

SINCLAIR

M P U L S

STICHTING
IMPULS

Nummer 7

Maart 1985

Stichting Impuls
POSTBUS 212
1740 AE SCHAGEN

Inhoud van IMPULS nummer Zeven.

07.1.9.001	07-04	Economen en ZX-81 (Mulder/Verduijn)
07.1.1.013	07-05	De zevende IMPULS (red)
07.1.3.028	07-06	Kleine instructie-grote gevolgen(vStaalduinen)
07.1.7.003	07-07	Belasting '84 (vStaalduinen)
07.2.3.018	07-10	EPROM's voor de Spectrum (Mosk/TIP)
07.2.3.019	07-11	Inverse video voor de ZX-81(Porceleijn)
07.2.3.021	07-13	SAVE-tip voor cassetterecorder(Ligthart)
07.2.3.023	07-15	(N)EPROM voor de ZX-81(Hoekman)
07.4.5.003	07-20	TASWORD en printer (v/d Maas)
07.4.5.004	07-21	Meer ervaringen met Tasword Twee (Bredenbeek)
07.4.5.006	07-23	Bijzondere symbolen en Tasword Twee(Breeden)
07.4.5.div	07---	Tasword tips (V/d Maas; Schim v/d Loeff)
07.5.1.007	07-29	Remstatement met veel bytes (vAbbe)
07.5.2.043	07-31	Lineaire zoekmethode (Breeden)
07.5.3.016	07-35	Vervolg Complete Basicode Disassembly (Radius)
07.5.3.019	07-38	Handleiding Basicode Schrijfroutine (Raats)
07.5.3.022	07-41	Basicode routine (Ament)
07.5.4.028	07-46	Een-en-twintigen (Sprengers)
07.6.4.001	07-48	Tiny Pascal Implementatie (Breedenbeek)
07.6.4.003	07-50	Geheugenindeling van Tiny Pascal(Breedenbeek)

ALGEMEEN

Inhoud van deze rubriek in dit nummer.

07.1.9.001	07-04	Economen en ZX-81 (Mulder/Verduijn)
07.1.1.013	07-05	De zevende IMPULS (red)
07.1.3.028	07-06	Kleine instructie-grote gevolgen(vStaalduinen)
07.1.7.003	07-07	Belasting '84(vStaalduinen)

Eerder gepubliceerd in de rubriek 1 ALGEMEEN

01.1.1.001	01-05	Introductie
02.1.1.005	02-05	Het tweede nummer
03.1.1.007	03-04	Computervrienden!
04.1.1.008	04-05	Klavertje vier (red)
04.1.1.009	04-51	Lidmaatschapskaart (red)
06.1.1.011	06-05	Stichting IMPULS(red)
01.1.3.001	01-07	Testen van cassettesignalen (Radius)
02.1.3.006	02-07	Het testen van cassettesignalen ZX-80 (Radius)
02.1.3.009	02-10	De Karaktergenerator van de ZX-81 (Breedem)
03.1.3.017	03-05	Verwarring over het gebruik van RAMTOP (v.Abbe)
03.1.3.019	03-07	Het gedrag van Stringvariabelen ZX-81 (Radius)
04.1.3.022	04-06	Vragen over SPECTRUM LOAD SAVE en BEEP (Radius)
04.1.3.023	04-07	Een duik op/in Uw Spectrum (vStaalduinen)
04.1.3.025	04-09	Kleurproblemen vanuit Uw SPECTRUM (Zimmerman)
04.1.3.026	04-10	SPECTRUM Mic, Ear en Speaker (Radius)
06.1.3.027	06-07	Intressante ROM-routines & instruc(Reddingius)
02.1.5.001	02-18	Meer geheugen voor de ZX-80 (Rietveld)
03.1.7.001	03-11	Boekbespreking "Voor galg en rad"
06.1.7.002	06-08	(Zak)Boekbespreking(Schim v/d Loeff)

ZX-81 ECONOMEN.

Tussen het geweld van de hardware-tips en machinecode-trucs vragen wij van een - waarschijnlijk klein - gezelschap even de aandacht voor onze ZX-81-passie. Wij zijn twee studenten die de economische wetenschap trachten te doorgronden en bezoeken daartoe regelmatig de Katholieke Hogeschool te Tilburg.

Veelal worden in studieboeken en vaktijdschriften analyses gepubliceerd die gefundeerd zijn op vrijwel oncontroleerbare berekeningen. Nogal sceptisch gooiden wij - de door de modale studiebeurs maximaal betaalbare - ZX-81 in de strijd en moeten nu concluderen dat de strijd niet bij voorbaat verloren is.

De twee programma's, die wij nu goed draaiende hebben, zijn: ZX-81-REGRESSION en ZX-81 MODELLENVEGER (beide in BASIC).

Het eerste programma stelt de gebruiker in staat meervoudige lineaire regressie-analyses te plegen. Het doel hiervan is voornamelijk het schatten van vergelijkingen op basis van reeksen statistische gegevens. Zelfs met de beperkte 16 K RAM vormen 5 verklarende variabelen en 40 data per variabele geen enkel probleem. Het programma genereert de structuur van de geschatte vergelijkingen, de correlatiecoëfficiënt tussen de verklarende variabelen, de coëfficiënten, de t-ratio's der coëfficiënten met de bijbehorende vrijheidsgraden, de F-ratio met de bijbehorende vrijheidsgraden, de R-kwadraat en de Durban-Watson-Statistic. Na een voltooide regressie biedt het programma de mogelijkheid één of meer verklarende variabelen weg te laten of te vertragen. Tamelijk gebruikersvriendelijk wordt het mogelijk, samenhangen op de statistische significantie te onderzoeken.

De modellenveger is een prima stuk gereedschap voor de oplossing van grote stelsels van lineaire vergelijkingen. De standaard 16 K RAM kan een stelsel van 50 lineaire vergelijkingen aan. Het programma is gebaseerd op het vegen van een matrix tot de rij-echelon-vorm is bereikt. Supersnel is de ZX-81 bij dergelijke klussen niet. Stelsels van bv. 35 vergelijkingen zetten hem 10 minuten aan het werk. Is men bereid dit beetje geduld op te brengen, dan verandert Uw ZX-81 al snel in een ware "spoorboekjes-fabriek". Iets wat wij lange tijd exclusief weggelegd achtten voor het Centraal Plan Bureau (CPB) en de rekencentra van Universiteiten.

Mocht men bezwaar hebben tegen de eis van de lineariteit van de modellen, dan moge het misschien troostend zijn te weten dat veel niet-lineaire modellen op niet al te ingewikkelde wijze te lineariseren zijn. Men denke bijv. aan modellen die luiden in procentuele verschillen.

Naast deze programma's, die het karakter van gereedschap dragen, mogen wij ook een data-base voor economische reeksen met PLOT-mogelijkheid tot onze productie rekenen. Bovendien denken wij aan educatieve programma's, die bepaalde theoriën interactief presenteren; zo werken wij aan een programma, dat de werking van een standaard IS-LM-model laat zien.

Onze bijdrage in dit blad heeft tot doel in contact te komen met ZX-81-gebruikers die hetzelfde interessegebied bestrijken om zodoende met een kleine beurs steeds grotere wetenschap te bedrijven.

Jan Verduijn en Rob Mulder
p/a Kalverstraat 14
5223 AD 's-Hertogenbosch

DE ZEVENDE IMPULS

Nummer zeven, het eerste nummer van 1985, ligt eindelijk, later dan verwacht, voor U. De planning 'in de loop van het eerste kwartaal' is dus niet gehaald. Uw redacteur biedt de donateurs van de Stichting IMPULS daarvoor zijn excuses aan. De Stichting zal ervoor zorgen dat de achterstand wordt ingelopen. U krijgt in 1985 dus in elk geval de vier nummers van IMPULS, als U donateur bent van de Stichting.

Wij, de schrijvers van de diverse artikelen en de redactie, hebben weer getracht een interessant nummer samen te stellen. Of die opzet geslaagd is zal elk van U misschien verschillend beantwoorden. U ziet dat vooral twee thema's ruime aandacht krijgen: het gebruik van TASWORD Two en de vele verborgen mogelijkheden die die tekstverwerker in zich bergt, en de gebruiksaanwijzingen van de beide meest bekende Hobbyscoop BASICODE vertaalprogramma's.

De Stichting beijvert zich om de beide routines in het IMPULSOFT-programma op te nemen. Daarvoor zal nog een barrière, die de NOS daartegen opwerpt, overwonnen moeten worden. Mogelijk is dat gelukt tegen de tijd dat U dit onder ogen krijgt.

Behalve de twee thema's biedt dit nummer echter weer 'voor elck wat wils'. Wij zouden echter graag zien dat U een onderwerp opgeeft dat nog onvoldoende in IMPULS aan de orde is gekomen. Vooral indien die opgave vergezeld gaat van Uw artikel (met Tasword ingetypt aan postbus 212, 1740 AE Schagen, s.v.p.) worden Uw mede-Sinclair-gebruikers daarmee gediend. En denkt U vooral niet te gauw dat Uw Tasword-vrucht niet interessant is: Deze IMPULS wordt door een breed scala computeraars gelezen en herlezen. Dit is geen 'tijd'schrift, in die zin dat de informatie, die er in staat van tijdelijke interesse is. U zult zien - wanneer U weer eens een oud nummer ter hand neemt, dat voorheen onbegrepen artikelen nu ineens Uw volle aandacht krijgen. Eerder een SINCLAIR-encyclopedie dan een nieuwsbode.

U weet wellicht, maar voor de nieuwelingen herhaal ik het nog maar: donateur wordt U alleen maar op één manier, te weten door f 25.- per jaar te storten op postgirorekening 5.693.776 of op bankrekening n° 45.40.87.446, beide ten name van de Stichting IMPULS te Den Haag. Deze nummers alléén gebruiken voor donaties! Uw bestellingen s.v.p. betalen op de rekening, die bij de aanbiedingen vermeld staan.

Ten slotte een afscheid. Na zes IMPULSen te hebben geproduceerd vind ik de tijd gekomen om mij weer eens aan actief werken met de computer (dus niet alleen met behulp van de computer mijn werk doen) bezig te houden en met vele andere zaken, die mijn aandacht hebben. Wie mij daarbij verwijt dat ik IMPULS in de steek laat, moet ik tegenwerpen dat wij twee zéér bekwame personen bereid gevonden hebben om die taak van mij over te nemen. Dat zijn Ed Weijgers uit Delft en Piet Zwager uit Den Haag. Zij zullen ongetwijfeld enkele veranderingen doorvoeren, maar niet ten koste van de tot nu toe bereikte IMPULS-kwaliteit.

Met dank voor Uw medewerking en de beste wensen voor Ed Weijgers en Piet Zwager laat ik U nu gauw aan de inhoud van IMPULS over.

Pieter Schim van der Loeff

KLEINE INSTRUCTIE - GROTE GEVOLGEN.

Bij de derde aflevering van een bepaald stukje vind ik dat je zo langzamerhand mag gaan spreken van een vaste rubriek. Dus hierbij openen wij dan de vaste rubriek met hints en tips voor de ZX-Spectrum. Echter ... om zo'n rubriek te vullen heb je het een en ander aan gegevens nodig. Ik ben ervan overtuigd dat er een groot aantal simpele (en ingewikkelde) instructies zijn om in zo'n rubriek te blijven vullen. Het probleem is alléén, dat ze verspreid zitten bij zo'n 2000 Spectrum-bezitters binnen onze gebruikersgroep. Daarom hier nogmaals een oproep om kleine kleine instructies (een regel) op te sturen, zodat die 1999 andere Spectrum-bezitters kunnen meeprofitieren van Uw zelfbedachte of zomaar ergens gevonden aanvulling in Sinclair-Basic-wonderland. Bedankt alvast! Om de smaak te pakken te krijgen volgen hier weer een paar van die "kleintjes". De vorige aflevering verschenen in IMPULS n° 4, blz. 7/8 en IMPULS n° 6, blz. 7.

PRINT-INSTRUCTIES.

→ Open 2,"P" - na deze instructie verschijnt de informatie, die normaal boven in het scherm zou komen, op de printer;

Open 1,"P" - als Open 2, maar dan met de informatie, die normaal onder in het scherm zou komen;

Open 3,"S" - alle informatie die met LPRINT of LLIST normaal op de printer zou komen, komt nu op het scherm.

Als U bovenstaande instructies ergens wilt toepassen dan verdient het aanbeveling om vóór de OPEN-instructie eerst een CLOSE (1, 2 of 3)-instructie te geven, al naar gelang de STREAM die U wilt gaan toepassen.

COPY-vervanging - Indien U in het bezit bent van een "betere" printer, kunt U het COPY-commando niet meer gebruiken. Als U alleen tekst van het scherm wilt kopiëren, dan is daarvoor een korte vervangende instructie die alléén letters en leestekens (dus met een ASCII-waarde tussen 32 en 127) omgezet naar de printer. Waar in 'n programma COPY staat kunt U dat vervangen door regel of subroutine:
 nnnn FOR R=0 TO 21: FOR K=0 TO 31: LPRINT SCREEN\$(R,K);: NEXT K:
 LPRINT: NEXT R (N/L) waarin "SCREEN\$" een code-woord is
 en U de letters R en K naar believen kunt aanpassen.

SCREEN-HANDLING.

RANDOMISE USR 3280: de hele pagina SCROLLt een regel mohoog;
 RANDOMISE USR 3582: SCROLLt de hele pagina 1 regel omhoog en voegt de vrijkomende toe aan het onderste deel van het scherm (probert U dit uit met een contrasterende border-kleur);

RANDOMISE USR 3583: als bij 3582, maar dan met een ½ blz ineens;

RANDOMISE USR 3652: als bij 3582, maar dan zonder SCROLL;
 INPUT ; wist de onderste twee regels (zoals bij Randomise USR 3483);

→ RANDOMISE USR 4317: verandert hoofdletters in kleine letters, en andersom.

Rob van Staalduinen,
 Den Haag

B E L A S T I N G 1 9 8 4

Een paar weken geleden op de "Haagsche Afdeling" van de Sinclair Gebruikersgroep liet een van de aanwezigen mij een belastingprogramma zien. Het was van een uitgeverij, waarvan de naam begint met een K en die onder deze naam een belastinggids uitgeeft. Omdat ik weet dat veel mensen toch wat moeite hebben met het invullen van hun belasting-aangifte, heb ik met dat uitgangspunt naar dat programma gekeken. Dat liep uit op een grote teleurstelling. Het programma veronderstelt het gebruik van de Gids om direct e.e.a. te kunnen naslaan. Maar zelfs dan is het zeer de vraag wat voor nut het programma heeft.

Heel anders verging het mij met het programma "Belasting 1984" van Filosoft, geschreven door de heer Diepmaat. Gezien mijn vorige ervaring moet ik zeggen dat ik van belastingprogramma's niet z'n hoge pet meer op had, maar nu weet ik dat er ook een echt goede is.

Na een vrij lange laadperiode veranderde de gebruikelijke 32 tekens per regel in 51 tekens. In goed leesbare tekst komen boven in beeld de omschrijvingen van het belastingformulier. Onderin de vragen, die nodig zijn voor de diverse berekeningen.

Stap voor stap, in de vorm van duidelijke vragen, komt dan op het scherm het antwoord op de betreffende vragen van het belastingformulier tot stand. Zeer snel voegt de computer er de antwoorden van berekeningen aan toe. Zijn alle vragen beantwoord dan verschijnt "ENTER" in beeld. Op het scherm staat dan - exact zó ingedeeld als op het belastingformulier - het antwoord op de betreffende vraag.

Hier komt het enige minpuntje in het programma naar voren. Zonder in het programma in te breken was ik niet in staat om van ieder scherm (vraag) een COPY op de printer te maken. Ik wilde eerst alle gegevens in de computer invoeren voordat ik het belastingformulier kon gaan invullen. Ik moest echter per vraag de gegevens naar mijn belastingformulier overbrengen.

Maar nu weer verder met het programma. Nadat U - steeds ondersteund door duidelijke vragen en aanwijzingen - alle vragen van

het belastingformulier hebt doorgeworsteld geeft het programma een resumé dat identiek is aan dat van het belastingbiljet. U kunt dat zo overnemen. Maar daarmee zijn we er nog niet!

Het programma vervolgt nu met vragen ter bepaling van Uw tariefgroep. Nadat U de antwoorden hebt gegeven op de aanvullende ja/nee-vragen geeft het ook nog even de berekening over de terug te ontvangen of alsnog te betalen belasting en premieheffing over het jaar 1984.

Dan stopt het programma en zit U met een ingevuld belastingbiljet 1984. Daar hoeft U alléén nog maar de betreffende stukken bij te zoeken.

Omdat mijn specialiteit toch een beetje ligt in de administratieve BASIC denk ik dat ik kan stellen dat vader & zoon Diepmaat met dit belastingprogramma een hoostandje hebben geleverd. Zij hebben tevens aangetoond dat onze Spectrum kan werken als een hoogwaardige computer. Alle lof aan de auteurs van dit programma. Daarom heb ik een advies aan allen, die hun aangifte nog moeten doen: Het programma kost minder dan een eerste consult met een belastingadviseur. Bovendien kunt U deze jaarlijks terugkerende klus thuis op Uw gemak klaren, dus deze aankoop zal zeker de moeite waard blijken.

