

# ZX USER CLUB SAMMELBAND

Nr.1/E4.

DM 9,80



Über 65 Programme  
für den SINCLAIR  
ZX81 und  
ZX-SPECTRUM  
Tips & Tricks  
Hardware-Erweiterungen



# SOFTWARE-POWER FÜR SINCLAIR SPECTRUM UND ZX81 VON HUEBER SOFTWARE



**Hueber Software**



**Tim Hartnell**  
**49 explosive Spiele**  
für den Sinclair ZX81  
In diesem Buch finden Sie weltberühmte Spiele, die Sie stundenlang an Ihren ZX81 fesseln werden.  
ISBN 3-19-008204-9, DM 29,80



**Trevor Toms**  
**Das ZX81 Buch**  
Programmierer in Maschinensprache — Optimierung — Viele Tips & Tricks für die Programmierung des ZX81.  
ISBN 3-19-008203-0, DM 29,80



**Alastair Gourlay**  
**34 TK-Superspiele**  
für den Sinclair ZX81  
Unglaublich was Courlay an Spielen und Programmen mit TK-RAM präsentiert.  
ISBN 3-19-008202-2, DM 19,80



**Tim Hartnell**  
**Entdecken Sie die unendlichen Dimensionen Ihres ZX81**  
Die Lesemuß für jeden ZX81-Anwender. Viele Beschreibungen.  
ISBN 3-19-008205-7, DM 29,80



**David Harwood**  
**SPASS & PROFIT SPECTRUM**  
60 Spiele und nützliche Anwendungen für das unendliche Spectrum Ihres ZX-SPECTRUM.  
ISBN 3-19-008201-4, DM 24,80



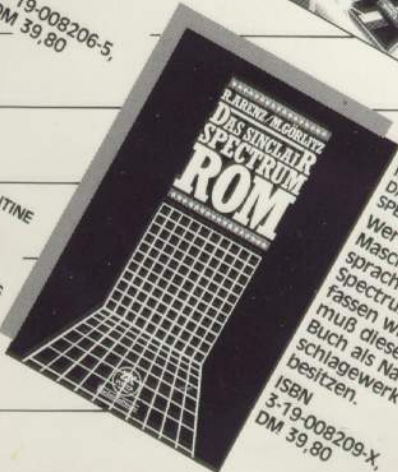
**H. Brandl / S. Sanver**  
**ROM**  
Das unentbehrliche Nachschlagewerk für ZX-Besitzer: komplettes do-Listing des ZX81.  
ISBN 3-19-008206-5, DM 39,80



**Hartnell / Jones**  
**SPECTRUM OHNE GRENZEN**  
Über 100 Programme und Routinen, die alle garantiert für den Spectrum geschrieben.  
ISBN 3-19-008208-1, DM 29,80



**Roger Valentine**  
**50 Programme für den ZX SPECTRUM**  
Wenn Sie alle Qualitäten Ihres Computers voll wollen — brauchen Sie dieses Buch.  
ISBN 3-19-008200-6, DM 29,80



**R. Arenz / M. Gohlitz**  
**DAS SINCLAIR SPECTRUM ROM**  
Wer sich mit Maschinensprache im Spectrum befassen will, muß dieses Buch als Nachschlagewerk besitzen.  
ISBN 3-19-008209-X, DM 39,80



**Trevor Toms**  
**DAS SPECTRUM BUCH**  
Spaß-, Spiel- und Nutzprogramme in BASIC, Maschinencode, Assembler, Disassembler und vieles mehr.  
ISBN 3-19-008207-3, DM 29,80

