

SINCLAIR

M P U L S



Nummer 3

7.50

INHOUDSOPGAVE IMPULS NUMMER 3

1. ALGEMEEN

1.1	Redactioneel	03-04
1.3	Verwarring over het gebruik van RAMTOP	03-05
5.2	Merge-routine	03-06
1.3	Het gedrag van String-variabelen bij de ZX-81	03-07
1.7	Boekbespreking	03-11

2. HARDWARE

2.1	Schema Spectrum	03-14
2.3	Zelfbouw geheugenuitbreiding (RAM/ROM)	03-16

5. SOFTWARE

5.2	Renumber	03-20
5.2	FAST-SAVE-LOAD 2400 BAUD	03-21
5.3	ZX en de centjes	03-26
5.3	BASICODE-2-protocol	03-27
5.4	Game of Life op z'n Booleaans	03-32
5.4	12 jockeys, deel I	03-37
5.4	Rekenoefeningen voor het jonge volkje	03-41
5.4	Yahtzee Scorekaart	03-44
5.9	Beschrijving van de IMPULSOFT-cassettes 1, 2 en 3	03-45
1.1	Wat is de SINCLAIR gebruikersgroep	03-52

ALGEMEEN

INHOUD

Redactioneel	03.1.1.007	03-04
Verwarring over het gebruik van RAMTOP	03.1.3.017	03-05
Merge-routine	03.5.2.	03-06
Het gedrag van stringvariabelen bij de ZX-81	03.1.3.019	03-07
Boekbespreking: "Voor galg en rad"	03.1.7.001	03-11

03-21 e.v.

sinclair impuls

03.5.2.

CORRECTIES OP 2400 BAUD FAST-SAVE-LOAD-ROUTINE

Helaas zijn enkele zeer storende foutjes door de correctieronde van dit artikel geslopen. Aan het eind van het artikel, op blz. 23, drie regels na het kopje VERIFY moet U lezen "Na PRINT USR 30465 ..."

Op de volgende bladzijde, in de listing van de LOAD deelroutine staat een regel die moet luiden 76E0 72 LD (HL),D (haakjes om HL i.p.v. om de D). De op twee na laatste regel van diezelfde deelroutine is de instructie moeilijk leesbaar. Daar moet staan 76FA 38EA JR C, SIGNAL.

In de VERIFY deelroutine tenslotte drie belangrijke kleine correcties: Vierde regel van de listing moet luiden: 76F7 38F8 JRC, TAPE. Een soortgelijke fout 16 regels lager: lees 7717 38C8 JRC, L/S UPDATE.. Minder essentieel maar toch fout is het een regel lager: 7719 BE CP (HL).

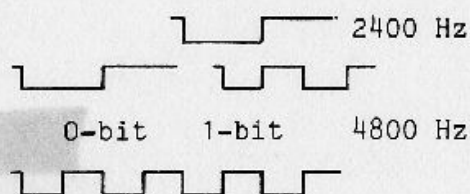
Met de excuses van de redactie, maar zó moet het wél werken.

COMPUTERVRIENDEN!

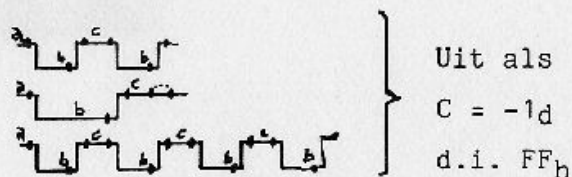
Bij dit derde nummer van IMPULS moeten helaas weer een paar foutjes uit het verleden worden verbeterd. Allereerst de codering: pag. 02-22 was gecodeerd 02.3.2.003, maar wij waren in die groep pas aan pag. 002 toe. Dat die bladzijde wat ongelukkig was geplaatst t.o.v. een bladzijde uit een andere groep kon helaas niet worden vermeden. Wèl had het programma Wisselkoersen van Ivo Breeden op een andere plaats dan op pag 02-28 moeten komen, het hoort 02.5.5.007 thuis. Het ontbreekt helaas ook in de inhoudsopgaven.

Op pag. 02.25, in de onderste listing, staan ook twee kleine foutjes. Na adres 16396 moet een) komen en op diezelfde hoogte moet een E weg, daar moet XOR staan.

Uw overijverige redacteur heeft iets teveel getekend in de tekeningen op blz 02-34. Die worden hier verbeterd afgedrukt.



Op diezelfde bladzijde, onderaan, onder punt 4 is het derde woord 'wordt'. Dat woord dient vervangen te worden door de woorden 'van 1200 Hz'.



Op de volgende bladzijde, iets onder het midden, in de SAVE-routine, kunt U adres 16466 opzoeken. Daarachter moet staan '56 Ld D,(HL)'. En zes regels lager, in het commentaar, staat 188 waar het 288 states had moeten zijn. In de grote LOAD-routine op pag 02-37 tenslotte ziet U op adres 16602 'F&' staan. Dat kan niet, dat moet F7 zijn.

Daarmee laten wij enkele minder essentiële typfouten graag aan U over. Tenslotte willen wij nog graag Uw aandacht erop vestigen dat de IMPULS-reeks eigenlijk een verzamelwerk is, die U compleet moet zien te maken (en te houden!). De voorgaande nummers (expres zeg ik niet 'oude') zijn nog te koop op de gebruikersdagen, bij Erik Visser in Schagen en in een aantal winkels die computer-boeken verkopen.

Goede artikelen voor IMPULS blijven nog steeds welkom, vooral in de categorieën Talen en Interfacing. Ook in de Hardware en Firmware categorieën is een achterstand in te halen. Dus zet U aan het werk en maak mede-Sinclair-gebruikers van Uw bevindingen deelachtig!

VERWARRING OVER GEBRUIK VAN RAMTOP

Over de consequenties ("gevaren") van het verlagen van RAMTOP d.m.v. POKE 16389,... (POKE 16388,... is meestal overbodig) is in mijn praatje van 3 september 1983 door kritische opmerkingen van een toehoorder verwarring ontstaan, ook bij mij. Daarom de feiten op een rijtje gezet.

In het ZX-81 handboek, pag. 177 (tweede uitgave, 1981) staat dat POKE gevolgd door NEW reserve ruimte boven de BASIC systeem area schept, terwijl POKE gevolgd door CLS gebruikt kan worden om een minimale display file (collapsed D-FILE) op te zetten. Dit laatste wordt ook bereikt door POKE te laten volgen door NEWLINE (N/L) omdat in de ROM dan o.a. de CLS routine wordt aangeroepen.

NEW wordt in de ROM gevolgd door de RAM-CHECK en INITIALISATION routines die o.a. de plaats van de machinestackpointer (SP) en GOSUB stackpointer (ERR-SP) bepalen en opslaan, (zie ook Handboek, p. 171). Na het inschakelen van de ZX-81 blijken de 18 adressen direkt onder RAMTOP te worden gebruikt door machinestack en GOSUB-stack. Op adres RAMTOP - 3 staat 6, op RAMTOP - 4 118 en het POKEn van andere waarden in die adressen leidt tot een crash. Het POKEn van andere adressen heeft geen blijvende gevolgen, behalve bij de eerste twee onder RAMTOP, die wel in waarde kunnen worden veranderd.

Door nu RAMTOP te verlagen beneden de initiële waarde (bij 16 k RAM is dat 128 voor 16389 en 0 voor 16388) gevolgd door NEW wordt de inhoud van de 18 bytes onder de RAMTOP verplaatst naar de 18 bytes onder de nieuwe RAMTOP en de oorspronkelijke 18 adressen kunnen nu zonder gevaar en blijvend worden gePOKEd. Voor de 18 adressen onder de nieuwe RAMTOP geldt wat ik hiervoor over de beginsituatie schreef.

Door de RAMTOP te verlagen, gevolgd door N/L wordt weliswaar de RAMTOP-pointer gezet op die nieuwe waarde, maar de "gevaarlijke" bytes blijven op de oorspronkelijke plaats. Met het op deze manier gebruiken van RAMTOP wordt wel het gebied boven RAMTOP beschermd tegen NEW en tegen het verdwijnen van de opgeslagen informatie (in machine-code of over-gePOKEd BASIC of MC-BASIC) bij het LOADen van een ander programma.

Verder kan op deze manier een collapsed D-FILE worden verkregen en wel door POKE 16389, 76 gevolgd door N/L. Iedere waarde boven 64 en onder 77 is overigens even bruikbaar. Het voordeel van deze handelwijze is dat de display file, die in geval van een beschikbare RAM 77*256 bytes opgevuld is met $33*24+1 = 793$ characters, gereduceerd wordt tot $24+1$ N/L characters, een besparing van 768 bytes, d.i. ca. 20 s SAVE

racters, een besparing van 768 bytes, d.i. ca. 20 s SAVE en LOAD tijd. Samenvattend: het is nuttig, voordelig en gevaarloos POKE 16389,76 gevolgd door N/L te gebruiken voor de beschreven doeleinden, gevaarloos zelfs nog als het BASIC-programma boven 7700_H=19712_D uitkomt. In dat laatste geval is het beter na het LOADen RAMTOP weer omhoog te POKEn, omdat anders hinderlijke traagheid bij programma wijzigingen en ook vertraging bij de uitvoering van programmaopdrachten optreedt. Het nut is dus bij grotere BASIC programma's betrekkelijk.

Wordt het omlaag POKEn van RAMTOP gevolgd door NEW, dan volgt alleen een crash, indien hetzij het BASICprogramma, hetzij machinecode-routines geschreven, gePOKEd of geLOAD worden in de twee adressen als eerder omschreven.

N.B. In de instructies van het FAST SAVE LOAD programma elders in deze IMPULS wordt POKE 16389, 76 gevolgd oor N/L effectief toegepast in het hulpprogramma. Ook als het te SAVEn programma groot is kan dit zonder het zoëven vermelde bezwaar van traagheid worden gedaan omdat na het LOADen van ieder programma RAMTOP automatisch op het maximum is gezet of op de bij het programma gewenste waarde. Alleen in het onwaarschijnlijke geval dat men het hulpprogramma 'vers' geschreven heeft (dus niet van TAPE ge LOAD) en onmiddellijk daaraan aansluitend ook een grote FSL te SAVEn programma schrijft is het raadzaam RAMTOP omhoog te POKEn.

H.H. van Abbe

Eveneens van H.H. van Abbe ontvingen wij een 'MERGE'-routine. Uit plaatsgebrek, maar omdat die toch zeer interessant is, plaatsen wij die op deze wat onlogische plaats.

"MERGE" BASIC VAN PROGR 1, MET
LAEE REGELNRS, IN PROGR 2
PROGR 1 WORDT OPGESLAGEN IN 707D
VERPLAATSGINGROUTINE IN REHSHT:
210D40 115070 012100 EDB0 C9
SAVE: POKE 16389,76 ^{N/L} GOTU 2 N/L

N 1 REH SLN RND)??'5 GOSUB STA
3 SAVE "MERGE"
3 RAND USR 16514

7050C	2A0C40	LD	HL,(400C)
7050E	117D40	LD	DE,407D
7050F	ED52	SBC	HL,DE
70510	225A70	LD	(705A) HL
70511	AF	XOR	A
70512	1502	JR	706C
70513	0E01	LD	A,01
70514	217D40	LD	HL,407D
70515	117D70	LD	HL,707D
70516	ED4B5A70	LD	BC,(7065A)
70517	A7	AND	A
70518	2801	JR	Z,707A
70519	EB	EX	DE,HL
7051A	EDB0	LDIR	
7051B	C9	RET	

OPERATION:
LET OP JUISTE NUMMERING PROGR 1
LEN PROGR 2
RAND USR 28764
BEPAL AANTAL BYTES MET PRINT
PEEK 28762+256*PEEK 28763
LOAD PROGR 2
ZORG VOOR JUISTE AANTAL BYTES
RUIMTE VOOR PROGR 1 OP EERSTE
REGENS PROGR 2; DENK AAN N/L CHRS
RAND USR 28778

HET GEDRAG VAN STRINGVARIABLEN BIJ DE ZX-81.

O.a. in het boekje "Byteing deeper into your ZX-81" staat een opsomming van de diverse typen variabelen. Deze zijn hier nog eens overzichtelijk bij elkaar gezet.

variabele	decimaal	binair
normale letter	code	001.
één-kar. numerieke variabele	code+64	011.
multikar.numerieke variabele	code+128	101.
numerieke array variabele	code+96	110.
control variabele (ROR/NEXT)	code+192	111.
gewone string variabele	code+32	010.
string-array variabele	code+160	110.
code voor 'einde variabelen'	128	1000 0000

Door op de eerste drie bits te testen zal de BASIC-interpreter ontdekken waar hij mee te maken heeft. Als het programma echter een stringvariabele en een ééndimensionale string van dezelfde naam bevat zal de machine de variabele alleen als array-naam herkennen en behandelen. Dit wordt geïllustreerd met het volgende kleine programma:

```

10 DIM B$(5)
20 FOR X=-4 TO 16
30 LET V= PEEK 16400 + 256*PEEK      (varspointer)
16401
40 PRINT PEEK(V+X)
50 NEXT X

```

Het RUN-commando geeft als resultaat: (zie tabel 1)

Nu voegen wij aan het programma de regel 5 LET B\$="AAAAAAA" toe. Het resultaat is weer tabel 1. Daaruit concludeer ik dat regel 10 het resultaat van regel 5 teniet doet.

Nu voegen wij bovengenoemde regel 5 in met als regelnummer 15. Het resultaat daarvan is in tabel 2 te zien. B\$ wordt kennelijk afgekapt tot 5 karakters en in het array gezet.

Veranderen wij nu regel 10 in: 10 DIM A\$(5) dan krijgen wij tabel 3 te zien. Het werkt normaal, de eerste de beste variabele komt bovenaan, de volgende eronder, enzovoort.

Laten wij nu eens het volgende programmatje proberen:


```

10 DIM B$(5)
15 LET B$="AAAAA"
16 FOR C=1 TO 6
20 FOR X=-4 TO 16
30 LET V= PEEK 16400 + 256*PEEK
16401
40 PRINT PEEK(V+X)
50 NEXT X
60 LET B$=B$(2 TO)
70 STOP
80 NEXT C

```

Dit programma middels het CONT-commando telkens een tabel laten printen:

	0	0	0	0	0	0	display file
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	118	118	118	118	118	118	Vars (eind)
	199	199	199	199	199	199	
totaal	8	8	8	8	8	8	B\$(5)
aant.bytes	0	0	0	0	0	0	
ant.DIM	1	1	1	1	1	1	
ant.kar's	5	5	5	5	5	5	
	0	0	0	0	0	0	
	38 A	38 A	38 A	38 A	38 A	0 f	(NB: f = spatie)
	38 A	38 A	38 A	38 A	0 f	0 f	
	38 A	38 A	38 A	0 f	0 f	0 f	
	38 A	38 A	0 f	0 f	0 f	0 f	
	38 A	0 f	0 f	0 f	0 f	0 f	
	232 c	232	232	232	232	232	naam
momentane	129	130	130	131	131	131	control- variabele
waarde	0	0	64	0	32	64	
	0 1	0 2	0 3	0 4	0 5	0 6	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
eind	131	131	131	131	131	131	C
waarde	64	64	64	64	64	64	
	0	0	0	0	0	0	
	0 6	0 6	0 6	0 6	0 6	0 6	
	0	0	0	0	0	0	
step	129	129	129	129	129	129	
= 1	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
lijn-nr+1	17	17	17	17	17	17	
	0	0	0	0	0	0	

Je kunt nu duidelijk zien, hoe de AAAAAA's naar boven toe uit de B\$(5) gedrukt worden. M.a.w. B\$(1) wordt in elke volgende stap verwijderd, de andere A's schuiven een plaats op en achter de A's wordt in B\$(5) een spatie toegevoegd. Ook kan men hiermee mooi zien hoe de controlvariabele C

genoteerd en veranderd wordt.

Als laatste een programma, zonder tevoren te dimensioneren, dus het programma van hierboven waaruit regel 10 is weggelaten. Het resultaat op het scherm verschilt in enkele opzichten van de vorige, maar het belangrijkste verschil is, dat in de tweede ronde al de gehele B\$ van het scherm is verdwenen. Dat komt doordat de string verandert door een LET-statement. Hij wordt dan van de oude plaats verwijderd en helemaal onderaan de lijst opnieuw toegevoegd..

Veranderen we nu regel 20 in: 20 FOR X=28 TO 52 dan komt de verloren zoon weer tevoorschijn:

x	0	0	0	0	0
x	0	0	0	0	0
x	71	71	71	71	71
x	4	3	2	1	0
x	0	0	0	0	0
x	38	38	38	38	128
x	38	38	38	128	143
x	38	38	128	143	8
x	38	128	143	8	204
x	128	143	8	206	0

128 en zo zou ik kunnen doorgaan, maar....

B\$ = "AAA"
B\$ = "AAA"
38 = " "
ENTER LINE, VARIABLEN WORKINGSPACE EIND BYTE

Er is duidelijk te zien dat het variabelengebied steeds een byte kleiner wordt, totdat de B\$ leeg is (""). De naam (71) en de lengtebytes (0 en 0) blijven bestaan óók als wij de lus bijv. 8x doorlopen. Het 128-byte markeert het einde van de variabelen. Wij zien dus ook nog een stukje van de ENTER-line verschijnen, ofwel de workspace.

