

# Wetenschappelijk Tijdschrift voor de Geschiedenis van de Nederlandsche Indië

**JAARGANG : 9**

**KWARTAAL : 1**

Verslijnt 4 maal per jaar.  
Losse verkoopprijs f 7,-.



"Sinclair Impuls", HET blad voor en door de gebruikers van ALLE Sinclaircomputers - ZX80, ZX81, ZX Spectrum, QL en aanverwanten - wordt uitgegeven door de "HCC Sinclair Gebruikers Groep" (SGG).

IMPULSREDACTIE:

Ed Weijgers  
H Marsmanlaan 29  
2624 TJ Delft

Kees Versluis  
Copernicuslaan 25  
2561 VA Den Haag

Jack Raats (eindredactie)  
(SGG-Infotel 01670-66845)

Noorddonk 107  
4651 ZD Steenbergen

IMPULSKOPIJ:

Voor OD naar Kees Versluis (OD-hulp: 070-3604185),  
alle andere kopij naar Ed Weijgers of Jack Raats.

IMPULSABONNEMENTEN:

Alleen voor HCC-SGG-leden f 25.00 per kalenderjaar  
(voor anderen f 30.00), over te maken op onze SGG-  
postrekening, ovv 'Abonnement Impuls 1991' (al ver-  
schenen nummers ontvangt u bij het volgende nummer).  
Adreswijzigingen: schriftelijk naar het SGG-adres.

SGG-ADRES EN -REKENING:

HCC Sinclair GG  
Postbus 76  
2260 AB Leidschendam

Postrekening 5374525  
tnv HCC Sinclair GG  
te Bunnik

SGG-TELEFOONNUMMER:

Infotelefoon 01670-66845

ma en do, 20-22 uur

SGG-databank "SINCLAIR BOX"  
300 1200 2400 1200/75 baud

dagelijks 23-6.00 uur  
za en zo vanaf 20 uur

Gebruik dit telefoonnummer uitsluitend bij problemen  
met hard- en software of voor de algemene informatie  
over onze SGG, Sinclaircomputers en Impulsartikelen.

SGG-ARTIKELEN BESTELLEN:

HCC-bestelservice  
Infotel 03403-78788

postbus 149  
3990 DC Houten

DUCBANK (OPUS DISCOVERY)  
via onze SGG-rekening  
bankier Kees Versluis  
Infotel 070-3604185

DD-BANK (DISCIPLE)  
via onze SGG-rekening  
bankier Rob Willig  
Infotel 02159-43087



In dit nummer kunnen we u onze nieuwe DD-programmabank voorstellen, staan gelukkig weer artikeltjes voor de QL en onze laatste P-omzetter, die voor de BD. Voorts het beloofde overzicht van de programma's uit IMPULS 61 t/m 84, die als P-files te downloaden zijn bij onze SINCLAIR BOX en te koop op IMPULSOFT-cassette 15, DUCDISK-21 en DDISC-21 (balieverkoop en de normale bestelwijze).

De nieuwste XCOM-programma's, versies .06, werken nog beter bij 2400 baud en hoger. Zoals beloofd zijn ze aangevuld met XCOM7 en XCOM8, voor het normale en het gemodificeerde VTX711-interface. Ze staan op DUCDISK-21 en DDISC-21, en op IMPULSOFT-cassette 16.

Voor OD-ers is er "dd>od" om DDiskettes te lezen en - u zag daar waarschijnlijk al verlangend naar uit - die 'supersnelle' OD-ROM is eindelijk niet alleen klaar, maar buitendien nog te koop ook!

Marcel van Dongen had indertijd een routine geschreven om alleen CODE vanaf speciaal geFORMATte disks zeer snel te kunnen LOADen. Sake Buwalda is bezig geweest om meer routines in ROM te bakken. Victor Vogelpoel heeft dat werk, een flink karwei, nu voltooid. Hij breidde het DOS van ROM 2.2 uit tot QUICKDOS in een ROM 2.3, met snelle routines voor zowel SAVE, LOAD als MERGE, bij zowel BASIC, CODE als ARRAYS. Het drivenummer wordt in 23728 gezet, nu ook bij RAM-disk (zonder 'last drive'-routine in het ankerblok). Zou er software zijn die nu niet meer werkt: QUICKDOS is in- en uitschakelbaar met de opdracht FORMAT "Q",s (s=1: on, s=0: off).

tijd in seconden	quickdisk		normdisk	
	on	off	on	off
SAVE	5	45	30	32
LOAD	4	43	29	31

quickdisk: disk onder QUICKDOS geFORMAT met interleave 1, skew 13  
normdisk: normaal met interleave 13, skew 5

Dit zijn testresultaten met files van 50 KByte. Hoewel het geen uitvoerige tests waren geven ze toch aan dat het SAVEN en LOADen nu gemiddeld ZEVEN KEER ZO SNEL gaat als vroeger! Bovendien gaat het met QUICKDOS zelfs bij normdisks nog een tikkeltje vlugger. Zonder QUICKROM gaat het bij quickdisks ook nog, maar langzamer. Een REFORMAT-programma om van N-disks Q-disks te maken komt nog.

Bestellen kan alleen door gireren op ons postbanknummer 5374525, tnv HCC SINCLAIR GG te Bunnik, o/v QUICKROM. Prijs slechts f 25, inclusief de bekende update-service van de DUCBANK. Komt er een nieuwe versie: opsturen naar Kees met gefrankeerde retouromslag. Wij verwachten dat wij deze chips voor 18 mei kunnen verzenden. Dan moet hij ook te koop zijn in Houten, waar altijd deskundigen aanwezig zijn om hem in te bouwen, als u dat zelf niet aandurft. Dat is overigens doodsimpel: oude uit het voetje en nieuwe erin.

In IMPULS 61-54 en 63-31 is reeds over dit onderwerp geschreven, maar dat is al lang geleden en tbv "dd>od" elders in dit nummer moest deze informatie uitgebreid worden. Voor DD- en OD-ers dus.

#### INDELING VAN EEN DDISK MET 1 OF 2 KANTEN EN 40 OF 80 TRACKS

B	0 1 2 ... 9 10 .....	390 .... 399 .... 790 .... 799
T	0 .....	1 ..... 39 ..... 79 .....
KANT 1	<u>1 2 3 ... 10 1 2 3 .....</u>	<u>40 tracks 80 tracks</u>
S	1 2 3 ... 10 1 2 3 .....	1 2 3 ... 10 .... 1 2 3 ... 10
KANT 2	<u>128 .....</u>	<u>129 ..... 167 ..... 207 .....</u>
T	128 .....	129 ..... 167 ..... 207 .....
B	800 .... 809 810 .....	1190 .. 1199 .... 1590 .. 1599
B	400 .... 409 410 .....	790 .... 799 <-- bij 40 tracks

We nemen aan dat er niemand meer SD met 256 bytes/blok gebruikt. S is het sector-, T het track- en B het bloknummer (voor de OD):

$$B = T \cdot 10 + S - 1 - \begin{cases} / (880 \text{ AND } T \geq 40) & / (881 \text{ OR } T < 40) \\ \backslash (480 \text{ AND } T \geq 80) & \backslash (481 \text{ OR } T < 80) \end{cases}$$

korter:  $B = T \cdot 10 + S - \begin{cases} / (881 \text{ OR } T < 40) \\ \backslash (481 \text{ OR } T < 80) \end{cases}$

Als  $N = (128 - \text{tracksaantal}) \cdot 10 + 1$  dan:  $B = T \cdot 10 + S - (N \text{ OR } T < 80)$

#### DE INSTELLING VAN DE OD-TABEL VOOR EEN DDISK

- 40 of 80 tracks/kant
- 10 blokken/track
- 512 bytes/blok
- geen ankerblok
- sector offset 1
- dubbelzijdig

Een enkelzijdige disk bevat toch geen tracknummers boven de 79.

DE DIRECTORY: T=0 TO 3, DUS 4\*10 BLOKKEN VAN 512 BYTES: 20KB  
PER FILE: 1/2 BLOK, DUS VOOR MAXIMAAL 80 FILES

001	Directory Description (DD-type 00 t/m 11)
002-011	Filenaam
012 013	Aantal der fileblokken (Hoog voor Laag)
014 015	Track- en Sectornummer van het eerste fileblok
016-210	Bitmap (zie de genoemde artikelen)
211	64K-vouden van de filelengte (0 bij Header-files)

Bij DD-type 01 t/m 04, 07 & 11 (laatste drie file-type 3) is het volgende 9-tal bytes een kopie van de header vooraan in de file.



```

212 1 file-type > 0 BASIC      1 N ARRAY  2 $-ARRAY  3 CODE
213 2 file-      >
214 3 lengte     >
215 4 begin-     >
216 5 adres      >
217 6            BASIC-      ARRAYletter/-type      255
218 7            lengte      255      255      255
219 8            regelnummer 255      255      exec adr
220 9            of 64K-1    255      255      of 0 0
221-256 alleen belangrijk bij snapshots ivm registerinhouden

```

Zie "System variables", Microdrive and Interface 1 manual, p 45.

#### DIRECTORY DESCRIPTION

```

00 ERASED          (herstelbaar indien niet overschreven)
01 BASIC           header file-type 0 met kopie in de dir
02 NUMERIEK ARRAY  header file-type 1 met kopie in de dir
03 STRINGARRAY     header file-type 2 met kopie in de dir
04 CODE            header file-type 3 met kopie in de dir
05 SNAPSHOT 48K    inhoud van de registers in de directory
06 MICRODRIVE FILE records van 540 bytes (zie hieronder)
07 SCREEN$         header file-type 3 met kopie in de dir
08 SPECIAL         ??? WIE HEEFT HIER INFORMATIE OVER ???
09 SNAPSHOT 128K   inhoud van de registers in de directory
10 OPENTYPE        data (headerless)
11 EXECUTE         header file-type 3 met kopie in de dir

```

In de filelengte in een header tellen noch die header, noch de 2 slotbytes in ieder blok (T & S volgend, of 00 laatste blok) mee. De DD-typen 05, 06, 08, 09 en 10 bezitten geen 9-bytes header.

DD-TYPE 06 M/DRIVE      RECORDS 27+512+1 BYTES      MAX INH 128 KB

```

001-010 DATA BLOCK (bij DD alle 0)
011 012 PREAMBLE (bij DD alle 255)
013 RECFLG bit 1=EOF bit 2=not a print file
014 RECNUM (0-255) \ bits 3-7 unused
015 016 RECLN (0-512)
017-026 RECNAM filename with trailing spaces
027 DESCHK checksum of the preceding 14 bytes

028-539 CHDATA 512 bytes of data

540 DCHK checksum of the preceding 512 bytes

```

Zie "Microdrive channel", Microdrive and Interface 1 manual p47.

Jack Raats

-

Infotel 01670-66845

-

Ed Weijgers



Kanalen (channels) dienen voor uitwisseling van gegevens (data) tussen de verschillende onderdelen van een computerconfiguratie. Hiervoor is zowel hard- als software (ingebakken of niet) nodig. INPUT en OUTPUT van data hebben altijd betrekking op de CPU, zeg maar de processor Z80. Aan de buitenkant (periferie) zitten alle interfaces; voor toetsenbord, scherm, CR, printer, drives enz.

Door een handig systeem in de SP kan een programmeur zich deze kanalen veel eenvoudiger voorstellen en ze ook zeer gemakkelijk in BASIC of MC benaderen via zogenaamde stromen (streams).

Een kanaal wordt gedefinieerd door een kaart (record), met alle informatie die nodig is voor het transport van de gegevens. Zo'n kanaal(kaart) bevindt zich in de CHANNEL AREA, waarvan het adres gegeven wordt door de systeemvariabele CHANS op 23631 en 23632.

Een stroom is een getal van twee bytes, opgeslagen in STRMS, een systeemvariabele op de adressen 23568 t/m 23605. Deze 19 streams in STRMS worden, in deze volgorde, aangeduid met #-3 t/m #15. De lage byte van #0 staat dus 3\*2 plaatsen verder dan het begin. Het STRMS-getal van #S begint derhalve op het adres  $23574+S*2$ . (In BASIC kunnen #-3 t/m #-1 niet gebruikt worden, in MC wel.)

