 5V) geldt als een binaire '1', en een lage spanning (ca. 0V) geldt als een binaire '0'.



# BULLETIN SGG

Om te kunnen bepalen of er 'gelezen', dan wel 'geschreven' moet worden, zijn er nog twee draden die de Z80-A met het geheugen verbinden: de 'readline', ( $\overline{RD}$ ) en de 'writeline' ( $\overline{WR}$ ). Verderop meer daarover.

Een computer zou weinig nut hebben wanneer het niet mogelijk was gegevens met randapparatuur uit te wisselen. Zo moet er van het toetsenbord, joystick, en cassette recorder gelezen kunnen worden, en er naar een printer, diskdrive, cassette recorder geschreven kunnen worden. De Z80-A kan dit op een slimme manier, zonder dat er veel extra aansluitingen nodig zijn. Dat gaat als volgt:

De 16 ADRESlijnen die het geheugen adresseren worden ook naar alle randapparaten gevoerd. Die krijgen allemaal een adres (nummer) toegewezen, en zijn dus net als het geheugen, te adresseren (te kiezen). Ook de  $\overline{RD}$  en de  $\overline{WR}$  signalen worden naar de verschillende randapparaten gevoerd. Om aan te geven of er bij een bepaald adres een geheugenplaats of een randapparaat bedoeld wordt gebruikt de Z80-A twee extra signalen, n.l. 'memory request' (MEMREQ) en 'In/Out request' ( $\overline{IOREQ}$ ). De een activeert het geheugen, de ander activeert de randapparatuur. In principe zijn voor randapparaten de adressen 0 - 65535 weer beschikbaar, en zouden er dus 65535 verschillende randapparaten kunnen worden aangesloten. (In een volgende aflevering zullen we zien dat dit in de SPECTRUM-praktijk niet haalbaar is.)

In tegenstelling tot ADRES- en DATA-BUS werken de signalen  $\overline{RD}$ ,  $\overline{WR}$ , MEMREQ en  $\overline{IOREQ}$  met z.g.n. negatieve logica, dat wil zeggen dat deze signalen 'laag' zijn (0V) wanneer ze geldig zijn, en 'hoog' (5V) wanneer ze niet gebruikt worden. Dit wordt aangegeven door het streepje boven deze namen.

Deze besturings-signalen op een rijtje:

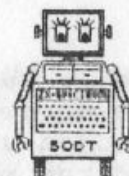
$\overline{RD}$  : laag wanneer de Z80-A DATA inleest.  
 $\overline{WR}$  : laag wanneer de Z80-A DATA verzendt.  
MEMREQ : laag wanneer er een geheugenplaats aangesproken wordt.  
 $\overline{IOREQ}$  : laag tijdens In/Out operaties met randapparatuur.

Duidelijk zal zijn dat (normaal)  $\overline{RD}$  en  $\overline{WR}$  nooit gelijktijdig laag zullen zijn, en evenmin MEMREQ en  $\overline{IOREQ}$ . Daarnaast moet voor elk DATA-transport een bepaald adres aangewezen worden.

Op grond van de in een (machinecode-)programma voorkomende instructies, zorgt de Z80-A chip er zelf voor dat de benodigde spanningen op de betreffende aansluitpennen komen te staan. Aan de geheugenchips en de div. randapparatuur wordt verder overgelaten om uit te maken voor wie wat bestemd is. Een kleine complicatie daarbij is, dat meerdere onderdelen niet tegelijkertijd van de DATA-BUS gebruik kunnen maken, de Z80-A zou dan hopeloos in de war raken. Elk onderdeel moet dus precies weten wanneer het 'aan de beurt' is, en het brokje techniek dat daarvoor zorgt heet 'adresdecodering'.

Daarover meer in de volgende aflevering.

## SODT IN APRIL



door: Johan Koning.

Vanwege de vele reacties die ik kreeg op mijn artikel over een nieuwe DISKDRIVE, die begin deze maand zou verschijnen, wil ik hier nog even duidelijk uitleggen dat dit een 1-APRIL-grap was. Jens en Olaf KER geeft JOKER.  
Begin (toen: volgende, is nu:) deze maand is 1 APRIL.  
De naam van het NIET bestaande IC 1661 is een verdraaiing van het WEL bestaande IC 6116.  
Als je 1661 een kwart slag met de klok mee laat draaien, krijg je het jaartal 1991.  
Samen geeft dit duidelijk aan dat het "slechts" een grapje was.

Om in de sfeer te blijven volgt nu een grappig programmaatje, dat je vertelt waar je ALLE informatie die je zoekt kunt vinden. De routines die gebruikt zijn komen allemaal uit de SINCLAIR-GIDS uitgave september 1987. Ze zijn door mij alleen iets aangepast en samen gevoegd zodat dit programmaatje ontstond.