Als voor de volgende keer het programma nog kan worden uitgebreid met een mogelijkheid voor copiën op papier of liever nog een printout direct op het belastingbiljet die direct opgezonden kan worden dan zou dat geweldig zijn.

Het programma is uitgebracht door: Filosoft, Postbus 1353, 9701 BJ Groningen en het kost f 49,50. Voor telefonisch advies en/of nadere informatie kunt U (iedere vrijdag tussen 10 en 16 uur) bellen (050)137746. Dit programma dat zich mogelijkerwijze dubbel en dik terugverdient wordt U van harte aanbevolen door:

Rob van Staalduinen
Den Haag

HARDWARE

Inhoud van deze rubriek in dit nummer.

07.2.3.018	07-10	EPROM's voor de Spectrum (Mosk/TIP)
07.2.3.019	07-11	Inverse video voor de ZX-81(Porceleijn)
07.2.3.021	07-13	SAVE-tip voor cassetterecorder(Ligthart)
07.2.3.023	07-15	(N)EPROM voor de ZX-81(Hoekman)

Eerder gepubliceerd in de rubriek 2 HARDWARE

01.2.1.001	01-15	Universale experimenteerprint
01.2.1.002	01-16	Schema ZX-80
01.2.1.004	01-18	Schema ZX-81
03.2.1.006	03-14	Schema Spectrum
01.2.3.001	01-20	Repeat Key (v Staalduinen)
03.2.3.003	03-16	Zelfbouw geheugenuitbreiding (RAM/ROM)
04.2.3.006	04-12	Herplaatsing: Repeat Key (vStaalduinen)
04.2.3.007	04-13	17 K RAM met 16 K RAM-pack (vDorp)
04.2.3.009	04-15	Swing de pan uit (vStaalduinen)
04.2.3.011	04-17	Joystick (Brons)
04.2.3.014	04-20	Tips voor gebruikers van grote toetsenborden
06.2.3.015	06-10	Geheugenuitbreiding (Keppel)
06.2.3.016	06-11	Trickstick (Achterkamp)
04.2.9.001	04-19	Massagheugen met spoelenrecorder (vStaalduinen)

ZX-SPECTRUM EPROMS

De meeste serieuze Spectrum gebruikers draaien maar een paar programma's op hun machine (bv. een wordprocessor of een spreadsheet). Die programma's laden ze dan steeds van cassette of Microdrive, wat grote verliezen van tijd en geheugen betekent. In z'n geval biedt een EPROM, die op adressen 0 tot 16384 geschakeld wordt, en het normale ROM uitschakelt uitkomst: er blijft ongeveer 41 Kb RAM vrij en men hoeft alleen de files te laden, die men wil gebruiken.

Dergelijke EPROMS worden nog niet gemaakt, maar qua hardware zouden wij dat zeker kunnen: een EPROM-kaart met daarop twee 8kB EPROMs, die aan de edge-connector geplugd kan worden, is vrij eenvoudig. De software zal ons ook geen grote problemen opleveren, maar het communiceren met andere randapparatuur dan ZX-Printer en cassetterecorder is nog een probleem voor ons. Wij hopen, dat u ons kunt helpen met schema's en listings van control-routines van printer-interfaces en Microdrives. Als u een schema of listing heeft, waar wij iets aan kunnen hebben, belt U ons dan a.u.b.

De programma's die we in een EPROM willen zetten zijn: een wordprocessor, een spreadsheet, een assembler en een Microsoft-achtige Basic. Bij onvoldoende belangstelling worden de programma's duurder of gaat het niet door. Als wij een programma af hebben, laten wij dat zien op de Sinclair gebruikersdagen, in de Bron of in dit blad. De prijs van de EPROM-kaarten komt (hopelijk) onder de f 100,- te liggen. De hardware alléén kost al zo'n f 85,-. Dat lijkt ons nogal overkomelijk.

Hopelijk tot ziens op de eerstvolgende Spectrum-dag.

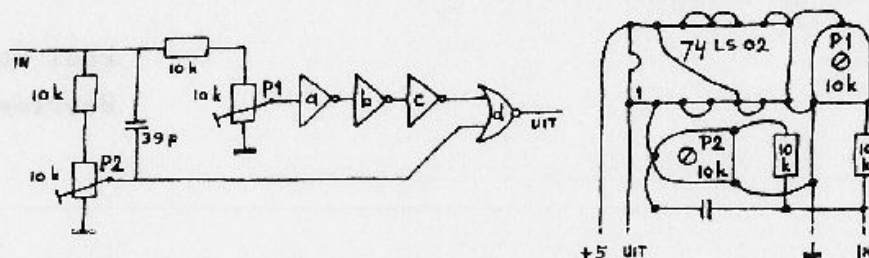
Allard Mosk	Andries Tip
Castricum	Castricum
02518-54081	02518-57057

INVERSE VIDEO MET DE ZX-81

De ZX-81 geeft van huis uit een zwarte tekst op een witte achtergrond. Mooi is dat niet. Als jouw TV niet zo best meer is, worden allerlei ongerechtigheden zichtbaar op de achtergrond. Bij de meeste computers is het beeld dan ook "invers": alleen de tekst licht op, de rest is donker.

"Omkeren" van het beeld kan niet zómaar. Je moet namelijk zorgen dat het synchronisatiesignaal niet mee wordt omgekeerd, want anders snapt de TV er niets meer van. Jammer genoeg is bij de ZX-81 het synchronisatiesignaal niet apart ergens van de print op te pikken.

Toch lukte het mij na enig experimenteren om het ZX-81 signaal om te keren. De hulpschakeling die ik daartoe bedacht heb is heel simpel. Voor ZX-81 fans die niet bang zijn voor wat ambachtelijke hardware is het nabouwen een fluitje van een knaak. Want zoveel was ik aan materiaalkosten kwijt. Een simpel TTL-tje met wat weerstanden. Links het schema, rechts hoe de zaak op een stukje gaatjesprint wordt gemonteerd. Gebruik géén IC-voetjes want dan wordt de zaak te dik voor montage in de ZX-81 kast.



Om de schakeling aan te sluiten verbind je de draden + en 1 met de voedingslijnen op de ZX-81 print (bijvoorbeeld bij de spanningregelaar). Knip vervolgens de draad middendoor die het videosignaal van de computerprint naar de modulator (het blikken doosje) voert. Die draad zit tussen het doosje en de EAR-plug. Soldeer "IN" op de draad die in de print verdwijnt en "UIT" aan (vervolg op pag. 12)

(vervolg van pag. 11)

het draadje dat in het doosje verdwijnt. Draai beide potmeters helemaal linksom (naar 1). Schakel nu de computer in en draai P1 langzaam open tót het lege beeld op de TV stilstaat. Draai dan P2 open tot je een mooi inverse tekstbeeld hebt. Klaar!

De werking is als volgt: met P1 wordt het videosignaal zó ingesteld dat de inverter alleen reageert op de sterkste positieve pulsen. Dat zijn die van het sync-signaal. Inverter a werkt daarvoor als synchronisatie-scheider. Met de inverters b en c wordt het sync-signaal vervolgens opgepoetst, op TTL-niveau gebracht en omgekeerd. Het netwerk rond P2 vormt een verzwakker. Daarmee wordt het videosignaal zó ingesteld, dat poort d gaat werken als een inverterende analoge versterker. Dat kan alleen als beide ingangen van deze NOR-poort hoog zijn. Het geïnverteerde sync-signaal op de andere poortingang zorgt nu dat die versterker bij elke (negatieve) sync-puls wordt uitgeschakeld. Aan de uitgang verschijnt aldus een invers videosignaal met flinke positieve sync-pulsen: precies wat wij nodig hebben.

De poorten a en d van de 74LS02 worden hier toegepast als analoge versterker. Dat is voor dit IC-type ongebruikelijk en het resultaat varieert dan ook enigszins per fabrikant. Ik gebruikte de uitvoering van Signetics. Die van Texas Instr. blijkt in elk geval ook te voldoen.

Paul Porcelijn,
Haarlem.

SPECTRUM PROGRAMMA "SAVEN" MET CASSETTERECORDER

Het "SAVEN" van een Spectrum programma met een cassetterecorder gaat in de regel met lastige handelingen gepaard. Zodra de opnametoets wordt ingedrukt, begint de zaak te fluiten. Dit kan worden opgeheven door de "EAR" plug uit de entree te trekken die er echter tijdens "VERIFY" weer in moet.

E.e.a. vindt z'n oorzaak in het feit, dat in- en uitgang voor de recorder op dezelfde pin van de ULA zitten, zodat ze met elkaar zijn doorverbonden (zie schema). Het gevolg daarvan is, dat het versterkercircuit gaat "rondzingen".

Met de pluggen op hun plaats kan het geheel feilloos werken, als U in de recorder een kleine wijziging aanbrengt. Deze wijziging bestaat uit het opnemen van een kontakt in de EAR-aan-sluiting, zodanig, dat dit kontakt verbreekt als de opnametoets wordt ingedrukt (zie recorderschema van de Philips recorder type D6620/P).

Daartoe wordt kontakt A7-9 gebruikt, dat in de originele staat de luidspreker uitschakelt. Met de aanwezige EAR-plug wordt de luidspreker gevoed over een trimpotmeter van 220 ohm, zodat altijd meegeluisterd kan worden op het gewenste volume.

Het ware te wensen, dat Philips (en andere fabrikanten) deze schakeling standaard in de data-cassetterecorder opnemen, daar het voor andere computers niet in de weg zit. Dat heb ik derhalve aan Philips Eindhoven, Afd. Consumentenbelangen, geschreven.

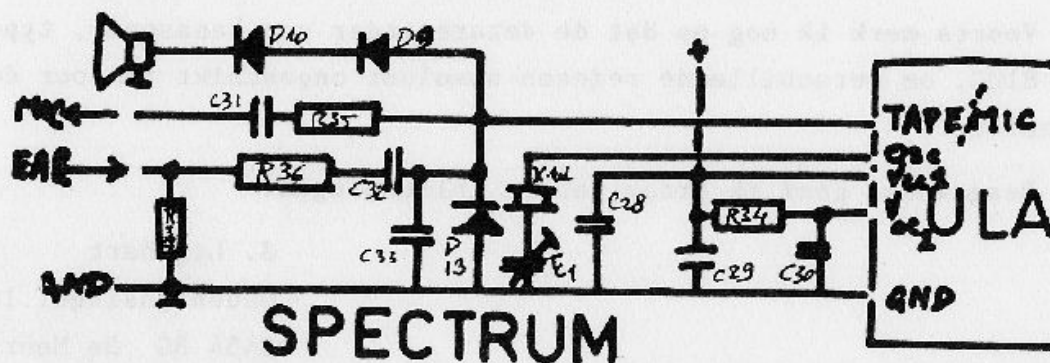
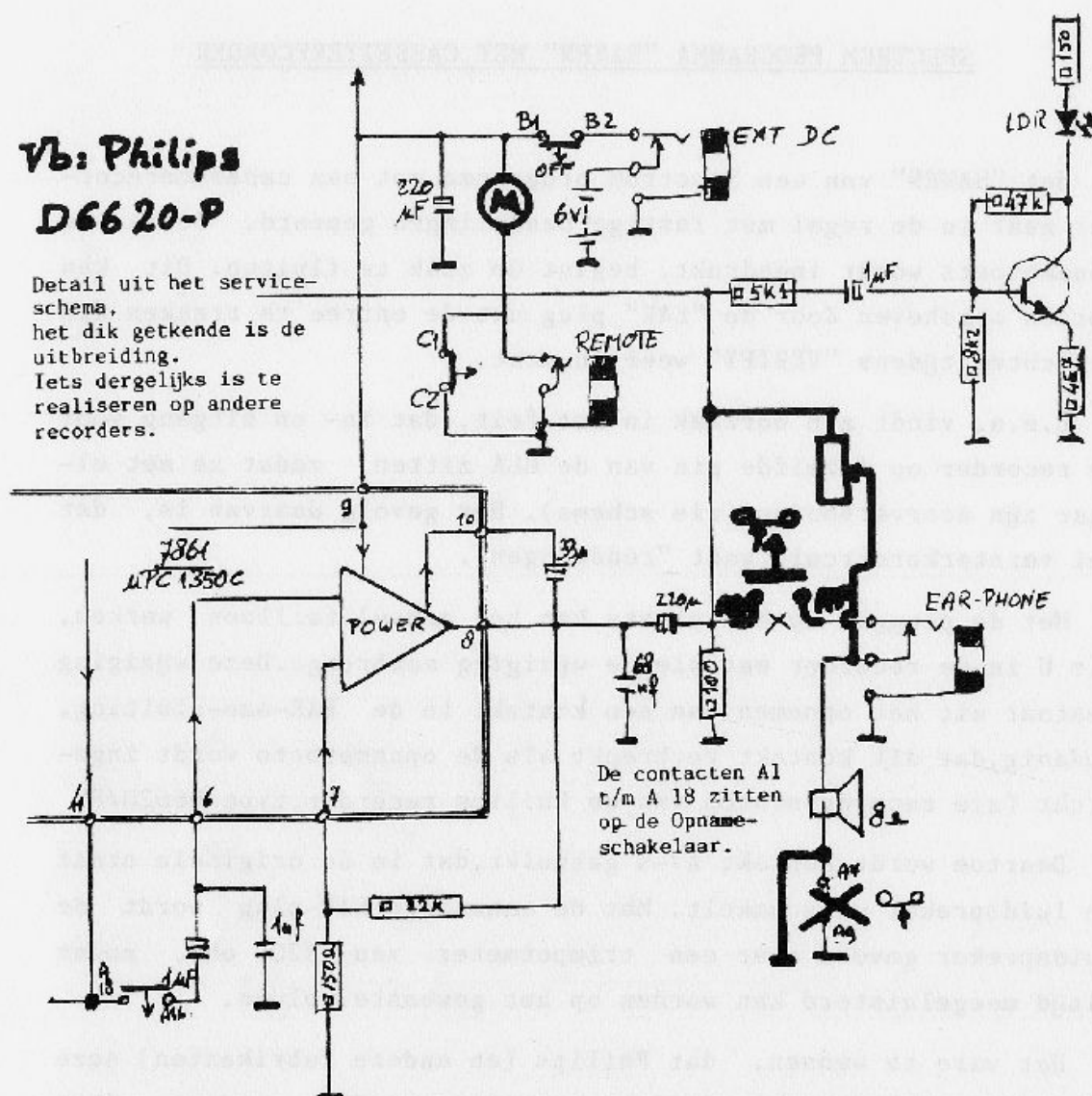
Voorts merk ik nog op dat de datarecorder van Panasonic, type RQ 8100, om verschillende redenen absoluut ongeschikt is voor de Spectrum.

Desgewenst geef ik graag nadere inlichtingen.

J. Ligthart
Oudenrijnsingel 12
3454 BG De Meern
Tel. 03406-2437

Vb: Philips
D6620-P

Detail uit het service-
schema,
het dik getkende is de
uitbreiding.
Iets dergelijks is te
realiseren op andere
recorders.



NEPROM VOOR DE ZX 81

Alweer een nieuw soort geheugen-IC? Nee, het is zoals u het leest: een nep-ROM of nee, eigenlijk is het een nep-RAM, nee toch ook weer niet...

Leest u maar verder, dan zal alles wel duidelijk worden.

Het is eigenlijk een uitbreiding van uw geheugen met 6 kB, maar dan in het gebied van 8 - 16 k. Ook is hij batterij gevoed, zodat alle informatie behouden blijft wanneer u de spanning van de ZX 81 afzet.

De ZX 81 kan men nu niet bepaald rekenen tot de snelste computers hoewel hij voor zijn prijs tot veel in staat is. Zoals u weet is hij het snelst in MC-programma's, maar die moeten in een REM-regel of boven RAMTOP staan. Dit heeft als nadeel dat de programma's overschreven kunnen worden of, wanneer het programma "crasht" of wanneer er even een spanningsonderbreking is, dan zal de CPU een RESET uitvoeren, waardoor het gehele geheugen wordt gewist. Gevolg: opnieuw beginnen met laden of, in het ergste geval, opnieuw programmeren.

Veel gebruikte MC-programma's, zoals een TOOLKIT of een snel-laadroutine of een high-resolution-programma, moeten eigenlijk steeds ter beschikking staan, zodra u de ZX 81 aanzet. Dit kan door middel van een EPROM, maar die moet eerst geprogrammeerd worden en er moet ook een EPROM-kaart voor aangeschaft of gemaakt worden en niet te vergeten een PROGRAMMER!

Reken maar uit wat dat gaat kosten.

De schakeling.

Deze schakeling maakt gebruik van CMOS-RAM's. Door de lage stroomopname zijn ze eenvoudig te voeden uit een 3-volt batterij of 2 NC-batterijen. Een RAM laat zich ook eenvoudiger overschrijven dan een EPROM.

De hier gebruikte IC's zijn van het type HM 6116 (max 200 ns). Ze zijn wel duurder dan een EPROM, maar gezien de hele schakeling bent u toch goedkoper uit. De HM 6116 heeft een geheugencapaciteit van 2 k x 8 bit. Er worden 3 gebruikt, zodat in totaal 6 kB ter beschikking staat. Behoorlijk wat voor MC-programma's!

