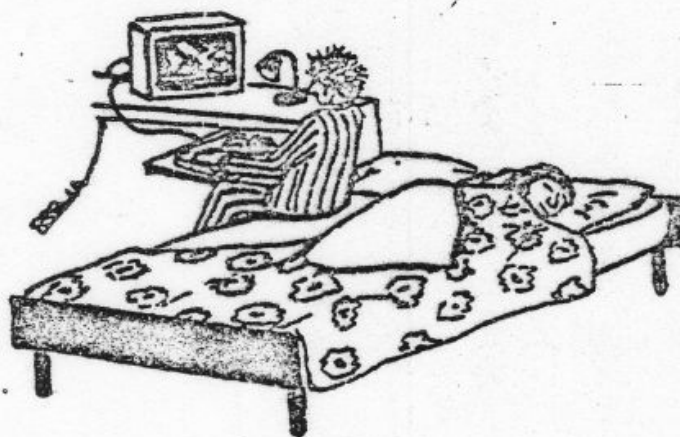


# DISCOVERY USERS CLUB

*The Netherlands*

Secretariaat :  
D.C.Kruihof  
Boelerkade 6  
2725 CH ZOETERMEER  
Tel: 079-416360

NUMMER 3 (HERDRUK)



# INHOUD

\*\*\*\*\*

Hedrick  
↓

|   |          |
|---|----------|
| COLOFON .....                                 | P. 1     |
| AFSCHEID OPRICHTER D.U.C. ....                | P. 2     |
| GEBRUIKERSDAG .....                           | P. 2     |
| PROBLEEM PAGINA .....                         | P. 3     |
| SMITH-CORONA FASTEXT 80 .....                 | P. 3     |
| DISCOVERY UTILITIES .....                     | P. 4     |
| NIEUWS .....                                  | P. 6     |
| TRANS EXPRESS en vreemde verschijnselen ..... | P. 7     |
| SCREENCOPIER (listing van de maand) .....     | P. 10    |
| DISCOVERIES (ingezonden advertenties) .....   | P. 11 12 |
| INGEZONDEN LISTINGS .....                     | P. 12 14 |
| OPMERKINGEN BETREFFENDE TASWORD 2.2 .....     | P. 13 14 |
| WERKGROEP KOMMUNIKATIE .....                  | P. 14 17 |
| REACTIES OP D.U.C. 2 .....                    | P. 16 19 |
| IC 6116 .....                                 | P. 19 20 |



Het magazine van de Nederlandse DISCOVERY USERS CLUB.

#### SAMENSTELLING BESTUUR

Interim voorzitter / penningmeester : R.O. Aalders  
Oprichter / erelid : D.C. Kruithof

#### REDACTIE DUC

H. Kramer  
M. Kuklewski

#### HULPLIJN

Rob Macare : tel. 010 -4862184 (tussen 1900 en 2330)

#### LEDENADMINISTRATIE

Dick Kruithof  
Boeierkade 6  
2725 CH Zoetermeer

#### ADVERTENTIES

Commerciële advertenties kunnen worden opgenomen in het magazine. Er zijn echter een aantal voorwaarden.

- 1) Advertenties moeten betrekking hebben op SINCLAIR producten of OPUS DISCOVERY.
- 2) Advertenties worden vooruit betaald. (zie LIDMAATSCHAP voor gironummer)
- 3) Kosten daaraan verbonden gaan uit van f 40,- per pagina.
- 4) Betalende leden mogen kleine advertenties plaatsen in het rubriek 'DISCOVERIES'. Hieraan zijn geen kosten verbonden maar wel is voorwaarde (1) hierop van toepassing.

#### ADVERTENTIE-ADRESS

D.U.C. Redactie  
Urkwil 92  
1324 HR Almere

#### LIDMAATSCHAP

Clubcontributie is vastgesteld op f 6,- per blad. Betaling dient per 3 bladen te gebeuren. (zie BETALINGEN)

#### NABESTELLINGEN

Nabestellingen van DUC kosten f 7,50 per blad.

#### BETALINGEN

Alle betalingen dienen te worden gegireerd op gironummer 5241782, t.n.v. R.O. Aalders  
Furkabaan 625  
3524 ZE Utrecht

#### AANSPRAKELIJKHEID

Het bestuur en redactie is niet aansprakelijk voor eventuele schade door ingezonden programmas, tips, etc. Eveneens is het bestuur noch redactie aansprakelijk voor het overtreden van de Copyright door leden.

#### COPYRIGHT

Op alle uitgaven van DUC rust COPYRIGHT van de vereniging. Voor overname van artikels is schriftelijke toestemming verplicht van R.O. Aalders (zie betalingen).







\*\*\*\*\*

- [illegible]

De benamingen in het kader op de help pagina (trouwens ook andere zaken) worden als DATA door de machine-code ingelezen .

Dit wetende zoek ik op waar die DATA staat en ben begonnen mijn eigen text erin te zetten . Wil jij dat ook doen , dan heb je hier alles betreffende de DATA ( '.' is een spatie ) :

Veranderen printernaam :

EpsonFX80.....staat op 55020 t/m 55030

Veranderen GRAPHISCHE acties :

vergroot.....staat op 55146 t/m 55159

```
verkleind.....staat op 55274 t/m 55287
```

onderstreept.....staat op 55402 t/m 55415

schuinschrift....staat op 55530 t/m 55543

extra vet.....staat op 55658 t/m 55671

```
schaduwscript...staat op 55786 t/m 55799
```

eliteschrift.....staat op 55914 t/m 55927

proportioneel... staat op 56042 t/m 56055

Om nu je eigen text erin te zetten , doe het volgende :

-CLEAR 54783

-laad de code van TASWORD

```
-typ het volgende programma in :
```

```
1 READ a, z$
```

```
2 IF a<1 THEN SAVE *1;"TWCODE "CODE 54784,10751 : STOP
```

```
3 FOR b= 1 TO 14
```

```
4 IF b<LEN z$ THEN POKE (a+b-1),CODE z$(b):GOTO 6
```

```
5 POKE (a-1+b),32
```

```
6 NEXT b :GOTO 1
```

7 DATA.....

Ik denk dat dit naar behoren zal werken .Veel plezier ermee !!

Rudie Aalders .

\*\*\*\*\*

Als je nog meer leuke ideeën hebt voor TASWORD (bv. een TASCOUNT zoals gepubliceerd in YOUR SPECTRUM nr.15) ; dan moet je niet verlegen zijn ! Gewoon opsturen !!!!!!!

Ales wat maar een verbetering inhoudt van een programma of een situatie, is gewoon fantastisch om met iedereen van de club te delen !

\*\*\*\*\*

P.S.

Er wordt in het DISCOVERY handleiding niet gesproken over het B-kanaal . Het B-kanaal is een P- of T-kanaal , maar dan voor losse bytes . Dar de printerpoort op het T-kanaal de controlcodes van de printer niet doorlaat en o het B-kanaal wel, raadt ik iedereen aan het B-kanaal bij TASWORD te gebruiken !!!

[illegible]

\*\*\*\*\*

Enige tijd geleden zijn er enkele Discovery utilities verschenen die door Sander Plomp geschreven zijn. In deze utilities zat echter een 'bug': ze waren alleen geschikt voor een printer-interface die de TAB-functie herkend. Helaas is dit met de Discovery-interface niet het geval zodat het uitprinten problemen gaf. Hij heeft echter deze programma's herschreven en ze werken nu rondt fantastisch. Hieronder een lijstje van de -tot nu toe- beschikbare programma's:

\*supercat : een utility om alle gegevens uit de Catalogue te halen en op beeld/printer weer te geven zoals:

- naam (ook als ze met CHR\$ 0 beginnen)

- start(adres/regel) en lengte van programma in bytes en blokken.



- lengte van programma en variabelen apart
- eerste en laatste blok en hoeveel bytes er in is geschikt voor iedere 80Kol printer (ook daisywheel)

\*smallcat : naam, startregel/adres en lengte programma. Geschikt voor elke 32 kol. printer (ZX + Seikosha 508 )

\*backup + : maak nu een tape back-up van uw belangrijke schijffjes. Saved in blokken van 32K de data van de disk naar tape. Bv. volle disk save? Slechts 12x cassette recorder bedienen ongeacht de hoeveelheid programma's.