## SGGreclame

```

10 DIM I(7): LET F$="INFORMATIE": LET G$="OVER"
20 LET I(1)=17: LET I(2)=8: LET E$(1)="MACHINECODECURSUS": LET
  E$(2)="PRINTERS"
25 LET I(3)=12: LET I(4)=10: LET E$(3)="TASWOORD 2/3": LET E$(
  4)="MASTERFILE"
30 LET I(5)=8: LET I(6)=14: LET E$(5)="DISCIPLE": LET E$(6)="O
  PUS DISCOVERY"
35 LET I(7)=3: LET E$(7)="SAM": LET E$(8)="S.G.G. NIEUWS"
40 PRINT INVERSE 1;" WAAR OVER WIL JE INFORMATIE ? "; INVERS
  E 0
50 FOR F=1 TO 8: PRINT AT F*2,10;F;" ";E$(F): NEXT F
60 INPUT "ENTER KEUZEGETAL => ";K: IF K<1 OR K>8 THEN GO TO 6
  0
65 IF K<8 THEN LET A$=E$(K)( TO I(K)): GO TO 90
70 POKE 23658,8
80 INPUT "WELK ONDERWERP ? (MAX. 18 KAR.)": TAB 6;"=> ":A$: IF
  LEN A$<1 OR LEN A$>18 THEN GO TO 70
90 LET D$="HET S.G.G. BULLETIN"
100 RESTORE : CLS : FOR a=30000 TO 30018: READ b: POKE a,b: NEX
  T a: PRINT AT 5,10; FLASH 1;F$: AT 8,13;G$: FLASH 0: PAUSE
  100: RANDOMIZE USR 30000: DATA 33,0,64,78,6,8,203,17,31,16,
  251,119,35,124,254,88,32,241,201
150 PAUSE 30: CLS : BEEP .1,10
200 RESTORE 210: FOR f=60000 TO 60025: READ a: POKE f,a: NEXT f
210 DATA 6,192,17,255,87,213,225,43,197,1,31,0,26,237,184,35,54
  ,0,43,43,27,193,16,240,201,254
220 FOR F=1 TO LEN A$: PRINT AT INT ((20- LEN a$)/2)+f,0;A$(
  F): NEXT F: PAUSE 5: FOR g=0 TO 30: RANDOMIZE USR 60000: NE
  XT g

```

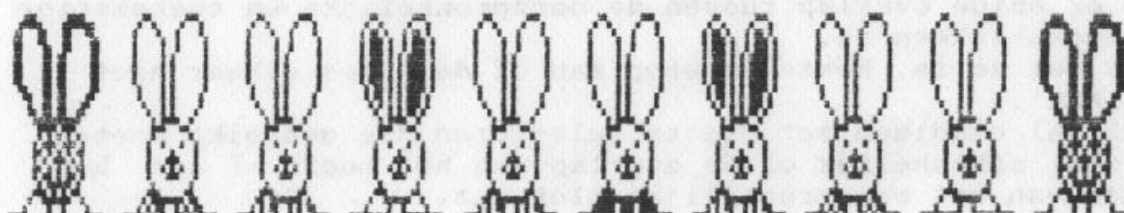


# BULLETIN SGG

```

300 RESTORE 310: FOR f=60000 TO 60025: READ a: POKE f,a: NEXT f
310 DATA 6,192,17,0,64,213,225,35,197,1,31,0,26,237,176,43,54,0
    ,35,35,19,193,16,240,201,254
320 FOR g=1 TO 32: RANDOMIZE USR 60000: NEXT g
400 RESTORE 450
410 FOR f=60000 TO 60018: READ a: POKE f,a: NEXT f
420 FOR g=1 TO 19 STEP 2: PRINT AT g,0;a$: AT g+1,32- LEN a$;a
    $: NEXT g
430 FOR g=0 TO 10: RANDOMIZE USR 60000: NEXT g
440 PAUSE 5: CLS
450 DATA 33,0,64,6,24,197,6,0,126,238,255,119,35,16,249,193,16,
    243,201
500 LET Z=2: FOR V=1 TO 2: CLS #: LET Y=64: LET b$="VIND": GO S
    UB 510: LET Y=72: LET b$=" JE": GO SUB 510: LET Y=80: LET b$
    =" IN": GO SUB 510: PAUSE 50: LET Z=Z-1: NEXT V
505 PAUSE 5: CLS : BEEP .1,10: GO TO 600
510 FOR X=Y TO Y+7 STEP Z: POKE 23681,X: LPRINT TAB 12;b$: NEX
    T X
520 RETURN
600 CLS : DIM C$(32): FOR i=11 TO 21: PRINT AT i,0: INK 7: PAP
    ER 0;C$: NEXT i: PRINT AT 10,1: INK 9: PAPER 9;D$: AT 11,31
    - LEN D$: PAPER 0: INK 7;D$
610 FOR F=1 TO 60: BORDER 0: BORDER 0: BORDER 0: BORDER 0: BORD
    ER 0: BORDER 7: BORDER 7: BORDER 7: BORDER 7: BORDER 7: NEXT
    F
700 RESTORE 710: FOR f=60000 TO 60037: READ a: POKE f,a: NEXT f
710 DATA 33,63,5,229,33,128,255,203,127,40,3,33,152,12,8,19,221
    ,43,243,62,2,71,16,254,211,254,238,15,6,10,45,32,245,5,37,32
    ,241,201
720 FOR h=1 TO 2: FOR g=0 TO 7: POKE 60006,10: POKE 60020,g: PA
    PER g: PRINT AT 10,7;d$: CLS : RANDOMIZE USR 60000: NEXT g
    : NEXT h
800 PRINT AT 0,0,: CLS #: FOR f=54 TO 87: POKE 23681,f: LPRINT
    " ";D$;" ": NEXT f
801 LET t=0
810 BORDER 2: BORDER 7: BORDER 1: BORDER 0: BORDER 0: LET t=t+1
    : GO TO 810+(1 AND t>125)
811 PRINT #0;"Druk een toets voor het vervolg."
812 BORDER 2: BORDER 7: BORDER 1: BORDER 0: PAUSE 1: BORDER 0:
    GO TO 812+( INKEY$ <> "")
850 RUN
900 STOP
999 SAVE *1;"SGGreclame" LINE 850

```



## MC STAP VOOR STAP

(C) J. v. Alteren, SGG; 210191.