Hierna geef ik een tabel van alle mogelijke variabelen met hun decimale en daarachter hun hexadecimale waarde. (tabel hierna)

Geen waarde hebben de getallen:

¹⁾ Als tweede t/m een na laatste kunnen in een multiple letter variabele (numeriek) nog cijfer- en lettercodes voorkomen van 28(1C) t/m 63(3F)
²⁾ en als laatste cijfer- of lettercode 156(9C) t/m 199(BF)

decimaal	hexadecimaal
0 t/m 38	00 t/m 25 ¹⁾
64 t/m 67	40 t/m 45
96 t/m 101	60 t/m 65
129 t/m 133	81 t/m 85
160 t/m 165	A0 t/m A5 ²⁾
192 t/m 197	C0 t/m C5
224 t/m 229	E0 t/m E5

H. Radius

Tot slot nog een tabel van alle mogelijke variabelen met hun decimale en daarachter hun hexadecimale waarde.

LETTER	alfa STRING VAR	numm. 1-LETR VAR	numm. m-dim ARRAY	numm. m-LETR VAR	numm. m-LETR VAR	alfa m-dim ARRAY	numm. CONTROL VAR
A	38-26	70-46	102-66	134-86	166-A6	198-C6	230-E6
B	39-27	71-47	103-67	135-87	167-A7	199-C7	231-E7
C	40-28	72-48	104-68	136-88	168-A8	200-C8	232-E8
D	41-29	73-49	105-69	137-89	169-A9	201-C9	233-E9
E	42-2A	74-4A	106-6A	138-8A	170-AA	202-CA	234-EA
F	43-2B	75-4B	107-6B	139-8B	171-AB	203-CB	235-EB
G	44-2C	76-4C	108-6C	140-8C	172-AC	204-CC	236-EC
H	45-2D	77-4D	109-6D	141-8D	173-AD	205-CD	237-ED
I	46-2E	78-4E	110-6E	142-8E	174-AE	206-CE	238-EE
J	47-2F	79-4F	111-6F	143-8F	175-AF	207-CF	239-EF
K	48-30	80-50	112-70	144-90	176-B0	208-D0	240-F0
L	49-31	81-51	113-71	145-91	177-B1	209-D1	241-F1
M	50-32	82-52	114-72	146-92	178-B2	210-D2	242-F2
N	51-33	83-53	115-73	147-93	179-B3	211-D3	243-F3
O	52-34	84-54	116-74	148-94	180-B4	212-D4	244-F4
P	53-35	85-55	117-75	149-95	181-B5	213-D5	245-F5
Q	54-36	86-56	118-76	150-96	182-B6	214-D6	246-F6
R	55-37	87-57	119-77	151-97	183-B7	215-D7	247-F7
S	56-38	88-58	120-78	152-98	184-B8	216-D8	248-F8
T	57-39	89-59	121-79	153-99	185-B9	217-D9	249-F9
U	58-3A	90-5A	122-7A	154-9A	186-BA	218-DA	250-FA
V	59-3B	91-5B	123-7B	155-9B	187-BB	219-DB	251-FB
W	60-3C	92-5C	124-7C	156-9C	188-BC	220-DC	252-FC
X	61-3D	93-5D	125-7D	157-9D	189-BD	221-DD	253-FD
Y	62-3E	94-5E	126-7E	158-9E	190-BE	222-DE	254-FE
Z	63-3F	95-5F	127-7F	159-9F	191-BF	223-DF	255-FF
endbyte		128-80					endbyte

BOEKBESPREKING

VOOR GALG EN RAD, Leren programmeren met de Spectrum
door W. van Engelen

Met dit boek zet de uitgever Wolfkamp een goede gewoonte voort om Nederlandstalige boeken voor de Sinclair computers uit te geven. Na ZX-HORIZON nu een boek voor de SPECTRUM, globaal opgebouwd volgens hetzelfde patroon. De rode draad vormen twee spelprogramma's, Galgje en Fruitmachine. Aan de hand hiervan worden vele programmeertechnieken beschreven. Het boek wil een inleiding in het programmeren in BASIC voor de SPECTRUM zijn.

Bij het schrijven is speciaal rekening gehouden met gebruikers die geen of weinig wiskundekennis hebben.

Door de voorbeelden krijgt de lezer inzicht in de manier waarop een probleem met behulp van de computer kan worden aangepakt. Zowel het gebruik van kleur als geluid krijgt veel aandacht, evenals het gebruik van bewegende beelden. In de 17 hoofdstukken komen vele onderwerpen aan de orde. Na een gebruikelijke inleiding wat programmeren is, het toetsenbord en de directe opdrachten wordt in hoofdstuk 4 begonnen met het echte programmeren. De programmavoorbeelden: kwadrateren en Fahrenheit naar Celsius verduidelijken het gebruik van de commando's: REM, LET, GO TO, CLS, INPUT en IF...THEN.

Daarna komt het werken met variabelen aan de orde. Het invoeren en weer-geven (op het scherm) van gegevens met behulp van de opdrachten INPUT, PRINT AT, TAB en INPUT "BERICHT" worden in hoofdstuk 6 besproken. Hoofdstuk 7 en 8 gaan in op lussen (loops) door middel van IF...THEN, FOR...NEXT en het werken met logische variabelen door middel van AND, NOT, OR. Na het werken met strings (\$, VAL\$, CHR\$, CODE, LEN) en de voor ZX80/81 onbekende READ, DATA en RESTORE volgt een uiteenzetting voor sorteren en het werken met arrays met de bijbehorende DIM-instructie.

Aan de hand van de voorbeeldprogramma's zoals: 23 lucifers en brandende lamp, wordt in de hoofdstukken 11 en 12 het grafisch werken met kleur en beweging besproken. Hierbij komen de instructies BORDER, PAPER, INK, FLASH, BRIGHT, LIST, GO SUB, PAUSE, PLOT, CIRCLE, DRAW, SCREEN\$, PI, INT, RND en POINT aan de orde.

Met een springend aapje als voorbeeldprogramma wordt het zelf karakters maken geïllustreerd. Hiervoor wordt ingegaan op het omzetten van decimaal naar binair en omgekeerd en de instructies POKE, USR en BIN. In het 14^e hoofdstuk komt het creëren van geluiden ter sprake. Als voorbeeld worden diverse kant en klare wijsjes gegeven.

Na al het voorbereidende werk komen in de volgende twee hoofdstukken de programma's waaraan het boek zijn titel ontleend aan de orde.

Zowel GALG als RAD zijn volop van uitleg en commentaar voorzien. Tevens worden aanwijzingen gegeven hoe beide programma's verfraaid kunnen worden.

Als toegift volgt het laatste (17^e) hoofdstuk waarin sterrenbeelden op het scherm verschijnen met de namen van de belangrijkste daarin voorkomende sterren.

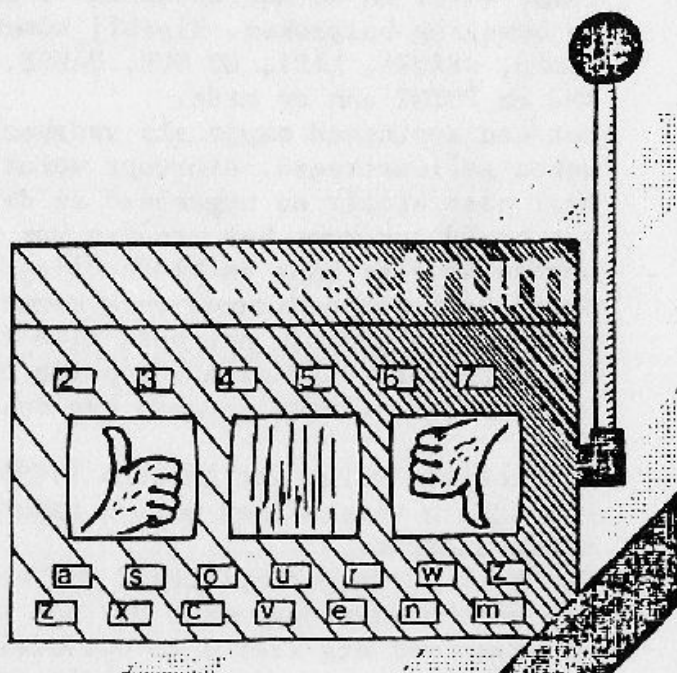
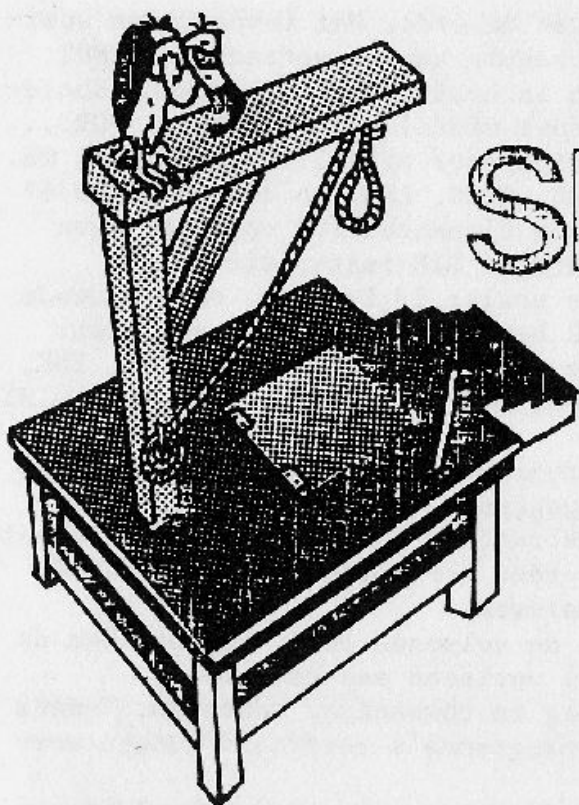
Het boek wordt gecompliceerd met een index en wat reserve-rasters. Hierbij moet worden opgemerkt dat het raster om Uw grafisch ontwerp op te schetsen wel erg kleine en onoverzichtelijk is.

De algemene conclusie is dat dit boek in zijn soort goed leesbaar en vlot geschreven is en voor zijn prijs (f 27,50) zeker in een behoefte zal voorzien.



leren programmeren met de

**ZX
SPECTRUM**



Verkrijgbaar bij:

Wolfskamp

Prijs: f 27.50

WETERINGSCHANS 221
POSTBUS 70254
1007 KG AMSTERDAM
TELEFOON 020 - 278931

HARDWARE

INHOUD

Schema Spectrum	03.2.1.006	03-14
Zelfbouw geheugenuitbreiding (RAM/ROM)	03.2.3.003	03-16

$R_{70} = 220\Omega$

sinclair

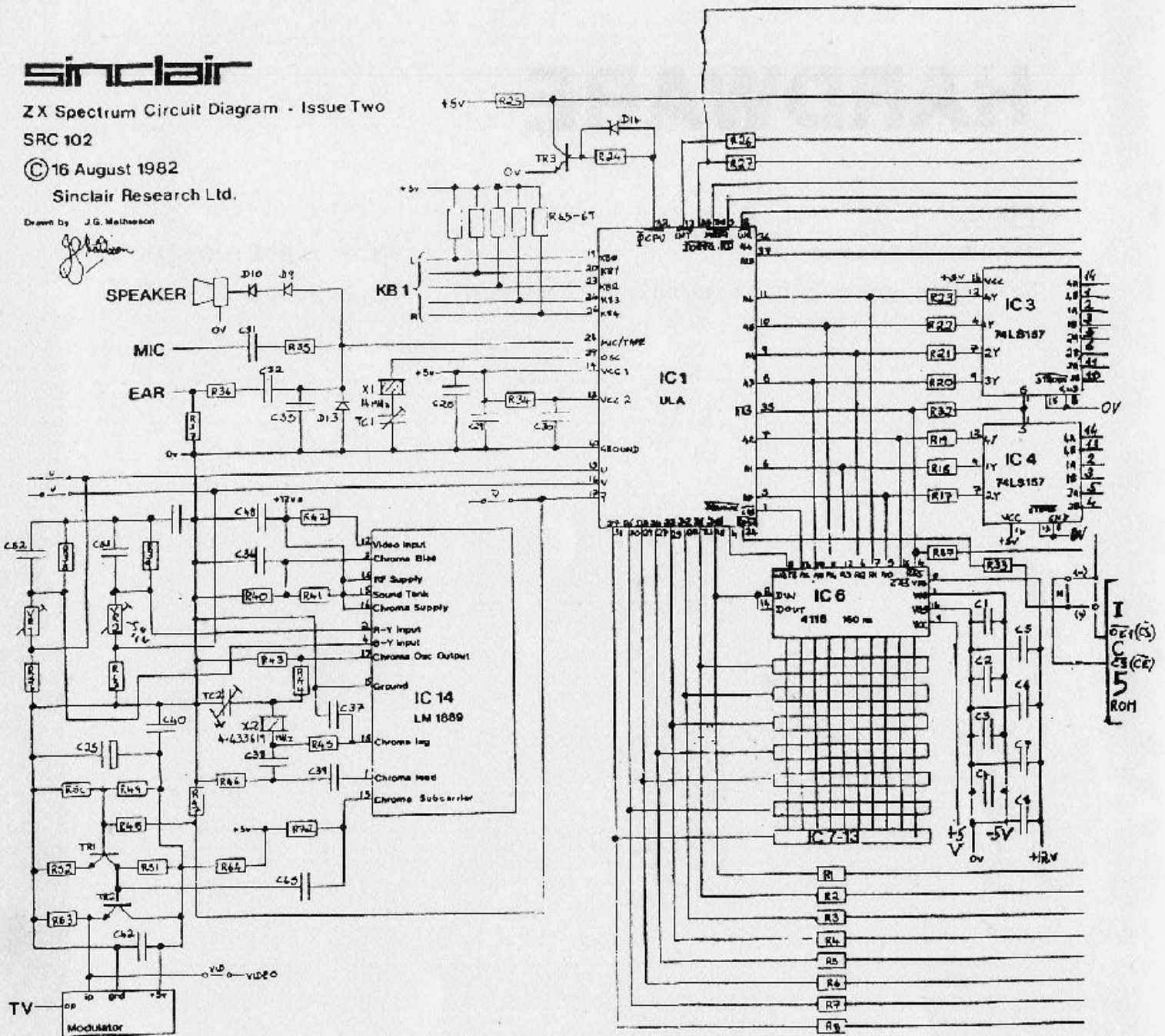
ZX Spectrum Circuit Diagram - Issue Two

SRC 102

© 16 August 1982

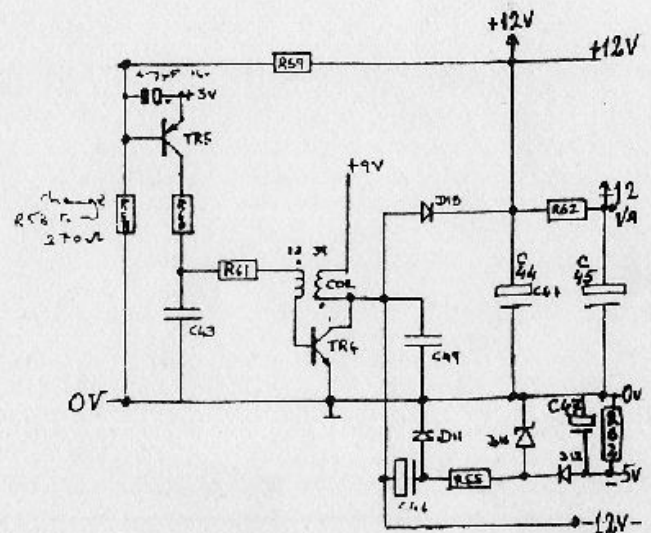
Sinclair Research Ltd.