\*probe 1 : probe 1/2 : kraakprogramma om niet al te lastige programma's vakkundig door te lichten en al te beveiligingen weg te werken. Probe 1 is voor tape en Probe 2 voor diskprogramma's. Incl. handleiding

\*disk-disk Copieer uw programma's tot 32K van disk naar disk u krijgt geen lamme hand van disk in en uit halen. Werkt met gebruik van RAM-disk.

Diegenen die in het bezit zijn van de oude uitvoering van de programma's, hebben recht op een (gratis) update. Stuur schijfje MET RETOURPORTO naar onderstaand adres met vermeld van 'UPDATE UTILITIES'.

Diegenen die belangstelling hebben om deze programma's te kopen kunnen op twee manieren aan deze programma's komen:

- 1) Maak een bedrag over van F 32,50 op onderstaand gironummer u ontvangt een schijfje + programma's + handleiding thuis.
- 2) Zend mij een schijfje + een betaalkaart van F 17,50 (ondertekend en ingevuld!) en u ontvangt het schijfje + programma's + handleiding thuis.

Gironummer 2706694 of  
Banknummer 41.04.65.321

tav. D.Kruihof  
Boeierkade 6  
Zoetermeer

Voorbeeld supercat:

WORKD. E/H Ruimte voor 112 entries 7 0

| file naam  |             | start | len   | prog  | vars  | blk's | frst |
|------------|-------------|-------|-------|-------|-------|-------|------|
| supercat   | LINE        | 1     | 4158  | 4158  | 0     | 17    | 7    |
| smallcat   | LINE        | 1     | 3453  | 3453  | 0     | 14    | 24   |
| *smallcat  | LINE        | 1     | 4101  | 3751  | 350   | 17    | 237  |
| *supercat  | LINE        | 1     | 5251  | 5219  | 32    | 21    | 254  |
| progs-file | LINE        | 100   | 40026 | 13309 | 26717 | 157   | 275  |
| (O) (CAT)  | LINE        | 1     | 5524  | 5202  | 322   | 22    | 432  |
| datafile   | DATA f\$( ) |       | 4     |       |       | 1     | 454  |

0 6 256 1792 0  
...frst last: use files : nr









Een zeer goede kant (de enige) van de TAPE-DISC-optie is de mogelijkheid om (via "O") programma's die AUTO-RUNNING zijn van deze optie te ontdoen. Hierdoor is het mogelijk programma's die NOT-MERGEABLE zijn op disc te zetten en ze daarna gewoon als dood programma te laden om ze wat beter te bekijken.

+DE DERDE REDEN is het feit dat de instructies zeer povertjes zijn. Bij het uitproberen van een (overgezette) programma kon 't de normale LOAD-instructie nog gemakkelijk overzetten naar een disc-LOAD. Maar andere weggetjes voor het binnenhalen van programma's werden rustig ingeslagen dus na BASIC-LOADER en SCREEN\$ stond mijn BORDER weer rustig te knippen. Hoewel het programma claimt HF aan te kunnen, zou het wel leuk zijn als ze ze ook oproepen kan van disc. In zo'n geval zouden goede instructies een uitkomst zijn.

Toch wil ik een pluim uitdelen naar het team welke TE in elkaar gedraaid heeft. De DISC-DISC-optie is excellent gewoon, hoe wel het overzetten met slechts een enkele drive wel een gedoe is met het in- en uitschuiven van de disc mbt. de drive. Dan had ik maar een dubbele drive moeten hebben!!!!

Maar nu de diepere gronden van TE. Op het moment dat ik met de TAPE-DISC-optie aan het rotzooien was en de knipperende BORDER op mijn scherm getoverd had ben ik dus gaan zoeken naar wat het programma gedaan had. De instructies boden totaal geen uitkomst dus was het tijd voor mijn "oude" SEIKOSHA GP50S en het MONITOR-programma van PICTURESQUE (16K-versie).

De CAT was het interessantste gedeelte. In plaats van de drie onderdelen (BASIC-SCREEN\$-HF) stonden er vijf! Snel nagaand was de HF gesplitst in twee delen, maar dan was er nog een deel teveel op disc gezet. Dus je neemt de eerste BASIC-LOADER en kijkt wat er veranderd is.

Wat ik zag was niet te beschrijven: twee regeltjes BASIC. De eerste begon met een REM-STATEMENT waarna een zwart balk zich enige regels in stand hield om te eindigen met knipperend "?K". In deze regel was dus een stuk CODE opgeslagen.

De tweede regel was interessanter.

CLEAR <getal>: RANDOMIZE USR 23760.

Dus bevond zich inderdaad in de eerste regel de CODE. Dit bevestigde alleen mijn vermoedens dus.

Mijn MONITOR-programma werd opgestart. Er kwamen verrassende zaken naar voren, na veel gepuzzel. De eerste tien BYTES verplaatsten de rest van de CODE naar lokatie 65280, en iets verderop ging de CODE verder.

Het volgende kreeg ik onder ogen :

```

5CDD LD HL,(5C53) ; HL:= begin BASIC-programma
5CD3 LD BC,0017 ; BC:= 23
5CD6 ADD HL,BC ; tel BC bij HL op
5CD7 LD DE,FF00 ; DE:= 65280
5CDA LD BC,007F ; BC:= 127
5CDD LDIR ; verplaats wat volgt naar (DE)
5CDF JP FF1A ; spring erachteraan
FF00 5CE2 LD A,48 ; A:= CODE "O"
FF02 5CE4 LD (FF7B),A ; Zet deze waarde als laatste in naam
FF05 5CE7 LD A,AF ; A:= CODE "CODE "
FF07 5CE9 LD (FF7D),A ; Zet deze achter LOAD-instructie
FF0A 5CEC POP HL ; ?
FF0B 5CED LD (FF62),HL ; zet momentane waarde HL weg op FF62H
FF0E 5CF0 LD DE,FF46 ; DE:=65350
FF11 5CF3 LD HL,(5C3D) ; HL:=(ERR_SP)
FF14 5CF6 LD (HL),E ; ) Daar waar (ERR_SP) naar wijst komt

```



|                |           |  |
|----------------|-----------|--|
| FF15 5CF7 INC  | HL        | ; ) de waarde van DE te staan. Igv een |
| FF16 5CF8 LD   | (HL),D    | ; ) ERROR springt alles naar FF46      |
| FF17 5CF9 JP   | FF2E      | ; )                                    |
| FF1A 5CFC LD   | HL,1238   | ; ) Een stuk wat de routine kan ge-    |
| FF1D 5CFF LD   | DE,FF7F   | ; ) bruiken uit S-ROM wordt opgehaald. |
| FF20 5D02 LD   | BC,004F   | ; ) Dit is niet een komplette routine, |
| FF23 5D05 LDIR |           | ; ) een RETURN komt er achter          |
| FF25 5D07 LD   | A,C9      | ; A:= RETURN-NMEMONIC                  |
| FF27 5D09 LD   | (DE),A    | ; A wordt achter ROM-CODE gezet        |
| FF28 5D0A LD   | HL,F883   | ; HL:=64816                            |
| FF2B 5D0D CALL | FF7F      | ; Roep ROM-routinetje aan              |
| FF2E 5D10 CALL | 16B0      | ; Minimaliseer EDIT-gebied (CLEAR)     |
| FF31 5D13 LD   | BC,0017   | ; BC:=23                               |
| FF34 5D16 LD   | HL,(5C59) | ; HL:=adres cursor-lijn                |
| FF37 5D19 INC  | HL        | ; )                                    |
| FF38 5D1A PUSH | BC        | ; )                                    |
| FF39 5D1B CALL | 1655      | ; ROM- "MAKE ROOM" voor executie       |
| FF3C 5D1E POP  | BC        | ; )                                    |
| FF3D 5D1F EX   | DE,HL     | ; ) Zet het BASIC-gedeelte op juiste   |
| FF3E 5D20 LD   | HL,FF68   | ; ) plaats in geheugen                 |
| FF41 5D23 LDIR |           | ; )                                    |
| FF43 5D25 JP   | 12CF      | ; )                                    |
| FF46 5D28 JP   | FF56      | ?                                      |
| FF49 5D2A LD   | DE,(FF64) | ?                                      |
| FF4D 5D2F LD   | HL,5CB6   | ?                                      |
| FF50 5D32 LD   | BC,(FF86) | ?                                      |
| FF54 5D36 LDIR |           | ?                                      |
| FF56 5D38 LD   | HL,1303   | ?                                      |
| FF59 5D3B PUSH | HL        | ?                                      |
| FF5A 5D3C LD   | HL,1B76   | ?                                      |
| FF5D 5D3F PUSH | HL        | ?                                      |
| FF5E 5D40 LD   | HL,(FF62) | ?                                      |
| FF61 5D43 JP   | HL        | ?                                      |
| FF62 5D44 HALT |           | ?                                      |
| FF63 5D45 DEC  | DE        | ?                                      |
| FF64 5D46 OR   | (HL)      | ?                                      |
| FF65 5D47 LD   | E,H       | ?                                      |
| FF66 5D,8 DJNZ | FF78      | ?                                      |
| FF68 5D4A DEFM | LOAD      | ## FF74 5D56 DEFM Y                    |
| FF69 5D4B DEFM | *         | ## FF75 5D57 DEFM H                    |
| FF6A 5D4C DEFM | "         | ## FF76 5D58 DEFM U                    |
| FF6B 5D4D DEFM | m         | ## FF77 5D59 DEFM N                    |
| FF6C 5D4E DEFM | "         | ## FF78 5D5A DEFM T                    |
| FF6D 5D4F DEFM | ;         | ## FF79 5D5B DEFM E                    |
| FF6E 5D50 DEFM | SGN       | ## FF7A 5D5C DEFM R                    |
| FF6F 5D51 DEFM | PI        | ## FF7B 5D5D DEFM O                    |
| FF70 5D52 DEFM | ;         | ## FF7C 5D5E DEFM "                    |
| FF71 5D53 DEFM | "         | ## FF7D 5D5F DEFM <SPATIE>             |
| FF72 5D54 DEFM | S         | ## FF7E 5D60 DEFM <ENTER>              |
| FF73 5D55 DEFM | P         | ## FF7F 5D61 NOP                       |