Uit ZX-Computing afl.: apr/mei '85, blz. 88 e.v.  
door David Nowotnik

```

1001101011011
1001101011011
11100110101101
1001101011011
1001101011011
1001101011011
1001101011011
1001101011011

```

8e aflevering:

## DEEL 3: Laden van blokken: (Vervolg)

Het laatste stuk theorie in dit gedeelte is het krachtige blok laad-opdrachten van de Z80. Het maakt het mogelijk, om een blokje Bytes (max 65535) in het geheugen te verplaatsen naar een ander gedeelte.

Het werkt als volgt:

Het startadres van het oorspronkelijke blok wordt geladen in het HL-registerpaar, het startadres van het blok waar het naar toe moet worden verplaatst gaat naar DE, en het aantal Bytes dat moet worden verplaatst, gaat naar BC. Daarna maakt de opdracht LDIR maakt de transactie compleet.

LDIR betekent: Load Increment Repeat

De waarde in BC werkt als een telmachine. De byte-waarde in het HL-register wordt overgeplaatst naar de Byte met adres DE; HL en DE worden hoger terwijl BC lager wordt. Daarna herhaalt deze opdracht zichzelf automatisch totdat BC nul wordt.

Er is een niet-repeterend alternatief LDI, waarbij het verplaatsen van de bytes plaatsvindt, de HL en DE adressen hoger worden, BC afneemt, maar waar de opdracht niet automatisch wordt herhaald.

Beginners met mc gebruiken meestal LDIR en dit wordt ook in de voorbeelden gedaan.

De Z80 heeft een bijna identieke opdracht genoemd LDDR. Om het te gebruiken worden de END-adressen van het oorspronkelijke blok en het adres van het blok waar het naar toe moet respectievelijk in HL en DE geladen, terwijl BC het aantal bytes nog vasthoudt dat moet worden verplaatst.

Elke cyclus opdrachten bevat verplaatsing van HL naar DE en alle drie registerparen die verlaagd zijn.

LDD in de niet-automatisch repeterende versie van LDDR.

Het doet er echt niet toe of U LDIR of LDDR gebruikt, behalve als er enige overlap tussen de oorspronkelijke en toekomstige geheugenblokken is.

Als dat zo is, komt het erop aan of de bytes elkaar niet 'opeten'.

Ik zal eindigen met die te selecteren die gebruikt moeten worden, afhankelijk of de overlap aan het begin of aan het einde van het oorspronkelijke blok zit.



# BULLETIN SGG

Fig. 2: Demonstratie SPECTRUM blok-verplaatsing.

Tik de hierna volgende listing in, SAVE hem en RUN het daarna. Nadat de mc boven de RAMtop geplaatst is (regel 10 - 80) wordt er een scherm-afdruk gemaakt, weggezet en daarna wordt het scherm schoongemaakt en komt er weer een nieuwe afbeelding via een ander blokje mc. Probeer ook hier weer de ASS.-Listing, die hierna staat, te betrekken.

Hieronder de HEX-loader SPECTRUM.

```

10 CLEAR 24999
20 FOR i=25000 TO 25023
30 READ x: POKE i,x
40 NEXT i
50 DATA 33,0,64,17,112,98
60 DATA 1,0,27,237,176,201
70 DATA 33,112,98,98,17,0,64
80 DATA 1,0,27,237,176,201
90 CLS
100 FOR i=1 TO 20
110 PRINT INK i/3;PAPER 8-i/3;" 111122223333444455556666777788"
120 NEXT i
130 RANDOMIZE USR 25000
140 CLS
150 PRINT AT 12,0;"Press any key "
160 IF INKEY$="" THEN GO TO 160
170 RANDOMIZE USR 25012
200 STOP
210 SAVE "Block move": PAUSE 50: VERIFY ""

```

Bijbehorende assambly listing:

25000	LD HL,16384	2100040	Begin van de schrem-file
	LD DE,25200	117062	Begin van het 'opberglaadje'
	LD BC,6912	01001B	Aantal verplaatste bytes
	LDIR	EDB0	Het verplaatsen en
	RET	C9	RETURN TO BASIC
25012	LD HL,25200	217062	Begin van het 'opberglaadje'
	LD DE,16384	110040	Begin van de scherm-file
	LD BC,6912	01001B	Aantal verplaatste bytes
	LDIR	EDB	Het verplaatsen en
	RET	C9	RETURN TO BASIC

Zo, dit was het voor deze keer. Als U er tot zover doorgekomen bent, lukt de rest de volgende keer ook, sterkte deze keer.

J. v. Alteren.

Noot van de redactie:

Dit artikel is al een deel van de tekst die in het bulletin van maart 1990 stond. Toendertijds is het niet goed overgekomen, in deze gewijzigde vorm hopen we dat het duidelijker is. Het overige deel van het originele artikel zal waarschijnlijk in een volgend bulletin verschijnen.

# BULLETIN SGG

Het voorbeeld geeft gelijke opdrachten om blokken te verplaatsen op zowel de ZX81 als de Spectrum; terwijl de afbeeld-file naar een veilige plaats in het RAM-geheugen wordt weggezet. De verschillen tussen de ZX81 en de Spectrum geven een verschillende uitwerking van de codes, dus de ZX81-gebruikers kijken naar fig. 1 terwijl de Spectrum-bezitters figuur 2 nemen.