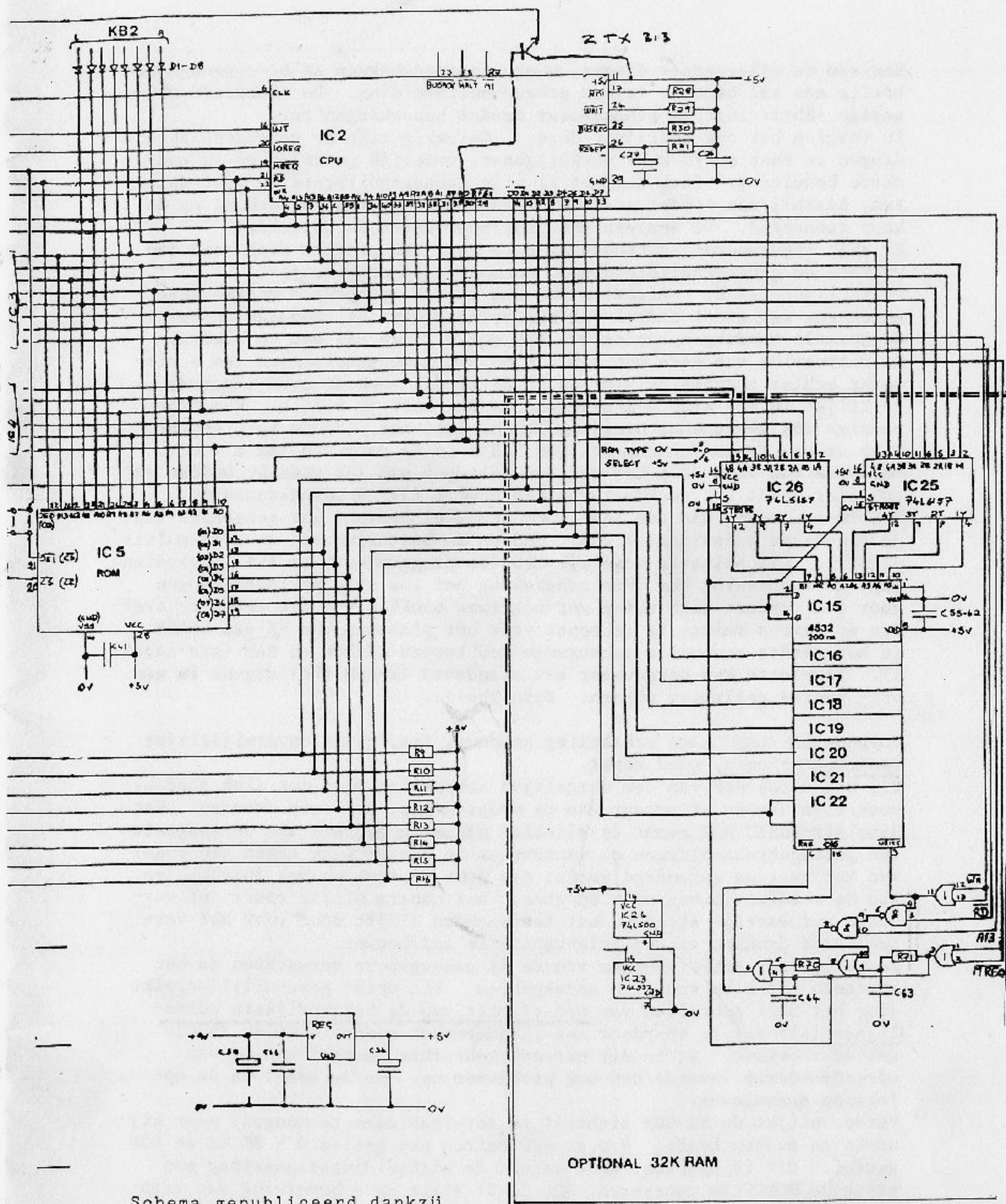
Drawn by J.G. Matheson



UNDERSIDE (A) COMPONENT SIDE (B)

A14	1	A15
A12	2	A13
15V	3	D7
19V	4	
SLOT	5	SLOT
0V	6	D0
TR	7	D1
2R	8	D2
A0	9	D6
A1	10	D5
A2	11	D3
A3	12	D4
10R04A	13	10R04
0V	14	10R04
VIDEO	15	10R04
V	16	10R04
U	17	10R04
10R04	18	10R04
RESET	19	10R04
A7	20	10R04
A6	21	10R04
A5	22	10R04
A4	23	10R04
10R04	24	10R04
10R04	25	10R04
10R04	26	10R04
10R04	27	10R04
10R04	28	10R04
10R04	29	10R04
10R04	30	10R04
10R04	31	10R04
10R04	32	10R04
10R04	33	10R04
10R04	34	10R04
10R04	35	10R04
10R04	36	10R04
10R04	37	10R04
10R04	38	10R04
10R04	39	10R04
10R04	40	10R04
10R04	41	10R04
10R04	42	10R04
10R04	43	10R04
10R04	44	10R04
10R04	45	10R04
10R04	46	10R04
10R04	47	10R04
10R04	48	10R04
10R04	49	10R04
10R04	50	10R04
10R04	51	10R04
10R04	52	10R04
10R04	53	10R04
10R04	54	10R04
10R04	55	10R04
10R04	56	10R04
10R04	57	10R04
10R04	58	10R04
10R04	59	10R04
10R04	60	10R04
10R04	61	10R04
10R04	62	10R04
10R04	63	10R04
10R04	64	10R04
10R04	65	10R04
10R04	66	10R04
10R04	67	10R04
10R04	68	10R04
10R04	69	10R04
10R04	70	10R04
10R04	71	10R04
10R04	72	10R04
10R04	73	10R04
10R04	74	10R04
10R04	75	10R04
10R04	76	10R04
10R04	77	10R04
10R04	78	10R04
10R04	79	10R04
10R04	80	10R04
10R04	81	10R04
10R04	82	10R04
10R04	83	10R04
10R04	84	10R04
10R04	85	10R04
10R04	86	10R04
10R04	87	10R04
10R04	88	10R04
10R04	89	10R04
10R04	90	10R04
10R04	91	10R04
10R04	92	10R04
10R04	93	10R04
10R04	94	10R04
10R04	95	10R04
10R04	96	10R04
10R04	97	10R04
10R04	98	10R04
10R04	99	10R04
10R04	100	10R04





Schema gepubliceerd dankzij
de heer J. Robers te Rijswijk

ZELFBOUW GEHEUGENUITBREIDING VOOR DE ZX-81

Een van de allereerste dingen, waar een nieuwbakken ZX 81-eigenaar behoefte aan zal hebben, is een geheugenuitbreiding. De standaard aanwezige 1Kbyte legt de programmeur teveel beperkingen op.

Zo verging het ook schrijver dezes. Gelukkig zijn er geheugenuitbreidingen te kust en te keur verkrijgbaar, redelijk geprijsd en in een nette behuizing. Toch besloot ik mijn geheugenuitbreiding zelf te maken, daarbij een minder gestroomlijnd uiterlijk van het geheel op de koop toenemend. De redenen voor zelfbouw waren de volgende.

Er zijn tegenwoordig geheugen IC's te koop van grotere dichtheid dan waarmee de geheugenuitbreidingen normaal zijn uitgerust:

64 K i.p.v. 16 K. Afgezien van hun grotere dichtheid missen deze geheugens een groot nadeel van hun voorgangers: de noodzaak om naast de normale voedingsspanning van +5V nog +12V en -5V aan te bieden.

De uitvoering van deze 64K RAM's - 65.536 geheugenplaatsen van 1 bit - maakt echter toepassing voor de ZX 81 minder voor de hand liggend:

je krijgt in één klap 64K geheugen en dat kost je 8 IC's. Een eenvoudige 16K geheugenuitbreiding kan er niet mee. Sinds kort echter zijn er van Texas Instruments 64K RAM's in de vorm van 16K x 4 bit.

Met twee van deze IC's kan dus een geheugen van 16K gemaakt worden en ook hier geldt het voordeel van één enkele (5V) voedingsspanning.

Een tweede reden was een idee voor een veel eenvoudiger generatie van de benodigde hulpsignalen (RAS, CAS en de besturing van de adresmultiplexer). Dit alles is mogelijk met één programmeerbaar IC. Bovendien laat de toepassing van deze schakeling het toe alle voorbereidingen voor een verdere uitbreiding met nog twee blokken van 16K reeds te treffen en tevens ruimte te scheppen voor het plaatsen van 8K aan EPROM in het verder onbenutte geheugengebied tussen ZX-ROM en RAM (zie tabel 1). Tenslotte kan desgewenst het standaard ZX-RAM (1K) ergens in een vrij gebied geplaatst worden. Zuinigheid...

Volgens deze opzet is een schakeling gebouwd, die, na de onvermijdelijke aanloopproblemen, goed werkt.

Bij het ontwerpen van een dergelijke schakeling moet men zich goed bewust zijn van de structuur van de programma's, die erop draaien. Het Sinclair-BASIC ROM maakt op allerlei manieren gebruik van de aangesloten geheugenschakelingen en wanneer er in het ontwerp tegen een regel van het systeem gezondigd wordt, dan komt de zaak na het inschakelen van de voeding domweg niet op gang; het scherm blijft zwart, of vertoont wat warrige strepen, het toetsenbord blijft doof voor het verwachte drukken van de teleurgestelde zelfbouwer.

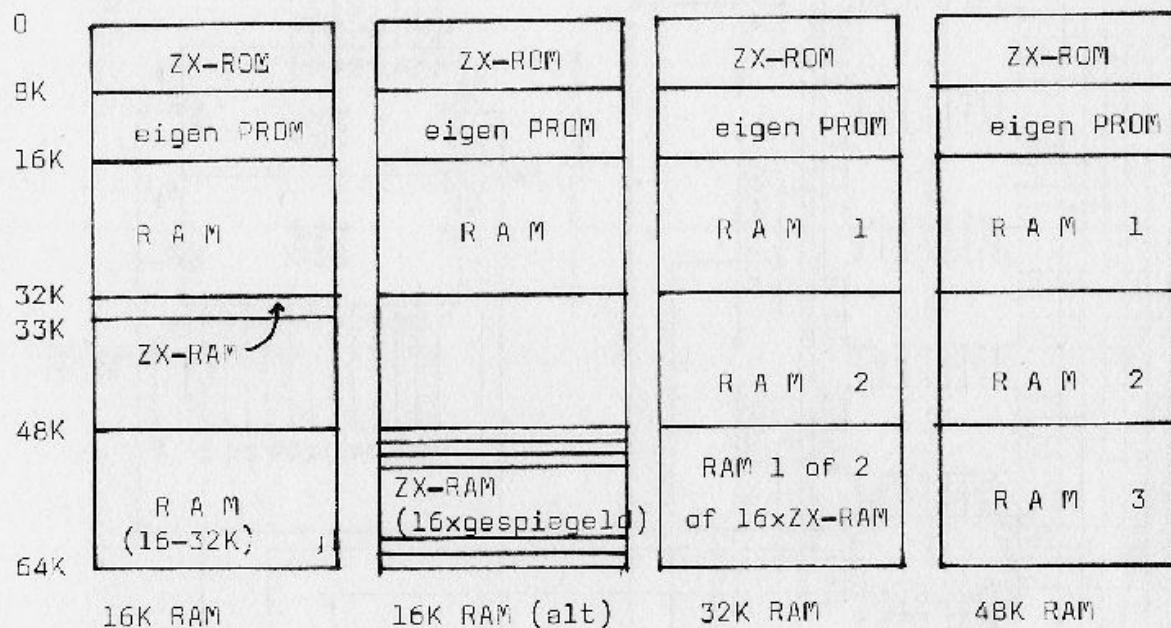
Zo blijkt het BASIC-systeem van de ZX geheugen te verwachten in het bovenste gedeelte van zijn adresgebied. Dit wordt gewoonlijk bereikt door het niet gebruiken van een of meer van de belangrijkste adreslijnen (A15 enz.), waardoor een geheugenblok meerdere malen in het gebied voorkomt. De in dit ontwerp gebruikte "nette" manier van adresdecoderen leverde dan ook problemen op. In de tabel is de oplossing aangegeven.

Verder blijkt de ZX-ROM zichzelf in adresgebieden te mengen, waar hij niets te zoeken heeft, d.w.z. ook buiten het gebied 0 - 8K is de ROM actief. Dit is opgelost door vanuit de uitbreidingsschakeling een passende ROMCS te genereren. De ZX 81 slikt deze beperking van zijn leefruimte zonder protesten, waarmee de reden van deze uitbreidings-

lust duister blijft

V o o r- e n n a d e l e n .

Om met de voordelen van de beschreven schakeling te beginnen: een 16K geheugenuitbreiding, die slechts 6 IC's vraagt (resp. 8 IC's voor 32K en 10 IC's voor 48K) met de mogelijkheid om tot 8K aan extra PROM toe te passen, die bovendien gevoed kan worden uit de standaard ZX-voeding (wel een eigen 5V stabilisator graag!) Nadelen: een 16Kx4 RAM is (nog wel) duurder dan vier 16Kx1 RAM's en de toegepaste 6331 is niet zomaar door iedereen te programmeren.



tabel 1. Diverse geheugenindelingen

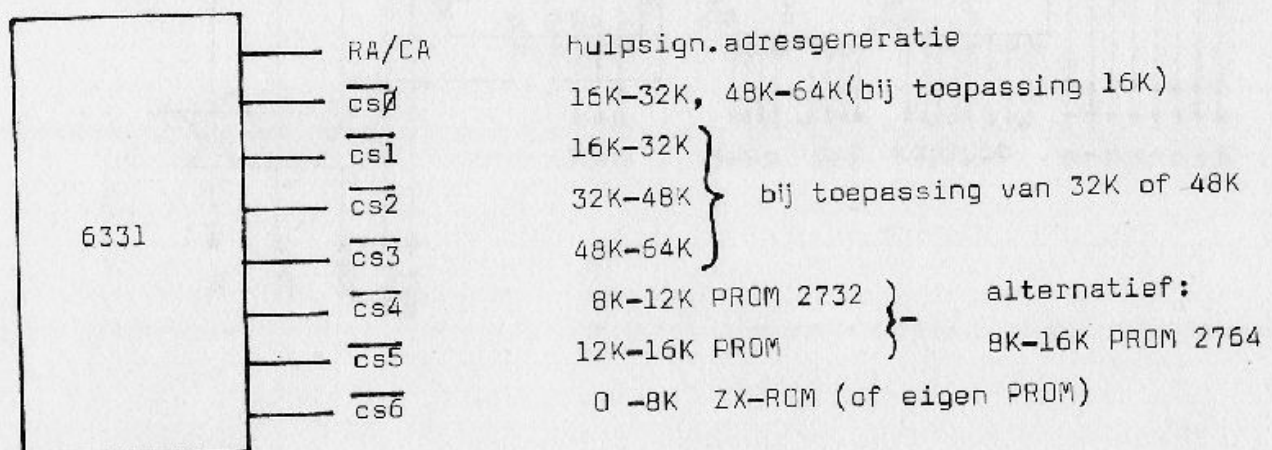


fig. 2. Indeling van de 6331 uitgangen tbv. de chip-select.

SOFTWARE

INHOUD

Renumber	03.5.2.012	03-20
Fast-Save-Load 2400 Baud	03.5.2.013	03-21
ZX en de centjes	03.5.2.018	03-26
BASICODE-protocol	03.5.3.001	03-27
Game of Life op z'n Booleaans	03.5.4.016	03-32
12 jockeys, deel 1	03.5.4.021	03-37
Rekenoefeningen voor het jonge volkje	03.5.4.024	03-41
YAHTZEE-scorekaart	03.5.4.027	03-44
IMPULSOFT-cassettes, beschrijvingen	03.5.9.001	03-45

R E N U M B E R P R O G R A M M A V O O R D E Z X - 8 1

Van D.J.van Ooijen ontvingen wij een programma om BASIC-regels te "re-numberen", dat - in tegenstelling tot het programma in IMPULS nr 1, pag. 01-49 - werkt. U laadt het met een HEX-LOADER van Uw keuze, bijvoorbeeld die uit IMPULS NR 1, pag 01-49.

adres	dec. code	mnemonic	toelichting
16509	--		MSB > nummer van eerste BASIC-regel.
16510	--		LSB >
11	34		LSB > lengte van eerste BASIC-regel:
12	0		MSB > (dat is i.c. 34 bytes (16546 - 16513)
13	234		REM
14	17	LD DE, 16509	In D komt MSB, in E LSB.
15	125	LSB > 16509	Het adres waar het RENUMBER-programma begint *)
16	64	MSB >	
17	33	LD HL, 100	
18	100	LSB > 100	Het eerste nieuwe regelnummer*) minus een stap.
19	0	MSB >	
16520	26	LD A, (DE)	In A komt de inhoud van het eerste adres van de
21	61	DEC A	volgende BASIC-regel. Is deze inhoud 118 (=HALT)
22	254	CP 117	dan is het einde van het BASIC-programma bereikt.
23	117		Men is bang voor '118'; daarom de omslachtige
24	200	RET Z	'truc' via 117.
25	60	INC A	Inhoud A herstellen.
26	254	CP 39	Kijken of je niet boven 39*256 komt; BASIC-regels gaan niet hoger dan 9999.
27	39		
28	200	RET Z	
29	1	LD BC, 10	IN B komt MSB, in c LSB.
16530	10	LSB > 10	gewenste stappengrootte*).
31	0	MSB >	
32	9	ADD HL, BC	Het eerste nieuwe regelnummer wordt gevormd.
33	235	EX DE, HL	
34	114	LD (HL), D	Op 16509 stop je MSB van het nieuwe regelnummer.
35	35	INC HL	HL wordt 16510.
36	115	LD (HL), E	Op 16510 stop je LSB van het nieuwe regelnummer.
37	35	INC HL	HL wordt 16511.
38	78	LD C, (HL)	IN C stop je LSB van 'lengte eerste BASIC-regel'.
39	35	INC HL	HL wordt 16512.
16540	70	LD B, (HL)	In B stop je MSB van 'lengte eerste BASIC-regel'.
41	9	ADD HL, BC	In HL komt laatste adres van eerste BASIC-regel.
42	35	INC HL	In HL komt eerste adres van volgende BASIC-regel.
43	235	EX DE, HL	
44	24	JR DIS	
45	230	-26	Zelfde verhaal voor de volgende BASIC-regel.
46	118		Einde van de eerste BASIC-regel.
47	--		MSB >
48	--		LSB > nr van tweede BASIC-regel etc.

*) Deze getallen kan je, als het MC-programma al is geschreven, wijzigen door op de juiste adressen het juiste byte (MSB, LSB) te POKEn.

2400 BAUD FAST-SAVE/LOAD ROUTINE VOOR DE ZX-81

In IMPULS 2 (pag 02-34 e.v.) heeft H. Radius een routine beschreven waarmee de ZX-80 kan LOADen en SAVEn met een snelheid van 2400 Baud. Toen ik deze wilde aanpassen voor de ZX-81 en geschikt maken voor normaal LOADen bleek dat het sneller, korter en gemakkelijker te bedienen ('user-friendly') kan. Voor een goed begrip verwijs ik naar dat artikel van Radius.

AANPASSING VOOR DE ZX-81

Allereerst moet de opdracht POP DE vervallen. Verder moeten begin- en eindadres (ENDBYTE) veranderd worden in het adres van VERSN - 1: 16392_d = 4008_h resp. LOAD/SAVE UPDATE: 508_d = 01FC_h.

SNELLER

Ik heb de stopbits laten vervallen. Per byte zijn nu nog maar 9 bits nodig, te weten een startbit en de 8 databits. Een bit duurt 1/2,4 ms (2,4 kHz puls of twee 4,8 kHz pulsen); 11 bits duren 4,58 ms (13 000 bytes per min) maar als er maar 9 bits overgeseind behoeven te worden (3 1/3 ms per byte) komt dat overeen met 16 000 bytes per minuut. De normale SAVE/LOAD-snelheid is ca. 2 300 bytes per min. omdat een byte gemiddeld 26,1 ms duurt.