Gebruikt wordt het adresbereik van 8192 tot 14336. Het volgende gebied tot 16384 wordt gemeden omdat veel I/O-uitbreidingen "memory mapped" zijn en enkele adressen vlak onder 16384 gebruiken. Omdat de beeldopbouw wordt verzorgd door de ULA, is gebruik gemaakt van de ROM-voet omdat hier de adreslijnen van de ULA ter beschikking staan en de ROM meestal in een voetje zit. Hierdoor is het eenvoudig om kleine letters of andere symbolen zelf te maken op uw ZX 81.

Het schema.

Het adresgebied van 8 kB wordt door de DEKODER 74 LS 138 in 4 stukken gedeeld van elk 2 kB. Dit wordt vrijgegeven wanneer ROMCS en A15 laag zijn. Het ROMCS-sigitaal bestaat uit een OF-koppeling van A14 en MREQ. Door een logisch 1-niveau op de ROMCS-lijn worden de ZX 81-ROM en ook onze NEPROM geactiveerd.

Vier silicium diodes vormen een EN-poort aan de uitgangen 1-4 van de decoder. Dit signaal wordt aan de ROMCS-lijn gelegd en de 3 volgende uitgangen komen aan de RAMCS-ingangen, die door PULL-UP-weerstanden hoog worden gehouden d.m.v. de batterijspanning.

Transistor T1 zorgt ervoor dat de OE-ingangen van de RAM's laag zijn, waardoor de uitgangen actief worden, dit alleen bij ingeschakelde voeding. Met T2 kan men de WE-ingangen hoog of laag maken, d.m.v. een enkelpolige schakelaar. Bij gesloten schakelaar is de RAM beveiligd tegen overschrijven. De diodes D1 en D2 zorgen voor het "omschakelen" naar de batterij. Dit moeten germanium typen zijn, omdat deze een lagere drempelspanning hebben. Weerstand R11 zorgt voor het bijladen van eventuele NC-batterijen. Bij alle andere soorten moet R11 weggelaten worden.

De print.

Door de kleine afmetingen kan het geheel in de originele behuizing van de ZX 81 worden geplaatst, hoewel er bij uitvoeringen 1 en 2 misschien wat problemen met de hoogte kunnen optreden. Bij type 3 gaat het prima. Een andere mogelijkheid is de schakeling in zijn geheel buiten het kastje aan te brengen d.m.v. een stukje flatcable. Bent u in het bezit van een groot toetsenbord met kast (bv. DKtronix), dan is er helemaal geen probleem: ruimte genoeg!

Omdat er onder de IC-voetjes weerstanden komen, moeten het "open" voetjes zijn; eventueel kunnen de weerstanden R1 t/m R5 en R7 ook aan de onderzijde van de print geplaatst worden.

Let op: er moeten 5 doorverbindingen gemaakt worden, omdat de print dubbelzijdig is. Wordt het printje op de ZX 81-print gezet, dan moeten de gaatjes voor het middelste voetje voorzien worden van zgn. HEADER-pennen; dit zijn dunne pennen die verbonden zijn door een plastic stripje. Ze zitten op IC-voet afstand van elkaar en kunnen gemakkelijk op de juiste lengte worden afgesneden of -gebroken.

Neem wel lange pennen (18 mm). Ze worden vanaf de onderzijde in de gaatjes gestoken en aan weerszijden gesoldeerd. Wanneer het printje niet op de ZX 81-print komt, maak dan gebruik van een stukje flatcable, voorzien van DIP-connectors. Eén zijde in het ROM-voetje en de andere zijde in het middelste voetje van de NEPROM-print. Het kabeltje mag niet langer zijn dan 20 cm i.v.m. stoorimpulsen en de capacatieve werking van de flatcable.

Omdat de decoder IC-74LS138 aan de boven- en onderkant van de print aansluitingen heeft, kan dit IC het beste rechtstreeks op de print worden gesoldeerd. vindt u dat te gevaarlijk, snij dan aan de zijkanten van een 16-pens voetje wat plastic weg om de pennetjes aan de bovenzijde te kunnen solderen. Let op: maak geen kortsluiting!

Aansluiten.

Na het openen van uw ZX 81 wordt het ROM-IC voorzichtig verwijderd, door aan beide korte zijden een klein schroefvedraaiertje er onder te steken en het voorzichtig omhoog te wippen. Het ROM-IC zit meestal in het midden van de print, in een 28-pens IC-voetje. Let goed op welke kontakten er niet worden gebruikt!

Om een defect door statische lading te voorkomen steekt u het IC in een stukje koolstofrubber. Ook goed gaat het in een stukje piepschuim dat gewikkeld is in aluminiumfolie.

Is het ROM-IC gesoldeerd, dan zit er niets anders op dan het er uit te solderen met behulp van désoldeer-litze en dan een 24- of 28-pens voetje te plaatsen.

Plaats nog geen IC's in de print, maar sluit de lege print aan op de ZX 81-print. Door middel van stukjes soepele draad moeten nu nog de punten 2, 3 en 4 met de ZX 81-print verbonden worden:

Punt 2 aan de WR-lijn (22 van de CPU of 17A van de connector).

Punt 3 aan A15 (5 van de CPU of 11B van de connector).

Punt 4 aan A13 (3 van de CPU of 13B van de connector).

De + van de batterij aan punt 5 en de - aan punt 6 op de print. Het schakelaartje komt aan de punten 1 en 6. Schakel de ZX 81 in en meet de spanning aan de punten 24 + en 12 1 van de IC-voetjes. Dit moet iets minder zijn dan 5 V. Bij uitgeschakelde voeding en aangesloten batterij moet dit minimaal 2,4 volt zijn, behalve de ROM. Plaats nu de IC-s met de stip of de inkeping gericht naar de 74LS138, in de voetjes.

Schakel de ZX 81 weer in en als alles goed is moet de cursor nu verschijnen. Gebeurt dit niet, dan is er iets mis en moet u alles nog eens goed controleren.

Testen.

Open de "schrijfbeveiligingsschakelaar" en typ het volgende

```
10 FAST                      Dit programma verplaatst
20 FOR N = 7680 TO 8191      de tekengenerator (512 b.)
30 POKE N + 512, PEEK N      vanuit de ROM naar onze
40 NEXT N                    NEPROM.
```

Geef nu een RUN. Wanneer het programma is afgewerkt schakel dan de ZX 81 uit. Na enkele minuten weer inschakelen en het volgende programma intypen:

```
10 FAST
20 FOR N = 7680 TO 8191
30 IF PEEK N ^ PEEK (N + 512) THEN STOP
40 NEXT N
50 PRINT "OK"
```

Wanneer alles in orde is (er zijn geen bytes veranderd) verschijnt er "OK" op het scherm. Is dit niet het geval dan zal alles nog eens gecontroleerd moeten worden op verkeerde aansluitingen.

De tekengenerator.

Zoals u wellicht weet, bestaat elk teken van de ZX 81 uit 8x8 punten, waarbij een horizontale lijn van 8 punten de inhoud is van 1 byte. De gehele tekengenerator bestaat dus uit 64 x 8 = 512 bytes. Het beginadres is 7680.

Het volgende programma geeft de ZX 81-CPU en -ULA de opdracht om met de tekengenerator in de NEPROM te gaan werken i.p.v. de ZX 81-ROM. Het wordt ook in de NEPROM geladen:

```
10 POKE 8704,62              40 POKE 8707,71
20 POKE 8705,32              50 POKE 8708,201
30 POKE 8706,237             60 RAND USR 8704
```

Geef nu een RUN - en er gebeurt niets! Hoe kan dat? De NEPROM tekengenerator is dezelfde als die in de ZX 81-ROM!

FIRMWARE

Inhoud van deze rubriek in dit nummer.

07.4.5.003	07-20	TASWORD en printer (v/d Maas)
07.4.5.004	07-21	Meer ervaringen met Tasword Twee (Bredenbeek)
07.4.5.006	07-23	Bijzondere symbolen en Tasword Twee (Breedem)
07.4.5.001	07-14	Tasword tips (V/d Maas; Schim v/d Loeff)

Eerder gepubliceerd in rubriek 4 FIRMWARE

06.4.5.001	06-15	Ervaringen met TASWORD TWEE (Schim v/d Loeff)
06.4.9.001	06-14	De Quantum Leap (Schim v/d Loeff)

TASWORD EN PRINTER

Na aanschaf van een printkabel (36 pins aan printer en een 9-way D-plug aan interface 1 - meteen meebestellen bij Impuls) zijn er nog de nodige handelingen te stellen alvorens geprint kan worden met de AVT-100 beta (ongeveer hetzelfde zal ook voor andere niet ZX-Spectrum printers gelden).

Na bestuderen van de handleiding bij de printer zult U concluderen dat de Dip-switches in de volgend stand moeten staan:

on		on	on	on	on	on	on
1	2	3	4	5	6	7	8
	off						

(Auto-feed on is prettig voor MF-file
maar lastig voor Tasword, oplossing hierna)

Het openen van de channels vindt U in de handleiding van de interface op bldz 35 -38.

Maar als U Tasword wil listen en aanpassen kunt U beter het programmatje op blz. 38 onderaan gebruiken. Nummer de regels van 1 tot 5 en laadt het BASIC-programma met MERGE "". Stop het programma zodra het niet mag (don't stop the tape). U kunt nu een listing van het basic-program maken, waarmee U de aanpassingen kunt volgen.

Laad TASWORD nu opnieuw. Na laden van TASWORD brengt STOP u in het menu. BREAKen met CAP SHIFT en SPACE. Bij alles wat U wilt veranderen krijgt U OUT OF MEMORY omdat er geen geheugenruimte meer is. PRINT 65535-USR 7962 geeft 31 Bytes. Nu LIST en ENTER.

Verwijder de regels 3000 (65 onnodige spaties) en de 8 regels die dienen voor side print, die hebt U toch niet meer nodig: 55, 699, 8000, 8010, 8020, 8030, 8040, 8050.

U beschikt nu over meer dan 300 bytes om te manipuleren. Voeg nu de regels toe:

```
279 FORMAT "b";1200:OPEN#3;"b"
281 CLOSE # 5
```

Zowel de printer als TASWORD geven n line-feed, dus de tussenruimte tussen de regels wordt te groot. Om de printer telkens aan te passen is lastig. Veel eenvoudiger is het om de feed-opdracht uit TASWORD te verwijderen.

PRINT PEEK 60926 geeft het getal 10, dat is de line-feed opdracht (Zie ook in het menu als U g kiest). POKE 60926,0 doet de line-feed verdwijnen en U hebt nergens last van. Past U wél eerst TASWORD aan vóórdat U een file typt of LOADt want U krijgt anders na de POKE een garnalensalade en het werk was voor niets.

U kunt blokken printen en deze blokken afwisselend een ander lettertype geven. Ga daartoe naar het menu - BREAK - en voeg aan het eind van regel 279; CHR\$ en een of andere code uit de manual toe. Terug naar menu met GOTO 25.

J.H.A.M. van der Maas
Egberinksweg 101, Boekelo

MEER ERVARINGEN MET TASWORD TWEE.

Enige maanden geleden heb ik een exemplaar van het nieuwe Nederlandstalige tekstverwerkingsprogramma TASWORD TWEE gekocht. Dit programma wordt door Filosoft / Groningen in licentie uitgebracht, het Engelse origineel is gemaakt door TASMAN Software.

Tasword twee is al uitgebreid besproken in IMPULS n° 6. Ook ik ben zeer tevreden met dit programma, ik heb er al diverse teksten mee geschreven (waaronder deze) en dit gaat uiteraard veel beter dan op een ouderwetse typemachine met alle bijkomende sores. Wel vind ik dat je, als je echt ten volle van dit programma wilt genieten, een behoorlijk toetsenbord voor je Spectrum moet kopen (zelf heb ik een DK'-Tronics).

Er kleeft aan Tasword Twee (en hoogstwaarschijnlijk ook aan het Engelse origineel) echter een bezwaar: Mensen die (zoals ik) een zwart scherm met witte letters prefereren boven het nogal vermoeiende wit-met-zwarte letters dat min of meer standaard is op de Spectrum komen met Tasword van een koude kermis thuis. Als je teruggaat naar BASIC en vervolgens typt BORDER 0: PAPER 0: INK 7 is het scherm netjes wit-op-zwart, maar als je dan weer RUN in-tikt wordt je BORDER weer wit en zit je ineens tegen een blauwe PAPER-kleur aan te hikken. Een "inverse" scherm schijnt kennelijk niet te mogen van meneer Tasman, en je moet dus maar weer gedwongen terug naar het voor tekstverwerking nogal ongebruikelijke zwart-op-wit scherm.

Nu ben ik op machinecodegebied ook geen beginnening meer en ik ben ook nog in het bezit van de Hisoft Devpac machinecode monitor/assembler/disassembler, dus ben ik daarmee maar eens wat gaan zoeken in de ruim 10K aan code in de Tasword. Even in de Spectrum ROM disassembly kijken waar de BORDER-routine zit; jawel, op 229B hex zit het vermoedelijke "entry point". Dus ga ik met de "Search"-optie van MONS3 (de monitor/disassembler van Devpac) kijken of er in de Tasword-code de instructie CALL 229B (= CD 9B 22) zit en jawel, op adres FD48_h (=64840_d) staat de gewraakte instructie. Daar maak ik dus drie NOP's van^d (00 00 00) en inderdaad blijkt de Tasword nu af te blijven van de BORDERkleur.

De onderste 2 regels van het scherm blijken echter nog steeds "zwart op wit te staan", dus moet er nog meer gezocht worden. Om precies te zijn naar het start-adres van de attributen van deze 2 regels, want die blijken ook beïnvloed te worden door onze vriend Tasword. Door het zoeken naar deze getallen (5ACO en 5AEO hex, uiteraard doe ik dit met MONS 3!) kom ik er uiteindelijk achter dat adres 59993_d de waarde bevat die de attributen van de op een na onderste schermregel krijgen. Voor de onderste regel blijkt dit adres 64570_d te zijn.

De op één na onderste attributen krijgen normaal de waarde "7" hetgeen duidt op PAPER 0 en INK 7 en de onderste attributen krijgen de waarde "56" hetgeen overeenkomt met PAPER 7 en INK 0. Dus doe ik POKE 59993,56:POKE 64570,7 en de onderste 2 regels zijn nu ook in de juiste kleur.

Nu is er nóg een probleempje: de achtergrondkleur van het scherm is nu blauw en als je de kantlijn zet wordt de BORDER zwart. Ik kan me voorstellen dat u dit precies andersom wilt.

Ik ga dus weer zoeken, ditmaal naar het start-adres van de attributen van de bovenste regel (5800 hex). En ja hoor, er blijkt een LD HL, 5800 instructie te zijn met erna een loop met SET 3, (HL) en RES 3, (HL) er in. Dit maakt ook meteen duidelijk waarom de PAPER-kleur verandert van wit in geel als je de kantlijn zet. Omdat bits 3 - 5 van het attribute-byte de PAPER-kleur bepalen verandert deze door de SET- en RES- instructies van 111 ("7") in 110 ("6") dus van wit naar geel vice versa. Heb je echter nu PAPER 0, dan zal de Tasword bit 3 op "1" zetten dus wordt je PAPER nu 1 (blauw). De SET- en RES- instructies werken nu dus "verkeerd-om".

Het rest ons dus slechts om met wat POKE-opdrachten de SET- in RES-instructies te veranderen en omgekeerd.

Het volledige aantal POKES om Tasword voor een "inverse" scherm geschikt te maken wordt dus: POKE 64840,0: POKE 64841,0: POKE 64842,0 POKE 59993,56: POKE 64570,7 POKE 58509,222: POKE 58518,222: POKE 58513,158: POKE 58522,158

Let wel: als je nu onverhoopt weer een "normaal" scherm wilt moet je die "56" en die "7" weer verwisselen, evenals de "222" en "158" en de laatste 6 POKES weer opnieuw doen met deze verwisselde bytes. (Bovengenoemde POKES gelden in ieder geval voor de Nederlandse versie van FiloSoft. Op de originele Engelse versie heb ik ze niet uitgetest.)

Als je tekst wilt uitprinten heb je uiteraard een printer nodig. Helaas heb ik zelf alleen maar een ZX Printer en aangezien 32 karakters op die van mij soms al niet meer leesbaar zijn verwachtte ik al niet veel van 64 karakters op de ZX printer. Dat bleek mee te vallen, maar hier viel me iets anders op: ondanks de bewering van de handleiding dat je op dubbele hoogte kon printen bleek dit bij mij niets uit te maken. Ik ben benieuwd naar andere ervaringen hieromtrent.

Dan nog iets over de karakterset: Die begint op adres EF00_h (61184_d) en begint net als bij de normale Spectrum karakterset met de spatie en gaat door tot en met de blok-graphics. Als je een mooiere karakterset wilt maken gebruik dan de PSION-karaktergenerator op de HORIZONS-cassette (door de UDG-pointer op 23675 en 23676 te POKEN kun je steeds andere karakters maken). Maak die karakters dan in een 3 bij 7-matrix rechtsboven in de hoek en laad ze in Tasword met LOAD""CODE 61184.

Ergens helemaal achteraan in het BASIC-programma staat een DEF FN-statement. Omdat een DEF FN voorin het programma veel sneller door de Spectrum wordt gevonden dan een DEF FN achterin heb ik die maar op regel 1 neergezet. De Spectrum springt hier gewoon overheen net zoals bij DATA-statements!