**Bestellen Sie im Innenteil auf Seite 60**



## Inhalt ZX-USER-CLUB Sammelband 1983

### PROGRAMME FÜR DEN ZX81

- 4 Tastaturabfrage  
Todeslabyrinth
- 5 Bombe  
Fangen
- 6 Schiff suchen  
Fakultätsberechnung  
Woche für Woche über  
13 Millionen Möglichkeiten
- 7 Siebzehn und vier
- 8 Endlosschlange
- 10 Gratwanderung  
Simon  
Mauer einwerfen
- 12 MC-Loader
- 14 Schachuhr
- 16 Eheinstitut  
Vergißmeinnicht
- 18 Monaco
- 20 Othello
- 22 Wer zuletzt lacht,  
lacht am besten  
Zahlenspiel  
Sternschlag im Weltraum
- 24 Biorhythmus
- 26 Adressverwaltung
- 27 Gehaltsabrechnung
- 28 Jetzt wird  
Groß-geschrieben
- 29 BASIC-Renumber-  
Programm
- 30 Scrollen ist wirklich  
ein Freizeitvergnügen
- 31 Schleifen  
Umwandlung in Hexadezimal
- 32 Pferderennen  
Boliden
- 34 Black Jack  
Kamikaze  
Reaktionstest
- 35 Albrecht Dürer  
Bildschirmvertierer
- 36 Australische Landung  
Turm von Babel  
Bowling
- 37 Lineare Regressions- und  
Korrelationsanalyse
- 38 Balkendiagramm
- 39 Umrechnung  
Der Spion, der aus der Kälte kam  
Astronautentests

- 40 Das Coingame
- 41 Hunde, wollt Ihr  
ewig leben
- 42 Frogger wie Frosch
- 46 SHIFT

### PROGRAMME FÜR DEN ZX-SPECTRUM

- 47 Muster

### PROGRAMME FÜR DEN ZX81

- 47 Die Sensation—  
Ein 1-K Maschinenprogramm
- 48 Hexadezimal-  
Umwandlung
- 49 Vier gewinnt

### TIPS & TRICKS

- 50 Stop  
Fließkomma  
Unendlich lange Pause  
JOJO oder die optische  
Täuschung des ZX81
- 52 Einführung in die  
Maschinensprache

### PROGRAMME FÜR DEN ZX81

- 58 Knobel mal wieder  
Berechnung des Ostersonntags  
bis zum Jahre 1991  
Banko
- 59 Data für ZX81
- 60 Mini-Packman

### PROGRAMME FÜR DEN ZX-SPECTRUM

- 61 Asteroids

### STORY

- 62 Rund um den  
SPECTRUM

### PROGRAMME FÜR DEN ZX-SPECTRUM

- 63 Festung

- 66 Scroll in alle Richtungen  
Für 16K und 48K-SPECTRUM

- 67 Farbig durch den Winter

- 68 Wer anderen eine Grube gräbt,  
fällt selbst hinein  
Sprachausgabe für den  
ZX-SPECTRUM

- 69 Picasso  
Dezimal-, Hexadezimalwandlung  
für den ZX-SPECTRUM  
Spectral Henker

- 70 Plotten und Zeichnen  
Trailer

- 71 READER 1  
READER 2

- 72 Sortieren mit dem  
ZX-SPECTRUM

- 74 Mondlandung

### HARDWARE

- 77 SPECTRUM  
MICRODRIVE

- 81 Akustische Eingabekontrolle  
für den ZX81  
RESET-Taste

### SELF-HARDWARE-MAN

- 82 Aufnahmeverstärker  
Änderung des  
Super-Moving-Keybord

Impressum:  
ZX USER CLUB  
Sammelband 1983  
Cooperation GmbH, Bruderstraße 2, 8000 München 22

Herausgeber: Josef Schaaf  
Word-Processor: Joe M. Hembus  
Chefredakteur: Wolfgang Bergmann  
Redaktion Technik: R. Brandl, S. Sanver  
Satz: Fotosatz Kretschmann, Bad Aibling  
Druck: Himmer, Augsburg

ZX-USER-CLUB-Sammelband 1983, Preis DM 9,80

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Die geltenden gesetzlichen und postalischen Bestimmungen bei Erwerb, Errichtung und Inbetriebnahme von Sende- und Empfangseinrichtungen sind zu beachten.

Die gewerbliche Nutzung, insbesondere der Schaltpläne und Programme, ist nur mit schriftlicher Genehmigung des Herausgebers zulässig. Die Zustimmung kann an Bedingungen geknüpft sein. Sämtliche Veröffentlichungen in ZX-USER-CLUB erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen ohne Gewährleistung einer freien Verwendung benutzt.  
Printed in Germany.