KORTER

Het programma kan ook korter zijn, omdat ik de stopbits heb laten vervallen en omdat ik een aantal 'rare' sprongen in het programma van Radius vermeden heb. De eindcontrole op het LOADen geschiedt via de E-LINE-check van de LOAD/SAVE UPDATE-routine uit de ROM.

Een FIND-LEADER routine en een BREAK-detectie zijn toegevoegd om deze routine gemakkelijker te bedienen te maken. Dit kost tweemaal 9 bytes.

<u>per saldo;</u>	<u>nieuw:</u>	<u>was:</u>
SAVE-deelroutine	66 bytes	87(+3) bytes
PULSES-deelroutine	21 bytes	21 bytes
LOAD-deelroutine	45 bytes	(10+)61 bytes
FIND-LEADER en BREAK-detectie	18 bytes	-
totaal:	150 bytes	169(+13)

GEBRUIKERSGEMAK

Door de FIND-LEADER-routine hoeft het laden niet binnen de 2,3 s van de leader gestart te worden, het mag ook voordien.

Door de BREAK-detectie-routine kunt U het hulp-programma gebruiken voor FAST-SAVEing. Dat hulpprogramma is nodig om met de normale LOAD-procedure te kunnen werken.

De routine wordt niet in een REM-statement geschreven, maar hoog in het RAM. Hierdoor wordt het gebruik ook vereenvoudigd. Als de routine in een REM-statement geschreven zou zijn, dan moest het zowel in het programma als 'los' (apart vooraf geladen) beschikbaar zijn. Vooral het eerste is een moeizame handeling. Die REM-regel is nodig voor de overname van de 'losse' routine. Bij een BASIC-programma groter dan 16 k en een 32 k RAM is het gebruik van een REM-statement echter onvermijdelijk.

OPSLAAN VAN ROUTINES, HULPPROGRAMMA

Eerst worden de deelroutines (machine-code) van de listing in de volgende adressen geschreven:

SAVE 7669_h(30313_d)-76AA_h(30378_d)

PULSES 76AB_h(30379_d)-76BF_h(30399_d)

LOAD/BREAK 76C0_h(30400_d)-76FE_h(30462_d)

desgewenst uit te breiden met: DISPLAY/VERIFY 76F3_h(30451_d)-761E_h(30494_d)

NB: de laatste deelroutine 'overschrijft' voor een deel de LOAD/BREAK-deelroutine. Boven dien moeten ook andere aanpassingen geschieden, zoals uit de listing blijkt.

Vervolgens wordt het volgende programma ingetypt:

```
2 DIM X$(150)      (resp. 182)
4 FOR X=1 TO 150    idem
5 LET X$(X)= CHR$(PEEK (30312+X))
6 NEXT X
```

Nadat dit programma is geRUNd wordt het veranderd in:

```
2 SAVE "FSL..."  (... = naam van het programma)
4 FOR X=1 TO 150   (resp. 182)
5 POKE 30312 +X, CODE X$(X)
6 NEXT X
7 RAND USR 30400
```

Door nu als 'direct command' te typen POKE 16389, 76 (N/L) wordt de D-FILE ineengeschrompeld. Hierdoor worden slechts 25 i.p.v. 793 bytes meegeseind. Dat bespaart bij het SAVEn en LOADen telkens ca. 19 s nutteloze S/L-tijd.

Dit programma moet nu in de FAST-mode worden geSAVEd met GOTO 2 (N/L), zodat het een 'auto-run'programma wordt. Ca. twee tellen nadat het scherm zwart wordt, wordt op de BREAK-toets gedrukt.

Hiermee is het hulp-programma voor het LOADen en het basis-programma voor het SAVEn vastgelegd. Voor alléén LOAD zou men kunnen volstaan met de routine vanaf 30400 (63 bytes).

FAST SAVING

Eerst wordt het hulp-programma ge-SAVEd, zoals hierboven beschreven. Dan wordt het programma, dat U FAST wilt SAVEn, ingetypt of (normaal) geLOAD.

Breng de computer dan (zodanig) met 'direct command' in de FAST mode. Als het programma 'auto-run' moet zijn, dan wordt de programmaregel, die de SAVE-opdracht bevat veranderd in "xx RAND USR 30313", waarna het programma direct ná het hulp-programma geSAVEd wordt met de opdracht GOTO xx (N/L).

Voor een niet-automatische start wordt "RAND USR 30313 (N/L)" als 'direct command' gegeven.

Het FAST-SAVEn is duidelijk zichtbaar op het scherm.

Het is beter de D-FILE niet ineens te laten schrompelen (zoals boven beschreven voor de normale S/L-snelheid) omdat de tijd, die nodig is om de D-FILE weer te herstellen langer is dan de tijdwinst van die 793 - 25 bytes met 16000 bytes per minuut.

U kunt het hulp-programma vóór elk FSL-programma opnemen. Het hulp-programma draagt dan de naam van het erna geSAVEde programma. U kunt echter ook één hulp-programma voor alle FSL-programma's op de band hebben, als U het ongerief van het heen-en-weer-spoelen en niet-op-naam-LOADen op de koop toe neemt.

LOADING

Het LOADen geschiedt alsof het een normaal geSAVEd programma was. Nadat het hulp-programma is geladen, wordt het scherm donker (of verandert in een FAST-LOAD-design' als U de DISPLAY-VERIFY-variant hebt gekozen). Laat de recorder gewoon doorlopen, totdat het programma geladen is.

AANTAL BYTES

De tijdsduur van het FSL SAVEn of LOADen wordt bepaald door het aantal bytes dat wordt overgeseind. Dat zijn 116 systeem-variabelen, het programma, de 793 bytes van de D-FILE en de VARS (variabelen). Het aantal VARS is 1 als de computer geCLEARed is. Met een 'direct command': PRINT PEEK 16404+256*PEEK 16405 - 16393 (N/L) kunt U achter het aantal komen.

VERIFY

Als U het uitgebreide programma hebt gekozen (182 bytes) dan kunt U na het SAVEn controleren of het goed op de band staat door de band direct weer af te spelen. Na RAND USR 30465 (N/L) vergelijkt de computer elke byte met die in de computer. Als er een verschil optreedt, verschijnt het adres, waar dat gebeurde (decimaal) op het scherm. Een foutloze VERIFY eindigt met "16509" (het begin adres van het programma). De systeem-variabelen worden bewust niet in de vergelijking betrokken (7711_h-7718_h). Met deze routine is het ook mogelijk het adres van een BASIC-regelnummer op te zoeken.

H.H. van Abbe

LISTING VAN DE DEELROUTINES VAN 2400 BAUD FAST-SAVE-LOAD

SAVE deelroutine

```

76600 310040 LD HL, VERSN-1 (STARTADRES-1)
76601 1E0000 LD E, 0
76602 010012 LD BC, 18 resp 128d
76603 0E1900 LD A, +25d
76604 32B176 LD (XX), A
76605 00B676 CALL PULSES
76606 10 DEC E
76607 20F020 JRNZ LEADER
76608 01040C STARTBIT
76609 3E3300 LD A, +51d
76610 32B176 LD (XX), A
76611 00FC01 CALL LOAD/SAVE-UPDATE 3)
76612 00B676 CALL PULSES
76613 56 (OUT-BYTE)
76614 1E0000 LD D, (HL)
76615 0B12 LD E, +8d
76616 000000 JRNZ, 0-BIT
76617 3E1900 LD A, +25d
76618 32B176 LD (XX), A
76619 010510 LD BC, 16 resp 5d
76620 000000 JR C, OUT-BIT
76621 3E3300 LD A, +51d
76622 32B176 LD (XX), A
76623 010412 LD BC, 16 resp 4d
76624 00B676 CALL PULSES
76625 10 DEC E
76626 20E040 JRNZ, EACH BIT
76627 10D101 JR STARTBIT

```

LOAD deelroutine

```

76600 061000 WAIT
76601 00E076 FIND LEADER CALL SIGNAL
76602 30F020 JRNZ, WAIT
76603 10F020 DJNZ, FIND LEADER
76604 310040 LD HL, VERSN (eerste gesavede byte)
76605 00E076 CALL SIGNAL
76606 30F020 JR C, STARTBIT (carry als HI < 153us)
76607 10F020 LD E, +8d
76608 00E076 CALL SIGNAL
76609 30F020 JRNZ, BIT
76610 00E076 CALL SIGNAL
76611 10F020 RL D
76612 10 DEC E
76613 30F020 JRNZ, READ
76614 00E076 LD HL, (D)
76615 00E076 CALL LOAD-SAVE UPDATE (60 T-states, INC HL tot E-LINE bereikt)
76616 30F020 JR, STARTBIT
76617 10F020 LD C, 0
76618 30F020 LD A, 7F (breakdetectie, zie Radius)
76619 00E076 IN A, (FE)
76620 00E076 RR A
76621 30F020 JRNZ, TAPE
76622 00E076 POP, HL
76623 00E076 RET (verwijder RET adres CALL)
76624 30F020 INC C
76625 00E076 LD A, (FE)
76626 30F020 JRNZ, TAPE
76627 00E076 LD A, C
76628 00E076 CP 6d
76629 00E076 JR C, SIGNAL
76630 00E076 CP 16d
76631 00E076 RET

```

³⁾ INC HL & check HL < E-LINE. 60 States

²⁾ (wacht op 24 pulsen ononderbroken van 4800 Hz: d.i. 5 ms.)

MC-routines, voor zover niet-essentieel veranderd t.o.v. IMPULS 02-35/37 wordt geen uitleg gegeven, behalve via LABELS tussen () -

Na sluitingsdatum kreeg ik een uitvoerige brief van Leo van Dorp uit Wageningen. Met een kortere routine kan men 22 000 bytes per min. bereiken. In een volgende IMPULS wordt hierop teruggekomen.

⁴⁾ 00 wordt gePOKEd, dus kan nu met elke (hex) waarde worden geschreven.

VERVOLG LISTING

HEX-LOADER voor diegenen die niet over een hex-loader beschikken. RUN en INPUT (onbeperkt even aantal) hex-codes. Om te stoppen: toets S.
Check: RUN 100 geeft 1 hex-adres en 8 hex bytes per regel.
Als U een ander Start-adres wilt, dan kunt U regels 10 en 100 veranderen.

VERIFY deelroutine

De VERIFY-routine kan aan de 150 bytes toegevoegd worden waardoor de gehele routine 182 bytes lang wordt. Dan beschikt U ook over een DISPLAY-faciliteit (het LOADen wordt zichtbaar op het scherm). Eerst komen enkele veranderingen in de LOAD-deelroutine:

76DE (was 20F3 JRNZ, READ) wordt: 1827 JR, VERIFY

76E0 72 IN-RAM

76E1 CDFC01 L/S UPDATE

en vervolgens de volgende listing:

```

76E3 03FF OUT (FF), A
76E4 00 INC A
76E5 17 RL A
76E6 58FA JRC, TAPE
76E7 79 LD A, C
76E8 FE24 CP 4d
76E9 58E5 JRC, SIGNAL
76EA FE2C CP 12d
76EB 00 RET
76EC 00 SET 0, VERSN
76ED FDCB09C6 JR, WAIT
76EE 18B9 JRNZ, READ
76EF 20CA BIT 0, VERSN
76F0 FDCB0946 JRNZ, IN-RAM
76F1 00 AND, A
76F2 00 LD A, D
76F3 7A LD BC, 407Dh
76F4 017D40 LD BC, BC
76F5 00 SBC HL, BC
76F6 00 ADD HL, BC
76F7 00 JRC, L/S UPDATE
76F8 00 CP HL
76F9 00 JRNZ, L/S UPDATE
76FA 00 PUSH HL
76FB 00 POP BC
76FC 00 RET
76FD 00
76FE 00
76FF 00
7700 00
7701 FDCB09C6, VERIFY
7702 18B9
7703 20CA
7704 FDCB0946
7705 00
7706 00
7707 7A
7708 017D40
7709 00
7710 00
7711 00
7712 00
7713 00
7714 00
7715 00
7716 00
7717 00
7718 00
7719 00
771A 00
771B 00
771C 00
771D 00
771E 00

```

```

10 LET A=30313
20 INPUT A$
30 IF A$="S" THEN STOP
35 PRINT A$
40 STOP
50 FOR X=1 TO LEN A$/2
60 POKE A,16*CODE A$/2
70 LET A=A+1
80 NEXT X
90 GOTO 20
100 LET A=30313
105 LET A=A/4096
110 FOR X=1 TO 4
120 PRINT CHR$(INT K/256);
130 LET K=16*(K-INT K/256)
140 NEXT X
145 PRINT " "
150 FOR X=0 TO 7
160 LET P=PEEK (A+X)/16
170 PRINT CHR$(INT (P+256));
180 NEXT X
185 LET A=A+X
190 GOTO 105

```


ZX EN DE CENTJES

In veel financiële programma's kom ik nog wel eens tegen, dat men moeite heeft om de neiging van de ZX, om nullen achter de komma te negeren, te overbruggen.

Vaak is het ook zo, dat het onder elkaar krijgen van de punt (lees komma) nogal wat problemen oplevert.

Vandaar onderstaand programmablokje, dat u ergens in uw financiële programma kunt opnemen om er naar toe te gosub-en als dat nodig is.

Met een eenvoudige PRINT TAB ?; B\$ komen dan alle getallen goed en op de cent nauwkeurig onder elkaar te staan.

HET PROGRAMMABLOKJE:

```
xx01 LET B$ = STR$ (bedragvariabele * 100+.5))/100
xx02 IF VAL B$ < > INT (VAL B$) THEN GOTO xx04
xx03 LET B$ = B$ + ".00"
xx04 IF LEN B$ - LEN (STR$ (INT (VAL B$))) = 2 THEN LET
    B$ = B$ + "0"
xx05 IF ABS (VAL B$) < 10 THEN LET B$ = "s" + B$
xx06 IF ABS (VAL B$) < 100 THEN LET B$ = "s" + B$
xx07 IF ABS (VAL B$) < 1000 THEN LET B$ = "s" + B$
xx08 en verder uit te breiden naar gelang de grootte
    van de getallen (dit gaat tot 9999.99).
xx09 IF VAL B$ >= 0 THEN LET B$ = "s" + B$
xx10 RETURN
```

De "s" hierboven staat voor "spatie".

Wilt U mooie(?) blokken getallen hebben, dan kunt U de spatie vervangen door een 0; behalve die in regel xx09.

Om ruimte te sparen kunt u, daar waar in het programma VAL B\$ of (VAL B\$) staat, dit vervangen door de bedragvariabele.

De B\$ is uiteraard te vervangen door iedere andere \$.

Veel cijferplezier toegewenst,

Rob van Staalduinen.

BASICODE-2 PROTOCOL

Het BASICODE-2 protocol, zoals het werd uitgezonden door HOBBYSCOOP op 9 januari 1983 en gedcodeerd op een ZX-80, 4 k ROM, 1 K RAM.

Door H. Radius werd dit programma gedecodeerd en als listing ingezonden. Daar wij ons voorstellen dat U de inhoud van het protocol wilt kennen, ook zonder dat het op Uw scherm verschijnt, is de tekst, zoals die op Uw scherm zou zijn verschenen, hieronder afgedrukt. Ook wordt later de listing afgedrukt van de routine, waarmee de tekst wordt aangepast aan de afmetingen van de diverse beeldschermen (aantal regels en karakters per regel).

Hallo HOBBYSCOOPVRIENDEN!

Hier is de uitleg over de nieuwe BASICODE-2.

Voortaan is dit de standaard, die bij de uit te zenden hobbyscoopprogramma's wordt gebruikt.

Wat betekent dat? Dat betekent dat het vanaf nu niet meer nodig is dat U zelf elk door HOBBYSCOOP uitgezonden programma moet aanpassen voordat het op Uw computer zonder fouten werkt.

Hoe kan dat? Een voorbeeld: De opdracht "CLEAR SCREEN" is in elke computer anders. Daarom zal in BASICODE-2 voortaan elk uitgezonden programma dat op een zeker ogenblik het scherm wil schoonmaken op die plek een instructie "GOSUB 100" bevatten. Als U dan op regel 100 een subroutine hebt staan die voor Uw machine het scherm schoonveegt dan is dus het probleem van "CLEAR SCREEN" opgelost.

Voordat U het uitgezonden BASICODE-2 programma gaat laden, moet wel eerst de benodigde subroutine voor "CLEAR SCREEN" worden klaargezet. Zo zijn er verschillende subroutines, die elk een bepaalde nuttige taak kunnen verrichten, en wel allerlei taken die in de diverse gangbare computers verschillende commando's nodig hebben.

Bijvoorbeeld: "Zet de cursor op een bepaalde plek";
 "Kijk of er een toets is ingedrukt";
 "Wacht tot er een toets is ingedrukt";
 "Print getallen volgens een opgegeven aantal
 cijfers voor en/of na de komma"

en zo voort.

Deze subroutines zijn zeer zorgvuldig uitgekozen. De ervaringen na een jaar uitzendingen in BASICODE liggen daar natuurlijk aan ten grondslag. Vrijwel alle eerder uitgezonden programma's zouden met deze subroutines

zonder enig probleem meteen op Uw computer kunnen werken. Wat er dan nog overblijft zijn dingen als bijvoorbeeld:

- de tijd opvragen;
- knipperende teksten;
- kleur;
- muziekjes
- graphics;
- en meer van dat moois.

De belangrijkste redenen om hiervoor in BASICODE-2 geen voorzieningen voor te treffen zijn:

1 Het komt te weinig voor;

2 De verschillen tussen de computers zijn te groot

Zodra de uitbreiding nodig blijkt zal de Redactie van HOBBYSCOOP dat zeker bekijken. Met de nu vastgelegde subroutines kunnen wij echter beslist een flinke tijd vooruit.