Is zo'n getal geen 0, dan is de stroom gekoppeld aan een kanaal. Het verwijst dan op de volgende manier naar het adres daarvan:

kanaaladres = CHANS-1 + STRMS-getal

De aan de stromen gekoppelde kanalen kunnen bekeken worden door middel van dit BASIC-programmaatje "kankijk":

```
1 DEF FN D(a)=PEEK a+PEEK (a+1)*256
2 INPUT "#";S: LET K=FN D(23574+S*2): PRINT "#";S:
  IF NOT K THEN PRINT "is vrij": STOP
3 FOR A=FN D(23631)-1+K TO 9E9:
  PRINT A;" ";PEEK A,CHR$ PEEK A AND PEEK A>32: NEXT A
```

RUNnen hiervan direct na inschakeling laat zien dat er bij enige streams naar eenzelfde kanaal verwezen wordt, dat #4 t/m #15 nog vrij zijn en dat de vijfde kanaalbyte bij een letter behoort:

stream number	0	1 -3	2 -2	-1	3
channel address	CHANS+0	CHANS+5	CHANS+5	CHANS+10	CHANS+15
ROM OUTPUT routine	244, 9	244, 9	129, 15	244, 9	
ROM INPUT routine	168, 16	196, 21	196, 21	196, 21	
channel identifier	CODE "K"	CODE "S"	CODE "R"	CODE "P"	

Op CHANS+20 staat de afsluiter van de CHANNEL AREA: 128 (80h). Daarna volgt het BASIC-programma (adres: systeemvariabele PROG). Dit is de initiele instelling van de streams en de SP-channels.



De functies van de drie vanuit BASIC bereikbare Spectrumkanalen:

#0/1	aan K(eyboard)	INPUT vanaf toetsenbord
		OUTPUT naar benedendeel van scherm
#2	aan S(creen)	OUTPUT naar bovendeeel van scherm
#3	aan P(rinter)	OUTPUT naar ZX-printer

De BASIC-opdracht om een stroom aan een kanaal te koppelen luidt

OPEN #[stream number],[channel spec]

Het bij het [stream number] behorende STRMS-getal gaat door deze opdracht naar het daarachter gespecificeerde kanaal verwijzen. Er mogen meerdere streams naar eenzelfde kanaal geOPENd worden. Het OPENen van de streams #0 t/m #2 kan moeilijkheden opleveren, bij de DD kan zelfs #3 niet helemaal naar behoren functioneren.

Voor de drie SP-kanalen bestaat een [channel spec] alleen uit de [channel identifier] "K", "S" of "P" (kleine letters mogen ook). Andere [channel identifiers], die bij diskinterfaces als MD, OD, BD of DD mogelijk zijn, moeten of mogen vaak nader worden gespecificeerd door toevoegingen. Dit verschilt echter per systeem. Soms kunnen de gegevens in een SP-kanaal zelf gewijzigd worden. Meestal wordt door het OPENen een extra kanaal toegevoegd achter het laatste. De afsluiter en de evt aanwezige BASIC met variabelen worden hierdoor opgeschoven (het 'optileffect' t/m STKEND).

Is stroom S geOPENd naar een voor OUTPUT geschikt kanaal, dan kunnen er gegevens naartoe 'geschreven' worden door de opdracht:

PRINT #S;[print items]

Merk op dat PRINT kort is voor PRINT #2 en LPRINT voor PRINT #3. Pas hierbij op met de scheidingstekens tussen de [print items]: een komma schrijft een CHR\$ 6 en een apostrof een CHR\$ 13 (CR). Sluit de PRINT-opdracht met een kommapunt af als u geen CR wilt. #S is zelf ook een [print item], u kunt dus van stream wisselen. Sommige kanalen bevatten een databuffer, bv ter grootte van een diskblok. Verwerking van geschreven [print items] vindt dan niet altijd direct plaats, maar pas wanneer zo'n databuffer vol is.

Is het een INPUT-kanaal, dan kan daaruit 'gelezen' worden met de functie INKEY\$#S (INKEY\$ is kort voor INKEY\$#0), of de opdracht:

INPUT #S;[variables]

Lezen met een stringvariabele kan alleen als er een CR voorkomt. Slechts met INPUT-variabelen kan gelezen worden (zie "VARIABLEN BIJ INPUT" in IMPULS 83-16). Andere [print items] kunnen tot ongewenste of onmogelijke OUTPUT leiden. Ook komma en apostrof schrijven, maar INPUT afsluiten met een kommapunt is niet nodig.



GeOPENde streams kunnen worden ontkoppeld door de BASIC-opdracht

CLOSE #[stream number]

Hierdoor wordt een bijbehorend STRMS-getal initieel, dwz dat het bij #0 t/m #3 naar het oorspronkelijke SP-kanaal gaat wijzen en bij #4 t/m #15 nul wordt (vrij). Was de stream geOPENd naar een extra kanaal dan wordt dit opgeruimd, maar pas nadat onverwerkte data uit een buffer geschreven werd. SP-kanalen blijven bestaan.

Alle streams in een klap CLOSEn is vaak mogelijk met CLEAR #. Nog niet verwerkte data in kanaalbuffers gaat hierbij verloren. De BASIC van de DD bevat helaas een incompatibiliteit. Daar moet wegens een systeemfout CLOSE #\*S en CLOSE #\* gebruikt worden.

#### OPMERKINGEN

De meeste systemen verstaan de MD-syntaxis om te kunnen lezen en schrijven bij een disk/cartridge. Daarbij dient aan de [channel identifier] "M" een [file specifier] te worden toegevoegd om een [channel spec] te verkrijgen. Die ziet er dan als volgt uit:

"M";[drive number];[name string]

Hierbij zijn nog toevoegingen (file type, IN, OUT, RND, EXP enz) of weglatingen ("M"; is default) mogelijk. Zie de handleidingen. Hierdoor ontstaat een diskkanaal met een buffer erin ter grootte van de blokken op de disk/cartridge in de drive (optileffect!).

MD bezit altijd 58 extra systeemvariabelen, van 23734 t/m 23791. Daarachter, maar voor CHANS, creëert MD per geOPENde file een zg 'MD map' van 32 bytes (met voor iedere cartridgesector een bit). Alleen MD en OD OPENen ook bij het LOADen, SAVen enz een diskkanaal, maar dan tijdelijk. Dat is in BASIC niet te bekijken, maar soms moet er wel rekening gehouden worden met een optileffect. BD gebruikt hiervoor de printerbuffer en DD zijn eigen RAM. HerLOADen van als CODE geSAVEde BASIC is daarom alleen mogelijk bij een 'kale' SP of met DD. Daarom introduceerden we H- en P-files.

Het is mogelijk om mbv MC een kanaal te creëren en er een stream naar te OPENen. Universele MC hiervoor is moeilijk te schrijven, omdat niet alle systemen de kanaallengte, nodig voor het verwijderen, op dezelfde plaats verwachten. CLOSEn moet dan ook met MC gebeuren. Dat moet zeker als in STRMS naar een kanaal buiten de CHANNEL AREA verwezen wordt (het kan zelfs voor CHANS liggen).

Kleuropdrachten met INK en PAPER werken op het 'current channel' en zullen daarom na gebruik van andere kanalen dan het S-channel niet werken. Maak dan eerst "S" current met de opdracht PRINT ;.



Bijna altijd wanneer er een disk begint te draaien voert de OD eerst een zg 'inquire' uit, om te weten te komen hoe hij de disk moet benaderen. Hiertoe worden de eerste 128 bytes uit het eerste blok van de disk, het zg ankerblok (blok -1), gekopieerd in IC0 van 10112 t/m 10239. Deze zg 'ankerblokroutine' wordt daarna geexecuteerd, waardoor deze vier diskgegevens in de bijbehorende disksubtabel van IC0 worden aangepast aan de disk:

A tracks/side	40	S+0	10114	
B blocks/track	18	S+1	10115	
C bytes/block	256	S+2	10116	$2^{(\text{bit}7*2+\text{bit}6)*128}$
D sides/disk	1	S+2	10116	$1+\text{bit}4$

De letters voor de omschrijvingen van de diskgegevens corresponderen met de opties in het menu van "inst&form" (zie de volgende IMPULS). Daarachter staan de standaardinstellingen, de adressen in de bijbehorende disksubtabel (met beginadres S) en in de gekopieerde ankerblokroutine, en eventueel verdere toelichtingen. Inquire geschiedt bij alle diskbenaderingen door opdrachten met:

```
SAVE *   LOAD *   MERGE *   VERIFY *   ERASE
OPEN #   CAT      " CAT "   MOVE D;.. TO ..  MOVE .. TO D;..
```

Dit gebeurt dus niet bij: MOVE #.. TO #.., POINT # en FORMAT. Verder worden onder andere ook nog deze diskgegevens gebruikt:

E block offset	0	S+3	
F anchor blocks	1	S+4 & S+5	laag voor hoog
G steprate	6	S+7 bit5bit4	00: 6 01: 12 (ROM 2.22) 10: 20 11: 30 ms
H motor delay	off	S+7 bit7	0: on 1: off

FORMAT kopieert de standaardankerblokroutine uit de ROM in het ankerblok, met daarin A t/m D uit de bijbehorende disksubtabel. Naast onder andere E t/m H gebruikt alleen FORMAT bovendien nog:

I interleave	13	S+11	
J skew	5	S+12	
K CAT blocks	7	S+13	aantal-1

In dit nummer staat het programma "format:z\*s", waarmee iedereen heel simpel disks kan FORMATTen met 1 of 2 maal 40 of 80 sporen. Het enige dat verkeerd gedaan kan worden is 80 sporen kiezen als de drive geschikt is voor 40; dat is door ons niet te voorkomen. Dit programma kan bovendien de schijf die geFORMAT wordt meteen van een ankerblok met de routine voor 'last drive' voorzien. In het volgende komt "inst&form", waarmee alle genoemde gegevens in te stellen zijn, maar dit eist meer kennis van de gebruikers. Andere, maar waarschijnlijk minder nuttige diskgegevens, treft u aan in de tekstfiles voor het programma "config" van DUCDISK-05.

E H F Weijgers - H Marsmanlaan 29 - 2624 TJ Delft



Eindelijk is onze laatste P-omzetter af. BD-ers behoeven zich nu niet langer meer te behelpen met "cr><p" voor cassetterecorder. Hierna volgen het BASIC- en het CODE-gedeelte van deze omzetter.

### HET SAMENSTELLEN

Neem de BASIC-regels zeer zorgvuldig over en SAVE het programma met de naam en het regelnummer zoals dat boven de LISTING staat.

Toets dan de 'hexdataregels' van de MC in, zonder de testsommen, maar wel met DATA achter elk regelnummer. MERGE onze "hexloader" uit IMPULS 81-20, geef CLEAR 27979, RUN en 65300 als beginadres. SAVE de CODE-file met naam, adres en lengte zoals aangegeven is.

HET BASICPROGRAMMA "bd><p" LINE 88 - EdW -

```

1 DEF FN L(A)=PEEK (A+11)+PEEK (A+12)*256+20:
  LET B=28E3: LET N$="12345678":
  LET T=FN L(23659)-20: LET A=T+FN L(T):
  LET D=PEEK 65313: LET L=A-B: LET F=65E3-A
2 INPUT ("0 CLEAR (IN ";L;" FREE ";F;")",
  "1 LOAD N>P", "2 SAVE P>N",
  "3 LOAD P" , "4 SAVE P",
  "5 CAT ";D , "6 <> DRIVE"),K
3 IF K<1 OR K>6 THEN GO TO 99
4 IF (K=1 OR K=3) AND D OR K=4 THEN
  INPUT " LOAD 2 LOAD SAVE "(K); "NAME",N$():
  IF K=1 THEN
    LET N$=N$+"9": POKE 23658,8:
    INPUT "BD-TYPE B / C / D ";N$(9):
    IF N$(9)<"B" OR N$(9)>"D" THEN RUN
5 GO SUB K*11+D: POKE T,PEEK T+128
6 IF PEEK A>=128 THEN
  LET A=A+FN L(A): GO TO 6
7 RANDOMIZE A: RUN

8 FOR F=1 TO 10:
  PRINT CHR$ PEEK (B+F);: NEXT F:
  PRINT " ";PEEK B, FN L(B);TAB 24;B: LET B=B+FN L(B):
  IF B<A THEN GO TO 8
9 PAUSE T: RUN

11 RANDOMIZE A: INPUT USR 65308: RETURN
15 FOR F=1 TO 9:
  POKE 23772+F,CODE N$(F): POKE A+F,CODE N$(F): NEXT F:
  RANDOMIZE A: INPUT USR 65304: POKE A+F,32
16 IF N$(9)="B" THEN
  POKE A+11,PEEK 23782: POKE A+12,PEEK 23783:
  POKE A,0: LET B=A+FN L(A):
  POKE A+13,PEEK (B+2): POKE A+14,PEEK (B+3):
  POKE A+15,PEEK 23784: POKE A+16,PEEK 23785: RETURN