De volgende keer krijgt U te maken met 'uitgangen' en enige 'bit'-opdrachten.

Fig. 1: Demonstratie ZX81 Blok-verplaatsing.

Tik de volgende listing in en SAVE haar daarna. Voordat je het programma RUNT, moet de RAMtop verlaagd worden, d.m.v. twee commando's (in de DIRECT-mode):POKE 16388,231 en POKE 16389,124.

De hexloader stuurt de mc die hierna volgt naar het R.A.M. Er wordt dan een beeld gevuld op het scherm en weggezet op een veilige plaats in het RAM. Het scherm wordt daarna weer schoongeveegd, en direct weer gevuld met een volgend mc-blokje in regel 180.

Hieronder de HEX-loader ZX81.

```

10 REM ..... > 34 SPATIES TOTAAL INTIKKEN < .....
20 LET X = 16514
30 LET A$="2A0C4011E77C011903EDB0C921E77CED5BOC40011903EDB0C9"
40 FOR I=1 TO LEN A$/2
50 LET J=16*(CODE A$-28)+CODE A$(2)-28
60 IF PEEK X=27 THEN POKE X,J
70 LET X=X+1L
80 LET A$=A$(3 TO)
90 NEXT I
100 CLS
110 FOR I=1 TO 20
120 PRINT "1234567890ABCDEFGHI"
130 NEXT I
140 RAND USR 16514
150 CLS
160 PRINT AT 20,0;"PRESS A KEY TO RESTORE DISPLAY"
170 IF INKEY$="" THEN GOTO 170
180 RAND USR 16526

```

Bijbehorende assambly listing:

SAVE	LD HL, (16396)	2A0C40	Laad de D-file in HL
	LD DE, 31975	11E77C	Laad de RAMTOP in DE
	LD BC, 793	011903	De lengte van de schermfile
	LDIR	EDB0	laad het blok
	RET	C9	RETURN naar BASIC
RESTORE	LD HL, 31975	21E77C	Laad de RAMTOP in HL
	LD DE, (16396)	ED5BOC40	Laad de D-file in DE
	LD BC, 793	011903	lengte van de schermfile
	LDIR	EDB0	laad het blok
	RET	C9	RETURNto BASIC



## ZICHTBAAR SORTEREN



De werking van een snelle sorteer-routine op het scherm afgebeeld.

Het sorteren van reeksen data is een karwei dat bij uitstek geschikt is om door computers verricht te worden. Geen wonder dat er dan ook een groot aantal verschillende sorteermethoden bedacht zijn. De bekendste is wel de z.g.n. Bubble-sort methode, deze heeft als voordeel z'n eenvoud, maar een groot nadeel is dat het nogal langzaam werkt.

In het hieronder staande programma wordt een string gesorteerd volgens de Shell-Metzner methode. Deze methode zoekt op een zeer speciale wijze naar de elementen binnen de string die verwisseld moeten worden, om zodoende het aantal benodigde 'tests' zo klein mogelijk te houden.

Door een extra regel (4052) heb ik geprobeerd de wijze waarop er getest (r.4050) en verwisseld (r.4060) wordt op het scherm zichtbaar te maken.

Roelof Koning.

```

10 CLS : PRINT "Even geduld...": DIM a$(64)
20 FOR f=1 TO 64: LET a$(f)=CHR$ (33+INT (RND*90)): NEXT f
30 CLS : PRINT a$'"Shell-Metzner sort is bezig"
4000 REM sortroutine
4010 LET la=LEN a$: LET m=la
4020 LET m=INT (m/2): IF m=0 THEN PRINT'"Klaar!"'"a$: STOP
4030 LET j=1: LET k=la-m
4040 LET i=j
4050 LET h=i+m: IF a$(i)<a$(h) THEN GOTO 4080
4052 PAUSE 5: PRINT AT 7,0;a$( TO i-1); PAPER 4;a$(i); PAPER 7;
      a$(i+1 TO h-1); PAPER 5;a$(h); PAPER 7;a$(h+1 TO )
4060 LET t$a$(i): LET a$(i)=a$(h): LET a$(h)=a$(t): LET i=i-m:
      IF i<1 THEN GOTO 4080
4070 GOTO 4050
4080 LET j=j+1: IF j>k THEN GOTO 4020
4090 GOTO 4040

```

## ESGEEGEETJES

Te koop aangeboden:

Spectrum +3 met ingebouwde diskdrive en seriële printeraansluiting. Met multiface +3 en circa 20 3" disks, inclusief software en boeken. Vaste prijs f 500,- Apple Scribe serielële printer tegen meerprijs erbij. Bel 050-410567.

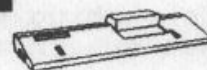
Gezocht:

Voor de Multiface 128 het kraakprogramma GENIE, of aanpassingen van de 48K versie voor deze. Bel 050-185054.

## IF1 OP DE DISCIPLE

IF 1

&



Het gebruik van de Interface 1 hookcodes op de DISCIPLE, deel 1. Door Tonnie Stap.

Een van de 'features' van de DISCIPLE is dat hij de microdrive hookcodes ondersteund. In dit artikel wil ik in gaan op wat dat precies inhoudt en wat de beperkingen zijn, later in de serie wil ik ingaan op het lezen van microdrive files en het vertalen hiervan in normale files.

