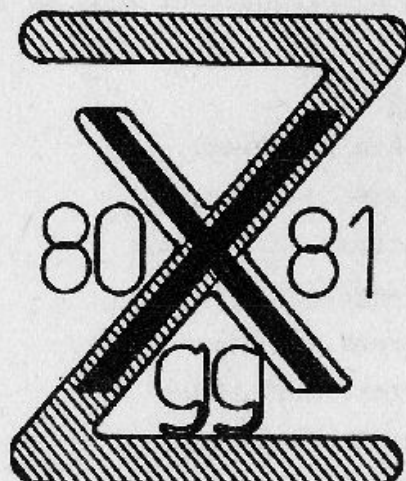


SINCLAIR

IMPULS



Nummer 1

HCC Sinclair Gebruikersgroep
Lijsterbesstraat 28
1741 SN Schagen.

SINCLAIR IMPULS

INHOUDS OPGAVE NUMMER 1

1. Algemeen

- 1.1 Introductie 01-05
- 1.3 Testen van cassette signalen van de ZX-80 01-07

2. Hardware

- 2.1 Universele experimenteer-print 01-15
 - Schema ZX 80 01-16
 - Schema ZX 81 01-18
- 2.3 Repeat key 01-20

3. Interfacing

- 3.2 8 Bits I/O poort 01-23
- 3.3 Data ingangspoort voor de ZX-80 01-24

4. Software

- 5.2 Teken programma 01-27
 - Cassette index 01-27
- 5.4 Nuclear Power Plant 01-28
 - Kiengetal trekken 01-33
 - Torens van Hanoi 01-36
 - Boter, Melk, Kaas 01-37
 - Getal omdraaien 01-38
 - Dobbelsteen gooien 01-38
- 5.5 Kasboek 01-39
 - N-faculteit berekenen 01-43

6. Talen

- 6.1 Hoe laden we machinecode in de ZX 80/81 01-47

ALGEMEEN

INHOUD

Introductie	01.1.1.001	01-05
Testen van cassettesignalen van de ZX-80	01.1.3.001	01-07

Z X - S O F T W A R E

Serieuze applicatieprogramma's van niveau en kwaliteit

Bovendien scherp geprijsd en gebruiksvriendelijk:

- Inkomsten en Uitgaven

45 posten, 70 financiële overzichten,
staafdiagrammen en diverse printroutines f 39.--

- Boodschappenlijst

Snal en handzaam in 't gebruik, en door een
unieke invoer/sorteer-procedure nu supersnel
winkelen aan de hand van een overzichtelijke,
gerangschikte BOODSCHAPPENLIJST f 34.--

- Kredieten

Berekening van alle variabelen.
Zeer flexibel en bijzonder compleet f 29.--

- Grafische Functieanalyse

Bepaling nulpunten en extremen, met behulp
van uitermate geavanceerde plotroutines f 34.--

In voorbereiding: Telefoon- en Adressenbestand, Inboedel,
Sparen, Wortelbepalen ($Re + Im$), Numerieke Functieanalyse.

Alle programma's zijn uitgesproken compleet, veelzijdig en
flexibel, waarmee een optimale prijs-/prestatieverhouding
is gerealiseerd.

Inlichtingen en/of Bestellingen:

ZX-SOFTWARE CENTRE, Weverij 32, 8401 PH GORREDIJK
(05133-1929 na 20.00 uur)

Levering na overmaking kostprijs op bank rek: 32.29.52859

Prijzen 'all-in'.

INTRODUCTIE

Het eerste "50-pagina"-boekje van de Sinclair gebruikersgroep is van de pers gerold.

Als naam voor deze uitgave is gekozen voor SINCLAIR IMPULS. De samenstellers hopen dat de inhoud voor u, Sinclairgebruiker, een impuls zal zijn om nieuwe wegen bij het gebruik van uw computer in te slaan.

De opbouw van het boekje is zodanig dat u het op twee manieren kunt bewaren.

- Als complete boekjes. De inhoudsopgave per boekje staat aan de binnenzijde van de omslag.
- Losbladig. Aan de linkermarge is ruimte gereserveerd voor perforatie.

Bij de indeling wordt rekening gehouden met de volgende hoofdstukken en paragrafen.

1. Algemeen

- 1.1 Redactioneel
- 1.2 "Op tilt"-verbeteringen
- 1.3 Systeembeschrijvingen
- 1.4 Modificaties
- 1.5 Memory map
- 1.9 Diversen

2. Hardware

- 2.1 Schema's
- 2.2 Modificaties
- 2.3 Uitbreidingen
- 2.4 RS 232-20 mA loop- IEEE 488
- 2.5 Buffering
- 2.6 Graphics
- 2.9 Diversen

3. Interfacing

- 3.1 Joystics
- 3.2 Parallel I/O
- 3.3 Serial I/O
- 3.4 Modems
- 3.5 Printers
- 3.6 Plotters
- 3.7 Microdrives, Floppy disks
- 3.9 Diversen

4. Firmware

- 4.1 4K Basic ROM ZX-80
- 4.2 8K Basic ROM ZX-81
- 4.3 16K Basic ROM Spectrum
- 4.4 Assembler, Disassembler
- 4.5 Teksteditor
- 4.9 Diversen

5. Software

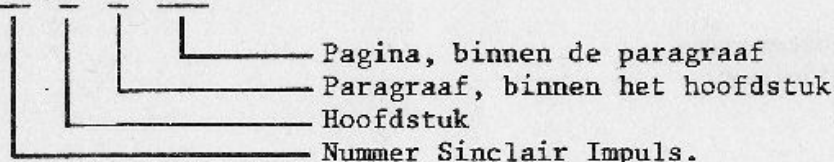
- 5.1 Subroutines
- 5.2 Utilities
- 5.3 Basicode
- 5.4 Spelen
- 5.5 Zakelijk gebruik
- 5.9 Diversen

6. Talen

- 6.1 Machine code
- 6.2 Basic
- 6.3 Forth
- 6.4 Pascal
- 6.5 Lisp
- 6.9 Diversen

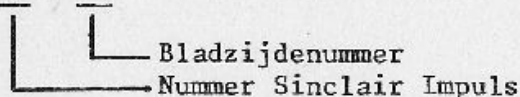
De paginanummering rechtsboven is als volgt opgebouwd:

01. 2. 3. 456



Rechtsonder vindt u een nummering per editie, deze is o.a. ten behoeve van de drukker.

01 - 23



De hoofdstukken worden steeds voorzien van een titelpagina met de totale inhoudsopgave van dat hoofdstuk.

E. R. Visser

```
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
& Wie helpt ons &
& GEZOCHT: 4K Basic ROM (uit de ZX80) &
& t.b.v. Software manager en &
& SINCLAIR IMPULS redactie &
& reacties aan: Lijsterbesstraat 28 &
& 1741 SN Schagen. &
```

HET TESTEN VAN CASSETTESIGNALLEN (OP DE EARBUS) VAN DE ZX-80

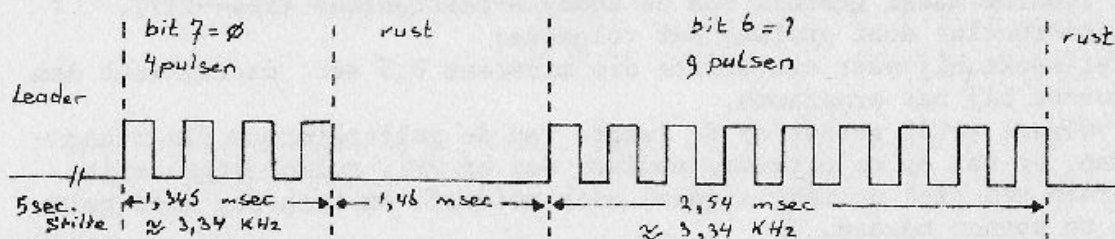
Er zijn al de nodige aanwijzingen gepubliceerd voor het laden van programma's met cassette-recorders en -radio's.

Hier is een methode die misschien ook zijn steentje kan bijdragen.

Het volgende slaat op de oude ZX-80 met 4K-ROM.

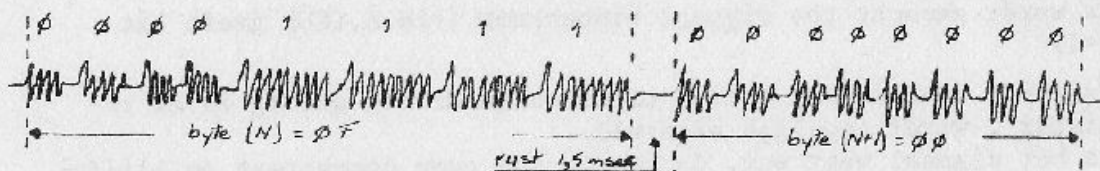
De SAVE-routine staat in de adressen 438 t/m 503 met daarbij horend de Endbyte-testroutine in adressen 504 t/m 517.

Het bleek mij dat de ZX-80 signalen uitzendt die er in een diagram als volgt uit moeten zien....



Is één byte uitgezonden dan volgt een rustpauze van 1,5 msec., dus praktisch geen verschil met die tussen de bits van een byte. Direct daarna wordt de volgende byte verzonden, dus zonder voorafgaande codesignalen, dat er een nieuwe byte zal komen.

In grafische vorm: b.v. byte(N)=0F en byte(N + 1)=00.



Deze patronen zijn ook op het TV-scherm te herkennen, neem b.v. het volgende programma:

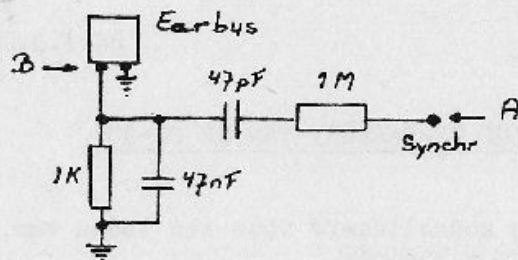
```
10 DIM A (100)
20 FOR X=0 TO 50
30 LET A (X)=-1
40 NEXT X
50 SAVE
```

-- (dus A(51) t/m A(100) blijven nul)

en RUN het.

Na enig onregelmatig gedoe op het scherm, dat zijn dan de systeemvariabelen en het programma, wordt het TV-beeld regelmatig van vorm, eerst ongeveer 4 zwarte balken (de rustpauzes) en iets bredere wit-zwart gestreepte banden ertussen. Dat zijn de 9 pulsen voor de 1-bits uit A(0) t/m A(50)=-1 (=FFFF_h = 1111 1111 1111 1111_h), daarna komen er wat dunnere witgestreepte banden tussen ongeveer 6 zwarte banden. Dat moeten de array-elementen A(51)---A(100)=0 zijn, dus 4 pulsen telkens voor de 0-bytes.

Uit deze proef blijkt nog, dat de rustpauze signaal-aan betekent, dus eigenlijk verwacht je spanning op de MIC-uitgang, maar de schakeling ziet er als volgt uit:



$R_{\text{rust}} = 5 \text{ Volt (IN A, (FE))}$

$1 \text{ puls} = 0 \text{ Volt (OUT (FF), A)}$

Deze Z-80 commando's
bedienen de I/O v.d. ZX-80

Blijft A 5 Volt gedurende enige tijd dan zal de spanning bij B tot 0 Volt weglekken door de 1K weerstand. Alleen snelle wisselingen van 3kHz en nog wat lager, b.v. 2kHz worden goed doorgegeven.

Nu het laden van programma's.

De LOAD-routine van de ZX-80 4K-ROM staat in adressen 518 t/m 597, ook deze routine maakt gebruik van de Endbyte-testroutine (504—517).

De Load-routine doet grofweg het volgende:

Eerst zoekt hij naar een stilte die minstens 0,5 sec. duurt, want dan verwacht hij het programma.

Vervolgens wordt getest op de lengte van de golftreintjes die binnenkomen, en wel op zo'n slimme manier, dat er vrij ruime marges zijn, waaraan een niet perfect signaal moet voldoen, om toch nog goed gelezen te kunnen worden.

De methode werkt als volgt: De Z-80 processor leest voortdurend de EAR-bus (met IN A, (FE)-instructie) en houdt een teller bij.

Hiervoor wordt het C-register gebruikt van de Z-80.

1. Het C-register wordt geïnitieerd op 148.
2. Er wordt gezocht tot signaal binnenkomt (=IN A, (FE) geeft bit 7=1)
3. Nu wordt voortdurend getest of het signaal nog geldig is en register C wordt tegelijk afgeteld.
Is het signaal weer weg, dan wordt nog even doorgetest en blijft het signaal weg, dan stopt de teller.

Er zijn nu 3 mogelijkheden:

- I C = 147 tot 86 : = geen geldig signaal
- II C = 85 tot 0 : = nul-bit gelezen (ideaal: C = 55)
- III C = -1 tot -128 : = één-bit gelezen (ideaal: C = -22).

Er is dus een ruime marge mogelijk.

Tot zover de ZX-80 routines.

Nu twee inleesroutines, die ook informatie kunnen geven over de kwaliteit van de cassette-signalen.

Het eerste programma leest achter elkaar 65536 maal de EAR-bus en telt daarvan het aantal binnengehaalde hoge signalen (dus énen). De routine kan geladen in een REM-statement, voor de ZX-80 heb ik beginadres 16430 genomen.

Adres	Mnemonics	Opcode hex.	States	Commentaar
16430	LD DE,0 LD HL,0	11 00 00 21 00 00		Init teller 65536 x Init teller hoog signaal
16436	IN A,(FE) RLA JR NC,+1 INC HL DEC DE LD A,D OR E JR NZ,-11	DB FE 17 30 01 23 1B 7A B3 20 F5	test om de 53 of 54 states 11 4 12/7 6 6 4 4 12/7	bit 7 in A-register bevat de cassette-input. Alleen verhogen als signaal is 1. Aftellen Al 65536 maal geweest?
16447	RET	C9		Terug naar Basic.

Het HL-register bevat nu het nodige gegeven.

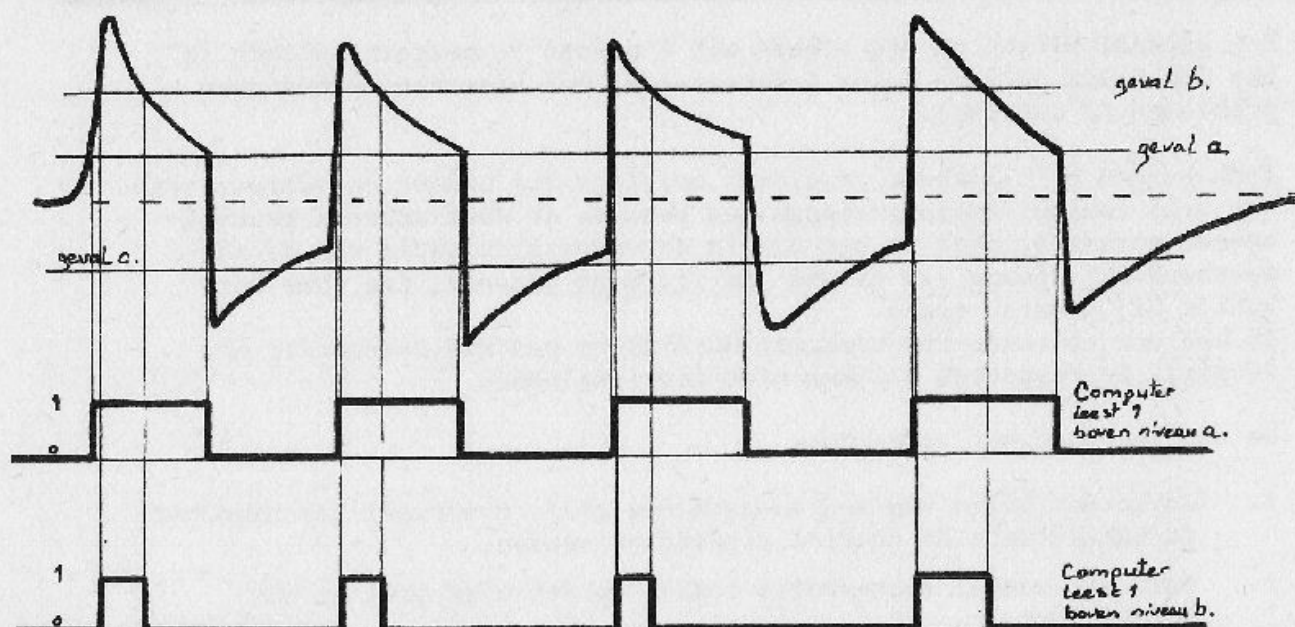
Met LET HOOG =USR (16430) kan dit gegeven in de Basic verder verwerkt worden.

