BULLETIN

SINCLAIR GEBRUIKERSGROEP GRONINGEN/ASSEN



10e jaargang nr 11
jubileumjaar '92

COLOFON



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

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

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

REDAKTIE CONTACT/
VICE PENNINGMEESTER/
MATERIAAL COMMISARIS:
Coen Ballintijn
B. Boermalaan 7
9765 AP Paterswolde
tel: 05907 - 91482

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

Kopij en vragen graag aan de redaktie contactpersoon!

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

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

Het lidmaatschap van onze gebruikersgroep bedraagt per kalenderjaar:
f 20,00 voor personen tot en met 17 jaar en
f 30,00 voor personen van 18 jaar en ouder.
Leden zijn automatisch abbonnee op het Bulletin.
Losse nummers f 4,00.

U kunt lid worden van de SGG

Door u op te geven bij de penningmeester.

VAN DE REDACTIE



Hallo allemaal!

In dit extra (jubileum) Bulletin vindt u een aantal artikelen uit de jaargangen 1 t/m 7, waar van wij (de redactie) dachten dat ze de moeite waard zijn om nog eens herhaald te worden. Juist ook om dat een groot aantal leden deze eerste jaargangen niet (meer) in hun bezit hebben, en de kwaliteit van de allerallereerste jaargangen zich eigenlijk te slecht is om ze te gaan copyeren voor evt. belangstellenden. Daar waar we dat nodig achtten zijn enkele artikelen aangepast c.q uitgebreid. Onze bedoeling is om de afleveringen van dit extra nummer te laten verschijnen in de nummers: 1, 3, 4, 6, 8 en 9 v/d 10e. jaargang. We hebben dit Jubileumnummer zo opgezet dat u de afleveringen kunt samen voegen tot het 11e. nummer van de 10e. jaargang. Mocht u nog belang hebben bij bepaalde artikelen uit de 8e, en/of 9e. jaargang, of lijkt het u leuk om ook in de toekomst weer zo'n 'verzamelnummer' te ontvangen, laat het ons dan even weten. Daar waar wij de naam van de schrijver van het artikel niet konden achterhalen hebben we de naam van de voor dat nummer verantwoordelijke redactie lid/leden ge(mis)bruikt. Het kan mogelijk zijn dat binnen een 'BLOK' de artikelen onderling van volgorde verwisseld zijn i.v.m een mooiere pagina indeling.

In dit JUBILEUM nummer:

	auteur	blz.
- Colofon	: zie blz. 2	
- Inhouds opgave	: redactie	3
- BASIC Programmeer tips:	:	4-11
Meer (gemakkelijker) doen met BASIC (div. routinetjes en prog. pokes)	: O. Heerkens	
Beschermen MC tegen foutieve CLEAR	: E. Draaisma	
Instant SCREEN	: bron: CRASH	
en nog veel meer		
- TW 2 en TW 3	:	12-19
aanpassingen van/voor TW2 of TW3		
printercontrolecodes TW3		
TW3 en IBM karakter set		
ook hier no veel meer		
- Super Spectrum? 256 K Ram en K ROM	: R. Koning	20-25
- Kleine zelfbouw projecten:	:	25-29
Resetknop, geluid via TV,		
TV als Monitor, Monitor uitgang.		
- Randapparatuur (4- delige serie)	:	30-36
- DATA communicatie :	: C. van Krimpen	37-43
via div. types telefoon	: H. Vesper	
via zelfbouw modem	: Lelieveld + Koevoets	
via zelfbouw FAX interface.	: H. Vesper	

BASIC PROGRAMEERTIPS EN POKES

O.Heerkens, drs.PC Sextrum e.a.

* TOETSENBORD BEEP:
Met POKE 23609,x krijg je een beep iedere keer als je een toets induwt. X ligt tussen 0 en 255. Het beste resultaat krijg je met een waarde tussen 50 en 100 .

* CAPSLOCK
POKE 23658,8 zet de computer automatisch in capslock mode, al je input wordt CAPITAL. POKE 23658,0 zet je computer weer in kleine letters. Gebruikt bij bv.:

```
    LINEnr POKE 23658,8: INPUT "Stoppen J/N";i$: IF i$<>"J" AND
           i$<>"N" THEN GO TO LINEnr
 LINEnr+5 IF i$="J" THEN STOP
LINEnr+10 POKE 23658,0
```

Je hoeft nu alleen maar op hoofdletters te testen. Een mooiere oplossing vindt je in een van de artikelen over GESTRUCTUREERD PROGRAMMEREN die in de 10e jaargang gepland staan.

* ATTRIBUTEN
Een attribute bevat de gegevens betreffende INK en PAPER kleur, en FLASH en BRIGHT van een PRINTpositie op het scherm. Een attribute waarde kan je als volgt vinden. Ieder attribute bestaat uit acht bits. Het meest linkse bit is bit nummer 7, het meest rechtse bit is bit nummer 0. Ieder bit heeft een functie die waar is als het bit 1 is en die niet waar is als het bit 0 is. De functies van de attribute bits zijn:

```
BIT NR.      7     6      5     4     3     2     1     0
FUNCTIE   FLASH BRIGHT Groen Rood Blauw Groen Rood Blauw
                       |_____________| |_____________|
                             PAPER           INK
```

Uit de kleuren groen, rood en blauw kunnen alle andere kleuren gemaakt worden. Je neemt het getal op het toetsenbord dat bij de desbetreffende kleur hoort. Bijvoorbeeld magenta is kleur nummer 3. Dit getal zetten we om naar de binaire waarde = 011 .
Als je dus een attribute wil hebben met de volgende functie, geen FLASH, wel BRIGHT, PAPER zwart (=0) en gele INK (=6), dan heeft die de binaire waarde 01000110 en dat is 70 decimaal. (Type maar eens in 10 PRINT BIN 01000110 en RUN dat dan.)
Het veranderen van een attribute waarde van een bepaalde beeldpositie kan op de volgende manier:

```
10 INPUT "regel ";y;"kolom ";x;"attribute ";a
20 POKE 22528+32*y+x,a
```

De attributen staan namelijk vanaf adres 22528 in het geheugen. Je POKEd de attribute waarde direct op de gewenste positie. Alle informatie op die positie blijft ook staan.

* INPUT
- Voor INPUT in een bepaalde kleur moet je adres 23624 POKEn met een attribute waarde. Geef van te voren de BORDER dezelfde kleur

als de PAPERkleur die je attribute waarde gebruikt. (INPUT gebeurt nu eenmaal in de BORDER niet waar?)
- INPUT LINE a$ print de " "niet. Een voordeel is dat 'STOP in INPUT' niet meer werkt. STOP wordt dan de inhoud van a$. Op deze manier wordt in veel programma's voorkomen dat ze stoppen door verkeerde INPUTgegevens. Toch is het nog mogelijk te stoppen in de INPUT, door op CAPSHIFT 6 te drukken. Dit geeft de foutmelding: H STOP in INPUT .

* SCROLL
- LET a= USR 328Ø SCROLLt het hele scherm één regel omhoog, inclusief de attributen.
- In PRINTopdrachten POKE je 23692 met de waarde 255. De computer vraagt dan niet meer om SCROLL? als het scherm vol is, maar SCROLLt automatisch.

* PRINTEN OP DE ONDERSTE TWEE REGELS
PRINT #Ø; AT Ø,Ø;"TEKST" print op de 23e regel. AT 1,Ø print op de 24e regel. Dit printen gebeurt net zo als een normale PRINT opdracht op het scherm, als er al iets stond dan wordt dit gewoon overschreven. Denk er wel om dat de tekst niet langer is dan één regel anders wordt de tekst naar boven gescrollt. Het onderste scherm deel wordt dan naar boven uitgerekt en komt dan op de plaats van het bovenste schermdeel terecht. Als daar als iets stond dan wordt dit omhoog gescrolt. Als de computer direct na zo'n PRINT opdracht zelf met een melding wil komen, zal hij eerst het hele onderste schermdeel wissen, en daarna de melding op regel 24 zetten. Met een PAUSE Ø opdracht na de PRINT opdracht zorgt er voor dat de tekst blijft staan totdat er een toets is ingedrukt.
PRINT #Ø;"tekst" print ook op regel 24, maar overschrijft een reeds aanwezige tekst niet doch SCROLLt deze omhoog.

* WORTELS
De derde machts worrel van 27 krijg je door PRINT 27^(1/3) in te typen en de vijfde macht wortel van 7776 door PRINT 7776^(1/5). In het algemeen geldt n^(1/r) geeft de r-de machts wortel van n. Let op ^ betekent tot de macht.

* VRIJE BYTES
- PRINT 65535-USR 7962 geft het aantal bytes dat nog vrij is. Dit geldt alleen voor de 48K Spectrum.
- PRINT PEEK 23429 + 256 * PEEK 23430 + 65536 * PEEK 23431 geeft bij de 128K aan hoeveel vrij geheugen er nog op de RAMdisk is. Houdt er wel rekening mee dat er voor elk FILE nog 29 bytes extra gebruikt worden voor filenaam, lokatie e.d. Dus een SCREEN$ van 6912 bytes gebruikt op een RAMdisk 6941 bytes.

* GEHEUGEN BESPARING
1 Meer opdrachten op één regel plaatsen.
2 In een rpogramma één keer LET a=1 gebruiken en verder overal waar het getal 1 nodig is de variable a gebruiken.
3 In plaats van PAPRE 1: INK 6: FLASH Ø: BRIGHT 1: CLS te gebruiken, type je in POKE 23693,78: CLS. Het getal 78 is een attribute waarde.

*** REGEL 0**

Een regel o maak je door eerst de door jou als regel 0 gewenste refel als eerste regel in het programma te zetten. Nu kan je hem d.m.v. een POKE in een regel 0 veranderen.

De adressen waarin je POKEn moet kan je vinden door: PRINT PEEK 23635 + 256 * PEEK 23636 . Bij een BASICprogramma geeft dit meestal adres 23755. Samen met 23756 bevat dit adres het eerste regel nummer van je programma en wel als volgt:
in 23755: INT(regelnr/256), bij regel 258 komt hier 1 te staan.
in 23756: regelnr-(INT(regelnr/256))*256, hier komt dan 2.
Om nu een regel Ø te krijgen POKE je 23755,Ø en 23756,Ø.
Een regel Ø kan je niet EDITen en DELETEn, als je hem wilt veranderen of verwijderen, moet je eerst weer van een normaal regelnummer voorzien. POKE daarvoor op 23755 en 23756 de hiervoor benodigde waarden.
N.B. gebruik altijd een niet in je programma voorkomend regelnr omdat je anders grote problemen krijgt.

* MID$, LEFT$, RIGHT$
- MID$(a$,3,2) neemt twee karakters van a$, te beginnen bij het derde karakter. Voor de Spectrum is dat dus a$(3 TO 5).
- LEFT$(a$,2) zijn de eerste twee karakters van a$. Dus a$(TO2)
- RIGHT$(a$,2) zijn de laatste twee karakters van a$. Dus a$(LEN a$-2 TO)

* GEEN BREAK

Hoe kan je je zelgemaakte programma beschermen tegen inbraak? Heel eenvoudig! Zorg er voor dat je computer geen informatie, zoals b.v. foutmeldingen, meer op het onderste schermdeel kwijt kan en hij zal crashen zodra hij er toch iets probeert neer te zetten. Er zijn twee simpele manieren om dit te doen:
- Het onderste deel van het scherm is gekoppeld aan stream #0, door nu als 1e regel van je programma de volgende regel te nemen, sluit je deze stream zodra je programma draait, en zie daar inbraak (BREAK-in) vrij. 1 CLOSE #Ø .
- Met POKE 23659,0 voorkom je ook dat de Spectrum nog gebruik van dit gedeelte van het scherm kan maken.

Deze truukjes hebben als nadeel dat je de statements: INPUT, CLS en BORDER niet meer kunt gebruiken. (Ze gebruiken het onderste schermgedeelte). Dit hoeft echter geen probleem te zijn, je kunt INPUT vervangen door INKEY$ mmet een printopdracht ergens boven in het scherm. En het effect van CLS en BORDER kun je krijgen door de kleurinformatie rechtstreeks in het displaygeheugen te POKEn en door OUT 254, COLOUR te gebruiken voor de BORDER kleur.

De adressen 23613 en 23614 vertellen de Spectrum hoe de zaken weer recht te zetten in geval van een fout (zoals BREAK). POKE 23613,4 :POKE 23614,61 zorgt er voor dat de computer zichzelf reset als Break wordt ingedrukt. Helaas overschrijven RUN, CLEAR, GO SUB en RETURN de gePOKEte waarden, dus de POKEs moeten door het hele programma na een van deze commando's herhaald worden.

GESTRUCTUREERD BASIC

C. v. Krimpen / H. Boon.

Een beetje BASICprogramma wordt al gauw een bord spaghetti, al die GOTO's en die GOSUB's maken een programma snel erg onoverzichtelijk. Veel chaos, en daardoor ellende, kan al worden voorkomen door bepaalde PASCAL commando's te simuleren. PASCAL kent bijvoorbeeld het REPEAT-mechanisme:

```
REPEAT
<opdracht>;
UNTIL <conditie>
```

<opdracht> wordt net zo vaak herhaald totdat <conditie> waar is. In BASIC Kan je dat simuleren door:

```
1000 REM REPEAT
.... <opdracht>
....
2000 IF NOT <conditie> THEN GO TO 1000: REM
     UNTIL <conditie>
2010 ...
```

PASCAL kent ook het WHILE mechanisme:

```
WHILE <conditie> DO BEGIN
<opdracht>
END;
```

Ook hier wordt <opdracht> net zo vaak herhaald totdat de <conditie> waar is. Bij REPEAT wordt de <opdracht> minstens één keer uitgevord. Bij WHILE kan <opdracht> direct worden overgeslagen. WHILE kan ook worden gesimuleerd in BASIC:

```
1000 IF NOT <conditie> THEN GOTO 2010: REM
     WHILE <conditie> DO BEGIN
.... <opdracht>
....
2000 GOTO 1000: REM END;
2010 ...
```

Door veel van deze structuren en subroutines gebruik te maken wordt je programma overzichtelijker. Zet veel gebruikte subroutines voorin het programma en de subroutines die het programma minder vaak aandoet achterin, dit in verband met de snelheid van het programmma. Het programma wordt altijd van af de 1e lINE naar achteren doorzocht totdat de gezochte regel gevonden is. (I.v.m. de snelheid kan je eventueel rekenwerk beter buiten de subroutines houden). (Cees van Krimpen).

Moet het programma echter op meer dan één conditie kunnen reageren, dan wordt het al een stuk moeilijker, meer regels met IF NOT geven vaak niet de gewenste oplossing.

```
1000 REM REPEAT
.... <opdracht>
....
2000 IF NOT <conditie1> THEN GO TO 1000: REM
     UNTIL <conditie1>
```

```
          2010 IF NOT <conditie2> THEN GO TO 1000: REM
               UNTIL <conditie2>
          2020 ...
```

(N.B. Regels 2000 en 2010 kunnen worden samen getrokken tot:

```
          2000 IF NOT <conditie1> AND NOT <conditie2>
               THEN GOTO 1000: REM UNTIL <conditie1> OR
               <conditie2> DO BEGIN )
```

Dit lijkt het nog wel te doen, zolang alletwee de condities niet waar zijn doorloopt het programma de lus 1000 - 2000/2010 , maar als één of meerdere condities waar zijn, gaat het programma altijd op de zelfde manier uit de lus, n.l. verder op LINE 2020.

Hoewel de check op het niet waar zijn van de condities zeer beslist nuttig kan zijn, is het vaak handiger om op de condities zelf te checken, zeker als je het waar zijn van de condities door middel van de waarde van een variabele aan kunt geven.
voorbeeld: Een Flatgebouw van 5 verdiepingen, elk met 9 appartementen. Nummer de etage's 1/m 5 en de woningen 1 t/m 9. Laat nu de variabele DOEL de waarde krijgen 10 x etagenr. + appart. nr. (zijn er maar dan 9 appartementen, neem dan 100 x etagenr.)
De variabele DOEL zal bij een gewenst bezoek aan appartm. 9 op de 4e. etage de waarde 49 hebben gekregen.

```
          1000 REM REPEAT
          1010 ....... :REM reageer op b.v. INPUT gegevens en
          1020 ....... :REM pas variabele DOEL zodanig aan,
          1030 ....... :REM dat er via regel 1080
          1040 ....... :REM naar de juiste GOSUB-routine
          1050 ....... :REM wordt gesprongen.
          1080 GO SUB 100 * DOEL: REM passende <opdracht>
          1090 GO TO 1000: REM UNTIL één of meerdere
               <conditie's> waar is/zijn. ('veiligheids' regel)
          2100 <opdracht conditie1>
          ....
          2190 RETURN
          2200 <opdracht conditie2>
          ....
          2290 RETURN
            enz.
```

Regel 1080 had ook als volgt mogen zijn:

```
          1080 GO SUB 100 * DOEL: REM passende <opdracht>
          2190 moest dan GO TO 1000 zijn even als regel
          2290 dan GO TO 1000 moet zijn.
```

De GO SUB-constructie heeft echter mijn voorkeur om dat je dan de diverse SUB-routines gemakkelijk vanaf een ander moment in je programma ook gemakkelijk aan kunt roepen, en je ze desgewenst ook nog kunt 'koppelen'. JWK.