Natuurlijk wilt U precies weten welke subroutines op welk plaats moeten staan. Die informatie komt zo direct. Allereerst echter dit:

Voor alle computers waarvoor een BASICODE vertaalprogramma bestaat zal dat programma opnieuw worden uitgezonden. De nieuw uit te zenden versie zal U dan tevens de benodigde subroutines kant en klaar leveren. Het is dus helemaal niet nodig dat U deze nieuwe subroutines zelf gaat zitten bedenken (maar het mag natuurlijk wel).

Het spreekt vanzelf dat U bij het maken van programma's deze nieuwe standaard dient toe te passen. Zonder dat zou Uw programma op geen enkele andere machine kunnen werken.

De nieuwe standaard zal nog niet meteen van kracht zijn: eerst moeten de nieuwe lees- en schrijfprogramma's zijn uitgezonden. Dat zal een maand of twee duren, dus vanaf pakweg eind maart zal het zover zijn.

Het juiste moment wordt tijdig aangekondigd en vanaf dat ogenblik vinden alle verdere uitzendingen plaats volgens het nieuwe BASICODE-2 protocol. Hoe is dat protocol nu precies? Laad daartoe het tweede deel van dit programma en RUN het.

Hallo, daar zijn wij weer met deel twee. Dit deel geeft een overzicht van het BASICODE-2 protocol.

Zodra de nieuwe lees- en schrijfroutines zijn uitgezonden zal HOBBYSCOOP verder volgens dit protocol werken. Het protocol werkt volgens een aantal vaste subroutines waarvoor de regels met nummers onder de 1000 zijn gereserveerd. Daarnaast enkele praktische regels, bedoeld om de lees-

baarheid van de programma's te verbeteren.

Luisteraars die programma's ter uitzending willen aanbieden (en dat zijn er gelukkig steeds meer), zullen hun programma's aan het protocol moeten aanpassen.

Allereerst een overzicht van de subroutines.

GOSUB 100. Met deze opdracht kunt U voortaan het scherm laten schoonwissen en de cursor linksboven in het scherm plaatsen. Op regel 100 zal voor Uw computer dus staan HOME of CLS of PRINT CHR\$(12) of PRINT CHR\$(147) of ... nou ja, ga zo maar door. Op dezelfde regel na een : of op de volgende regel staat dus een RETURN.

GOSUB 110. Deze opdracht kunt U gebruiken om de cursor op een gewenste plaats op het scherm te zetten. Daar hoort dus nog wat bij. De bovenste schermregel heet voortaan regel 0, de volgende is 1, enzovoort.

De meest linkse plaats op elke regel is plaats 0 en evenzo oplopend tot 39 bij een scherm met 40 karakters op een regel. De bedoeling is nu dat U in de variabele HO aangeeft op welke plaats in de horizontale richting de cursor moet komen en in de variabele VE op welke regel in de verticale richting. (Voorbeeld: 'VE=2 : HO=4 : GOSUB 110' Hiermee komt de cursor op regel 2 (dus de derde regel van boven) en op die regel op plaats 4 (dat is de vijfde plaats van links). Hierna zal het eerste karakter van een volgend PRINT-statement op die aangegeven plaats terecht moeten komen.

GOSUB 120. Deze subroutine doet juist het omgekeerde van de vorige: nadat Uw programma deze subroutine heeft aangeroepen staat in VE en HO de juiste positie van de cursor op dat moment. Door dan bij VE of HO iets op te tellen plaatst U de cursor ten opzichte van de oude plaats.

GOSUB 200. Dit is een subroutine die voor U kijkt of er een toets van het toetsenbord was ingedrukt. Als er eentje ingedrukt was, dan wordt in de variabele IN\$ aangegeven welke toets, als er geen was ingedrukt dan zal IN\$ een lege string worden.

GOSUB 210. Deze subroutine lijkt wat op de vorige. Nu wordt echter door de subroutine gewacht totdat er een toets wordt ingedrukt en dan pas wordt IN\$ weer aan Uw programma afgegeven.

GOSUB 250. Met deze subroutine kan een BASICODE voortaan een piepje afgeven. Bij computers waar geen luidspreker is ingebouwd doet deze subroutine natuurlijk niets. Met deze subroutine kan geen muziek worden gemaakt. De toonhoogte en de duur van het piepje is niet vastgesteld. De subroutine is alleen bedoeld de mogelijkheid tot een hoorbaar signaal te bieden.

GOSUB 260. Het aanroepen van deze subroutine leidt tot een antwoord op de variabele RV. In RV wordt namelijk een RANDOM-getal afgeleverd. Het kleinste getal dat deze routine kan afleveren is 0, alle afgeleverde getallen zijn kleiner dan 1. Deze subroutine is opgenomen omdat de RND-functie in diverse micro's nogal verschillend schijnt te werken.

GOSUB 270. Met deze aanroep worden afgedankte strings opgeruimd en wordt in de variabele FR als antwoord gegeven hoeveel bytes er nog ongebruikt zijn. Tekstprogramma's kunnen hiermee tijdig zien aankomen of het geheugen vol dreigt te raken en eventueel een passende melding produceren.

GOSUB 300. Alvorens deze subroutine aan te roepen moet U de variabele met de naam SR een getalswaarde hebben gegeven. Als antwoord geeft deze subroutine U de STR\$(SR). Dit lijkt wat overbodig, maar er is nogal wat variatie in wat de STR\$-functie op diverse machines doet: de een geeft wel een spatie na het laatste cijfer, de ander niet. Sommige geven bij positieve getallen een spatie op de plaats van het teken, de andere beginnen met eerste cijfer. Deze nieuwe BASICODE subroutine geeft geen spaties, niet ervoor en niet erna.

GOSUB 310. Dit is wel een hele mooie, althans voor een aantal computers, die dit niet van huis uit kennen: het getal dat U in de SR meegeeft wordt omgezet in een SR\$ door de subroutine. De lengte van SR\$ is na afloop gelijk aan wat U tevoren in CT had opgegeven. Als U in CN een getal groter dan 0 had vermeld, dan bevat SR\$ een decimale punt met daarna nog CN cijfers. (Enkele voorbeelden: SR=123.4567 : CT=7 : CN=2 : GOSUB 310. Dit levert als antwoord SR\$=' 123.46'. Ander voorbeeld: SR=-1E-3 : CT=7 : CN=3:GOSUB 310 ' levert als resultaat SR\$='-0.001'.

SR=98765 : CT=5 : CN=1 GOSUB 310' dan wordt SR\$='*****'

'SR=98765 : CT=5 : CN=0 : GOSUB 310' geeft als reactie SR\$='98765')

U ziet: CT is steeds het totale aantal cijfers in het antwoord en na de punt staan nog CN cijfers. Als het niet mogelijk is, dan komen er CT sterretjes en het antwoord is steeds in cijfernotatie en netjes afgerond. Dit is natuurlijk bedoeld om mooie tabellen te kunnen maken.

GOSUB 350. Zorgt ervoor dat SR wordt geprint op de printer. SR\$ kan zijn bepaald door de subroutine op regel 300 of 310 maar natuurlijk ook door Uw eigen programma. Na het printen van SR\$ wordt niet automatisch een LINE FEED naar de printer gestuurd. Daartoe dient GOSUB 360 die dus niet anders doet dan de CARRIAGE RETURN en de LINE FEED naar de printer geven.

Dat waren de subroutines. De vrije regelnummers zijn voor eventuele latere

uitbreidingen.

De functies die deze routines vervullen mogen dus niet meer rechtstreeks worden gedaan. Uw programma zou op minstens enkele micro's niet goed werken.

BASICODE-2 stelt naast deze subroutines nog enkele eisen:

Ten eerste: Uw programma moet starten op regel 1000. Daartoe staat op regel 10 de opdracht 'GOTO 1000'. Op regel 1000 kan dan een opdracht staan als: 'A=1000 : GOTO 20'. Schrik dan niet: op de regel 20 staat voor sommige computers, die dat nodig hebben een CLEAR A - opdracht waarmee stringruimte wordt vrijgemaakt / andere computers hebben dat niet nodig: die hebben op regel 20 alleen maar staan GOTO 1010

Wat Uw programma na regel 1000 doet is niet van belang. We doen enkele aanbevelingen. Veel programma's beginnen met initialisatie. Dat kan zijn: allerlei DIM-opdrachten, enkele variabelen op een beginwaarde zetten, met READ allerlei DATA inlezen, uitleg geven aan de gebruiker, enzovoort. Aangezien al deze dingen door hun aard slechts eenmaal gedaan worden, verdient het aanbeveling al deze opdrachten aan het eind van het programma te zetten. Op regel 1010 staat dan bij U bijv. GOTO 8000. Vanaf regel 8000 staat de initialisatie en de laatste regel daarvan is dan weer GOTO 1020. Op die regel 1020 begint dan de kern van het programma.

Een tweede advies: plaats alle DATA-regels weer na de initialisatie.

Een derde tip is om daarna dan weer de benodigde REM-opdrachten met programma-uitleg te geven, afgesloten met REM's, waarin Uw naam en adres staan vermeld. Iemand die het programma LIST't ziet dan vanzelf de laatste 20 regels, waarin juist staat hoe het programma heet en wie het heeft gemaakt en wanneer. Bekijkt U gewoon eens een aantal nieuwe programma's. om dit in de vingers te krijgen.

Tot slot: de benodigde subroutines worden U door het nieuwe BASICODE leesprogramma geleverd. Hoe dat precies gaat staat in dat leesprogramma, neemt U het zorgvuldig op wanneer het wordt uitgezonden. Als het opnemen niet lukt kunt U na de uitzending een listing opvragen bij HOBBYSCOOP, Postbus 1200, 1200 BE Hilversum.

Tot zover de uitleg, waarbij zij vermeld dat het vrij van copyright is.

In de volgende IMPULS zal H. Radius ingaan op het tekstopmaakgedeelte van die HOBBYSCOOP-uitzendingen.

Deze uitleg kon hier worden afgedrukt dankzij het werk van H. Radius.

"GAME OF LIFE" op z'n BOOLEAANS voor ZX-80, 4 K ROM, 1K RAM

De ZX-81 kan een heleboel, echter op één punt is de 'echte' ZX-80 met de integer BASIC verre de baas. Alleen machines als PET, VIC en Acorn Atom zijn op dat punt zijn gelijke. Zelfs de Apple integer BASIC laat het afweten.

Ik doel hier op de manier waarop de functies AND, OR en NOT werken bij de ZX-80, dat is echte BOOLE-algebra.

Neem: 10 AND 7, de ZX-81 geeft 10, de ZX-80 geeft 2;

7 AND 10, de ZX-81 geeft 7, de ZX-80 geeft 2;

Het antwoord 2 is juist, wat aldus blijkt:

schrijf de som binair:

$$\begin{array}{r} 1010 \quad (=10) \\ 0111 \quad (=7) \\ \hline 0010 \quad (=2) \end{array} \text{AND}$$

Bij de OR-functie doet de ZX-81 het al niet veel beter:

neem: 10 OR 7, de ZX-81 geeft 1, de ZX-80 geeft 15;

of: 7 OR 10, de ZX-81 geeft 1, de ZX-80 geeft 15.

Binair:

$$\begin{array}{r} 1010 \quad (=10) \\ 0111 \quad (=7) \\ \hline 1111 \quad (=15) \end{array} \text{AND}$$

Zo is het ook bekend, dat bij de ZX-81 NOT 0 = 1, bij de ZX-80 is dat -1

Welnu, in BOOLE-algebra is dat laatste juist: $0_d = 0000 \ 0000 \ 0000 \ 0000_2 = 0$
 $\text{NOT } 0 = 1111 \ 1111 \ 1111 \ 1111_2 = -1$

Er is met de ZX-80 ook een XOR (exclusive OR) te maken:

waarheids-	A	1100	**	probeer de formule:
tabel:	B	1010	**	A XOR B = (A OR B) AND NOT (A AND B)
	A XOR B	0110	**	Vul maar eens in:
			**	1 XOR 1 = (1 OR 1) AND NOT (1 AND 1) =
			**	1 AND NOT 1 = 1 AND 0 = 0
			**	1 XOR 0 = (1 OR 0) AND NOT (1 AND 0) =
			**	1 AND NOT 0 = 1 AND 1 = 1
			**	0 XOR 1 = (0 OR 1) AND NOT (0 AND 1) =
			**	1 AND NOT 0 = 1 AND 1 = 1
			**	0 XOR 0 = (0 OR 0) AND NOT (0 AND 0) =
			**	0 AND NOT 0 = 0 AND 1 = 0

De antwoorden kloppen met de waarheidstabel, dus moet de formule ook wel goed zijn.

In het programma staat ook een XOR-functie en wel op regel 40. De formule is echter iets anders gemaakt, een mengsel van BOOLE- en gewone algebra.

Er is een - teken gebruikt: $A \text{ XOR } B = (A \text{ OR } B) - (A \text{ AND } B)$

Ingevuld:

$$\begin{array}{l} 1 \text{ XOR } 1 = (1 \text{ OR } 1) - (1 \text{ AND } 1) = 1 - 1 = 0 \\ 1 \text{ XOR } 0 = (1 \text{ OR } 0) - (1 \text{ AND } 0) = 1 - 0 = 1 \\ 0 \text{ XOR } 1 = (0 \text{ OR } 1) - (0 \text{ AND } 1) = 1 - 0 = 1 \\ 0 \text{ XOR } 0 = (0 \text{ OR } 0) - (0 \text{ AND } 0) = 0 - 0 = 0 \end{array}$$

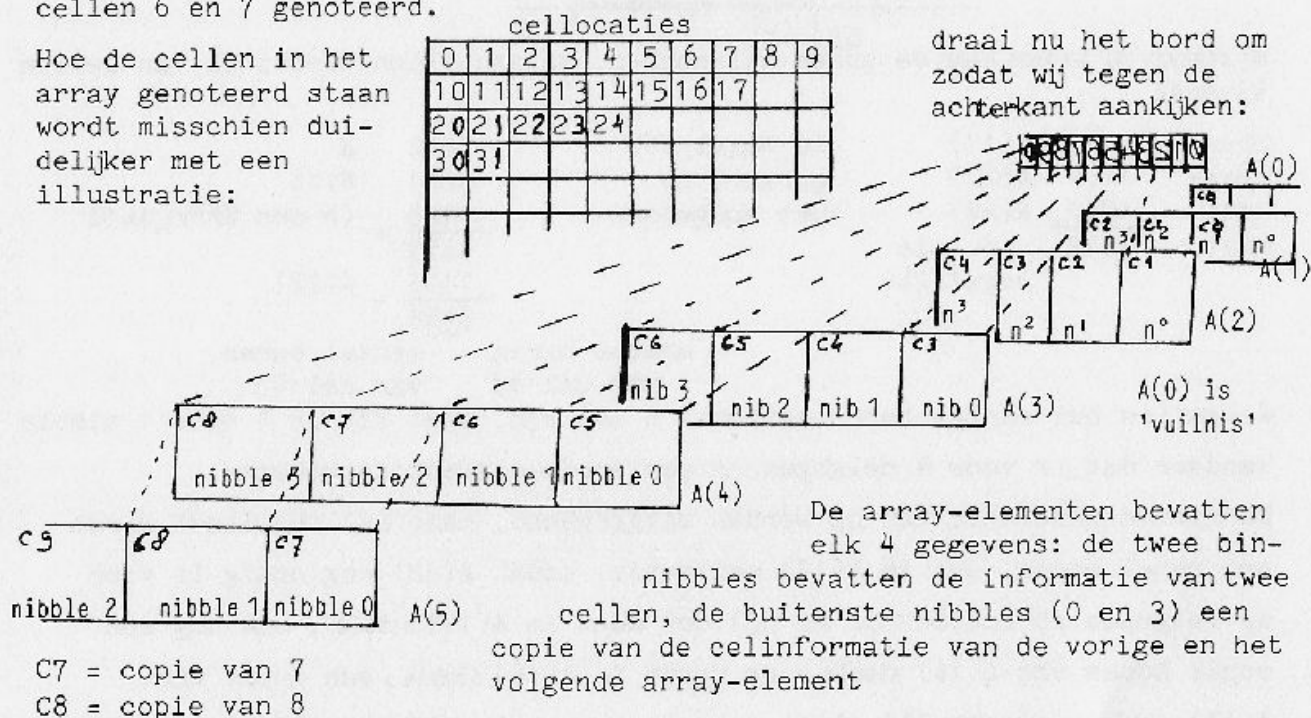
Zoals U ziet klopt ook deze formule met de waarheidstabel.

Laat U niet misleiden door de vorm $16*16**(X \text{ AND } 1)$ in regel 40. Als wij bijvoorbeeld cel 7 willen veranderen, dus $X=7$, dan levert deze vorm op:

$$16*16' = 256 = 0000\ 0001\ 0000\ 0000_2 \ (n^3; n^2; n^1; n^0)$$

Zo hebben wij dus het eerste bit van nibble 2 te pakken (n^2), en deze bit staat in array-element $A(\frac{1}{2}X+1)$, dat is $A(4)$ en daarin staan nu net de cellen 6 en 7 genoteerd.

Hoe de cellen in het array genoteerd staan wordt misschien duidelijker met een illustratie:



Wie het tot nu toe volkomen duister is waar dit verhaal over gaat verwijs ik naar andere artikelen over LIFE (van J.Conway) die de bedoeling van het spel uitleggen. Het probleem van het spel is namelijk:

er wordt een eerste generatie cellen opgezet ('de oude wereld')
uit het aantal burens van elke cel is te berekenen welke cel

in de volgende generatie geboren wordt, overleeft of sterft.