Testresultaten op eigen opnames (goede) leverden waarden als 21098, 21739, 21442, 22032, 19556, 19978,.....

Conclusie: Sinclair code geeft signaal gedurende 25 à 30% van de tijd.

De tweede routine werkt wat verfijnder.

Stel, een binnenkomend signaal ziet er als volgt uit:

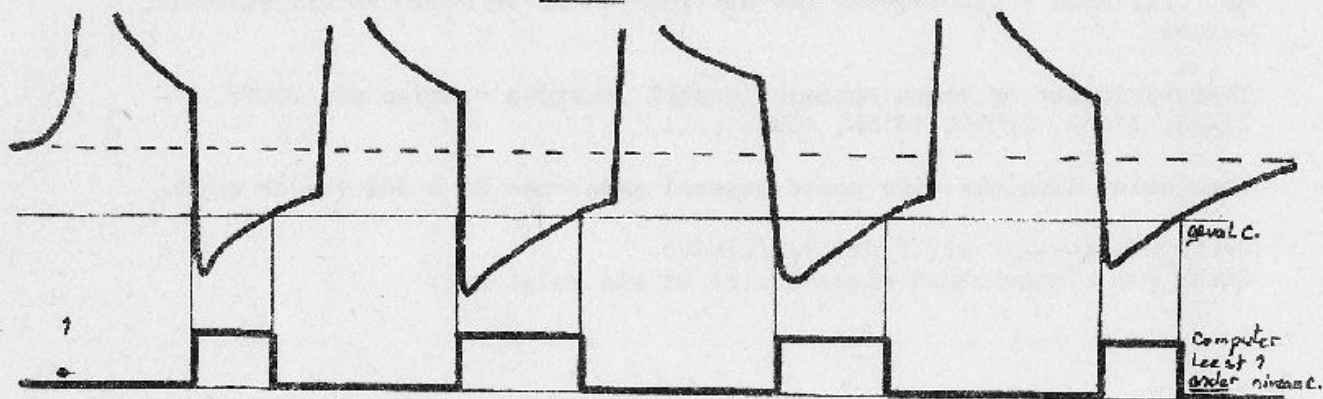


In geval a) leest de computer signaal boven niveau a) als énen. Dat levert een mooi resultaat, hoog en laag ongeveer even lang.

In geval b) leest de computer pas signaal boven niveau b). Minder mooi, maar de signalen kunnen nog best goed gelezen worden, als er maar geen stoorsignalen doorheen komen.

Pas op: Niveau a) is even hoog als niveau b,
alleen in geval b) zijn de cassette-signalen zwakker.

Ook kan het voorkomen dat de cassette-recorder de signalen omkeert, wat als positieve impuls bedoeld is, komt als negatief resultaat terug, we meten dan b.v. alles onder niveau c) als éenen.



Het signaal blijkt nu nog steeds uit 4 pulsen te bestaan, alleen is het een halve periode later gedetecteerd, ook hier hoeven dus geen problemen te ontstaan.

Toch kunnen er blijkbaar problemen ontstaan bij opname en weergave van het vrij zwakke Sinclair-signaal en behalve de door anderen gepubliceerde oorzaken, zoek ik het ook in de vóórmagnetisatie van de magneetband bij opname van geluid, de zg. bias genoemd, die soms verschilt bij diverse typen.

Ik heb een cassette-recorder met DC-bias en een radio-recorder met AC-bias. De cassettes blijken niet uitwisselbaar.

De routine doet het volgende:

1. Zoekt het begin van een hoogstaande golf, eventueel kan hier met de BREAK-toets de routine afgebroken worden,
2. telt het aantal succesvolle testen op het hoge deel in het H-register,
3. telt aansluitend het aantal geldige proeven in het L-register,
4. keert terug naar Basic, waar de gegevens verwerkt worden.

Deze golftest-routine heb ik laten beginnen op adres 16450.

Adres	Mnemonics	Opcode hex.	States	Commentaar
16450	LD HL,0	21 00 00	16	Init tellers H en L
16453	IN A,(FE)	DB FE	11	
	RLA	17	4	bit 7 = cassette input
	JR C,-5	38 FB	12/7	test tot signaal is laag
16458	LD A,7F	3E 7F	7	a ₁₅ = 0 voor Breakdetectie
	IN A,(FE)	DB FE	11	
	RRA	1F	4	bit ₀ in carry
	JR NC,+16	30 10	12/7	Carry=0 als BREAK ingedrukt.
16465	RLA	17	4	
	RLA	17	4	bit ₇ = cassette- input
	JR NC,-11	30 F5	12/7	test tot signaal weer hoog
16469	IN A,(FE)	DB FE	11	
	RLA	17	4	test en tel
	INC H	24	4	aantal hoge signalen
	JR C,-6	38 FA	12/7	in H-register
16475	IN A,(FE)	DB FE	11	
	RLA	17	4	test en tel
	INC L	2C	4	aantal lage signalen
	JR NC,-6	30 FA	12/7	in L-register
16481	RET	C9	10	Terug naar Basic

In het basic-programma kunnen we nu b.v. zeggen:

```
LET SIGNAAL = USR (16450)
PRINT "HOOG"; SIGNAAL/256
PRINT "LAAG"; SIGNAAL AND 255
```

Tests op goedwerkende Sinclairopnamen leverden als waarden op HOOG 14 tot 16 en LAAG 15 tot 17, dus praktisch even lang en zo hoort het ook. Soms werd voor LAAG 125 of 151 of iets dergelijks gemeten, dan werd kennelijk een rustpauze gemeten.

Ook BASICODE is te onderzoeken en dit levert meestal uitstekende waarden op, 1200 Hz levert op 40 t/m 46
2400 Hz levert op 20 tot 22.

Sommige basicodes geven echter ook storingen te zien.

H. Radius.

COMPUTER - BOEKEN

Uitgaven van Uitgeverij WOLFKAMP op computergebied :

2X 81 HORIZON - Watney , 70 blz. + programmacassette ,

splitaalband ; Fl. 59,50

- zet elk 'board-game' om in een computerspel
 - animated - ball - display
 - overschrijven van BASIC in machinetaal
 - animatie - graphics
 - tekstverwerker
 - overzetten van complete programma-blokken
 - kopiëren van beveiligde programma's
- Nodig : enige kennis van BASIC programmeren, 16 K geheugen.

ENGELSTALIG

FORTH OK - Meljer/Lemaire , 133 blz. softcov. , Fl 29,50

- Nederlandse inleiding op de taal van de toekomst

FORTH

APPLE HANDBOEK - Van Kampen/De Kuiper , 390 blz. ringband

inmen omslag , Fl 89,50

- Volledig Nederlands HANDBOEK voor de APPLE-gebruiker
- inclusief een volledige beschrijving van de D.O.S.
- losbladig t.v.m. update's en aanvullingen

VERWACHT : MACHINETAAL PROGRAMMEREN , APPLE - Inman/Inman

300 blz, softcov. Fl. 49,50

- Volledig Nederlands leerboek voor machinetaal programmeren van Uw APPLE.

Al deze boeken zijn verkrijgbaar bij de goede (technische) boekhandel , bij veel computerwinkels , en bij de uitgever :



WOLFKAMP

WETERINGSCHANS 221
POSTBUS 70254

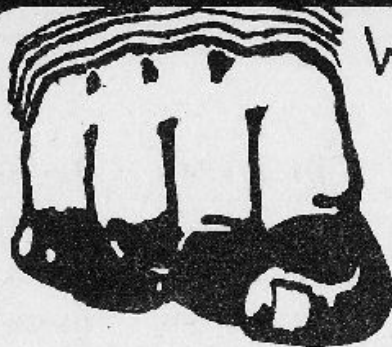
1007 KG AMSTERDAM
TELEFOON 020 - 278931

Haal meer uit Uw computer

HARDWARE

INHOUD

Universele experimenteer-print	01.2.1.001	01-15
Schema ZX-80	01.2.1.002	01-16
Schema ZX-81	01.2.1.004	01-18
Repeat Key	01.2.3.001	01-20



WIJ VECHTEN DOOR TEGEN DE **LOAD-** PROBLEMEN !

NIEUW !!! LOAD/SAVE/VERIFY-ROUTINE f 20,00

- 1200 baud (dus 4x zo snel als normaal !)
- Is gebruikersvriendelijk !
- Stelt zich in op het volume !
- Kan loaden en save onder naam !

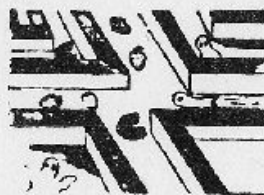
BETER 1200 BAUD IN DE HAND

DAN 4000 SLECHT OP DE BAND !...

GAME AND UTILITY

f 23,00

- Laten de monsters je niet te pakken krijgen !

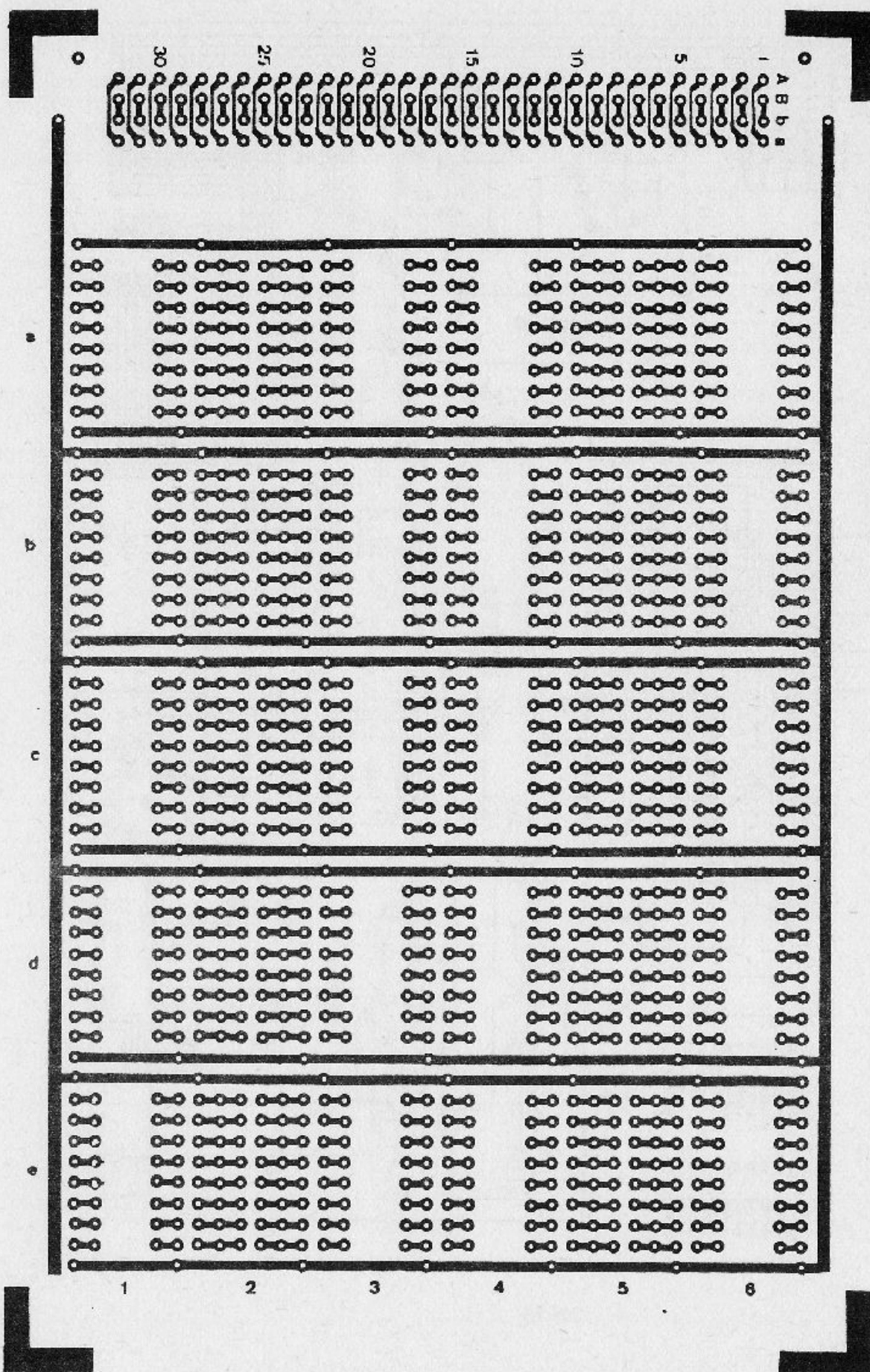


U kunt deze programma's bestellen bij
SOLIDWARE - postbus 52 - 3840 AB Harderwijk
tel 03410-20653 - postrek. 5281547 tnv R.Klopman

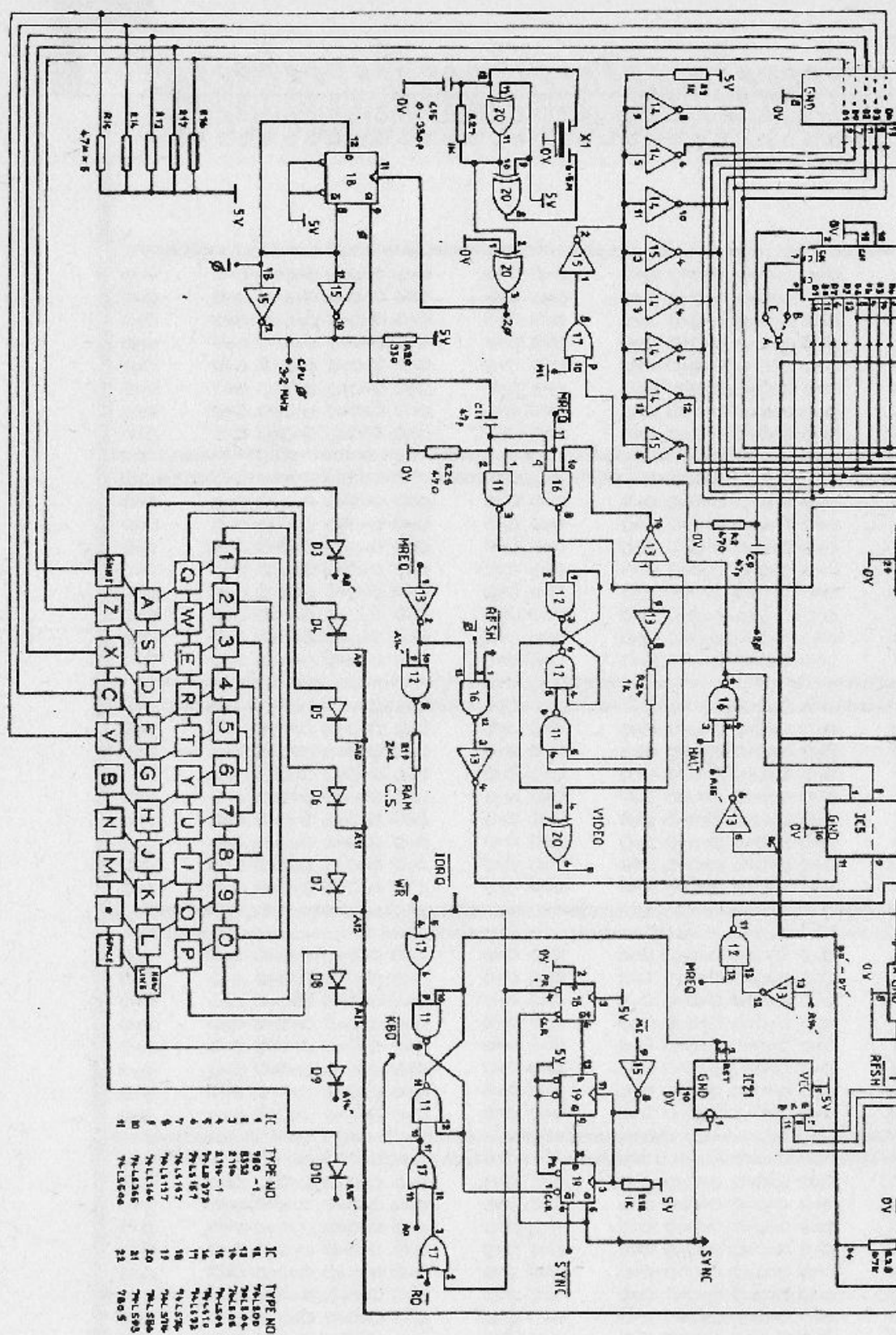
SINCLAIR IMPULS

UNIVERSELE EXPERIMENTEER-PRINT.