Nog een paar tips bij het programeren in BASIC:
- Je kunt voorkomen dat je de RAMTOP per ongeluk te hoog zet door een verkeerde CLEAR <adres> opdracht. Dit kan als volgt: zet RAMTOP op de hoogste toegestane waarde, (dit is één minder dan de start van je machinecode), door middel van CLEAR <adres>.

Voer daarna uit:

```
POKE 23732, PEEK 23730: POKE 23733, PEEK 23731 .
```

Indien je, of je programma, daarna een CLEAR opdracht geeft boven de toegestane waarde, dan verschijnt de foutmelding:

```
'RAMTOP no good'
```

Erg handig met TOOLKITs, assemblers, monitors enz. CvK.

- Je hebt een printer en wilt wel eens wat .....m

De mogelijkheid van LPRINT en LLIST is aanwezig, maar hetgeen er op je beeldscherm staat kan je niet met behulp van COPY laten printen. Voeg dan deze sub-routine eens aan je programma toe:

```
10 FOR r = 0 TO 21
20 FOR k = 0 TO 31
30 LPRINT SCREEN$ (r,k);
40 NEXT k
50 NEXT r
```

Ook te gebruiken als printertestje. H. Boon.

INSTANT SCREEN Deze keer gaan we het hebben over SCREEN$. De Z80 processor die in de Spectrum zit kan 64k geheugen tegelijk aan. Dat is totaal 65536 bytes. Een byte wordt normaal gesproken gebruikt om een karakter op te slaan, en is de hoeveelheid geheugen die gemanipuleerd wordt door PEEK of POKE.

Je kunt het gemakkelijkste deze 65536 bytes voorstellen als 256 bladzijden van elk 256 bytes. De eerste 64 bladzijden, genummerd van 0 tot en met 63, worden gebruikt voor de ROM van de Spectrum. De RAM zit op bladzijde 64 t/m 255.

Een SCREEN gebruikt 27 bladzijden, en de bladzijden 64 t/m 90 zijn de pagina's die gebruikt worden voor LOAD ""SCREEN$.

De eerste 24 pagina's bevatten het monochrome raster van 256 * 192 puntjes en de laatste 3 bladzijden zijn voor de Kleuren-informatie. Als je extra SCREEN$ in het geheugen wilt stoppen moet je geheugenruimte CLEARen om te voorkomen dat deze overschreven wordt door BASIC.

Er is ruimte in het geheugen van de 48K Spectrum voor 5 SCREEN$, en je houdt dan 7K over voor een kort controle-programmaatje. Het is verstandig deze 5 SCREEN$ te starten op de bladzijden 120, 147, 174, 201 en 228. Om ruimte te CLEARen voor 5 SCREEN$ type je CLEAR 120*256-1 . Als je maar 2 SCREEN$ wilt opslaan, gebruik je ClEAR 201*256-1. Deze twee SCREEN$ moeten dan starten op blz. 201 en 228.

Je kunt ook een SCREEN rechtstreeks in het geheugen zetten met LOAD "" CODE <pagina> *256,6912 . Neem voor de waarde <pagina> 120, 147, 174, 201 of 288, al naar gelang van op welke plaats je het SCREEN wilt opslaan/gebruiken. Bladzijde 64 zet het geladen SCREEn direct op je tv of monitor. De hieronder geplaatste korte routine past op bladzijde 255, vlak na het laatste SCREEN en net voor het UDG-gebied.

```
10 CLEAR 120*256-1:REM (C) juni 1987 Simon N Goodwin
20 LET move =65280: LET from=56282: LET target=65285
30 RESTORE 50: FOR p=65280 TO 65291
```

```
 40 READ x: POKE p,x: NEXT p
 50 DATA 33,0,64,17,0,288,1,0,27,237,176,201
 60 LIST
 70 POKE from,64: POKE target,228: RANDOMIZE USR move
 80 CLS: PAUSE 0
 90 POKE from,228: POKE target,64: RANDOMIZE USR move
100 PAUSE 0: GO TO 80
```

De regels 10 t/m 50 van het voorbeeld-programma reserveren het geheugen en maken de Machinecode. Regel 60 zet iets op het scherm (je kunt ook LOAD ""SCREEN$ gebruiken). Regel 70 STOREd de display op bladzijde 288, terwijl regel 80 het scherm weer schoonmaakt en wacht op een toets indruk. Regel 90 brengt het SCREEN weer terug van bladzijde 228 naar het display-geheugen (bladzijde 64) en regel 100 wacht weer op een toetsdruk om het vanaf regel 80 allemaal weer over te doen. Het bladzijde nummer van de te verplaatsen bladzijde kan gemakkelijk worden veranderd door een nieuwe waarde in 65285 te POKEn.

Nog een programmaatje, dat bijna het zelfde doet, dit keer als assembler-listing:

```
 10 ;STORE / RESTORE SCREEN
 20 ;COPYRIHGT 1985  P.E. PRAK
 30
 40          ORG  40000
 50
 60          JR   START
 70 STORE    LD   HL,16384
 80          LD   DE,40100
 90          LD   BC,6411
100          LDIR
110          LD   A,#00
120          LD   (UP),A
130          RET
140 LOAD     LD   HL,40100
150          LD   DE,16384
160          LD   BC,6411
170          LDIR
180          LD   A,#01
190          LD   (UP),A
200          RET
210 UP       DEFB #01
220 START    LD   A,(UP)
230          CP   #01
240          JR   Z,STORE
250          CP   #00
260          JR   Z,LOADE
270          RET
```

Dit programma zet een beeld dat op het scherm staat tijdelijk in het geheugen. Dat wil zeggen dat het beeld na RANDOMIZE USR 4E4, het beeld in het geheugen is opgeslagen. Maak je nu de scherm-afbeelding anders en je wilt de oude afbeelding weer terug, dan nog een keer RANDOMIZE USR 4E4 en hij is weer terug op het scherm. De eerst volgende RANDOMIZE USR 4E4 zal de schermafbeelding van dat moment weer opslaan. (om en om dus).

MEER BASIC PROGRAMMEER TRUKJES

door: Johan Koning

* PRINT wordt LPRINT.
Met een simpele opdracht kan je PRINT-commando's in een programma wijzigen in LPRINT-commando's. Voer dan de volgende regel in: LET p=0: OPEN #2;"p"
Alle opdrachten gelden nu voor de printer en neit meer voor het beeldscherm. Wil je weer terug naar het beeldscherm, toets dan in: CLOSE #2, en het PRINT-commando voor het scherm geldt weer.

Om tijdens het SAVEn van meerdere files naar de cassetterecorder niet telkens gevraagd te worden om op een toets te drukken, zet u POKE PEEK 23631+PEEK 23632*256+2,181 tussen de SAVE-opdrachten Dit SPAARt u bovendien ruimte op de tape en tijd bij het LOADEN.

Met
POKE 23791,(gewenste aantal copiën) zorgt u er voor dat er meer dan 1 copie (nl. aantal), naar de cassette gesaved worden. Dit neemt dan wel meer ruimte in op de cassette, maar daarvoor krijgt u dan gelijk een aantal backups terug.

Melden we al eerder de POKE's 23658,8 (C-mode) en 23658,0 (L-mode), nog gemakkelijker kan het met RANDOMIZE USR 4217, dit geeft namelijk een wisseling tussen deze beide modes.

Ook op/met het scherm valt leuk te stoeien.
Eerst enkele regels om te scrollen en te wissen:

De bovenste 'helft' van het scherm (regels 0 t/m 21) van kanaal "S" is gekoppelt aan stroom #2, de onderste 'helft' (regels 22 en 23) is gekoppel aan #0 en #1, regel 22 en 23 zijn op die manier regels 0 en 1 van kanaal "K". (IPUTs en meldingen komen dus in "K"-regel 1).

RANDOMIZE USR 3280 geeft elke regel de inhoud van de volgende: die van "S"-regel 0 verdwijnt, die van "K"-regel 0 komt nu in "S"-regel 21 met "S"-kleuren, en de (nu) lege "K"-regel 1 houdt de "K"-kleuren.

RANDOMIZE USR 3583 doet zowat hetzelfde, alleen krijgt "S"-regel 21 nu tijdelijk de "K"-kleuren (BORDER-) en "K"-regel 1 krijgt nu (tijdelijk) de "S"-kleuren.

RANDOMIZE USR 3330 doet alweer iets dergelijks, zij het dat nu elke schermregel 21 keer de inhoud van de volgende regel krijgt. (Je laat dus eigenlijk bijna het hele scherm regel voor regel leeg scrollen).

RANDOMIZE USR 3652 wist de schermregels 10 t/m 23, waarbij ook de "K"-regels tijdelijk de kleur van het "S"-scherm te zien geven.

RANDOMIZE USR 3438 wist het "K"-scherm. Een instructie die het zelfde doet maar minder bytes 'verbruikt', is INPUT ; na deze instructie heeft het "K"-scherm zijn initiele 2 regels weer terug.

Als RANDOMIZE USR 3652 te veel en RANDOMIZE USR 3438 te weinig regels wist, dan kan de volgende tip misschien helpen.
INPUT AT <regel>,0; wist het scherm schoon van af regel <regel>.
LET OP! De laatste PRINT positie voor deze instructie moet altijd boven de <regel> zijn. Dus met b.v.:

```
.... PRINT AT 0,0:INPUT AT 12,0;
```

wordt het scherm vanaf regel 12 probleemloos schoon geveegd.

Bovenstaande tip gebruikt heel wat minder geheugen en is desondanks ook nog veel sneller dan de lus:

```
FOR f=12 TO 21: PRINT AT f,0;"32 spacies": NEXT f
```

Ook zijn met deze lus de regel 22 en 23 nog niet gewist.
Toch kan deze FOR/NEXT-lus nuttig blijken, en wel als je niet tot en met de onderste regel wilt wissen. PRINT AT f,0;, geeft het zelfde resultaat als PRINT AT f,0;"32 spacies" en scheelt weer enige bytes.
U wilt geen hele regels wissen, maar slecht een klein gedeelte (window?), gebruik dan: PRINT AT f,<kolom>;"gewenst aantal spacies" en op het scherm zullen in de regels, die aan de f-waarden voldoen, vanaf kolom <kolom> het gewenste aantal karakterblokken gewist worden.
Ook hier kan het weer korter: de , in PRINT, betekent eigenlijk PRINT op deze regel vanaf de laatste kolompositie op deze regel tot AAN de eerst volgende schermhelft (16 of 32) spacies.
Dus PRINT AT f,3;, zal op de regels, die aan de voorwaarde f voldoen, telkens vanaf kolom 3 tot aan kolom 16 spacies printen, en daarmee de vorige inhoud wissen.
Om nu bijvoorbeeld een window dat start op regel 7, kolom 10 (linkerbovenhoek), en eindigt op regel 15, kolom 20 (rechterbenedenhoek), te wissen, gebruik je de volgende FOR/NEXT-lus:

```
FOR f=7 TO 15: PRINT AT f,10;,"5 spacies": NEXT f
```

De FOR/NEXT-lus bepaalt de regels, de AT f,10;, zorgt voor het wissen van de kolomposities 10 t/m 15 en de 5 spacies zorgen voor het wissen van de posities 16 t/m 20.

Door UDG's en (tekst)schermen als code-files op te slaan (saven) en dus ook weer als code te laden en gebruiken, met b.v. RANDOMIZE USR xxxx, hoef je deze gegevens niet meer in het BASIC-programma te hebben staan, dus minder regels, en daardoor een sneller programma.
N.B. Als je SCREEN$ moet laden en vaak gebruiken, laad ze dan op een 'vrije' geheugenruimte als CODE, en zet ze zodra ze nodig zijn met een bij "INSTANT SCREEN" omschreven routine op het scherm, dat is sneller dan ze telkens te moeten laden.

TW III EN PRINTERCONTROLCODES

TASWORD III PRINTECONTROLCODES:

Idereen die met een tekstverwerkingsprogramma wekt, krijgt vroeger of later te m2ken met printercontroletekens. In Tasword II en III kunnen we een aantal grafische tekens gebruiken, om opdrachten naar de printer te sturen. Heel vaak worden bijvoorbeeld stukken tekst onderstreept. Handig is het, dat een aantal van de printercontroletekens al zijn gedefinieerd in het programma, zoals je dat koopt. Enkele daarvan zijn uitstekend bruikbaar. Andere heb niets aan, omdat in de Tasword tekstverwerkings programma's de tekens zijn gedefinieerd voor een Epson printer. En met printers is het net zo als met computers, er zijn geen twee gelijk. Er zit niets anders op dan de tekens zelf te definieren. En dan komen gelijk de problemen, want er zijn zoveel mogelijkheden, dat je door de bomen het bos niet meer ziet. Daarom lijkt het mij handig om het een en ander uit te leggen, door midel van een artikel in het Bulletin.

Tijden het voorafgaande gedeelte heb ik al gezegd, dat de codes voor het onderstrepen van tekst heel vaak worden gebruikt. Hoe ik deze heb gedefinieerd zal ik het eerst uitleggen. Met de codes: 27 45 1 geef je aan de printer de opdracht om alle volgende tekst te onderstrepen, met de codes: 27 45 0 zet je het onderstrepen weer uit. Tot zover niets nieuws. In de tekst nemen de controle tekens uiteraard de plaats in van een letter. De controle tekens worden echter niet op het scherm afgedrukt. Ook de ruimte die een controleteken op het scherm inneemt, wordt niet als spacie naar de printer gestuurd. Dit heeft twee nare gevolgen.

1e. De rechterkantlijn gaat tijdens het afdrukken verloren, omdat er enkele tekens minder worden afgedrukt, dan op het beeldscherm staan.

2e. Je moet na het plaatsen (of er voor) van een controleteken altijd een spacie gebruiken, omdat anders de woorden, waartussen een controleteken staat, aan elkaar worden gedrukt.

Zelf heb ik dit probleem opgelost door bij het definieren van de grafische tekens, aan de reeksen van controletekens telkens een teken voor een spacie toe te voegen.
De tekens voor het onderstrepen zien er bij mij dan ook zo uit:

onderstrepen AAN: 32 27 45 1 (32 is de code voor een spacie)

onderstrepen UIT: 27 45 0 32

Het zal sommigen zijn opgevallen, dat de code vooe een spacie de ene keer aan het begin van de reeks codes staat, en de andere keer aan het einde van de reeks.
De uitleg is simpel: een spacie wordt gezien als een letterteken en de printer onderstreept alle lettertekens. Het zou erg slordig staan, als de printer te vroeg begint, of er te laat mee eindigt. Door het toevoegen van een spaciecode aan de controletekens, wordt er op de plek van de controleteken netjes een spacie afgedrukt. De rechterkantlijn blijft netjes, je hoeft geen extra spacie meer te gebruiken bij een controle teken om te voorkomen dat woorden een elkaar vast geprint worden en, door de plaats van deze spaciecode in de reeks begint en eindigt het onderstrepen altijd op de gewenste plaats.

Bert Westenburg.

TW III EN NLQ OP DE FASTEXT 80

PSEUDO NLQ VOOR DE COMBINATIE TASWORD III EN DE FASTEXT 80.

Met het volgende stukje Basic is het mogelijk een tekst van T.W. 3 twéé keer te printen. De letters worden vetter afgedrukt. Het is echter niet mogelijk om de printercodes te verwerken.
Omdat er in T.W. 3 niet genoeg ruimt is voor alle Basicregels, heb ik er twee stukjes van gemaakt.
Ook moet je even nadenken als je een gewijzigde versie wilt wegschrijven. Bert Westenburg bracht mij op het idee, twee artikelen uit de Sinclair Impuls hebben mij hierbij geholpen en E.H.F. Weijers beschreef op welke adressen de tekstlengte staat.

Het standaard stukje Basic heb ik aangevuld met:

```
70 MERGE *1;"TWmerge"
```

Het volgende deel wordt weggeschreven met:

```
SAVE *1;"TWmerge"
```

```
 5 GO TO VAL "20"
10 RANDOMIZE USR VAL "25000"
20 DIM a$(1,65): LET L=VAL "1"
30 FOR a=VAL "47875" TO VAL "47875"+PEEK VAL "25171"+PEEK VAL
"25172"*VAL "256": LET d=PEEK a: IF d=VAL "0" THEN GO TO VAL
"50"
40 LET a$(VAL "1",L)=CHR$ d: LET l=l+VAL "1": NEXT a
50 LET t=VAL "0"
```

```
60 LPRINT "        ";a$(VAL "1" TO L-VAL "1"): LET t=t+VAL "1":
IF t=VAL "2" THEN LET l=VAL "1": LPRINT CHR$ (VAL "10"): NEXT
a: GO TO PI*PI
70 GO TO VAL "60"
```

Om het te gebruiken:
Eerst via het Hoofdmenu naar Basic en dan 2x RUN.
de 2e. keer vanwege het MERGEn.

F. Grunefeld.

TW III EN JE EIGEN PRINTER

Het aanpassen van Tasword III aan je eigen printer

Sinds enige tijd heb ik een STAR NL 10 printer. Deze printer kent een groot aantal karakters die niet op de SPECTRUM voorkomen, waaronder breuken, griekse letters en diverse graphics. (De z.g.n. IBM karakterset.) Tasword III kan deze tekens weliswaar naar de printer sturen d.m.v. zijn aan te passen tweede karakterset, maar deze tekens wilde ik nu ook graag op mijn beeldscherm.