In deze serie ga ik er hierbij van uit dat u tenminste een beetje kennis heeft van machinetaal en het liefst ook van de DISCIPLE command codes. Voor het eerste verwijs ik u naar de serie over machinetaal van dhr. van Alteren en wat betreft het tweede staat er op pagina 59 van de originele DISCIPLE handleiding een voorbeeld programma. Uit de (onuitputtelijke) DISCIPLE Nieuwsbrief kunt u tenslotte informatie halen over beide onderwerpen.

De reden voor het bestaan van de hookcodes (als ik spreek over hookcodes bedoel ik zowel die van het Interface 1 als de command codes van de DISCIPLE) is het verlangen om de SPECTRUM uit te kunnen breiden. Zo wel men graag snellere gegevensopslag, printen op een echter printer en computers verbinden met een netwerk. Echter, de standaard SPECTRUM kan dat niet en de ROM zit al zo goed als vol. Nu zijn er interfaces die de SPECTRUM RAM gebruiken voor het programma dat deze randapparatuur aanspreekt maar ideaal is dat niet. Ook zijn er oplossingen mogelijk zoals bij de Beta Disk en die komt al aardig in de buurt van de door Sinclair en Miles Gordon gebruikte.

Een diskcommando levert op de 'kale' SPECTRUM immers een foutmelding op, bij elke van deze meldingen wordt een RST 8 uitgevoerd. Deze instructie is een korter alternatief voor CALL 8, er bestaan 8 an deze instructies voor de adressen 0,8,16, ..,56. In genoemde interfaces zit nu een stukje logica dat ingrijpt op het moment dat de instructie op adres 8 wordt opgehaald. Dan wordt namelijk het SPECTRUM ROM geheugen vervangen door het geheugen van het interface.

Na de RST 8 staat gewoonlijk het nummer van de foutmelding, je kunt er echter ook een hookcode neerzetten, een hookcode is immers een gewoon getal. Zoals u wel zult begrijpen kan dit alleen vanuit machinetaal, die moet u schrijven of (als onderdeel van commerciële programma's) kopen. Op adres 8 neemt het interface de controle over, de Z80 krijgt zijn instructies uit het nu 'ingepagede' geheugen. Hier staat een routine die precies uitzoekt waarom de RST 8 werd uitgevoerd, was het bijvoorbeeld een gewone foutmelding of een foutmelding veroorzaakt door een interface commando (zoals CAT 1). Of was het een hookcode, of printen met het interface etc etc.



# BULLETIN SGG

Laten we er van uit gaan dat het een hookcode was, dat is voor de routine makkelijk af te leiden want die hebben waarden van 27 t/m 52 (IF1) of 51 t/m 71 (DISCiPLE). Een overzicht hiervan zal ik in een toekomstige aflevering geven. In een tabel staat voor elk van deze hookcodes een adres, dit is het adres van de routine waar de hookcode afgehandeld wordt. In de DISCiPLE zijn er twee van die tabellen, één voor de command codes en één voor de IF1 hookcodes.

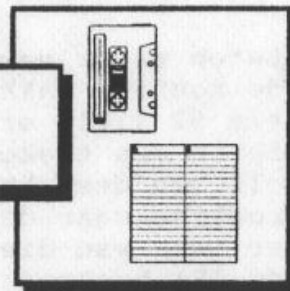
Stel nu dat u een programma heeft dat gebruik maakt van de IF1 hookcodes, als het programma dan wil SAVEN op microdrive wordt eerst een hookcode uitgevoerd die aangeeft dat er een file met de gespecificeerde naam moet worden geladen. Dit lijkt vreemd maar is erg handig want als er al een file bestaat met die naam moet die eventueel worden gewist. Mislukt dit 'openen' van de leesfile dan wordt er een hookcode uitgevoerd om de gemaakte rotzooi op te ruimen en één om een file te schrijven. Er is tot nu toe geen byte op cartridge of disk terecht gekomen maar dat gaat nu veranderen. Het programma verplaatst steeds 512 bytes naar in buffer in het normale SPECTRUM geheugen en geeft dan via een hookcode de opdracht om deze 'sector' weg te schrijven. Zijn alle bytes geschreven dan volgt de hookcode om het file af te sluiten, d.w.z. dat alle relevante informatie over de file in de directory wordt gezet en de gemaakte rotzooi opgeruimd.

Tot zover een versimpelde uitleg van hoe de hookcodes gebruikt worden, voor de al verdr georderde lezer ik wil nog iets kwijt over de beperkingen van hookcodes op de DISCiPLE. Door verschillen tussen cartridges en disks zijn namelijk niet alle hookcodes relevant, verder heeft de DISCiPLE geen RS232 poort en is het netwerk iets anders. De RS232 codes (29 en 30) werken dus niet, net zoals de codes om direkt sectoren te lezen en te schrijven (40, 41 en 42). Zoals al gezegd werkt ook het netwerk niet (hookcodes 45 t/m 48) en een CALL uitvoeren in de IF1 ROM is ook onmogelijk (nr 50). Dit laatste is omdat de routines in de DROM op andere adressen staan dus dat loopt van geen kant. De hookcodes 51 en 52 werken allen op Issue 2 van het IF1 en worden overschreven door de eerste twee DISCiPLE command codes. Geen groot verlies want de eerste haalt een 'beschrijving' op van het 'record' en de tweede werkt met de RS232 poort.

Tot zover de eerste aflevering van 'IF1 op de DISCiPLE', ik hoop deze serie nog een half jaartje vol te kunnen babbelen en vertrouw daarbij op u voor vragen en reacties. Is er dus iets onduidelijk, zit u met een probleem, aarzel niet en vraag 't, op de bijeenkomsten of over de post, alles is welkom!