Ik hoop met deze praktijk-ervaringen en -tips ook anderen te helpen om de uitstekende tekstverwerker ook prettiger te maken.

Jan BREDENBEEK
Hilversum

BIJZONDERE SYMBOLEN VIA TASWORD TWO PRINTEN.

Enige tijd geleden heeft de redacteur van dit, uw en mijn, lijfblad mij verzocht u een stukje te schrijven over Tasword two en printers. Met name ging het er om hoe gecombineerde lettertekens zoals letters met umlaut, trema, accent-grave etc. afgedrukt kunnen worden.

De moeilijkheid van het schrijven van een artikel over dit onderwerp is, dat iedere printer zo zijn eigen (on)mogelijkheden heeft. Zo kennen sommige printers alleen de ASCII-karakterset (American Standard Code for Information Interchange), andere printers kennen daarnaast een groot aantal bijzondere letters (é, è, ë, ð enz.) en bij weer andere printers kan men de karakterset geheel of gedeeltelijk zelf definiëren.

Ik wil beginnen een truc te beschrijven, die mijns inziens op iedere printer werkt.

Bij de ASCII standaard is CHR\$ 8 een backspace. Het printen van dit karakter heeft tot gevolg dat de printpositie één positie terug gezet wordt. Een voorbeeld, dat u met uw Sinclair Spectrum gewoon op het beeldscherm kunt uitproberen, zal dit verduidelijken. Geef de commando's:

```
OVER 1:PRINT "u";CHR$ 8;" ""
```

(U weet toch dat u, om een aanhalingsteken in een print-opdracht te plaatsen, twee maal aanhalingstekens moet intoetsen. Samen met de omsluitende aanhalingstekens dus totaal 4 maal).

Op het scherm verschijnt dan: u

De verklaring is als volgt. Eerst wordt de "u" gewoon op het scherm afgedrukt. De separator ";" zorgt ervoor dat de positie vlak achter de "u" als volgende print-positie wordt aangewezen. Vervolgens wordt opdracht gegeven CHR\$ 8 af te drukken. Dit is geen letterteken, doch wordt door de print-routine opgevat als commando om de print-positie één plaats terug te stellen. De positie, waar de u al staat, wordt zodoende weer als volgende print-positie aangewezen. Daarom wordt het aanhalingsteken op de zelfde positie afgedrukt als de "u". Het commando 'OVER 1' zorgt dat de "u" niet uitgewist wordt bij het afdrukken van het aanhalings-teken.

Zeer waarschijnlijk kunt u het zelfde principe toepassen bij uw printer. Het commando 'OVER 1' is dan natuurlijk overbodig, omdat de letter al op papier is afgedrukt, en toch niet meer uitgewist wordt (tenzij u een ZX-printer gebruikt). Werkt dit niet, raadpleeg dan de gebruiks aanwijzing van uw printer en zoek de juiste code voor backspace.

Er bestaat ook een delete code. Bij mijn printer is dat CHR\$ 127. Delete stelt ook de print-positie één plaats terug, doch wist daarmee het voorgaande letterteken uit (zoals het woord al doet vermoeden). Dit kan omdat delete werkt op de inhoud van de

printerbuffer, terwijl backspace eerst de inhoud van de printerbuffer op papier zet, voordat de print-positie een plaats terug gesteld wordt.

Mijn printer kent de volgende leestekens:

aanhalingsteken of umlaut:	CHR\$ 34
circonflexe:	CHR\$ 94
trema:	CHR\$ 130
accent-aigu:	CHR\$ 131
accent-grave:	CHR\$ 132

In combinatie met CHR\$ 8 kan ik op iedere letter één van deze tekens afdrukken.

Overigens heb ik bij het afdrukken van trema's en accenten wat problemen gehad met mijn interface (Kempston Centronics S). De software van dit interface weigert namelijk de karaktercodes 128 t/m 164 (graphics) door te geven aan de printer. U zult begrijpen dat ik weinig waardering kon opbrengen voor dit soort censuur. Gelukkig werkt dit interface met software, die van cassette geladen moet worden, en ben ik er in geslaagd de werking van deze software te doorgronden, zodat dit probleem opgelost kon worden. Mocht u problemen van die aard tegen komen, denk er dan aan de gebruiksaanwijzing van uw interface na te lezen.

De theorie met de backspace kan bij Tasword met succes in praktijk worden gebracht, door de graphic codes op handige wijze te definiëren. Hoe ik dit zelf heb ingedeeld, blijkt uit afb. 1. Kijkt u maar naar de definities van de graphic-codes 133, 134 en 136 t/m 138. In de praktijk werkt dit zo dat ik, door in de tekst een graphics-shift 6 (CHR\$ 137) op te nemen, een trema krijg op de letter die juist voor de graphic staat. Deze methode heeft voor- en nadelen. Het voordeel van mijn methode is dat ik zo vanuit tasword bijna alle mogelijkheden van mijn printer kan benutten. Het nadeel is, dat de tekst, zoals die op het beeldscherm verschijnt, niet zo mooi is doordat er allemaal zwarte blokjes in de tekst staan. Een tweede nadeel is dat de justification niet meer klopt. Tasword zorgt er immers voor dat alle regels 64 lettertekens bevatten, (en voegt zonodig spaties toe) zodat er een mooie rechterkantlijn ontstaat. Doordat er nu op het scherm tekens staan, die niet op papier worden afgedrukt, wordt dit teniet gedaan.

Schim van der Loeff heeft in Impuls nr. 6 uitgelegd hoe hij dit opgeloste. Enkele graphic codes definieerde hij als dubbele resp. drievoudige spatie. Die tekens neemt hij in een regel op als de justificatie in gevaar komt. Bij underline, enlarged, condensed en dergelijke heb ik dit overigens anders opgelost: in afb. 1 ziet u namelijk verschillende definities beginnen met 32 (=spatie). Dit heeft tot gevolg, dat eerst een spatie wordt afgedrukt en vervolgens de control-code naar de printer gezonden wordt. Zo doende wordt voor een graphic op het scherm een spatie op papier afgedrukt. Bij de accenten, trema's en dergelijke gaat deze truc natuurlijk niet op omdat deze in een woord worden opgenomen. Binnen één woord mogen natuurlijk geen spaties voorkomen. Daarvoor beveel ik dan ook de methode Schim van der Loeff aan. Zelf kon ik die niet meer toepassen omdat ik alle graphics al gebruikt had en van geen enkele afstand kon doen.

Ik wil ook nog een andere aanpak beschrijven, waarmee mooiere resultaten te boeken zijn. De methode is vooral interessant als u slechts een beperkt aantal letters met leestekens nodig hebt. Stel dat u alleen maar een é, ë en een ü nodig hebt. U kunt dan bijvoorbeeld graphic-5 (CHR\$ 133) definiëren als: 97 8 34 (dus: "a", backspace, umlaut). Wilt u in uw text een "ä" hebben, dan typt u gewoon in graphic-mode een 5 in. Er ontstaan dan geen problemen met de justificatie.

Als u het helemaal mooi wilt maken, dan kunt u ook de karakterset van Tasword nog aanpassen, zodat ook op het scherm een a-umlaut verschijnt als u een graphic-5 invoert. Om dat voor elkaar te krijgen moet u begrijpen hoe de vorm van een letter bij een Sinclair in het geheugen is vastgelegd. (Lees hierover ook het hoofdstuk over User Defined Graphics in de Sinclair manual). Om een Sinclair-letter te ontwerpen dient u een rooster van 8x8 te tekenen en daarin een aantal hokjes zwart te maken, totdat de gewenste letter ontstaat. Vervolgens schrijft u daarnaast hetzelfde patroon over, maar nu met enen en nullen: voor de hokjes die wit waren gebleven vult u een nul in en voor de zwarte hokjes een 1. (Zie afb. 3) Zo ontstaan acht binaire getallen. Met de functie BIN kunt u ze eenvoudig omrekenen naar decimale getallen en op de juiste plaats POKEn.

Eerst moet ik echter nog wat over Tasword-letters vertellen. In de eerste plaats zijn de letters van tasword maar half zo breed als normaal. Daarom worden slechts de vier rechtse (=least significant) bits gebruikt. Enkele praktijkvoorbeelden treft u aan in afb. 4. Ten tweede moet u weten dat het basis-adres van de Tasword-letters 60928 is. Dat betekent dat de lettermodellen zich uitstrekken van adres 61184 tot 62079. Stel dat u wilt weten waar de letter "a" is opgeslagen, dan gaat u als volgt te werk:

- de ASCII-code van "a" is 97 (dit kunt u vinden in appendix A van het Sinclair manual);
- vermenigvuldig de ASCII-waarde met 8 (omdat iedere letter in acht bytes is vastgelegd) $8 \times 97 = 776$;
- tel het gevonden getal op bij het basisadres van de karakter-generator: $60928 + 776 = 61704$;
- dit betekent dat de vorm van de letter "a" is vastgelegd op de adressen 61704 t/m 61711. Met het POKE-commando kunt u hier nu zelf veranderingen aanbrengen als u dat wilt.

Om een voorbeeld te geven wil ik terugkomen op mijn voorstel om CHR\$ 133 te definiëren als "a-umlaut". Eerst berekenen we daar toe waar CHR\$ 133 in het geheugen staat: $(133 \times 8) + 60928 = 61992$. Wat we op de adressen 61992 t/m 61999 moeten poken, kunnen we al goeddeels zien in afb. 4. Alleen moeten er nog twee puntjes boven komen. Met een timmermansoog schat ik: BIN 00000101=5 decimaal. Voeg aan het Tasword-programma de volgende regels toe:

```
1 FOR i=61992 TO 61999:READ d:POKE i,d:NEXT i
2 DATA 5,0,6,1,7,5,7,0
```

Na het RUN-commando kunt u meteen uitproberen hoe het er op het beeldscherm uit ziet als u graphic-5 intoetst. Volledigheids halve zullen we dan ook nog de 32-letter-per-regel-optie aanpassen (ik bedoel extended-C: change window on text). Het basis-

adres daarvoor is: 59904. De berekening is dus: $133 \times 8 + 59904 = 60968$. Voor het model kijken we naar afb. 3. Hier moet alleen BIN 00101000 = 40 decimaal boven gezet worden om de puntjes toe te voegen. Het programma van zojuist kan aangepast worden:

```
1 FOR i=60968 TO 60975:READ d:POKE i,d:NEXT i
2 DATA 40,0,56,4,60,68,60,0
```

Zowel bij 64 als bij 32 tekens per regel verschijnt nu een ä op uw scherm als u graphics-5 intoetst. Als u tevreden bent over het resultaat kunt u de regels 1 en 2 weer uit het programma verwijderen en Tasword 'SAVE-n'. Voor het op deze manier veranderen van de Tasword-karakterset kan ik het programma "Tasdef" van de heer Nagel en Dr. Kam aanbevelen (afb. 5). Dit programma kunt u gewoon toevoegen aan het BASIC-programma van Tasword.

Wel moet ik nog opmerken, dat u de gehele 64-letter/regel karakterset van Tasword zelf kunt definiëren, van de 32-letter-per-regel-karakterset uitsluitend de graphics, CHR\$ 128 t/m 143.

Ik hoop u hiermee voldoende te hebben ingelicht over de mogelijkheden van Tasword. Zijn er echter nog vragen onbeantwoord gebleven, dan verneem ik dat gaarne van u.

I. Breeden
Beatrixlaan 30
4213 CK Dalem

SOFTWARE

Inhoud van deze rubriek in dit nummer.

07.5.1.007	07-29	Remstatement met veel bytes (vAbbe)
07.5.2.043	07-31	Lineaire zoekmethode (Breedem)
07.5.3.016	07-35	Vervolg Complete Basicode Disassembly (Radius)
07.5.3.019	07-38	Handleiding Basicode Schrijfroutine (Raats)
07.5.3.022	07-41	Basicode routine (Ament)
07.5.4.028	07-46	Een-en-twintigen (Sprengers)

Eerder gepubliceerd in de rubriek 5 SOFTWARE

02.5.1.001	02-24	Implementatie ON ... GO TO ... ELSE (Kuiper)
02.5.1.002	02-25	Video inverse (Myers)
02.5.1.002	02-25	SCROLL-programma (Myers)
02.5.1.002	02-25	Screen Invert (Borm)
02.5.1.003	02-26	Smalle letters printen met de ZX-81 (vDorp)
02.5.1.004	02-27	READ - DATA (Surtel)
04.5.1.005	04-23	Het wissen van een blok BASIC (vOoyen)
04.5.1.006	04-24	Nogmaals: Merge en Autowipe (v.Abbe)
06.5.1.008	06-19	Programmeertips HEXLOAD/DUMP, FFFS (Weygers)
06.5.1.010	06-21	Twee Hulproutines v/d ZX-81 (Weygers)
01.5.2.001	01-27	Teken-programma (v/dHorst)
01.5.2.002	01-27	Cassette-Index (vKerkhoff)
02.5.2.003	02-29	Macrofotografic (deKorte)
02.5.2.004	02-31	Vergelijkingen met 2 of 3 onbekenden (v/dWorp)
02.5.2.005	02-32	Afstanden tussen twee plaatsen (Luysterb/Forger)
02.5.2.007	02-34	ZX80 4K ROM 2400 Baud Save (Radius)
02.5.2.011	02-38	Lineaire regressie (deZeeuw)
03.5.2.012	03-20	Renumber (vOoyen)
03.5.2.013	03-21	Fast-Save-Load 2400 Baud (vAbbe)
03.5.2.018	03-26	ZX en de centjes (vStaalduinen)
04.5.2.019	04-25	Fast SAVE/LOAD/VERFY 2400 Baud (v.Dorp/v.Abbe)
04.5.2.022	04-28	Nogmaals: ZX en de centjes (Weijgers)
04.5.2.023	04-29	Romeinse getallen (Bosselaar)
04.5.2.024	04-31	Bestanden (Breedem)
06.5.2.032	06-22	Functies (Noordman)
06.5.2.035	06-25	Sorteren en Bubblesort (Breedem)
06.5.2.041	06-31	Formatteren van getallen (Planheel)
06.5.2.042	06-32	SnelSAVEbare Psion's CHESS (Weygers)
03.5.3.001	03-27	BASICODE protocol (Radius)
04.5.3.006	04-39	BASICODE schrijfroutine (Radius)
06.5.3.012	06-33	The complete BASICODE disassembly (Radius)
01.5.4.001	01-28	Nuclear Power Plant (Berggren)
01.5.4.006	01-33	Kiengetal trekken (Kerkhoff)
01.5.4.007	01-36	Torens van Hanoi (Vierhout)
01.5.4.008	01-37	Boter, Melk, Kaas (Boerdijk)
01.5.4.009	01-38	Getallen omdraaien (Boerdijk)
01.5.4.009	01-38	Dobbelstenen gooien (Boerdijk)
02.5.4.010	02-39	Een-en-twintigen (deRoo)
02.5.4.013	02-42	Yahtzee (vAbbe)
02.5.4.015	02-44	23 Lucifers (Boerdijk)
03.5.4.016	03-32	Game of Life op z'n Booleaans (Radius)
03.5.4.021	03-37	12 Jockeys deel 1 (Radius)
03.5.4.024	03-41	Rekenoefeningen voor 't jonge volkje (Hielkema)
03.5.4.027	03-44	Yahtzee-Scorekart
01.5.5.001	01-39	Kasboek
01.5.5.005	01-43	Berekening van n-faculteit (Verhoeven)
03.5.9.001	03-45	IMPULSOFT-cassettes, beschrijvingen

REM-STATEMENTS MET VEEL BYTES

Voor machinecode programma's die worden geschreven in een REM-statement moet eerst het benodigde aantal bytes ruimte worden gemaakt. De primitieve manier is: 1 REM 123456.... etc., en dan evenveel karakters als benodigde mc-bytes. Dit kan reeds worden versneld door: 1 PRINT 1+2+3+4+5+6+7+8+9+ ... etc. Op die manier komen per cijfer met + teken 8 bytes ter beschikking.

Herhaling van deze regels d.m.v. EDITing versnelt het maken van ruimte verder. Rekening houden met de bytes, die door de regelnummers, het aantal karakters per regel en de N/L-karakters worden ingenomen, is hierbij essentieel en maakt e.e.a. tot een moeizame klus. Met een 27 bytes uitbreiding van mijn programma "MERGE" (zie hiervoor*) gaat het heel gemakkelijk.

Eerst de uitbreiding: laad "MERGE" met RAND USR 837 zoals beschreven. POKE 16510, 0; POKE 16521, 65. Voeg toe: 1 PRINT 1+2+ZZZZZ. POKE 16511, 86. Schrijf volgens de aanvullende hexdump vanaf adres 40BB_h=16571_d. SAVE als in "MERGE" is beschreven.

Natuurlijk kan ook direct worden uitgegaan van een REM-statement van 84 bytes, waarin dan achtereenvolgens de "MERGE"-hexdump en de onderstaande hexdump worden geschreven.

Het gebruik van deze extra mogelijkheid is als volgt. Schrijf 'ergens' - maar NIET in regel 1 - de BASIC-regel INPUT A. RUN deze regel en beantwoord de INPUT-prompt met het gewenste aantal bytes, dat de REM-statement moet omvatten. Geef dan het 'direct command': RAND A, RND USR 28802, RAND USR 28771 en regel 1 met de verlangde ruimte is daar!