Om de naar het scherm te printen karakters te veranderen heb ik opgezocht waar die waren opgeborgen. Dat is in TASTABLE, maar..... op 2 plaatsen. TW III kent immers de mogelijkheid van 32 of 64 karakters op een regel, en gebruikt daarvoor twee verschillende karaktersets!

Op hokjes-papier heb ik toen 2 keer 75 karakters "getekend": de kleine karakters van 4 pixels breed en 8 pixels hoog en de grote van 8 x 8. Voor elke karakter kun je dan 8 getallen vinden aan de hand van de zwartgemaakte hokjes.

Om deze getallen nu in TASTABLE te kunnen POKEN heb ik een basicprogrammaatje gemaakt en de decimale getallen, per karakter 8, in DATA regels geplaatst. Dat is weer handig om foutjes te korrigeren of veranderingen uit te voeren. Daaromheen een aantal basic-regels om op het scherm de vorm van de letters en tekens te kunnen kontroleren.

Op de volgende pagina vind je de listing. Als je deze ingetypt hebt, (en gesaved), load dan het codeblok "TASTABLE" van TW III. Dit kan via r.200 in mijn programma. Dan worden na GOTO 500 de nieuwe karakters ingepoked. Nu kun je het gewijzigde codeblok saven als "TASTABLE2" CODE 37888,6656. (r.220) Met RUN kun je de aangemaakte karakterset op het scherm afbeelden, ook wordt steeds het betreffende geheugenadres vertoond . Hierbij wel steeds een toets indrukken totdat beide sets op het scherm zijn verschenen.

Het TW III "run"programma moet nu gewijzigd worden door een "2" toe te voegen aan "TASTABLE" (r.50). Na loaden van TW III moet je nu via de menukeuze :AANPASSEN PROGRAM, aan je tweede karakterset de kodes meegeven die voor je printer nodig zijn, deze zijn te vinden in het printerhandboek. Om ook deze nieuwe wijzigingen blijvend te maken opnieuw "TASTABLE2" CODE saven. (Via de TERUG naar BASIC optie.)

N.B.:1e:Er worden een in de karakterset-tabel een aantal geheugenplaatsen niet gebruikt, daar heb ik dus een aantal extra nullen toegevoegd aan de DATA.
2e:De pijltjes achter i,j,k,l in de "2e kar set" worden ook gebruikt in de "cursorbesturing". Deze dus liever niet veranderen.(of gebruik mijn getruukte oplossing; zie programma)
3e:Het vertalen van karakters in decimale getallen wordt hier bekend verondersteld. Maar uitleg hiervan is natuurlijk altijd nog mogelijk.
4e:Voor de (Engelse) versie van TW III waarbij vanuit het menu een backup-save gemaakt wordt, geldt dat de aangepaste TASTABLE weer gewoon TASTABLE heet, zonder de "2" dus. Denk hierom in het "run"prog.
5e:LOAD en SAVE opdrachten in het basicprogramma moet je evt. aanpassen, ik werk n.l. met de OPUS Discovery.

Veel plezier met dit m.i. nuttige programma.

b.koning (bert) Niehove 05949-396

```
  1 REM * B.©.Koning -1988-
 10 REM **karakters naar scherm
 15 LET a=39424: REM kleine kar
 16 REM a=40704 grote kar
 17 LET t=0
 20 LET f=16384
 60 FOR i=0 TO 7
 70 POKE f+256*i,PEEK (a+i)
 80 NEXT i
 85 PRINT AT 21,1;a
 90 LET f=f+1: LET a=a+8
100 PAUSE 0: GO TO 60
200 CLEAR 37800: LOAD *1;"TASTA
BLE2"CODE : STOP
220 SAVE *1;"TASTABLE2"CODE 378
88,6656: STOP
500 REM **poken 2e set klein
505 RESTORE : REM 75*8-1 erbij
510 FOR f=39424 TO 40047
550 READ b: POKE f,b
560 NEXT f
570 BEEP .2,30
600 REM data kleine letter
601 DATA 15,15,15,15,15,15,15,1
5
602 DATA 0,0,0,3,2,2,2,2
603 DATA 0,0,0,15,2,2,2,2
604 DATA 0,0,0,14,2,2,2,2
605 DATA 2,2,2,3,2,2,2,2
606 DATA 2,2,2,15,2,2,2,2
607 DATA 2,2,2,3,0,0,0,0
608 DATA 2,2,2,15,0,0,0,0
609 DATA 2,2,2,14,0,0,0,0
610 DATA 2,2,2,14,2,2,2,2
635 DATA 0,2,5,5,6,5,4,0
636 DATA 0,0,15,5,4,4,4,0
637 DATA 0,0,2,5,5,5,5,0
638 DATA 7,4,2,1,2,4,7,0
639 DATA 0,3,2,3,2,2,6,0
640 DATA 2,5,5,2,5,5,2,0
641 DATA 12,12,12,12,12,12,12,1
2
642 DATA 3,3,3,3,3,3,3,3
643 DATA 2,5,5,4,4,4,4,0
644 DATA 15,15,15,15,0,0,0,0
645 DATA 0,0,0,0,15,15,15,15
646 DATA 5,5,13,0,13,5,5,5
647 DATA 0,7,5,5,0,0,0,0
648 DATA 0,2,5,5,5,2,7,0
649 DATA 0,0,0,15,5,5,5,0
650 DATA 0,7,2,5,5,2,7,0
651 DATA 0,6,4,2,5,5,2,0
652 DATA 0,0,1,6,5,5,2,0
653 DATA 0,0,1,2,6,2,2,0
654 DATA 0,0,0,5,5,5,10,0
655 DATA 1,6,9,11,13,9,6,0
656 DATA 0,3,2,10,10,10,4,0
657 DATA 2,2,14,2,14,2,2,2
658 DATA 5,5,13,0,15,0,0,0
659 DATA 3,6,10,15,10,10,11,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0
660 DATA 0,2,2,7,7,2,2,0
661 DATA 2,7,2,7,7,2,2,0
662 DATA 5,5,5,4,5,5,5,5
663 DATA 1,2,7,1,3,5,3,0
664 DATA 5,5,13,1,13,5,5,5
665 DATA 5,0,6,2,2,2,7,0
```

wordt vervolgd.

```
611 DATA 0,0,15,0,15,0,0,0
612 DATA 2,2,7,2,2,0,7,0
613 DATA 5,5,5,5,5,5,5,5
614 DATA 0,0,0,15,0,0,0,0
615 DATA 2,2,2,2,2,2,2,2
616 DATA 0,0,2,0,7,0,2,0
617 DATA 2,5,2,0,0,0,0,0
618 DATA 4,12,4,4,3,1,2,7
619 DATA 2,5,1,2,7,0,0,0
620 DATA 3,4,4,7,4,4,3,0
621 DATA 4,4,4,0,5,5,7,1
622 DATA 0,0,7,4,5,5,5,5
623 DATA 0,0,15,0,15,2,2,2
624 DATA 0,0,15,1,13,5,5,5
625 DATA 5,5,5,4,7,0,0,0
626 DATA 5,5,13,1,15,0,0,0
627 DATA 0,0,7,0,7,0,7,0
628 DATA 10,10,5,5,10,10,5,5
629 DATA 1,2,4,2,5,2,1,0
630 DATA 0,0,0,10,5,10,5,0
631 DATA 4,2,1,2,5,2,4,0
632 DATA 10,5,10,5,10,5,10,5
633 DATA 2,2,15,0,15,0,0,0
634 DATA 0,0,0,5,10,10,5,0
666 DATA 10,10,10,11,10,10,10,1
0
667 DATA 5,0,2,5,7,4,3,0
668 DATA 5,5,5,13,5,5,5,5
669 DATA 0,0,15,0,13,5,5,5
670 DATA 10,0,5,0,10,0,5,0
671 DATA 0,1,3,6,3,1,0,0
672 DATA 0,4,6,3,6,4,0,0
673 DATA 0,0,6,6,6,15,6,0
674 DATA 0,6,15,6,6,6,0,0
675 DATA 2,2,3,2,3,2,2,2
800 REM poken grote kar
805 RESTORE 900
810 FOR f=40704 TO 41327
830 READ b: POKE f,b
840 NEXT f
850 BEEP 1,35
861 DATA 0,40,20,10,20,40,0,0
900 REM data grote kar
901 DATA 255,255,255,255,255,25
5,255,255
902 DATA 0,0,0,15,8,8,8,8
903 DATA 0,0,0,255,8,8,8,8
```

```
904>DATA 0,0,0,248,8,8,8,8
905 DATA 8,8,8,15,8,8,8,8
906 DATA 8,8,8,255,8,8,8,8
907 DATA 8,8,8,15,0,0,0,0
908 DATA 8,8,8,255,0,0,0,0
909 DATA 8,8,8,248,0,0,0,0
910 DATA 8,8,8,248,8,8,8,8
911 DATA 0,0,255,0,255,0,0,0
912 DATA 8,8,62,8,8,0,62,0
913 DATA 20,20,20,20,20,20,20,2
0
914 DATA 0,0,0,255,0,0,0,0
915 DATA 8,8,8,8,8,8,8,8
916 DATA 24,24,0,124,0,24,24,0
917 DATA 24,36,24,0,0,0,0,0
918 DATA 64,72,80,44,82,4,8,14
919 DATA 24,36,8,16,28,0,0,0
920 DATA 12,16,32,60,32,16,12,0
921 DATA 64,72,80,36,84,20,30,4
922 DATA 0,0,31,16,23,20,20,20
923 DATA 0,0,255,0,255,8,8,8
924 DATA 0,0,252,4,244,20,20,20
925 DATA 20,20,23,16,31,0,0,0
926 DATA 20,20,244,4,252,0,0,0
927 DATA 0,0,60,0,60,0,60,0
928 DATA 170,85,170,85,170,85,1
70,85
929 DATA 4,8,16,8,4,16,8,4
930 DATA 0,0,50,76,0,50,76,0
931 DATA 16,8,4,8,16,4,8,16
932 DATA 168,84,168,84,168,84,1
68,84
933 DATA 8,8,255,0,255,0,0,0
934 DATA 0,0,49,74,68,74,49,0
935 DATA 28,34,34,36,34,38,32,3
2
936 DATA 0,62,34,32,32,32,32,0
937 DATA 0,28,34,34,34,34,34,0
938 DATA 62,34,16,8,16,34,62,0
939 DATA 12,18,16,24,16,16,48,0
940 DATA 0,0,54,73,73,54,0,0
941 DATA 240,240,240,240,240,24
0,240,240
942 DATA 15,15,15,15,15,15,15,1
5
943 DATA 12,18,16,16,16,16,16,1
6
944 DATA 255,255,255,255,0,0,0,
0
945 DATA 0,0,0,0,255,255,255,25
5
946 DATA 20,20,247,0,247,20,20,
20
947 DATA 44,52,36,36,36,0,0,0
948 DATA 0,28,34,34,34,20,54,0
949 DATA 0,0,62,20,20,20,20,0
950 DATA 28,8,20,34,34,20,8,28
951>DATA 24,36,16,56,68,68,56,0
952 DATA 0,1,62,36,36,36,24,0
953 DATA 0,0,62,72,8,8,8,0
954 DATA 0,0,0,36,36,36,58,64
955 DATA 0,26,36,44,52,36,88,0
956 DATA 7,4,100,36,36,40,16,0
957 DATA 8,8,248,8,248,8,8,8
```

```
 958 DATA 20,20,247,0,255,0,0,0
 959 DATA 30,40,40,62,40,40,46,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0
 960 DATA 8,28,62,127,62,28,8,0
 961 DATA 28,28,42,127,42,8,28,0
 962 DATA 20,20,23,16,23,20,20,2
0
 963 DATA 4,8,56,4,60,68,58,0
 964 DATA 20,20,244,4,244,20,20,
20
 965 DATA 20,0,24,8,8,8,28,0
 966 DATA 20,20,20,23,20,20,20,2
0
 967 DATA 36,0,24,36,60,32,28,0
 968 DATA 20,20,20,244,20,20,20,
20
 969 DATA 0,0,255,0,247,20,20,20
 970 DATA 170,0,170,0,170,0,170,
0
 971 DATA 0,20,40,80,40,20,0,0
 972 DATA 0,40,20,10,20,40,0,0
 973 DATA 34,119,127,62,28,28,8,
0
 974 DATA 8,28,62,62,119,8,28,0
 975 DATA 8,8,15,8,15,8,8,8
9000 STOP
9999 SAVE *1;"KARSCHERM"
```

TW III EN DE IBM KARAKTERSET

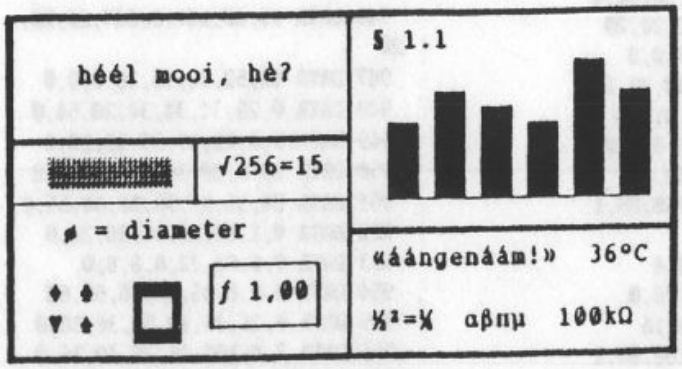
Het is bekend dat alle goede dingen langzaam komen, en dus kwam ik er nu pas toe om het programma van Bert Koning (inderdaad familie) uit het bulletin van maart 1988 nuttig te gaan gebruiken.

Ik heb de data in TASTABLE gepoked, nog een paar wijzigingen aangebracht omdat ik ook graag de 'é' en de 'è' wilde gebruiken, en de nieuwe TASTABLE gesaved.

Toen kon ik beginnen met het aangeven van de kodes voor uitvoer naar de printer. Dat viel nog even tegen, zevenenzeventig nummers opzoeken in m'n printerhandboek, terwijl al die graphics zo verdraaid veel op elkaar lijken! Maar nu draait het spul, en oh wat gaat dat mooi!

Ter illustratie heb ik even iets in elkaar gezet waarvan je hiernaast twee afdrukken kunt zien, met, ter vergelijking, een screendump van het oorsponkelijke TASWORD-scherm.

Omdat bijna alle printers wel een IBM karakterset bezitten en de code's standaard zijn, geef ik hieronder een lijstje waar uit af te lezen valt welke code bij welk karakter hoort, en welke toets ervoor ingedrukt moet worden. Ook de bijbehorende DATA-regel (uit Bert's programma) is vermeld.



TOETS	CHAR	CODE	REGno.s	TOETS	CHAR	CODE	REGno.s
space		219	601/901	F	ƒ	159	639/939
!	┌	218	602/902	G	╼	236	640/940
"	┬	194	603/903	H	╽	221	641/941
#	┐	191	604/904	I	┃	222	642/942
$	├	195	605/905	J	⌠	244	643/943
%	┼	197	606/906	K	■	223	644/944
&	└	192	607/907	L	▄	220	645/945
'	┴	193	608/908	M	╪	206	646/946
(	┘	217	609/909	N	**	252	647/947
)	┤	180	610/910	O	Ω	234	648/948
*	▬	205	611/911	P	π	227	649/949
+	±	241	612/912	Q	Φ	232	650/950
,	▌	186	613/913	R	δ	235	651/951
-	_	196	614/914	S	σ	229	652/952
.	|	179	615/915	T	τ	231	653/953
/	÷	246	616/916	U	µ	230	654/954
0	°	248	617/917	V	ø	237	655/955
1	½	171	618/918	W	√	251	656/956
2	²	253	619/919	X	┤	181	657/957
3	ε	238	620/920	Y	┴	202	658/958
4	¼	172	621/921	Z	**	146	659/959
5	┏	201	622/922	[	♦	4	660/960
6	┳	209	623/923	\	♣	5	661/961
7	┓	187	624/924	]	╞	204	662/962
8	┗	200	625/925	a	á	160	663/963
9	┛	188	626/926	b	┫	185	664/964
:	■	240	627/927	c	ī	139	665/965
;	▒	178	628/928	d	╠	199	666/966
<	≤	243	629/929	e	ë	137	667/967
=	≈	247	630/930	f	▒	182	668/968
>	≥	242	631/931	g	╥	203	669/969
?	**	177	632/932	h	┸	176	670/970
@	┴	207	633/933	i	«	174	671/971
A	α	224	634/934	j	»	175	672/972
B	β	225	635/935	k	♥	3	673/973
C	Γ	226	636/936	l	♠	6	674/974
D	∩	239	637/937	m	╞	198	675/975
E	Σ	228	638/938	** = niet afgebeeld hier			

Misschien kan de redaktie, voor wie deze uitbreiding toch onmisbaar moet zijn, het programma in een komplete vorm nog eens bij de casette-tape-service te doen?

Roelof Koning.

P.S. Toch niet iedere printer blijkt over een IBM-karakterset te beschikken, o.a. de trouwe Smith-Corona laat het afweten, evenals (naar het schijnt) de Brother M1009.