# PROGRAMME

## Tastaturabfrage



Das folgende Programm fragt die Tastatur ab. Sie wartet so lange, bis eine Taste gedrückt wird.

Hier das Listing in Z80-Mnemonics:

```
4082 C0BB02 CALL 02BB
4085 7D LD A,L
4086 FEFF CP FF
4088 28F8 JR Z,4082
408A 44 LD B,H
408B 4D LD C,L
408C C0BD07 CALL 07BD
408F 7E LD A,(HL)
4090 D7 RST 10H
4091 C9 RET
```

```
16514 205
16515 187
16516 2
16517 125
16518 254
16519 255
16520 40
16521 248
16522 68
16523 77
16524 205
16525 189
16526 7
16527 126
16528 215
16529 201
```

Dieses Programm ist also zuerst einmal die Grundlage unserer Tastaturabfrage.

Nun wollen wir die Vorgänge beim Ablauf dieses Programmes klären. Zuerst wird mit CALL 02BB die Routine im ROM aufgerufen, die die Tastatur abfragt.

Wenn eine Taste gedrückt wurde, dann ist im HL-Register der zugehörige Wert; ansonsten ist das HL-Register mit FF geladen. Mit den nächsten Befehlen:

LD A,L, CP FF und JR Z.4082 wird wieder zum Anfang des Programmes zurückgesprungen, wenn keine Taste gedrückt wurde. Wenn also nun eine Taste gedrückt wurde, dann wird BC mit HL geladen und mit CALL 07BD die Dekodierungsroutine im ROM aufgerufen.

Sie lädt das HL-Register mit der Adresse, in der der Code der gedrückten Taste steht. Nun wird der Akkumulator mit LD A. (HL) mit dem Code geladen. Nun kann man das Programm nach den eigenen Bedürfnissen fortsetzen. Hier seien nur 2 Beispiele erwähnt:

### 1. SCHREIBMASCHINE

Erläuterungen:

RST 16 ist die Druckroutine, die den Code ausdrückt, der sich gerade im Accumulator befindet. Der Rest des Programmes fragt wieder ab, ob eine Taste gedrückt wurde und wartet ab, bis keine Taste mehr gedrückt wird.

```
4090 D7 RST 10H
4091 C0BB02 CALL 02BB
4094 7D LD A,L
4095 FEFF CP FF
4097 28F8 JR NZ,4091
4099 C9 RET
```

```
16528 215
16529 205
16530 187
16531 2
16532 125
16533 254
16534 255
16535 32
16536 248
16537 38
16538 201
```

### ACHTUNG!!!

Beachten Sie, daß nur die Tasten gedrückt werden dürfen, deren Code einstellig ist!

Beispiel: 'A,B,C,.,?,-,+ , etc.

NICHT: «=, THEN, AND, \*\*, ect.

### 2. ABFRAGE EINER BESTIMMTEN TASTE

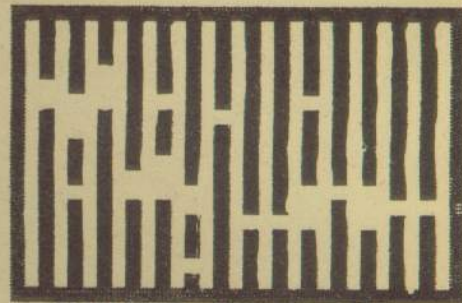
Erläuterungen:

Zuerst muß der abzufragende Code mit POKE 16529,XX eingepoked werden (wenn das Programm an einer anderen Stelle im Speicher steht, dann ist die entsprechende Adresse zu ermitteln).

```
4090 FE00 CP 00
4092 C8 RET Z
4093 18ED JR 4082
```

```
16528 254
16529 0
16530 200
16531 24
16532 237
16533 254
```

## Todeslabyrinth



SPIELEND PUNKTE: 4859  
BISHERIGER BESTAND: 5109

Wie Sie auf dem Computerausdruck sehen, erzeugt der ZX81 ein Labyrinth. Sie (das S-Zeichen) starten auf der linken Seite und sollten möglichst schnell nach rechts gelangen, indem Sie Q (hoch), Z (tief) und L (rechts) drücken. Die Punktzahl nimmt dabei ständig ab. Wenn Sie gegen einen schwarzen Strich stoßen (statt durch die Leerräume zu gehen), werden Ihre Punkte erheblich weniger. Wenn Sie an eine Außen-Umrandung stoßen (ausgenommen die rechte), endet das Spiel automatisch.