*) Impuls nr 4, pag. 04-24

H.H.van Abbe

TASWORD TIP n° 2:

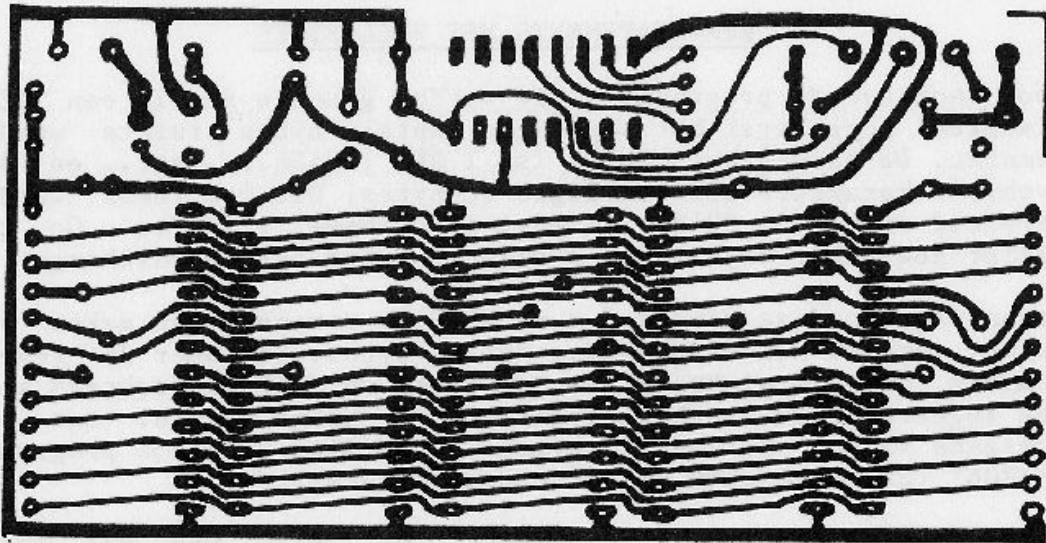
Als ik een TASWORD-file wil gaan printen kies ik altijd regelafstand 1 (default). Ik begon het dus lastig te vinden om steeds maar weer een 'overbodige' ENTER te moeten geven.

Dit is heel gemakkelijk te overbruggen. Delete regel 210 en verander regel 215 in: '215 POKE 62235, 1' Dat is nl. wat er feitelijk gebeurt bij het commando 'IF a\$="" THEN LET a\$="1"'.
 *)

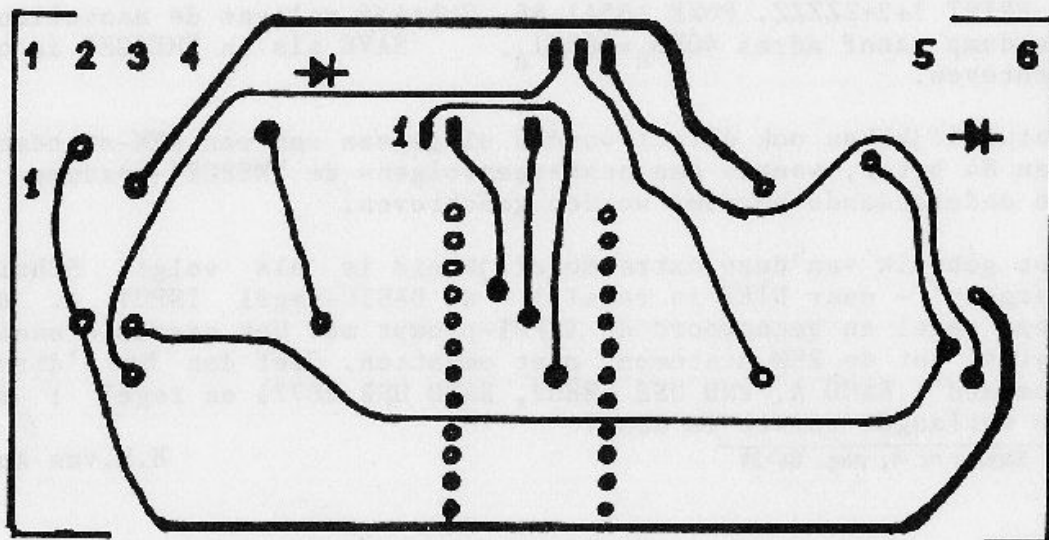
Toen ik TASWORD voor een discdrive geschikt ging maken heb ik zo ook een aantal andere regels overbodig gemaakt (wat bovendien ruimte bespaart). Daartoe behoren o.a. regels 800, 900, 920, 930 en 940. Maar PAS OP! U moet dan wel de GOSUB-opdrachten uit de regels 1030, 1050, 1100, 700 (ontbreekt in de Hollandse versie) verwijderen, terwijl voor GOSUB 920 in regel 400 een oplossing moet worden gezocht. Ik heb ook die regels geschrapt - nu ik eenmaal die code's heb ingevoerd hoef ik die niet zo nodig meer te veranderen. Dat ruimt ook al weer op!

Pieter SCHIM van der LOEFF
Den Haag.

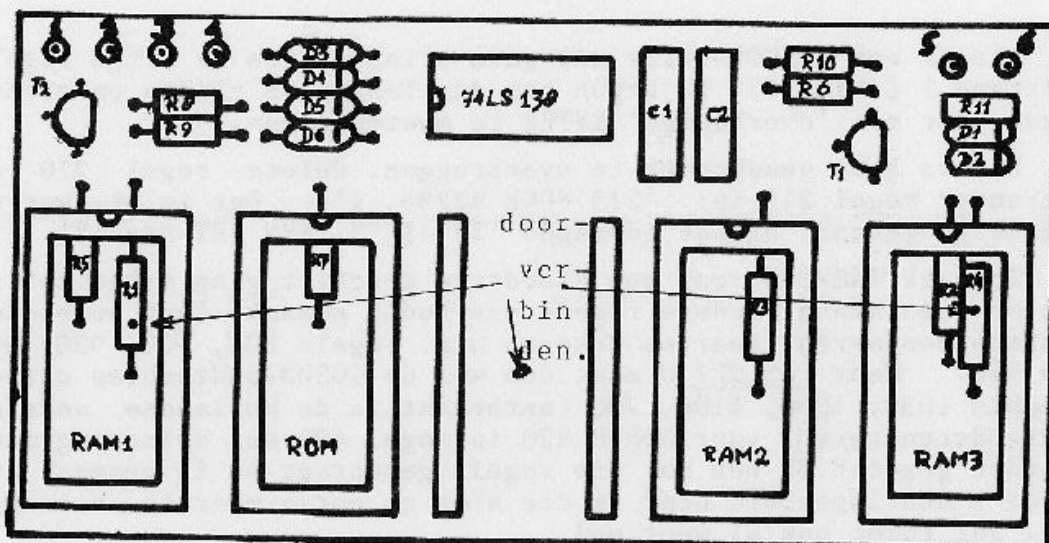
(behoort bij "(N)EPRON" van pag. 15 - 18)



onderzijde



componentenzijde



onderdelen-opstelling

LINEAIRE ZOERMETHODE

Wellicht hebt u al eens geprobeerd om een bestand op te zetten met behulp van het DIM-commando. Dit kan heel eenvoudig. Een bestandje voor namen en adressen kan gedimensioneerd worden met: DIM n\$(50,15):DIM a\$(50,15):DIM w\$(50,15)

(Bij andere BASICs dan Sinclair-BASIC hoeft het tweede subscript (15, voor het aantal lettertekens per string) gewoonlijk niet aangegeven te worden.) Daarbij is n\$ voor de namen, a\$ voor de adressen en w\$ voor de woonplaats. Zo krijgen we ruimte voor 50 namen en adressen, van elk 15 lettertekens. (Dat laatste dus alleen bij Sinclair).

U kunt zo'n array natuurlijk ook in één keer dimensioneren als een meer-dimensionale array: DIM a\$(50,3,15). Ik neem aan dat ik u hier niets nieuws mee vertel.

Wil zo'n bestand nut hebben, dan moet het mogelijk gemaakt worden om in zo'n bestand de gegevens van een bepaalde persoon op te zoeken. Dit kan met een eenvoudig programmaatje.

```
10 REM n$ voor namen, a$ voor adressen en w$ voor woonplaatsen
100 PRINT "Gegevens opzoeken"
110 INPUT "Geef een naam op", h$
120 FOR n=1 TO eind :REM eind moet natuurlijk eerst een waarde
      gekregen hebben, en wel het volgnummer van
      het laatste record.
130 IF n$(n)=h$ THEN PRINT n$(n)'a$(n)'w$(n)
140 NEXT n
```

Dit programma zal op de Sinclair niet geheel voldoen. Stel dat U de gegevens van mijnheer Mol wilt opzoeken. Als u deze naam intypt zult u zijn adres niet op het scherm krijgen, zelfs al komt de naam wel in het bestand voor. Dit komt doordat de Sinclair 15 letters heeft gereserveerd voor de naam, en daarom de ontbrekende letters heeft opgevuld met spaties. Om het adres van mijnheer Mol te weten te komen zult u dus "Mol" en twaalf spaties in moeten tikken. Dit is natuurlijk te gek. Vervang daarom regel 130 door:

```
130 IF n$(n, TO LEN h$)=h$ THEN PRINT n$(n)'a$(n)'w$(n)
```

De computer kijkt nu nog maar naar de eerste drie letters van de namen in het bestand (als u een naam van 3 letters gevraagd had). Deze constructie heeft nog een voordeel: stel dat in uw bestand de namen: Mol, J.W., Mol, J.A., Mol, J.W.Ch. en Mol, J. voorkomen, dan kan het voorkomen dat het u niet meteen te binnen wil schieten wie van de vier uw vriend Jan is. Door de bovenstaande wijziging in het programma krijgt u de gegevens van al deze personen op het scherm. Als u echter "Mol, J.A." opgeeft, dan krijgt u alleen zijn gegevens.

(Om die reden is het niet gek om ook bij Microsoft-achtige BASICs deze constructie toe te passen. In dat geval moet regel 130 luiden: 130 IF LEFT\$(n\$(n), LEN(h\$))=h\$ THEN PRINT n\$(n);

```
CHR$(13);a$(n); CHR$(13);w$(n)
```

De aldus beschreven zoekmethode staat bekend als "lineaire zoekmethode". De voordelen zijn duidelijk: het programma is een-

voudig en het bestand hoeft niet gesorteerd te zijn. De nadelen komen aan het licht, als u met deze zoekmethode grotere bestanden te lijf wilt gaan: de methode is erg langzaam. Gemiddeld moet 50% van het bestand doorzocht worden om de gewenste gegevens te vinden. Bij grotere bestanden wil ik daarom de logaritmische zoekmethode aanbevelen.

Logaritmische zoekmethode.

Om te begrijpen hoe deze methode werkt moet u eens nagaan, hoe u te werk gaat als u een kaart uit een alfabetisch gesorteerde kaartenbak wilt hebben. U gaat natuurlijk niet alle kaarten een voor een bekijken (Zo werkt de lineaire methode). U pakt ongeveer de middelste kaart, en aan de hand daarvan weet u of u verder naar achter of naar voor moet kijken. Zo doorloopt u met grote stappen de kaartenbak, totdat u bijna de gevonden naam hebt en dan pas bekijkt u de kaarten stuk voor stuk. Op die manier kunt u uw computer ook laten zoeken. Uit het voorgaande heeft u begrepen dat deze zoekmethode alleen toegepast kan worden bij gesorteerde bestanden.

Laat mij u eens een programma voorschotelen.

```
10 REM n$ voor namen, a$ voor adressen en w$ voor woonplaatsen
100 LET begin=1
110 REM zorg dat de variabele 'eind' het laatste record van uw
    bestand aangeeft.
120 INPUT "Welke naam moet ik opzoeken? ", h$
130 LET lengte=LEN h$
140 LET midden=INT ((begin + eind)/2)
150 IF n$(midden, 10 lengte)=h$ THEN GOTO 200
160 IF h$'n$(midden) THEN LET begin=midden+1
170 IF h$'n$(midden) THEN LET eind=midden-1
180 IF begin=eind THEN GOTO 140
190 PRINT "Die ken ik niet.":STOP
200 PRINT n$(midden)'a$(midden)'w$(midden)
```

Bij andere BASICS wordt regel 150: IF LEFT\$(n\$(midden), lengte=lengte)=h\$ THEN 200

Het herhalingsgedeelte van de zoekbewerking treffen we aan tussen de regels 140 en 180. We zien dat aanvankelijk het hele bestand als zoekgebied ('begin' tot 'eind') gekozen wordt. Na elke herhaling wordt het zoekgebied gehalveerd. Van het aantal herhalingen dat maximaal nodig is, kunnen we het volgende staatje maken:

bestands grootte	max.aantal tests (regel 150)
1	1
2-3	2
4-7	3
8-15	4
16-31	5

A.J. Kool beschrijft in HCC-nieuwsbrief nr.64 blz. 29, dat het aantal tests volgt uit de formule:

tests= 1 + LOG (bestands grootte) (let wel: logaritme met grondtal twee.)

Door zijn artikel weet ik nu ook dat deze zoekmethode "binary search" of "logarithmic search" genoemd wordt.

Doordat de logaritmische zoekmethode (in vergelijking met de lineaire methode) tamelijk ingewikkeld is, heeft die geen voordeel bij kleine bestanden. Volgens mijn berekeningen ligt het keerpunt ongeveer bij een bestand van 23 elementen. Bij kleinere bestanden is de lineaire methode in het voordeel, voor grotere de logaritmische.

Zelf gebruik ik de logaritmische methode in een programma dat momenteel 31 records bevat. De maximale zoektijd bedraagt nu 0.24 sec. Dit betekent dat in een tweemaal zo lange tijd (0.48 sec) een record in een bestand met 1023 elementen gevonden zou worden. Als u met werkelijk gigantische bestanden goochelt, en de zoektijd u nog te lang is, wil ik u de volgende filosofie aan de hand doen. Stel dat de beginletter van uw eerste record een "A" is en van uw laatste record een "Z". Als u uit dit bestand mijnheer Aalders wilt zoeken, dan zal de zoekroutine in het midden van het bestand beginnen te zoeken. Uw computer zou echter veel meer kans op succes hebben als hij meer aan het begin zou gaan zoeken. Voor het berekenen van een betere gissing zou u gebruik kunnen maken van de operator 'CODE' (bij andere BASICs heet dit meestal (ASC')). 'CODE' geeft de ASCII code van de eerste letter van een string. Probeert u daarom voor regel 140 eens:

```
140 LET midden=INT(begin+(eind-begin)*(CODE h$
-CODE n$(begin)+1)/(2+CODE n$(eind)-CODE n$(begin)))
```

Ik heb het zelf niet uitgetoetst. Mocht u foutmeldingen of verkeerde resultaten krijgen, bel mij niet doch gebruik uw fantasie om er een goedwerkende routine van te maken en schrijf daar een stukje over in dit, uw en mijn, lijfblad.