UITBREIDING: 256K ROM +96K RAM

door: Roelof Koning

Twee vragen: Is dit mogelijk? En zo ja, wat kun je ermee? De eerste vraag is snel beantwoord: ja het kan, maar er is een tamelijk techniese ingreep in de SPECTRUM nodig. Het antwoord op de tweede vraag is moeilijker te geven, de gebruiksmogelijkheden hangen voornamelijk af van je eigen inventiviteit en vaardigheid in MachineCode. Zelf heb ik Ramdisk*6 van de Discovery versie2.2 "omgebouwd", ik beschik nu over een ramdisk van 223K (!) op mijn 48K SPECTRUM. Verder valt er te denken aan zaken zoals Basic-kommando uitbreidingen in een ROMbank, of het "implanteren" van een extra computertaal.

In dit artikel wil ik me overigens beperken tot de techniese kant. Het hier beschreven ontwerp is bedoeld om als hulpprint op de Spectrum-printplaat te worden gestoken, in een IC-voet die op de plaats van de ROMchip moet komen. Deze wijze van monteren bergt een aantal nadelen in zich, zoals moeilijk soldeerwerk en een kast die niet meer past. Aan de andere kant zijn daar de voordelen: een eenvoudige hulpprint, korte leidingen, minder kontaktovergangen en een vrije edge-connector. Doorslaggevend bij de keuze is uiteindelijk de overweging geweest dat er ook bij externe montage een portie soldeerwerk op de Spectrumprint nodig zou blijven. Voor het uitgebreide geheugen is namelijk gebruik gemaakt van dynamiese 256K Ramchips die op de Spec-print moeten worden gemonteerd op de plaats van de normale 32K-chips, en er moeten een paar aansluitingen gemaakt worden op signalen die niet op de edgeconnector aanwezig zijn.

Een probleem dat bij elke uitbreiding voor de Spectrum weer opduikt is het vinden van vrije IN/OUT adressen. Ik heb afgezien van compatibiliteit (kwa besturing) met de SPECTRUM-128, omdat de 8 Rambanken elk 32K groot zijn en het alternatief zou zijn geweest 16 banken van 16K en dit zou nogal wat extra chips gevergd hebben. Gekozen is om de Rambanken te schakelen dmv de kommando's IN 0 tm 7, en de Rombanken dmv IN 8 tm 10 en IN 12 tm 14.

Een ander probleem was het refreshen van zoveel geheugen, de architectuur van de Spectrum is daar niet op ingericht. Een ontwerp-idee uit Happy Computer bracht uitkomst, adreslijn 7 van de RAMchips wordt tijdens de refreshpuls aangesloten op een oscillator die deze lijn beurtelings hoog en laag maakt.

Voor de ROM's ben ik uitgegaan van 32K EPROMs waarvan voor elke bank de helft gebruikt wordt. Overigens kunnen ook 16K- en 8K EPROMs gemonteerd worden, deze bevinden zich dan onder de hoge bankno's. Uiteraard is het ook mogelijk de originele SPECTRUM-ROM te plaatsen, die zich dan (vanwege z'n extra ROMSELEKT aansluiting) onder de lage bankno's laat aanspreken.

Hier volgt de beschrijving van het schema.
IC1 (een comparator) decodeert IN-opdrachten waarbij de waarde op de ADRES-lijnen kleiner is dan 16. IC2a decodeert verder voor ADRESsen groter of kleiner dan 8. Afhankelijk van dit laatste onderscheid wordt IC3 of IC4 (latch) geaktiveerd, die dan de waarden van ADRES-lijnen 0-2 opslaat.
Decoder IC6a kiest dan de juiste (EP)ROM mbv de door A0 en A1 vastgelegde bits,en het door A3 vastgelegde bit bepaalt of de hoge of de lage helft werkzaam zal zijn. Een extra voorziening is nodig om de gekozen ROMbank te kunnen uitschakelen als Interface 1 of (in mijn geval) Opus Discovery in aktie komt. Dit wordt verzorgd door IC6a die hier als NANDpoort fungeert.
Welke RAMbank aktief is wordt bepaald door de in IC4 vastgelegde waarde, deze wordt gebruikt door de multiplexer IC5 die als het ware een verlengstuk vormt van de multiplex-adressering van de SPECTRUM zelf.
IC7 is geschakeld als oscillator (frekwentie ergens rond de 500 Herz) en deze verzorgt samen met IC2b, D1, en D2 het "knipperen" van A7 tijdens het refreshen. Om dit mogelijk te maken moet er wel een voorrangskwestie opgelost worden,dit kan met behulp van een weertandje a la Clive (R2). Deze moet (op de SPEC-print) opgenomen worden in de A7-lijn van de nieuwe RAMchips.
Verder is er voorzien in een automatiese reset van de latches bij het inschakelen. De timing (C1 en R1) hiervan is tamelijk krities wanneer de Opus Discovery aangesloten is.

Wenken mbt het nabouwen.
De hierbij afgebeelde layout van de benodigde hulp-printplaat spreekt (hoop ik) voor zichzelf, de IC's worden rechtstreeks op de print gesoldeerd en voor de (EP)ROM voeten moeten eksemplaren met gedraaide bussen gebruikt worden, ivm het solderen aan de bovenzijde van de print.
Dan nu puntsgewijs de werkzaamheden die op de SPEC-printplaat uitgevoerd moeten worden.

a. De 8 chips van het 32K geheugenblok uitsolderen, IC-voetjes insolderen, hier 8 chips van het type 41256 plaatsen nadat eerst pen 1 van deze chips omhoog gebogen is. Deze pennen komen dus niet in de voetjes, maar worden met elkaar verbonden en op aansluiting A8 vd hulpprint aangesloten.

b. De draadbrug tbv de H-L bankselekt verwijderen, de TI-OKI draadbrug moet over OKI. De vrije TI aansluiting (het dichtst bij de MIC-plug) verbinden met aansluiting BANK.

c. Pen 1 van IC26 (R/C selekt) verbinden met aansluiting SELEKT.

d. De verbinding van IC26 pen 9 met de SPECprint onderbreken, en deze onderbreking overbruggen met een weerstand van 470 ohm. (Dit is R2 in het schema.) De printplaatzijde van deze weerstand verbinden met de aansluiting A7bis.

e. Ioreq, Rd, Refresh, en eventueel Wr, (te vinden in

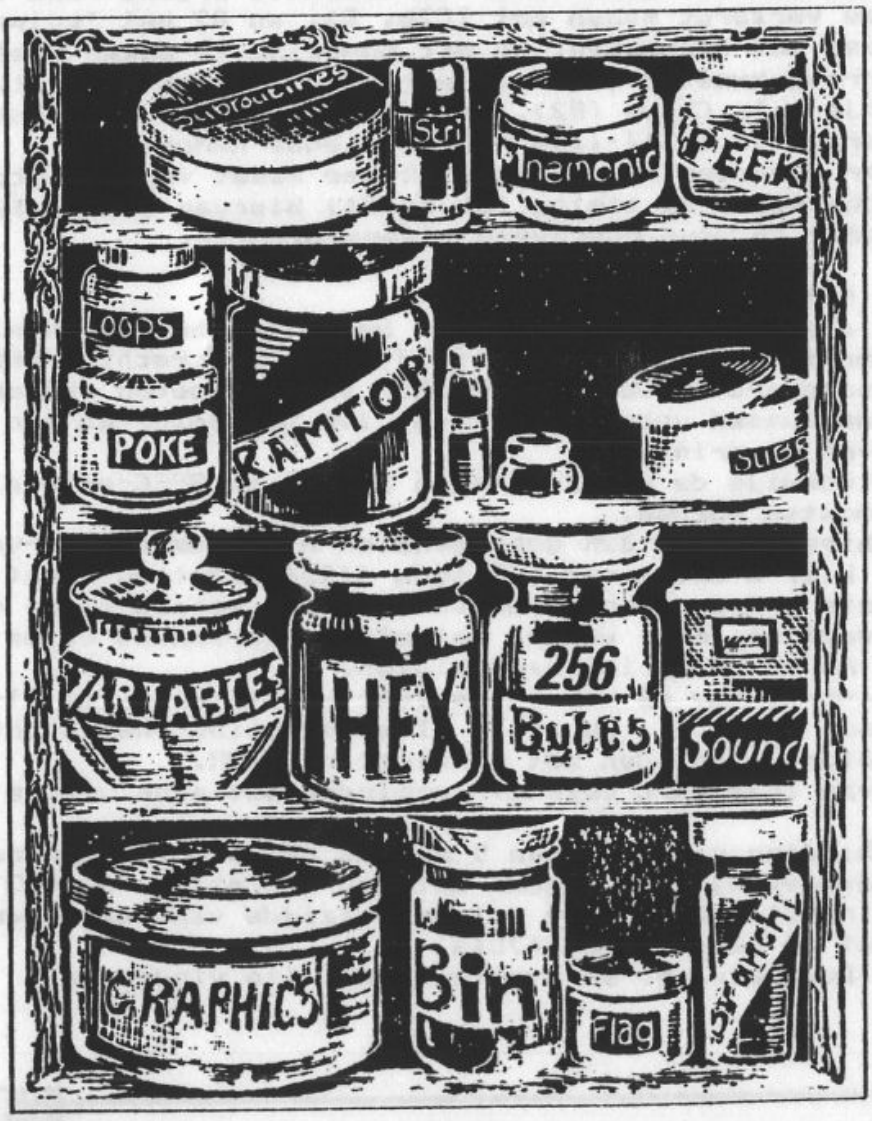
de buurt van de edgeconnector) verbinden met de betreffende aansluitingen op de hulpprint.

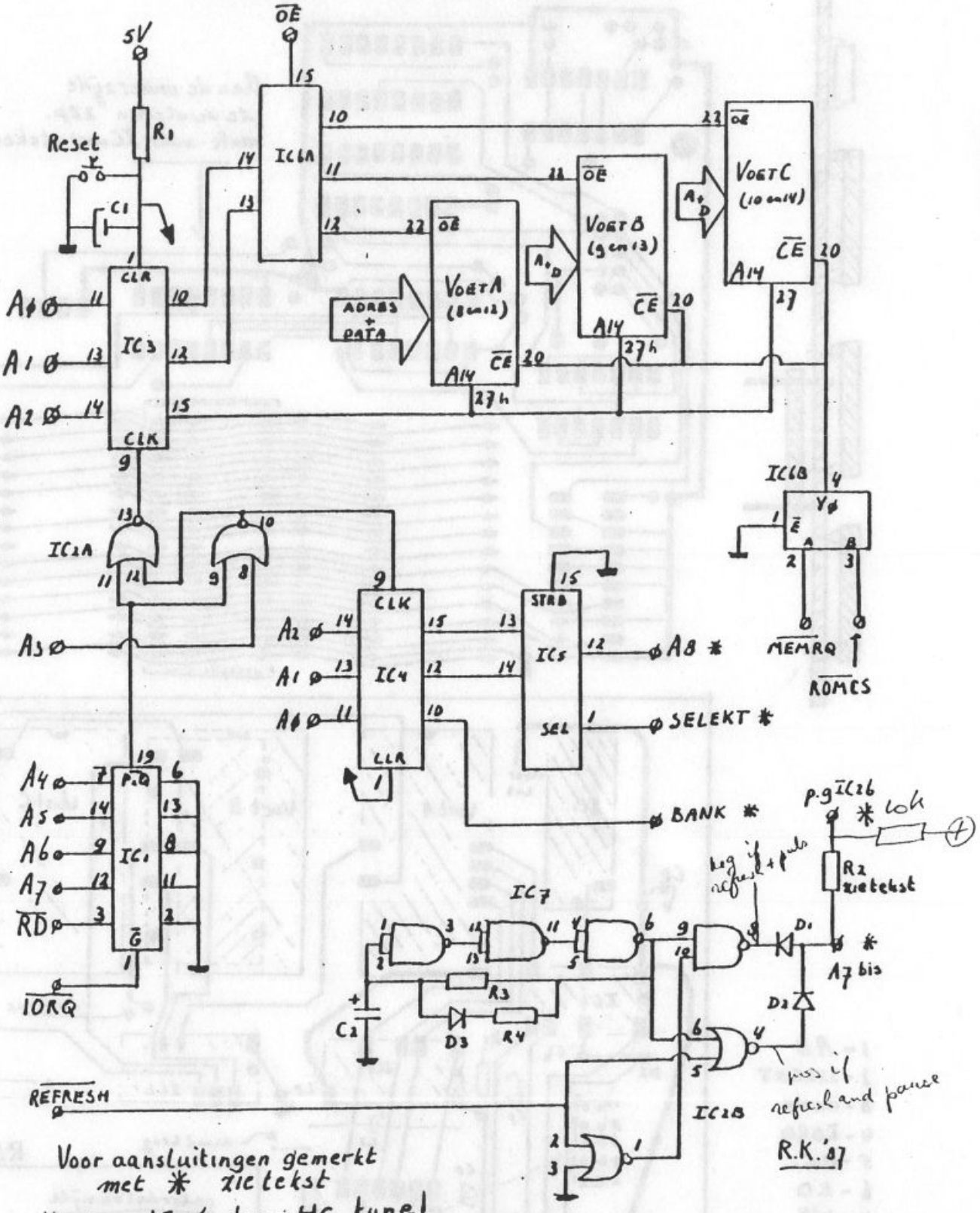
f. De SPEC-ROM (voorzichtig!).uitsolderen, hier een Wirewrap IC-voet zodanig insolderen dat deze ca. 1cm. boven de print uitsteekt. (Een normale voet kan ook, deze dan ophogen door er een extra voet op te steken.)

h. De bij de print-layout aangegeven Wr aanrluiting is niet nodig bij het werken met EPROMs, maar is bedoeld om te kunnen experimenteren met statiese RAMs ipv EPROMs.

i. Met de SPEC-ROM in voet A van de hulpprint zal, wanneer alles korrekt is aangesloten, de computer normaal opstarten en funktioneren.

Tot slot nog twee tips voor de HH technici die zich aan het nabouwen wagen: Door de aanwezigheid van een oscillerende chip op de print is een deugdelijke ontkoppeling van de 5V lijn uitermate belangrijk, met name voor een betrouwbare refresh.
En het tweede: gebruik bij het uitsolderen een goede tinzuigpomp en vet het zuigertje daarvan af en toe eens in.





Voor aansluitingen gemerkt met * zie tekst

Vervang IC 26 door HC type!

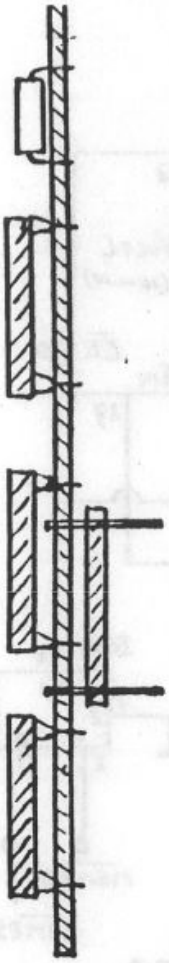
IC1 = 74LS688
IC2 = 74LS02
IC3 = 74LS174
IC4 = 74LS174
IC5 = 74LS157
IC6 = 74LS139
IC7 = 74LS132

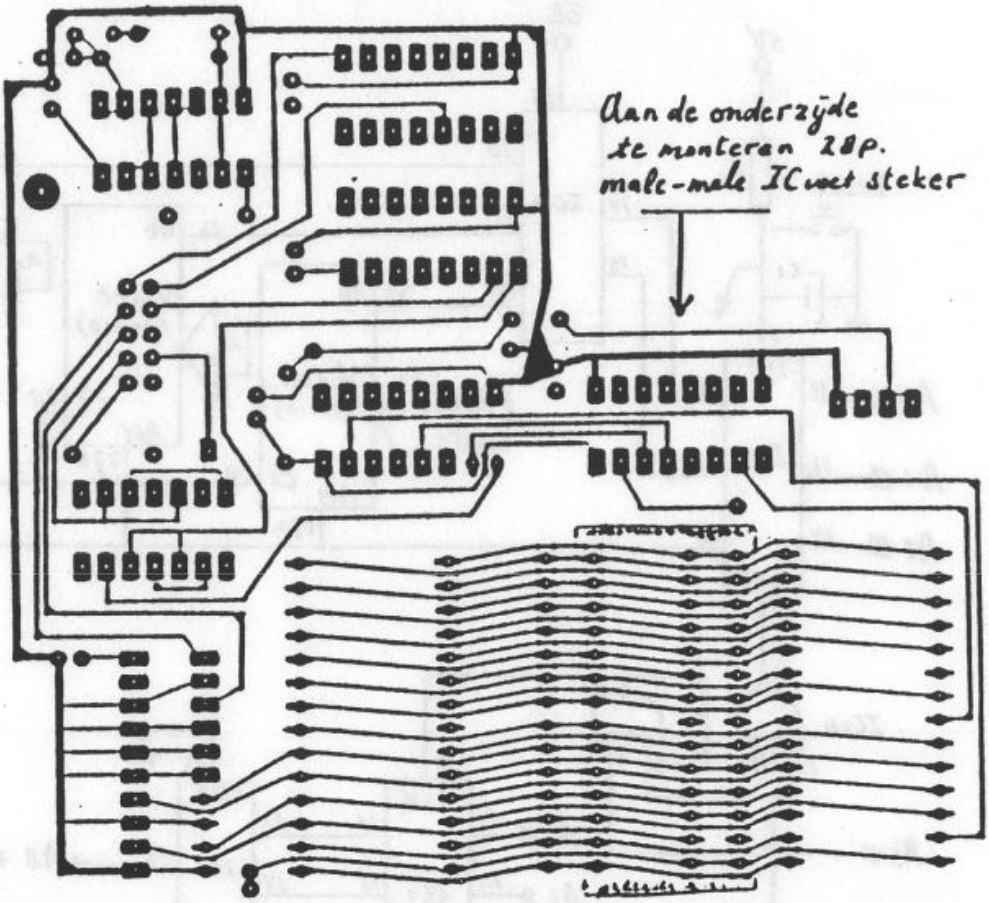
R1 = 2k7
R2 = 470Ω/330Ω
R3 = 1k
R4 = 100Ω