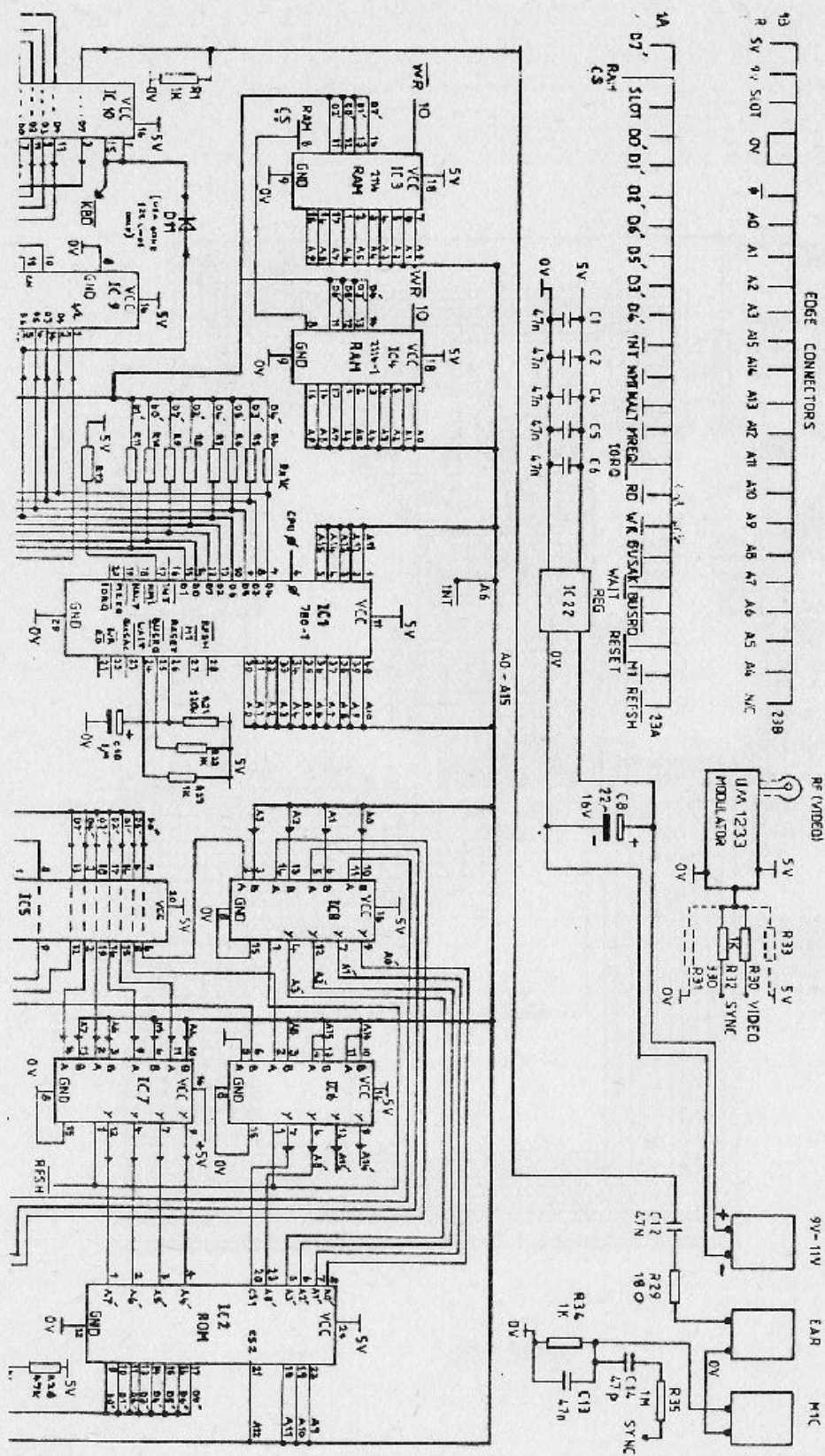
(maximaal 30 16pins-ic's)



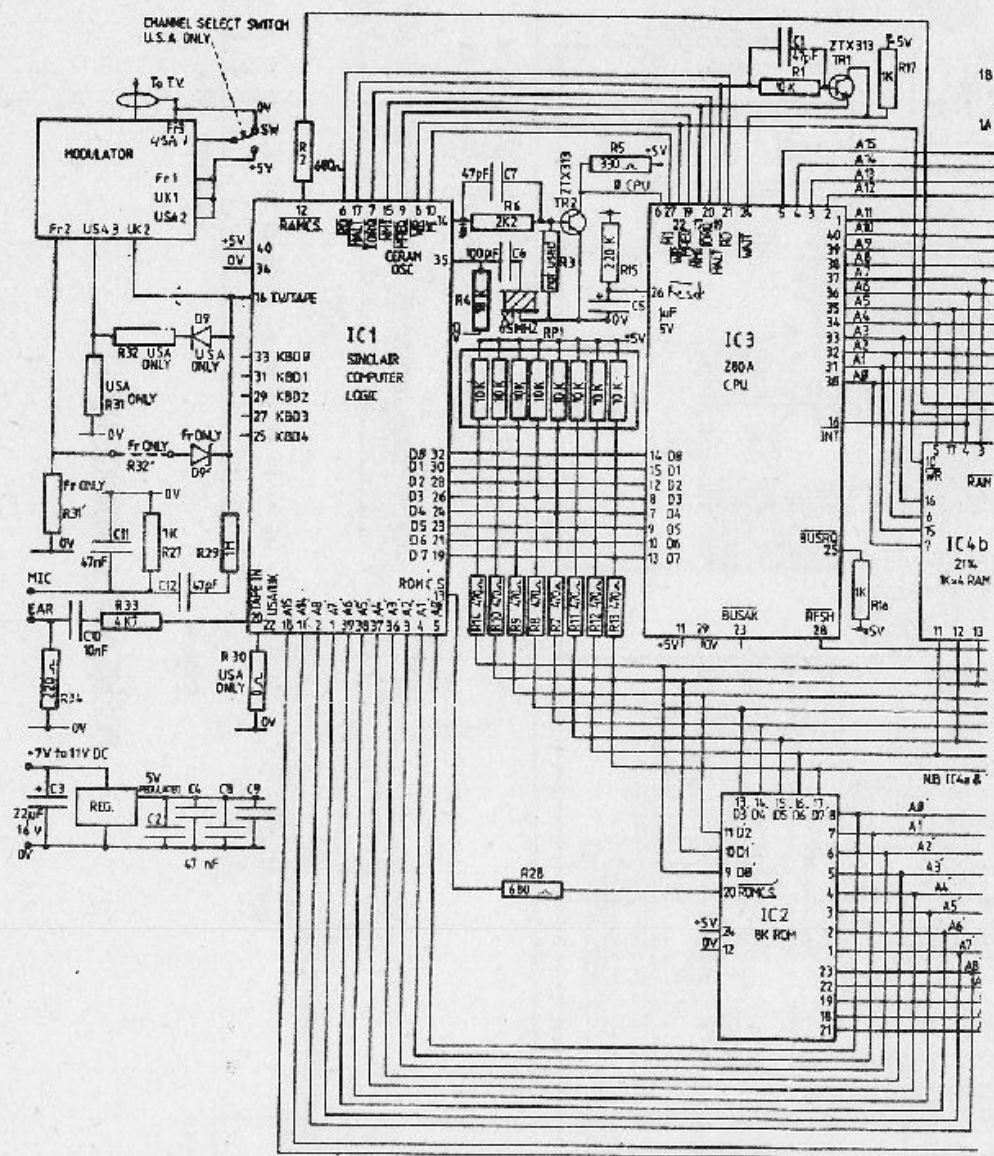
SCHEMA ZX-80



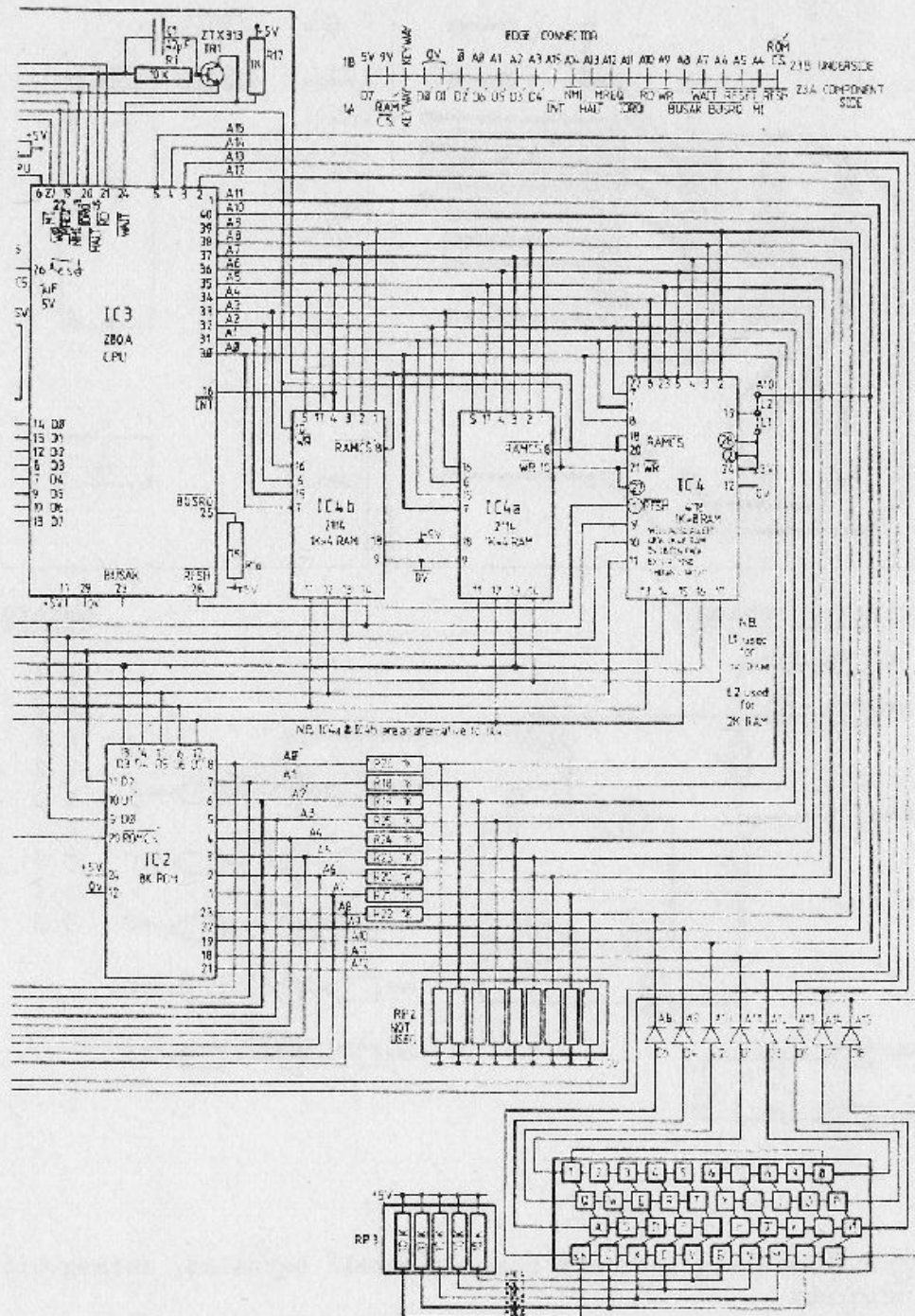
ZX80 CIRCUIT DIAGRAM



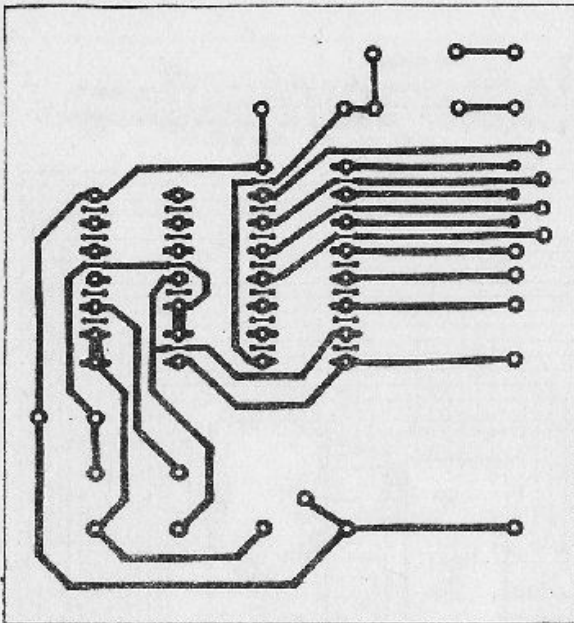
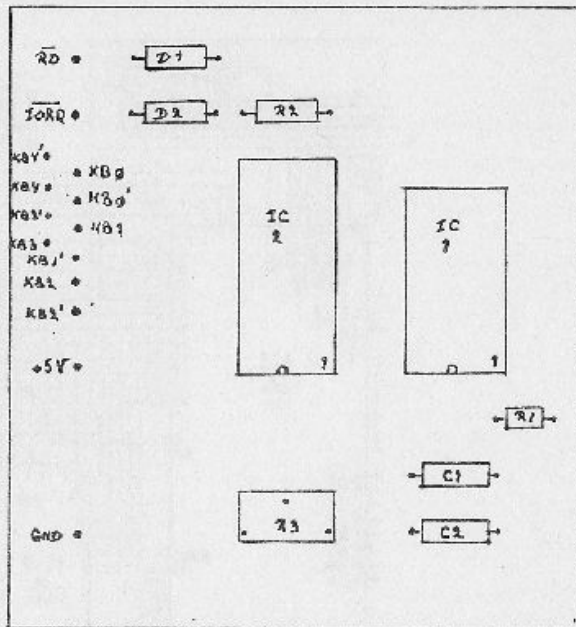
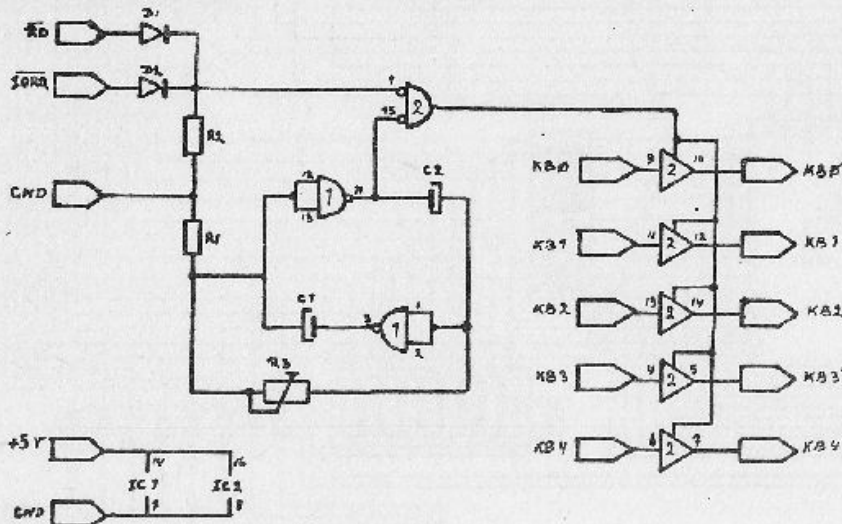
SCHEMA ZX-81



SINCLAIR ZX 81



REPEAT KEY

Print Lay-outComponenten opstellingPrincipe schemaOnderdelen

IC 1	74LS00
IC 2	74LS365
R 1	1K
R 2	1K
R 3	2K5 instel
C 1	220µF 165
C 2	220µF 165
D 1	Si-diode
D 2	Si-diode

Deze schakeling laat een toets zichzelf herhalen, zolang hij wordt ingedrukt.

Dit is handig bij cursor-besturingen, het intypen van vele spaties en dergelijke.

De Repeat-key schakeling moet worden opgenomen in de 5-strobe lijnen van het toetsenbord.

Met behulp van R 3 kan de herhalings-frequentie van een toets worden afgeregeld.