De moeilijkheid is nu: als een nieuwe cel uitgerekend wordt kan deze niet meteen in het oude generatie-bord ingevuld worden, omdat bij de berekening van nog onbehandelde plaatsen de buurcellen ondertussen veranderd kunnen zijn. Een oplossing is: Reken uit in de 'oude wereld' en zet het resultaat in een 'nieuwe wereld'. Als alles dan klaar is, vervang dan de 'oude wereld' door de 'nieuwe wereld' en zet het resultaat in het display. De 'nieuwe wereld' is dan weer vrij voor het uitrekenen van een nieuwe generatie.

Het nadeel van deze methode is voor de 1 K RAM, dat dit veel geheugenruimte (twee beelden) kost. Hier is het anders gedaan.

De cellen worden - twee tegelijk - weergegeven door een array-element, dat 4 nibbles bevat. Waarom niet elke bit van het array-element gebruikt voor een cel? Dat is 15 cellen voor een element ($d_0 - d_{14}$), d_{15} is tekenbit(niet gebruikt).

Dit copieer-gedoe is gedaan om een makkelijke methode te krijgen voor het uitrekenen van het aantal burens van twee cellen tegelijk in een bewerking, zie programma-regels 240 - 260

Hoe het programma te gebruiken?

Na RUN vraagt het om DATA.

Nummers van 0 tot 70 zetten het celpatroon op, hetzelfde nummer nog een keer verwijdert de cel weer (70 t/m 79 worden overschreven).

Om te zien wat je gemaakt hebt, typ Z (van ZIEN) in.

Om de volgende generatie te krijgen, tik V (van VOLGENDE) in.

Om het programma te stoppen, typ S in.

Tussen regel 340 en 380 was eerst een routine om het (sc)rollen te voorkomen, maar met (sc)rollen vind ik het eigenlijk nog leuker, zodoende.

Wat me eigenlijk verbaasde was, dat ondanks de lange berekeningen, het programma vrij snel is, vergeleken met het overeenkomstige programma dat in het boek "30 programs for the ZX-80" staat.

En ... heb ik te veel gezegd wat betreft de superieure ZX-80 4 K ROM BASIC? Maar één troost: de ZX-81 kan dit ook wel, echter alleen in machinetaal als het om BOOLE-algebra gaat. De XOR-functie kan dan zelfs met één instructie.

H. Radius

```

1 LET M=40
2 LET K=5
3 LET G=0
4 LET V=-1
5 LET Z=-2
10 DIM A(M+K)
20 INPUT X
25 CLS
30 IF X < 0 THEN GOTO -X*200
De nu komende echte BOOLEANSE operatie werkt als een
flipflop op de nibble van de gekozen locatie en laat de
andere onaangetast.
40 LET A(X/2+1)=(A(X/2+1) OR 16*
16*(X AND 1))-(A(X/2+1) AND 16*16
**(X AND 1))
Nu moet de veranderde nibble nog in het naastliggende array-
element gecopieerd worden. Om het programma wat kleiner te
houden copieren we beide burens, weliswaar één overbodig,
maar alle .....
50 FOR L=0 TO 1
60 LET A(X/2+2*L)=(A(X/2+1) AND
D NOT 4096**((1-L))+(A(X/2+1) AND
16*16**L)*256**((1-L)/256**L
70 NEXT L
80 GOTO 20
4096 = 0001 0000 0000 0000b; NOT 4096 = 1110 1111 1111 1111b
Een AND-operatie met dit masker maakt bit 12 altijd 0 voor-
dat de locatie wordt gecopieerd, Nu eerst de displayroutine:
400 PRINT "GENERATIE ";G
410 FOR X=1 TO M
420 FOR L=1 TO 2
430 PRINT CHR$(128 OR -20*(A(X
) AND 16**L)>0)):
Zie hier hoe gemakkelijk je op één bit kunt testen van een ar-
ray-element om te weten of daar een cel zit of niet.
440 NEXT L

```


bit gezet, dat betekent dood of leeg blijven.
 Dit zijn de regels van het spel 'LIFE', maar iets anders
 geformuleerd dan in het Amerikaanse tijdschrift "SYNCR".
 Nu de eerste generatie in A(X-K) staat moet nibble 1
 nog gecopieerd worden in de voorgaande buur (dat is
 A(X-K-1)) en vanuit die buur moet er ook nog gecopieerd
 worden naar nibble 0

```

310 FOR L=0 TO 1
320 LET A(X-K-L)=A(X-K-L)+(A(X-
K-1+L) AND 256/16**L*256**L/256*
*(1-L)
330 NEXT L
340 NEXT X

380 LET G=G+1
..... en daar is dan de nieuwe generatie .....
```

```

450 IF (X/K)*K=X THEN PRINT
460 NEXT X
470 GOTO 20

De opzetroutine kan nu alvast gecontroleerd worden
op zijn goede werking. Hier komt de rest (rollend display)
200 FOR X=1 TO 2*K
210 LET A(M-K+X)=A(X)
220 NEXT X

Het berekenen van het aantal burens:
230 FOR X=K+1 TO M
240 LET N=A(X-K)+A(X)+A(X+K)
250 IF N=0 THEN GOTO 310
260 LET N=N+N/16+(N AND 4095*1
6-A(X))
```

Regel 250 was om nutteloos werk te vermijden.

.. 'AND 4095 maakt nibble 3 0 vóór de vermenigvuldiging
 om mogelijke overflow te vermijden.

De aantallen burens van twee locaties zijn nu in de nibbles
 1 en 2, dus nu kan geboorte, leven of dood van de plaat-
 sen, vertegenwoordigd door A(X) berekend worden, maar wij
 zetten de nieuwe generatie één rij hoger, omdat A(X) nog
 nodig is bij het berekenen van de volgende rij.

Daarvoor waren ook de regels 200-220 nodig. Het effect is
 een (sc)rollend plaatje.

```

270 LET A(X-K)=0
280 FOR L=1 TO 2
290 LET A(X-K)=A(X-K)-16**L*((N
AND 14*16**L)=2*16**L AND (A(X)
AND 16**L)=16**L OR (N AND 15*1
6**L)=3*16**L)
300 NEXT L
```

In bovenstaande loop bepalen wij eerst of nibble 1, bit 4,
 moet worden gezet of niet. Dankijken wij naar nibble 2, bit 8.
 De regels zijn hier: a) alleen als er 2 of 3 burens zijn en er
 was reeds een cel, dan wordt de bit gezet op 1;
 b) zijn er 3 burens, dan wordt de bit gezet;
 dus als a) en b) onwaar zijn (beiden), dan wordt er geen

12 JOCKEYS voor de ZX80 4KROM 1 KRAM

Dit is een soort puzzle die het volgende inhoudt:

Er zijn 12 JOCKEYS, waarvan bekend is, dat ze precies even zwaar zijn, behalve één..., die is nl. óf zwaarder, óf lichter.

De jockeys hebben de nummers 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B en C.

De oorspronkelijke versie bestond nu hieruit dat je met behulp van 3 wegingen op een wip de juiste man eruit moest zien te halen en tevens wist te zeggen of hij zwaarder dan wel lichter was.

Hier is nu een computerversie van gemaakt met een kleine wijziging om het wat spannender te maken.

Na elke weging mag je nu een gok doen, er is dan een kans dat je het in één keer raadt.

De bonuspunten worden als volgt berekend:

in 1 keer goed = 100 punten

2 keer goed = 50 punten

3 keer goed = 25 punten

4 keer goed = 12 punten

5 keer goed = 6 punten

6 keer goed = 3 punten

7 keer goed = 1 punt

8 keer goed = geen punt.

Mocht de bonus nihil zijn, dan geeft de computer zelf de oplossing.

Ook wordt het aantal ronden bijgehouden en de totaalscore.

Wil je het programma stoppen, dan geef je na opgave van de score i.p.v. enkel N/L een willekeurig teken in.

Het ingeven van het plaatsen van jockeys op de wip gaat als volgt:

b.v. de jockeys 1, 2 en 3 links en de nr.'s A, B en C rechts op de wip wordt ingegeven als

123-ABC (dus minteken ertussen).

Als er gegokt (of zéker geweten) wordt, dan geef je in...

b.v. 6L als bedoeld wordt: jockey nr. 6 is lichter en

b.v. AZ als bedoeld wordt: jockey nr. A is zwaarder.

Het programma is niet 'foolproof', dus wel opletten.

AZWAARDER of 6LICHTER, dus voluit, accepteert hij wél, maar aan elkaar geschreven.

De PRINT-statements in 290, 300 en 310 kunnen als volgt gemaakt:

tik in 1 PRINT " _--1234"

Het midden van de wip is nu adres 16431 in de RAM.

Doe nu rechtstreeks POKE 16431,128 POKE 16432,131

POKE 16433,131 POKE 16434,131

Nu is er ontstaan: 1 PRINT " _--1234"

EDIT 1 nu tot 290 en klaar is kees.

Zoiets kan nu ook met de twee andere PRINT-statements.

Dit is het programma, het is getest en het werkt.

```

1 LET T=0 ; totalpunten
2 LET R=0 ; rondeteller
5 Cls
8 Let R=R+1
10 Let B=100 ; bonuspunten
20 Let J$=Chr$(Rnd(12)+28) ; random-jockey
30 Let G=2xRnd(2)-1 ; 3=zw. 1=lichter
40 Print "12 JOCKEYS: NRS-12345678ABC-"
45 Print "nummers links-rechts op wip"
50 Input G$
55 Print G$
60 Let W=0 ; wipindicator
70 If G$="" Then Goto 110
80 Let W=W+2x(W+1)x(Code(G$)=220)
   -Gx(code(G$)=Code(J$))-2xNot(Code(G$)=Code(J$))
90 Let G$=TL$(G$)
100 Goto 70
110 Gosub 300+10x(W<0)-10x(W>0)
120 Print "GOK: BV.AZ of 6L>";
130 Input G$
135 Print G$
140 If Code(G$)=Code(J$) and Code(TL$(G$))
   =49-14x(G=3) then Goto 200
150 Let B=B/2
160 If B 0 then Goto 45
200 Print "HET WAS"; J$;(Chr$(49-14x(G=3));"..";B;" PUNT(EN)VERDIEND"
205 Let T=T+B
206 Print "TOTAAL=";T;" RONDE ";R
210 Input G$

```

```

230 If G$="" then Goto 5
240 Stop

290 Print "  _ _ _ _ _"
295 Return
300 Print " _ _ _ _ _"
305 Return
310 Print " _ _ _ _ _"
315 Return

```

Dit is de versie waarin de computer het raadsel aan jou opgeeft.

Als je het goed speelt, moet je het minstens in drie keer raden per ronde weten, dus doe je best.

De rollen kunnen ook omgedraaid, maar dat wordt een ander programma.

Over het programma nog het volgende:

In regel 80 wordt de wip-indicator uitgerekend door van de G\$ telkens het volgende karakter te onderzoeken en de gewichtswaarde of de betekenis ervan aan W toe te voegen.

Wordt uiteindelijk W positief, dan is rechts zwaarder,
wordt W tenslotte nul, dan is de wip in evenwicht en
wordt W tenlaatste negatief, dan is links zwaarder.

W kan veranderen door 3 termen, hier genoemd a, b en c.

```

b=: -Gx(Code(G$)=Code(J$))          betekent
    if code(G$)=code(J$) then let W = W + G
c=: -2xNot(Code(G$)=Code(J$))        betekent
    if code(G$) ≠ code(J$) then let W = W + 2
                                   (ook voor het minteken wordt dus 2 bijgeteld).
a=: +2x(W+1)x(Code(G$)=220)          betekent
    if code(G$) + code "-" then let W = -W - 2
                                   (de extra -2 is nodig omdat bij C deze er te-
                                   veel bijkomt).

```

Een vreemde zaak vind ik dat, als je de term a achteraan zet,

dus $Let W = W + b + c + a$,

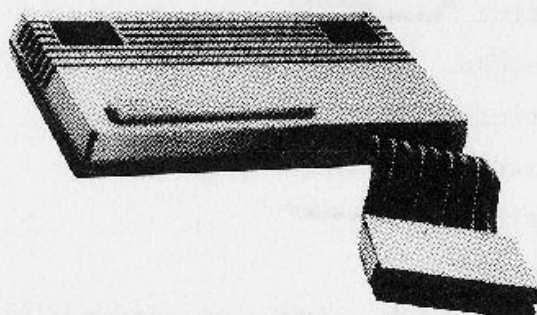
er geheel verkeerde resultaten komen.

Waarschijnlijk wordt dan niet de oorspronkelijke W, maar een half geëvolueerde W genomen. Uitkijken dus.

In regel 200 komt de uitdrukken Chr(49...J)$ tot het printen van een "L" of een "Z".

H. Radius

ZX SPECIAL PRODUCTS

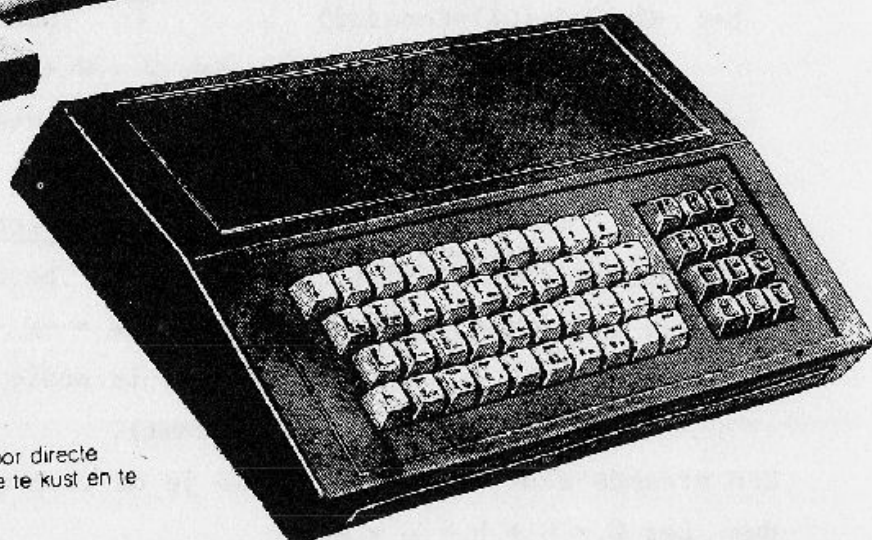


Bouwsteenuitbreiding voor ZX en Spectrum tot 1 Megabyte!!!

PERSONA ZX81	f 162,-	DROM	f 212,-
PERSONA SPECTRUM	f 243,-	TOOLKIT	f 119,-
hierna kunt u uitbouwen met:		PERICON a I/O	f 150,-
MINIMAP	f 193,-	PERICON b I/O	f 208,-
RAM 08	f 131,-	PERICON c I/O	
RAM 16	f 144,-	centronics	f 224,-
RAM 64	f 395,-	SONUS	f 162,-

VERWACHT: FLOPPY-DISCOCONTROLLER

KORTING voor combinatieaankopen op aanvraag.



Wij leveren een uitgebreide serie Printers voor directe
AANSLUITING OP ZX81 en Spectrum. Software te kust en te
keur en verdere randapparatuur.

KOMIN B.V. VRAAG ONZE UITGEBREIDE
INFORMATIE!!! GRATIS.

KANTOORADRES: BORDEAUXLAAN 86, 5627 GW Eindhoven
Postadres: Postbus 8100, 5601 KC Eindhoven
Tel. 040-428179 ABN Veidhoven Rek. nr. 52.82.75.615

f228,-

KEYBOARD

Voor Inbouw van ZX81
of Spectrum,
Extra cijferblok.