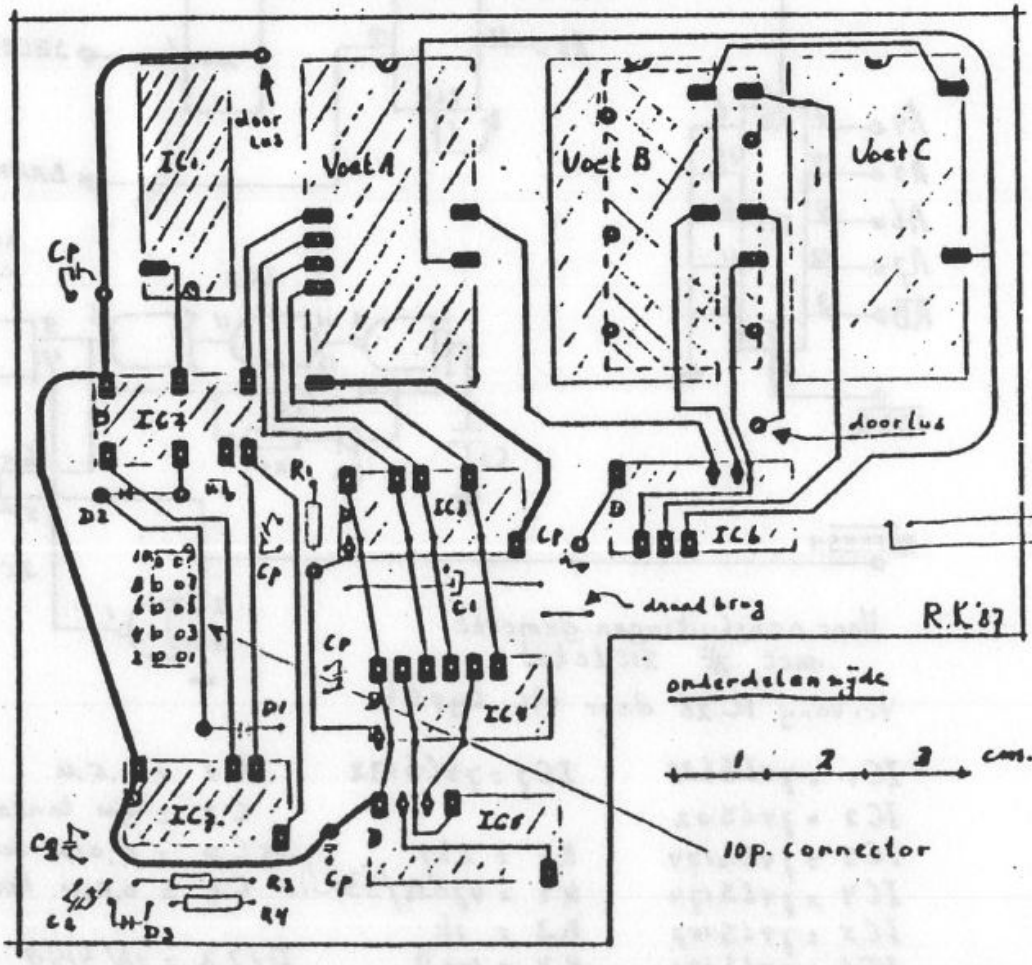
C1 = 0,5 µ
C2 = 1 µ tantaal
5x Cp = 0,01 µ Kemmier zie print
Cg = 0,6 µ Keramies " "

D1,2,3 = 1N 4148

R.K.'87







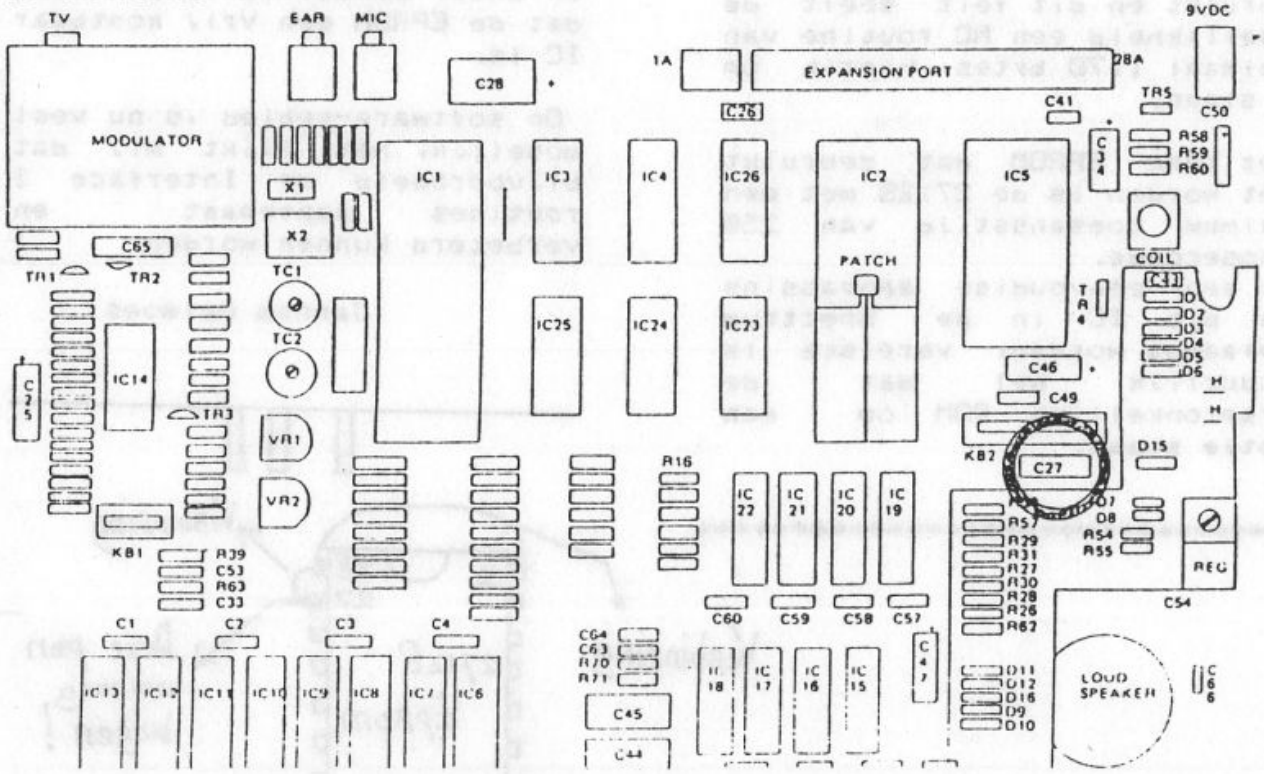
1 - A8
2 - SELEKT
3 - BANK
4 - IORQ
5 - n.c.
6 - RD
7 - WR
8 - REFRESH
9 - n.c.
10 - A7 bis

ZELFBOUW: RESET-SWITCH

door: Eddie Draaisma.

De reset switch ontbreekt op de Spectrum. Toch is deze heel eenvoudig te maken. Schroef de Spectrum maar eens open. Verwijder het toetsen bord van de rest door de strippen uit de connectors te trekken. Onder de rechter connector zit en blauwe ELCO C-27. Door deze condensator kort te sluiten wordt de Spectrum gereset. Het grootste probleem is dus eigenlijk een veilige plek te zoeken voor deze alle software verwoestende noodrem. Ik geloof dat een mooie oplossing is een 3mm gaatje te boren rechts boven op de Spectrum en aan de binnenkant, dus moeilijk bereikbaar een microprintschakelaar te monteren. Resetten is dan alleen mogelijk met een ballpoint o.i.d.

COMPONENT LAYOUT FOR ISSUE 2 BOARD



De Reset condensator

SPECTRUM-ROM VERVANGING

Een nadeel van veel gebruikte utility's is, dat deze na het inschakelen steeds geladen moeten worden en daarbij soms zeer nuttige RAM- gebieden in beslag nemen.
Een mogelijkheid om deze routines meteen aan 'boord' te hebben is de oorspsronkelijke ROM te vervangen door een EPROM. In deze EPROM komt het (eventueel aangepaste) Spectrum monitor programma te staan plus in het nog vrije ROM gebied de te gebruiken utility, bijvoorbeeld een printerroutine voor de Centronics interface. De adressen 386E t/m 3CFF Hex worden in de ROM namelijk niet gebruikt en dit feit geeft de mogelijkheid een MC routine van maximaal 1170 bytes hierin op te slaan.

Het type EPROM dat gebruikt moet worden is de 2712B met een maximum toegangstijd van 250 nanoseconde.
Met een eenvoudige aanpassing kan dit IC in de Spectrum geplaatst worden, vereiste is natuurlijk wel dat de oorspronkelijke ROM op een voetje staat.

****

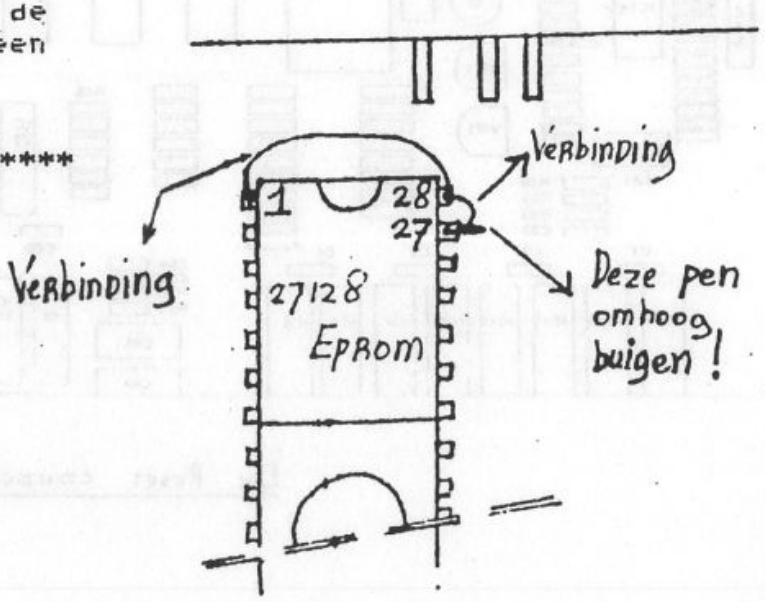
Bovendien moet deze ROM van het Hitachi of AMI type zijn, de NEC ROM maakt een kleine ingreep op de print noodzakelijk.

Pen 27 van de EPROM moet weggebogen worden, zodanig dat deze niet in het voetje komt. Verbindt nu pen 27 met pen 28; verbindt daarna pen 28 met pen 1.
Het beste is hier geen soldeerverbinding van te maken omdat bij het opnieuw programmeren van de EPROM ze weer ongedaan gemaakt moet worden.

Bijgaande tekening maakt een en ander duidelijk, bedenk wel dat de EPROM een vrij kostbaar IC is.

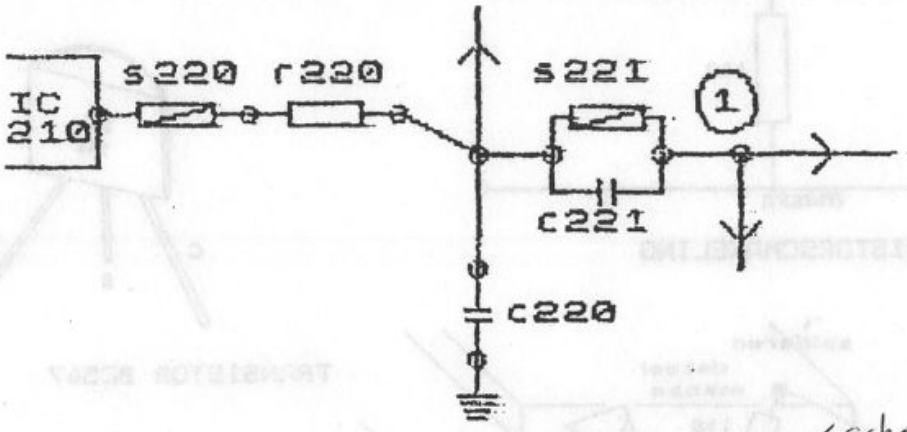
Op software-gebied is nu veel mogelijk, het lijkt mij dat bijvoorbeeld de Interface I routines aangepast en verbeterd kunnen worden.

Jannes Aalmoes



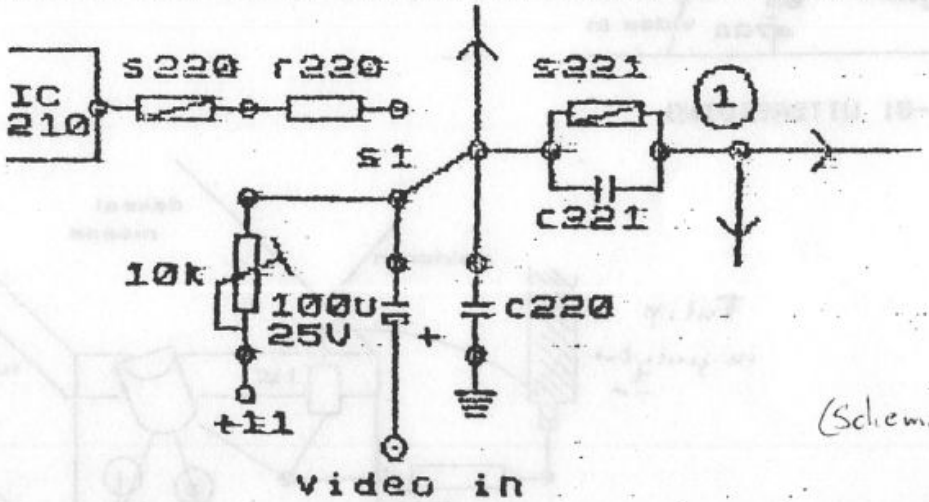
ZELFBOUW: TV ALS MONITOR + ==>

Het is vrij eenvoudig om een Philips televisie type TX met afstembalk om te bouwen tot monitor.
Het eerste schema laat de originele situatie zien en is te vinden in het bij de televisie geleverde schema.
Het tweede schema is voorzien van de uitbreiding.
Tevens op de volgende pagina in schema's de mogelijkheid om een ZX-81 en Spectrum van een monitoruitgang te voorzien.



(schema 1)

PHILIPS TV TX MET AFSTEMBALK ZONDER VIDEO AANSLUITING



(Schema 2)

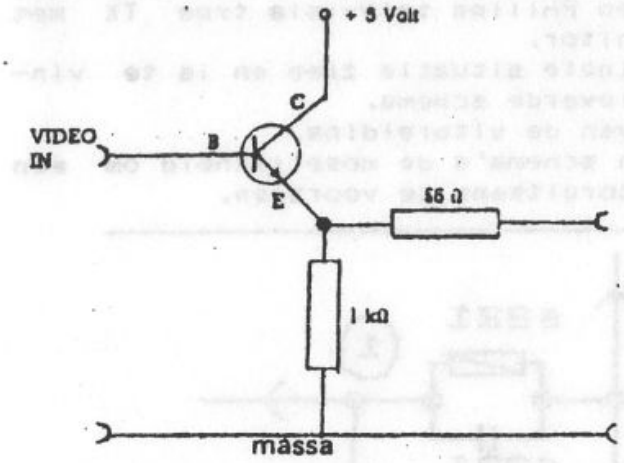
+11 ZIE SCHEMA (voeding)

PHILIPS TV TX MET AFSTEMBALK MET VIDEO AANSLUITING

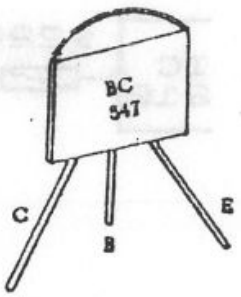
J.O.POSTEMA.