Das Programm gibt den jeweiligen Punkt-Rekord an. Mit einiger Übung sollten Sie über 13504 Punkte erreichen. Wenn Sie zusehen wollen, wie der ZX81 zwischen den einzelnen Runden das Labyrinth aufbaut, löschen Sie die Zeilen 10 FAST und 9490 SLOW.

```
3 LET U=0
10 FAST
15 FOR B=2 TO 28 STEP 2
20 FOR A=0 TO 19
30 PRINT AT A,B;" "
40 NEXT A
45 PRINT AT RND*14+3,B;" "
50 NEXT B
60 FOR A=0 TO 30
70 PRINT AT 0,A;" " AT 19,A;" "
80 NEXT A
90 FOR B=1 TO 18
100 PRINT AT B,0;" " AT B,30;" "
110 NEXT B
120 GOSUB 9000
125 GOTO 167
150 PRINT AT A,B;" "
155 LET Z=Z-673
160 PRINT AT 20,0;"PUNKTE: ";Z
165 RETURN
167 LET Z#=INKEY#
170 IF Z#="" THEN LET Z#=A#
180 LET Z=Z-50
200 LET Y=A
```

K. Reger, München

# ZX-USER CLUB



# PROGRAMME

```

210 LET X=B
215 REM ZUR SPIELBESCHLEUNIGUNG
NACHSTE ZEILE LOESCHEN
217 LET R=RND*RND*RND
220 LET A=A+(Z#="Z")-(Z#="Q")
230 LET B=B+(Z#="L")
232 PRINT AT Y,X;" "
233 IF PEEK (PEEK 16396+256*PEE
K 16397+33*A+B+1)=128 THEN GOSUB
150
235 PRINT AT A,B;"#"
255 IF A>18 OR A<2 OR B<1 THEN
LET Z=INT (Z/3)
260 IF A>18 OR A<2 OR B<1 OR B>
29 THEN GOTO 510
490 LET A#=Z#
500 GOTO 167
510 PRINT AT 20,0;"SIELENDE P
UNKTE: ";Z
520 IF Z>U THEN LET U=Z
530 FOR G=1 TO 6
540 PRINT AT 21,3;"BISHERIGER R
EKORD: ";U
545 PRINT AT A,B;"#";AT A,B;"#"
;AT A,B;"#"
550 PRINT AT 21,14;"REKORD"
560 NEXT G
570 CLS
580 GOTO 10
9000 LET A=10
9010 LET B=1
9020 LET Z=20000
9030 LET Y=A
9040 LET X=B
9050 LET A#="Z"
9490 SLOW
9500 RETURN

```

Aus Tim Hartnells »49 explosive Spiele«, erschienen im Verlag Cooperation, München

## Bombe

Sie suchen auf einer durchgehenden schwarzen Fläche (einem 9x9) Raster von CHR\$(128) nach einer Bombe, wobei Sie nur mit Ihrer Tastatur bewaffnet sind.

Ein »Bombendetektor« — eine unter dem großen Quadrat erscheinende Zahl — gibt Ihnen Hinweise, wo Sie sich relativ zur Bombe befinden. Sie bewegen sich durch Drücken der Taste 5, 6, 7 und 8, die auf der jeweiligen Taste angegebene Pfeilrichtung.