Wat alles precies doet weet ik nog niet. Een paar vreemde zaken zijn me echter opgevallen:

1. De aanwezigheid van de BASIC-regel in de CODE.
2. Wat er voor deze CODE staat is volslagen niets. Het programma echter verandert nog wat in dat stuk. Maar wat moet ik nog uitvinden, en het zal heel wat moeite kosten.

Er zijn twee entry-punten in deze routine: op FF1A en FF00. De eerste entry is voor als er een BASIC-routine binnengehaald moet worden, de tweede als dit een CODE-programma is.

Toch is dit niet het belangrijkste van alles . Aan het einde ga ik TE gebruiken voor iets dat belangrijker is voor de club : ik ga (proberen) in te breken in de DISC-DISC optie van TE , om een beter begrip te kweken van hoe het werkt , en op slag binnen te zijn wat betref MACHINE-CODE instructies wat betreft de disc-handelingen .

[illegible]

\*\*\*\*\*

MIXER had tot doel om vier verkleinde plaatjes tot 1 grote te mixen, in acht nemend de plaats waar de kleine plaatjes op het grote scherm stonden.

- MEGABASIC werkt niet goed op ons systeem. We hebben dus onze "vertrouwde" DISC-OPDRACHTEN niet bij de hand. Alles wat we verrichten gaat met CASSETTES. En bij de cassette-handelingen verliezen we de controle over de onderste 2 regels.

Dit alles is omslachtig. Je kan beter een routine hebben die het hele proces zelf in de hand houdt. En die heb ik ook geschreven.

Ik moet zeggen dat de routines snel zijn. Voor een scherm verkleinen en de verkleinde versie links bovenin het scherm te zetten is minder dan 4 seconden nodig.

Voor het wegzetten naar DISC van 5 schermen, het laden van 8 schermen en 4x beide routines laten lopen heeft het geheel niet meer dan anderhalve minuut nodig. In het geval van MEGABASIC zou



je ten tijde van dezelfde handelingen koffie kunnen gaan drinken bij de burelen.

Het verkleinde scherm is opgeborgen in bytes 51001 t/m 52536.

De routines heb ik laten beginnen op 50000; er is nog ruimte over om desnoods een heel scherm boven het geheel op te slaan. Voor mij is dat ook heel belangrijk want in een ander programma van mij moet ik de gelegenheid hebben hele schermen op te slaan. Misschien heeft U ook van zulke programma's.

Het programma.

U tikt Listing 1 in en SAVET het onder een eigen naam naar disc, laat het programma wel automatisch starten op regel 9999. Indien U klaar bent met de eerste listing tikt U de HEXALADER in (listing 2). U RUNT deze en geeft de regels uit listing 3 in als INPUT. Bij foutieve INPUT meldt het programma dat en U krijgt nog een kans. De CODE SAVET zichzelf naar disc.

#### Listing 1 (Hoofdprogramma).

```
1 GOSUB 1000 : GOSUB 1000
2 GOSUB 1000 : GOSUB 1000
3 SAVE *1;"KLEINER" SCREEN$
4 GOTO 4
1000 SAVE *1;"KLEINER" SCREEN$ : CLS : CAT 1
1010 INPUT "WELK SCHERM ? " : a$
1020 LOAD *1;a$ SCREEN$
1030 RANDOMIZE USR 50000
1040 LOAD *1;"KLEINER" SCREEN$
1050 RANDOMIZE USR 50175
1060 READ a1,a2,b1,b2,c1,c2
1070 POKE 50182,a1 : POKE 50183,a2
1080 POKE 50188,b1 : POKE 50189,b2
1090 POKE 50194,c1 : POKE 50195,c2
1100 RETURN
2000 DATA 16,64,144,64,16,72
2001 DATA 128,72,0,80,128,80
2002 DATA 144,72,16,80,144,80
2003 DATA 0,64,128,64,0,72
9999 CLEAR 49999: LOAD *1;"MC" CODE
```

#### Listing 2

```
1 LET a=10: LET b=11: LET c=12 LET d=13: LET e=14: LET f=15
2 CLEAR 49999
3 FOR z=50000 TO 50240 STEP 16
4 let tot=0
5 PRINT z;" ";
6 INPUT b$: IF LEN b$ <> 36 THEN GOTO 6
10 FOR y=0 TO 15
11 LET p=16* VAL b$(1)+ VAL b$(2)
12 LET tot=tot+p : POKE (z+y),p
13 LET b#=b$(3 TO ) : NEXT y
15 LET check=0: LET b#=b#+
16 FOR y=1 TO 4
17 LET check=16*check+ VAL b$(1): LET b#=b$(2 TO )
18 NEXT y : IF check <> tot THEN BEEP 1,1: GO TO 6
19 PRINT check
30 NEXT z
50 SAVE *1;"MC" CODE 50000,256
```