## SKIPLOAD

Een programma om lange files te kraken.  
Door Tonnie Stap.



Lang, lang (of toch niet zo lang) geleden werkte iedereen met tapes. Toen ik net bij de club kwam (in 1987) had een enkeling iets anders, zelf had ik al eens mogen ruiken aan de Rotronics Wafadrive en dat was al heel wat. De OPUS was er (geloof ik) al wel maar de DISCiPLE nog lang niet.

Maar genoeg over disks en dergelijke, op tape had je soms lange files die in één keer het geheugen vulden, kopiëren kon dan alleen met dubbele cassettedeks o.i.d. Daarom schreef ik destijds het programma SKIPLOAD. Wat doet het programma, heel eenvoudig, van een file op tape kunnen willekeurige stukken geladen worden. Bijvoorbeeld van een 48K file alleen het screen en de UDG's of van een screen het eerste derde deel en het derde zoals in de demo. Voor u de MC aanroept zet u in a\$ achtereenvolgens een CHR\$ 0 om te beginnen met overslaan, en iets anders om te laden, dan twee bytes voor het startadres (lage eerst) en vervolgens de lengtes van de te laden blokken (steeds twee bytes). De laatste lengte moet nul zijn. Veel plezier!

Demonstratie programma:

```

10 DATA 1,16384,2048,2048,2048,768
20 DATA 0
30 READ a
40 LET a$=CHR$ a
50 READ a
60 LET a$=a$+CHR$ (a-256*INT (a/256))+CHR$ INT (a/256)
70 IF a<>0 THEN GO TO 50
80 CLS
90 PRINT "Start de tape en druk op een toets na de header."
100 PAUSE 0: CLS
110 LET l=USR 65000
120 PAUSE 0: CLS
130 IF NOT l THEN PRINT "Waarschuwing: Tape loading error"
999 STOP
8000 CLEAR 64999
8010 LOAD d*"SKLOAD.COD"CODE 65000
8020 CLS
8030 PRINT '"Type RUN om een demonstratie te zien, doe daarvoor
een bandje met een SCREEN$ in de recorder."'
8040 PRINT '"Verander de DATA in regel 10 om te experimenteren."'
8050 PRINT '"Het eerste getal is 0 om te beginnen met oversl
aan."'
8060 PRINT '"Het tweede nummer is het laad adres."'
8070 PRINT '"Dan volgen de lengtes van de te laden en over te sl
agen blokken."'
8080 PRINT '"De eindmarkering is een nul (regel 10)."'

```



# BULLETIN 566

```

10 *L-
20 *D+
30 *C-
40 ;This routine loads some
50 ;parts of a file from
60 ;tape at a given address.
70 ;
80 ;A$(1) contains 0 to skip
90 ;and any other value to
100 ;load the first part.
110 ;
120 ;The load address has to
130 ;be put in A$(2 TO 3).
140 ;
150 ;The length of the blocks
160 ;has to put in A$(4 TO ),
170 ;low byte first.
180 ;
190 ;The string has to be
200 ;terminated with a zero
210 ;block length.
220 ;
230 CH_ADD EQU 23645
240 ORG 65000
250 *L+
260 DI
270 LD HL, (CH_ADD)
280 ;First get address of A$.
290 PUSH HL
300 LD HL, CHAR
310 LD (CH_ADD), HL
320 CALL #28B2
330 EX (SP), HL
340 ;Restore CH_ADD.
350 LD (CH_ADD), HL
360 ;If C then A$ not found.
370 JR NC, FOUND
380 EI
390 RST 8
400 DEFB 1
410 ;Put IY to 2nd word.
420 FOUND POP IY
430 LD BC, 8
440 ADD IY, BC
450 ;
460 ;Get indicator for skip
470 ;or load first.
480 ;
490 ;255 is added to it, if
500 ;it was zero this gives
510 ;NC i.e. skip.
520 ;
530 ;If it wasn't it gives
540 ;a carry i.e. load.
550
560 LD A, (IY-5)

```

# BULLETIN 566

```

570      ADD    A,255
580 ;Get first length in DE.
590      LD     E,(IY-2)
600      LD     D,(IY-1)
610 ;Get start address in HL.
620      LD     H,(IY-3)
630      LD     L,(IY-4)
640      PUSH   HL
650      POP    IX
660      LD     A,255
670      CALL   SKLOAD
680      LD     BC,0
690      RET     NC
700 ;NC means tape error.
710      INC    C
720 END     RET
730 CHAR    DEFM "A$ "
740 *L-
750 ;This routine loads a
760 ;file from tape skipping
770 ;parts if desired.
780 ;
790 ;It is to be called with
800 ;the following constants:
810 ;
820 ;IX= Start address.
830 ;DE= Length 1st part.
840 ;C   If loading 1st part.
850 ;NC  If skipping 1st.
860 ;
870 ;IY Points to a block of
880 ;words giving the lengths
890 ;of the blocks to be
900 ;loaded or skipped.
910 ;
920 ;This routine is almost a
930 ;copy of the ROM routine
940 ;at #0556.
950 ;
960 *L+
970 SKLOAD INC    D
980      EX      AF,AF'
990      DEC     D
1000     DI
1010     LD      A,#0F
1020     OUT     (#FE),A
1030     LD      HL,LD_RET
1040     PUSH    HL
1050     IN      A,(#FE)
1060     RRA
1070     AND     #20
1080     OR      #02
1090     LD      C,A
1100     CP      A
1110 L056B RET     NZ
1120 L056C CALL   #05E7