INTERFACING

INHOUD

8 Bit I/O-poort	01.3.2.001	01-23
Data ingangspoort voor de ZX80	01.3.3.001	01-24

***** COMPUTERCOLLECTIEF

Amstel 312a (t/o Carre) 1017 AP Amsterdam giro 4.475.158 NMB 69.79.15.646
winkel open: wo t/m za van 11.00 - 17.00 tijdschriften, boeken en software

* - T I J D S C H R I F T E N - *

* INTERFACE maandblad van de Engelse ZX Users' Club met programmaas f 8,-- *

* voor de Spectrum, ZX81 en ZX80. *

* SYNC tweemaandelijks Amerikaans tijdschrift voor Sinclair f 12,50 *

* gebruikers. *

* ZX COMPUTING tweemaandelijks Engels Sinclair tijdschrift f 15,-- *

* *

* - B O E K E N (dit is slechts een selectie) - *

* TOEPASSINGEN EN SPELLEN VOOR DE ZX81 (Voorburg) ca. f 22,50 *

* ZX HORIZON (Watney) inclusief cassette met de programmaas f 59,50 *

* BYTEING DEEPER INTO YOUR ZX81 (Harrison) f 36,-- *

* ZX81 PROGRAMS FOR REAL APPLICATIONS (Hurley) f 35,-- *

* MASTERING MACHINE CODE ON YOUR ZX81 (Tony Baker) f 39,-- *

* 49 EXPLOSIVE GAMES FOR THE ZX81 (Hartnell) f 32,-- *

* 20 SIMPLE ELECTRONIC PROJECTS FOR THE ZX81 (Adams) f 35,-- *

* STRETCHING YOUR ZX81 OR ZX80 TO ITS LIMITS (Hartnell) f 39,-- *

* THE EXPLORERS GUIDE TO THE ZX81 (Lord) f 29,-- *

* MACHINE LANGUAGE PROGRAMMING MADE SIMPLE FOR YOUR SINCLAIR f 49,-- *

* UNDERSTANDING YOUR ZX81 ROM (Logan) f 49,-- *

* HET PROGRAMMEREN VAN DE Z80 (Rodney Zaks) in het nederlands! f 58,-- *

* THE ZX SPECTRUM EXPLORED (Hartnell) f 39,-- *

* 20 PROGRAMS FOR THE ZX SPECTRUM (Altwasser) f 39,-- *

* OVER THE SPECTRUM f 40,-- *

* SPECTRUM MACHINE LANGUAGE FOR THE ABSOLUTE BEGINNER f 40,-- *

* 60 GAMES AND APPLICATIONS FOR THE ZX SPECTRUM f 32,-- *

* PROGRAMMING YOUR SPECTRUM (Logan) f 46,-- *

* *

* - S O F T W A R E (dit is slechts een selectie) - *

* MULTIFILE een database van Bug-Byte voor een 16 K ZX81 f 95,-- *

* COMPUTACALC van de Software Masters een Visicalc voor de Sinclair f 55,-- *

* MAZOGS mooi grafisch monstermaze van Bug-Byte. vind de schat f 60,-- *

* QS-INVADERS Quicksilva's versie van het bekende invaders f 27,50 *

* QS-ASTEROIDS van Quicksilva, de arcade-game specialist f 27,50 *

* QS-DEFENDER alle Quicksilva games zijn in machinetaal f 27,50 *

* QS-SCRAMBLE snel, actie, bewegende graphics, Quicksilva f 27,50 *

* 3D MONSTERMAZE van Greye f 27,50 *

* PUCKMAN van de Software Masters, een Packman voor de ZX81 f 39,-- *

* ZX CHESS van de Software Masters, met 6 niveaus f 49,-- *

* NIGHTMARE PARK een adventure van Software Masters f 49,-- *

* ADVENTURE Damsel & the Beast van Bug-Byte f 39,-- *

* DICTATOR van Bug-Byte een fantastisch nieuw Adventure f 55,-- *

* PILOT een ZX81 flight simulator f 39,-- *

* ZX FORTH een FORTH voor de ZX81 van Artic f159,-- *

* ZX AS een assembler van Bug-Byte f 30,-- *

* ZX DB een machinetaal debugger van Bug-Byte f 39,-- *

* ZX TK toolkit van Bug-Byte f 36,-- *

* PROGMERGE renumber en merge van de Software Masters f 39,-- *

* *

* - S P E C T R U M S O F T W A R E - *

* SPECTRAL INVADERS van Bug-Byte voor de SPECTRUM. high res, color f 30,-- *

* verwacht: Spectrum Assembler, Spectrum Chess, Meteor Storm, Intruders *

alles incl. BTW / verzendkosten f 6,- / vraag onze nieuwe CATALOGUS aan

8 BIT I/O-POORT

Een I/O port is één van de nuttigste uitbreidingen voor de ZX81. Wegens de hoge prijs van de in Engeland verkrijgbare ports, heb ik er zelf een ontworpen.

De hardware

De I/O port is vrij simpel van opzet en daardoor gemakkelijk en goedkoop te bouwen. Het schema spreekt voor zich. Ik moet echter wel enkele kanttekeningen maken :

1. Het is raadzaam om tussen de ZX81 en de schakeling een spanningsregulator (5 volt, 1 amp.) op te nemen en deze op een koelblok te monteren. Dit om oververhitting van de computer te voorkomen. Hierop kunnen dan ook andere schakelingen zoals A/D converters e.d. aangesloten worden.
2. Gebruik ALLEEN 74LS ic's; met SN74.. ic's werkt de schakeling niet !!
3. Niet gebruikte ingangen worden als logische "1" gezien en zijn nogal storingsgevoelig. Het is dus beter om niet gebruikte ingangen aan "0" te hangen (aan-5V).

De software

De poort is zeer eenvoudig te gebruiken :

In basic: op adres 19500 met Peek & Poke

b.v. LET A=PEEK 19500

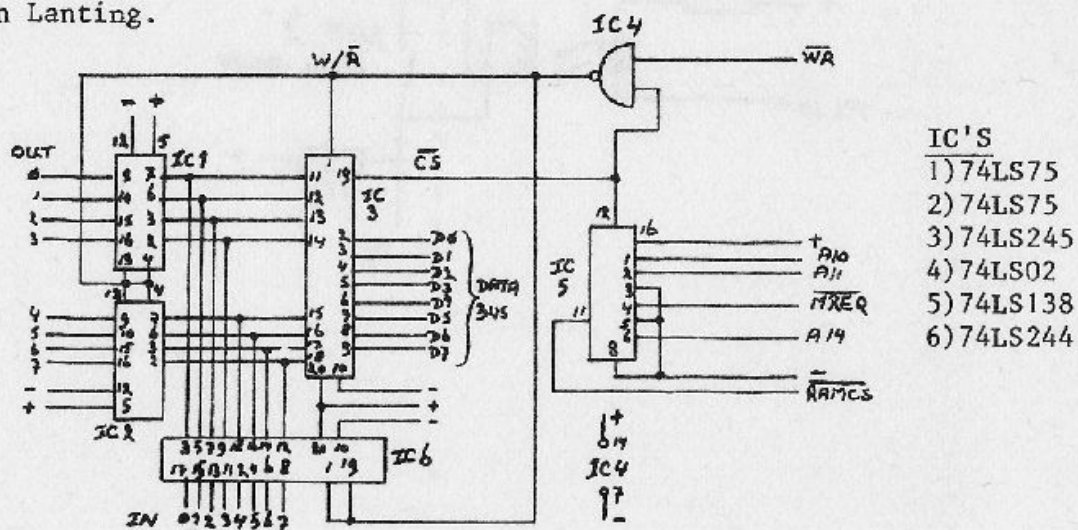
POKE 19500;A als A de waarde is die op de uitgang moet komen

In machinetaal: op adres 4C2C(hex.)

b.v. LdA, (4C2C)

Ld(4C2C),n ,waar n de waarde is die op de uitgang moet komen.

Erwin Lanting.



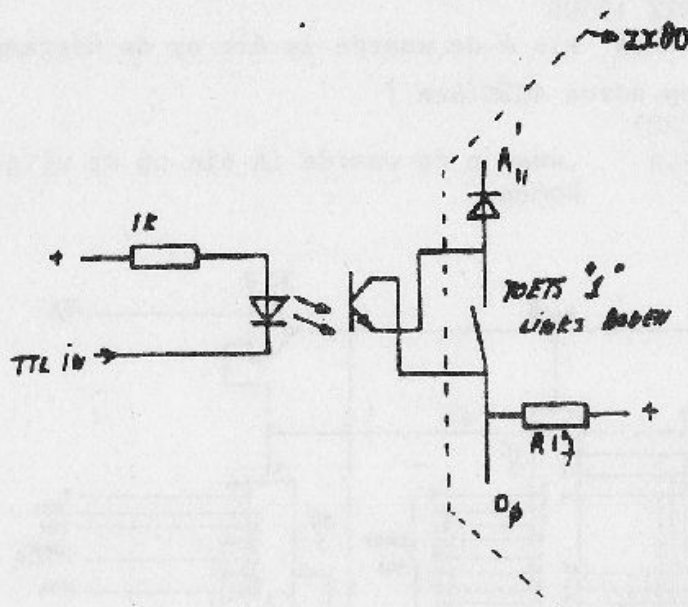
DATA INGANSPOORT VOOR DE ZX80

Om met de ZX80/81 bijzondere dingen te doen zoals marsedecodering, telex en het inlezen van hobbyscoop programma's is het noodzakelijk dat men beschikt over een data-ingang (voor seriele invoer).

De hier beschreven methode maakt gebruik van het toetsenbord circuit. Het bestaat uit een stukje software (machinetaal) en een simpele hardware schakeling. De hardware bestaat uit een optocoupler en een weerstand (bij mij was een weerstand van 1K nodig). De weerstand is zo gekozen, dat de optocoupler (elk type voldoet) met een TTL-poort aangestuurd kan worden. De transistor van de optocoupler sluit als het ware de toets. Het stukje software maakt A₁₁ laag en leest D₀ in. Hier is gekozen voor de toets 1 maar elke toets is te testen (zie schema ZX80/81). Als de toets gesloten is, dan wordt D₀ laag. Dit bit staat dan in de accumulator A en wordt getest. Onder de voorwaarde, dat het bit laag is, wordt er uit de lus gesprongen.

J.A.M. Kleijn

Assembler	HEX.	
LD A, %11110111	3E F7	maak A ₁₁ laag
IN A, %11111111	DB FE	lees D ₀ in
BIT 0,A	CB 47	test bit
RET,Z	C8	
JR,-9	18 F7	



SOFTWARE

INHOUD

Teken programma	01.5.2.001	01-27
Cassette index	01.5.2.002	01-27
Nuclear Power Plant	01.5.4.001	01-28
Kiengetal trekken	01.5.4.006	01-33
Torens van Hanoi	01.5.4.007	01-36
Boter, Melk, Kaas	01.5.4.008	01-37
Getallen omdraaien	01.5.4.009	01-38
Dobbelsteen gooien	01.5.4.009	01-38
Kasboek	01.5.5.001	01-39
Berekening van N-faculteit	01.5.5.005	01-43

S O F T W A R E - B I B L I O T H E E K

Laat de software-bibliotheek groeien !!!!

Zend Uw cassette met:

Eigen programma's

of Bewerkingen van programma's geschreven
voor andere computers (met bronvermelding)

Aan: H.C. Telman
Postbus 258
5460 AG Veghel.

De cassette krijgt U op de eerstvolgend bijeenkomst terug.

£			£
£	PRINTER-papier	f 11.50 per rol	£
£		f 22.50 per 2 rol	£
£			£
£	C-15 CASSETTES	f 10.- per 3 stuks	£
£			£
£	Alleen verkrijgbaar op de bijeenkomsten.		£
£			£

LISTING TEKEN PROGRAMMA, CASSETTE INDEX

```

90 FOR B=9 TO 19
100 PRINT AT 8,0;"1";AT 8,15;"1"
110 NEXT B
120 PRINT AT 19,0;"1"
130 PRINT AT 9,4;"KANT 1";AT 9,
140 PRINT AT 10,0;"1"
150 FOR Z=1 TO 16
160 IF Z<=8 THEN PRINT AT 20,0;"
170 IF Z>8 THEN PRINT AT 20,0;"
180 INPUT B$ THEN PRINT AT 10+Z,
190 IF Z<=8 THEN PRINT AT 2+Z,16
200 IF Z>8 THEN PRINT AT 20,0;"
210 NEXT Z
220 REM (C) 1982 H.V. KERKHOFF
    SINCLAIR GEBRUIKERSGROEP

```

ZX 81 CASSETTE NUMMER 1

ZX 81 CASSETTE NUMMER 1 INDEX

KANT 1

KANT 2

1200+5007-0

1200+5007-0

```

5 REM "TEKENPROGRAMMA"
6 REM MET 0 KUN JE PLOT AAN-
7 OF UITSCHAKELLEN, MET 5 T/M 8 GEE
8 JE DE RICHTING VAN DE LIJNEN A
9 AN
10 LET X=0
11 LET Y=0
12 LET B=1
13 IF INKEY$="" THEN GOTO 40
14 LET A$=INKEY$
15 IF A$="0" THEN LET B=-B
16 UNPLOT X,Y
17 PLOT X,Y
18 IF B=1 THEN UNPLOT X,Y
19 IF A$="" THEN GOTO 50
20 LET X=X-(VAL A$=5)+(VAL A$=
21 LET Y=Y-(VAL A$=6)+(VAL A$=
22 IF X<0 THEN LET X=0
23 IF X>63 THEN LET X=63
24 IF Y<0 THEN LET Y=0
25 IF Y>43 THEN LET Y=43
26 PLOT X,Y
27 GOTO 50

```

AUTEUR: G UD HORST
 HENGELOLAAN 1158
 2544 GT DEN HAAG

10 PRINT "CASSETTE NO ?"

110 INPUT A

120 GOTO 10

50 LET A\$="ZX 81 CASSETTE NUMM

60 PRINT AT 1,4;A\$;A

70 PRINT AT 6,0;"1"

80 PRINT AT 7,0;"1";A\$;A;" IND

EX",AT 7,31;"1"

LISTING NUCLEAR POWER PLANT.

```

1 REM APPLE NUCLEAR POWER PLA
2 VAN:STEPHEN R BERGGREN
3 UUT:CREATIVE COMPUTING
4 DECEMBER 1980
5 REM RANGEPAST VOOR ZX81 16K
6 LET DMGE=0
7 LET RD=0
8 LET ED=0
9 LET PD=0
10 LET XD=0
11 LET SD=0
12 LET GD=0
13 LET PB=0
14 LET SB=0
15 LET XB=0
16 LET GB=0
17 LET DAY=0
18 RAND
19 SLOW
20 PRINT TAB 12; "EX-81-"
25 PRINT
30 PRINT TAB 6; "NUCLEAR POWER
PLANT
40 PRINT "ENTER "I" OR "S""
50 PRINT
60 PRINT "I" = INSTRUCTIONS
70 PRINT "S" = START"
80 INPUT X$
90 IF X$="S" THEN GOTO 390
100 IF X$="I" THEN GOTO 50
110 GOSUB 6000
120 PRINT "OPERATE CONTROLS BY
TYPING IN"
130 PRINT "DESIRED CONTROLLED E
SETTING AND"
140 PRINT "FLOW RATES (0-100)"
150 PRINT "NO CHANGE WHEN NO EN
TRY
160 PRINT "USE N/L TO STEP TO D
IFFERENT"
170 PRINT "FUNCTIONS."
180 PRINT "AFTER ENTRIES, USE "
A" FOR"
190 PRINT "NEXT DAY"
200 PRINT "OPERATION ENDS WHEN
MELTDOWN"
210 PRINT "OCCURS OR FUEL IS EX
HAUSTED"
220 PRINT "WHEN FUEL IS EXHAUST
ED"
230 PRINT "AN EVALUATION WILL B
E MADE"
240 PRINT AT 20,3;"HIT NEW-LINE"
250 INPUT X$
255 CLS
260 PRINT "FOR DAMAGE-REPAIR OR
COOLANT"
270 PRINT "REPLACEMENT, BRING RE
C-TEMP."
280 PRINT "BELOW 100 AND SHUT C
OOLANT FLOW"
290 PRINT "THIS GIVES AUTOMATIC
MAINTENANCE"
300 PRINT "SHUTDOWN AND COOLANT
WILL BE"
310 PRINT "REPLENISHED AND REPA
IR IS MADE"
320 PRINT "THE GREATER THE DAMA
GE"
330 PRINT "THE LONGER THE REPAI
R IS"
340 PRINT "WARNING: NO AUTOMATI
C SAFETY"
350 PRINT "DEVICES"
360 PRINT AT 20,0;"HIT NEW-LINE"
380 INPUT X$
382 CLS
385 GOTO 50
390 REM --INITIATE--
400 GOSUB 2000
410 LET RH=0
420 LET RL=0
430 LET DAY=0
435 LET TT=0
440 LET DMGE=0
445 LET A1=0
450 LET A2=0
455 LET A3=0
460 CLS
470 CLS
475 LET DAY=DAY+1
480 PRINT TAB 3;"STATUS REPORT-
DAY"
490 PRINT
510 PRINT "WARNINGS:"
520 IF AT>500 THEN PRINT "REACT