- \* Wie kan A.L. meyer helpen met het opzetten van wiskundige en goniometrische programma's ?  
Adres : Meidoornstraat 13 , 2371 VC , Roelofarendsveen
- \* Wie kan Onno Witkop helpen ? Listing met BETA BASIC gaat soms vreemd bij hemals de printer staat afgesteld met 'ORND32' en list format 2; lange regels worden soms wel en soms niet vanaf het begin geprint . Adres : Rembrandtlaan 36 , 7545 ZH Enschede .
- \* Alexander Sramek lukt het niet files van FINANCE MANAGER te printen . Wie helpt hem daarbij ? Adres : Trompenbergweg 2 , 1217 BE , Hilversum .
- \* Wie kan J.H. Buisman helpen : de printer-aanpassingen van AACKO-GROOTBOEK , VU-FILE en VU-CALC werken niet . Adres : Vlietstraat 35 , 4254 AD , Sleeuwijk .
- \* Gezocht : programmas om elektrotechnische problemen op te lossen . Reacties aan P.Vermaas , Nachtegaallaan 20 , 3362 NL Glijedrecht .
- \* Wie kan Jaap vn Doorn helpen met het overzetten d.m.v. save van zelfgemaakte programmas in HISOFT PASCAL .Adres : St. Janstraat 3 , 4901 LR , Oosterhout .
- \* Gezocht : hulp voor het maken van screendumps in machine -code Hennie van Vliet , van Ostadestraat 315 III , 1074 VV A'dam .
- \* Gezocht in telefoondistrict 030 : DISCOVERY-gebruiker met modem . Adres : Rudie Aalders , Furkabaan 625 , 3524 ZE , Utrecht .
- \* Te koop aangeboden : Opus Discovery 1 , 10 discs met o.a. Tasword 2 (Nederlandse versie) , Trans Express , Beta Basic 1.9 , Aackocalc , een auto-onkostenprogramma , een databank en 22 spelletjes . Elke schijf is voorzien van een re-organisatie-programma en het laden van elk programma is menu-gestuurd . Bijgeleverd word een Centronics kabel en een cassette waarin de discs opgeborgen zijn . Ook bij de prijs inbegrepen is een programmeerbare joystick interface (incl. software) . Prijs : Hfl 750,- . Tel. intl. H. Kruse 02152 - 56556 .
- \* Te koop aangeboden :Spectrum 48K met DK-Tronics toetsenbord , Opus Discovery 1 en de nodige kabels .Veel serieuze software , boeken en tijdschriften . Prijs : Hfl. 1500,- .Tel. intl. 020 - 120923 (tussen 18.00 en 1900 uur) .
- \* Te koop : TASWORD III (nederlandse versie) op 3.5" , Bij minimale belangstelling van 25 personen prijs : Hfl. 75,- . Adres : Rob Macare , Gaesbeekstr. 77 , 3081 NE Rotterdam .
- \* Gevraagd : de oscilloscop uit het blad HOBBIT (7/8/'84) . Een foto van de print of de print zelf zou ook welkom zijn . K. Zondervan , Smaragd 8 , 2651 RW , Berkel-Rodenrijs
- \* Te koop : speciale doosjes voor het verzenden van discettes , Hfl. 3,10 per stuk bij afname van 10 stuks . Als er in totaal 200 of meer besteld worden bij onze voorzitter , zullen ze nog goedkoper uitvallen .!! U kunt ze ook zelf bestellen : Fa. Pressel , B.J. Blomstraat 1 , 2225 HL Katwijk .  
Tel. 01718 - 73232 .





Voorts is het aantal files in een kolom teruggebracht van 22 naar 21. Dit komt door een SPECTRUM-bug. Ik ging ervan uit dat bij de opdracht "PRINT #0;" slechts de 22-ste en 23-ste regel van het scherm gebruikt zouden worden. Niets is minder waar. Ook regel 21 wordt schoongeveegd. We moeten daarom TASWORD 2.2 op drie plaatsen verbeteren :

REGEL 5: in de definities van de functies q en r moet 22 vervangen worden door 21.

REGEL 1019: hierin staat IF PEEK VAL"23296"=VAL"65". Deze 65 moet vervangen worden door 62.

#### OVER VERSIE 2.2.

Er was eerst sprake van een versie 2.1, degene die die versie aangevraagd hebben ontvingen echter de versie 2.2! De nieuwe versie was compacter dan de vorige, sneller door beter BASIC (en het feit dat enkele subroutines naar voren geplaatst waren) en WORD-COUNT was geïmplementeerd.

Karakteristiek aan TASWORD 2.2 was echter zijn neefje: een BASIC-programma genaamd TWHULP. Dit neefje is noodzakelijkerwijze erbij gekomen omdat anders ons TASWORD niet meer in zijn BASIC-huisje paste. Omdat de DISCOVERY informatie betreffende zijn annalen kwijt moet, kon TASWORD niet veel meer groeien. Dan maar een neefje maken !

#### OVER VERSIE 2.3 (DE NABIJE TOEKOMST).

Wat U nu leest is verwerkt met de allernieuwste versie van TASWORD, de versie 2.3. Met deze versie is de lengte van TASWORD weer iets groter geworden helaas! We zitten nog niet tegen de limiet maar ik denk dat bij de volgende versie van TASWORD (2.4) zijn neefje groter gegroeid is.

TASWORD 2.3 wordt voorlopig niet vrijgegeven. Ze stikt op dit moment van de copyright en ik wil eerst bepaalde zaken goed geregeld hebben voordat ze wordt vrijgegeven.

Ook moet deze versie van TASWORD eerst grondig getest worden want we spelen met deze versie op de rand van het kunnen van TASWORD (wat dit moment betreft). Ik wil niet achteraf weer een paar fouten eruit moeten halen zoals bij de vorige versies. Wel weet ik dat ze op een speciale manier straks gedistribueerd moet

den.

Wat kan deze nieuwe versie dan meer?

Er zijn veel veranderingen aangebracht in deze nieuwe versie.

Een belangrijke verandering is dat bij het uitprinten van een textfile nu gedacht wordt aan paginering. Paginanummers met HEADERS of zonder behoren ook tot de mogelijkheden.

Een tweede belangrijke verandering is dat WORD-COUNT automatisch gegeven wordt. Dit zorgt wel voor een kleine vertraging bij het hoofd-(STOP-)menu maar dat moet maar op de koop toe genomen worden.

De derde verbetering heet FLEXIBEL.

FLEXIBEL stelt je in staat je TASWORD-files in een compactere vorm om discette op te slaan en na eenmaal ingelezen te hebben weer netjes te laten groeien. Dit levert gemiddeld een besparing op van 20%, dus bij 100K aan TASWORD-files scheelt dat 20K!

Omdat veel van mijn TASWORD-files BASIC- of ASSEMBLY-listingen bevatten haal ik wel een gemiddelde besparing van 35%. Mijn record-besparing is op dit moment 68%, dus is dit mooi meegenomen op de lange duur.

Ook is er een speciale TASWORD-versie geschreven voor degene die in bezit zijn van een SMITH-CORONA FASTEXT 80-printer. Deze



versie heeft namelijk een achste menu-optie meegekregen die een paar zaken met de printer regelt buiten TASWORD zelf om. In de toekomst zullen ook handleidingen geschreven worden voor andere printers, maar mij ontbreekt daarover nu alle informatie.

#####  
# Nog steeds is de versie 2.2 bij mij te bestellen. Maak dan #  
# fl 17.50 over op mijn GIRO-rekening en binnen twee weken #  
# wordt U uw versie per discette toegestuurd. #  
# GIRO : 3846279 tnv R.O. Aalders - Furkabaan 625 - Utrecht #  
# onder vermelding van TASWORD 2.1 of TASWORD 2.2 #  
#####

## STANDAARDEN.

Bij het lezen van de vorige DUC ben ik geschrokken. Dick had het artikel van Benno Baksteen in het blad opgenomen, iets waarvoor ik hem nog wou waarschuwen maar vergeten was. Ik had liever een aangepaste versie van dat artikel gezien.

Het gevaar van het artikel van Menno was dat hij specifiek zijn situatie besprak. Ik had het verstandiger gevonden dat hij zijn artikel zo algemeen mogelijk gehouden had zodat iedereen makkelijk zijn eigen aanpassingen had kunnen aanbrengen. Het artikel had een wegwijzer moeten zijn voor allen die een printer aan de dierbare DISCOVERY hebben hangen.

Maar goed, het kwaad was al geschied en ik kon mezelf alleen verwijten dat het geplaatst was. Ik kan alleen proberen de schade wat te beperken in dit artikel.

Ik had een paar weken voordat de tweede DUC uitkwam het artikel van Benno al op discette van Dick ontvangen. Ik kon het echter niet goed door de printer krijgen want Benno had zijn grafische tekens andere functies meegegeven dan ik. Dus mijn printer gaf een vreemde mengeling van allerlei schriftsoorten! Benno liet in zijn TASWORD-file ons verwerkers duidelijk voelen wat de mogelijkheden van zijn printer waren. Dat moet dus niet de bedoeling zijn van het opsturen van artikelen.