```



```

17 POKE A+11,PEEK 23784: POKE A+12,PEEK 23785: LET L=1:
  IF N$(9)="C" THEN
    POKE A,3:
    POKE A+13,PEEK 23782: POKE A+14,PEEK 23783: RETURN
18 FOR F=A+10 TO A+8+PEEK (A+20)*2 STEP 2:
  LET L=L*(FN L(F)-20): NEXT F:
  POKE A,2 OR F+11+L<>A+FN L(A):
  POKE A+14,PEEK A*64+CODE "A": RETURN

22 RANDOMIZE B: LET B=USR 65306: GO TO 22+(10 AND B>T)
26 FOR F=1 TO 8:
  LET N$(F)=CHR$ PEEK (B+F): NEXT F: PRINT N$:
  LET L=PEEK B-(128 AND B<>T): LET A=B+FN L(B)
27 IF NOT L THEN
  FOR F=0 TO 3:
    POKE 27980+F,PEEK (A+F): NEXT F:
    POKE A,128: POKE A+1,170:
    POKE A+2,PEEK (B+13): POKE A+3,PEEK (B+14)
28 INPUT USR 15363: REM : SAVE N$CODE B+20,A-B-21+(5 OR L)
29 INPUT USR 65300: POKE 23781,CODE "BDDC"(L+1):
  IF NOT L THEN
    POKE A,PEEK 27980: POKE A+1,PEEK 27981:
    POKE A+2,PEEK 27982: POKE A+3,PEEK 27983:
    POKE 23782,PEEK (B+11): POKE 23783,PEEK (B+12):
    POKE 23784,PEEK (B+15): POKE 23785,PEEK (B+16)
30 IF L=3 THEN
  POKE 23782,PEEK (B+13): POKE 23783,PEEK (B+14)
31 INPUT USR 65302:
  IF B<T THEN
    LET B=A: GO TO 26
32 RUN

33 LOAD "CODE A,F: RETURN
41 INPUT USR 15363: REM : LOAD N$CODE A,F
42 RETURN

44 SAVE N$CODE B,L: RUN
52 INPUT USR 15363: REM : SAVE N$CODE B,L
53 RUN

55 GO TO 8+NOT L
63 PAUSE USR 15363: REM : CAT
64 RUN

75 INPUT "0 CR / 1-4 BD",D:
  POKE 65313,D OR D<0 OR D>4: INPUT USR 65314: RUN

88>CLEAR 27979: INPUT USR 15363: REM: LOAD "Cbd"<p"CODE
99 RANDOMIZE USR 65310: RUN

```

NB: gebruik waar mogelijk KEYWORDS. Dit moet beslist in regel 4.



DE MC "Cbd><p"CODE 65300,203 IN HEXDATAREGELS - JaRa -

1	"18 38 18 2D.18 13 18 4F.18 53 C3 CF.FF 00 3A 21"	1150
2	"FF A7 C8 3D.0E 01 C3 63.FF CD 4E FF.2A 76 5C 11"	3204
3	"14 00 19 ED.5B EB 5C 3A.EA 5C 47 0E.05 CD 63 FF"	4937
4	"C9 3A 0F 5D.0E 09 CD 63.FF C9 0E 0A.CD 63 FF 3A"	6728
5	"0F 5D FE FF.28 06 0E 08.CD 63 FF C9.FB CF 0E E5"	8874
6	"CD 06 3C E3.C3 FD 3B FD.CB 47 86 18.10 FD CB 47"	11107
7	"C6 DD 2A 76.5C AF 37 11.11 00 CD 56.05 3E 02 CD"	12607
8	"01 16 11 C0.09 2A 76 5C.E5 E5 7E CB.BF CD 0A 0C"	14305
9	"06 0A 23 7E.D7 10 FB E1.11 14 00 19.DD E1 FD CB"	16153
10	"47 46 20 14.DD E5 DD 7E.00 F5 DD CB.00 BE CD 70"	18319
11	"09 F1 E1 77.DD 2B 18 0F.DD 5E 0B DD.56 0C 3E FF"	20178
12	"37 E5 DD E1.CD 56 05 DD.E5 C1 C9 21.4C 6D E5 11"	22512
13	"4D 6D 01 C6.91 36 00 ED.B0 C1 C9"	23903

### HET GEBRUIK

De gebruiksaanwijzing, die al in IMPULS 81-21 werd gepubliceerd, is geldig voor alle P-omzetters, dus ook voor dit BD-programma. Aangezien de BD met namen van acht ipv tien tekens werkt, moeten we nog enkele opmerkingen over het gebruik van filenames maken.

Laat een filenaam niet beginnen met een driveletter gevolgd door een dubbele punt. Gebruik om van drive te wisselen uitsluitend de daarvoor bestemde optie "6 <> DRIVE". De drivenummers 1 t/m 4 hebben betrekking op de BD-drives A t/m D, nummer 0 is voor CR.

Bij "1 LOAD N>P" vanaf BD is alleen een filenaam soms onvoldoende. Daarom wordt er ook om een BD-typeletter B, C of D gevraagd. Deze wordt op de negende positie van de naam in de P-file gezet. Dit is noodzakelijk om gelijknamige files van verschillend type te kunnen onderscheiden bij het uitpakken met de P-omzetters van andere disksystemen, daar ze elkaar anders zouden overschrijven.

Tijdens het uitpakken van een pakket met "2 SAVE P>N" op BD worden de laatste twee tekens van iedere filenaam niet overgenomen. Zijn die tekens significant, dan kunt u die namen in de P-buffer eerst wijzigen, zoals beschreven is in TRANS 22.3 in dit nummer. Een C-file die eenzelfde naam krijgt als een B- of D-file dient als laatste uitgepakt te worden, anders wordt deze overschreven.

Ook bij "4 SAVE P" naar CR zijn de namen maar acht tekens lang. Bij "1 LOAD N>P" en "3 LOAD P" vanaf CR wordt er niet naar namen gevraagd. CR-filenames komen wel volledig in de P-files terecht.

Wanneer dit programma wordt gebruikt mag ZX-IF1 niet aangesloten zijn, aangezien anders de adressen van de gebruikte BD-systeemvariabelen niet meer kloppen (ZX-IF1 schuift ze 58 plaatsen op).

Jack Raats

- SGG-infotel 01670-66845

- Ed Weijgers



Gebruiksaanwijzing overbodig: het simpele menu is overduidelijk.

HET BASICPROGRAMMA "format:z\*s" LINE 99 - EdW -

```

1 DEF FN I()=CODE INKEY$#3+CODE INKEY$#3*256: GO SUB 5
2 POINT #3;S: LET T=CODE INKEY$#3:
  POINT #3;S+2: LET B=CODE INKEY$#3/16:
  LET Z=INT (B-INT (B/2)*2):
  PRINT AT 9,0;"0 drive"      ,D''
        "1 zijden"          ,Z+1''
        "2 sporen"          ,T''
        "3 FORMAT ";
        #0;"?": BEEP .1,9      (laat ? knipperen)
3 LET N$=INKEY$: IF N$<"0" OR N$>"3" THEN GO TO 3
4 INPUT ;: GO SUB CODE N$: GO TO 2

5 LET D=PEEK 23728: CLEAR #: OPEN #3;"CODE "
6 LET D=D OR NOT D OR D>4-(USR 8=2.1): POKE 23728,D:
  POINT #3;8192:POINT #3;FN I()+10:POINT #3;FN I()
7 LET N$=INKEY$#3: LET S=FN I():
  IF CODE N$<>D THEN GO TO 7
8 RETURN

48 LET D=D+1: GO TO 6
49 POINT #3;S+2:
  LPRINT CHR$ ((INT (B/2)*2+B-INT B+NOT Z)*16);: RETURN
50 POINT #3;S: LPRINT CHR$ (40+(40 AND T=40));: RETURN
51 INPUT "DISKNAAM",N$: FORMAT D;N$: RETURN

```

99>RUN

Wilt u disks meteen zo kunnen FORMATTEN dat ze later altijd het laatst gebruikte drivenummer in 23728 achterlaten? Dat kan ook.

Plaats de 'last drive'-routine uit IMPULS 74-23 in het ankerblok van een disk en installeer in IC0 tenminste de extra opdrachten LOAD @ en SAVE @ uit IMPULS 84-20. Er mogen er dus best meer in. SAVE dan de IC0-inhoud als "Cic0:"+STR\$ USR 8 (CODE 8192,2048). (Op de DUCDISK komen versies voor OD-ROM 2.1, 2.2, 2.22 en 2.3).

SAVE tenslotte "format:z\*s" met de als volgt gewijzigde regels:

```

51 INPUT "DISKNAAM",N$: FORMAT D;N$: POINT #3;S:
  LET N$=INKEY$#3+INKEY$#3+CHR$ (INT (CODE INKEY$#3/16)*16):
  POINT #3;10114: LPRINT N$;: SAVE @D,65535,10112: RETURN

```

```

99>GO SUB 5: LOAD *D;"Cic0:"+STR$ USR 8CODE 4E4:
  POINT #3;8185: SAVE *#3CODE 4E4,2048: RUN

```

E H F Weijgers - H Marsmanlaan 29 - 2624 TJ Delft



Sommigen zullen in de afgelopen tijd in de echomail de tear-line

--- QBoss vl.01b

(of iets dergelijks) tegengekomen zijn. Die is dan afkomstig van een node of point draaiend op een Sinclair QL-computer. Hoewel Erik Slagter een tijdje terug (in nummer 3 van juni 1989) al het een en ander heeft verteld over de QL bij de bespreking van zijn terminalprogramma "Qommunicado", zal ik toch nog even een korte introductie geven over de QL, voor diegenen onder jullie die misschien denken dat de QL een PC-clone is :-).

De QL, oftewel "Quantum Leap", werd begin 1984 gelanceerd door Sir Clive Sinclair, een Engelse computerfabrikant die destijds veel succes oogstte met zijn goedkope computertjes, zoals de ZX80, ZX81 en ZX Spectrum. De QL was bedoeld voor de "serieuze" markt: 128K RAM met de mogelijkheid tot uitbreiding naar maximaal 640K, opslag van gegevens op twee ingebouwde microdrives die elk ruim 100K konden opslaan op cartridges, een gestructureerde BASIC met procedures en functies, en last but not least een multitasking operating system, QDOS genaamd. Bovendien werden vier applicatieprogramma's meegeleverd: een tekstverwerker, bestandsbeheerder, spreadsheet en grafisch tekenprogramma. Dit soort features waren in 1984, toen je voor een gewone PC nog meer dan tienduizend gulden moest betalen, uitermate spectaculair voor een machine van f 2000 en veel computerfreaks bestelden meteen na de aankondiging een QL bij Sinclair.

De introductie van de QL ging echter niet bepaald van een leien dakje. Allereerst werden de eerste exemplaren pas na drie maanden leverbaar, hetgeen de QL al de bijnaam "Quite Late" opleverde. Erger was dat die machines veel bugs in het (ROM-)operating system bevatten en de bijgeleverde programma's er wel mooi uitzagen, maar zeer traag en onbetrouwbaar werkten. Pas een jaar na de introductie was Sinclair in staat behoorlijk werkende QL's te produceren. Helaas kwam hij op dat moment ook in financiële problemen door de inzakkende markt voor home-computers en een paar geflopte projecten. Begin 1986 moest hij zijn bedrijf verkopen aan Amstrad. Die bracht nog wel enkele vernieuwde versies van de Spectrum uit, maar had voor de QL geen belangstelling meer. De produktie werd gestaakt en de grote voorraad QL's werd tegen dumprijzen verkocht.

Betekent dit nu dat de QL dood is? Geenszins. Anno 1990 is er nog steeds een ruime keus aan hard- en software voor de QL te koop, alhoewel je daar wel voornamelijk in Engeland of Duitsland voor moet zijn. Het is zelfs zeer goed mogelijk een QL te emuleren op een Atari ST. Omdat de hardware van een ST veel sneller is dan die van de QL, is die emulator zelfs drie- a viermaal zo snel als een "echte" QL!

Een van de redenen waarom computerliefhebbers nog steeds op de QL freaken is het operating system, het al eerder genoemde QDOS.



QDOS was in 1984 zeer revolutionair, en is dat nu nog. Dit komt omdat QDOS volledig multitasking is: programma's (in QDOS-terminologie een "job") kunnen gelijktijdig draaien, en als een job op input van de console of een ander I/O-apparaat moet wachten, wordt zijn "time-slice" automatisch toegewezen aan andere jobs. Ook is QDOS "device-independent": een programma kan altijd dezelfde I/O-aanroepen gebruiken ongeacht of dat nu naar een seriële poort, schermwindow, floppydisk of harddisk gaat.