```



```

OR OVERHEATED:
522 IF RT>800 THEN LET RD=RD+1+
(RT>850)+(RT>900)+2*(RT>950)
523 IF RT>900 THEN LET PD=PD+1
527 IF RT>800 THEN LET ED=ED+1+
(RT>850)
530 IF XT>500 THEN PRINT "HEATE
XCHANGER OVERHEAT"
532 IF XT>500 THEN LET XD=XD+1+
(XT>600)
535 IF XT>500 THEN LET PD=PD+1
537 IF XT>500 THEN LET SD=SD+1
540 IF GO>2000 THEN PRINT "TURB
INE OVERLOADED"
542 IF GO>2000 THEN LET GD=GD+1
+ (GO>2500)
545 IF GO>2000 THEN LET SD=SD+1
550 IF CT>300 THEN PRINT "COOLI
NGTOWER OVERHEAT"
552 IF CT>300 THEN LET SD=SD+1
555 IF GO<1000 THEN PRINT "POWE
ROUTPUT LOW"
570 IF EV<200 THEN PRINT "EMERG
COOLANT LOW"
580 IF PV<100 THEN PRINT "PRIM.
COOLANT LOW"
585 IF PV<100 THEN LET PD=PD+1
590 IF SV<100 THEN PRINT "SEC.
COOLANT LOW"
595 IF SV<100 THEN LET SD=SD+1
610 PRINT "DAMAGE"
620 IF RD>5 THEN PRINT "REACTOR
CORE DAMAGED"
630 IF PD>4 THEN PRINT "PRIM. C
COOLANT LEAK"
635 IF PD>4 THEN LET PV=(PV-PD)
*( (PV-SD)>0)
640 IF SD>4 THEN PRINT "SEC. CO
OLANT LEAK"
645 IF SD>4 THEN LET SV=(SV-SD)
*( (SV-SD)>0)
650 IF ED>2 THEN PRINT "EMERG.
COOLANT LEAK"
655 IF ED>2 THEN LET EV=(EV-2*E
D)+( (EV-2*ED)>0)
660 IF PD>8 THEN PRINT "PRIM. COO
LANT PUMP FAILURE"
665 IF PD>8 THEN PRINT "SEC. COOL
ANT PUMP FAILURE"
670 IF SD>8 THEN PRINT "SEC. COOL
ANT PUMP FAILURE"
675 IF SD>8 THEN PRINT "HEATEXCHA
NGER FAILURE"
680 IF GO>5 THEN PRINT "TURBINE F
AULTURE"
700 IF RD>5 THEN PRINT AT 18.0;
"REACTOR CORE OVERHEATED"
710 IF RD>5 THEN PRINT AT 18.0;
"REACTOR PUMP FAILURE"
715 IF RD>5 THEN PAUSE 200
717 IF RD>5 THEN GOTO 3000
719 PRINT AT 20.0; "HIT NEW-LINE"
718 INPUT X$
719 CLS
720 PRINT "INDICATORS:"
725 PRINT "REACTOR TEMP. (MAX. 800
)"; RT
730 PRINT "HEATEXCHANGER TEMP. (MA
X. 500)"; XT
735 PRINT "COOLINGTOWER TEMP. (MA
X. 300)"; CT
740 PRINT "POWEROUTPUT (MAX. 2000
KW)"; GO
750 LET KU=INT (TT/DAY)
760 PRINT "AVERAGE OUTPUT "; KW)
770 DAY=DAY+KU
780 PRINT "CONTROL RODS "; R
790 PRINT "COOLANTS"
800 PRINT "EMERG.LEVEL "; EV; " F
LOW "; EF
830 PRINT "PRIM.LEVEL "; PV; " FL
OW "; PF
840 PRINT "SEC.LEVEL "; SV; " FLO
W "; SF
842 IF (100-RL)<5 THEN CLS
850 IF (100-RL)<5 THEN PRINT AT
12.5; "REACTOR CORE OVERHEATED"
855 IF (100-RL)<5 THEN PAUSE 20
0
858 IF (100-RL)<5 THEN GOTO 400
900 REM --GET NEW CONTROL VALUE
950 LET A2=A1
951 LET A1=A
952 PRINT "CONTR. RODS?";
955 LET B$=" "
956 FOR I=1 TO 2
960 INPUT A$
961 LET Z=CODE A$
965 IF (Z<>38 AND Z<>0) AND (Z<
37 OR Z<28) THEN GOTO 960

```

```

1151 LET SF=SF+(100-SF)*(SF>100)
1160 PRINT A$;
1161 NEXT I
1162 PRINT
1163 CLS
1165 GOTO 800
1170 IF PF=0 AND SF=0 AND RH<1 A
ND RT<100 AND A=0 THEN GOSUB 200
0
1171 IF PF=0 AND SF=0 AND RH<1 A
ND RT<100 AND A=0 THEN CLS
1172 IF PF=0 AND SF=0 AND RH<1 A
ND RT<100 AND A=0 THEN PRINT AT
12.5: "EMERGENCY - CHECK FLOW!"
1175 IF PF=0 AND SF=0 AND RH<1 A
ND RT<100 AND A=0 THEN PRINT AT
14.14: "NO. "
1178 IF PF=0 AND SF=0 AND RH<1 A
ND RT<100 AND A=0 THEN PAUSE 150
AND;
1180 IF EF=EV THEN LET EF=EV
1200 REM -- DAMAGE ASSESSMENT AM
0 OPERATION CALCULATIONS --
1205 LET EV=EV-EF-2*ED*(ED>3)
1210 LET PD=PD+(PF>90)*(RND>.95)
1220 LET SD=SD+(SF>90)*(RND>.92)
1230 LET PB=PD>5
1240 LET SB=SD>5
1250 IF PF>(100-PD*10) AND PB TH
EN LET PF=(100-PD*10)*(100-PD*10
>0)
1260 IF SF>(100-SD*10) AND SB TH
EN LET SF=(100-SD*10)*(100-SD*10
>0)
1270 LET RL=RL+RH/50
1280 LET RH=(A*30+A1*60+A2*10)/2
1300*(100-RL)
1300 LET PH=PF*(100*(PV)100)+PU*
(PV<100)/350
1310 LET EH=EF/200*(RT-25)
1320 LET RT=INT (RT+RH-EH-PH-S*(
RT>25))
1325 LET RT=INT (25+(RT-25)*(RT>
25))
1330 LET XT=INT ((RT-25)*PF+(CT
-25)*SF)/(PF+SF+1)+25)
1340 IF XB THEN LET XT=INT (RT+.
5+5)
1350 LET SH=SF*(100*(PV)100)+PU*
(PV<100)/350*(XT-CT)
1360 IF XB THEN LET SH=SH*.2

```

```

970 LET B$=B$+A$
971 IF Z=38 THEN GOTO 1170
975 IF Z=0 THEN GOTO 988
980 LET A=VAL B$
981 LET A=A+(100-A)*(A>100)
985 PRINT A$;
986 NEXT I
990 PRINT "EMERG.FLOW?";
1000 PRINT B$="";
1005 FOR I=1 TO 2
1006 INPUT A$
1010 LET Z=CODE A$
1011 IF (Z>38 AND Z<0) AND (Z>
1015 OR Z<28) THEN GOTO 1010
1020 LET B$=B$+A$
1021 IF Z=38 THEN GOTO 1170
1025 IF Z=0 THEN GOTO 1045
1030 LET EF=VAL B$
1031 LET EF=EF+(100-EF)*(EF>100)
1035 IF EF=EV THEN LET EF=EV
1040 PRINT A$;
1041 NEXT I
1045 PRINT "PRIN.FLOW?";
1050 LET B$="";
1055 FOR I=1 TO 2
1060 INPUT A$
1070 LET Z=CODE A$
1071 IF (Z>38 AND Z<0) AND (Z>
1075 OR Z<28) THEN GOTO 1070
1080 LET B$=B$+A$
1081 IF Z=38 THEN GOTO 1170
1085 IF Z=0 THEN GOTO 1110
1090 LET PF=VAL B$
1091 LET PF=PF+(100-PF)*(PF>100)
1100 PRINT A$;
1101 NEXT I
1110 PRINT "SEC.FLOW?";
1120 LET B$="";
1125 FOR I=1 TO 2
1130 INPUT A$
1131 LET Z=CODE A$
1135 IF (Z>38 AND Z<0) AND (Z>
1140 OR Z<28) THEN GOTO 1130
1141 LET B$=B$+A$
1142 IF Z=38 THEN GOTO 1170
1145 IF Z=0 THEN CLS
1150 IF Z=0 THEN GOTO 800
1153 LET SF=VAL B$

```



```

3090 GOTO 5000
3100 PRINT "REACTORCORE MELTED D
OWN"
3110 PRINT "STEAM EXPLOSION. CON
TAINMENT"
3120 PRINT "BUILDING RUPTURED."
3130 PRINT "RADIOACTIVE GASES ES
CAPED."
3160 PRINT "EVACUATE AND CLEANUP
3170 PRINT "GET MEDICAL ASSISTAN
CE."
3190 GOTO 5000
4000 REM -- EVALUATION OF GAME RE
SULTS
4010 CLS
4020 PRINT "OVER "; DAY; " DAYS, YO
U PRODUCED "
4030 PRINT "AN AVERAGE OUTPUT OF
"; KU
4040 PRINT "KU PER DAY."
4050 LET AKU=INT (1+(KU>1000)+(K
U>1200)+(KU>1500)+(KU>1800))
4060 PRINT
4070 PRINT "YOUR POWERPROD.RATE
IS:"
4080 IF AKU=1 THEN PRINT "HORRIB
LE. FIND ANOTHER JOB."
4085 IF AKU=2 THEN PRINT "UAY BE
LOW YOUR POWER NEEDS"
4090 IF AKU=3 THEN PRINT "ADEQUA
TE. YOU COULD DO BETTER"
4100 IF AKU=4 THEN PRINT "EXCELL
ENT. POWER COSTS IN YOUR"
4110 IF AKU=4 THEN PRINT "AREA I
T WILL NOT BE INCREASED"
4140 IF AKU=5 THEN PRINT "NEAR M
AXIMUM. POWER COSTS"
4150 IF AKU=5 THEN PRINT "WILL D
ROP SIGNIFICANTLY."
4200 REM -- DAMAGE EVALUATION
4210 GOSUB 2000
4220 LET D=INT (1+(DMGE>10)+(DMG
E>20)+(DMGE>30))
4225 PRINT
4230 PRINT "EQUIPMENTDAMAGE SUST
AINED WAS:"
4240 IF D=1 THEN PRINT "VERY LIG
HT"
4270 IF D=2 THEN PRINT "MODERATE

```

```

1370 LET GO=INT (SH/XT*(XT-CT)*2
/3)
1375 IF GO>2500 THEN LET GO=2500
1380 LET GO=GO*(GO>0)*(GB=0)
1390 LET CT=INT (25+(XT-25)*(SH
-100)/(SH+1)*.75)
1395 LET CT=25*(CT<=25)+CT*(CT>2
5)
1400 IF XB<1 THEN LET XB=INT (IX
D/2)*(AND>.9)
1410 IF GB<1 THEN LET GB=INT (IG
D/4)*(AND>.9)
1420 LET TT=TT+GO
1430 GOTO 470
1500 REM -- MAINTENANCE REPAIR SU
BROUTINE
20010 LET EV=300
20020 LET PV=120
20030 LET SV=120
20040 LET RT=25
20050 LET XT=25
20060 LET CT=25
2070 LET DMGE=DMGE+2*RD+ED+PD+XD
+SD+GD
2080 LET MD=5+3*(10*(RD>3)+(ED>3
)+(PD>3)+(SD>3)+2*PB+2*SB+3*XB+3
*GB)
2085 LET DAY=DAY+MD
2090 LET RD=0
2100 LET ED=0
2110 LET PD=0
2120 LET XD=0
2130 LET SD=0
2140 LET GD=0
2150 LET PB=0
2160 LET SB=0
2170 LET XB=0
2180 LET GB=0
2190 LET EF=0
2200 LET SF=0
2210 LET GO=0
2220 RETURN
3000 REM -- MELTDOWN ENDING
30010 CLS
30020 IF RD>6 THEN GOTO 3100
3030 PRINT "REACTORCORE DESTROYE
D"
3050 PRINT "CONTAINMENT BUILDING
NOT"
3050 PRINT "RUPTURED. EVACUATE..."

```



```

4280 IF D=3 THEN PRINT "HEAVY"
4290 IF D=4 THEN PRINT "SEVERE"
5000 REM -*-END
5010 PRINT "TRY AGAIN(J/N)?"
5020 INPUT X$
5045 IF X$="J" THEN GOTO 390
5060 CLS
5070 STOP
6000 REM EXPLANATION
6005 CLS
6010 PRINT "SIMULATION OF A NUCL
6020 PRINT "REACTOR. OPERATE THE
6030 PRINT "AT A"
6040 PRINT "MAX. AVERAGE POWEROU
6050 PRINT "AVOID REACTOR MELTDOW
6060 PRINT "CONTROLRODS ADJUST A
6070 PRINT "HEAT PRODUCED"
6080 PRINT "PRIM. COOLANT TRANSF
6090 PRINT "HEAT TO"
6100 PRINT "HEATEXCHANGER"
6110 PRINT "SEC. COOLANT TRANSFE
6120 PRINT "HEATEXC. TO COOLINGT
6130 PRINT "EMERG. COOLANT IS US
6140 PRINT "SHUTDOWN WHEN OTHERS
6150 PRINT "COOLANTS CAN BE RECY
6160 PRINT "EXCEPT THE EMERGENCY
6170 PRINT "COOLANT"
6180 PRINT AT 20,0:"HIT NEW-LINE
6190 INPUT X$
6200 RETURN
6210 REM VARIABLE PREFIXES
6220 REM A-COOLANT RODS
6230 REM C-COOLINGTOWER
6240 REM E-EMERGENCY COOLANT
6250 REM G-TURBINE
6260 REM P-PRIMARY COOLANT
6270 REM R-REACTOR

```

```

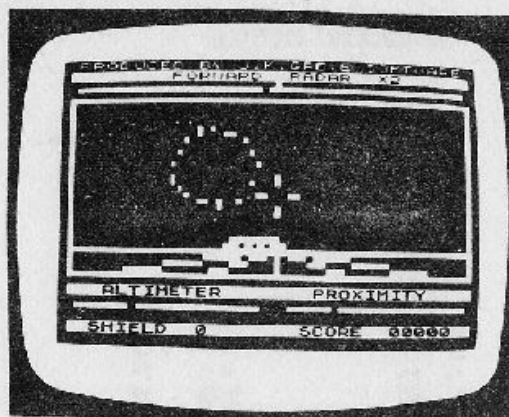
7017 S-SECONDARY COOLANT
7018 X-HEAT EXCHANGER
7019 VARIABLE SUFFIXES
7020 A-BROKEN
7021 D-DAMAGE
7022 F-FLOW RATE
7023 H-HEAT FLOW
7024 J-LIFE
7025 O-OUTPUT
7026 T-TEMPERATURE
7027 V-VOLUME
7028 OTHER VARIABLES
7029 TOT-TOTAL POWER OUTPUT
7030 KW-AVERAGE POWER OUTPUT
7031 DAY-DAY OF OPERATION
7032 DMGE-TOTAL EQUIPMENT DA
7033 *** APPLE NUCLEAR ***
7034 *** POWER PLANT ***
7035 REM.BY STEPHEN R BERGGREN

```

— 0 — 0 —

**MEER DAN
100**

**PROGRAMMA'S
VAN**

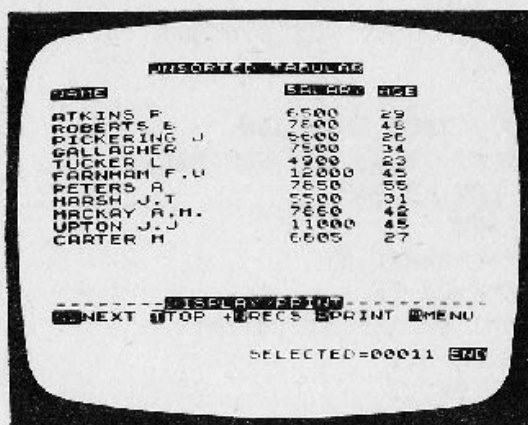


3D DEFENDERS

-MICHAEL ORWIN-
-J.K.GREYE SOFTWARE-
-CAMPBELL SYSTEMS-
-ARTIC COMPUTING-
-CONTROL TECHNOLOGY-
-J.R.S. SOFTWARE-
-QUICKSILVA-

AACKOSOFT

**SINCLAIR
ZX 81
SOFTWARE**



THE FAST ONE

► PROFESSIONELE TOEPASSINGEN

► AVONTURENSPELPROGRAMMA's

► TOOLS & PROGRAMMERS AID's

► EDUCATIEVE PROGRAMMA's

► RUIMTESPELPROGRAMMA's

► BEHENDIGHEIDSPROGRAMMA's

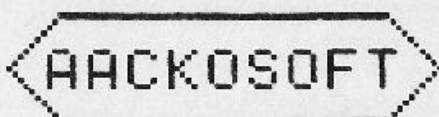
► DENKSPORTPROGRAMMA's

ONZE SOFTWARE IS TE KOOP BIJ :

Amsterdam	ALLWAVE Damrak 91, Utrechtsestraat 110-112
Amsterdam	Computercollectief Amstel 312
Arnhem	ALLWAVE Grote Oord 4
Breda	ALLWAVE Ginnekenstraat 131
Delft	ALLWAVE Oude Langendijk 13
Eindhoven	ALLWAVE Markt 34a
Den Haag	ALLWAVE Passage 54
Gouda	KASS Systeemhouse Walenstraat 4
Haarlem	ALLWAVE Grote Houtstraat 16
Hoogvliet	Radio Oudeland Binnenban
Leiden	ALLWAVE Haarlemmerstraat 113
Rotterdam	ALLWAVE Hoogstraat 171
Rotterdam	Bijenkorf
Rotterdam	DIL Electronica Jan Ligthartstraat 59-71
Rotterdam	ELRA Zwartjanstraat 38
Utrecht	ALLWAVE Oude Gracht 163

INLICHTINGEN:

BESTELLINGEN:



Postbus 3111
 2301 DC Leiden
 Tel. 01880-11446

01-36


```

1160 LET T=A
1170 PRINT AT 19,0"#####
#####"
1180 RETURN

```

Opmerking: & = spatie.