Ik stel een paar standaarden voor. In artikelen die naar ons opgestuurd worden mogen dan ook alleen deze standaarden gebruikt worden zodat uniformiteit van de artikelen gegarandeerd blijft.

gr 1 : spatie + begin vergrote afdruk  
gr ^1 : einde vergroot afdrukken + spatie  
gr 2 : spatie + begin ELITE-schrift  
gr ^2 : spatie + begin PICA-schrift (normaal)  
gr 3 : spatie + begin onderstreping  
gr ^3 : einde onderstreping + spatie  
gr 4 : spatie + instelling regelafstand op 18/72 inch (dit komt neer op de schrijfmachine 1.5 regel)  
gr ^4 : spatie + instellen op 1/8 inch regelafstand  
gr ^5 : spatie + instellen op 1/6 inch regelafstand

U houdt nog zeven mogelijkheden over voor eigen printerstellingen, dat moet tenminste goede mogelijkheden geven. Indien een bepaald iets niet op uw Printer mogelijk is (kan gebeuren), dan implementeerd U alleen de spatie.

## NOG EEN LAATSTE OPMERKING.

Tasword 2.2 is complex. Ze gaat geheel van andere zaken uit dan het oude Tasword. Het is een kwestie van wennen. Zelfs ik ga nogal eens in de fout.

Hoezeer het ook wennen is, je zult verder moeten. Indien al bestelde copien in de hoek verdwijnen dan heeft het geen zin om door te gaan met verbeteringen aan te brengen in TASWORD.

Naar alle waarschijnlijkheid wordt TASWORD 2.3 op de komende DISCOVERY GEBRUIKERS DAG getoond, maar nog niet vrijgegeven he-

[illegible]

\*\*\*\*\*



Je ontvangt zsm. de kopieën in de brievenbus.

Het VTX modem verbergt nog zeer veel geheimen, mocht er iemand zijn die er een berg uit heeft, even met mij in kontakt willen treden ?

De snelheid waarop het VTX modem werkt is 1200/75 baud. Als je het modem wel eens opengemaakt heb zie je op het printje rechts tussen een paar IC's in de getallen 75,150,300,600,800 en 1200 staan. Denk nou niet dat je hiermee de snelheid van het modem kunt veranderen ! Soldeer hier als-je-blijft niet aan, je blaast zo je modem op... De USART, het grote IC linksboven op hetzelfde printje, de RS 232 interface, crasht zodra het iets anders als 1200 of 75 baut tegenkomt. Ook het modem IC zelf, de TCM 3101, kan geen andere snelheden aan als 1200 of 75 baut.

Volgens de geruchten moet het toch mogelijk zijn het VTX 5000 modem op 300 baut te laten werken, ja zelfs met auto-answer faciliteit ! Mocht het ooit zover komen dan zal ik jullie uitvoerig inlichten hoe het ombouwen geschiedt.

Momenteel zijn wij bezig het standaard ROM programma dusdanig te veranderen dat de Spectrum ook nog wat mag doen als computer. Namelijk wat meer beelden op te slaan in het geheugen en dit als een "bank" weg te SAVEN op disk. Zijn hiervoor nog enige suggesties ? Laat het mij ook dan even weten.

Er zijn plannen om in Rotterdam een databank te starten met een Spectrum en een Discovery (met 5 1/4 ~ drive natuurlijk). Hiervoor moet wel een programma komen compleet met telesoftware, ID kode, opslaan van zo'n 800 pagina's en wie wanneer welke pagina ge-edit heeft. Wie denkt dat hij zo'n programma kan schrijven in MC-Code en iets verdienen wil kan een demo opsturen of in kontakt treden met mij.

Het wil wel eens gebeuren dat het modem niet de goede karakters doorzendt, de ASCII code staat ernaast. Hiervoor is een simpel POKEje in regel 20 nodig om dit te verhelpen. POKE 23611,204.

Starten van het modem vanuit een Basic programma is heel eenvoudig en gaat foutloos; OUT 255,21. Let niet op het vreemde scherm dat je gelijk ziet. Dit verdwijnt en daar is het vertrouwde Micronet 800 beeldje weer. Druk op een toets en ja hoor, daar is het hoofdmenu.

Alle reacties zijn natuurlijk van harte welkom, ik zal ze zo veel mogelijk in de DUC-bladen opnemen zodat iedereen er wat aan heeft.

Rob Macare  
Gaesbeekstraat 77b  
3081 NE Rotterdam  
010-862184 19:00-23:00 uur

#### BUITENLAND

Ja, ook in het buitenland zijn we druk bezig bekend te worden. Ik heb al diverse brieven uit Engeland gehad. Korrespondentie met België, Spanje, Ierland, Schotland, Italië en zelfs Zimbabwe loopt. Het is echt overweldigend, al die brieven. Soms tot diep in de nacht zitten tikken, printen, vouwen en plakken om hun vragen, programma's hints en tips te bekijken en te beantwoorden

Maar ja, dat hoort erbij. Momenteel ben ik bezig om alle info die ik uit het buitenland toegestuurd krijg, te verwerken, samenvatten en weer in te tikken zodat jullie ook de tips die ik krijg kunnen gebruiken.

## BINNENLAND

Wat er in Nederland gebeurd is ook onvolstelbaar ! Dick Kruithof stapt op, importeurs zijn doodstil, maar het ledenaantal is weer gestegen. Dus we gaan met volle moed door om er een nog dikkere DUC van te maken. Dat wil zeggen, niet vol met vragen en antwoorden. Nee, alle vragen kunnen jullie nu aan mij stellen tussen zeven en elf uur 's avonds. Bel je voor zevenen, dan ben ik nog niet thuis en na elfen wil ik ook even wat nachtrust houden. Ja, anders kan ik geen antwoorden verzinnen voor jullie problemen.

Rob Macare  
Gaesbeekstraat 77b  
3081 NE Rotterdam  
010 - 4862184 19:00-23:00 uur

## REACTIES OP DUC 2

\*\*\*\*\*

In deze rubriek zult u ingezonden reacties aantreffen n.a.v. het vorige nummer van DUC. Het zijn meestal aanvullingen c.q. correcties op stellingen, maar ook kritische kanttekeningen zijn welkom. Niet in de zin van "dat wist ik al, interessant, ik kan dat wel beter"; wel in de zin van opbouwende kritiek waar wij als redactie iets mee kunnen. (Als we er niets mee kunnen, gaat het toch in de prullemand, en heb je voor niets geschreven).

Enfijn, de reacties, dus.

\* "K" en "S"-channels kunnen niet ge-opend en gesloten worden. Het zijn interne kanalen. Gebeurt dit terwijl de DISCOVERY-ROM bezig is dan geeft deze de opdracht door aan de SPECTROM-ROM. Deze kan er niets mee beginnen en er volgt weer een terugkeer naar de DISCOVERY-ROM. Het is zelfs zo: de SPECTROM-ROM kent de foutmelding "STREAM ALREADY OPEN" niet, het is een DISCOVERY-melding.

\* De DISCOVERY kent geen systeembatiabelen, anders zou ze SPECTROM-geheugen in beslag nemen. Het ROM werkt totaal anders. Probeer maar eens wat uit:

CLEAR 24000

LOAD \*1;"file naam".

Mooi dat de tweede regel niet geaccepterd wordt. Het ROM heeft vooraf evaluatie (en opzetten werkruimte) geheugen nodig en dat is er niet. Ook de STACK groeit nogal bij de uitvoering van een DISCOVERY instructie:

RANDOMIZE USR 0

CLEAR 24000: LOAD \*1;"file naam"

en weer gaat het mis, hoewel nu op een ander punt. (De DISCOVERY kan nu zijn channel-info niet kwijt. Spijtig, maar waar).

De systeemvariabelen zijn (soms) gebruikersvriendelijk, ze bewaren in het geval van I1 bepaalde DEFAULT waarden (zoals met welke disc er gewerkt wordt, welke netgebruiker, etc.). Wij hebben die dus niet nodig.





string van regelnummers , en alle GOSUBs via deze string kan laten lopen . Als er later een nieuwe versie van dit (subroutine) programma uitkomt , hoeft hij niet al zijn programmas opnieuw te schrijven , omdat de (bijgaande) string van regelnummers aangepast is aan de nieuwe versie .

In BASIC is aan zo'n string eigenlijk weinig behoefte omdat er meestal wel genoeg regelnummers zijn om subroutines uit te breiden , en er toch geen subroutine-programmas zijn (hoewel , kijk maar eens naar BETABASIC) , maar met machinetaal ligt dat anders . Alle routines liggen aaneengesloten in het geheugen , en als er een routine gewijzigd moet worden , schuiven alle navolgende routines mee , met als gevolg dat de (absolute) sprong-instructies mee moeten veranderen . Als de start-adressen (of een deel ervan) van deze routines in tabellen opgeslagen zijn , hoeven alleen deze tabellen veranderd te worden , op voorwaarde dat elke aanroep van deze subroutines via deze tabellen loopt . En dit is nu precies wat er in de OPUS-ROM gebeurt .