Wenn Sie die Bombe finden, erscheint ein inverses »B« an der Stelle, wo sie versteckt war. Dieses 1K-Programm teilt Ihnen auch mit, wie lang Sie für die Suche gebraucht haben.

```

10 LET S=0
20 LET A=1+INT (RND*8)
30 LET B=1+INT (RND*9)
40 LET K=16396
50 LET Q=128
60 FOR Z=1 TO 10
70 PRINT ....
80 NEXT Z
90 FOR X=2 TO 10
100 FOR Y=1 TO 10

```

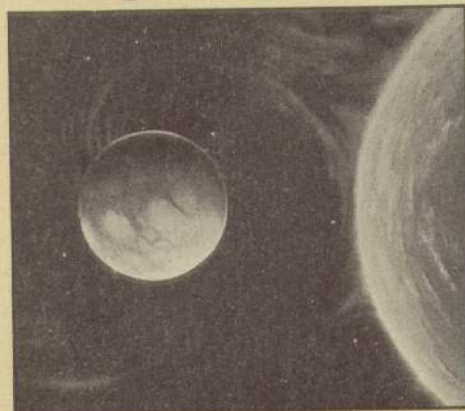
```

110 POKE Y*33+X+PEEK (K)+PEEK (
K+1)*256,Q
120 NEXT Y
130 NEXT X
140 LET C=1
150 LET D=1
160 POKE C*33+D+1+PEEK (K)+PEEK
(K+1)*256,189
170 LET S=S+1
180 IF INKEY#="" THEN GOTO 180
190 POKE C*33+D+1+PEEK (K)+PEEK
(K+1)*256,Q
200 IF INKEY#="7" THEN LET C=C-
1
210 IF INKEY#="5" THEN LET D=D-
1
220 IF INKEY#="6" THEN LET C=C+
1
230 IF INKEY#="8" THEN LET D=D+
1
240 IF C<1 THEN LET C=1
250 IF C>9 THEN LET C=9
260 IF D<1 THEN LET D=1
270 IF D>9 THEN LET D=9
280 IF A=C AND B=D THEN GOTO 31
0
290 POKE 187+PEEK (K)+PEEK (K+1
)*256,ABS (B-D)+ABS (A-C)+156
300 GOTO 160
310 PRINT "ERFOLG NACH ";C;" ";
D;" IN ";S;" VERSUCHEN"
320 POKE C*33+D+1+PEEK (K)+PEEK
(K+1)*256,167

```

Aus Tim Hartnells »49 explosive Spiele«, erschienen im Verlag Cooperation, München

## Fangen



Das Spiel beginnt mit einem rollenden Ball, den Sie mit den Tasten »6« und »7« so lenken sollen, daß er möglichst viele schwarze Kästchen berührt. Das Programm gibt den jeweiligen Punktrekord an — da jedoch die Schwierigkeit während des Spiels ansteigt, wird es immer schwerer, die Punktzahl zu erhöhen. Wenn Sie keine Lust mehr haben, das Spiel nach diesen Regeln zu spielen, versuchen Sie, möglichst vielen schwarzen Kästchen auszuweichen.

```

5 LET U=0
10 FOR Z=1 TO 60

```

```

20 PRINT AT 3+RND*16,2+RND*29;
" "
30 NEXT Z
40 LET E=16396
50 LET F=16397
60 LET T=0
70 LET M=0
100 LET X=10
102 LET K=1
105 LET B=X
107 LET B#="0"
110 LET Y=10
111 LET A=X
117 POKE 33*B+A+1+PEEK E+256*PE
EK F,0
118 IF PEEK (33*Y+X+1+PEEK E+25
6*PEEK F)=128 THEN GOSUB 500
120 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
121 LET T=T+3
122 LET A=X
123 IF T>200 THEN GOSUB 500
124 LET B=Y
130 LET X=X+K
135 IF X<2 OR X>30 THEN LET K=-
K
136 LET A#=INKEY#
137 IF A#="" THEN LET A#=B#
138 LET Y=Y-(A#="7")+(A#="6")
139 IF Y<2 THEN LET A#="6"
140 IF Y>18 THEN LET A#="7"
150 LET B#=A#
170 GOTO 117
500 LET M=M+1
502 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
505 PRINT AT 0,0;"IHRE PUNKTZAHL
IST ";M;AT 1,5;"ZEIT: ";T
513 FOR Z=1 TO 7
515 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
516 POKE 33*Y+X+1+PEEK E+256*PE
EK F,23
518 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
520 NEXT Z
522 LET R=117*M
525 PRINT AT 20,0;"GESAMTPUNKTE
";R
527 FOR H=1 TO 6
528 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
529 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
530 NEXT H
535 PRINT AT 20,0;"
"
540 IF T>200 THEN GOSUB 1000
560 PRINT AT 0,0;"
"
570 RETURN
1000 IF R>U THEN LET U=R
1010 PRINT AT 0,0;"SIELENDE - G
ESAMTPUNKTE: - ";R;AT 1,5;"REKOR
D - ";U
1020 FOR H=1 TO 23
1022 POKE 33*Y+X+1+PEEK E+256*PE
EK F,189
1025 POKE 33*Y+X+1+PEEK E+256*PE
EK F,52
1030 NEXT H
1035 LET M=0
1040 LET T=0
1045 LET R=0
1050 RETURN