BULLETIN 10j

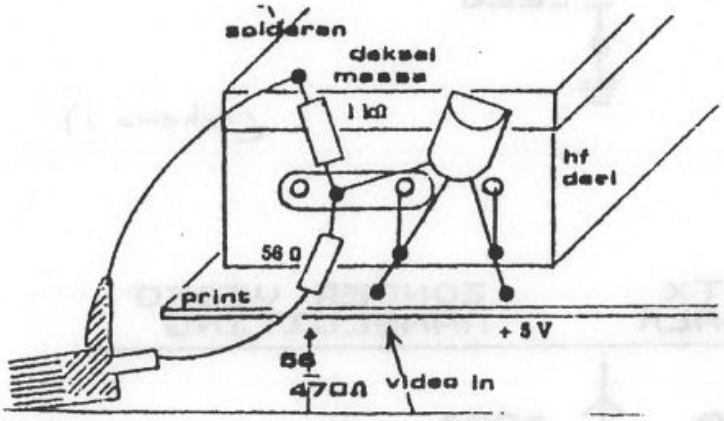
MONITOR UITGANGEN (SPEC. ZX81)



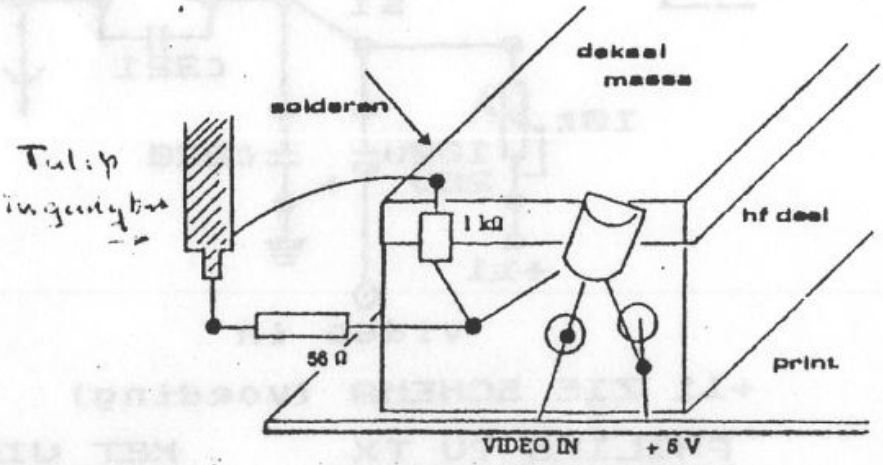
TRANSISTORSCHAKELING



TRANSISTOR BC547



Tulp ingangbus ZX-81 UITBREIDING



SPECTRUM UITBREIDING

RANDAPPARATUUR

door: C. van Krimpen.

INLEIDING:

Randapparaten (of periferie) onderscheiden zich van de eigenlijke computer door een aantal punten. Meestal is deze apparatuur door een kabel en toebehoren met de computer verbonden. Bij de Spectrum zijn zeker als randapparatuur aan te merken: Printers, lichtpennen, microdrives, diskdrives, floppydrives, joysticks, Eprom schrijf en lees apparatuur, Telexontvangers, modems, A/D converters, meetinstrumenten, weerstations, treinbesturingen, robotbesturingen en ga zo maar door.

Of de televisie, de cassetterecorder en het toetsenbord ook tot de randapparatuur behoren laat ik in het midden (huiswerk). in ieder geval is de voeding Geen randapparaat, maar gewoon de voeding. Keihard Ware.

Alle randapparatuur wordt aangesloten via tussenstations, ook wel interfaces genoemd. De interfaces zorgen voor "onderlinge aanpassing". De diverse verbindingen gebeuren met twee of meer draden. De gebruikte signalen worden meestal vertaald in een standaard.
De meest gebruikte is RS-232 (Serieel), ook erg veel gebruikt is Centronics (parallel). De bekendste Joysticks- standaard is die van Kempston.

Er zijn een aantal technische redenen, en misschien ook wel een aantal ethische en/of estetische redenen, om voor een bepaalde standaard te kiezen of om er juist van af te wijken. De hobby-gebruiker en de semi- professionele gebruiker zal bij de keuze letten op:

snelheid - afstand - al aanwezig materiaal.

Kortweg: RS-232 (serieel; informatie achter elkaar) voor grotere afstanden (voorbeelden: modem, fax), vereist meer elektronica. Centronics (paralllel; informatie over meer draden tegelijk) voor intensief gebruikte lijnen (voorbeeld: printer), waar eventueel timing een rol speelt.

Ik zal in deze serie de randapparatuur in drie groepen bespreken nl.:

1: RS-232 (algemeen).

2: Centronics (algemeen).

3: Speech sythesisers.

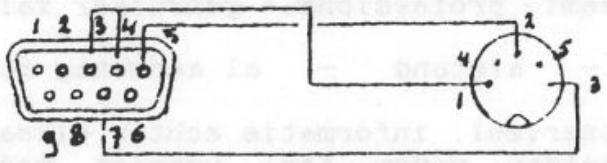
1: RS-232 :

De belangrijkste standaard in computerland vind ik de RS-232c. Zoals zoveel termen is "R S 2 3 2" OF "R S 2 32" afkomstig uit een groter verband. Dat verband ligt tussen twee standaarden: Een Amerikaanse EIA RS-232c (jaar 1969) en een Europese CCITT-V23/28. Het zijn populair gezegd, lijstjes met zorvuldig gedefinieerde afspraken (over een verbinding tussen een computer en een terminal), om toch eens iets dat niet direct familie is aan elkaar te kunnen knopen. Deze standaard kent om 1 reden (de beperkte snelheid: 19.200 bits/sec) een opvolger, de RS 423.

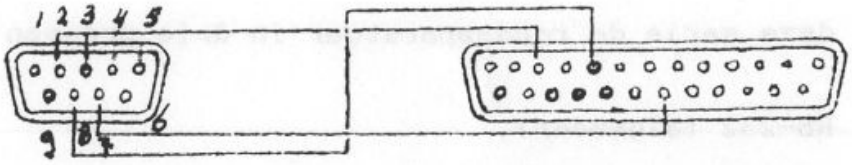
In de RS-232c is ondermeer vastgelegd dat de enen en de nullen stuk voor stuk (bit voor bit), dus serieel de signaalgeleider opgaan. Dat gebeurt met een bepaalde snelheid: de baudrate (=bits/sec; bps). Voor de baudrate kan men een keuze doen uit een reeks mogelijkheden:
75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 .

Helaas was men in 1969 nog niet zover om de afspraken te baseren op 1 connector. Zodoende is er altijd sprake van een DTE RS-232c connector en een DCE connector. De "T" is van terminal de "C" van communicatieapparaat (computer of modem). Voor velen is het de laatste jaren een probleem geworden, wat een communicatie apparaat is en wat een terminal. Eeen microcomputer kan bijvoorbeeld allebei zijn.....

Voor de DTE connector geldt dat een draad gemerkt met TD, SD, TX (pen 2) aan zenderzijde op + 3 Volt tot + 15 Volt ten opzichte van massa (pen 7) wordt gezet om een "LAAG" signaal, een "0", over te zenden. Voor een "HOOG" signaal, een "1", moet de spanning - 3 Volt tot - 15 Volt zijn. Aan ontvangstzijde heet die draad RD, RX. Is de connector een, zoals het oorspronkelijk hoorde, DCE connector, dan is dat pen 2. Op de DTE connector is het pen 3.



Voor Seikosha GP 250/550



Universele printerkabel (o.a. Brother CT - 50 BT)

(schema's overgenomen uit 't maandblad voor SINCLAIR gebruikers)

Conclusie 1: Voor een éénrichtig verbinding (besturing van een printer) zijn minimaal 2 draden nodig. Voor een zend en ontvang verbinding zijn minimaal 3 draden nodig.

Conclusie 2: Men kan een RS-232c verbinding gebruiken om TEGELIJKERTIJD over te zenden en te ontvangen. Wanneer het systeem van zender en ontvanger dit onafhankelijk kan doen heet dat FULL DUPLEX.

De standaard RS-232c makt gebruik van een bepaalde plug: de V24 met 25 pennen (of gaten). De standaard beschrijft behalve de drie genoemde pennen nog 11 andere pennen. Met name voor het gebruik van een modem zijn andere signalen van belang.

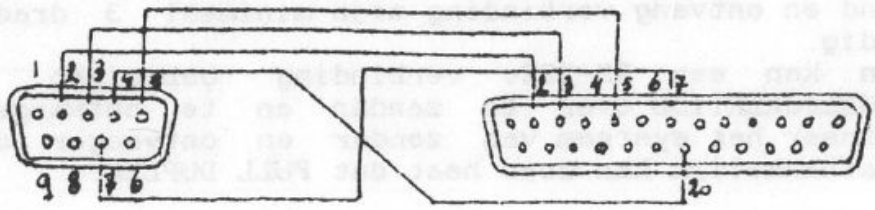
Het MODEM MOduleert het te verzenden signaal en DEModuleert het te ontvangen signaal. Moduleren wil zeggen, dat van een bepaalde spanning een bepaalde frequentie wordt gemaakt. Een "1" was dus - 12 Volt geworden, nu wordt het bijvoorbeeld 2400 Hz. Een "0" was + 12 Volt en wordt nu 1200 Hz. Dat zijn hoorbare tonen, die u per zender, per telefoon kunt overzenden, of op een cassette-bandje kunt opnemen.

De extra signaalladers van de RS-232c Maken het mogelijk om het modem (of printer, fax of cassetterecorder) voor te bereiden op zijn taak. Pen 6 (DSR= Data Set Ready) wordt hoog zodra het modem ingeschakeld is.
Pen 8 (DCD= Data Carrier Detect) wordt hoog zodra de verbinding gemaakt is, zodra er een drager (draaggolf) is.
Pen 5 (CTS= Clear To Send) wordt hoog even voordat de data overgezonden wordt.
Die signalen worden op de microcomputer ontvangen op respectievelijk pen 20 (DTR= Data Transmit Ready), pen 8 (DCD= Data Carrier Detect) en pen 9 (RTS= Request To Send).

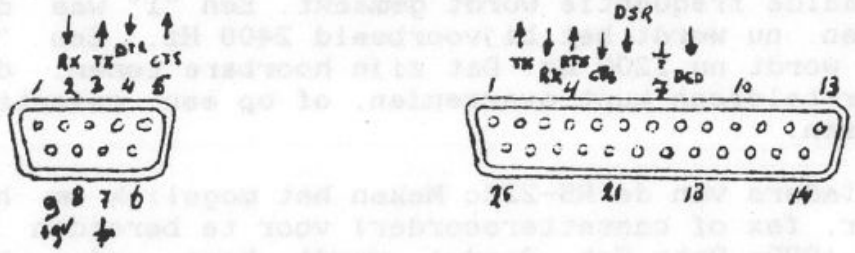
In onderstaand tabelletje is output = ! en input = ? :

	DTE				Interface 1 (DCE)
2	2	SD	TD, TX	(!)	RX (?)
3	3	RD	RD, RX	(?)	TX (!)
4	4	RS	RTS	(!)	DTR (?)
5	5	CS	CTS	(?)	CTS (!)
6	6	DR	DSR	(?)	--
7	7	SG	massa		massa
8	8	CD	DCD	(?)	--
9	9	--	--		+ 9 Volt
20	20	ER	DTR	(!)	--

Omdat het dus een ingewikkelde afspraak is (vooral door DTE en DCE) leken mij een aantal voorbeelden verhelderend te kunnen werken, twee voorbeelden (uit de "Sinclair Gebruiker" overgenomen) zag u reeds, hier volgen nog twee van eigen hand.



Volgens SINCLAIR (werkt o.a. bij Brother EP-44)



De pennen en hun funktie

Bij een RS-232c verbinding moet (toch) een aantal dingen worden vastgelegd op beide apparaten: de buadrate, code(s) voor het springen naar een volgende regel. Ook het aantal bits per karakter (PACKET) en de controle methode (PARITY) liggen meestal niet vast. Een goed voorbeeld is een Interface 1 - Brother EP-44 verbinding:

	bits	parity	CR+LF	code	error
"t" kanaal (voor text)	8	N	CR	7 of 8	Y of N
"b" kanaal (voor bits)	8	N	CR+LF	7 of 8	Y of N

De baudrate van de EP-44 is 1200 maximaal, van het Interface 1 maximaal 19200. De beste keuze (snelste) is dus 1200. Let daar vooral altijd op, maar ook op het aantal bits (gecombineerd met parity) en CR of CR+LF.

De code gebruikt bij RS-232c en bij bijna alle home-computers is vrijwel zonder uitzondering de zogenaamde ASCII code (7 bits). Daarmee zijn 96 (= 128 - 32) karakters gedefineerd, de rest is aangevuld. Zie daarvor de Appendix van uw handleiding (blz. 183 en verder).

Rest in dit verband nog een opmerking: ook synchrone communicatie per RS-232c is mogelijk (dus timing met speciaal kloksignaal), maar meestal is er sprake van asychronische seriele communicatie.

2: CENTRONICS INTERFACE:

In de inleiding en deel 1 werd al duidelijk (hoop ik) dat:
1. Er (te) veel standaarden zijn.
2. Dat het onderscheid tussenserieel en parallel interfaces het belangrijkste verschil is.
3. Het bekendste serieel interface is op basis van RS-232c.
4. Het bekendste parallel interface is volgens de standaard van Centronics.

Over de Centronics standaard is maar weinig te schrijven in vergelijking tot de RS-232c. Daarom hier kort over de standaard en daarnaast een gedeelte van een artikel van vorig jaar (febr. 1985 red.) over het "WILKENS"-interface.

Centronics is een firma op het gebied van computers. Waar veel bedrijven naar streven is hen gelukt: de standaard door deze firma gebruikt, werd door veel andere fabrikanten overgenomen. Deze standaard wordt voor zo ver ik weet alleen voor printers gebruikt, en dan voor het versturen van data (karakters) naar de printer. Eenrichting verkeer dus.

Bij deze standaard worden de acht bits van een byte tegelijk via acht parallelle lijnen naar de printer gestuurd. Dit in tegenstelling tot een RS-232c interface waar de bits één voor één worden overgestuurd. Het oversturen van de data (karakters) gaat volgens een "handshake" procedure. Bij deze procedure wordt voor ieder karakter dat geprint moet worden, via twee aparte lijnen (busy en strobe) een vraag en antwoord spelletje tussen de computer en de printer gespeeld. De printer moet immers weten wanneer de computer een karakter wil printen (met strobe) en de computer moet weten of de printer een karakter kan ontvangen (met busy).

* *

"WILKENS"-interface

* *

Van de gebruikte schakeling staat het schema in figuur 1. De scakeling is erg eenvoudig gehouden door een Z80A PIO te gebruiken. Dat is een lid van de familie Z80A waarvan de CPU het hoofd is. Dus:

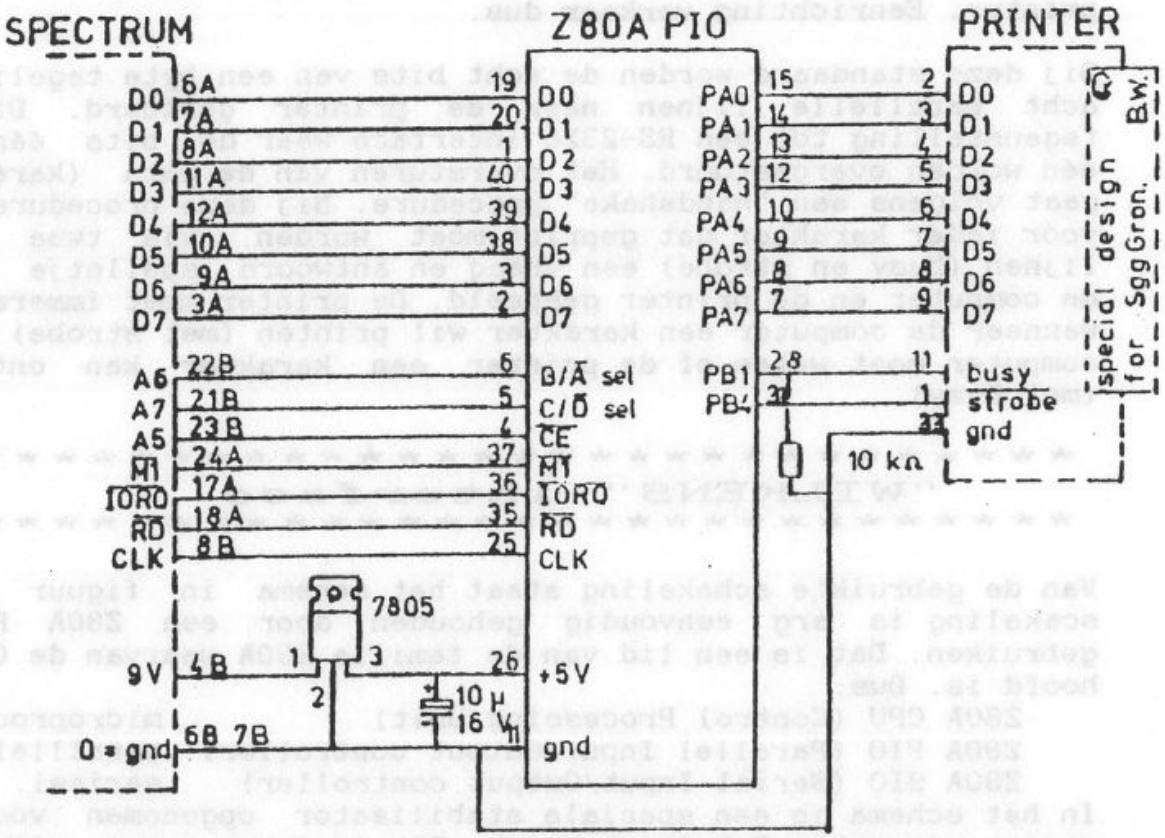
Z80A CPU (Control Processing Unit)	microprocessor.
Z80A PIO (Parallel Input/Output controller)	parallel.
Z80A SIO (Serial Input/Output controller)	serieel.