Er zijn in totaal 13 tabellen , genummerd #00 tot #18 , waarvan er drie (#10 , #12 , #14) vermeld staan in section 4 . (Deze bevatten start-adressen van belangrijke subroutines) . Voor het opzoeken van een waarde in een tabel zijn twee dingen nodig ; namelijk het nummer van de tabel en de index in die tabel , die beiden meegegeven moeten worden met de instructie RST 30 (een speciale vorm van CALL #30) . In het B-register moet de index staan , en op het adres direct na de RST 30 moet een byte staan met het nummer van de tabel (meestal mij een assembler via de pseudo-instructie DEFB .) Het opzoeken van de eerste index (index 0) in de negende tabel (tabel #10) gaat dus als volgt :

```
START    LD    B,0
          RST   30
          DEFB  #10
          RET
```

De gezochte informatie staat nu in het HL-registerpaar .

Nu is het zo dat er te veel en te lange tabellen zouden komen indien er rekening gehouden zou moeten worden met alle mogelijke syntaxen (het OPEN #en naar 10 verschillende kanalen , het MOVEn , etc.) , dus is er een mogelijkheid open gelaten om in elk tabel onderscheid te maken tussen verschillende dingen (zoals de 16 foutmeldingen , de 12 commando's en 11 hook-codes) . De eerste byte van elk tabel bevat een bepaald getal die (al dan niet) overeenkomt met het gezochte waarde (fout nummer , ASCII-waarde van het commando , hook-code) gevolgd door twee bytes voor het startadres van de subtabel behorend bij deze waarde . Zo bestaat elk tabel uit een aantal drie-tallen (zoekwaarde (1) en adres (2)) , waarbij het laatste drietal 0(nul) als zoekwaarde heeft (om aan te geven dat het einde van het tabel bereikt is) en het adres wijst naar een tabel met foutmeldingen .

Aangezien er 13 tabellen zijn (met ieder hun eigen sub-tabellen) ligt het voor de hand om aan te nemen dat de startadressen van de tabellen ook in een tabel opgeslagen liggen en dat het getal achter de DEFbyte-instructie de index in deze (hoofd)tabel is .

Dan is het nu tijd om aan te geven wat RST 30 ongeveer doet. Eerst wordt de waarde van de byte na de aanroep )DEFB) opgehaald en in het C-register opgeborgen , dan wordt gekeken naar de laatste twee bytes van de OPUS-ROM (of de eerste twee van de (extra) 2K RAM , maar daarover later meer) , en deze wordt in HL (registerpaar) opgeborgen. Dan wordt C+1 bij HL opgeteld



en staan op de adressen HL-1 en HL het startadres van het C-tabel (van subtabellen) . Vervolgens wordt HL gelijk gemaakt aan dit startadres en wordt de zoekwaarde in de tabel telkens vergeleken met het A-register totdat de zoekwaarde gelijk aan A of gelijk aan 0 is . Bij een succesvolle zoekpoging (via A) of een niet-succesvolle (via 0) wordt HL gelijk gemaakt aan de twee waarden die direkt achter de zoekwaarde staan (deze waarden vormen het startadres van het subtabel) , zodat HL nu eindelijk naar de gezochte subtabel wijst. Tenslotte wordt B+1 bij HL opgeteld , en HL gelijk gemaakt aan de adressen HL-1 en HL , dus bevat dan HL het startadres van de gezochte routine .

De reden waarom aan A geen waarde hoeft te worden toegekend in het voorbeeld is , dat tabel #10 als eerste zoekwaard het getal NUL bevat , en er dus geen onderscheid gemaakt wordt tussen waarden in het A-register (dit geldt ook voor de tabellen #12 en #14) .

Indien B (of C) 255 bevat wordt teruggekeerd uit de RST 30 subroutine vlak voor het moment dat B+1 (of C+1) bij HL zou worden opgeteld . Dit heeft als doel dat het startadres van het subtabel (B=255) of de hoofdtabel (C=255) in HL gevonden kan worden , in plaats van de waarde in de subtabel .

Na dit ingewikkelde verhaal een voorbeeldje : we zoeken in de negende tabel (tabel #10) de eerste index (index #00) , zie voorbeeld boven .

(N.B: de adressen die in dit voorbeeld gebruikt worden , zijn adressen zoals ze in versie 1.2 voorkomen , en kunnen dus voor andere versies verschillen . ) Eerst word de waarde #0 in C gezet . Op adres #1FFE staat #74 en op adres #1FFF staat #17, dus HL word #1774 . Op adres #1774 start (blijkbaar) de hoofdtabel . C+1 wordt bij HL opgeteld ,  $HL=1774+10+1=1785$  . Op #1785 staat #71 , daarom wordt HL nu #1771 , het eerste adres van tabel #10 . Op dit moment moeten we gaan zoeken naar NUL , of en waarde die gelijk is aan A , maar aangezien de eerste waarde al NUL is , wordt HL gelijk gemaakt aan de waarden op adressen 1772 en 1773 . Dit is 1774 , dus de feitelijke tabel #10 (de enige subtabel) begint er direct na . We tellen B+1 op bij HL , dus  $HL=1774+0+1=1775$  . Op adres 1775 staat #03 , op adres 1774 staat #CC , dus tenslotte wordt HL #03CC , het startadres dat we zochten . Uit de informatie uit section 4 van de manual blijkt dat dit het startadres van de routine CALCHN is .

Zoals gezegd zijn er 13 tabellen , waarvan er drie vermeld staan in section 4 . Wat staat er nu in de andere tabellen ?

In tabel #00 staan de startadressen voor de hook-codes . Tabel #02 bevat die voor de commando's . In tabel #04 staan de adressen voor alle channels ("m" , "#" , "CODE" , etc.) en in tabel #06 die voor de channels "m" en "j" . Tabel #0A bevat informatie over de drives 1 t/m 3 en 5 (!) en tabel #0C de startadressen voor de ASCII-waarden die afgedrukt moeten worden bij een foutmelding . Uit tabel #04 blijkt dat er een channel " " bestaat , die precies gelijk is aan channel "m" (probeer maar eens CAT " ";1) en een channel "LOAD" , wat (na uitproberen ) "CAT" voorsteld . ('CAT' is trouwens niet vermeld , en 'LOAD' is het keyword , wat een beetje moeilijk in te typen is , tussen twee aanhalingstekens .)

Het opzoeken van de ASCII-waarden van een foutmelding gaat als volgt : START LD A,nummer van de foutmelding

LD B,255

RST 30

DEFB #0C

RET

In tabel #0C wordt dan en subtabel gezocht, behorende bij A. en omdat B 255 is, wordt van de RST-30 teruggagekeerd met in HL het startadres van deze subtabel. Hierin staan de ASCII-waarden van de letters die afgedrukt moeten worden bij de foutmelding waarvan de waarde in het A-register opgeslagen is. Het laatste af te drukken karakter heeft bit 7 geSET (er is 128 bij de ASCII-waarde opgeteld), om het einde van de string aan te geven.