Spectrum Bestelnr. 002
ZX 81 Bestelnr. 001

Bestellingen d.m.v. vooruitbetaling of cheques
met vermelding bestelnr. Verzendkosten f 6,50
Prijzen inclusief BTW.

```

***>*****
*** REKEN-OEFENINGEN ***
***      VOOR HET      ***
***      JONGE VOLKJE  ***
*** *****
Hoe is jouw naam Piet
Wil je uitleg ?  J N

BESTE PIET
Vij gaan samen eens wat sommen
Maken jij kunt straks een getal
Opgeven tussen 10 en 90.
Je krijgt dan vermenigvuldin-
gingen op het scherm met getallen
Dit maximaal 9 hoeser zijn dan
het opgegeven getal.
Je hoeft niet bang te zijn om
fouten te maken, want ik zal je
met raad en daad terzijde staan.
Je zult zien dat het dan van een
leien dakje gaat en dat je met
plezier opgaven zult oplossen.
Druk op NEULINE

76*42=      3456
NIET GOED, IK HELP JE.
70*42=      2890
Vij gaan verder splitsen
70*40=      2800
70*2=       140
2800+140=   2940

GOED, NU DE VOLGENDE STAP
6*42=       252
Vij gaan weer verder splitsen
6*40=       240
6*2=        12
2*0+12=     252
NU ALLES BIJ ELKAAR Optellen
2940+252=   3092
FOUT, OOK DIT GAAN WE WEER SPLITS
EN
2900+200=   3100
40+50=      90
0+2=        2
3100+90=    3190
5190+2=     3192
GOED, DUS
76*42=      3192

1000 REM    ==REKENOEFENINGEN==
1010 REM    ==HENK EN MEINDERT
1020 REM    HIELKE==
1030 REM    ==ENGELANDLARM 24==
1040 REM    ==9501 AR STADSKANAAL==
1050 REM    ==GEMAAKT OP VICE20
        BASIC02==
1060 REM    ==RANGEPAST AAN BASI-
1070 REM    CODE OKTOBER 1982==
        ==BEWERKT VOOR ZX 81
        DOOR M. I.F.FORGER
1080 REM    ==IECERE BELUKENIS MET
        BESTANDE OF TOT HET
        VERLEDEN BEHORENDE
        VARIABELEN IS TOEVALLIG

```



```

1180 GOSUB 2210
1190 GOSUB 1850
1200 GOSUB 2210
1210 PRINT "IK WENS JE VEEL SUCC
1220 N$=INT (RND*(Q)+10
1230 PRINT "WELK GETAL "
1240 PRINT Q
1250 IF Q<10 OR Q>90 THEN GOTO
1260
1270 LET U=INT (RND*(Q)+10
1280 LET V=INT (RND*(Q)+10
1290 LET W=STR$ (U)
1300 GOSUB 2210
1310 PRINT U;"*";V;"="";TAB 15;
1320 PRINT T
1330 IF T<U*V THEN GOTO 1550
1340 GOTO 1820
1350 PRINT "NIET GOED, IK HELP JE
1360
1370 GOSUB 2250
1380 PRINT LU;"*";V;"="";TAB 15;
1390 INPUT E
1400 IF E<LU*V THEN GOTO 1380
1410 GOTO 1470
1420 PRINT "WIJ GAAN VERDER SPLI
1430 TSEN"
1440 GOSUB 2300
1450 PRINT LU;"*";LV;"="";TAB 15;
1460 INPUT O
1470 IF O<LU*LV THEN GOTO 1400
1480 GOSUB 2400
1490 PRINT LU;"*";RV;"="";TAB 15;
1500 INPUT P
1510 IF P<LU*RV THEN GOTO 1430
1520 PRINT Q;"*";P;"="";TAB 15;
1530 INPUT E
1540 IF E<Q*P THEN GOTO 1450
1550 PRINT
1560 PRINT "GOED, NU DE VOLGENDE
1570 STAP"
1580 GOSUB 2350
1590 PRINT SU;"*";V;"="";TAB 15;
1600 PRINT K
1610 INPUT K
1620 IF K<SU*V THEN GOTO 1520
1630 GOTO 1620
1640 PRINT "WIJ GAAN NEER VERDER
1650 TSEN"
1660 GOSUB 2300
1670 PRINT SU;"*";LV;"="";TAB 15;
1680 PRINT F
1690 INPUT F
1700 IF F<SU*LV THEN GOTO 1540
1710 GOSUB 2400
1720 PRINT SU;"*";RV;"="";TAB 15;
1730 PRINT H
1740 IF H<SU*RV THEN GOTO 1570
1750 PRINT K
1760 F;"*";H;"="";TAB 15;
1770 INPUT K
1780 IF K<F+H THEN GOTO 1550
1790 PRINT "NU ALLES BIJ ELKAAR
1800 TTELLEN"
1810 PRINT E;"*";K;"="";TAB 15;
1820 PRINT R
1830 IF R<E+K THEN GOTO 1650
1840 GOTO 1790
1850 PRINT "FOUT, OOK DIT GAAN WE
1860 NEER SPLITSSEN"
1870 GOSUB 2450
1880 PRINT BE;"*";LK;"="";TAB 15;
1890 INPUT LB
1900 IF LB<BE+LK THEN GOTO 1670
1910 GOSUB 2550
1920 PRINT CE;"*";CK;"="";TAB 15;
1930 INPUT CC
1940 IF CC<CE+CK THEN GOTO 1780
1950 PRINT

```

[illegible]

Het gebruik van het YAHTZEE-spel uit IMPULS 2 blijkt veel makkelijker te gaan indien men daarbij kan beschikken over de score-kaart voor de punten-telling. Daarom drukken wij die hier alsnog af. Er wordt gewerkt aan een gebruikersvriendelijker versie van YAHTZEE.

SCORE CARD



MINIMUM REQUIRED FOR BONUS		HOW TO SCORE		GAME					
1 = 2 = 3 = 4 = 5 = 6				1	2	3	4	5	6
Aces	• • • • • = 3	COUNT AND ADD ONLY ACES							
Twos	• • • • • = 6	COUNT AND ADD ONLY TWOS							
Threes	• • • • • = 9	COUNT AND ADD ONLY THREES							
Fours	• • • • • = 12	COUNT AND ADD ONLY FOURS							
Fives	• • • • • = 15	COUNT AND ADD ONLY FIVES							
Sixes	• • • • • = 18	COUNT AND ADD ONLY SIXES							
TOTAL = 43			→						
Bonus IF 43 OR OVER			SCORE 35						
TOTAL OF UPPER HALF			→						

3 of a kind	ADD TOTAL OF ALL DICE								
4 of a kind	ADD TOTAL OF ALL DICE								
Full House	SCORE 25								
Sm. Straight (any 5)	SCORE 30								
Lg. Straight (any 6)	SCORE 40								
YAHTZEE (5 of a kind)	SCORE 50								
Chance	SCORE TOTAL OF ALL 5 DICE								
TOTAL OF LOWER HALF	→								
TOTAL OF UPPER HALF	→								
GRAND TOTAL	→								

```

REM -TIENTAL-VAN-U
LET Y$=VAL$(1 TO 1)
LET UV=VAL (Y$) #10
LET URN
REM -EENHEID-VAN-U
LET X$=VAL$(2 TO 1)
LET SU=VAL X$
LET URN
REM -EENHEID-VAN-U
LET N$=VAL$(2 TO 1)
LET RU=VAL N$
LET URN
REM -HONDERDTAL-VAN-E
LET E$=" "+STRA$(E)
LET U$=E$(1 TO (LEN E$-2))
LET BE=VAL U$ #100
REM -HONDERDTAL-VAN-K
LET K$=" "+STRA$(K)
LET R$=K$(1 TO (LEN K$-2))
LET RK=VAL R$ #100
LET URN
REM -TIENTAL-VAN-E
LET S$=E$( (LEN U$+1) TO (LE
CE=VAL S$ #10
REM -TIENTAL-VAN-K
LET T$=K$( (LEN R$+1) TO (LE
CK=VAL T$ #10
LET URN
REM -EENHEID-VAN-E
LET P$=E$(LEN E$ TO 1)
LET SE=VAL P$
REM -EENHEID-VAN-K
LET Q$=K$(LEN K$ TO 1)
LET RK=VAL Q$
LET URN
SAVE "REKENEN"

```

BESCHRIJVING IMPULSOFT-1 CASSETTE

De IMPULSOFT-1 cassette bevat 12 programma's welke hier kort besproken worden.

Alle programma's kunnen gestart worden met RUN tenzij anders is aangegeven.

1. READ-routine

Deze routine zet in regel 1 een machinecode waarmee de READ-functie wordt gesimuleerd.

2. DATA-routine

Starten met: Rand USR 16514.

In regel 1 moet de machinecode van de READ-routine staan. De rest van het programma is een voorbeeld van "DATA-statements". Deze worden ingelezen in A\$.

3. Lineaire regressie

Het programma vraagt om de eerste X-waarde, daarna de bijbehorende Y-waarde. Vervolgens de 2^e X-, 2^e Y-waarde, etc. De getalinvoer wordt gestopt door alleen een New-Line te geven. Het programma berekent nu de meest waarschijnlijke rechte door de punten en geeft van deze rechte: helling (slope), het snijpunt met de Y-as (intercept) en de correlatiecoëfficiënt, deze laatste zegt iets over het afwijken van de punten ten opzichte van de berekende lijn (hoe dichter bij de waarde: 1, hoe beter). Daarna vraagt het programma om een X- of Y-waarde in de vorm: X=... of Y=... en berekent dan de bijbehorende Y- of X-waarde van de berekende rechte lijn.

4. Faculteit berekening

Gevraagd wordt om een getal. Hiervan wordt de faculteit berekend, bijvoorbeeld: getal: 4 $4! = 4 \times 3 \times 2 \times 1 = 24$. Dit programma bevat een aardige routine om getallen van meer dan 8 cijfers volledig op het scherm te zetten.

5. Supermind

Een vorm van Mastermind met 4 speelsterktes. Na 18 beurten volgt de oplossing. Deze is na een BREAK ook te bereiken met GOTO 284.

6. Cassette index printen

Dit programma vraagt na het cassettenummer om tweemaal 8 programmanamen. Deze namen mogen maximaal 12 letters lang zijn. Als de index goed is moet u zelf COPY geven.

7. Kiengetallen trekken

Door op New-Line te drukken wordt een nieuw getal getrokken. De toets: S met New-Line geeft een schermkopie en start een nieuwe ronde. De toets: A met New-Line geeft alleen een schermkopie, het trekken kan daarna voortgezet worden.

8. Nuclear Power Plant

Opmerking: Regel 410 moet zijn: LET RH=0.

In de instructies (toets I) staat hoe de parameters gewijzigd kunnen worden. Met behulp van toets: A volgt een berekening en komt het resultaat daarvan op het scherm.

9. Credietberekening

De cursor: > is met toets 6 en 7 te besturen. Na toets: I kunt U een nieuwe waarde invoeren. Met toets: 2 wordt de onbekende grootte uitgerekend. Mocht er een foutmelding optreden dan kan U herstarten met: GOTO 2000 of met: RUN.

10 Yahtzee

Dit programma is zelfstartend. Per beurt kun je driemaal gooien. In de eerste zes beurten geef je op welke ogen geteld moeten worden. In beurt 7 t/m 13 kun je kiezen of je op: 3 of 4 van 1 soort, 2 van de ene en 3 van de andere soort (full house), kleine of grote straat (4 of 5 run), Yahtzee of kans (chance) speelt. Er zijn twee spelsoorten: met 2 spelers of met 1 speler. De telling wordt op het scherm bijgehouden.

11. Torens van Hanoi

Dit programma is zelfstartend. De uitleg verschijnt op het scherm.

12. Maze monster

Eerst worden de afmetingen van het doolhof gevraagd. Nadat de computer het doolhof gemaakt heeft, dit kan bij grote afmetingen 1 à 2 minuten duren, krijg je het te zien. Als het "blind" geworden is verschijnt je eigen positie (□), de uitgang en het monster (*). Met behulp van de toetsen: 5, 6, 7 en 8 kun je jezelf naar de uitgang manoeuvreren. Het is de bedoeling het monster te ontwijken. Elke keer dat je een muur ontmoet verschijnt deze op het scherm. Een ontmoeting met het monster beëindigt het spel, waarna het doolhof opnieuw verschijnt voor een volgende poging.

De IMPULSOFT-2 cassette bevat 5 voor ZX-81 - BASIC bewerkt. Zij worden allen geladen met LOAD "P". ^{programma's}

1. CIDER RECEPTUREN

* Drankproductie

(laadtijd 6'40", + 15 K)

In ruim 200 regels BASIC met veel tekst wordt voor U uitgerekend hoe naar eigen wens cider kan worden gemaakt. Enige kennis van scheikunde helpt daarbij.

Door reSAVEN met GOTO 2700 N/L kan het programma 'auto-run' worden gemaakt onder de naam "CIDER".

2. HYPOTHEEKKOSTEN

* Kostencalculatie

(laadtijd 6'45", +15 K)

In ruim 200 regels BASIC met veel tekst wordt voor U uitgerekend de kosten van een hypotheek op annuïteitenbasis of levensverzekering. Rente, aflossing, restant schuld en het verschil in kosten tussen beide vormen van hypotheek worden per jaar bepaald, rekening houdend met belasting en rente-aspecten.

Via GOTO 9990 N/L kan door reSAVEN het programma 'auto-run' worden gemaakt, naam "HYPOTHEEK".

Via GOTO 9990 N/L kan door reSAVEN het programma 'auto-run worden gemaakt, naam "HYPOTHEEK".

3. VERGELIJKINGEN OPLOSSEN

* Wiskunde

(Laadtijd 7'10", + 16 K)

In ruim 100 regels BASIC worden vergelijkingen met 2 of 3 onbekenden opgelost met de regel van Cramer. De listing is gepubliceerd in IMPULS 2, (pag 02-31/32).

reSAVEN met GOTO 1700 N/L maakt het programma 'auto-run', met de naam "VERGELIJKINGEN".

4. AFSTANDEN BEPALEN.

* Wiskunde

(Laadtijd 7'10", + 16 K)

In ruim 150 regels BASIC wordt de afstand berekend tussen twee plaatsen op aarde. De listing is te vinden in IMPULS 2 (pag 02-32/33).

'Auto-run is het programma te maken met GOTO 1900 N/L, waarna reSAVEN onder de naam "AFSTANDEN".

De IMPULSOFT-3 cassette bevat 10 programma's, die allen 'auto-run' zijn.

LOAD "0" MENU-PROGRAMMA

* Beginners

(Laadtijd 0'55", ruim 1 K)

Met dit programma op het begin van Uw cassettes kunt U meerdere programma's op de cassette zetten, waarbij U met dit eerste programma een duidelijk overzicht heeft van wat er op die cassette staat.

Tevens kunt U vanuit dit programma meteen doorstarten naar het gewenste programma.

LOAD "1" AUTOMATISCH STARTEN

* Beginners

(Laadtijd 0'35", ruim $\frac{1}{2}$ K)

Deze routine achter Uw eigen geschreven programma geplaatst zorgt voor automatisch beginnen van het programma als U het heeft geladen en is een makkelijke hulp bij het SAVEN.

LOAD "2" MAANLAND-ER VISUEEL

* Beginners

(Laadtijd 1'45", krap 4 K)

In het Kluwer boek voor de ZX-81 staat een aardig 1 K spelletje genaamd "Maanlander". Dit programma laat zien dat er van zo'n spelletje nog veel meer te maken is door er wat leuke grafische voorstellingen aan toe te voegen.

LOAD "3" ADRESSENPROGRAMMA

* Administratie

(Laadtijd 4'50", + 11 K)

De meeste programma's waarin adressen worden verwerkt zijn gebaseerd op de naam van de betreffende persoon. Echter niet dit programma. Uitgaande van het idee dat U er Uw familie in kwijt wilt, is er hier uitgegaan van het adres waar meerdere personen kunnen wonen. Behalve opzoeken van voor- en achternaam, adres en telefoonnummer kunt U met dit programma ook verjaardagskalenders samenstellen.

LOAD "4" TELLER AMERIKAANS JOKEREN

* Algemeen

(Laadtijd 2'10", ruim 4 K)

Een hulp-tel-programma als er bij U gegokt en gekaart gaat worden om centen (of meer).

Dit programma houdt voor (maximaal) 6 mensen de stand bij tijdens het Amerikaans Jokeren en geeft aan aan wie en hoeveel moet worden uitbetaald.

LOAD "5" REKLAME / LOOPKRANT

* Algemeen

(Laadtijd 2'30", krap 2 K + lange array)

Met dit programma kunt U Uw eigen reclametekst over het scherm laten lopen, zolas U wel eens bent tegen gekomen in etalages en bij kranten.

LOAD "6" KOLENMIJNSCHAT

* Spellen

(Laadtijd 4'10", ruim 6 K)

Een avonturenspeel in het hart van de computer.

Driedimensionaal gaat U op zoek naar de schat, maar overal dreigen gevaren.

Geen mooie plaatjes op het scherm, maar wel duidelijke tekst.

LOAD "7" STADHOUDER

* Spellen

(Laadtijd 3'15", krap 7 K)

In dit programma regeert U 14 jaar een land. D.m.v. beslissingen brengt U het land voor- of tegenspoed en zal het volk wel of niet in opstand komen.

Evenals nr 6 een grandioos tekstspel.

LOAD "8" SALARISPROGRAMMA (explicatie)

* Administratie

(Laadtijd 2'10", 4½ K)

Om de ruimte in het geheugen optimaal te kunnen gebruiken, is dit programma niets meer dan de uitleg voor het volgende programma. Start automatisch door naar "SALARIS".

LOAD "SALARIS"

* Administratie

(Laadtijd 4'40", 10½ K)

Dit programma is geschreven naar aanleiding van de nieuwe berekeningswijze van AOW een LOONBELasting per 1 januari 1983. Indien Uw werknemers een per jaar vastgesteld salaris hebben en U loonbelasting inhoudt volgens de 'witte' tabel, dan is dit programma een grote tijdsbesparing voor U. Het enige wat U hoeft te doen, is eenmaal per jaar de vaste gegevens in te vullen en per maand het bedrag aan AOW en Loonbelasting.

Het programma MAAKT DAN VOOR U een compleet salarisstrookje.

In een programma kunt U maximaal 6 werknemers kwijt. Als U meerdere werknemers hebt gebruikt U dit programma gewoon meerdere malen.

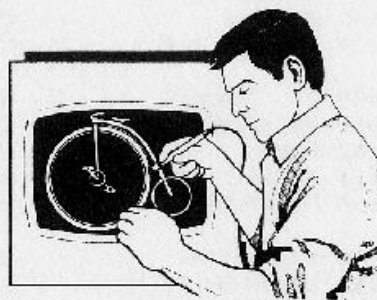
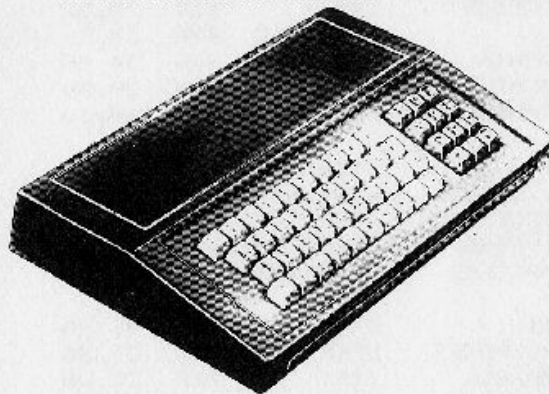
Alle programma's zijn volledig in BASIC geschreven. De programma's 6 en 7 zijn bewerkingen van programma's uit het Kluwer BASIC spellenboek. Wij kunnen U dit boek van harte aanbevelen, indien U geïnteresseerd bent in het zelf schrijven van spelprogramma's, maar ook als U grotere programma's wilt gaan schrijven in BASIC. Zeer leerzaam en in duidelijk Nederlands geschreven.