LISTING BOTER-MELK-KAAS

```

1 REM BOTER-MELK-KAAS, 12-2-19
82, W. BOERDIJK, SINCLAIR ZX-80
2 CLS
3 PRINT "NAAM EERSTE SPELER?"
4 INPUT A$
5 PRINT "NAAM TWEEDE SPELER?"
6 INPUT B$
7 LET A=1
8 LET B=2
9 LET C=3
10 LET D=4
11 LET E=5
12 LET F=6
13 LET G=7
14 LET H=8
15 LET I=9
16 FOR J=1 TO 4
17 CLS
18 PRINT A,B,C
19 PRINT
20 PRINT D,E,F;" ";A$
21 PRINT
22 PRINT G,H,I
23 INPUT K
24 CLS
25 IF K=1 THEN LET A=10
26 IF K=2 THEN LET B=20
27 IF K=3 THEN LET C=30
28 IF K=4 THEN LET D=40
29 IF K=5 THEN LET E=50
30 IF K=6 THEN LET F=60
31 IF K=7 THEN LET G=70
32 IF K=8 THEN LET H=80
33 IF K=9 THEN LET I=90

```

```

34 PRINT A,B,C
35 PRINT
36 PRINT D,E,F;" ";B$
37 PRINT
38 PRINT G,H,I
39 IF Z+B+C=60 OR D+E+F=150 OR
G+H+I=240 OR A+D+G=120 OR B+E+H
=150 OR C+F+I=180 OR A+E+I=150 O
R C+E+G=150 THEN GOTO 57
40 INPUT M
41 IF M=K THEN GOTO 40
42 IF M=1 THEN LET A=100
43 IF M=2 THEN LET B=200
44 IF M=3 THEN LET C=300
45 IF M=4 THEN LET D=400
46 IF M=5 THEN LET E=500
47 IF M=6 THEN LET F=600
48 IF M=7 THEN LET G=700
49 IF M=8 THEN LET H=800
50 IF A+B+C=600 OR D+E+F=1500
OR G+H+I=2400 OR A+D+G=1200 OR B
+E+H=1500 OR C+F+I=1800 OR A+E+I
=1500 OR C+E+G=1500 THEN GOTO 55
52 NEXT J
53 PRINT "GELIJKSPEL"
54 GOTO 58
55 PRINT A$;" HEEFT GEWONNEN"
56 INPUT X
57 PRINT B$;" HEEFT GEWONNEN"
58 INPUT X
59 GOTO 2
60 STOP

```


LISTING GETALLEN OMDRAAIEN

```

1 REM GETALLEN OMDRAAIEN, 29-1
-1982, W. BOERDIJK, SINCLAIR ZX80
2 PRINT "GETALLEN OMDRAAIEN"
3 PRINT
4 DIM A(20)
5 PRINT "HOEVEEL GETALLEN? (MAXIMAAL 20)"
6 INPUT N
7 LET A(1)=RND(N-1)+2
8 FOR K=2 TO N
9 LET A(K)=RND(N)+1
10 FOR J=1 TO K-1
11 IF A(K)=A(J) THEN GOTO 9
12 NEXT J
13 NEXT K
14 PRINT "DAAR GAAN WE..."
15 LET T=0
16 GOSUB 40
17 PRINT "HOEVEEL GETALLEN OMDRAAIEN? (GOEDE VOLGORDE? DUW OP 0)"
18 INPUT R
19 CLS
20 IF R=Q THEN GOTO 34
21 IF R<N OR R=N THEN GOTO 24
22 PRINT "HO, MAXIMAAL "; N
23 GOTO 17
24 LET T=T+1
25 FOR K=1 TO (R/2)
26 LET Z=A(K)
27 LET A(K)=A(R-K+1)
28 LET A(R-K+1)=Z
29 NEXT K
30 GOSUB 40
31 FOR K=1 TO N
32 IF A(K)<K OR A(K)>K THEN GOTO 17
33 NEXT K
34 PRINT "JE HEBT GEWONNEN IN "; T; " BEURTEN"
35 PRINT "NOG EEN KEER? (J/N)"
36 INPUT A$
37 IF A$="J" THEN GOTO 4
38 PRINT "IK HOOP DAT JE HET LUK VOND"
39 GOTO 44
40 FOR K=1 TO N
41 PRINT A(K)-1; " ";
42 NEXT K
43 RETURN
44 STOP

```

LISTING DOBBELSTENEN GOOIEN TEN BEHOEVE VAN JAHTZEE

```

1 REM JAHTZEE, 25-1-1982, W. BOERDIJK, SINCLAIR ZX-80
2 PRINT "JAHTZEE"
3 PRINT "====="
4 PRINT
5 PRINT "IK ZAL 3x GOOIEN MET MAXIMAAL 6 STENEN"
6 FOR Q=1 TO 3
7 PRINT "MET HOEVEEL STENEN GOOI IK?"
8 INPUT Z
9 IF Z>6 OR Z<1 THEN GOTO 28
10 IF Z=6 THEN FOR X=1 TO 6
11 IF Z=5 THEN FOR X=1 TO 5
12 IF Z=4 THEN FOR X=1 TO 4
13 IF Z=3 THEN FOR X=1 TO 3
14 IF Z=2 THEN FOR X=1 TO 2
15 IF Z=1 THEN FOR X=1 TO 1
16 LET A=RND(6)
17 IF A=1 THEN PRINT "EEN"
18 IF A=2 THEN PRINT "TWE"
19 IF A=3 THEN PRINT "DRI"
20 IF A=4 THEN PRINT "VIER"
21 IF A=5 THEN PRINT "VIJF"
22 IF A=6 THEN PRINT "ZES"
23 NEXT X
24 NEXT Q
25 INPUT M$
26 IF M$="" THEN CLS
27 GOTO 2
28 PRINT "IK ZEI: MAXIMAAL 6 EN NIET "; Z
29 GOTO 24
30 STOP

```

LISTING KASBOEK.

```

0000 PRINT AT 10,9;"KASBOEK";
0005 PRINT AT 11,9;" ";
0007 PRINT AT 12,9;" ";
0010 FOR X=1 TO 40
0020 NEXT X
0030 CLS
0040 PRINT AT 0,6;"MENU";
0050 PRINT AT 1,6;" ";
0060 PRINT AT 2,6;" ";
0070 PRINT AT 3,6;"OVERZICHT..";
0080 PRINT AT 4,6;"ABONNEMENT.";
0090 PRINT AT 5,6;"ONTSPANNING";
0100 PRINT AT 6,6;"BELASTING..";
0110 PRINT AT 7,6;"VERZEKERING";
0120 PRINT AT 8,6;"DIVERSEN...";
0130 PRINT AT 9,6;"ENERGIE....";
0140 PRINT AT 10,6;"HUUR.....";
0150 PRINT AT 11,6;"HUISHOUDIN";
0160 PRINT AT 12,6;"KLEDING...";
0170 PRINT AT 13,6;"LENINGEN..";
0180 PRINT AT 14,6;"RESERVE...";
0190 PRINT AT 15,6;"STOPPEN...";
0200 PRINT AT 16,6;" ";
0210 PRINT AT 17,6;" ";
0220 PRINT AT 19,7;"MARK UW KEU";
0230 INPUT 0;
0240 PRINT AT 21,22;"STAND BY*";
0250 GOSUB 0*500+500
0260 GOTO 300
0270 CLS
0280 PRINT "OVERZICHT"
0290 PRINT AT 0,20;"DATUM: ";Y(1)

```



```

11500 RETURN
11501 FOR Z$(X)=1 TO 16
11502 LET Z$(X)=B$(X)
11503 NEXT X
11504 LET M(14)=M(2)
11505 LET Y(14)=Y(2)
11506 LET ZA=TB
11507 GOSUB 7500
11508 FOR X=1 TO 16
11509 LET B(X)=Z(X)
11510 LET B$(X)=Z$(X)
11511 NEXT X
11512 LET M(2)=M(14)
11513 LET Y(2)=Y(14)
11514 LET TB=ZA
11515 RETURN
11516 FOR X=1 TO 16
11517 LET Z(X)=C(X)
11518 LET Z$(X)=C$(X)
11519 NEXT X
11520 LET M(14)=M(3)
11521 LET Y(14)=Y(3)
11522 LET ZA=TC
11523 GOSUB 7500
11524 FOR X=1 TO 16
11525 LET C(X)=Z(X)
11526 LET C$(X)=Z$(X)
11527 NEXT X
11528 LET M(3)=M(14)
11529 LET Y(3)=Y(14)
11530 LET TC=ZA
11531 RETURN
11532 FOR X=1 TO 16
11533 LET Z(X)=D(X)
11534 LET Z$(X)=D$(X)
11535 NEXT X
11536 LET M(14)=M(4)
11537 LET Y(14)=Y(4)
11538 LET ZA=TD
11539 GOSUB 7500
11540 FOR X=1 TO 16
11541 LET D(X)=Z(X)
11542 LET D$(X)=Z$(X)
11543 NEXT X
11544 LET M(4)=M(14)
11545 LET Y(4)=Y(14)
11546 LET TD=ZA
11547 RETURN
11548 FOR X=1 TO 16
11549 LET Z(X)=E(X)
11550 LET Z$(X)=E$(X)
11551 NEXT X
11552 LET M(14)=M(14)
11553 LET Y(14)=Y(14)
11554 LET TD=ZA
11555 RETURN
11556 FOR X=1 TO 16
11557 LET Z(X)=F(X)
11558 LET Z$(X)=F$(X)
11559 NEXT X
11560 LET M(14)=M(14)
11561 LET Y(14)=Y(14)
11562 LET TD=ZA
11563 RETURN
11564 FOR X=1 TO 16
11565 LET Z(X)=G(X)
11566 LET Z$(X)=G$(X)
11567 NEXT X
11568 LET M(14)=M(14)
11569 LET Y(14)=Y(14)
11570 LET TD=ZA
11571 RETURN
11572 FOR X=1 TO 16
11573 LET Z(X)=H(X)
11574 LET Z$(X)=H$(X)
11575 NEXT X
11576 LET M(14)=M(14)
11577 LET Y(14)=Y(14)
11578 LET TD=ZA
11579 RETURN
11580 FOR X=1 TO 16
11581 LET Z(X)=I(X)
11582 LET Z$(X)=I$(X)
11583 NEXT X
11584 LET M(14)=M(14)
11585 LET Y(14)=Y(14)
11586 LET TD=ZA
11587 RETURN
11588 FOR X=1 TO 16
11589 LET Z(X)=J(X)
11590 LET Z$(X)=J$(X)
11591 NEXT X
11592 LET M(14)=M(14)
11593 LET Y(14)=Y(14)
11594 LET TD=ZA
11595 RETURN
11596 FOR X=1 TO 16
11597 LET Z(X)=K(X)
11598 LET Z$(X)=K$(X)
11599 NEXT X
11600 LET M(14)=M(14)
11601 LET Y(14)=Y(14)
11602 LET TD=ZA
11603 RETURN
11604 FOR X=1 TO 16
11605 LET Z(X)=L(X)
11606 LET Z$(X)=L$(X)
11607 NEXT X
11608 LET M(14)=M(14)
11609 LET Y(14)=Y(14)
11610 LET TD=ZA
11611 RETURN
11612 FOR X=1 TO 16
11613 LET Z(X)=M(X)
11614 LET Z$(X)=M$(X)
11615 NEXT X
11616 LET M(14)=M(14)
11617 LET Y(14)=Y(14)
11618 LET TD=ZA
11619 RETURN
11620 FOR X=1 TO 16
11621 LET Z(X)=N(X)
11622 LET Z$(X)=N$(X)
11623 NEXT X
11624 LET M(14)=M(14)
11625 LET Y(14)=Y(14)
11626 LET TD=ZA
11627 RETURN
11628 FOR X=1 TO 16
11629 LET Z(X)=O(X)
11630 LET Z$(X)=O$(X)
11631 NEXT X
11632 LET M(14)=M(14)
11633 LET Y(14)=Y(14)
11634 LET TD=ZA
11635 RETURN
11636 FOR X=1 TO 16
11637 LET Z(X)=P(X)
11638 LET Z$(X)=P$(X)
11639 NEXT X
11640 LET M(14)=M(14)
11641 LET Y(14)=Y(14)
11642 LET TD=ZA
11643 RETURN
11644 FOR X=1 TO 16
11645 LET Z(X)=Q(X)
11646 LET Z$(X)=Q$(X)
11647 NEXT X
11648 LET M(14)=M(14)
11649 LET Y(14)=Y(14)
11650 LET TD=ZA
11651 RETURN
11652 FOR X=1 TO 16
11653 LET Z(X)=R(X)
11654 LET Z$(X)=R$(X)
11655 NEXT X
11656 LET M(14)=M(14)
11657 LET Y(14)=Y(14)
11658 LET TD=ZA
11659 RETURN
11660 FOR X=1 TO 16
11661 LET Z(X)=S(X)
11662 LET Z$(X)=S$(X)
11663 NEXT X
11664 LET M(14)=M(14)
11665 LET Y(14)=Y(14)
11666 LET TD=ZA
11667 RETURN
11668 FOR X=1 TO 16
11669 LET Z(X)=T(X)
11670 LET Z$(X)=T$(X)
11671 NEXT X
11672 LET M(14)=M(14)
11673 LET Y(14)=Y(14)
11674 LET TD=ZA
11675 RETURN
11676 FOR X=1 TO 16
11677 LET Z(X)=U(X)
11678 LET Z$(X)=U$(X)
11679 NEXT X
11680 LET M(14)=M(14)
11681 LET Y(14)=Y(14)
11682 LET TD=ZA
11683 RETURN
11684 FOR X=1 TO 16
11685 LET Z(X)=V(X)
11686 LET Z$(X)=V$(X)
11687 NEXT X
11688 LET M(14)=M(14)
11689 LET Y(14)=Y(14)
11690 LET TD=ZA
11691 RETURN
11692 FOR X=1 TO 16
11693 LET Z(X)=W(X)
11694 LET Z$(X)=W$(X)
11695 NEXT X
11696 LET M(14)=M(14)
11697 LET Y(14)=Y(14)
11698 LET TD=ZA
11699 RETURN
11700 FOR X=1 TO 16
11701 LET Z(X)=X(X)
11702 LET Z$(X)=X$(X)
11703 NEXT X
11704 LET M(14)=M(14)
11705 LET Y(14)=Y(14)
11706 LET TD=ZA
11707 RETURN
11708 FOR X=1 TO 16
11709 LET Z(X)=Y(X)
11710 LET Z$(X)=Y$(X)
11711 NEXT X
11712 LET M(14)=M(14)
11713 LET Y(14)=Y(14)
11714 LET TD=ZA
11715 RETURN
11716 FOR X=1 TO 16
11717 LET Z(X)=Z(X)
11718 LET Z$(X)=Z$(X)
11719 NEXT X
11720 LET M(14)=M(14)
11721 LET Y(14)=Y(14)
11722 LET TD=ZA
11723 RETURN
11724 FOR X=1 TO 16
11725 LET Z(X)=AA(X)
11726 LET Z$(X)=AA$(X)
11727 NEXT X
11728 LET M(14)=M(14)
11729 LET Y(14)=Y(14)
11730 LET TD=ZA
11731 RETURN
11732 FOR X=1 TO 16
11733 LET Z(X)=AB(X)
11734 LET Z$(X)=AB$(X)
11735 NEXT X
11736 LET M(14)=M(14)
11737 LET Y(14)=Y(14)
11738 LET TD=ZA
11739 RETURN
11740 FOR X=1 TO 16
11741 LET Z(X)=AC(X)
11742 LET Z$(X)=AC$(X)
11743 NEXT X
11744 LET M(14)=M(14)
11745 LET Y(14)=Y(14)
11746 LET TD=ZA
11747 RETURN
11748 FOR X=1 TO 16
11749 LET Z(X)=AD(X)
11750 LET Z$(X)=AD$(X)
11751 NEXT X
11752 LET M(14)=M(14)
11753 LET Y(14)=Y(14)
11754 LET TD=ZA
11755 RETURN
11756 FOR X=1 TO 16
11757 LET Z(X)=AE(X)
11758 LET Z$(X)=AE$(X)
11759 NEXT X
11760 LET M(14)=M(14)
11761 LET Y(14)=Y(14)
11762 LET TD=ZA
11763 RETURN
11764 FOR X=1 TO 16
11765 LET Z(X)=AF(X)
11766 LET Z$(X)=AF$(X)
11767 NEXT X
11768 LET M(14)=M(14)
11769 LET Y(14)=Y(14)
11770 LET TD=ZA
11771 RETURN
11772 FOR X=1 TO 16
11773 LET Z(X)=AG(X)
11774 LET Z$(X)=AG$(X)
11775 NEXT X
11776 LET M(14)=M(14)
11777 LET Y(14)=Y(14)
11778 LET TD=ZA
11779 RETURN
11780 FOR X=1 TO 16
11781 LET Z(X)=AH(X)
11782 LET Z$(X)=AH$(X)
11783 NEXT X
11784 LET M(14)=M(14)
11785 LET Y(14)=Y(14)
11786 LET TD=ZA
11787 RETURN
11788 FOR X=1 TO 16
11789 LET Z(X)=AI(X)
11790 LET Z$(X)=AI$(X)
11791 NEXT X
11792 LET M(14)=M(14)
11793 LET Y(14)=Y(14)
11794 LET TD=ZA
11795 RETURN
11796 FOR X=1 TO 16
11797 LET Z(X)=AJ(X)
11798 LET Z$(X)=AJ$(X)
11799 NEXT X
11800 LET M(14)=M(14)
11801 LET Y(14)=Y(14)
11802 LET TD=ZA
11803 RETURN
11804 FOR X=1 TO 16
11805 LET Z(X)=AK(X)
11806 LET Z$(X)=AK$(X)
11807 NEXT X
11808 LET M(14)=M(14)
11809 LET Y(14)=Y(14)
11810 LET TD=ZA
11811 RETURN
11812 FOR X=1 TO 16
11813 LET Z(X)=AL(X)
11814 LET Z$(X)=AL$(X)
11815 NEXT X
11816 LET M(14)=M(14)
11817 LET Y(14)=Y(14)
11818 LET TD=ZA
11819 RETURN
11820 FOR X=1 TO 16
11821 LET Z(X)=AM(X)
11822 LET Z$(X)=AM$(X)
11823 NEXT
```

```

4) " -";M(14)
500 PRINT
505 PRINT
510 PRINT
520 PRINT
530 PRINT "TOTAL IN KAS
F";TK
540 PRINT
550 PRINT A$(16)
560 PRINT B$(16)
570 PRINT C$(16)
580 PRINT D$(16)
590 PRINT E$(16)
600 PRINT F$(16)
610 PRINT G$(16)
620 PRINT H$(16)
630 PRINT I$(16)
640 PRINT J$(16)
650 PRINT K$(16)
660 PRINT L$(16)
670 PRINT M$(16)
680 PRINT N$(16)
690 PRINT O$(16)
700 PRINT P$(16)
710 PRINT Q$(16)
720 PRINT R$(16)
730 PRINT S$(16)
740 PRINT T$(16)
750 PRINT U$(16)
760 PRINT V$(16)
770 PRINT W$(16)
780 PRINT X$(16)
790 PRINT Y$(16)
800 PRINT Z$(16)
810 PRINT
820 PRINT
830 PRINT
840 PRINT
850 PRINT
860 PRINT
870 PRINT
880 PRINT
890 PRINT
900 PRINT
910 PRINT
920 PRINT
930 PRINT
940 PRINT
950 PRINT
960 PRINT
970 PRINT
980 PRINT
990 PRINT