```



# BULLETIN 566

```

1130      JR    NC,L056B
1140      LD    HL,#0415
1150 L0574  DJNZ L0574
1160      DEC   HL
1170      LD    A,H
1180      OR    L
1190      JR    NZ,L0574
1200      CALL  #05E3
1210      JR    NC,L056B
1220 L0580  LD    B,#9C
1230      CALL  #05E3
1240      JR    NC,L056B
1250      LD    A,#C6
1260      CP    B
1270      JR    NC,L056C
1280      INC   H
1290      JR    NZ,L0580
1300 L058F  LD    B,#C9
1310      CALL  #05E7
1320      JR    NC,L056B
1330      LD    A,B
1340      CP    #D4
1350      JR    NC,L058F
1360      CALL  #05E7
1370      RET   NC
1380      LD    A,C
1390      XOR   #03
1400      LD    C,A
1410      LD    H,#00
1420      LD    B,#B0
1430      JR    L05C8
1440 L05A9  EX    AF,AF'
1450      JR    NZ,L05B3
1460 ;Jump to here for a new
1470 ;block to win some time.
1480 NWBLCK JR    NC,L05BD
1490      LD    (IX+0),L
1500      JR    L05C2
1510 L05B3  RL    C
1520      XOR   L
1530      RET   NZ
1540      LD    A,C
1550      RRA
1560      LD    C,A
1570      INC   DE
1580      JR    L05C4
1590 ;
1600 ; Replace verify with
1610 ; instruction for skip
1620 ; and two dummies for the
1630 ; timing.
1640 ;
1650 L05BD  DEC   IX
1660      LD    R,A
1670      LD    R,A
1680 ;Continue as useall.

```

# BULLETIN SGG

```

1690 L05C2 INC IX
1700 L05C4 DEC DE
1710 EX AF,AF'
1720 LD B,#B2
1730 L05C8 LD L,#01
1740 L05CA CALL #05E3
1750 RET NC
1760 LD A,#CB
1770 CP B
1780 RL L
1790 LD B,#B0
1800 JP NC,L05CA
1810 LD A,H
1820 XOR L
1830 LD H,A
1840 LD A,D
1850 OR E
1860 JR NZ,L05A9
1870 ;
1880 ; Now fetch length of
1890 ; next block to be loaded
1900 ; or skipped. If this is
1910 ; zero then that's all.
1920 ;
1930 LD E,(IY+0)
1940 LD D,(IY+1)
1950 LD A,E
1960 OR D
1970 INC IY
1980 INC IY
1990 JR Z,LD_END
2000 ;Invert skip/load flag.
2010 EX AF,AF'
2020 CCF
2030 JP NWBLCK
2040 ;Continue as useall.
2050 LD_END LD A,H
2060 CP #01
2070 RET
2080 ;Restore IY and jump to
2090 ;normal return address.
2100 LD_RET LD IY,#5C3A
2110 JP #053F
2120 ;
2130 ;This routine loads a
2140 ;header into the
2150 ;printerbuffer.
2160 ;
2170 HEADLD LD IX,23296
2180 LD DE,17
2190 XOR A
2200 SCF
2210 CALL #0556
2220 ;In case of error, retry.
2230 JR NC,HEADLD
2240 RET

```



# BULLETIN 566

```

2250 ;
2260 ;This routine saves a
2270 ;block of bytes.
2280 ;
2290 ;The header information
2300 ;is in B$(1 TO 17) and
2310 ;the address of the bytes
2320 ;in B$(18 TO 19).
2330 ;
2340 ;The routine is almost a
2350 ;copy of the ROM routine
2360 ;at #0970.
2370 ;
2380         LD     HL, (CH_ADD)
2390 ;First get address of B$.
2400         PUSH  HL
2410         LD     HL, CHARB
2420         LD     (CH_ADD), HL
2430         CALL  #28B2
2440         EX     (SP), HL
2450 ;Restore CH_ADD.
2460         LD     (CH_ADD), HL
2470 ;If C then B$ not found.
2480         JR     NC, SFOUND
2490         RST     8
2500         DEFB  1
2510 SFOUND POP  IX
2520 ;Let IX point to B$(1).
2530         LD     BC, 3
2540         ADD    IX, BC
2550         LD     A, #FD
2560         CALL  #1601
2570         XOR     A
2580         LD     DE, #09A1
2590         CALL  #0C0A
2600         SET    5, (IX+2)
2610         CALL  #15D4
2620         PUSH   IX
2630         LD     DE, 17
2640         XOR     A
2650         CALL  #04C2
2660         POP    IX
2670         LD     B, #32
2680 WAIT1S HALT
2690         DJNZ  WAIT1S
2700         LD     E, (IX+11)
2710         LD     D, (IX+12)
2720         LD     L, (IX+17)
2730         LD     H, (IX+18)
2740         PUSH   HL
2750         POP    IX
2760         LD     A, #FF
2770         JP     #04C2
2780 CHARB  DEFM  "B$ "

```

## DRIVE DE WEG KWIJT



Door Rudy Biesma

De DISCiPLers onder ons hebben de foutmelding **FORMAT data lost** soms tot hun afgrijzen weleens gezien. De DISCiPLE laat hiermee weten dat er niets zinnigs (geen sectoren laat staan data) op de huidige track te vinden is.