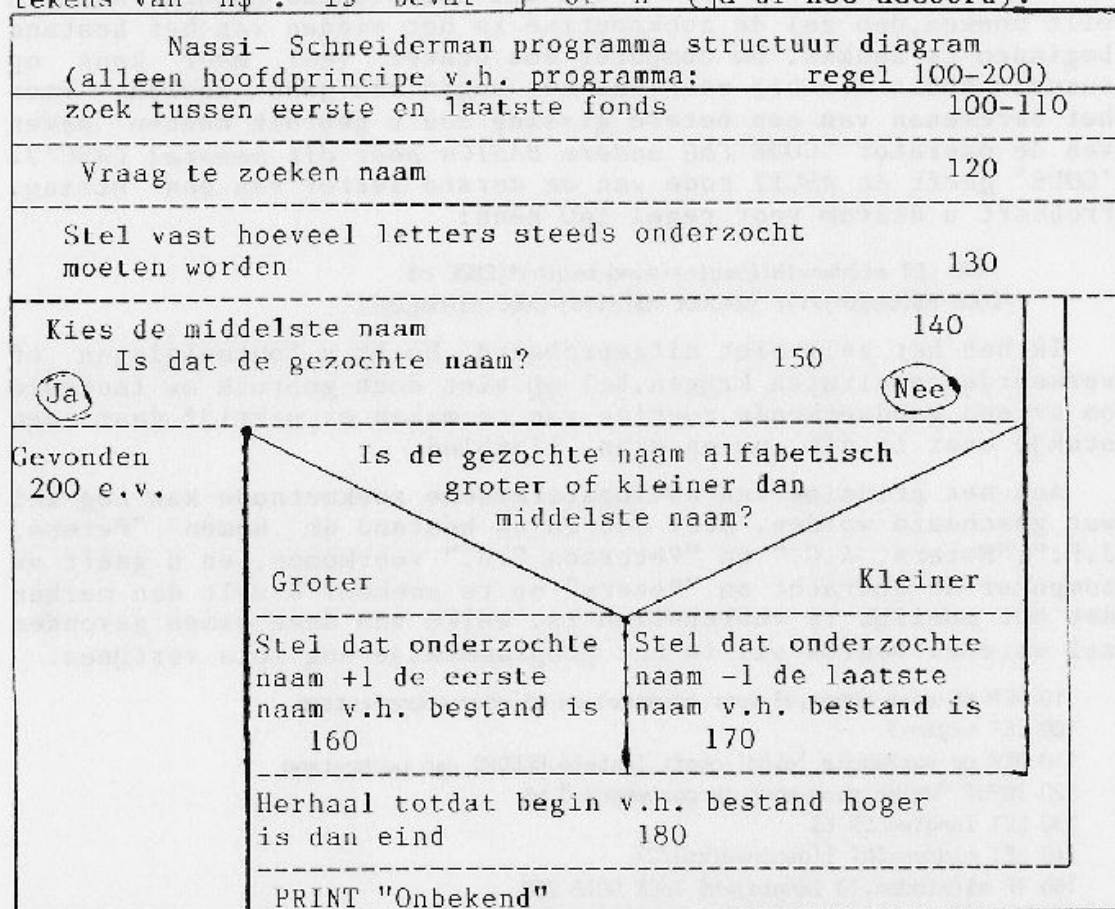
Aan het principe van de logaritmische zoekmethode kan nog wel wat geschaafd worden. Stel dat in uw bestand de namen "Peters, J.P.", "Peters, A.G." en "Peterson C.G." voorkomen, en u geeft uw computer de opdracht om "Peters" op te zoeken. U zult dan merken dat het moeilijk te voorspellen is, welke van deze namen gevonden zal worden. Daarom wil ik het programmaatje nog iets verfijnen.

```
10 REM n$ voor namen, a$ voor adressen en w$ voor woonplaatsen
100 LET begin=1
110 REM de variabele 'eind' geeft laatste RECORD van uw bestand
120 INPUT "Welke naam moet ik opzoeken? "; h$
130 LET lengte=LEN h$
140 LET midden=INT ((begin+eind)/2)
150 IF n$(midden, TO lengte)=h$ THEN GOTO 200
160 IF h$^n$(midden) THEN LET begin=midden+1
170 IF h$^n$(midden) THEN LET eind=midden-1
180 IF begin^eind THEN GOTO 140
190 PRINT "Die ken ik niet.":STOP
200 IF midden^1 THEN IF n$(midden-1, TO lengte)=h$ THEN LET
  midden= midden-1:GOTO 200:REM dit zorgt ervoor dat het eerste
  element, dat aan de gestelde eis voldoet, gevonden wordt.
210 PRINT n$(midden)
220 INPUT "Akoord? (j/n) "; i$
230 IF i$="j" THEN GO TO 270
240 LET midden= midden+1
250 IF n$(midden, TO lengte)= h$ THEN GO TO 210
260 GO TO 190 : REM niet gevonden
270 PRINT a$(midden)' w$(midden)
```

Tot slot wil ik nog de variabelen-definitie geven en het programma-structuur diagram (P.s.d.). Wellicht hebt u nog geen ervaring met het lezen van z.g. Nassi-Schneiderman diagrammen. U leest ze in principe van boven naar beneden. Voor een goede uitleg van deze manier van noteren wil ik verwijzen naar HCC-nieuwsbrief nr.39 (mei'82) blz. 12, waar Frank Gieske een artikel over dit onderwerp heeft geschreven.

Variabelen-definitie.

in: 'h\$', bevat te zoeken woord
uit: 'midden', bevat positie waar gezochte naam staat.
bijwerking: 'begin' en 'eind' bepalen het begin en eind van het gebied, waarin gezocht wordt. 'lengte' is het aantal lettertekens van 'h\$'. 'i\$' bevat "j" of "n" (ja of nee accoord).



Ivo Breeden
Dalem

TASWORD TIP n° 1:

Hebt U dat ook wel eens meegemaakt: zoh 'garnalenslatje' dat U kunt krijgen als U bijv. met TASWORD gaat printen zonder dat de printer aan staat? Daarna moet BREAK het programma weer klaren. Is dan de ingetypte tekst verloren? Ik heb dat een paar keer aldus kunnen oplossen:

Printer uit / (BREAK) / GOTO 25 / menu / kies p en 3x ENTER /
 OUT OF MEMORY / GOTO 25 / STREAM ALLREADY OPEN / GOTO 280 en
 alles is weer op orde. Misschien nog een keer naar GOTO 25.

J.H.A.M. van der Maas
 Egberinksweg 101, Boekelo

Vervolg op: Radius' BASICODE-DISASSMBLY uit vorig nummer.

De sluitpost van de hoofdroutine, beginnend met "endnr" op 16733

16733	254	32	40	38	229	42	32			
	endnr						(lyn-n)			
16740	66	86	35	94	237	83	36	66	225	24
	(lyn-n)						(lynnr)			
16750	179	54	118	35	34	32	66	54	0	35
		n-rem					(lyn-n)			
16760	54	0	35	34	30	66	54	2	35	54
					(lyn-l)					
16770	0	35	54	234	35	205	193	65	254	58
						regnr (byte)				
16780	48	207	254	48	56	203	203	175	203	167
16790	229	42	32	66	86	35	94	98	107	203
			(lyn-n)							
16800	37	203	20	203	37	203	20	25	203	37
16810	203	20	22	0	95	25	237	91	32	66
									(lyn-n)	
16820	124	254	36	212	22	66	18	19	125	18
					(newnr)					
16830	225	24	198							

De laatste instructie op adres 16831/2 luidt: JR,REGNR. De rest zult u zelf moeten decoderen.

In bovenstaand stuk van de hoofdroutine vinden we de draad van het verhaal weer terug, die we in vorige aflevering kwijt waren geraakt. Wij gaan nu dus verder bij "n-rem" op adres 16751.

Het feitelijke lezen van de BASICODE gaat hier beginnen. Maar eerst nog even iets verduidelijken:

U zult al wel hebben begrepen, dat aanduidingen onder sommige getallen LABELS voorstellen die in assemblertaal gebruikt worden. Maar sommige woorden staan hier tussen haakjes. Wel, dat zijn dan adresaanduidingen waar die labels te vinden zijn, dus CALL-instructies, maar ook adressen van HULPVARIABLEN.

Deze zijn:

STCKP (16924/5) bevat copie v.oorspronkelijke stackpointerwaarde
LYN-L (16926/7) adres LSByte v/d regellengtebytes der te vormen Basicregel.

LYN-N (16928/9) adres 1^e byte waar het regelnummer moet komen.

FIRST (16930/1) het adres van N/L-byte na COPY COPY; onthoudt waar begonnen is met laden.

LYNNR (16932/3) laatste gelezen regelnummer, als dat groter wordt dan 9216, dan wordt hier een vervangend nummer gemaakt.

De programmaloop.

"n-rem": Zet een N/L neer (eerste keer op adres (first);
bewaar adres regelnummer in "lyn-n";
maak regel nummer gelijk aan nul;
bewaar adres regellengte-bytes in "lyn-l";
maak startwaarde regellengte = 2;
zet hierna een REM-byte.

"regnr": Nu wordt een byte gelezen van de taperecorder.

Is het een cijfer?

Zo ja, stel het lijnnummer (regelnummer) samen;
(blijkt het lijnnummer groter dan 9216 te worden, dan
wordt een nieuw lijnnummer gemaakt dat telkens met 1 op
hoogt, met behulp van "lynnr en subr. "newnr").

Is het geen cijfer meer, dan springen naar "endnr".

"endnr": Blijkt het een spatie te zijn, dan terug naar "regnr".
Anders zet het gemaakte lijnnummer in de hulpvariabele
"lijnnr" op zijn plaats in de REM-regel, die in de maak is.
Als laatste teken was er iets gelezen dat geen cijfer en
ook geen spatie was, daarom is nu het regelnummer klaar
en de routine springt naar "alfa" op adres 16674.

We gaan dus twee bladzijden terug naar bekend terrein.

De hoofdschotel hebben we gehad, wat er nog rest zijn een aantal
subroutines, er is zelfs een (sub)**4-routine, dus hou je vast.

Van een subroutine weten we de naam al, de subroutine "byte".
Gelukkig komt die het eerst, op adres 16833 begint hij.

16833	205	222	65	56	251	22	7			
	byte		(bit)							
16840	205	222	65	203	27	21	32	248	203	59
	byte		(bit)							
16850	205	222	65	48	251	205	222	65	48	251
	stp1						stp2			
16860	123	201								

De subroutine "byte" werkt als volgt:

byte : Zoekt het startbit (=een hele blokpuls van 1200 Hz).

byte2: Haalt 7 bits binnen die de data vormen in ASCII,
eenbit = 2 volledige blokpulsen van 2400 Hz,
nulbit = 1 volledige blokpuls van 1200 Hz,
het achtste databit is eenbit, maar er wordt
niet naar gezocht.

stp 1: Zoekt eerste stopbit (is een eenbit).

stp 2: Zoekt tweede stopbit (is een eenbit),
levert de byte af aan de aanroep routine.

De subsubroutine "bit" start op adres 16862

16862	205	255	65	14	0	6	38	5		
	bit		(high)				==	een		
16870	202	238	64	12	219	254	203	127	32	245
			(rst)							
16880	205	9	66	16	251	62	9	185	48	228
	nul		(brk)							
16890	62	15	185	63	201					

Dit is de cruciale subroutine! Heb ik moeilijkheden met inlezen,
dan wil het veranderen van byte 38 op adres 16868 in 33,34 of 35
wel eens helpen.

Dus: POKE 16868, 33 of 34 of 35 bij leesmoeilijkheden.

Werking van de subroutine "bit" die een bit moet lezen:

bit : Zoekt de opgaande flank van een puls en als deze (door
"high") gevonden is;
zet teller C op nul,
zet teller B op 38. (zie opmerking voor cursieve regel.)

Er wordt nu voortdurend getest en geteld hoe lang de puls duurt, C telt op, B telt af.

Als de puls te lang duurt, gaat het programma weer helemaal opnieuw beginnen bij "rst" op adres 16622.

Er wordt dan zeker een stuk popmuziek of de Zevend van Beethoven ingelezen met zware trommelslagen.

De Breakkey wordt ook nog getest ("brk").

Is de puls weer weg, dan wordt de C teller gestopt, maar B telt verder af tot nul.

Hierna wordt gekeken of de puls voldoende lang geweest is, (misschien een sopraan geweest?), zo niet : Overdoen!

Wel goedgekeurd kan het een 1- of 0 bit voorstellen, daarmee gaat hij terug naar de aanroeper.

De sub³-routine "high" vanaf adres 16895:

```
16895 205 9 66 219 254
      high (brk)
16900 203 127 40 247 201
```

Voor de werking van deze subroutine verwijs ik naar Sinclair IMPULS nummer 1, artikel 01.1.3.005. Daar staat iets wat hier op lijkt en dat geldt ook voor de nu volgende

sub⁴-routine "brk", die trouwens ook gebruikt wordt als sub³-routine bij de call op adres 16880.

Subroutine "brk" begint op adres 16905:

```
16905 62 127 219 254 31
      brk
16910 216 237 123 28 66 195 71 65
      (stckp) (end)
```

De laatste instructie maakt dus een sprong naar 16711.

De kleine subroutine "newnr" op adres 16918, kan nodig zijn als de gelezen regelnummers te groot dreigen te worden.

```
16918 42 36 66 35 124 210
      newnr (lynnr)
```

Tot slot nog de gebruikte hulpvariabelen, genoemd twee bladzijden geleden:

```
16924 0 0 =stckp
16926 0 0 =lyn-1
16928 0 0 =lyn-n
16930 0 0 =first
16932 0 0 =lynnr
```

Van het klaargezette REM-statement voor het herbergen van het in machinecode geschreven vertaalprogramma voor BASICODE-2, blijken de laatste 7 puntjes (code 27) van de 427 gereserveerde bytes nog over te zijn. Het N/L-karakter ligt op adres 16941.

Wilt U de machinecode zonder meer intypen: ga Uw gang! Maar wilt u leren begrijpen, hoe de routines werken, ga dan zorgvuldig de decimale codes vertalen in mnemonics, o.a. te vinden in Chapter 28 van het ZX81 Manual. Ook vorige Sinclair IMPULS-nummers kunnen u op weg helpen. Na enige avonden noeste arbeid zal dan de dageraad gloren.

(zie Appendix op blz.)

H. Radius

H A N D L E I D I N G

van de BASICODE-2 schrijfroutine VERSIE-5.2A en 5.2B

De cassette bestaat uit twee kanten: Kant A bevat de standaard Basicode-2 routines en kant B bevat het conversie-programma. Laad de cassette kant A met LOAD"". Het programma presenteert zich met een menu en met de foutmelding 7/90. U kunt nu uw BASICODE-2 programma maken vanaf regel 1000. Voor kant A kunt u ook het inleesprogramma gebruiken daar dit ook de BASICODE-subroutines bevat en de vervangroutine op regel 400 en verder.

Voor de inhoud van het BASICODE-2 programma bent U als programmeur volledig verantwoordelijk, het vertaalprogramma wijzigt niets aan uw programmatekst. Gebruik dus bij voorkeur het BASICODE-protocol. Een aantal dingen, waar U zelf op moet letten, zijn:

- Het eerste regelnummer moet 1000 zijn en er aldus uit zien:

```
1000 LET A=100
1001 GOTO 20
1002 REM PROGRAMMA-NAAM
```

'A' krijgt hier de waarde die alle STRING-variabelen (bv. A\$) bij elkaar aan geheugen ruimte zullen gebruiken. In dit geval 100 bytes. Door de sprong naar regel 20 wordt deze ruimte gereserveerd. Sommige computers hebben dit nodig.

- Elke variabele moet uit hoofdletters bestaan.

- Een regel mag, inclusief regelnummer en spaties, niet langer zijn dan 60 tekens. Dat zijn twee regels op het ZX-81 scherm minus vier tekens.

- STRING-SLICING is niet toegestaan, zoals bv. LET I\$=A\$(6 TO 8). Het stukje "(6 TO 8)" mag niet, het zou in principe uitgedrukt moeten worden met behulp van LEFT\$, RIGHT\$ en MID\$. Dit kan een probleem vormen.

- U moet de BASICODE-2 standaard-routines gebruiken voor de functies: CLS, PRINT AT, INKEY\$, RND EN LPRINT. Zie hiervoor ook het HOBBYSCOOP BASICODE-2 protocol.

Bij het gebruik van INKEY\$ (GOSUB 200/210) wordt er gewerkt met IN\$ en bij GOSUB 300/310 is er zelf sprake van SR\$. Wilt u hiermee werken in uw eigen programma (regels 1000-....), vervang dan IN\$ door "spatie I\$" (of een andere letter). De spatie is van belang, omdat u later, als u uw programma uitgetest heeft op de juiste werking met de standaard-routines, die "spatie I\$" weer kan vervangen door IN\$. Houd dus in uw eigen programma ook "spatie I\$" aan. Doe dit tijdelijk vervangen van een variabele met twee letters en terug steeds door middel van de subroutine van regel 400.

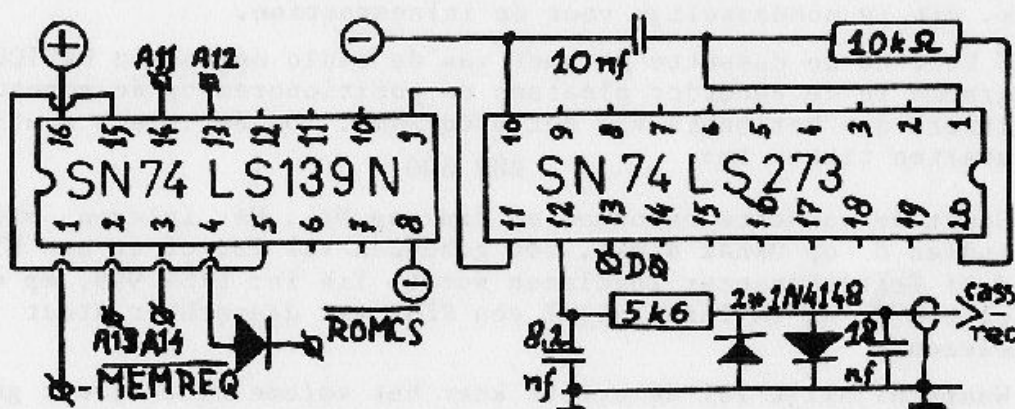
Voorbeeld: RUN 400
 "2" voor vervangen door het hele programma.
 "IN\$" voor woord.
 "spatie I\$" voor vervangen door
en terug RUN 400
 "2" voor vervangen.
 "spatie I\$" voor woord.
 "IN\$" voor vervangen door.

Wanneer uw programma klaar is moet u het op cassette SAVEN. Allereerst moet U regel 1 Deleten. Daarna kunt U het programma SAVEN door middel van RUN 30. Deze routine SAVet het programma automatisch 3X met tussenpauzes van 30 seconden.

Reset de ZX-81 door middel van PRINT USR 0 en laad dan kant B. Het programma presenteert zich evenals dat van kant A met een scherm en met de foutmelding 7/9100. Start het programma d.m.v. RUN. Het programma vertelt precies wat het doet en U kunt dat ook controleren. U mag echter nooit een regel toevoegen of weghalen als er een programma in tabel A\$ zit.

TECHNIEK 2.A

Voor Basicode 5.2A hebt uw een interface nodig, een I/O-poort, die memory-mapped is. Dit wil zeggen dat u hem kunt aanroepen met bijvoorbeeld POKE 14336,1.