Tot zover mijn introductie van de QL die toch wel weer aardig lang geworden is. Nu snel over naar de kern van het verhaal:

#### DE QL EN DATACOMMUNICATIE

Voor het begin hiervan gaan we terug naar de zomer van 1987. Er bestond toen een gecompileerd BASIC-programma van Engelse make-lij, waarmee op de QL, in samenwerking met een speciaal voor de QL gemaakt modem, een Viewdata host-systeem kon worden opgezet. Het programma was nou niet bepaald een juweeltje van programmeertechniek: alle pagina's werden bijvoorbeeld van tevoren in RAM geladen zodat je een flinke RAM-uitbreiding nodig had om een paar honderd pagina's online te hebben (en je geheugen zat dan ook meteen vol, zodat je geen andere programma's meer kon draaien!). De opzet was ongeveer gelijk aan het "Beebtel"-programma dat de oudere datacom-freaks zich misschien nog wel zullen herinneren: als je een bericht wilde opgeven moest dat via een "antwoordbeeld" gebeuren, waarna het in het bestand werd gezet. Het kon niet meer veranderd kon worden, behalve door de sysop.

Aangezien ik nou niet bepaald onder de indruk was van dat programma, en ueberhaupt een BBS-programma a la Fido wilde (Opus bestond nog nauwelijks, om over QuickBBS maar te zwijgen...), begon ik zelf maar wat te programmeren in 68000 assembler. Juist ja, assembler, want ik was er toch al aardig bedreven in en de twee voor de QL verkrijgbare C-compilers waren te beperkt en buggy. En zo gebeurde het dat op de avond van zaterdag 27 juni 1987 QBOX versie 0.01 voor het eerst online kwam, uitsluitend op 1200/75 bps bereikbaar (een 2400 bps-modem kostte toen nog een vermogen!). Overigens was je op die eerste versie snel uitgekeken: er stonden alleen bulletins in, en het kende nog niet eens een gebruikers registratie!

De nieuwe versies van QBOX volgden elkaar snel op: soms was het zelfs zo dat een om 19.55 uur door de assembler gehaalde versie meteen om 20.00 uur online kwam (en ook weer snel offline ging vanwege de bugs...). Er kwam een mail- en een file-afdeling, een user-systeem a la Fido en de mogelijkheid voor de sysop om tegelijk met een user lokaal online te zijn (heel eenvoudig op een multitasking machine!).

Toen in de loop van 1988 Echomail in Nederland een grote vlucht nam werd het maar eens tijd om QBOX ook daarvoor aan te passen.



In februari 1989 waren de programma's SCANMAIL en TOSSMAIL af; deze konden echomail scannen en tossen. Maar er was echter nog geen FidoNet mailer, dus moesten de echomail packets met de hand worden gedown- en upload! Er kwam een link van QL-georiënteerde echo's tussen mijn systeem en KU-EL Tel van Frank Troost. Ook gingen "points" meedoen, o.a. Andre van de Wijdeven en Erik Slagter lagen bij mij constant met elkaar in de clinch over de concept-verschillen tussen QDOS en MS-DOS :-). We gebruikten eerst geen officiële FidoNet adressen maar een "fake net" 9999; toen QBOX later FidoNet-compatible werd hebben we echter wel officiële nodenummers gekregen in het DFF-net.

FidoNet mail werd uiteindelijk mogelijk met twee programma's, "Poll" en "TempFront". Poll was een programma om een andere node te mailen, en TempFront was een front-end programma waarmee QBOX inkomende FidoNet calls kan beantwoorden. Beide programma's zijn overigens in hun huidige vorm nogal primitief; alle mailsessies gaan volgens de FTS-1 norm met Xmodem en Telink, maar ik hoop binnenkort wel SEALink erin te hebben. Door o.m. Erik Slagter zijn ook een aantal utilities gemaakt waaronder "autopoll", die als een soort scheduler dient voor Poll. Het is overigens de bedoeling dat Poll en TempFront ooit worden vervangen door een "echte" mailer a la Binkley of FrontDoor, maar aangezien ik bijna de enige QL'er ben die geïnteresseerd is in het schrijven van FidoNet software (BBS, mailer, tosser/ scanner en utilities zoals UNZIP/UNARC etc. heb ik allemaal zelf moeten schrijven) kan dat nog wel enige tijd duren. QBOX zelf is bijvoorbeeld hoognodig aan een update toe; de huidige versie kan slechts 32 area's aan met maximaal 255 berichten per area (tja wist ik twee jaar geleden veel dat echomail zo'n vlucht zou nemen...).

Momenteel zijn de Sinclair-georiënteerde echo's verkrijgbaar op diverse systemen in Nederland, Duitsland, Oostenrijk, Zweden en Groot-Brittannië. Zelf draai ik met twee BBSen: een PC-systeem (node 2:283/500, 035-237178; 300, 1200, 2400 en HST-snelheid) en een QL-systeem met QBOX (node 2:283/501, 035-216520; 300, 1200, 2400 en 1200/75). Nederlandse QBOX-BBSen zijn Sinclair Box (node 2:285/103, 01670-66845 (23.00-5.00)), KU-EL-TEL (node 2:285/102, 01650-37105) en QLAT BBS (node 2:283/508, 030-962265).

De internationale echo's hebben als onderwerp: QL algemeen, Spectrum algemeen, Minerva operating system, programmeren in C en QBOX support. De voertaal is (uiteraard) Engels. Daarnaast zijn er nog een aantal Nederlandstalige echo's, maar die zijn helaas de laatste tijd erg rustig.

Inlinken op de echo's is mogelijk, maar aangezien ze niet op de backbone verkrijgbaar zijn zal je zelf mijn systeem (of een van de andere nodes) moeten pollen. Voor meer informatie kun je altijd een netmailtje sturen naar mij op node 2:283/500, of inloggen op een van mijn BBSen.

Jan Bredenbeek

BRON : "Nederlands Net Nieuws" - nummer 20 - januari 1991



In de vorige IMPULS (84-05) werd het VTX711-interface gepresenteerd, waarbij oa software werd beloofd. Ook de modemsoftware is inmiddels beschikbaar. Van elk programma is er tevens een versie voor het gemodificeerde VTX711-interface. Het betreft de titels:

VIDI711, FIDO711 en XCOM7 & VIDI711MO, FIDO711MO en XCOM8

De software is verkrijgbaar (cassette, f 7,50) tijdens onze gebruikersdagen in Houten, maar is ook simpelweg (gratis) te downloaden als P-file in Sinclair Box (01670-66845 - file-area 6).

Natuurlijk is ook de meegeleverde software bruikbaar. Voor het gemodificeerde interface moeten wel de poortadressen worden verPOKEt. Eerst een lijstje van de software op de VTX711-cassette.

- ukbas (viewterm)	- LINE 1
- code	- CODE 26624,12288
- newterm	- LINE 10
- newcode	- CODE 24832,15360
- teleterm	- LINE 10
- teleterm (telecode)	- CODE 59136,6400

VIEWTERM is een 'viditel'-programma.

NEWTERM is een 'fido'-programma, maar kan (met beperkingen) ook voor viditel-banken gebruikt worden.

TELETERM is voor het downloaden uit viditel-banken. In Nederland is dit programma onbruikbaar, door verschillen in de Nederlandse en de Engelse 'telesoftware'-norm.

De programma's kunnen op elk disksysteem overgezet worden, maar LOADen en SAVEn van files kan alleen via cassette of Microdrive. In de handleiding is vermeld hoe dat in zijn werk gaat. De programma's kunnen werken met Hayes-modems, zij het tot 1200 baud. Belangrijk hierbij is: EXTEND MODE fungeert als CONTROL-toets.

Heeft u de VTX711 gemodificeerd? LOAD dan de betreffende CODE en RUN de onderstaande regels. Eerst voor VIEWTERM en NEWTERM:

```
1 FOR f=1 TO 7: READ a,b: POKE a,b: NEXT f: STOP
2 DATA 27979,183,27990,183,28011,183,28020,183,27938,183
3 DATA 27971,247,27965,247
```

Hoewel onbruikbaar in Nederland, kunt u TELETERM aanpassen voor gebruik met het gemodificeerde VTX711-interface. RUN dan:

```
1 FOR f=1 TO 7: READ a,b: POKE a,b: NEXT f: STOP
2 DATA 59211,183,59222,183,59243,183,59252,183,59170,183
3 DATA 59203,247,59197,247
```

Kees Versluis (070-3604185) - Jack Raats (Infotel 01670-66845)



U hebt hem misschien al gezien, in onze stand op de HCC-beurs in Utrecht verleden jaar, onze DD-programmabankier: Rob Willig. Hij beheert de spullen die de SGG heeft overgenomen van de voormalige "DISCIPLE NIEUWSBRIEF", kopieert en verzendt de diskettes, en verkoopt aan de balie op onze SINCLAIRGEBRUIKERSDAGEN in Houten. (Alleen daar is nog een beperkt aantal nieuwsbrieven leverbaar.)

Er is dus wel wat gebeurd sinds onze oproep in IMPULS 83. U kunt verderop zien dat er al heel wat diskettes zijn. Nieuwe software is uiteraard zeer welkom. Neem daartoe contact op met DD-bankier

R Willig, J v Stolbergln 74, 1412 BJ Naarden, 02159-43087.

Diskettes bestellen kan alleen via onze postgirorekening 5374525 tnv HCC SINCLAIR GG te Bunnik, ovv DDiSC (nodig!) met nummer(s), 3.5 of 5.25 inch en 80 of 40 tracks (enkelzijdig beschreven).

Prijs f 10 (HCC-SGG-leden en IMPULS-abonnees) of f 12.50 / disk, plus verzendkosten: f 2.50 (1 of 2 disks) of f 4.50 (3 of meer).

Gireer wel het JUISTE BEDRAG, anders ontvangt u minder dan u bestelde, en vermeld uw lidmaatschapsnummer voor de ledenkorting.

#### EEN OVERZICHT VAN ALLE TWINTIG THANS VERKRIJGBARE DDiSCS

- 01 BIT-MAP2 Toont overzicht der vrije en gebruikte sectors.  
BASIC-TAS Zet LISTings van BASIC-files in TW2-files.  
SNELLE-RAM Test de SP48-RAM op defecten.
- 02 WP48/128 Specword, tekstverwerker voor SP128 met DD/+D.  
TAPE>DISK Kopieert programma's van tape naar disk.
- 03 MFACE 128 LOAD-programma voor MultiFace-128.  
SNEL-ROM Test de SP48-ROM op fouten.  
NAMEDISC Geeft SYSteemfiles zinvolle namen.  
AUTOLOAD Menugestuurde AUTOLOADER.  
CLONEN Kopieerprogramma.
- 04 NAAM-CAT Verandert de CATALOGUS-header ("DISC x DIR")  
RECORD 1.3 Bestandsbeheer mbv OPEN en CLOSE #, SP48.  
LOADER-HOL Nog een AUTOLOADER.  
BITMAP3 Toont per track welke sectoren vrij/bezet zijn.  
TW Fraaie Tasword-Twee-BASIC, met character-set.
- 05 TRNS-DT.BS Kopieert programma's van disk naar tape.  
CP-OTYP Kopieert OPENTYP-bestanden.  
E.CAT2 .BS Peutert informatie uit de directory.  
TRNS-TD.BS Kopieert programma's van tape naar disk.  
G2-LIST.BS Toont GENS-files zonder GENS ("viewer").  
SNAP-SPY Toont BASIC in SNAPSHOTS op scherm/printer.