Na deze informatie dan nu het interessante werk, maar om de voorbeeldprogrammas uit te kunnen voeren is het wel nodig om eerst een IC 6116 (statisch RAMmetje, zie DUC #2) aanschafft te hebben, en deze in de disc-drive te plaatsen. Het kopen van de IC moet niet zo moeilijk zijn. Er schijnen echter twee versies van verkrijgbaar te zijn, een zogenaamde CMOS en een MMOS. De een (CMOS, geloof ik) kost tussen de 7.50 en 10.00 gulden, de ander is ongeveer 10 gulden duurder. Ik heb zelf de goedkopere genomen, en die werkt prima. Na de aanschaf komt de plaatsing. Schakel hiervoor de disc-drive uit met de schakelaar achterop, en verwijder het toetsenbord en alle andere randapparatuur. Draai de drive op zijn rug. Draai de kruiskopschroef op elk hoekpunt los, doe hetzelfde met de schroef achterop, onder het woordje GREAT. Leg de 5 schroeven ergens veilig neer. Nu moet het omhulsel los van de onderkant zitten. Schuif de omhulsel naar achteren (weg van de plaats waar het toetsenbord hoort) als een luciferdoosje. Pas op voor de grote kruiskopschroef en de connectors (buig het omhulsel desnoods iets uit). Als het omhulsel verwijderd is, draai dan de drive weer om, zodat je er van bovenaf in kunt kijken. Op de bodem van de drive is nog een vrij voetje voor het IC 6116 zichtbaar. Houdt nu het IC zo, dat de halve-maan vormige uitsparinkje naar drive 1 wijst, net als het IC dat reeds aanwezig is, meer naar de kant van het toetsenbord (de letters van de beide IC's moeten dan dezelfde kant op te lezen zijn). Druk nu het IC 6116 (met de pinnetjes naar beneden) zo recht mogelijk in het voetje. Dit gaat misschien wat moeilijk, maar met een paar keer voorzichtig proberen moet het wel lukken. Kijk of tussen het IC en het voetje nog ongeveer 1 millimeter over is, druk hem anders nog dieper. Als alles er goed uitziet schuif je het omhulsel weer terug, draai de schroeven vast, zet het toetsenbord er weer aan, en (last but not least) schakel je de drive weer aan.

De drive kan naspeuren dat het RAMmetje aanwezig is, en de schaduw-ROM reageert daarop door de hoofdtabel in het RAMmetje te kopiëren. Als er nu een RST 30 instructie uitgevoerd wordt, worden niet de laatste twee bytes van de ROM in het HL gezet, maar de eerste twee van het RAMmetje. HL wijst daardoor naar het tabelletje in de 2K RAM, die (bij mijn versie 1.2) begint op adres #2004, dus de 5e byte in RAM. Deze tabel is gelijk aan die in ROM, met een uitzondering: tijdens de initialisatie is de verwijzing in de hoofdtabel naar tabel #0A gewijzigd in een verwijzing naar een (tweede) tabel in RAM, die meegekopieerd is. Hierin staan 4 verwijzers naar subtabellen (voor elk drive-nummer een) die tevens in RAM gezet zijn. Deze subtabellen zijn achter volkomen gelijk aan die in ROM. Tabel #0A bevat informatie over de discs 1 t/m 3 en 5, dus kan relatief eenvoudig gewijzigd worden, maar daarover later meer.

Ik wil eerst laten zien hoe de foutmeldingen van de OPUS in het Nederlands vertaald kunnen worden. Dit kan door tabel #0C te veranderen, en de startadressen in RAM te laten wijzen, en daar de (ASCII-waarden van de) foutmeldingen neer te zetten.



Hiervoor is het nodig om een kopie van de 8K ROM + 2K RAM in (de 48K) RAM te hebben. SAVE daartoe de eerste 10K op disc (niet op tape), met SAVE \*1;"10K"CODE 0,10240;laad het daarna weer in op adres 32768 met CLEAR 32767:LOAD \*1;"10K"CODE 32768.

We beginnen met de eerste twee regels van het programma, waarin we een konstante en twee variabelen van een startwaarde voorzien.

```
10 LET ramstart = 8*1024
20 LET tabel = ramstart + 96
30 LET vrij = ramstart + 256
```

Tabel wijst naar de eerste vrije plaats in RAM waar we informatie kunnen toevoegen. De gekopieerde tabellen nemen 92 bytes in beslag en beginnen op het 5e adres in RAM, dus de totale ruimte is 96 bytes. Ik wil de eerste 256 bytes in RAM gebruiken voor tabellen, de rest voor andere informatie (hier: ASCII-waarden voor de foutmeldingen).

We bepalen eerst de startadres van tabel #0C, de zevende tabel.

```
1000 LET HL = (ramstart + 4) + (12*1) + (32768)
1010 LET tabel10C = PEEK (HL-1) + 256*PEEK(HL)
```

Deze wijst (nog) in ROM. We gaan deze tabel nu in RAM kopiëren.

```
1020 FOR f= 0 TO 3*16-1
1030 POKE (tabel+32768)+f,PEEK(tabel10C+32768+f)
1040 NEXT f
```

waarbij 16 het aantal foutmeldingen is, en 3 het aantal bytes per foutmelding. Verander nu in de hoofdtabel de wijzer naar de oude tabel #0C (in ROM) naar de nieuwe (in RAM).

-56 99  
256 11  
→ 1050 POKE HL-1,tabel+256\*INT(tabel/256)  
1055 1060 POKE HL, INT(tabel/256)

Merk op nu tabel #0C in RAM gekopieerd is, de wijzer in de hoofdtabel ernaar wijst, maar de subtabellen (die de ASCII-waarden van de foutmeldingen bevatten) in tabel #0C nog steeds in ROM staan. We gaan nu de 16 foutmeldingen in RAM zetten. We lezen daartoe de strings vanuit de DATA-regels in.

```
1060 FOR f= 1 TO 16
1070 READ a$
```

Verander de wijzer naar de foutmelding in tabel #0C.

```
1080 POKE (tabel+32768)+1,vrij-256*INT(vrij/256)
1090 POKE (tabel+32768)+2,INT(vrij/256)
1100 LET tabel=tabel+3
```

Zet nu de karakters in RAM.

```
1110 FOR g= 1 TO LEN a$
1120 PKE (vrij+32768),CODE a$(g)
1130 LET vrij = vrij + 1
1140 NEXT g
```

SET bit 7 van het laatste karakter.

```
1150 POKE (vrij+32768-1),128+PEEK (32768+vrij-1)
```

En herhaal dit nog 15 maal

```
1160 NEXT f
```

Vervolgens om de 16 regels.

```
1500 DATA "Foute device naam"
1510 DATA "Stream is al open"
1520 DATA "Fout drive nummer"
1530 DATA "Verboden te schrijven"
1540 DATA "Geen plaats op disc"
1550 DATA "Disc I/O fout"
1560 DATA "File niet gevonden"
```

```
1570 1670 DATA "Foute hook code"
1580 DATA "Foute file grootte"
1590 DATA "Verifiëren faalde"
```

```

1600 DATA "Verkeerde file type"
1610 DATA "Verkeerde disc"
1620 DATA "1986 Marcel van Dongen"
1630 DATA "Doe niet zo dom joh "
1640 DATA "RAM is verknoeid"
1650 DATA "Geen bericht"

```

Deze kunnen naar believen veranderd worden .

Run het programma (vergeet niet het file "10K" in te laden op adres 32768) , en SAVE daarna de 2K RAM op disc met

```
SAVE *1;"IC 6116" CODE 32768+8*1024,2*1024
```

SAVE voor alle zekerheid ook het Basic-programma .

Om nu te kijken of de wijzigingen gelukt zijn , resetten we de SPECTRUM met RANDOMIZE USR 0 (of de RESET toets) , en laden we daarna het geSAVEde blok code in over het RAMmetje , met

```
LOAD *1;"IC 6116"CODE 8*1024,2*1024
```

Als de SPECTRUM crasht , heb je waarschijnlijk het verkeerde blok geheugen gesaved, of op het verkeerde adres ingeladen . Als alles goed is gegaan , kunnen we testen of de foutmeldingen inderdaad in het Nederlands (afhankelijk van de DATA-regels 1500 t/m 1650) zijn . Probeer dus een file dat niet op disc staat in te laden , bijvoorbeeld .

```
LOAD *1;"ggg"
```

Dit geeft dan (hopelijk) een Nederlandse foutmelding . Probeer ook andere foutmeldingen te genereren , en kijk of deze allemaal werken . Probeer tenslotte

```
RANDOMIZE USR 10134
```

en zie , de copyright-melding !!