Figuur 1: Voorbeeld van een I/O-poort. Poort-adres 14336

TECHNIEK 2.B

Bij BASICODE - 2B dient u het filter, dat in de computer achter de ULA komt aan te passen. U hoort daarvoor de computer open te maken en een condensator kort te sluiten. Even een waarschuwing vooraf: GA IN DE GARANTIEPERIODE NIET IN DE COMPUTER SOLDEREN, maar wacht tot die periode voorbij is. Er is geen importeur bereid kosteloos mee te werken aan experimenten die door onkunde de mist zijn ingegaan. Tevens kan de schrijver van dit artikel niet aansprakelijk worden gesteld voor de eventuele schade. Wanneer u seccuur te werk gaat, zal de kans op defecten nihil zijn.

Leg de computer met de onderzijde naar boven op uw werktafel. Verwijder de rubberen voetjes en schroef de kruiskopschroefjes los. De plastic bodemplaat zit nu los. Til hem op en u ziet nu de print en het koellichaam van de spanningsregelaar zitten. Schroef hierna de twee andere kruiskopschroefjes los. De print zit nu alleen nog vast met de aansluitingen voor het toetsenbord. Laat deze verbinding zitten en draai de print VOORZICHTIG!!! om. U ziet nu de print aan de componentzijde. Vlak bij de TV-modulator (dit is het zilverenkleurig doosje op de printplaat) zit een condensator C12 (dit staat ook op de print). Het is de bedoeling dat deze condensator wordt kortgesloten. Dit gaat het beste door er een draadje overheen te solderen. Wanneer dit gedaan is kunt u de kast weer voorzichtig dichtbouwen en is de hardware-aanpassing klaar. U kunt nu met uw computer basicode-2 programma's maken en toch ook nog Sinclair programma's saven.

Handelwijze.

Laad de inlees- en vertaal routine dmv. LOAD"", zodra die geladen zijn presenteert de routine zich en geeft de "fout"melding 7/9100 (dit is goed). Verder moet u aan de foutmeldingen van de ZX-81 niet te veel waarde hechten. Uw ZX-81 staat nu in de FAST-mode, dit is noodzakelijk voor de inleesroutine.

U kunt nu de cassette met het van de radio opgenomen BASICODE programma in de recorder plaatsen en positioneren op de monotone fluittoon aan het begin van het programma. Om de inlees routine te starten tikt u in:

- RUN 600

Start de cassette-recorder en druk op N/L. Het inlezen stopt nu indien U op BREAK drukt, het geheugen vol is, of er een ETX- (End of Text)character ingelezen wordt. Tik in: LIST 999, op die regel zie U nu in ieder geval een REM, wat daarachter staat is ingelezen.

Waarschijnlijk zal de eerste keer het volume niet direct goed gestaan hebben, hetgeen zal blijken uit een REM-regel met onzin en fouten in de statements (PRINT bv. als PHINT). Stel dan het volume bij en probeer het nog eens. U hoeft niet steeds de band terug te spoelen, u mag ook halverwege beginnen. Is het programma zo goed mogelijk ingelezen, dan kunt u het gaan vertalen. Tik daartoe in:

-RUN 700

Het vertaal-programma zal nu iedere regel die met een REM begint gaan onderzoeken, de statements ed. door tokens vervangen, zodra hij een tweede statement op dezelfde regel tegenkomt een nieuwe regel maken, zonodig zelf een ontbrekende LET of GOTO statement invoegen en uiteindelijk het REM statement aan het begin van de regel verwijderen. Het volgende valt dus te verwachten:

```
1210 REM A=400 : B=2      1210 LET A=400
                          1211 LET B=2
```

februari 1985

Jack Raats

BASICODE VERTAAL-PROGRAMMA.

voor de SPECTRUM 48 K.

Er zijn twee versies: versie 2.2 zoals uitgezonden op 12 december 1984 en 2.2m (zie bijlage '42 Kolommen met microdrive').

Als men het programma laadt dan wordt automatisch de machinecode voor het inlezen, vertalen en de codes om 42/51 kolommen op het scherm te krijgen geladen.

Om een copie van het geheel te krijgen hoeft men alleen het programma te stoppen d.m.v. BREAK nadat alles is geladen en daarna RUN 890 te laten uitvoeren.

Om in het menu te komen kunt U elke keer als U het programma heeft onderbroken GOTO 810 intypen. Normaal komt U echter vanzelf in het menu.

Dit menu bestaat uit 7 gedeelten:

- 0 een read tape routine
- 1 een inlees routine
- 2 een list routine
- 3 een vertaal routine
- 4 een routine om automatisch de variabelen aan te passen
- 5 een routine om een variabele naam aan te passen
- 6 een DELETE routine

0 De read tape routine.

Deze routine print direct op het scherm wat er zou worden ingelezen. Er wordt niet daadwerkelijk geladen. Deze routine kunt U gebruiken om te zien op welke plaats het bandje zit maar ook om te zien of er geen onzin geladen wordt. Door het SCROLLen is het begin van een nieuwe regel meestal verminkt.

1 De inleesroutine

De inleesroutine verwijdert eerst de regels 900 t/m 919 en de regels boven de 1000. Daarna wordt het BASICODE programma ingelezen en in een REMregel opgeslagen, en wel op regel 900. Als de inleesroutine wacht op de begintoon is de BORDER paars-groen, als de inleesroutine inleest is de BORDER wit/zwart. Na het inlezen verschijnt op het scherm een van de volgende 'fout-meldingen':

- O.k. checksum=0
- break gedurende laden
- programma te groot
- tape loading error

2 De LIST routine

Met deze routine kan het originele BASICODE programma gelIST worden. Characters kleiner dan een spatie worden als een zwart blokje (CHR\$ 143) afgedrukt.

3 De vertaal- en renumber-routine

De vertaal routine. Voor het vertalen worden U drie vragen gesteld. Door de originele subroutines te laten neerzetten heeft U IN\$ en SR\$ etc weer terug. De andere twee vragen gaan over geheugen-gebruik. VAL "getal" bespaart 3 bytes per getal in de DATA statements. FN M\$ en FN R\$ besparen 9 bytes of meer per functie.

Bedenk wel dat de snelheid tijdens het runnen langzamer is. De vertaal-routine vertaalt de eerste regel die ligt tussen 900 en 919. Elke regel wordt in SPECTRUM-BASIC omgezet, op de juiste plaats in het programma gezet, en het stukje wat vertaald werd wordt verwijderd. Als er commando's met kleine letters (bijv. if) voorkomen dan worden die ook herkend. Er worden tijdens het vertalen enkele zaken aangepast:

- END wordt STOP:STOP:STOP:
- RIGHT\$ en MID\$ kunnen op twee manieren vertaald worden, als slicers met TO of als FN R\$ en FN M\$. LEFT\$ wordt altijd met slicer TO vertaald.
- ON a GOTO etc deze opdracht wordt ook aangepast. In verband met het renumberen moet elke GOTO/GOSUB een aparte statement worden. Zie tevens bijlage.
- DIM. Omdat bij de BASICODE de indices van array's bij nul beginnen wordt bij elke index '+1' toegevoegd. Hierdoor gaat FOR I=0 TO 12:READ A(I):NEXT altijd goed.
- Een speciaal geval zijn de gedimensioneerde string array's. Als in BASICODE de volgende uitdrukking voorkomt:
- DIM A\$(10) dan declareert men 11 strings waarbij elke string string een variabele lengte van 0 tot 255 kan hebben. Bij de SPECTRUM declareert men dan 1 string van lengte 11. Er moet dus een extra dimensie toegevoegd worden. Als extra dimensie wordt TO genomen. Dit geeft een foutmelding bij de DIM statements.
Men moet alleen in de DIM-statements "TO" weghalen en vervangen door een getal waarvan men de grootte moet gokken. Zie tevens bijlage.

Het renumberen. De renumber routine renumbert alle regels boven 1000 opnieuw met een stapgrootte van 10. Alle GOTO's, GOSUB's, SAVE...LINE's, LIST, LLIST en RESTORE's worden indien nodig aangepast. Als de foutmelding 'Subscript wrong' verschijnt dan is de stapgrootte te groot. Men kan op adres 62453 een kleinere stapgrootte POKEN. Men kan zelfs GOTO statements voordat er vertaald wordt in het programma zetten. Deze wijzen na het renumberen naar het nieuwe regelnummer. Er staan al een aantal op regel 790 e.v.

4 Aanpassen variabelen-namen

Met deze routine worden alle variabelen met 2 letters die niet in SPECTRUM-BASIC mogen voorkomen omgezet in namen met één letter. Men kan een lijst van de oude en de nieuwe namen naar de printer of beeldscherm sturen. Als er als nieuwe naam een sterretje verschijnt dan betekent dat dat die naam niet aangepast kan worden omdat de namen op zijn. men zal dan zelf iets moeten verzinnen om dat op te lossen. Bij een aanpassing kunnen de volgende vijf veranderingen verschijnen:

A\$=n betekent: er komen in het programma zowel een string A als een string-array A voor. De string-array behoudt zijn eigen naam en alle strings A krijgen de naam n.

F1 =x betekent de FOR..NEXT variabele F1 krijgt de naam x.

A2(=y betekent het numerieke array A2 krijgt de naam y.

S1\$=z betekent de simpele string S1 krijgt de naam z.

SA\$(=t betekent het string array SA\$ krijgt de naam t.

N.B. De namen mogen zowel in hoofdletters als in kleine letters staan.

5 Een variabele aanpassen

In plaats van alle variabelen automatisch aan te laten passen kunt U ook één op te geven variabele naam veranderen in een variabelen-naam met één letter. U krijgt op het scherm alle niet gebruikte strings, array's en FOR..NEXT variabelen.

6 De DELETE optie.

Als het programma vertaald en hernummerd is en alle variabelen zijn aangepast, dan heeft men het menu-programma niet meer nodig. Door DELETE te kiezen worden de regels 800 t/m 919 verwijderd.

U kunt door RUN 700 Basic & 42 koloms routine SAVEn en VERIFYen.

De 42/51 koloms routine.

Speciaal voor de SPECTRUM wordt 'n routine meegeleverd die 42 of 51 characters per regel op het beeldscherm afdrukt. Om deze routine te initialiseren moet men het volgende doen:

-RANDOMIZE getal dit getal geeft 't kanaal aan waarlangs men wil printen. Normaal is kanaal 2 de PRINT-routine en kanaal 3 de LPRINT routine. RANDOMIZE 2 zorgt er dus voor dat alles wat met PRINT wordt afgedrukt naar de 42 koloms-routine gaat
 -RANDOMIZE USR 6E4 nu wordt kanaal 2 naar de 42 kolomr-routine geopend. Normaal zijn er 3 kanalen maar door deze routine komen er 5 kanalen en het begin van het Basic programma is hierdoor niet op adres 23755 maar op adres 23765 (zonder microdrive). Op het moment van het maken van dit epistel weet ik niet of die 42 koloms routine compatibel is met de interface 1 plus microdrive (zie bijlage).

Men kan speciale characters naar deze print routine sturen. Deze zijn, behalve de characters voor de kleuren en print posities op het scherm:

CHR\$ (1)	print 42 characters op 1 regel
CHR\$ (2)	print 51 characters op 1 regel
CHR\$ (0)+CHR(I)	print op de eerste (I+1) regels
CHR\$ (30)	vraag indien nodig voor scroll
CHR\$ (31)	vraag niet om scroll

N.B. INPUT werkt niet om de scroll-counter te resetten.

Om de normale 32 characters per regel weer terug te krijgen hoeft men alleen maar CLOSE £2 in te typen. Op regel 111 en 121 staan de BASICODE routines voor 32 kolommen op het scherm. Als men deze routine in het geheel niet wil dan moet men ook regel 890 tot 899 aanpassen:

Verwijder regels 892 897 898 en de laatste "VERIFY" "CODE in r.893
 In regel 895 CLEAR aanpassen en laatste LOAD "CODE verwijderen.

Als men deze routine wil gebruiken zonder BASICODE moet men het volgende doen:

-Zorg dat de RAMTOP lager is dan 60000 en dat geen machinecode wordt gebruikt op adres 60000 t/m 61611;
 -LOAD "CODE de bytes voor de 42 kolommen per regel;
 -RANDOMIZE 2:RANDOMIZE USR 6e4;
 -PRINT CHR\$(1);CHR\$(0);CHR\$(21);CHR\$(30);: REM PRINT PER REGEL
 42 CHARACTERS;PER SCHERM 22 REGELS;EN VRAAG OM SCROLL

Bijlagen

bij de gebruiksaanwijzing van Ament's BASICODE-routine

1. Geheugen vrijmaken.

- 1 CLEAR 59999 subroutines kunnen niet meer vervangen worden
 2 CLOSE#2: CLEAR 61729 geen 42 kolommen op het scherm

2. Machine-code routines

Adres	functie
60000	initialisatie 42 kolommen
61730	read tape zonder laden
61755	PRINT vrije variabelen
61849	verander alle variabelen automatisch
62330	verander 1 variabele (POKE's zijn hiervoor nodig!)
62459	renumber alle variabelen moeten gewist zijn (CLEAR)
63000	lees BASICODE in
63284	LIST origineel
63373	vertaal zonder FN M\$ en FN R\$
63375	vertaal met FN M\$ en FN R\$
63377	DELETE eerste regel tussen 900 en 919
63380	DELETE regel 1000 en hoger
63383	DELETE regel 900 t/m 919
63386	DELETE regel 800 t/m 919
63389	zet regel 1 t/m 799 op adres RAMTOP en lager
63392	zet originele subroutines neer

RUN 700 - SAVE basic + 42 koloms routine

RUN 810 - enter menu programma

RUN 890 - SAVE basic en alle machine-code routines.

3. ON...GOTO/COSUB

Voorbeeld: BASICODE-programma, vertaald en henummerd

1030 PRINT:ON A COSUB	1030 PRINT
1100,1110,1300:	1040 IF INT (A+.5)=1 THEN GOSUB 1100
PRINT"KLAAR";	1050 IF INT (A+.5)=2 THEN GOSUB 1110
	1060 IF INT (A+.5)=3 THEN GOSUB 1300
	1070 PRINT "KLAAR";

Indien er als eerste statement 'IF X=2 THEN' i.p.v. PRINT had bestaan dan was nu regel 1030: IF X=2 THEN. Dat moet worden veranderd in: 1030 IF NOT (X=2) THEN GOTO 1080

N.B. Als A=1 en in SUBROUTINE 1100 wordt A=2 dan zal bij het RUN-nen van het programma SUBROUTINE 1110 ook aangeroepen worden, terwijl dat eigenlijk NIET moest!

4. Gedimensioneerde array's

	BASICODE		Onaangepast		Verbeterd	
	DIM A\$(2) A\$(2)="PIET"		DIM A\$(2+1) LET A\$(2+1)="PIET"		DIM A\$(2+1,5) LET A\$(2+1,TO) ="PIET"	
index	inhoud	lengte	inhoud	lengte	inhoud	lengte
0	""	0	" "	1	" "	5
1	""	0	" "	1	" "	5
2	"PIET"	4	"P"	1	"PIET "	5

Voor TO is 5 gekozen. Bekijk of het klopt!

5. FOR .. NEXT-LOOP pauzes

Vaak wordt in BASICODE programma's lege FOR..NEXT loops als pauze functie gebruikt, bijv FOR I=1 TO 2000:NEXT I. Deze duren op een SPECTRUM vaak veel langer dan bedoeld, en U zit maar te wachten. BREAK in het programma als U denkt dat dit het geval is en pas de bovenlimiet aan. Meestal zal men het met een factor 10 moeten verkleinen, maar dat is even uitproberen. Ik hoop dat u hiermee met succes programma's die u via Hobbyscoop of vrienden of vriendinnen krijgt kunt omzetten voor uw SPECTRUM.

6. 42 Kolommen met microdrive

(Op Uw bandje komt dit als tweede programma voor als "BASICODE 2.2m" met alle wijzigingen die hieronder staan !)
Als de 42 koloms routine problemen geeft met de microdrive kunt U na het BASIC gedeelte BREAKen als volgt wijzigen:

Voeg toe de regels (als U geen printer hebt m.u.v. regels 50-53)

```
50 LET ool=PEEK 23631+256*PEEK 23632+15
51 FOR o=0 TO 4:LET oo2=PEEK (ool+o)
52 POKE ool+o,PEEK (65300+o): POKE 65300+o,oo2
53 NEXT o
54 RETURN
351 GOTO 50
361 GOTO 50
```

en verander de volgende regels als volgt

```
350 REM:GOSUB 50:LPRINT SR$;
360 GOSUB 50:LPRINT
710 CLEAR 59999:GOSUB 50:LOAD ""CODE : CLOSE#2:
RANDOMISE 2:RANDOMIZE USR 6E4:PRINT CHR$ 1;
CHR$ 0;CHR$ 23;: RUN: REM BEGIN
879 GOSUB 50:LPRINT "'OUDE/NIEUWE VARIABLEN:'":
LPRINT STR$ USR 61849 AND NOT PI :GOSUB 50:RETURN
883 GOSUB 50:LPRINT USR 63284:GOSUB 50:RETURN
896 RANDOMIZE USR VAL"63389": POKE 60005,0:
POKE 60029,15: POKE 60060,16:GOSUB 50
```

voer vervolgens de volgende opdracht uit

```
RESTORE 350:POKE PEEK 23639+256*PEEK 23640+5,32
```

RUN 895 en start de tape.