- 06 FIXAFILE Haalt gewiste programma's terug (mits intact).  
 OKIE-POKIE Zet valsspeelPOKES rechtstreeks in SNAPSHOTS.  
 BETA>DISCI Kopieert van BDisk naar DDisk.  
 GNSLIST.BS Toont GENS-files, maar sneller en beter.  
 +D T<>D.BS Aanpassingen voor kopierprogramma's, voor +D.  
 SELECT.BS Softwarematige drivewissel, met POKE ipv CAT.
- 07 +D HACK.BS Aanpassingen voor Hacker aan de +D.  
 COPIEER.BS Hoofdmenu voor reeks kopieerprogramma's.  
 SECTTEL.BS Controleert of programma's nog intact zijn.  
 SNAP 48.BS Kopieert 48K-SNAPSHOTS naar tape.  
 SNAP128.BS Kopieert 128K-SNAPSHOTS naar tape.  
 SNPL28D.BS Kopieert 128K-SNAPSHOTS naar tape, +D-versie.  
 SNAP-NAAM Toont SNAPSHOTSCREEN\$ en laat naam aanpassen.  
 AUTO-NAAM Geeft AUTOLOADERS een andere naam (in DOS).  
 GNS-TAS.BS Van GENS- naar TW2-files.  
 ROM1.GNS t/m ROM3.GNS ROM-disassembly (deel 4 is zoek).
- 08 H-COPY.BS Headerless CODE-blokken van tape naar disk.  
 MULTIFACER Stopt SNAPSHOTS vol met valsspeelPOKES.  
 SYS3D.CONV Maakt van Sys 3b en 3c de 'final' versie 3d.  
 DUMP.HR5 Screendumps op oa BROTHER-HR5, x/y instelbaar.  
 SCRNMAKER Haalt grapjes uit met SNAPSHOTSCREEN\$.  
 BEDS>DI1 Nog een kopieerprogramma van BDisk naar DDisk.  
 WP48+128 Specword-tekstverwerker, SP48/128/+2 met DD/+D.  
 PRFUNCTIES Past de printerinstellingen in Specword aan.
- 09 DIRLAAD.MC Razendsnelle directoryLOADER (naar 40.000).  
 SMARTKLOON Maakt 1-op-1 kopie, maar alleen gevulde tracks.  
 @VAR-LIST Toont overzicht van de DD-SYSteeminstellingen.  
 DISCDOC+1 Fraaie 3-in-1 toolkit: bitmap, unERASE, dokteren.  
 D>D48 Disk-kopieerprogramma voor SP48 met DD.  
 D>D128 Disk-kopieerprogramma voor SP128 met DD.  
 SECTORPIEK Bekijk en/of wijzig sectoren op schijf.  
 AUTO JACK AUTOLOADER met naar keuze een selectieve CAT.  
 RAM A.GNS4 t/m RAM D.GNS4 DD-RAM-disassembly.  
 CONVERT.BS Aanpassingen tbv "The Art Studio" met DD.
- 10 SECTORDUMP Toont de inhoud van de sectoren op een diskette.  
 SC-REPLACE Manipuleert SNAPSHOTSCREEN\$.  
 AUTO-(C)JV Joystick/cursorgestuurd menusysteem, LOADt snel.  
 LABELER Drukt etiketten af voor uw schijvenverzameling.  
 POKESYS.BS Speciaal DD-DOS voor hackers/valsspelers.  
 +D POKE.BS Als POKESYS, maar voor +D.  
 T>D2 Kopieerprogramma van tape naar disk, versie 2.  
 BETERGRIJS Mooiere grijstinten voor schermafdrukken.  
 Voorzien van testscherm en uitleg in TW2-file.
- 11 +D GREY Mooiere grijstinten voor de printer, nu voor +D.  
 SLOPER Hakt SNAPSHOTS in hapklare brokken.  
 DI#TEST Test zelf wanneer de meeste crashes optreden.  
 E.CAT4 Alle informatie uit de directory over files.  
 BASTAS.MC BASICLIST in TW2-files, sneller en eenvoudiger.



- +BASTAS.MC Als BASTAS.MC, maar voor +D.  
 LOADDIR/ Haalt een met SAVEDIR gemaakte kopie  
 SAVEDIR van de directory terug.
- 12 FC 80-DS Snel kopieerprogramma voor SP128.  
 SNEL-ERASE Wist alle files zonder te FORMATten.  
 DISKSEARCH Doorzoekt schijf op voorkomen van tekenreeks.  
 TAS128 Zet een werkende TW128 op uw DDisk.  
 PEEK- DEMO Demo-MC voor PEEK @ (kent DD niet).  
 M1 LDR.BS Universele LOADER voor MultiFace-One.  
 M128LDR.BS Idem voor MultiFace-128.
- 13 GETBITMAP Haalt bitmap van schijf ter ontleding.  
 CHECK.BS Meldt defecte sectoren op een diskette.  
 HIDEAFILE Maak een file (on)zichtbaar.  
 NETWBATTLE Een spel voor een netwerk.  
 D-CONTROLE Demonstratie DD "ON ERROR GO TO".  
 BASTAS.CON Verplaatst Bastas (voor gebruik in TW2 bv).  
 ADRES-1989 Database, 5 velden, snel en eenvoudig.
- 14 LINE Wijzigt de AUTOLINE in BASIC-files op disk.  
 SAVER Reddingsboei voor gegevens uit defecte sectors.  
 OPUSLD .BS Kopieert van ODisk naar DDisk.  
 OPUSCAT.BS Toont de CATalogus van een ODisk.  
 MAPPER Toont bij welke files de sectors behoren.  
 FRMAT84 FORMAT 80-tracks disk met 84 tr tbv dir-kopie.  
 TOOLKIT.BS Snelle, multifunctionele toolkit met D>D-copier.  
 PINCODE Bewaart uw pincode.
- 15 SCTORSAVE De echte redder van gegevens in defecte sectors.  
 FASTRAC.BS Kopieert supersnel, alleen voor twee drives.  
 DIRSORT.BS Sorteert de directory op naam/type van A-Z.  
 EXEC.MAKER Maakt EXECUTE-files.  
 REN5 5 Hernummert SP128-BASIC-programma dmv EXECUTE.  
 SYSLIST.BS Toont SYSteeminstelling in woorden ipv getallen.  
 AUTOLOAD.B Selector en LOADER voor SYSteemfiles.
- 16 HCC 1989 graphic special. Teken/inkleur/letter-routines.  
 + demonstratie + ontwerpprogramma  
 + voorbeeldscherm + uitleg in TW2-file
- 17 The Best of DISCIPLE NIEUWSBRIEF 1 t/m 10. Bevat een selectie van eerder uitgebrachte programma's.
- 18 DISKFILER Catalogiseerprogramma voor DDiskettes, maximaal 2000 programma's, met handleiding in TW2-file.
- 19 Eros (18+) Verzameling erotische programma's voor de liefhebber, alleen beeld.
- 20 Lancet Diverse programma's, de meeste speciaal voor de DD en +D, voorzien van ingebouwde hulpschermen.



Op het volgende paar bladzijden treft u ons tweede programma aan om systeemvreemde schijven te kopiëren: van Disciple naar Opus.

Het werkt bijna net zo als de programma's "bdss>od" en "bdds>od" uit IMPULS 84-28 en kent dus bijna geen beperkingen. Alle files, van alle elf DD-types, kunnen gekopieerd worden naar ieder soort ODisk (ongeacht diameter, blokgrrootte, SS/DS, 40/80 tracks enz):

00 ERASED	->	wordt overgeslagen
01 BASIC	->	0 BASIC (met evt autostartregelnummer)
02 NUMERIEK ARRAY	->	1 NUMERIEK ARRAY (met originele letter)
03 STRINGARRAY	->	2 STRINGARRAY (met originele letter)
04 CODE	->	3 CODE (met beginadres)
05 SNAPSHOT 48K	->	DATA (met DIR-record en DD-blokken)
06 MICRODRIVE	->	DATA (met DD-blokken)
07 SCREEN\$	->	3 CODE (16384,6912)
08 SPECIAL	->	DATA (met DIR-record en DD-blokken)
09 SNAPSHOT 128K	->	DATA (met DIR-record en DD-blokken)
10 OPENTYPE	->	DATA (met juiste lengte)
11 EXECUTE	->	3 CODE (met begin-, zonder startadres)

De DATA-file uit de types 05, 08 en 09 begint met het directory-record van 256 bytes (met de registerinhouden). Daarna - bij het type 06 alleen - 510 bytes uit ieder fileblok, ook het laatste. Als u zich afvraagt wat u hier in 's hemels naam aan hebt op een OD: u behoeft ze immers niet in uw kopieerselectie op te nemen.

Verderop in deze IMPULS vindt u een programma om uit een op deze manier gekopieerde 06-file een normale DATA-file te verkrijgen. Opnemen van deze conversie in "dd>od" gaf veel teveel rekenwerk.

De enige beperking geldt de veronderstelling dat er niemand meer DDisk van SD (Single Density) gebruikt. Dit betekent dat er dus altijd 10 blokken van 512 bytes op een track voorkomen, ongeacht diameter, enkel- of dubbelzijdig, met 40 of 80 tracks per zijde. Neem gerust contact op wanneer deze veronderstelling onjuist is.

#### - GEBRUIKSAANWIJZING -

Na het LOADen wordt u eerst het drivenummer voor de DDisk (kies uit 1 t/m 4) en het aantal van de tracks per zijde gevraagd (40 of 80) en dan het drivenummer voor de ODisk (kies uit 1 t/m 6). De DDisk moet in een passende drive zitten (SS of DS, 40 of 80).

Na beantwoording verschijnt de genummerde CATALOGUS van de DDisk met het verzoek de nummers van de gewenste files te noteren tbv de nu volgende repeterende selectievraag. DD-type (00 t/m 11) en (bruto) KBytes van de files worden ook getoond, achter de namen.

Elke vraag toont het aantal der KBytes van de reeds geselecteerde files, opdat u kunt zien of de selectie nog op uw ODisk past. Geef na NEXT ? een filennummer en na TO ? een (hoger) nummer als



HET BASICPROGRAMMA "dd>od" LINE 88 - EdW - 190191

HET LEZEN EN SCHRIJVEN DER TUSSENLIIGENDE FILEBLOKKEN

```

1 LOAD @D,T*10+CODE B$(512)-(N OR T<80),A: DD-blok naar B$
  LET T=CODE B$(511):                      volgend tracknr
  IF T THEN                                geen eindblok:
    LPRINT B$( TO 510);:                  blok -> ODisk
    GO TO 1:                              volgende blok
  DEF FN I( )=CODE INKEY$#3+CODE INKEY$#3*256 tbv adreszoeken
2 RETURN

```

HET LEZEN EN TONEN VAN DE DD-CAT

```

3 FOR F=0 TO 39:                          de dir-bloknrs
  LOAD @D,F,A+F*512:                      naar C$(oneven)
  IF CODE C$(F*2+2,9) THEN                even gebruikt:
    NEXT F                                volgend bloknr
4 PRINT "NOTEER DE NRS DER GEWENSTE FILES""': laatste filenr
  LET L=F*2:                             dir niet vol:
  IF L<80 THEN                            oneven gebr +
    LET L=L+NOT NOT CODE C$(L+1)         alleen filenrs:
5 FOR F=1 TO L:                          toon filenummer
  PRINT F;TAB 4;                          naam ongeERASEd
    C$(F,2 TO 11) AND CODE C$(F),        DD-type (00-11)
    CODE C$(F); TAB 20;                  blokkenaantal/2
    CODE C$(F,12)*128+CODE C$(F,13)/2:   (bruto KBytes)
  NEXT F:                                adres van B$(1)
  CLEAR #: LET A=A-529:                  selecteer files
  GO SUB 80

```

HET VERWERKEN DER GESELECTEERDE FILES

```

6 FOR F=1 TO LEN S$:                      de S$-posities
  LET F$=C$(CODE S$(F)):                 bijbehorende C$
  OPEN #3;O;F$(2 TO 11):                 file op ODisk
  LET T=CODE F$(14)                      tracknr beginbl
7 LOAD @D,T*10+CODE F$(15)-(N OR T<80),A: beginblok -> B$
  LET B=10:                              beginpos in B$
  LET L=CODE F$(213)+CODE F$(214)*256+9: file+headlengte
  GO SUB CODE "BAACFMCFDC"(CODE F$)      volgens DD-type
8 LET L=L-INT (L/510)*510:               restlengte
  LET T=CODE B$(511):                    volgend tracknr
  IF T THEN                              beginblok, mits
    LPRINT B$(B TO 510);:                geen eindblok
    GO SUB 1: LET B=1                     B tbv eind-B$
9 LPRINT B$(B TO L+(510 AND NOT L));:    eindblok -> OD
  CLOSE #3:                              sluit OD-file
  NEXT F:                                volgende file
  GO TO 90                               ook als S$=""

```



EVT OD-HEADER, BEGINPOSITIE IN B\$ EN TE KOPIEREN LENGTE

```

65 LPRINT B$( TO 4);B$(6 TO 7): RETURN      ARRAY (NUM / $)
66 LPRINT B$( TO 3);B$(8 TO 9);B$(6 TO 7);: BASIC
    RETURN
67 LPRINT B$( TO 6): RETURN                  CODE (& SCR EX)
68 LET B=1: LET L=L+CODE F$(211)*65536-9:    DATA (OPENTYPE)
    RETURN
70 LPRINT F$;                                SNAPSH SPECIAL
77 LET B=1: LET L=0: RETURN                  MICRODRIVE