Als dat op een lege schijf gebeurt dan is er niet zoveel aan de hand, maar owee als die foutmelding juist optrad terwijl je een belangrijk programma aan het laden was. Nog veel erger is het als de foutmelding optreedt als de kop van de drive op de, blijkbaar wijlen, CAtalogus staat.

In het tweede geval, als CAT \* het nog normaal doet, ligt (hopelijk) alleen het betreffende file in de vernieling, maar als de fout tijdens het "CAtten" optreedt dan zijn er veel meer files min of meer verloren.

Meestal valt er niets meer te redden van datgene dat op de "FORMAT data lost"-track staat. Alleen als na CAT \* de 'CAT-boodschap' en een aantal seconden later de "FORMAT data lost"-melding verschijnt, is er nog een sprankje hoop. Het kan dan namelijk zijn dat de drive, en daarmee de DISCiPLE, gedesorienteerd is. Dit is heel lastig als je net je systeem file wilde laden, de enige oplossing was dan het opnieuw, van kassette, initialiseren van de DISCiPLE en een schijf formatteren. Vooral de "Data-Skip drives" kan dit nog weleens overkomen als de steprate onder de 6ms wordt gezet. (POKE @3,n waarbij 0<=n<6)

DE OPLOSSING is vrij eenvoudig en werkt ook als er geen systeem aanwezig is (het werkt zelfs als de DISCiPLE mbv de inhibitknop uitgezet is). Voer met de hand de volgende twee regeltjes na elkaar in (na elke regel 'ENTER' geven).

OUT 219,10

OUT 27,16

Met deze twee kommando's wordt de Floppy Disk Controller in de DISCiPLE rechtstreeks "opgedragen" om de kop van de drive naar track 10 te verplaatsen. Als nu CAT of RUN gegeven wordt moet alles weer normaal functioneren anders is er werkelijk een fout op de schijf.

De de drive raakt gedesorienteerd doordat de kop blijkbaar voor track 0 terecht komt ("track -1"), maar dat de drive toch signaleert dat hij op track 0 staat. Omdat de DISCiPLE denkt dat de drive op de juiste plaats staat, wordt er geprobeerd te lezen. Dit gaat mis omdat track -1 niet geformatteerd is.

Het probleem is te voorkomen door na het laden van elk "vreemd" systeem eerst POKE @3,'juiste waarde' te geven.



## BESTE MAATJES MET SCREEN-DUMPS

Door: Johan Koning.

Er zijn vele maten screendumps net als vele maten papier. Een klein aantal heb ik hier onder op een rij gezet.

	papiermaat br x ho in mm	afmetingen dump ho x br in mm	voorkeur dump naam(prog)/ Bulletin ed.
A6 :	105 x 149	105 x 160	Greyscale(uit Kopprint)
A7 :	75 x 105	65 x 105	maart'90 blz:9 (reloc.)
A8 :	53 x 75	? x ?	mij niet bekend.
A9 :	38 x 53	38 x 53	Kleinedump (Kopprint)

LET WEL ! Bij papier is de hoogte groter dan de breedte, doch bij de hiergenoemde dumps is dat net andersom.

DUS print zo'n dump op minimaal 1 maat groter papier.

En dan nu het BELANGRIJKSTE:

WIE voelt zich in staat om een programma te maken waarmee naar

\*\*\*\*\*

GEVRAAGD: Programmeer-pokes. Hier mee bedoel ik pokes zoals :

POKE 23756,0 waarmee je de 1e. regel van een BASIC-programma in regel 0 veranderd of POKE 23658,8 waar mee je alle INPUT automatisch in HOOFD LETTERS veranderd.

Dus pokes die je helpen bij het programmeren in BASIC.

Ook andere TIPS voor het programmeren in BASIC zijn van harte welkom. Stuur ze naar de REDACTIE, zie adres in het COLOFON.

Als er genoeg binnen komen wordt hier van een MINIBOEKJE gemaakt dat in het maartnummer van het BULLETIN zal zitten.

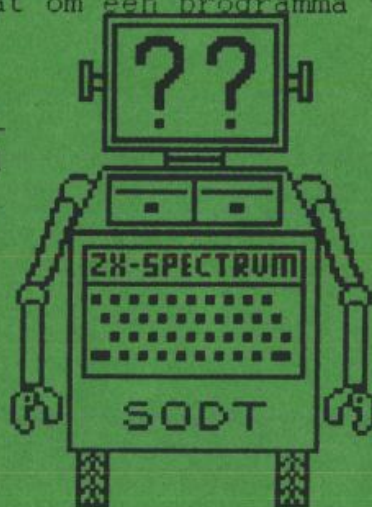
Anders worden de wel binnengekomen pokes en tips in de SODT-rubriek in het zelfde nummer vermeld.

Vermeldt bij de pokes die je instuurt even wat hij doet en eventueel ook met welke poke je hem weer opheft.

Insturen voor 1 januari 1992.

Bij voorbaat dank.

Johan.



keuze in een van deze afmetingen een SCREEN\$ geprint kan worden. OF WIE levert mij een of meerdere van deze of hier mee te vergelijken screendumps. Maar dan wel graag als een R E L O C A T A B L E machinecodeprogramma. Zodat ik ze in een programma kan samenvoegen. Johan.

\*\*\*\*\*



DRUKWERK

C.M. Ballintijn  
B. Boermalaan 7  
9765 AP Paterswolde

PORT BETAALD  
GRONINGEN