In het schema is een speciale stabilisator opgenomen voor het vormen van extra 5 Volt vermogen. Dit is zeker geen overbodige luxe, als u meer apparaten gebruikt.

De PIO heeft 2 achts bits poorten A en B. Deze poorten kunnen in groepjes als ingang en uitgang gebruikt worden. In ons geval wordt poort A gebruikt om de data van de computer naar de printer te sturen. Van poort B worden alleen de lijnen PB1 en PB4 gebruikt. Deze lijnen (busy en strobe) controleren de overdracht van de data tussen PIO en de printer. Lijn PB1 wordt

hierbij als ingang gebruikt en lijn PB4 als uitgang. De weerstand van 10 kio-ohm tussen PB1 en ground (gnd) zorgt ervoor dat de computer niet blijft wachten als de printer uit staat.
De verbinding van PIO naar Spectrum is heel eenvoudig omdat het familie van elkaar is. Gewoon doorknopen dus. Het meest bijzonder zijn de adreslijnen A5, A6 en A7. De adreslijn A5 wordt gebruikt om de PIO "wakker te maken" (CHIP ENABLE); hij moet gebruikt worden. Met A6 wordt aangegeven of de komende data voor poort A of B is (B/A). Met A7 wordt de PIO verteld of de data een commando voor de PIO is of data die doorgestuurd moet worden

Voor dit interface is behalve het schema ook software nodig. Deze software wordt (werd? red.) commercieel op de markt gebracht door de welbekende Groningse firma Filosoft.



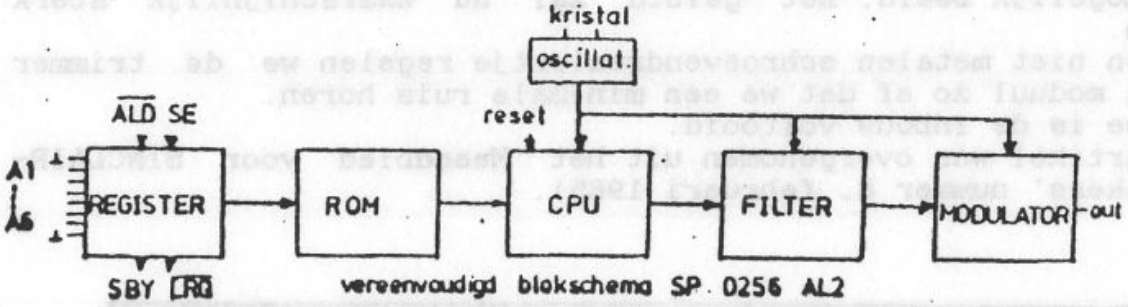
Printer interface (Centronics)

3. DE SPEECH-SYNTHESIZER

De speech-synthesiser bevat een chip (de SP 0256 AL2) waarmee engelse klanken geproduceerd kunnen worden. Zo kunnen ook woorden worden gevormd. De klanken kunnen gevarieerd worden i lengte. De mogelijkheden zijn enorm. Sommige Spectrum programma's maken daar (zonder dat u het weet ?) dan ook gebruik van. Nederlandse woorden zijn met de engelse klanken verstaanbaar, soms na lang experimenteren.

Het proces om klanken te produceren wordt voor het grootste deel afgewikkeld in een CHIP, de SP 0256 AL2. In feite heeft deze chip 6 digitale ingangen en 1 analoge uitgang. De ingangen kunnen alle 0 of 1 zijn, zodat er 2^6 = 2*2*2*2*2*2 = 64 mogelijkheden zijn aan de uitgang. Vijf daarvan zijn pauzes van verschillende lengtes.

De niet zo belangrijke ingangen zijn ALD (Adres LaDen) en SE (Strobe Enable). Is de SE nul, dan hoeft de ALD ingang niet te worden gebruikt, maar adrespulsen moeten gelijktijdig op de parallelle ingang worden aangeboden. Twee uitgangen zijn beschikbaar voor zorgvuldige besturing ("handshaking"), dat zijn LRQ (Load ReQuest=buffer vol) en SBY (StandBY).

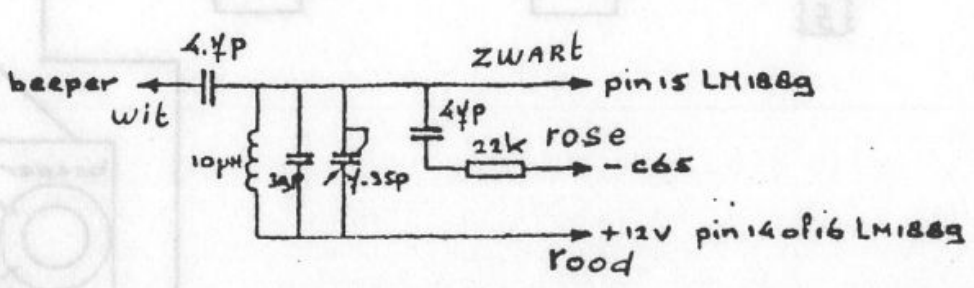


vereenvoudigd blokschema SP 0256 AL2

Een blokschema van de chip is getekend in bovenstaande figuur. Met een eenvoudige schakeling is de speech-chip op de Spectrum en ZX-81 aan te sluiten. Voor de Spetrum kunnen klanken geproduceerd worden door OUT 127, <getal> in te toetsen.

C. v. Krimpen.

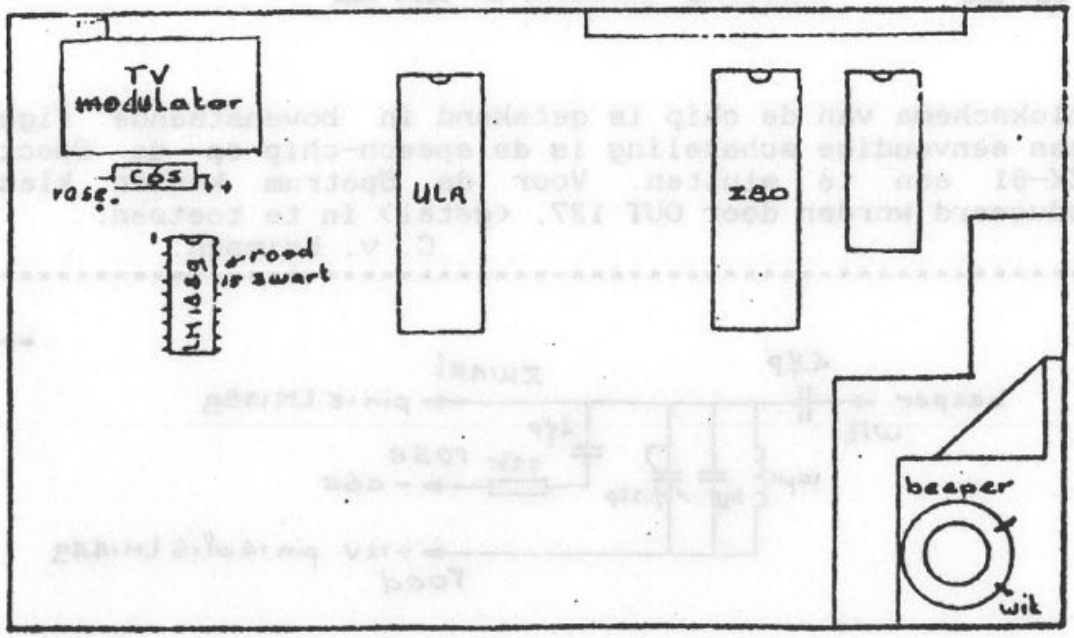
****



SPECTRUM GELUID VIA TV

door: H. Boon.

De Spectrum maakt gebruik van het IC LM1889 voor het generen van het video signaal, dit IC heeft echter ook de mogelijkheid om met behulp van enkele onderdelen een geluidsignaal te moduleren. Het is onbegrijpelijk dat Sinclair hier geen gebruik van heeft gemaakt.
Wij kunnen deze mogelijkheid eenvoudig alsnog aanbrengen. Na het openen van de Spectrum lokaliseren we eerst het IC LM1889. Hiervan moet pin 15 losgemaakt worden van de print, dit kan eenvoudig door het pootje door te knippen.
Aan het overgebleven stompje solderen we de zwarte draad van het moduul, de rode draad solderen we aan pin 14 of 16 van het IC.
De rose draad solderen we aan de minpool van condensator C65.
Tenslotte solderen we de witte draad aan de onderste aansluiting van de beeper.
Plats het moduul zo dicht mogelijk bij het IC LM1889 en houd de draden zo kort mogelijk, let op dat het moduul geisoleerd is van andere onderdelen. Gebruik hiervoor b.v. een stukje tape.
Schakel nu de Spectrum en de TV in en regel de TV af op een zo goed mogelijk beeld, Het geluid zal nu waarschijnlijk sterk ruisen.
Met een niet metalen schroevendraaiertje regelen we de trimmer op het moduul zo af dat we een minimale ruis horen.
Hiermee is de inbouw voltooid.
(Dit artikel was overgenomen uit het 'Maandblad voor SINCLAIR-gebruikers' nummer 8, februari 1985).



ZELFBOUW: RS 232 INTERFACE

door: H.J. Koevoets.

Voor hen die met de Spectrum aan telecommunicatie willen gaan doen heb ik een eenvoudig RS232 modem-interface ontworpen, dat ik samen met Jack Lelieveld heb uitgewerkt en getest.
Het interface is voor ongeveer F15.- aan onderdelen na te bouwen en iedereeen die wel eens een soldeerbout heeft vastgehouden zou het aan de praat moeten kunnen krijgen. Er hoeft bovendien niet perse een printje ontworpen te worden, een dubbelzijdige banen-print voldoet al. De schakeling bevat het minimum aan onderdelen dat nodig is om een modem op de Spectrum aan te sluiten.

Onder de kreet 'telecommunicatie kan de computerhobyist het werken met databanken verstaan, zoals VIDITEL, de zogeheten 'VIEWDATA SYSTEMS' of de 'BULLETIN BOARD SYSTEMS', zoals bij-voorbeeld het FIDONET van de HCC. Er bestaan tussen deze twee systemen een paar technische en een paar gebruikersverschillen.

De VIEWDATA systemen sturen steed volle pagina's met infor-matie: 24 regels van ieder 40 characters. U kunt dan de informatie lezen en de volgende pagina laten komen, of een keuze maken uit de geboden alternatieven.
Technisch worden de VIEWDATA systemen gekenmerkt, doordat zij gebruik maken van de ontvangst-/zendsnelheden 1200/75 baud. Dat wil zeggen ontvangen met 1200 baud, d.i. 1200 bits/sec, hetgeen bij 10 bits per character 120 tekens per sec. oplevert.
Die 10 bits zijn als volgt samengesteld: 1 startbit, 7 databits, 1 paritybit en 1 stopbit.
Het verzenden gebeurt met 75 baud. (7,5 char./sec.). Dat de ontvangstsnelhied groter is dan de zendsnelheid is bepalend voor het gebruik van het gehele VIEWDATA systeem.
De gebruiker zal steeds met slechts enkele toetsaanslagen zijn keuze duidelijk maken, waarna het VIEWDATA systeem antwoordt met volle pagina's. U ontvangt veel meer informatie dan u verstuurt. Als u als gebruiker van een VIEWDATA systeem zelf een volledige pagina wilt versturen, dan doet u daar 2 minuten en 8 seconden over, (24 * 40 = 960 characters op een vol scherm, 960 / 7,5 = 128 seconden, en 128 / 60 = 2 min. en 8 sec.).
Het VIEWDATA systeem stuurt u diezelfde pagina in 8 sec. terug.

Toch is de communicatie op 1200/75 baud best werkbaar gebleken omdat er, zoals gezegd, relatief weinig informatie wordt verzonden. Als antwoord of reactie op een ontvangen beeld tikt u immers slechts enkele toetsen in, die u de gewenste vervolg-beelden brengen.

Bulletinboards werken heel anders, zij sturen u steeds enkele regels tekst. Bijvoorbeeld:

BBS: Welkom bij dit Bulletin Board System.
U bent beller nummer 1517.
U hebt nog 19 minuten, wat is uw naam ?
U: Piet.
BBS: Goedenavond Piet, wat is uw woonplaats ?
U: Bergen op Zoom.
BBS: En met wat voor computer werkt u ?
U: Sinclair Spectrum.
BBS: Dank u Piet. Er zijn momenteel geen berichten speciaal aan u gericht, wel is er een algemeen bericht van de sysop aan alle gebruikers. Desgewenst kunt u .....

Als het beeld vol is zal de tekst een regel omhoog scrollen. De lengte van een regel varieert van BBS tot BBS. Er zijn er die van 40 tekens per regel uitgaan, maar er zijn er ook die met 64, 72 of 80 tekens per regel werken, terwijl de wat betere BBSen de mogelijkheid bieden, dat u zelf de te gebruiken regellengte instelt. Uit het voorbeeld was al gebleken dat de ontvangen en verzonden informatie elkaar qua hoeveelhied niet veel ontlopen. Vandaar dat de voor VIEWDATA systemen gebruikte 1200/75 baud hiervoor nier zo in trek is. Er wordt doorgaans gebruik gemaakt van gelijke zend- en ontvangstsnelheden, n.l. 300/300 baud.

Het hier bijgevoegde ontwerp van een RS232 modem-interface voor de Spectrum is tegebruiken in combinatie met elke RS232 modem. En er is keuze te over. Bij voorbeeld: WS2000, Teletron 1200, Protek, VICCOM, PHILIPS of het PTT-modem. De laatste is helaas, evenals het bekende VTX5000 modem niet omschakelbaar naar andere snelheden dan 1200/75 baud.

De software: het terminalprogramma, ("TERMINAL" geheten, is uitgebracht door de HCC-sgg en stond/staat? samen met andere bij dit interface te gebruiken programma's op een diskette. (Mocht het programma niet meer via de HCC-sgg verkrijgbaar zijn, neem dan even contact op met JOHAN Koning).
Dit programma kan goed gecombineerd worden met elk van deze modems en biedt de mogelijkheid om tal van dingen, zoals de baudrate, in te stellen via een menu. Iedere baudrate is mogelijk, maar bepalend voor de te kiezen baudrate is de computer, waarmee u in contact wenst te komen. Werkt die op 300 baud, dan zal u ten eerste uw modem op 300 baud moeten instellen en vervolgens uw programma. Dan kunt u bellen en contact maken.
Naast de baudrate kunt u nog een aantal dingen via het menu van het terminalprogramma instellen, zoals de parity. Parity is een techniek, die controle op een juiste ontvangst mogelijk maakt.

Ook heeft het programma een stand half-duplex/full-duplex. In de stand full-duplex zal de computer, waarmee u in verbinding staat, alle tekst die u intikt terugzenden, waarna het op uw scherm verschijnt. Bij half-duplex verbindingen doet de computer aan de andere kant van de telefoon dit niet, en u ziet dus niets meer op uw scherm verschijnen. In dat geval kunt u het terminal-programma in de half-duplex stand zetten, waarna deze de verzonden tekst zelf op het scherm plaats.

Om op zowel VIEWDATA systemen als op Bulletinboardsystemen probleemloos te kunnen werken, kan de functie van de ENTER-toets gewisseld worden. Of de ENTER-toets geeft een hekje (# voor VIEW DATA), of de ENTER-toets geeft een CR (voor BBS). Daarnaast kan de interpretatie van de CR gewisseld worden tussen 'wagen terug' (print positie aan het begin van de zelfde regel) en CR+LF (print positie aan het begin van de volgende regel).
Al deze mogelijkheden maken dat u met dit programma en een van de genoemde modems, via uw Spectrum en zelfbouw interface met zo goed als elk ander computer systeem in contact kunt komen.

Helaas is het niet mogelijk om met dit programma FIDO-files te down- of uploaden. Wel buffert het alles wat van FIDO afkomt en zodra FIDO met het zenden stopt, displayt het wat er ontvangen is. Daarom is er nog een tweede programma gekomem, "FIDOTERM" genaamd, dat zoals de naam al doet vermoeden, speciaal voor FIDO geschreven is.
Dit programma heeft in het kort de volgende kenmerken:

- 64 tekens op een regel.
- Instelbare baudrates: 1200/75, 75/1200, 300/300, 1200/1200.
- Instelbare ENTER-toets om een CR te versturen of een # .
- CR kan naar keuze ook gevolgd worden door een LINEFEED.
- Instelbare Parity (none, odd of even).
- Log-sessie. Hierbij is het mogelijk om de hele FIDO-sessie in het geheugen op te slaan. nadat de verbinding verbroken is kan dan alles op het scherm of op papier geprint worden. Deze buffer is 34000 bytes groot. Wanneer men echter gaat downloaden, dan wordt deze buffer leeggemaakt om plaats te kunnen bieden aan de te ontvangen file(s).

Een nadeel van dit programma is dat het up- en downloaden alleen via de logbuffer gaat, dus in CODE. Dit betekend dat we een programma eerst op moeten slaan. (B.v. op cassette met: SAVE "naam"CODE 23552,lengte).
Voorst kent het programma bij het XMODEM-protocol geen timeouts, wat bij communicatie tussen twee Spectrums een nadeel kan zijn.

Van dit programma zijn ook versies voor het PRISM VTX5000 modem, het Micrsource-interface, het ZX-Interface 1 en voor de ZX Spectrum 128K.