Nadat alles geladen is kan men door RUN 890 alles weer SAVEn. De LPRINT, PRINT, LIST en LLIST routines sturen nu de te printen characters óf naar de printer óf naar het scherm.

Door GOSUB 50 'verwisselt U beurtelings printer en beeldscherm.

---oooOooo---

ZX-80-EENENTWINTIGEN (1 K RAM, 4 K ROM)

```
10 LET A=0
30 CLS
40 LET D=0
50 LET C=0
60 PRINT "BET = ? "
70 INPUT E
80 CLS
90 PRINT "HIT=N/L-----";
100 PRINT "STAND=1","DOUBLE=2"
110 GO SUB 920
120 PRINT "YOUR CARDS:"
130 PRINT
140 LET F=RND(11)
150 PRINT "<";F;">",
155 LET D=D+F
160 IF D>21 THEN GO TO 400
170 INPUT A$ (hier keuze be-
      kend maken)
180 IF A$ = "1" THEN GO TO 210
190 IF A$ = "2" THEN LET E=E*2
200 GO TO 140
210 PRINT
220 PRINT
230 PRINT
240 PRINT
250 PRINT
270 PRINT "MY CARDS:"
280 PRINT
290 LET G=RND(11)
300 PRINT "<";"G";">",
310 INPUT B$
320 LET C=C+G
330 IF C>21 THEN GO TO 500
340 IF C>16 THEN GO TO 540
350 GO TO 290
400 GO SUB 900
410 PRINT,"--YOU LOSE"
415 LET A=A-E
420 PRINT "-----POINTS IN GA
      ME:";A
425 GO SUB 910
430 INPUT B$
440 GO TO 30
500 GO SUB 910
510 PRINT,"---YOU WIN"
520 LET A=A+E
530 GO TO 420
540 GO SUB 900
550 LET V=ABS(C D)
560 IF V=1 THEN GO TO 410
570 IF C=D THEN GO TO 590
580 GO TO 510
590 PRINT,"-----DRAW"
600 GO TO 420
900 PRINT
910 PRINT
920 FOR H=1 TO 32
930 PRINT CHR$(3);
940 NEXT H
950 PRINT
960 PRINT
970 RETURN
```

TALEN

Inhoud van deze rubriek in dit nummer.

- 07.6.4.001 07-48 Tiny Pascal Implementatie (Breedendeek)
07.6.4.003 07-50 Geheugenindeling van Tiny Pascal(Breedendeek)

Eerder gepubliceerd in de rubriek 6 TALEN

- 01.6.1.001 01-47 Hoe laden we machinecode in de ZX-80/81(Visser)
02.6.1.006 02-47 Het laden van machinecode en het uitvoeren van
testroutines (ZX-80; Radius)
04.6.2.001 04-46 Werken met DEF FN en FN (Arts)
04.6.2.002 04-47 CALL RELATIVE (Verhoeven)

ZX SPECTRUM TINY PASCAL VERSIE 1.0 IMPLEMENTATIEInleiding:

Dit is de Spectrum-implementatie van de NOS/TELEAC Tiny-Pascal compiler. De versie van de compiler die op de Tiny-Pascal cassette stond was gemaakt voor een Exidy Sorcerer computer en moest aanzienlijk worden aangepast om op de Spectrum te kunnen draaien. Dit werk hebben we echter voor u gedaan en u kunt dus nadat u deze Spectrum-versie hebt geladen meteen beginnen met programmeren in Tiny-Pascal. De voorbeeldprogramma's op de NOS/TELEAC cassette kunt u ook inladen en draaien op deze versie.

Tiny-Pascal op de cassette bestaat uit twee delen: een kort BASIC-programma dat de LOAD- en SAVE-routines bevat en een 22K lang machinecode-deel dat de eigenlijke compiler inclusief editor, operating system e.d. bevat. Als deze twee blokken ingeladen zijn wordt het scherm gewist en verschijnt enkele sekonden lang een copyright-melding, gevolgd door de melding: "TINY PASCAL OPERATING SYSTEM". Dit operating system bestuurt in feite het hele programma, vanuit het operating system kunt u naar de diverse andere delen springen zoals de Editor en de compiler zelf.

Dit springen naar een bepaald gedeelte wordt gedaan door een bepaalde toets in te drukken, bv. E voor editor. Indien u een toets indrukt welke geen commando kent geeft het operating system aan welke commando's het wel kent.

Omdat u nu eerst enige Pascal-tekst moet maken voor de compiler moet u naar de editor, druk dus op "E". Er verschijnt nu de melding "EDITOR" gevolgd door de cursor-prompt.

De werking van het toetsenbord van de Spectrum is nu iets anders dan u normaal in BASIC gewend bent. Alle toetsen geven alleen letters, ook indien ze met SYMBOL SHIFT gebruikt worden. Bijvoorbeeld SYMBOL SHIFT-G geeft " " en niet "THEN". Verder zijn de E- en G-modes niet aanwezig. Wel kunt u op de gebruikelijke manier met CAPS SHIFT-2 de hoofdletters vast zetten, de cursor verandert daarbij van een "L" in een "C".

Indien u grafische karakters wilt weergeven in een Pascal-programma moet u daarvoor de betreffende codes gebruiken. Met de controlecodes 16 t/m 23 van de Spectrum kunt u INK, PAPER, FLASH, BRIGHT, OVER, INVERSE, AT en TAB simuleren, zie hiervoor het Spectrum handboek. Het scherm wissen gebeurt met WRITE(12); maar de cursor wordt echter niet linksboven gezet.

OPMERKING: Voor de tekens "<=", ">=" en "<>" moet u niet de enkele toetsen Q, W en E gebruiken omdat de symbolen dan niet herkend worden door de compiler. U moet deze symbolen intikken als twee aparte tekens, dus ">=" als ">" gevolgd door "=".

Nu u in de editor bent kunt u tekst intypen of laden. Intypen doet u door een regelnummer te typen gevolgd door de Pascal-tekst. Hierbij kunt u de DELETE-toets normaal gebruiken. CAPS SHIFT-8 geeft twee spaties voor het inspringen aan het begin van de regel. Dit kunt u veranderen door "Options" te kiezen en vervolgens het nieuwe aantal te printen spaties te geven.

EDITen van een regel is niet mogelijk, een foute regel zult u dus opnieuw in moeten typen.

De LIST-optie van de editor werkt in deze Tiny-Pascal versie iets anders dan in de oorspronkelijke versie. In deze versie zal nadat 12 regels geLIST zijn de listing pauzeren en verdergaan na een toetsdruk, tenzij hierbij CAPS SHIFT-1 werd ingedrukt. In het laatste geval stopt het listen en keert de editor terug naar zijn normale prompt.

Indien u L100 intypt zal de tekstfile vanaf regel 100 gelist worden, dus niet zoals oorspronkelijk alleen regel 100. Wilt u dit laatste toch, typ dan in L100,100 (of L100a100 enz.).

Als u twee getallen na L intypt gescheiden door een letter of ander teken zullen de regels tussen het eerste en het tweede regelnummer gelist worden.

Het LOADen en SAVEn van Pascal-tekst naar tape kunt u op twee manieren doen. De eerste manier maakt gebruik van het normale Spectrum cassetteformaat en werkt met de editor-commando's R voor lezen en W voor schrijven. Als u deze commando's intikt wordt een filenaam gevraagd die max. 10 karakters lang mag zijn. Bij het LOADen van de band mag u hierbij ENTER intikken, dan wordt de eerstvolgende file geladen.

Deze commando's werken via BASIC-statements, u kunt ze daarom aanpassen voor de Microdrive door de LOAD- en SAVE-statements in het BASIC-programma aan te passen. Tevens dienen dan de regels 30 en 40 verwisseld te worden, aangezien het nu niet mogelijk is om bij LOADen een "lege" naam op te geven. Hierna kunt u het aangepaste programma SAVEn met RUN 80.

U kunt echter ook uw programma's in BASICODE wegschrijven en weer inlezen. Hierdoor kunnen bezitters van Tiny-Pascal met een andere computer uw programma's ook inladen en draaien. Ook kunt u hiermee de voorbeeldprogramma's op de NOS/TELEAC cassette inladen en draaien. U kunt deze optie kiezen door in de editor "B" in te toetsen. Vervolgens krijgt u de vraag "Read (R) or Write (W)" voor LOADen resp. SAVEn in BASICODE.

Zowel bij het LOADen als SAVEn kunt u op BREAK drukken, er verschijnt dan de melding "BREAK pressed". Tijdens laden is het mogelijk dat door bv. een drop-out een of meer karakters fout worden ingelezen, er verschijnt dan de melding "Checksum ERROR". Indien u om een of andere reden in Basic bent teruggekomen (bijv. door een "Tape loading error") kunt u Tiny-Pascal d.m.v. RUN weer opstarten, u komt dan terug in de editor zonder dat de Pascal-tekst wordt gewist.

Voor verdere bijzonderheden over de Editor verwijzen we naar de informatie op de Pascal-cassette.

Als u een Pascal tekst in de Editor hebt kunt u hem vertalen naar een 8080/Z80 machinetaalprogramma. Daartoe moet u in de editor X typen gevolgd door ENTER, u komt nu weer terug in het operating system. Van daaruit moet u "T" toetsen voor "translate". Er verschijnt nu een copyrightmelding van de compiler gevolgd door de vraag "Want code printed (Y/N)". Als u "Y" intoetst zal de compiler na elke gecompileerde regel de P-code gegenereerd door die regel uitprinten (de P-code is een soort tussenvorm tussen de Pascal-brontekst en de 8080 machine-code). Toetst u "N" dan zal alleen de listing van de Pascaltekst op het scherm voorbijkomen bij het compileren.

DE NIEUWE GEHEUGENINDELING VAN TINY-PASCAL.

Omdat in de oorspronkelijke versie van Tiny-Pascal geheugen-adressen werden gebruikt die op de Spectrum niet vrij zijn zijn er een aantal wijzigingen gemaakt in de geheugenindeling van het programma.

De nieuwe indeling is als volgt (alle adressen in hex):

Startadres tekstfile : BE00
P-code startadres : E000
Objectcode startadres: F000
Stack startadres : E000
Stack eindadres : EFFF

De laatste vier waarden zijn de default-waarden. Het is mogelijk om in het Operating System de "C"-optie te kiezen i.p.v. de "T"-optie. In dat geval moeten de adressen voor de runtime module en de vier laatstgenoemde adressen vanaf het toetsenbord worden ingevoerd. Indien in zo'n geval ENTER wordt getoetst wordt de betreffende default-waarde genomen. Deze optie dient alleen gekozen te worden als men bijvoorbeeld de objectcode op een andere plaats in het geheugen wil plaatsen.

Verder zijn er nog een aantal wijzigingen in de diverse routines van het Tiny-Pascal programma:

6303-7CFF Editor
7D00-7EFF Basicode stuur- en leesroutine
7F00-7FFF Keyboard routine
9E63-9F09 Basicode schrijfroutine.

Ik hoop dat u veel programmeerplezier zult hebben met deze Pascal-versie.

Jan BREDENBEEK
Hilversum

(vervolg van TINY PASCAL IMPLEMENTATIE van pag.49)

Indien de compiler een fout vindt in de Pascaltekst verschijnt een foutmelding in woorden op het scherm waarbij het foute karakter wordt aangegeven met een pijltje. Na het indrukken van een toets komt u dan weer terug in het Operating System.

Als de tekst zonder fouten gecompileerd is zal onmiddellijk worden doorgegaan met het vertalen van de P-code naar 8080 code. Hierna komt u weer terug in het Operating System, van waaruit u het programma met "R" kunt runnen.

Als u in het Operating System de "P"-optie kiest zal alle verdere output naar de ZX-printer gaan. Optie "S" schakelt het scherm weer in.

Jan BREDENBEEK
Hilversum

SPECTRUM WAFADRIE

is gelijk aan 2 st. Micro Drives
+ 1 st. Micro Interface
+ 1 st. Centronics Interface
+ 1 st. RS-232 Interface
+ 1 st. Expansion Bus voor Joy Sticks
enz.
en dat alles voor **f 575,-**

U krijgt daarvoor totaal 256K bytes opslagcapaciteit. U werkt gewoon met BASIC programma's, machinetaal en databestanden groter dan het SPECTRUM geheugen.

Voor de cassette drives zijn z.g. Wafers (eindeloze cassettebandjes) leverbaar van 16K tot 128K. De WAFADRIE gedraagt zich als een dubbele Floppy Disk met 128Kb (geform.) per drive en wordt aangesproken met 25 nieuwe DOS-achtige BASIC commands.

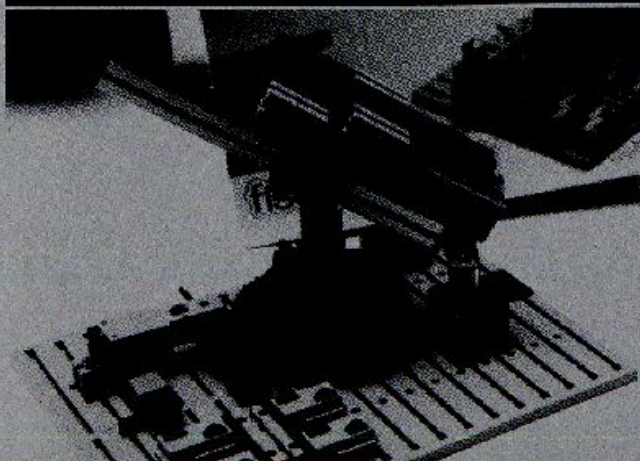
In één woord: **SUBLIEM**

MicroSales c.v.

Forellendaal 352
2853 LN Den Haag
Tel. 070 - 97 71 69

MICRO@SOURCE

NIET ALLE INCLUSIEF IN GEDRUKTE SINCLAIR SPEC. KUNNEN SPECIAALZAKEN
OPNIMINGEN VAN DE WERK. TEL. 070 12 16 11 24 73 17 11
DO AVOND 19 21 11 OMSCHRIJVINGEN, POSTBUS 1243, 8501 BG ZWOLLE
TEL. 039-223498. GEE SPECIAAL SEEDR. IN POSTORDER SERVICE.



FISCHER TECHNIK COMPUTING DOOS f 239,50.
SPECTRUM INTERFACE VOOR FISCHER COMPUTING f 195,-.

MICRO@SYSTEEM

DOORGEMETALISEERDE PRINT EN BOEKJE

MOEDERKAART f 35,-
BUSVERLENGKAART f 30,-
2 KANAALS SERIEEL INTERFACE f 30,-
2 KANAALS PARALLEL INTERFACE (NIET DOORGEMET) f 25,-

GRATIS INFORMATIE

VRAAG ONZE UITVOERIGE DOCUMENTATIE OVER DE
SPECTRUM, RANDAPPARATUUR, SOFTWARE EN BOEKEN

VIDI@SOURCE

VIDITELPAKKET VOOR ZX SPECTRUM 48 K.
NU VOOR INTERFACE EEN INCLUSIEF TELESOFTWARE

VIDITEL EEN **f 99,-** EEN VIDITEL
MODIEMKABEL f 50,-

INCLUSIEF KABEL f 149,-

INCL. UNIVERSIT. RS-232 INTERFACE, VIDITEL KABEL, SOFTWARE VOOR
VIDITEL, LIST EN PRINT, EN TERMINAL GEBRUIK f 295,-

NIET EEN FOLDER

WAT IS MSX?

MSX COMPUTERS, RANDAPPARATUUR EN SOFTWARE NIET LEVERBAAR.
KOM ZIEN MSX MACHINETAALPAKKET f 95,-

AANBIEDINGEN

BETA DISK INTERFACE DOUBLE DENSITY f 550,-
INCL. MITSUBISHI 80 TRACKS DEVD'S DRIVE (640 K NETTO), KAST, KABEL
EN VOEDING VOOR 2 DRIVES f 1700,-
CURRAH MICROSTITCH f 125,-
GEHEUGEN UITVERHIDING VOOR SPECTRUM 16 K f 118,-
INTELLIGENT JOYSTICK INTERFACE f 129,-

MICRO@SOURCE

BANK: ABN 59 52 44 948, POSTGIRO 36 27 209, VIDITEL PAC. # 6170400 #
VERZENDEN: PTT BREF 5,-; PAK. f 4,50, REMBOURS 10,-
AL ONZE PRIJZEN ZIJN VRIJBLIJVEND INCL. BTW EXCL. VERZENDKOSTEN

DATA@LINK

TEL. 038-223345.

INFORMATIE, PRIJKORD- EN TELESOFTWARESYSTEEM
MET ELKE VIDITEL TERMINAL KUNT U VRIJBLIJVEND 5 MINUTEN IN
DATA@LINK RONDKIJKEN!

EEN SERVICE VAN MICRO-SOURCE, ONDERGEBRACHT BIJ
STICHTING PARAPLUI, POSTBUS 1163, 8001 BD ZWOLLE.
TEL. DATA@LINK 038-223345, INFORMATIE: 038-221698.
LIDMAATSCHAP f 35,- PER JAAR VRAAG ONZE INFORMATIE.

Stichting IMPULS
Postbus 212
1740 AE SCHAGEN

PORT
BETALD
SCHAGEN