```

HET VULLEN VAN DE SELECTIESTRING S\$ MET FILENUMMERS

```

80 LET S$="": LET B=0                        blokkenaantal
81 INPUT "SELECTED",+B/2;" KB",              gekozen files
    " NEXT "; LINE F$;" TO ";LINE T$:        filenr t/m nr
    IF NOT LEN F$ THEN RETURN                selectie klaar
82 IF NOT LEN T$ THEN LET T$=F$              slechts 1 file
83 LET F=VAL F$: LET T=VAL T$:
    IF F<1 THEN GO TO 80
84 FOR F=F TO T AND T<=L:                    begin opnieuw
    IF CODE C$(F) THEN                       mits F<=T<=L:
        LET S$=S$+CHR$ F:                   als ongeERASEd:
        LET B=B+CODE C$(F,12)*256+CODE C$(F,13) filenr -> S$
85 NEXT F: GO TO 81                          blokkenaantal
                                              daarna herhalen

```

HET INSTELLEN VAN DE OD-DISKTABLE TBV DE DDISK

```

90>CLEAR : DIM B$(512): DIM C$(80,256):      DD-blok en -dir
    INPUT "DD IN ";D,"40/80 TRACKS ";T:      drivenummers en
    "OD IN ";O:                               DD-tracksaantal
91 CLEAR #: OPEN #3;"CODE ":                 tilt 9 bytes
    OPEN #4;" CAT ";O:                       tilt als in 7:1
    POINT #3;FN I()+10:POINT #3;FN I():       -> diskinfo tbl
92 LET N=CODE INKEY$#3: LET A=FN I():        zoek adres van
    IF N<>D THEN GO TO 92                     DDisksubtabel
93 POINT #3;A: LPRINT CHR$ T;CHR$ 10;:       T tr, 10 bl/tr
    LET N=CODE INKEY$#3:POINT #3;A+2:        -> alle bits I
    LPRINT CHR$ (144+N-INT (N/16)*16);        blgr 512 en DS
    CHR$ 1;CHR$ 0;CHR$ 0;                    offs 1, 0 anker
94 LET A=PEEK 23627+PEEK 23628*256+526:      adres C$(1,1)
    LET N=1281-T*10: GO TO 3                  aftrek tbv kant

```

u ook nog daaropvolgende files wilt selecteren, of alleen maar ENTER na TO ? als het bij deze vraag slechts een file moet zijn. Deze selectievraag blijft terugkeren totdat u na NEXT ? alleen maar ENTER geeft, waarop de files in de selectievolgorde naar de ODisk gekopieerd worden. Bij een vergissing kunt u herbeginnen met de selectie door 0 na NEXT ? te geven. U hoeft nummers van geERASEde files niet over te slaan: die worden niet gekopieerd.>



HET BASICPROGRAMMA "ddmdf>oddf" LINE 3 - EdW -

1 FOR L=1 TO L:	de rec-posities
LPRINT INKEY\$#4;: NEXT L:	lees schrijf
LET P=P+540	tbv volgend rec
2 IF R-INT (R/4)*4<2 THEN	mits geen EOF:
POINT #4;P: LET R=CODE INKEY\$#4:	lees RECFLG &
POINT #4;P+2:	lengte in dit
LET L=CODE INKEY\$#4+CODE INKEY\$#4*256:	record POINT
POINT #4;P+15: GO TO 1	op rec-begin
3>CLOSE #3: CLEAR #:	
INPUT "FROM DRIVE",M,"NAME",M\$,	D=M toegestaan,
" TO DRIVE" ,D,"NAME",D\$:	mits dan D\$<>M\$
OPEN #3;D;D\$: OPEN #4;M;M\$:	recordlengte 1
CLEAR : FOR L=0 TO 0:	adres L is VARS
LET P=13: LET R=0: GO TO 2	tbv pointer #4

De DD-MD-file, met records van 540 bytes, wordt met recordlengte 1 geopend om de POINTER binnen de records te kunnen verplaatsen. Bit 1 van byte 13 geeft namelijk EOF aan en byte 15 en 16 (L H) de lengte van de eigenlijke records van 512 bytes die op byte 28 beginnen. Het laatste eigenlijke record kan echter korter zijn.

#### - GEBRUIKSAANWIJZING -

Na het LOADen van dit conversieprogramma worden u drivenummer en naam tbv de gekopieerde en dan tbv de nieuwe DATA-file gevraagd. Zijn deze drivenummers gelijk, dan moeten die namen verschillen. Daarna vindt conversie plaats, waarna de vragen terugkeren. Geef EDIT STOP wanneer er geen files meer zijn die u om wilt zetten.

>> slot van het artikel DD>OD >>

NB: De MC tbv LOAD @ wordt verondersteld in IC0 aanwezig te zijn en tbv de drive met de ODisk erin kan Motor-On-Delay nodig zijn. Desgewenst kunt u hiertoe dit programma zelf vooraf de CODE-file met die opdrachten in IC0 laten LOADen door dat in een regel 88 op te nemen, zoals (met PEEK last-drivenummer uit IMPULS 74-23):

88>LOAD PEEK 23728;"Cic0"CODE	naar IC6116 tbv
90 ....: *MOD 0,1	LOAD @ en *MOD

De opdracht voor de Motor-On-Delay kunt u toevoegen in regel 90.

E H F Weijgers - H Marsmanlaan 29 - 2624 TJ Delft



De Disciple kent twee soorten PRINT-files: OPENTYP en M/DRIVE. De eerste soort komt overeen met de DATA-files bij de MD en OD. De tweede werd gecreeerd om programma's die oorspronkelijk voor de MD geschreven zijn, zoals TW128 en GENS en PASCAL van HiSoft, meteen zonder problemen ook via DDisk's te kunnen laten werken.

Zolang je die MD-files alleen maar voor die programma's gebruikt gaat dit ook probleemloos, maar verder is er niets mee te doen. Ze zijn niet normaal te OPENen om te lezen of naar het scherm te MOVEn, laat staan te gebruiken voor communicatie via een modem. Daarom hebben we het conversieprogramma "ddmd><ot" geschreven, om uit een file van de ene soort er een van de andere te maken.

Wilt u bv een TW3-MD-tekst versturen, maak er dan eerst een TW3-OT-tekst van met "ddmd><ot". LOADt deze daarna in "tuwvsd", met type 5 (d). Het menu toont dan type v (met CR en LF). Desgewenst wijzigt u nu het teksttype. Tenslotte SAVET u de file als CODE. Maak ontvangen tekst in omgekeerde volgorde voor TW128 geschikt.

Doe hetzelfde met source-files van GENS of PASCAL, maar gebruik "tuwvsd" alleen voor DATA><CODE, voer nooit tekstconversie uit! Om tekstfiles te maken van GENS-CODE, of andersom, kunt u "g><t" gebruiken. Van t-files kan "tuwvsd" wel andere teksttypes maken.

Misschien komen er nog programma's om alles in een keer te doen.

HET BASICPROGRAMMA "ddmd><ot" LINE 9 - EdW -

```
1 CLOSE #4; DIM M$(10);
  INPUT "0 OT TO MD / 1 MD TO OT" 'C'
  "NAME MD",M$,"DRIVE",M,"NAME OT",O$,"DRIVE",O:
  POKE 65384,M
2 FOR M=1 TO 10: POKE 65373+M,CODE M$(M): NEXT M
3 IF C THEN OPEN #4;do;O$ OUT: RUN NOT USR 65368
4 RANDOMIZE USR 65370: MOVE do;O$ TO #4: RUN NOT USR 65372
9>LOAD d*"Cddmd><ot"CODE : RUN
```

DE MC "Cddmd><ot"CODE 65368,138 IN HEXDATAREGELS - JaRa -

1	"18 37 18 7A.18 51 20 20.20 20 20 20.20 20 20 20"	650
2	"01 00 00 00.CF 31 2A 68.FF 22 D6 5C.21 0A 00 22"	1725
3	"DA 5C 21 5E.FF 22 DC 5C.D9 E5 D9 CF.22 D9 E1 D9"	4326
4	"DD 22 6A FF.DD CB 04 BE.C9 CD 6C FF.DD 2A 6A FF"	6953
5	"DD 6E 45 DD.66 46 CD BA.FF DD 2A 6A.FF DD CB 43"	9507
6	"4E 20 08 CF.25 18 E5 DD.2A 6A FF CF.23 AF CF 21"	11403
7	"FB C9 11 52.00 DD 19 DD.E5 E5 3E 04.CD 01 16 E1"	13398
8	"D1 1A E5 D5.D7 D1 E1 13.2B 7C B5 20.F4 C9 CD 6C"	15881
9	"FF E5 D1 21.1E 5C 73 23.72 C9"	17194

Jack Raats

- Infotel 01670-66845

- Ed Weijgers

**sinclair**

**impuls**



De QL heeft de zeer handige eigenschap meegekregen dat grafische commando's niet met primitieve pixelcoördinaten, maar met meer geavanceerde (voorgeschaalde) grafische coördinaten werken. Nadat een Window van bepaalde grootte ( $W_x$  pixels horizontaal en  $W_y$  pixels verticaal) is geopend, kunnen met het SCALE-commando de plaats van de oorsprong van het coördinatenstelsel alsmede de te gebruiken schaafactor  $S_y$  ingesteld worden. Deze schaafactor geeft aan hoeveel schaaeenheden in verticale richting in het Window passen. Het aantal schaaeenheden dat in horizontale richting in het window past ( $S_x$ ) ligt dan eenduidig vast. Voor optimaal gebruik van het window-oppervlak is het noodzakelijk deze  $S_x$  vrij precies te weten. Gegeven de verhouding  $W_x/W_y$  en de aspect-ratio  $r$  van de pixels (de verhouding breedte/hoogte van een enkel pixel in Mode 4; bij benadering  $r=0.74$ ) zou men verwachten  $S_x$  te kunnen berekenen uit

$$S_x = r \cdot S_y \cdot W_x / W_y$$

Al experimenterend kwam ik tot de conclusie dat deze relatie weliswaar ongeveer klopt, maar dat met name bij windows waarin  $W_x/W_y$  veel groter of veel kleiner dan 1 is, deze relatie zeker niet toegepast mag worden. Om dat nader te onderzoeken definieerde ik een grootheid  $S$ , gegeven door

$$S = (S_x \cdot W_y) / (S_y \cdot W_x)$$

en schreef een programmaatje dat automatisch (via het principe van intervalhalvering) deze grootheid  $S$  bij een honderdtal ( $W_x, W_y$ )-combinaties bepaalde en naar een file schreef (zie de bijgevoegde listing). Vervolgens was het zaak te zoeken naar een functioneel (wiskundig) verband dat voor elke ( $W_x, W_y$ )-combinatie direct  $S$  oplevert. Uit de rij geproduceerde getallen bleek weliswaar dat er een verband van de volgende vorm bestond:

$$\begin{aligned} S &= A_y(W_x) + B_y(W_x)/W_y & (W_x \text{ constant}) \\ S &= A_x(W_y) + B_x(W_y)/W_x & (W_y \text{ constant}) \end{aligned}$$

met  $A_y$  en  $B_y$  functies van  $W_x$ , en  $A_x$  en  $B_x$  functies van  $W_y$ . Beide relaties combinerend volgt uit de Ceteris-Paribusvoorwaarde dat de relatie tussen  $S$ ,  $W_x$  en  $W_y$  moet luiden:

$$S = P + Q/W_x + R/W_y + T/(W_x \cdot W_y)$$

met  $P$ ,  $Q$ ,  $R$  en  $T$  vooralsnog onbekende constanten. Via een zogenaamde Kleinste-Kwadratenaanpassing van het bovenstaande functionele verband aan de eerder verkregen set getallen vond ik

$$\begin{aligned} P &= 0.73701 & R &= 0.82014 \\ Q &= -1.1055 & T &= -1.2316 \end{aligned}$$