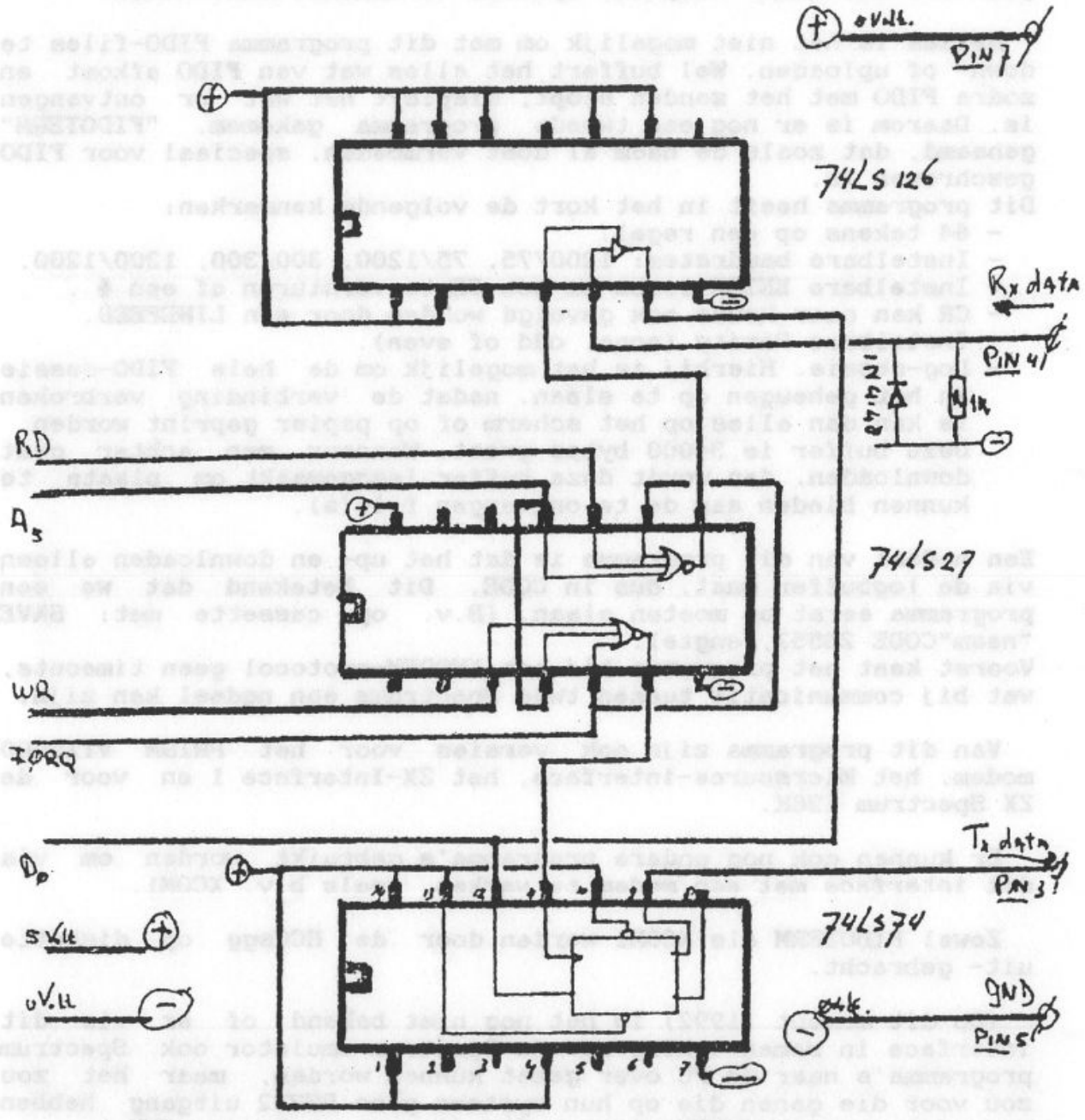
Er kunnen ook nog andere programma's gebruikt worden om via dit interface met een modem te werken, zoals b.v. XCOM1.

Zowel FIDOTERM als XCOM1 worden door de HCCsgg op diskette uit- gebracht.

(Op dit moment (1992) is het nog niet bekend of er via dit interface in samenwerking met de Spectrum emulator ook Spectrum programma's naar de PC over gezet kunnen worden, maar het zou zou voor die genen die op hun systeem geen RS232 uitgang hebben de moeite waard KUNNEN zijn om dit eens te proberen. JWK.).

Spectrum
IC adres: 223

modem
RS232



J.D. Lelieveld / H.J. Koevoets SEPT'85 INFO 010-846320

DATA VIA 'NORMALE' TELEFOON

door: J.O. Postema

Al enkele keren kwam tijdens de gebruikers bijeenkomsten (in 1984. red.) het onderwerp DATAcommunicatie via de telefoonlijn tersprake. Zou dit ook voor de Spectrum mogelijk zijn, en zo ja hoe?. En zou er ook op dit gebied, net zo als bij de meeste andere Spectrumzaken, een 'doe het zelf' oplossing mogelijk zijn?

Zoals uit het hier onderstaande zal blijken, is het zowel mogelijk als zelf te maken, en ook het hoe wordt duidelijk gemaakt.

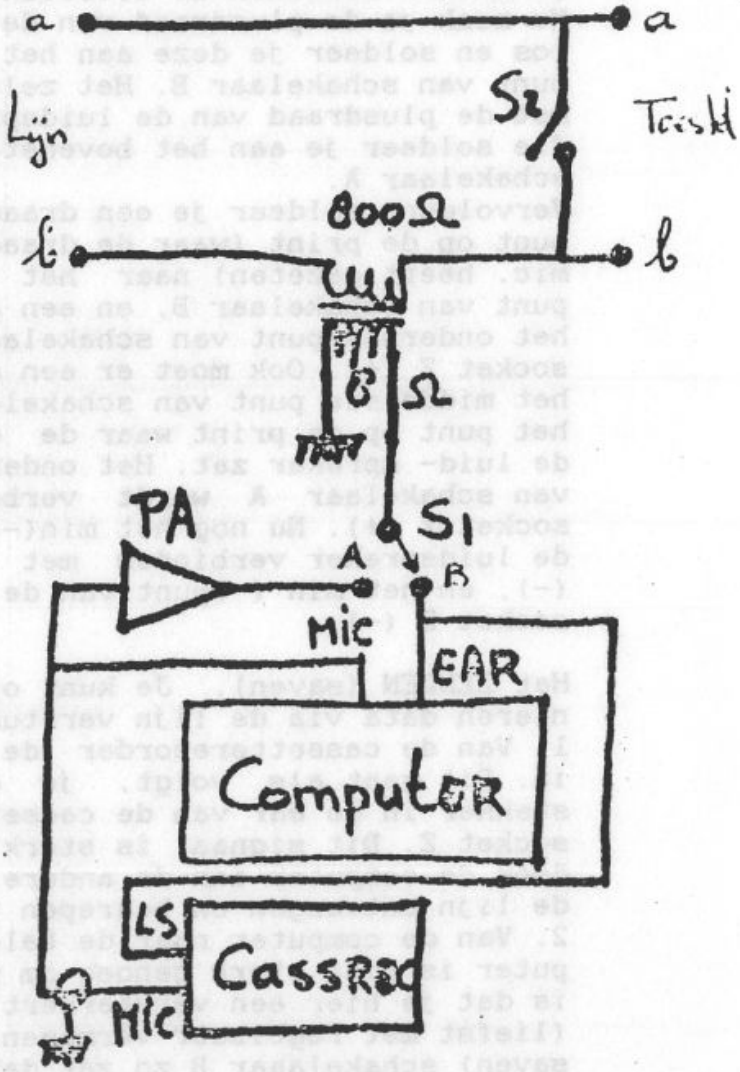
Het schakelingetje is te gebruiken voor bijna elk type toestel, en er hoeft niet aan de bestaande PTT bedrading gesleuteld te worden. Je kunt het in een kastje inbouwen, waarop je aan de ene kant een telefoonstekker aansluit en aan de andere kant een zgn telefoonwandcontactdoos, zodat je het geheel tussen het bestaande aansluitpunt en je telefoon kunt plaatsen.

Voor de trafo kan in principe elke laagspanningstrafo worden gebruikt

Tijdens het zenden 'S2' sluiten om het telefoonhoorn geluid te dempen. Deze schakelaar is niet perse noodzakelijk, alleen moet er bij het weglaten hiervan wel stilte worden betracht tijdens het zenden.

S1-A = vanuit computer verzenden
+ gelijktijdig zelf saven.

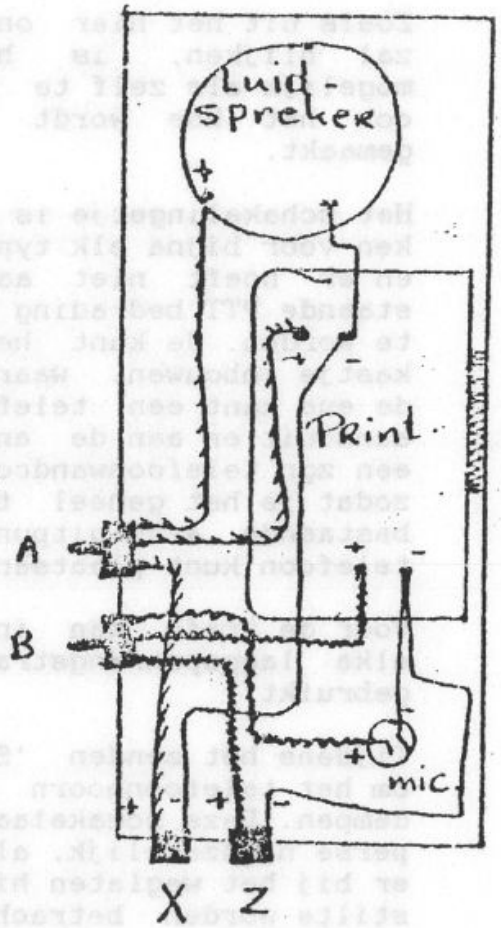
S1-B = vanaf cassette zelf laden
+ gelijktijdig verzenden.



VIA DRUKTOETSEN TOESTEL

door: H. Vesper

Hier nog een zelfbouw modempje, te maken van een goedkope druktoets telefoon (± f 15,00). Verder heb je alleen nog twee inbouwsockets nodig, (gelijk aan de mic. en ear sockets die achter op de computer zitten), en twee wissel (switch) schakelaartjes. Lees de onderstaande tekst en bekijk de tekening.
Bouw eerst de twee sockets aan de onderkant en de twee schakelaartjes aan de zijkant van de telefoon in, voordat je draden binnenin de telefoon losmaakt.
Nu maak je de plusdraad van de microfoon los en soldeer je deze aan het bovenste punt van schakelaar B. Het zelfde doe je met de plusdraad van de luidspreker, en die soldeer je aan het bovenste punt van schakelaar A.
Vervolgens soldeer je een draad van het punt op de print (waar de draad naar de mic. heeft gezeten) naar het middelste punt van schakelaar B, en een draad van het onderste punt van schakelaar B naar socket Z (+). Ook moet er een draad van het middelste punt van schakelaar A naar het punt op de print waar de draad van de luid- spreker zat. Het onderste punt van schakelaar A wordt verbonden met socket X (+). Nu nog het min(-)punt van de luidspreker verbinden met socket X (-), en het min (-)punt van de mic. met socket Z (-).



Het ZENDEN (saven). Je kunt op twee manieren data via de lijn versturen:
1. Van de cassetterecorder de telefoon in. Dit gaat als volgt, je drukt een stekker in de ear van de cassette en in socket Z. Dit signaal is sterk genoeg om door de computer aan de andere kant van de lijn ontvangen en begrepen te worden.
2. Van de computer naar de telefoonlijn. Het signaal van de computer is niet sterk genoeg om zo de lijn in te saven. Het beste is dat je hier een versterkertje van ca. 1 Watt tussen zet. (liefst met regelbaar vermogen.) Let wel op dat je (als je gaat saven) schakelaaar B zo zet dat tijdens het SAVEN de microfoon is uitgeschakeld, want anders gaat hij stuk.(Bij het saven kan je weer socket Z gebruiken, maar aangesloten op de mic-uitgang van de computer.
HET LOADEN. Als je het signaal uit de telefoon in de computer wilt loaden, verbind dan socket X met de ear-ingang van de computer.

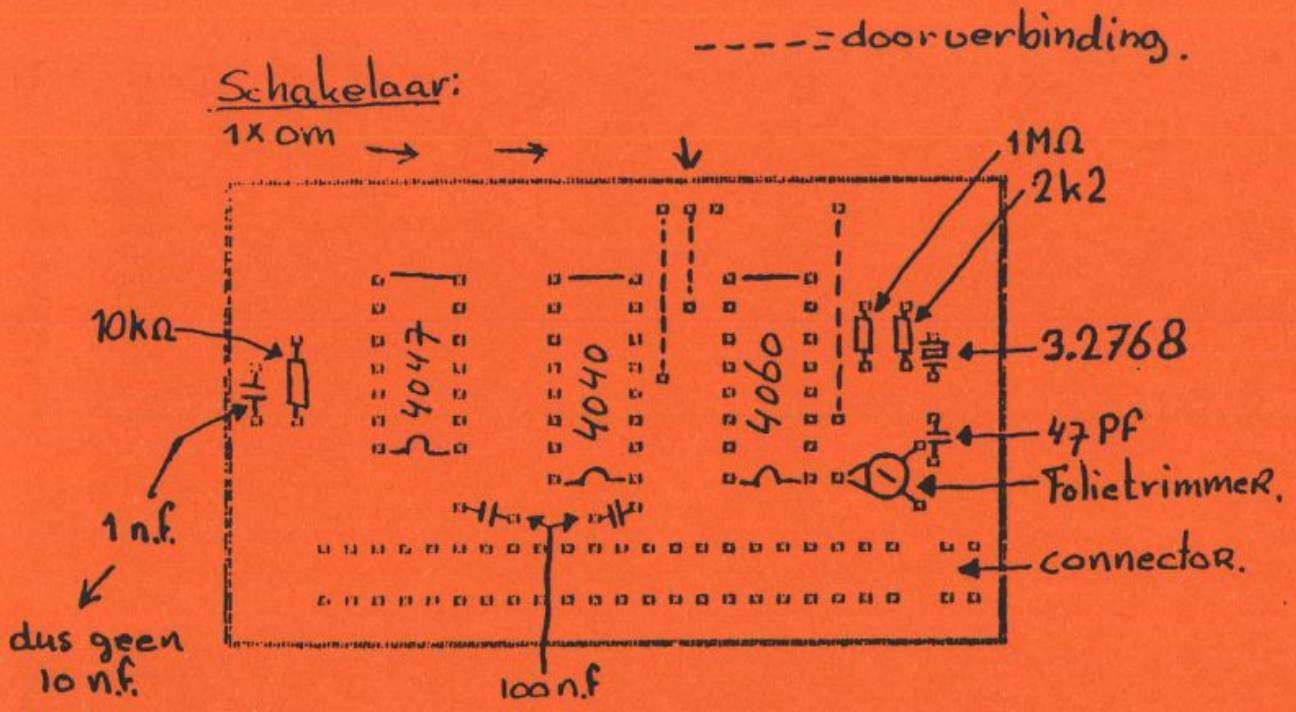
FAXINTERFACE VOOR 48K SPECTRUM

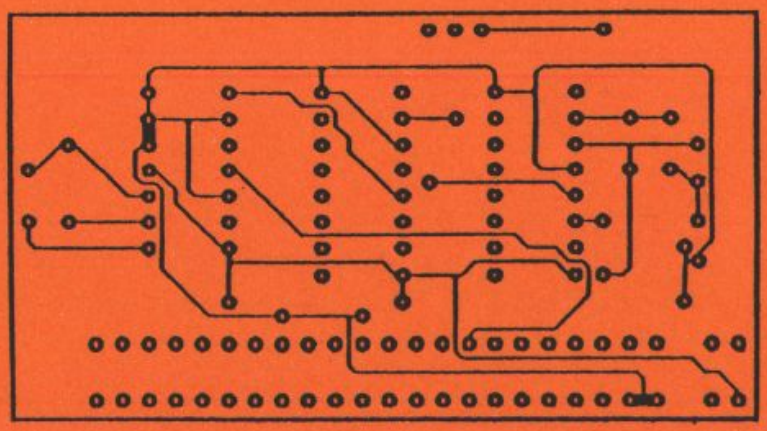
door: H. Vesper

Als je aan FAX ontvangst wilt gaan doen heb je daar een interface voor nodig, omdat de Spectrum daar te snel voor is. Aan de software alleen heb je niets. Zonder interface krijg je scheve beelden op het scherm. Hieronder staat de layout voor een zelfbouw print.

Onderdelen:

1 x 4060 I.C.	1 x weerstand 1MΩ	2 x condensator 100nf
1 x 4040 I.C.	1 x weerstand 2KΩ	1 x condensator 47nf
1 x 4047 I.C.	1 x weerstand 10KΩ	1 x condensator 1nf
1 x kristal 3.2768	1 x ZX-81 connector	1 x Folietrimmer(geel)





DRUKWERK

C. M. BALLINTYN
B. BOERMALAAN 7
9765 AP PATERSWOLDE

PORT BETAALD
GRONINGEN