```



```

45600 LET ZA=TI
45700 GOSUB 7500
45800 FOR X=1 TO 16
45900 LET I(X)=Z$(X)
46000 NEXT X
46100 LET Y(8)=Y(14)
46200 LET H(8)=H(14)
46300 LET TI=ZA
46400 RETURN
46500 FOR X=1 TO 16
46600 LET Z$(X)=L$(X)
46700 NEXT X
46800 LET M(14)=M(9)
46900 LET Y(14)=Y(8)
47000 LET ZA=TL
47100 GOSUB 7500
47200 FOR X=1 TO 16
47300 LET L$(X)=Z$(X)
47400 NEXT X
47500 LET Y(9)=Y(14)
47600 LET H(9)=H(14)
47700 LET TL=ZA
47800 RETURN
47900 FOR X=1 TO 16
48000 LET Z$(X)=W$(X)
48100 NEXT X
48200 LET XA=TY
48300 LET M(14)=H(10)
48400 LET Y(14)=Y(10)
48500 GOSUB 7500
48600 FOR X=1 TO 16
48700 LET W$(X)=Z$(X)
48800 NEXT X
48900 LET Y(10)=Y(14)
49000 LET H(10)=H(14)
49100 LET TY=ZA
49200 RETURN
49300 FOR X=1 TO 16
49400 LET Z$(X)=S$(X)
49500 NEXT X
49600 LET M(14)=M(11)
49700 LET Y(14)=Y(11)
49800 LET ZA=TS
49900 GOSUB 7500
50000 FOR X=1 TO 16

```

```

50100 LET S(X)=Z(X)
50200 NEXT X
50300 LET M(14)=M(5)
50400 LET Y(14)=Y(5)
50500 LET ZA=TE
50600 GOSUB 7500
50700 FOR X=1 TO 16
50800 LET E(X)=Z$(X)
50900 NEXT X
51000 LET M(5)=M(14)
51100 LET Y(5)=Y(14)
51200 LET TE=ZA
51300 RETURN
51400 FOR X=1 TO 16
51500 LET Z$(X)=G$(X)
51600 NEXT X
51700 LET M(14)=M(6)
51800 LET Y(14)=Y(6)
51900 LET ZA=TG
52000 GOSUB 7500
52100 FOR X=1 TO 16
52200 LET G$(X)=Z$(X)
52300 NEXT X
52400 LET M(6)=M(14)
52500 LET Y(6)=Y(14)
52600 LET TG=ZA
52700 RETURN
52800 FOR X=1 TO 16
52900 LET Z$(X)=H$(X)
53000 NEXT X
53100 LET Y(14)=Y(7)
53200 LET H(14)=H(7)
53300 LET HA=TH
53400 GOSUB 7500
53500 FOR X=1 TO 16
53600 LET H$(X)=Z$(X)
53700 NEXT X
53800 LET M(7)=M(14)
53900 LET Y(7)=Y(14)
54000 LET TH=ZA
54100 RETURN
54200 FOR X=1 TO 16
54300 LET Z$(X)=I$(X)
54400 NEXT X
54500 LET M(14)=M(8)
54600 LET Y(14)=Y(8)

```

```

6100 LET S$(X)=Z$(X)
6110 NEXT X
6120 LET Y(11)=Y(14)
6130 LET M(11)=M(14)
6140 LET TS=ZA
6150 RETURN
6500 CLS
6510 PRINT AT 21,22;"*STAND BY*"
6520 PRINT AT 9,0;"U MOET HET PR
OGRAMMA NOG OP CASSETTE LADE
N."
6530 PRINT
6540 PRINT "START UW RECORDER (O
PN.)"
6550 FOR X=1 TO 50
6560 NEXT X
6570 CLS
6580 PRINT AT 9,10;"GEBEURD? J/N
"
6590 INPUT Q$
6600 IF Q$="N" THEN GOTO 6500
6610 CLS
6620 SAVE "KASBOEK"
6630 GOTO 200
6640 GOSUB 8209
6650 PRINT AT 20,19;"WILT U BOEK
EN?"
6660 INPUT K$
6670 IF K$="N" THEN PRINT AT 20,
10;"*STAND BY*"
6680 IF K$="N" THEN GOTO 8380
6690 CLS
6700 PRINT "WAT IS DE DATUM? ";
6710 INPUT Y(14)
6720 PRINT Y(14)
6730 PRINT "-----"
6740 PRINT "WAT IS DE MAAND? ";
6750 INPUT M(14)
6760 PRINT M(14)
6770 PRINT AT 3,0;"-----"
6780 PRINT "GEEF OMSCHRIJVING (M
AX. 20)."
6790 PRINT
6800 INPUT Q$
6810 PRINT Q$
6820 PRINT "-----"
6830 PRINT "WAT IS HET BEDRAG?"
6840 INPUT R

```

```

7685 PRINT AT 21,22;"*STAND BY*"
7690 PRINT AT 9,20;"F";R
7700 LET Z(16)=Z(16)+R
7710 LET TK=TK+R
7720 PRINT
7730 PRINT
7745 IF R>0 THEN GOTO 8060
7750 PRINT "AF F ";R;" ";0$
7755 PRINT
7760 IF Z(16)>0 OR Z$(16)=S$(16
) THEN GOTO 8030
7790 PRINT
7800 PRINT "U HAD SALDO TEKORT"
7810 PRINT
7820 PRINT "ER IS F ";-Z(16);" U
AN RESERVE AF"
7840 PRINT
7850 PRINT
7880 LET S(16)=S(16)+Z(16)
7890 LET S(TS)=Z(16)
7900 LET S$(TS)="TEKORT "+Z$(16)
7910 LET TS=TS+1
7920 LET Z(ZA)=-Z(16)
7930 LET Z$(ZA)=-Z$(16)
7940 LET ZA=ZA+1
7950 IF TS<15 THEN GOTO 8020
7960 LET S(1)=S(TS-1)
7970 LET S$(1)=S$(TS-1)
7980 FOR X=2 TO 15
7990 LET S(X)=0
8000 LET S$(X)=""
8005 NEXT X
8010 LET TS=2
8020 LET Z(16)=0
8030 FOR X=1 TO 50
8040 NEXT X
8050 GOTO 8090
8060 PRINT "BIJ F ";R;" ";0$
8065 PRINT
8070 FOR X=1 TO 50
8080 NEXT X
8090 IF TK<0 THEN GOSUB 8400
8100 LET Z(ZA)=R
8110 LET Z$(ZA)=0$
8120 LET ZA=ZA+1

```


LISTING BEREKENING N-FACULTEIT.

Listing BEREKENING N-FACULTEIT.

MET VOLGENDE PROGRAMMA BEREKENT
N-FACULTEIT UIT, ONDER NEDENEMING
VAN ALLE CIJFERS

```

1 REM FACULTEIT/FACTORIAL
2 REM J. VERHOEVEN MOLENSTR.5
3 REM 8881 HQ OISTERWIJK
4 REM 01-03-1982
10 PRINT "BEREKENING VAN N FAC
ULTEIT TOT"
120 PRINT "EN MET HET LAATSTE C
IJFER."
130 PRINT "VOORWAARDE IS DAT N<
15 EN WAAR"
155 PRINT "SCHIJNLIJK IS DIT NO
G TE GROOT."
60 PRINT "WÉLKÉ WAARDE VAN N ?
70 INPUT NO
80 PRINT NO
81 IF NO>0 THEN GOTO 85
82 PRINT "0 FACULTEIT = 1 *150
"
83 GOTO 80
85 FAST
86 LET NO=INT ABS NO
87 LET C1=(NO*(LN NO-1))+.5*LN
(2*PI*NO)/LN 10+1
110 LET CO=NO/5
120 LET C=INT ((C1-CO)/4)+2
140 LET A(C)=1
145 LET N=0 TO 2 STEP -1
150 FOR X=C TO 1 STEP -1
160 LET A(X)=A(X)*N
170 NEXT X
180 LET CARRY=0
190 FOR X=C TO 1 STEP -1
200 LET A(X)=A(X)+CARRY
210 LET CARRY=INT (A(X)/1E4)
220 LET A(X)=A(X)-1E4+CARRY
230 NEXT X

```

```

8130 IF ZA<15 THEN GOTO 8200
8140 LET Z(1)=Z(ZA-1)
8150 LET Z(1)=Z(ZA-1)
8160 FOR X=2 TO 15
8170 LET Z(X)=0
8180 NEXT X
8190 LET ZA=2
8200 PRINT AT 21,0;"BENT U KLAAR
DEZE POST?"
8201 INPUT K$
8202 IF K$="J" THEN GOTO 8209
8203 CLS
8204 PRINT " DATUM: ";Y(14)
8205 PRINT " MAAND: ";M(14)
8206 PRINT "
8207 GOTO 7610
8208 CLS
8209 PRINT Z$(16)
8210 PRINT AT 0,12;"DD AFREKENIN
G";Y(14);";";M(14)
8235 PRINT "
8237 PRINT "
8240 PRINT "BEDRAG:"
8250 PRINT AT 3,11;"OMSCHRIJVING
"
8260 PRINT "
8270 LET Q=5
8280 FOR X=1 TO 13
8290 PRINT "F";Z(X)
8300 PRINT AT 0,11;" ";Z$(X)
8310 LET Q=Q+1
8320 NEXT X
8330 PRINT "
8340 PRINT "SALDO: F";Z(16)
8350 PRINT AT 20,25;"*COPY?*"
8370 INPUT Q$
8375 PRINT AT 20,22;"*STIND ON"
8380 IF Q$="J" THEN COPY
8390 RETURN
8400 CLS
8410 FOR X=1 TO 10
8420 PRINT "U HEEFT SALDO TEKORT
"
8430 PRINT
8440 NEXT X
8445 PRINT "
8450 RETURN

```



```

WELKE WAARDE VAN N ? 10
10 FACULTEIT = 3628800+1E 0
WELKE WAARDE VAN N ? 20
20 FACULTEIT = 2432902000817564+
1E 4
WELKE WAARDE VAN N ? 69
69 FACULTEIT = 1711224524281413
11372468336861272639092270544893
5203693364804002325737975414064
7424000+1E 12
WELKE WAARDE VAN N ? 125
125 FACULTEIT = 100267717688009
26099743767703491600657595403646
714922425887596231506353156331613
5968666293266949592313364640544
593005774063016191034136805978188
63457858547035524326376565007131
77068000+1E 26
WELKE WAARDE VAN N ? 8
8 FACULTEIT = 40320+1E 0

```

```

3000 IF A(C) < 0 THEN GOTO 390
3010 LET NUL=NUL+4
3020 FOR X=C-1 TO 1 STEP -1
3030 LET A(X+1)=A(X)
3040 NEXT X
3050 LET A(1)=0
3060 NEXT N
3070 SLOW
3080 PRINT NO;" FACULTEIT = ";
4100 LET FLAG=0
4110 FOR X=1 TO C
4120 DIM A$(14)
4130 IF A(X) < 0 THEN LET FLAG=FL
AG+1
4140 IF FLAG=0 AND A(X)=0 THEN G
OTO 530
4150 IF NOT (FLAG > 0 AND A(X)=0)
THEN GOTO 455
4160 LET A$="0000"
4170 GOTO 520
4180 LET C=4-INT (LN A(X)/LN 10)
4190 LET B$=STR$ A(X)
4200 IF FLAG > 1 THEN LET A$( TO 0
-1)="0000"
4210 LET A$10 TO 4)=B$
4220 PRINT A$;
4230 NEXT X
4240 PRINT " "+1E ";NUL

```

RUN

BEREKENING VAN N FACULTEIT TOT
 EN MET HET LAATSTE CIJFER.
 VOORWAARDE IS DAT N CIJES EN WAAR-
 SCHIJNLIJK IS DIT NOG TE GROOT.

```

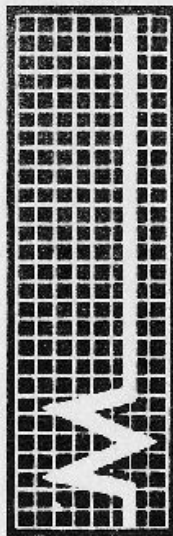
WELKE WAARDE VAN N ? 0
0 FACULTEIT = 1 +1E0
WELKE WAARDE VAN N ? 5
5 FACULTEIT = 120+1E 0

```

TALEN

INHOUD

Hoe laden we machinecode in de ZX-80/81 01.6.1.001 01-47



ZX81 UITBREIDINGEN

— in landem te gebruiken —

MEMOPAK 16K 1178,-

MEMOPAK 32K 1278,-

MEMOPAK 64K 1418,-

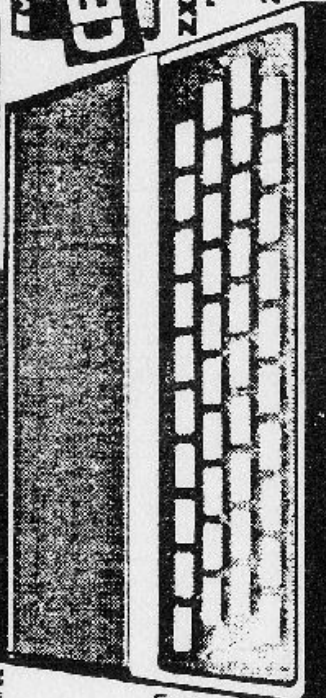
MEMOPAK 32K RAM

Operates in tandem
with Sinclair 16K Ram
to give a full 48K!