Wat direct opvalt aan deze getallen is dat

$$Q/P = T/R = -1.50$$



Dat betekent dat de gegeven vergelijking voor S vereenvoudigd kan worden tot een vergelijking met nog slechts drie in plaats van vier constanten, te weten F, G en H:

$$S = F * (1 + G/Wx) * (1 + H/Wy)$$

zodat

$$Sx = S * Sy * Wx / Wy = F * Sy * (Wx + G) * (1 / Wy + H / (Wy^2))$$

Deze laatste vergelijking voor Sx kan voor voldoende hoge waarden van Wy zeer goed benaderd worden door

$$Sx = F * Sy * (Wx + G) / (Wy - H)$$

en die formule lijkt "verdacht" veel op de allereerste in dit artikelje, zij het dat Wx en Wy een soort offset-waarden hebben gekregen (-G resp H). Het toepassen van de eerder genoemde Kleinste-Kwadratenmethode op deze formule levert het overtuigende eindresultaat (de nullen zijn significant!):

$$F = 0.7380074 \quad G = -1.500000 \quad H = 1.000000$$

Het bovenstaande verhaal gaat uit van een window zonder border. Indien U met het BORDER-commando een rand ter grootte b hebt ingesteld, dan luidt de formule voor Sx:

$$Sx = F * Sy * (Wx - 4 * b + G) / (Wy - 2 * b - H)$$

Mocht U de wiskundige inhoud van dit verhaal niet helemaal hebben kunnen volgen, dan volgt hier de

#### CONCLUSIE

Opent U op de QL een window met breedte Wx pixels (Wx wordt even verondersteld) en hoogte Wy pixels (Wy groter dan 1), stelt U de verticale schaling in op Sy (default-waarde Sy=100), en geeft U het window een border b (default-waarde 0), dan wordt het aantal horizontale schaaldelen Sx gevonden uit:

$$Sx = 0.7380074 * Sy * (Wx - 4 * b - 1.5) / (Wy - 2 * b - 1)$$

Hierin stelt het getal 0.7380074 de aspect-ratio van de pixels in Mode 4 voor. De relatie geldt natuurlijk zowel in Mode 4 als in Mode 8. Overigens is het waarschijnlijk niet toevallig dat 0.7380074 handiger te schrijven is in de vorm 1/1.355. De reden voor de offset-waarden van Wx en Wy is mij echter volkomen onduidelijk (correctie voor de kromming van het monitorbeeld?). Wellicht dat andere QL-gebruikers hier een verklaring voor weten te bedenken.

- listing zoz -

Carlo Delhez - Emmastraat 3 - 4651 BV Steenberg - 01670-65114



LISTING VAN HET BASICPROGRAMMA DAT GEBRUIKT WERD OM BIJ DIVERSE WINDOW-AFMETINGEN (Wx,Wy) AUTOMATISCH S (EN DUS Sx) TE VINDEN:

```

100 REMark PIXELS-2-GRAPHICS scaling
110 REMark by Carlo Delhez, 02/91
120 :
130 MODE 4
140 OPEN_OVER #3,flpl_scales_dat
150 PAPER 0 : INK 7
160 :
170 FOR Wx=10 TO 40 STEP 10,50 TO 200 STEP 50,300,400,500
180 REMark Wx must be even
190 FOR Wy=10 TO 40 STEP 10,50 TO 250 STEP 50
200 PRINT #0,Wx,Wy,
210 WINDOW Wx,Wy,512-Wx,0
220 SCALE 1,0,0 : REMark Sy=1
230 Smin=0 : Smax=2
240 REPEAT find
250 S=(Smin+Smax)/2
260 Sx=S*Wx/Wy
270 IF Smin==Smax THEN EXIT find
280 CLS
290 FOR j=0,1
300 FILL j: LINE 0,0 TO Sx,0 TO Sx,1 TO 0,1 TO 0,0 : FILL 0
310 END FOR j
320 REMark Read upper-right pixel of screen:
330 p%=PEEK(131199) MOD 2
340 IF p% : Smax=S : ELSE : Smin=S
350 END REPEAT find
360 PRINT #3,Wx,Wy,Sx
370 FLUSH #3
380 PRINT #0,Sx
390 END FOR Wy
400 END FOR Wx
410 :
420 CLOSE #3
430 STOP

```



Onlangs bereikte mij de klacht dat het programma "dd><p" af en toe problemen had met het uitpakken van een P-file in N-files. Wat was het geval? Het bleek zich voor te doen bij files met namen waarin een asterix of een vraagteken voorkwam. Bij nader inzien was dat logisch, want de DD kent aan dit tweetal tekens een speciale betekenis toe ivm de zg 'Wild-Card Files'. Wanneer u in BASIC een file wilt SAVEN met een naam die een asterix bevat, en er staat al een file op de disk met een naam die tot aan de positie van die asterix hetzelfde is, dan vraagt de DD of u die al bestaande file wilt overschrijven. De MC van "dd><p" vraagt echter niets, gebruikt nog eens dezelfde naam en dus gaat het mis. Aangezien de andere opslagsystemen (nog) geen 'Wild-Card Files' kennen hadden we daar dus ook geen rekening mee gehouden.

Ook bij de andere opslagsystemen kan het voorkomen dat u een N-file uit een (ontvangen) P-file niet met de oorspronkelijke naam op uw disk wilt of kunt zetten. De naam kan al voorkomen of, bij de BD, kan dat met de eerste acht tekens ervan het geval zijn. De oplossing die ik de DD-klager voor zijn probleem verschaftte werkt gelukkig niet alleen bij "dd><p", maar bij alle conversieprogramma's "...><p". Ze bestaat uit de onderstaande drie BASIC-regels die gEMERGED kunnen worden, waarna u over een extra optie beschikt om filenamen in de P-buffer van uw omzetter te wijzigen voordat u de P-file uitpakt. (Deze optie komt niet in het menu.)

---

HET MERGEPROGRAMMA "rename><p" TBV "...><p"

```

8 FOR F=1 TO 10: PRINT CHR$ PEEK (B+F);: NEXT F:
PRINT " ";PEEK B,FN L(B);TAB 24;B: LET B=B+FN L(B):
IF B<A THEN GO TO 8

999 DIM N$(10): INPUT "ADRES",A,"NAAM",N$
1000 FOR P=1 TO 10: POKE A+P,CODE N$(P): NEXT P: RUN

```

---

#### DE WERKWIJZE

- LOAD uw omzetter, geef STOP en MERGE de regels in uw omzetter (of wijzig regel 8 als hierboven en voeg de andere twee toe).
- RUN naar het menu en LOAD een P-file met optie 3 (LOAD P).
- Maak zonodig met optie 6 (<> DRIVE) het drivenummer 0 (CR).
- Kies optie 5 (CAT 0). De toevoeging in regel 8 levert nu van elke file ook telkens het adres in de buffer (geheel rechts). Noteer het adres van de file waarvan u de naam wilt wijzigen.
- Geef BREAK na de CAT en daarna RUN 999. Regel 999 vraagt om het genoteerde adres en de nieuwe naam. Regel 1000 POKet de nieuwe naam op de juiste plaats, waarna het menu terugkeert.
- Stel zonodig met optie 6 (<> DRIVE) een ander drivenummer in.
- Pak met optie 2 (SAVE P>N) de P-file op uw disk of tape uit.



Naar ons nu pas bleek wordt bij de P-omzetter "dd><p" uit IMPULS 81-20 niet altijd de gewenste drive geselecteerd tgv een DD-bug. Dit kan heel nare consequenties hebben. Onder excuses geven wij u daarom hier een nieuwe MC waarbij deze fout niet kan optreden.

Overal waar "Cdd><p" voorkomt, al of niet in P-files, op DDiSCs, IMPULSOFT-cassettes en in SINCLAIR BOX, is deze reeds vervangen.

DE MC "Cdd><p" CODE 65E3,409 VOOR "dd><p" IN HEXDATA - JaRa -

1	"C3 0F FE C3.5D FE C3 13.FF C3 0D FF.C3 71 FF 01"	2502
2	"00 00 64 00.00 00 00 00.00 00 00 00.00 00 00 00"	2602
3	"00 00 00 00.00 00 00 CD.E6 FE 21 FC.FD DD 2A 76"	4210
4	"5C DD E5 D1.13 01 0A 00.ED B0 2A 07.FE DD 75 0B"	6056
5	"DD 74 0C 3A.06 FE DD 77.00 FE 00 28.15 FE 03 28"	7675
6	"07 3A 0B FE.DD 77 0E C9.2A 09 FE DD.75 0D DD 74"	9553
7	"0E C9 2A 0B.FE DD 75 0F.DD 74 10 2A.0D FE DD 75"	11428
8	"0D DD 74 0E.C9 DD 2A 76.5C DD E5 E1.23 11 FC FD"	13698
9	"01 0A 00 ED.B0 DD 6E 0B.DD 66 0C 22.07 FE DD 7E"	15441
10	"00 CB BF 32.06 FE 3C 32.FB FD FE 01.28 2A FE 04"	17354
11	"28 11 DD 7E.0E 32 0B FE.21 FF FF 22.0C FE 22 0D"	18977
12	"FE 18 2D DD.6E 0D DD 66.0E 22 09 FE.21 FF FF 22"	20855
13	"0B FE 23 22.0D FE 18 18.DD 6E 0D DD.66 0E 22 0D"	22232
14	"FE DD 6E 0F.DD 66 10 22.0B FE 21 CB.5C 22 09 FE"	24095
15	"DD 21 F7 FD.DD 7E 00 CF.21 CF 35 2A.76 5C 11 14"	25985
16	"00 19 EB ED.4B 07 FE CF.37 CF 38 CF.40 C9 DD 21"	28069
17	"F7 FD DD 7E.00 CF 21 CF.3B 11 06 FE.06 09 CF 3C"	29981
18	"12 13 10 FA.2A 76 5C 11.14 00 19 EB.ED 4B 07 FE"	31406
19	"CF 3D CF 40.C9 FD CB 47.86 18 10 FD.CB 47 C6 DD"	33793
20	"2A 76 5C AF.37 11 11 00.CD 56 05 3E.02 CD 01 16"	34897
21	"11 C0 09 2A.76 5C E5 E5.7E CB BF CD.0A 0C 06 0A"	36588
22	"23 7E D7 10.FB E1 11 14.00 19 DD E1.FD CB 47 46"	38561
23	"20 14 DD E5.DD 7E 00 F5.DD CB 00 BE.CD 70 09 F1"	40836
24	"E1 77 DD 2B.18 0F DD 5E.0B DD 56 0C.3E FF 37 E5"	42729
25	"DD E1 CD 56.05 DD E5 C1.C9 21 4C 6D.E5 11 4D 6D"	44965
26	"01 9B 90 36.00 ED B0 C1.C9"	46126

#### WELLICHT INTERESSANT VOOR MC-PROGRAMMEURS

Er wordt gebruik gemaakt van een zg UFIA, zie DD-manual page 58. Hiervan bevat de eerste byte, DSTR1, het gewenste drivenummer. Met het adres van DSTR1 in register IX wordt nu command-code 35h 'Open a file' gebruikt. Volgens page 56 brengt dit de filegegevens uit de UFIA naar het diskfilekanaalgebied DFCA over (33h), wordt een 'file sector map' conform deze DFCA-info geopend (34h) en de 9-bytes fileheader conform HD00-HD11 uit de UFIA verzorgd. In de nieuwe MC wordt bovendien hookcode 21h 'Select Microdrive' gebruikt, met het drivenummer in register A (zie IMPULS 63-52).



Als vier P-files te downloaden bij "SINCLAIR BOX" (01670-66845) en te koop op DUCDISK-21, DDISC-21 (de bekende bestelwijze) en IMPULSOFT-cassette 15 (HCC-bestelnummer 5454115, prijs f 10,-).