Het vertalen van de foutmelding in het Nederlands is niet echt nuttig , maar het geeft wel duidelijk aan wat je zoal kan doen met het RAMmetje . Ik hoop nu een iets nuttiger toepassing te laten zien , namelijk het maken van extra commando's . De mogelijkheden zijn legio , voor het gemak houdt ik de syntax eenvoudig . Het is weer de bedoeling om het geheugenblok "10K" op adres 32768 in te laden , na verlagen van RAMTOP met CLEAR 32767 . We maken het commando FLASH , zonder parameters , dat ervoor moet zorgen dat de kleur van de BORDER korte tijd knippert . Mischien zie je al dat we hiervoor tabel #02 moeten aanpassen , en dit kan alleen maar door deze naar RAM te kopiëren , waarbij we ruimte open moeten laten voor het extra commando : 1 plaats voor de Ascii-warde van 'FLASH' , en 2 voor het start-adres van de subroutine die het eigenlijke FLASHen doet .

De initialisatie van de variabele en de konstante is gelijk aan die van 't vorige voorbeeld , dus je kan regels 10 t/m 30 direct overnemen . Vervolgens moeten we de plaats opzoeken waar tabel #02 in ROM staat . We zoeken in de hoofdtabel :

```
2000 LET HL =(32768+ramstart+4)+2+1
```

```
2010 LET tabel102 = PEEK (HL-1)+256*PEEK(HL)
```

Tabel #02 staat nog in ROM . We verplaatsen hem dus naar de vrije plaats voor tabellen in het RAMmetje , die in het variabele 'tabel' wordt bijgehouden .

```
2020 FOR f = 0 TO 3*13-1
```

```
2030 POKE (tabel+32768)+f,PEEK ((tabel102+32768)+f)
```

```
2040 NEXT f
```

De eerste 36 bytes (voor de eerste 13 commando's) zijn in RAM gezet . Nu voegen we de gegevens voor het nieuwe commando in , de eerste byte is de Ascii-warde van 'FLASH' , de twee volgende vormen het startadres waar de subroutine moet komen , en dit adres staat in de variabele 'vrij' .

```
2050 POKE (tabel+32768)+36, CODE "FLASH"
```

```
2060 POKE (tabel+32768)+37,vrij-256* INT(vrij/256)
```

```
2070 POKE (tabel+32768)+38, INT(vrij/256)
```



Vervolgens moeten we de laatste waarde , met het adres van de fout-routine , in RAM plaatsen .

```
2080 FOR f=0 TO 2
```

```
2090 POKE (tabel+32768)+39+f, PEEK((tabel02+32768)+36+f)
```

```
2100 NEXT f
```

We verplaatsen de wijzer in de hoofdtabel , die naar tabel #02 in ROM wijst , naar onze nieuwe tabel #02 in RAM .

```
2110 POKE HL-1,tabel-256* INT(tabel/256)
```

```
2120 POKE HL, INT(tabel/256)
```

We moeten 'tabel' mog laten wijzen naar de nieuwe vrije plaats voor tabellen .

```
2130 LET tabel = tabel + 42
```

Nu moeten we nog de machine-code routine op z'n plaats zetten . Degene die niet zo bekend is met machinetaal kan verder gaan na de laatste assembly-instructie , vlak voor regel 2140 .

De machine-code routine moet met twee gevallen rekening houden nl. met syntax-checking (controle of het commando wel goed is ingetypt) en run-time (uitvoering van het commando) . Eerst kijken we of er geen syntax-fouten zijn . Het A-register bevat het ASCII-waarde van het symbool volgende op FLASH . De (goede) mogelijkheden zijn :A = 'ENTER' en A = ':' .

```
START    CP 13                ; symbool is 'ENTER' ?
          JR Z,GOED
          CP 58                symbool is ':' ?
          JR Z,GOED
FOUT      RST 10              ; keer terug naar SPECTRUM-ROM en
          DEFW #1C8A          ; geef melding 'nonsense in BASIC'
GOED      RST 18              ; syntax-checking ?
          JR NZ,RUN
SYNCHK    LD (IX+0),255       ; er is geen fout opgetreden
          LD SP,(#5C3D)       ; zet SP-register weer goed
          LD BC,#1BF4         ; zet het adres van de subroutine
          PUSH BC             ; NEXT-STATEMENT op stapel
          JP #1748            ; en keer terug naar de SPECTRUM-ROM
```

Hier kunnen we de eigenlijke subroutine zetten . Dit word dan :

```
RUN      LD A,(#5C48)         ; A=BORDER-kleur
          LD B,A              ; bewaar A in B
          AND 7
          LD C,A              ; C=INK kleur
          LD A,B              ; restore A
          RRA
          RRA
          RRA
          AND 7
          LD B,A              ; B=PAPER kleur
          XOR C               ; de BORDER moet knippen , dus
          LD C,A              ; nemen we de XOR-waarde van B en C
          LD A,B
          LD L,0              ; aantal malen door de LOOP
LOOP 1    LD B,0              ; aantal malen door het kleine loop
LOOP 2    OUT (#FE),A         ; het feitelijke knippen
          XOR C               ; wissel de kleur
          DJNZ LOOP 2         ; B=B-1 : als b<>0 dan naar loop 2
          DEC L
          JR NZ,LOOP 1
          LD A,(#5C48)         ; zet nu de BORDER weer goed
          RRA
          RRA
```

```

RFA
AND 7
OUT (#FE),A
RET ; klaar !!

```

Als we dit in en DATA-regel zetten en laten inlezen m.b.v. READ krijgen we

```

2140 READ a
2150 POKE vrij,a
2160 LET vrij = vrij+1
2170 READ a
2180 IF a<>999 THEN GOTO 2150
2190 STOP
2200 DATA 254,13,40,7,254,58,40,3,215,138,28,223,32,15,253,
54,0,255,237,123,61,92,1,244,27,197,195,72,23
2210 DATA 58,72,92,71,230,7,79,120,31,31,31,230,7,71,169,79,
120,46,0,6,0,211,254,169,16,251,45,32,246,58,72,92,31,
31,31,230,7,211,254,201
2220 DATA 999

```

SAVE ook dit Basic-programma naar disc . RUN het programma (nadat het file "10K" ingeladen is op adres 32768 , en wacht tot de STOP-boodschap verschijnt . SAVE dan de 2K voor het RAMmetje met

```
SAVE *1;"IC 6116"CODE 32768+8*1024,2*1024
```

Laad het weer in over het RAMmetje , met

```
LOAD *1;"IC 6116"CODE 8*1024
```

en toets nu FLASH in (E-modus , SYMBOL-SHIFT V) en druk op ENTER Nu moet de BORDER kort flitsen . Als dit niet zo is , kijk dan of je geen typ-fouten hebt gemaakt in regels 2200 en 2210 . Als het werkt , probeer dan het volgende :

```
10 FOR f=0 TO 63: POKE 23624,f:FLASH:NEXT f
20 BORDER 7

```

Op adres 23624 worden de PAPER , INK , FLASH en BRIGHT-attributen van de BORDER en het onderste deel van het scherm bijgehouden en deze waarde wordt gebruikt door FLASH .

Indien je de versie 1.2 van de DISCOVERY-ROM hebt , kun je de DATA in regel 2200 vervangen door :

```
2000 DATA 205,114,4
```

Dit is veel korter , maar werkt misschien niet bij alle versies , dus heb ik maar een kort routinetje geschreven wat hetzelfde doet als deze CALL . Als je een ander versie-nummer hebt , kun je het ook wel proberen . Laat me dan horen of het werkt .

De twee voorbeeld-programmas hebben verschillende regelnummers en kunnen dus met elkaar GEMERGED worden . Als het programma (bestaande uit de twee deel-programmas) dan GERUND wordt , krijg je zowel Nederlandse foutmeldingen als het nieuwe commando 'FLASH' .

Heb je nu zelf ideeën voor nieuwe commando's die nuttiger zijn dan heb je misschien iets aan de volgende informatie : Er zijn minstens 6 adressen die als systeem-variabelen dienst doen , nl. #2800, #2801, en #3000 t/m #3003 . Van adres #3000 bevat

```

bit 1 :0 als IC 6116 niet aanwezig is
1 " " " " wel " " " "
bit 3 :0 " " disc 1 het laatst gebruikt is
1 " " " " 3 " " " " " "
bit 7 :0 " " "j" wel geformateerd is
1 " " " " niet " " " "

```

De betekenis van de rest is mij niet duidelijk . Ik vermoed echter dat bepaalde bits gebruikt worden om aan de hardware signalen door te geven . Ook tast ik in het duister wat betreft de informatie over de discs (subtabellen van tabel #0A) . Voor