MEMOPAK 64K 1418,-

MEMOPAK PRO

PRUIZEN PER 1/12/82



Wij leveren
nog steeds
**DE ZX81
voor 1298,-
COMPLEET!**
Prijswijzigingen
voorbehouden

Hoe te bestellen? door vooruitbetaling op giro 13 61 173 van DATALINK.
Of op bankrekening 43 51 97 010 ten name van DATALINK HEINO.
Of onder rembours (15,- extra). Alle prijzen zijn inclusief BTW. Tel daar nog
wel de portokosten bij op: 18,50.

Telefonische orders kunnen wij helaas niet aannemen.

DATALINK POSTBUS 61, 8140 AB HEINO (OV.)

DEALERS GEZOCHT

MEMOTECH

IN NEDERLAND: VRAAG UW ZX-DEALER OF BIJ:
DATALINK HEINO (Overijssel)

ZX81

MEMOPAK RAM
16-KB GRAPHICS
CENTRONICS I/F

ZX81

A complete range of ZX81 plug-in peripherals

CENTRONICS I/F 1228,-

**ZX SOFTWARE
'AACKOSOFT'**

Postbus 3111

2301 CH Leiden

**HIGH RES.
GRAPHICS**

1328,-

oplossend vermogen
192x248 pixels.

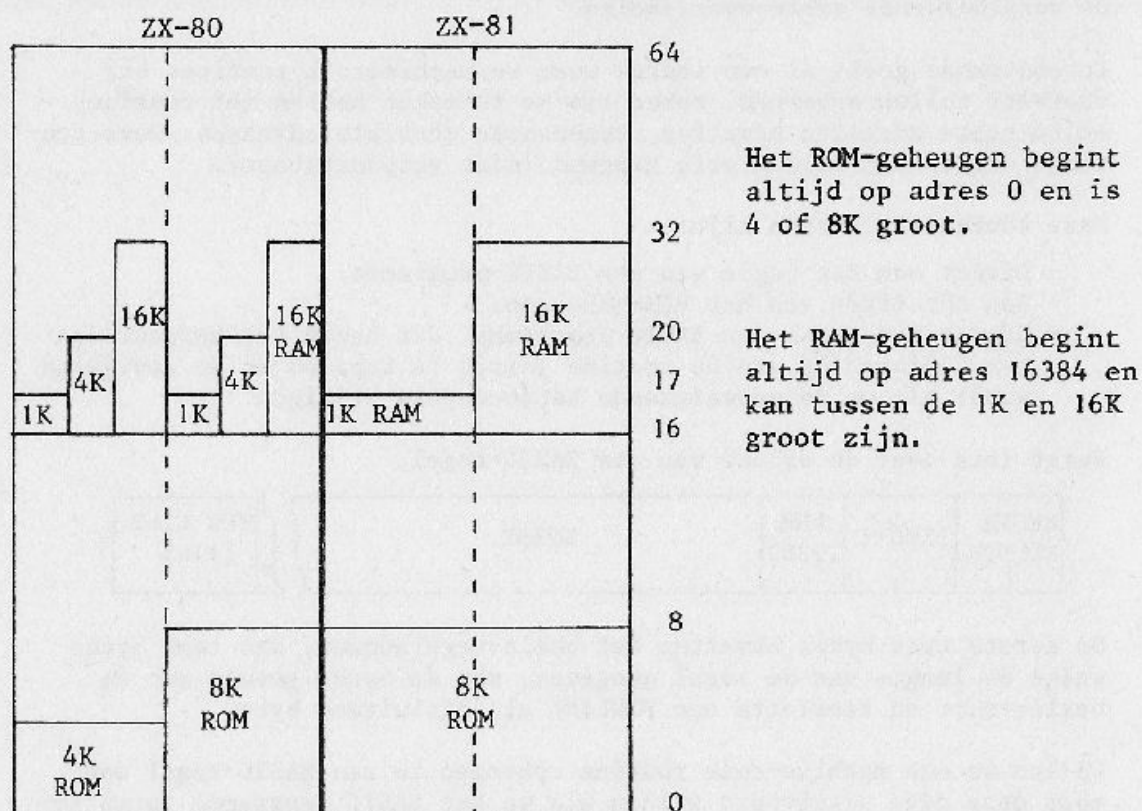
**plug-in toetsenbord
prijs ca. 228,-**

NIUW

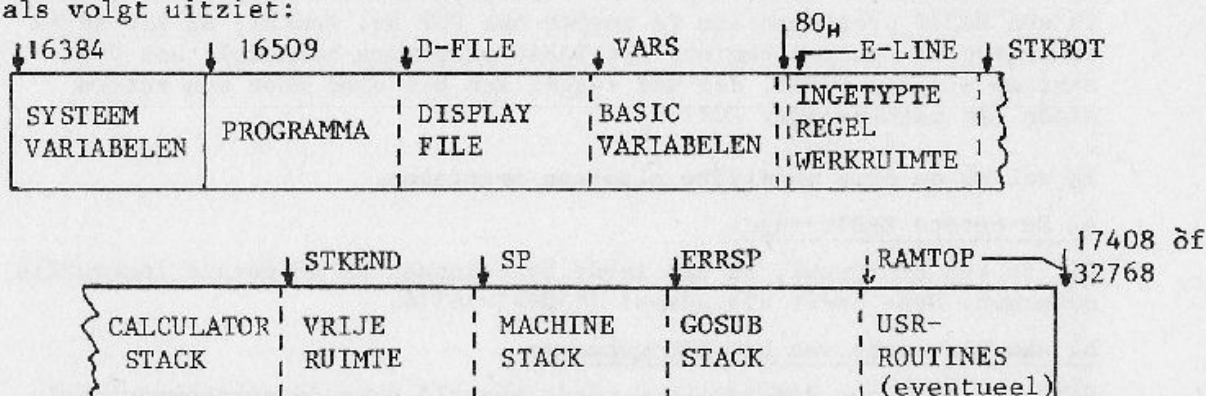
HOE LADEN WE MACHINECODE IN DE ZX-80/81

Voor we het eigenlijke probleem, het laden van machinecode in de ZX-80/81, bekijken eerst wat theorie over de werking van de Sinclair.

De micro-processor kan een adres-bereik van 0-65535 direct adresseren. Dit adressenbestand is als volgt ingedeeld.



Binnen het beschikbare RAM-geheugen is er een vaste indeling welke er als volgt uit ziet:



Aangezien de totale lengte afhangt van het gebruikte RAM-geheugen zijn er maar 3 grenzen welke eenduidig vastliggen.

- Het begin nl. 16384.
- Het begin van de programma-ruimte 16509, het aantal systeemvariabelen is namelijk constant.
- Het einde van de beschikbare RAM-ruimte. Dit wordt bij het inschakelen door de machine zelf bepaald en vastgelegd in de

systeemvariabele: RAMTOP (deze bevat de waarde van het eerste niet bestaande byte).

Alle andere grenzen kunnen tijdens het gebruik verschuiven. Door bijvoorbeeld het programma uit te breiden zal ook de DISPLAY-FILE opschuiven, enz. Uiteindelijk wordt er van de VRIJE RUIMTE wat afgeknabbeld.

De grenzen worden voortdurend aangepast en de waarden opgeslagen in de verschillende systeemvariabelen.

Bovenstaande geeft al een indruk waar we machinecode routines bij voorkeur zullen opbergen, zeker als we te maken hebben met routines welke echte adressen bevatten, zogenaamde absolute adressen. Deze routines worden non-relocatable genoemd (niet verplaatsbaar).

Deze voorkeursplaatsen zijn:

- Direct aan het begin van een BASIC-programma.
- Aan het einde van het RAM-geheugen.
- Ergens midden-in een BASIC programma, dit heeft tot bezwaar dat het begin-adres van de routine lastig te bepalen is en gewijzigd wordt als we de voorafgaande BASIC-regels wijzigen.

Eerst iets over de opbouw van een BASIC-regel.

REGEL NUMMER	LENGTE	REM (234)	TEKST	NEW LINE (118)
-----------------	--------	--------------	-------	-------------------

De eerste twee bytes bevatten het basic-regelnummer, dan twee bytes welke de lengte van de regel aangeven, dan de bytes gevuld met de basic-tekst en tenslotte een NEWLINE als afsluitend byte.

Willen we een machine-code routine opbergen in een BASIC-regel dan moet deze niet uitgevoerd worden als we het BASIC programma laten lopen, hetgeen betekent dat het een REM-regel wordt van voldoende lengte om de Machinecode te bevatten. Machinecode routines starten we door ze in een BASIC programma aan te roepen met `USR nn`, waarbij `nn` het adres is waarop de routine begint. Het BASIC programma behandelt een `USR` aanroep als een `GOSUB`, dat wil zeggen aan het eind moet een return staan (in machinecode: 201).

Ik zal nu de drie mogelijke plaatsen bespreken.

a. De eerste BASIC-regel

Dit is een REM-regel, in het zesde byte kunnen we de eerste instructie opbergen. Deze heeft als adres: $16509+5=16514$.

b. Aan het einde van het RAM-geheugen

Het einde van het RAM-geheugen wordt bepaald door de systeemvariabele RAMTOP, deze wordt opgeborgen op de adressen: 16388 en 16389.

Stel dat RAMTOP de waarde heeft van 17408 (een 1K RAM) dan bevat adres 16389 de waarde: $\text{INT}(17408/256)=68$ en adres: 16388: $16388-256*\text{INT}(17408/256)=0$.

Als we nu in de adressen 16388 en 16389 een kleinere waarde 'POKE'n gevolgt door het commando NEW, dan reserveren we een ruimte die niet

overschreven wordt door NEW of LOAD maar welke ook niet geSAVED wordt. Deze waarde(v) brengen we als volgt in:

```
POKE 16388, v-256*INT(v/256)
POKE 16389, INT(v/256)
NEW.
```

c. In een BASIC-regel ergens in het programma

Het bezwaar hiervan is dat we het begin-adres van onze USR-routine niet weten. Gelukkig is er een systeemvariabele NXTLIN genaamd welke het begin-adres van de BASIC-regel bevat welke volgt op die welke in uitvoering is. NXTLIN staat in de adressen: 16425 en 16426. Als we nu in de aanroep van de USR-routine, die direct voor de regel moet staan waarin de machinecode staat, het begin-adres bepalen door gebruik te maken van NXTLIN gaat alles zoals het moet gaan.

We weten nu waar we de machine-codes kunnen opbergen. Het hoe volgt nu. Hiervoor gebruiken we meestal een BASIC hulpprogramma, dit kan na gebruik weer verwijderd worden. Ik zal twee manieren geven om met behulp van een hulpprogramma machine-code te laden.

- a. De machinecode wordt in een string gezet en van daaruit door het programma over ge-POKED.
- b. Het programma vraagt steeds om de invoer van een M.C.-instructie of in decimale of in hexadecimale vorm en zet dat direct weg door middel van een POKE.

Programma 1 zet een M.C.-routine in een REM-statement aan het begin van een BASIC-programma. Het vraagt steeds om invoer in decimale vorm.

```
1 REM 1234567890123456789012345
6789012
2 LET S=16513
3 FOR I=0 TO 31
4 SCROLL
5 LET B=16514+I
6 PRINT I;"=";B;"-->";
7 INPUT A
8 IF A<0 THEN GOTO 12
9 POKE S+I,A
10 PRINT A
11 NEXT I
12 STOP
```

Toelichting: De machine-code is 32 bytes lang. In de REM-statement worden dus een 32 bytes lange tekst gemaakt.
De FOR-NEXT-Loop vraagt 32 maal om invoer en laat zien het hoeveelste byte op welk adres wordt weggeschreven en de waarde daarvan. Als een negatief getal wordt ingetypt stopt de invoer-vraag.

Na RUN worden de volgende waarden ingevoerd:

17	125	64	33	100	0	26	61
254	117	192	19	26	154	39	208
1	10	0	9	235	114	35	115
35	78	35	70	9	235	24	230

SINCLAIR IMPULS

01.6.1.004

Deze routine hernummert de basic-regels van een programma (GOTO's en GOSUB's worden niet aangepast).

De onderstreepte adressen bevatten de volgende waarden:

- 1: Beginadres van waaraf hernummerd wordt ($125+64*256=16509$).
- 2: Beginwaarde nieuwe BASIC-regelnummers ($100+0*256=100$).
- 3: Toename basic-regelnummers (increment) ($10+0*256=10$).

De routine wordt aangeroepen met: PRINT USR 16514.

Programma 2 zet een M.C.-routine achter RAMTOP. Het laad de machine-code uit een string, in hexadecimale vorm. De routine is 13 bytes lang.

Verplaats eerst RAMTOP met 13 bytes dus op adres: $17408-13=17395$ (voor een 16K geheugen 32768 in plaats van 17408).

```
POKE 16388, 17395-256*INT(17395/256)
POKE 16389, INT(17395/256)
NEW.
```

Type nu het volgende BASIC-hulpprogramma in:

```
100 LET A$="B7ED5B1C40ED6239ED5
2E5C1C9"
110 FOR I=1 to 13
120 POKE 17394+I, 16*CODE A$+CO
DE A$(2)-476*
130 LET A$=A$(2TO)
140 NEXT I
```

*De waarde: 476 is verkregen omdat de CODE van een getal in de ZX 28 groter is dan de waarde dus moeten we $26*28+16=476$ van het resultaat aftrekken.

RUN het programma en verwijder daarna het programma.

Deze routine kunt U aanroepen met:

```
PRINT USR 17395. U krijgt dan op Uw scherm het aantal BYTES dat
nog vrij is in Uw geheugen.
```

Programma 3 geeft de mogelijkheid een machine-code routine middenin een BASIC programma te zetten. De enige beperkende voorwaarde is dat de aanroep (USR-statement) er onmiddellijk voor moet staan.

```
370 LET A=(PEEK 16425+256*PEEK
16426)+5
371 REM 123456789012345678901
380 FOR N=A TO A+21
381 INPUT B
382 POKE N,B
383 PRINT PEEK N
384 NEXT N.
```

Nadat we dit hebben ingetypt geven we het commando: GOTO 370.
Er wordt dan om de volgende decimale invoer gevraagd:

175	103	111	57	237	75	28	64	237	66
229	193	253	33	0	64	62	30	237	71
201									

Verwijder nu regel 380 t/m 384. Verander regel 370 in:

370 PRINT USR(PEEK 16425+256*PEEK 16426)+5.

Type nu de rest van het programma in.

Deze routine geeft bij aanroep de vrije geheugenruimte.

Ik hoop dat het maken van USR-routines, en het plaatsen daarvan in BASIC programma's na de 3 voorbeelden iets duidelijker is geworden.

E.R. Visser.

— o — o —

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      S I N C L A I R - B I J E E N K O M S T E N      %
%                                                                 %
%              1 9 8 3                                     %
%                                                                 %
%    15 januari                                     5 maart %
%    16 april                                       11 juni  %
%    3 september                                   15 oktober %
%              10 december                             %
%    Aanvang: 11 uur.                                %
%    Technische school "De Bron"                     %
%    Vadderijndreef 7, Utrecht.                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```