---

PIMP8-43	20 OD Xl&s:@	37 OD read:txt
	26 OD Xformatbt	Cread:txt
84-04 OD teas+od	27 OD bdss>od	39 SP C5000><
05 SP C711><	bdds>od	43 OD odrinst
C711mod><	35 OD lees:su	49 OD Xpar>ser+
14 SP Cquick	lees:vd	
16 OD blmovedemo	lees:stuvw	
83-06 SP tuwvsd	10 SP b/t>if1	26 OD flen<
Ctuwvsd	Cb/t>if1	flen><
09 SP convtkns	13 DD snap:s&l	30 SP fpr/int

---

PIMP8-21	14 MD Cmd><p	28 SP Ch64cpt1
	16 OD Xclear2.1	32 OD odpphex
82-12 SP reloc\$	18 SP toolkit	33 SP listvar0
Creloc\$	28 SP ln><cpt1	listvar1
14 MD md><p	Cln><cpt1	Cvars
81-06 DD Cdd-if1	23 SP cr><p	30 SP gens><t
Cdd-p>s	24 OD od><p	Cgens><t
16 OD Xcls	25 DD dd><p	33 OD odpeekpoke
Xrundr2	Cdd><p	38 SP C><sgif1
18 SP hsftbas	28 OD modtest	
22 SP Ccr><p	X*mod	

---

PIMPULS7	19 OD Xscrn\$#	24 OD Cmessage
	22 OD runherstel	39 DD autoldmenu
74-13 OD mc>odram	23 OD Clastdrive	
73-05 OD tr0herstel	07 SP Cvtx>s	09 OD Canker><
07 SP vtx>s	09 OD anker><	
72-05 OD dirlezer	55 SP Cswser	58 SP Cif3par
dir>\$	Ctw3ser	Cif3ser
cat>\$3k	56 OD tegek	62 MO 2dim\$>dm

---

PIMPULS6	23 BD l&s:tr0	29 DD peek @
	Cl&s:tr0	35 SP d>kol
64-05 OD diskdek	27 SP datr-st	36 SP t>kol
12 SP hexloader	29 DD peek d*	
63-05 SP list>s	11 SP rsl28bt	37 DD sys3d
62-15 DD cat #3	61-23 DD auto @cat+	55 DD scat&herst

---



In dit tweede deel krijgt u het aangekondigde programma waarmee u de LISTing van een BASIC-file kunt lezen zonder dat programma echt te LOADen, en een combinatie daarvan met de tekstlezers uit het eerste deel, waarnaar ik u verwijs voor een aantal gegevens.

HET BASICPROGRAMMA "lees:bas" LINE 5 - EdW -

```

1 LET I$=INKEY$#3: IF I$=CHR$ 14 THEN
  LET I$=INKEY$#3+ ... (5x) ... +INKEY$#3: GO TO 1
2 PRINT I$;: IF I$<>CHR$ 13 THEN GO TO 1
4 IF L THEN
  LET I$=STR$ (CODE INKEY$#3*256+CODE INKEY$#3):
  LET L=L-CODE INKEY$#3-CODE INKEY$#3*256-4:
  PRINT TAB 4-LEN I$;I$;: GO TO 1

5>CLOSE #3: INPUT "NAAM",N$;"DRIVE",D: OPEN #3;D;N$: CLEAR :
  POINT #3;6: LET L=CODE INKEY$#3+CODE INKEY$#3*256:
  GO SUB 4: GO TO 5

```

Regel 5 zet, na het OPENen, de BASIC-lengte uit de header in L. Een EOF-test is hier niet op zijn plaats, aangezien een BASIC-file immers ook, of zelfs alleen maar, variabelen kan bevatten.

Regel 4 onthoudt, zolang er nog BASIC is (L<>0), het regelnummer in I\$, trekt dan de regellengte af van L en PRINT vervolgens I\$ rechts op de eerste vier regelposities (dus evt spaties ervoor). Als enige uitzondering komt bij een regelnummer de hoge voor de lage byte. De volgende paar bytes bevatten de resterende regellengte t/m de afsluitende CR. De regellengte zelf is dus 4 meer.

Regel 1 leest telkens de volgende byte en zet die in I\$. In een BASIC-programma wordt elk getal gevolgd door een CHR\$ 14 en zijn FPR- of INT-voorstelling. Deze regel slaat al zulke bytes over.

Regel 2 PRINT elke andere byte uit een regel tot zijn afsluitende CR. Pas wanneer I\$ een CR bevat wordt regel 4 weer bereikt.

Omdat dit programma voor elk teken een aparte PRINT-opdracht gebruikt kunnen eventuele INK- en PAPER-kleuren, FLASH, BRIGHT en INVERSE in een LISTing niet weergegeven worden. Om dit wel mogelijk te maken moeten de gelezen tekens eerst worden opgeslagen in een string, die pas bij een CR in zijn geheel wordt gePRINT. Voor zo'n programma kunt u het vorige zo wijzigen en uitbreiden:

```

2 IF I$<>CHR$ 13 THEN LET P$=P$+I$: GO TO 1
3 PRINT P$
4 IF L THEN
  LET P$=STR$ (CODE INKEY$#3*256+CODE INKEY$#3):
  LET L=L-CODE INKEY$#3-CODE INKEY$#3*256-4:
  PRINT TAB 4-LEN P$;: GO TO 1

```



Daar de lengte van P\$ vooraf onbekend is wordt hij opgebouwd dmv concatenatie (+). Dit is een langzame stringoperatie vanwege het herhaaldelijk vervangen, verwisselen en verschuiven van P\$ en I\$ in het variabelengebied en het meeschuiven van gebieden erboven. Dat maakt dit programma zo'n tien procent trager dan het vorige.

HET BASICPROGRAMMA "lees:alles" LINE 8 - EdW -

```

1 LET I$=INKEY$: IF I$>=" " THEN PRINT I$;: GO TO 1
2 IF V$="" THEN GO TO 5
3 IF I$=V$ THEN PRINT : GO TO 1
4 GO TO 8 OR LEN I$

5 IF I$=CHR$ 14 THEN LET I$=INKEY$#3+ ... (5x) : GO TO 1
6 PRINT I$;: IF I$<>CHR$ 13 THEN GO TO 1
7 IF L THEN
    LET I$=STR$ (CODE INKEY$#3*256+CODE INKEY$#3):
    LET L=L-CODE INKEY$#3-CODE INKEY$#3*256-4:
    PRINT TAB 4-LEN I$;I$;: GO TO 1

8>CLOSE #3: INPUT "NAAM",N$;"DRIVE",D: OPEN #3;D;N$:
    CLEAR : LET V$=INKEY$#3: LPRINT ;:
    IF CODE INKEY$#3+CODE INKEY$#3*256+CODE V$=USR 432-4 THEN
        POINT #3;6: LET L=CODE INKEY$#3+CODE INKEY$#3*256:
        LET V$="": GO SUB 7: GO TO 8

9 LET I$=CHR$ (8 OR V$<>CHR$ 3):
    POINT #3;CODE I$: LET V$=INKEY$#3:
    IF CODE V$ THEN
        LPRINT ;:POINT #3;CODE I$+USR 432: LET V$=INKEY$#3:
        IF V$>=" " THEN LET V$=CHR$ 0
10 POINT #3;CODE I$: GO SUB 1: GO TO 8

```

Regel 8 slaat de eerste byte in V\$ op en geeft LPRINT ; om hier-na USR 432 te kunnen gebruiken. Neem even aan dat het BASIC is. In het linkerlid van de voorwaarde wordt uit de header de file-lengte achter de header gelezen, waarbij nog 0 (typebyte) wordt opgeteld. Er zijn nu 3 van de 7 headerbytes gelezen, dus USR 432 in het rechterlid geeft nog maar 4 bytes meer aan dan de gelezen filelengte achter de header. Voor gelijkheid moet daar dus 4 af. Daarna wordt de pointer op de BASIC-lengte in de header gericht. Nadat die gelezen en in L gezet is wijst de pointer precies naar de eerste BASIC-regel (als er tenminste BASIC-regels voorkomen). Tot slot wordt V\$ leeg gemaakt om BASIC aan te geven bij de verdere verwerking, die bijna net zo geschiedt als in "lees:bas". Is de conditie niet vervuld, dan vervolgen we met CODE of DATA.

Regel 9 slaat de pointerpositie voor het tekstbegin mbv V\$ op in I\$, richt de pointer daarop, leest de eerste byte en plaatst die in V\$. Is het geen t-file, dan moet V\$ de laatste byte bevatten. Daarvoor moet de pointer gericht worden op USR 432, verhoogd met



De volgorde die de SP hanteert bij het evalueren (zeg maar het uitrekenen) van een expressie ligt vast in 'prioriteitsregels'. Deze staan in de SP-handleiding op bladzijde 205 of op page 155. U ziet daar dat de logische functies NOT, AND en OR de laagste prioriteit bezitten, dwz dat alle andere functies en operaties eerder worden uitgevoerd, tenzij haakjes dat anders aangeven. Van deze drie logische functies heeft NOT de hoogste prioriteit. Wanneer u zich dat niet goed realiseert kan dit aanleiding geven tot fouten; ik heb mij tenminste wel eens vergist. Bijvoorbeeld:

IF A OR NOT B THEN ... en IF NOT B OR A THEN ...

hebben dezelfde werking, want NOT gaat voor OR. Maar dat is bij

IF A+NOT B THEN ... en IF NOT B+A THEN ...

niet zo, aangezien nu de NOT rechts pas na B+A wordt uitgevoerd!

- EdW -

>>> VERVOLG VAN DE VORIGE BLADZIJDE >>>

CODE I\$, het aantal der bytes die inmiddels al zijn gepasseerd. (Na gebruik van POINT moet LPRINT ; tbv USR 432 herhaald worden) Bij een s-file is deze slotbyte hoger dan 31. Dan wordt er in de vergelijkingsstring V\$ een CHR\$ 0 gezet. Deze komt niet voor, en kan dus later niet ten onrechte voor een nieuwe regel zorgen.

Regel 1 leest telkens een teken, plaatst het in I\$ en PRINT het als het een spatie of een hoger teken is; de meeste tekens dus.

Regel 2 geeft een sprong naar regel 5 als het een BASIC-file is.

Regel 3 begint een nieuwe tekstregel als aan I\$=V\$ is voldaan.

Regel 4 geeft een sprong terug naar regel 1 indien I\$<>"", maar anders door naar herstartregel 8, aangezien dan EOF bereikt is.

Regel 5 t/m 7 verwerken andere BASIC-tekens zoals in "lees:bas".

Al deze programma's bevatten zo weinig mogelijk variabelen, voor een zo groot mogelijke executiesnelheid. Voor de duidelijkheid bevatten ze alleen datgene wat voor de goede werking ervan noodzakelijk is, dus niets voor kleur- en geluidseffecten. Toeters en bellen mag u er zelf aan toevoegen wanneer u dat leuk vindt.

E H F Weijgers - H Marsmanlaan 29 - 2624 TJ Delft



--

## DE INHOUD VAN IMPULS 91 - DE SGG- EN HCC-DAGEN

--

02	COLOFON .....	--
03	VAN DE REDACTIE .....	--
04	DE INHOUD VAN EEN DISCIPLE-DISK .....	DD
06	HET SYSTEEM VAN CHANNELS EN STREAMS .....	SP
09	DISKSUBTABEL EN ANKERBLOK .....	OD
10	TRANS 22.2 - VAN N-FILES NAAR P-FILES EN TERUG .....	BD
13	FORMATTEEN MET 1 OF 2 ZIJDEN EN 40 OF 80 SPOREN .....	OD
14	SINCLAIR QL IN FIDONET .....	QL
17	HET VTX711-RS232-INTERFACE - DEEL 2 .....	SP
18	ONZE NIEUWE SGG-BANK VOOR DISCIPLE EN PLUS-D .....	DD
21	DD>OD - VAN DISCIPLE- NAAR OPUSDISK .....	OD
24	VAN GEKOPIEERDE DD-MD-FILE NAAR DATAFILE .....	OD
25	VAN MD-FILE NAAR OPENTYPE OF ANDERSOM .....	DD
26	PIXELS-2-GRAPHICS SCALING .....	QL
29	TRANS 22.3 - RENAME IN P-FILES VOOR ALLE P-OMZETTERS .....	SP
30	AD TRANS 22 - NIEUWE MC VOOR DD><P .....	DD
31	P-FILES MET IN IMPULS GEPUBLICIEERDE PROGRAMMA'S .....	SP
32	PROGRAMMA'S OM TEKST- EN BASICFILES TE LEZEN - DEEL 2 .....	OD
34	NOTA BENE NOT .....	SP
35	DE INHOUD VAN IMPULS 91 - DE SGG- EN HCC-DAGEN .....	--

-- ALGEMEEN

80 ZX80

81 ZX81

CR CASSETTERECORDER

SP ZXSPECTRUM

MD MICRODRIVE

OD OPUS DISCOVERY

BD BETA DISK

DD DISCIPLE

QL QUANTUM LEAP

88 Z88

PC SINCLAIR PC

## DE SINCLAIRDAGEN IN HOUTEN

ZATERDAG

10-16 UUR

18 MEI

22 JUNI

24 AUGUSTUS

26 OKTOBER

## DE HCC-MICROCOMPUTERDAGEN

VRIJDAG

ZATERDAG

DE JAARBEURS

29 NOVEMBER

30 NOVEMBER

TE UTRECHT

ONDER VOORBEHOUD - BEKIJK STEEDS DE AGENDA IN DE HCC-NIEUWSBRIEF



SINCLAIR IMPULS

Postbus 76  
2260 AB Leidschendam

PORT BETAALD  
PORT PAYE  
DEN HAAG