μS MICRO SOURCE μS

MICRO-ELEKTRONIKA en SOFTWARE

MICRO-SOURCE HEEFT STEEDS HET NIEUWSTE VOOR UW SINCLAIR COMPUTER

 WIJ ZIJN NU OP VASTE TIJDEN BEREIKBAAR, DUS GEEN ANTWOORD MEER
 DT TM VR: 12 - 16 UUR, ZATERDAG: 10 - 16 UUR, ALTIJD ANTWOORD



DE' TRONICS **

TOETSENBORD VOOR DE ZX SPECTRUM F 228.00
 LICHTPEN VOOR DE SPECTRUM (high res tekenen op scherm) F 117.00

EUROELECTRONICS **

ZX LPRINT SPECTRUM CENTRONICS INTERFACE softw in epron F 300.00

KEMPSTON ***

CENTRONICS INTERFACE VOOR DE SPECTRUM softw cassette F 250.00
 COMPELION PRO JOYSTICK MET SPECTRUM INTERFACE F 125.00

C SYSTEMS **

Modulair input / output systeem voor de zx spectrum

ADAP	DUBBELE CONNECTOR VOOR DE SPECTRUM	F	40.00
BBP3	GEBUFFERD 3 SLOTS MOEDERBOARD	F	197.50
BP24	4 SLOTS BUS VERLENGING	F	139.50
I/O	2 X 8 BITS PROGRAMMEERBAAR PARALLEL INPUT/OUTPUT	F	163.00
CENT	CENTRONICS UPGRADE VOOR PARALLEL KAART	F	24.50
232D	TWEE KANAALS PROGRAMMEERBAAR RS 232	F	189.00
PROT	DUBBELZIJDIG DOORCETALTSEERD EXPERIMENTEER PRINT	F	75.00

BINNENKORT: VIDITEL INTERFACE, DISK DRIVE I.F., HARTSLAG MONITOR

***** VOOR DE SPECTRUM EX ZX 81 *****

FINELATA **

SURFAKSGENERATOR Allophonen met SP 0256 IC F 157.50
 SOUNDGENERATOR met AY 3-8912, 2 kanaals, incl I/O poort F 185.00

***** VOOR DE ZX 81 *****

MEMOTECH **

16 K RAM	F 128.50	MEMOCALC	F 162.50	HIGH RES GRAPH	F 175.00
32 K RAM	F 195.00	MEMOEXT	F 162.50	CENTRONICS I F F	155.00
64 K RAM	F 297.50	ASSEMBLER	F 162.50	CENTR KABEL	F 55.00
KEYBOARD MET BUSBUFFER	F 199.00				
NIEM: RS 232 INTERFACE (voor communicatie)	F 222.50				

DE' TRONICS ***

TOETSENBORD VOOR INBOUW VAN DE ZX 81 F 228.00
 GRAPHICS IN ROM VOOR DE ZX 81 F 137.50
 TOOLKIT IN EPROM (te gebruiken met graph rom board) F 59.50
 FLEXIBELE BANDKABEL KONNEKTOR (voor storingsvrij gebruik) F 55.00

MICRO SOURCE OSSENMARKT 25 POSTBUS 1243 8001 BE ZWOLLE

TELEFOON 038 - 22 36 98

ABN 59 82 44 948 POSTGIRO 36 77 209

VERZENDEN *****
 PAKKET 25T (vrij vooruitbetaling) F 8.75
 REHBORS (betalen aan de postboor) F 12.75
 AL ONZL PRIJZEN ZIJN VRIJBLIJVEND, INCL BTW, EXCL VERZENDKOSTEN
 prijzen per 1.9.83 WIJ GEVEN PRIJSGARANTIE, VRAAG DE VOORWAARDEN
 De met ** gekleurde prijzen zijn niet gegarandeerd, bel bij twijfel

PRIJSGARANTIE.
 ALS U TETS BTJ ONS HEEFT GEKOCHT, EN U KUNT ONS BINNEN 7 DAGEN
 NA VERZENDING VAN HET PAKKET AANTOENEN DAT HET/DE/DE PRODUCT ERGENS
 ANDERS OP DE DAG VAN AFLEVERING GELIJKOEPER UIT VOORRAAD LEVERBAAR
 WAS, DAN KRIJGT U HET VERSCHIL TERUG. NIJ BEHOUDEN ONS HET RECHT
 VOOR OM HET ARTIKEL TEGEN DE DOOR U BETALDE PRIJS TERUG TE KOPEN.



1	035 ZX CHESS 2	ARTIC	ZX81	16K	39.50
2	062 3D DEFENDERS	NEW.GEN.	ZX81	16K	29.50
3	073 ZX ASSEMBLER	ARTIC	ZX81	16K	49.50
4	160 MINI INFOR	AACKOSFT	SPECTRUM	16K	25.00
5	061 3D MONSTERMAZE	NEW.GEN.	ZX81	16K	29.50
6	096 ZX COMPILER	AACKOSFT	ZX81	16K	29.50
7	099 4D TIMEGATE	OS	SPECTRUM	48K	39.50
8	033 CASSETTE 4	M.ORWIN	ZX81	16K	34.50
9	071 GULP 2	CAMPBELL	ZX81	16K	29.50
10	109 MASTERFILE	CAMPBELL	SPECTRUM	48K	69.50
11	049 ZX TOOLKIT	ARTIC	ZX81	16K	25.00
12	010 ZX COMPUTACALC	SIL.TRIDS	ZX81	16K	34.50
13	142 FROGS	AACKOSFT	ZX81	16K	25.00
14	111 GROUND ATTACK	SILVERSOFT	SPECTRUM	16K	29.50
15	108 GULPMAN	CAMPBELL	SPECTRUM	16K	25.00
16	078 FIGFORTH	ARTIC	ZX81	16K	87.50
17	112 DRBITER	SILVERSOFT	SPECTRUM	16K	29.50
18	021 DISASSEMBLER	CAMPBELL	ZX81	16K	25.00
19	020 DATABASE	CAMPBELL	ZX81	16K	34.50
20	127 SUPER C COMPILER	SOFTK	SPECTRUM	16K	49.50
21	114 MICROCHESS	ARTIC	SPECTRUM	16K	34.50
22	076 INVASION FORCE	ARTIC	ZX81	16K	25.00
23	072 CASSETTE 5	M.ORWIN	ZX81	16K	34.50
24	098 THE CHESSPLAYER	OS	SPECTRUM	48K	34.50
25	059 T F D	CAMPBELL	ZX81	16K	49.50
26	106 VOICE CHESS	ARTIC	SPECTRUM	48K	39.50
27	057 GOBBLEMAN	ARTIC	ZX81	16K	25.00
28	066 TRANSCODER	AACKOSFT	ZX81	16K	29.50
29	101 SPACE INTRUDERS	OS	SPECTRUM	16K	29.50
30	107 S P D E	CAMPBELL	SPECTRUM	48K	29.50
31	100 METEOR STORM	OS	SPECTRUM	16K	29.50
32	113 MINED OUT	OS	SPECTRUM	48K	29.50
33	118 SPECTRUM ASSEMBLER	ARTIC	SPECTRUM	48K	49.50
34	143 BEEP-CHESS	COMP.CONC.	BBC B 1.2	32K	45.00
35	087 SUPER WUMFUS	SILVERSOFT	ZX81	16K	29.50
36	052 SCRAMBLE	OS	ZX81	16K	25.00
37	046 GALAXIANS	ARTIC	ZX81	16K	25.00
38	070 JRS GRAPH TOOLKIT	JRS	ZX81	16K	29.50
39	124 COSMIC SWARM	SOFTK	SPECTRUM	16K	29.50
40	032 CASSETTE 3	M.ORWIN	ZX81	16K	34.50
41	117 INVASION FORCE	ARTIC	SPECTRUM	16K	29.50
42	119 ARCADIA	J.K.GREYE	SPECTRUM	16K	29.50
43	139 WORDWISE FIRMWARE	COMP.CONC.	BBC B 1.2	32K	199.00
44	105 FROG/SHOWDOWN	ARTIC	SPECTRUM	16K	29.50
45	038 ZX BUG	ARTIC	ZX81	16K	29.50
46	126 METEORIDS	SOFTK	SPECTRUM	16K	29.50
47	116 INVADERS	ARTIC	SPECTRUM	16K	29.50
48	154 DATABASE	SOFTW.F.ALL	BBC B 1.2	32K	39.50
49	069 JRS 16K TOOLKIT	JRS	ZX81	16K	29.50
50	140 ASTEROID BELT	COMP.CONC.	BBC B 1.2	32K	34.50

- LAGE PRIJZEN
 - TOPKUALITEIT SOFTWARE
 - 80% MET NEDERLANDSE
 HANDLEIDING
 - OVERAAL VERKRYGBAAR

**ACKOSFT
SOFTWARE**

INLICHTINGEN
 TEL: 071-213071

computercollectief

Amstel 312 (t.o. Carré) / 1017 AP Amsterdam / Giro 4 475 158 / Bank NMB 69.79.15.646

tijdschriften voor de ZX81 en ZX SPECTRUM:

YOUR COMPUTER f 7,- SYNC f 16,-- ZX COMPUTING (2-maandelijks) f 13,50
 nieuw: SPECTRUM COMPUTING (2-maandelijks magazine op cassette) f 20,--
 SYNCLAIR PROJECTS (hardware projekten) f 9,-- SINCLAIR USER f 7,--

nieuwe en actuele boeken:

ASSEMBLY LANGUAGE ASSEMBLED FOR THE ZX-81 (Woods/Macmillan) f 44,--
 leert programmeren in machinetaal mbv ZX-AS assembler van Bug-Byte.
 ZX-81 ELECTRONICAPROJECTEN (Kluwer) inclusief printplaatje f 27,50
 ADVANCED GRAPHICS FOR THE ZX SPECTRUM (Angell/Macmillan) f 49,--
 GAMES ZX COMPUTERS PLAY (Interface) f 16,--
 BEYOND SIMPLE BASIC - DELVING DEEPER INTO YOUR ZX SPECTRUM (Jones) f 39,--
 20 SIMPLE ELECTRONIC PROJECTS FOR THE ZX SPECTRUM (Adams) f 32,--
 CREATING ARCADE GAMES ON THE ZX SPECTRUM (Haywood) f 25,--
 SPECTRUM MICRO DRIVE BOOK (Logan) f 35,--
 SUPERCHARGE YOUR SPECTRUM (Webb) f 34,--
 SPECTRUM PROGRAMMES 1 (Hampshire) f 35,--
 60 GAMES AND APPLICATIONS FOR THE ZX SPECTRUM (Harwood) f 25,-
 UNDERSTANDING YOUR SPECTRUM (Logan) f 39,-
 SPECTRUM MACHINELANGUAGE FOR ABSOLUTE BEGINNERS (Tang) f 34,-
 COMPLETE SPECTRUM ROM DISASSEMBLY (Logan & o'hara) f 46,--

op 21 oktober, tijdens de HCC dagen, komen we uit met onze nieuwe
 WINTER 83/84 catalogus. deze bevat nu meer dan 100! boektitels voor Sinclair.

nieuwe en actuele software voor de ZX81:

Bug-Byte: ZX AS assembler f 36,-- ZX-DB debugger f 36,- ZX Toolkit f 36,-
 Quicksilva: SUBSPACE STRIKER f 20,- GALAXIANS f 20,- 3D BLACK STAR f 26,-
 Quicksilva: TRADER TRILOGY f 48,- MUNCHEES f 20,- COSMIC GUERRILLA f 20,-
 CP-software: SUPERCHESS 16K f 28,-
 New generation Software: 3D MONSTERMAZE f 28,- 3D DEFENDER f 28,-

nieuwe en actuele software voor de ZX Spectrum:

Melborne House: THE HOBBIT f 70,- PENETRATOR f 38,- TERROR DAKTIL 4D f 38,-
 Melborne House: MELBORNE DRAW f 48,- hiermee zijn de Hobbit-plaatjes gemaakt!
 Imagine: JUMPING JACK f 28,- ARCADIA f 28,- AH DIDDUM f 28,-
 Imagine: ZIP ZAP f 28,- MOLAR MAUL f 28,- ZZOOM 3D f 28,-
 Ultimate: JET PAC f 30,- PSSST f 30,- COOKIE f 30,- TRANSAM f 30,-
 Ocean: CRAZY KONG f 28,- ARMAGEDDON f 28,- ROBOTICS f 28,-
 Hi-soft: DEVPAC (assembler/monitor) f 75,- HISOFT PASCAL compiler f 125,-
 Quicksilva: AQUAPLANE f 36,- TRADER TRILOGY f 48,- FRENZY f 26,-
 Quicksilva: 4D TIMEGATE f 36,- THE CHESSPLAYER f 36,- XADOM f 36,-
 CP-software: SUPERCHESS II 48K f 44,- SUPERCHESS II 16K f 38,-
 CP-software: COLOSSAL CAVES - het originele adventure! f 40,-
 Hewson: 3D SPACE WAR f 30,- SPECTRAL PANIC f 30,- MAZE CHASE f 25,-
 Hewson: SPECVADERS f 25,- NIGHTFLITE f 30,- AIRTRAFFIC CONTROLLER f 40,-
 Microsphere: OMNICALC spreadsheet f 49,- TRAIN GAME f 28,-
 Sheperd: TRANSYLVANIAN TOWER f 30,- SHIP OF THE LINE f 30,- EVEREST f 30,-

SUZO SPECTRUM JOYSTICK INTERFACE voor alle Atari-type joysticks f 99,-
 SUZO COMPETITION-PRO JOYSTICK met micro-switches en metalen pen f 75,-

winkel open : wo t/m za van 11.00 t/m 17.00 - alle prijzen zijn incl. BTW
 verzendkosten f 6,- per bestelling - vraag onze nieuwe WINTER83/84 catalogus aan

microcomputer tijdschriften boeken en software

WAT IS DE "SINCLAIR GEBRUIKERSGROEP?

Dat is een onderdeel van de Hobby Computer Club (HCC) Nederland. Die HCC verenigt reeds meer dan 10.000 computergebruikers in alle delen van ons land, gebruikers van allerlei merken hobbycomputers. De HCC werkt ook nauw samen met haar Belgische zustervereniging. De HCC is onderverdeeld naar regio en naar gebruikers van bepaalde merken computers. In zo'n gebruikersgroep (gg) komen computerhobbyisten samen die allen geïnteresseerd zijn in hetzelfde merk computer, of zij zelf zo'n computer hebben of niet, dat maakt niet uit.

De HCC wil het gebruik van hobbycomputers bevorderen door een aantal activiteiten te organiseren. Er worden HCC-dagen gehouden in het najaar (de Belgische HCC organiseerde zo'n manifestatie in het najaar 1983). De HCC-nieuwsbrief is het verenigingsorgaan, dat voor 'elck wat wils' bevat. De HCC Hardwareservice levert vele soorten chips e.d.. Gedurende de afdelingsbijeenkomsten (regionaal) legt men persoonlijke contacten met gelijkgezinden.

Eén van de HCC Gebruikersgroepen is de SINCLAIR GEBRUIKERS GROEP. Deze organiseert regelmatig bijeenkomsten met lezingen en demonstraties van hardware en software. Maar vooral is er dan ook gelegenheid om ervaringen uit te wisselen met andere gebruikers van een Sinclair ZX-80, ZX-81 of Spectrum.

Deze gebruikersgroep beheert ook een Software-Bibliotheek. Die wordt regelmatig uitgebreid met programma's en routines voor Sinclair's die door leden van de gg zijn gemaakt. Enkele daarvan worden op IMPULSOFT-cassettes verkocht. En 'last but not least' geeft deze gg op onregelmatige tijden een boekje als dit uit. De omvang van 50 pagina's wordt steeds nagestreefd. Het bevat artikelen, programma's en advertenties over Uw Sinclair computer, op allerlei terreinen, zoals hoe Uw Sinclair werkt, hoe het anders kan, wat U er nog meer mee kan, wat machinetaal is en het geeft spelprogramma's, zakelijke, leerzame en andere toepassingen in programma-vorm. Ook vindt U er schema's in voor uitbreidingen. Ja, eigenlijk alles wat de gg-leden er in willen stoppen kan via SINCLAIR IMPULS grotere bekendheid krijgen.

vandaar:	Wordt lid van de HCC	Kosten:
	Wordt lid van de SINCLAIR GEBRUIKERS GROEP	voor
	Maak méér van Uw hobby!	héél
NU DOEN!	Meld U aan als lid.	1983
		f 40.-

CORRECTIES OP 2400 BAUD FAST-SAVE-LOAD-ROUTINE

Helaas zijn enkele zeer storende foutjes door de correctieronde van dit artikel geslopen. Aan het eind van het artikel, op blz. 23, drie regels na het kopje VERIFY moet U lezen "Na PRINT USR 30465 ..."

Op de volgende bladzijde, in de listing van de LOAD deelroutine staat een regel die moet luiden 76E0 72 LD (HL),D (haakjes om HL i.p.v. om de D). De op twee na laatste regel van diezelfde deelroutine is de instructie moeilijk leesbaar. Daar moet staan 76FA 38EA JR C, SIGNAL.

In de VERIFY deelroutine tenslotte drie belangrijke kleine correcties: Vierde regel van de listing moet luiden: 76F7 38F8 JRC, TAPE. Een soortgelijke fout 16 regels lager: lees 7717 38C8 JRC, L/S UPDATE. Minder essentieel maar toch fout is het een regel lager: 7719 BE CP (HL).

Met de excuses van de redactie, maar zó moet het wél werken.