```

# ZX-USER CLUB



# PROGRAMME

## Schiff suchen

(Speicherbedarf 923 Bytes)

Mit dem folgenden kleinen Spielprogramm wird ein 8 mal 8 Felder umfassendes Gebiet nach einem dort verborgenen, 2-Felder großen U-Boot, abgesehen.

Wer dafür die wenigsten Versuche benötigt, hat das Spiel gewonnen.

Durch RUN wird das Spiel gestartet, worauf sich der Bildschirm für etwa 20 Sekunden verdunkelt. In dieser Zeit wird das Spielfeld gezeichnet und das U-Boot in Position gebracht.

Anschließend erwartet der ZX81 die Eingabe der Suchkoordinaten, und zwar Zeilen-Nr, N/L — Spalten-Nr, N/L. Einen Fehlschuß quittiert der Computer durch eine Null an der angegebenen Stelle, bei einem Treffer wird ein schwarzes Quadrat ausgegeben.

Bei Spielende wird die Anzahl der Versuche angezeigt.

K. Bruger, Augsburg

```

1 REM SCHIFF SUCHEN BY K.BRUGER
2 FAST
10 FOR A=1 TO 8
20 FOR B=1 TO 8
30 PRINT AT A,B;" "
40 PRINT AT 0,B;"B"
50 PRINT AT A,0;"A"
60 NEXT B
70 NEXT A
80 LET Y=0
90 LET Z=0
100 LET A=INT (RND*8)+1
105 LET B=INT (RND*8)+1
110 LET X=RND
115 SLOW
117 PRINT AT 20,0;"ZAHL ZW. 1 U
    "
120 INPUT K
125 INPUT K1
130 PRINT AT K,K1;" "
135 LET Y=Y+1
140 IF X<.5 THEN GOTO 200
145 IF A=K AND B=K1 OR K=A AND
    B=K1+1 THEN GOTO 300
150 GOTO 120
200 IF A=K AND B=K1 OR K=A+1 AN
    D B=K1 THEN GOTO 300
210 GOTO 120
300 PRINT AT K,K1;"■"
310 LET Z=Z+1
320 IF Z>1 THEN GOTO 350
330 GOTO 120
350 PRINT AT 10,8;"BEIM ";Y;"
    VERSUCH"
```

## Fakultätsberechnung mit ZX81

Als Fakultät einer Zahl bezeichnet man das Produkt der Zahl und aller positiven ganzen Zahlen, die kleiner als sie sind.

So ist etwa die Fakultät von 6 (man schreibt »6!«)  $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ . Diese Funktion wird unter anderem in Formeln der Wahrscheinlichkeitsrechnung benutzt. Während wissenschaftliche Taschenrechner heutzutage meist über eine entsprechende Funktion verfügen, ist dies bei Computern nur sehr selten zu finden.

Mit diesem Programm für den Sinclair ZX81 lassen sich auch Fakultäten berechnen, die normalerweise den Rechenbereich überschreiten würden, z. B. die Fakultät von 111. Das hängt mit der Berechnungsmethode zusammen: Die Logarithmen der Faktoren werden addiert.

Bedienung: Nach Start gewünschte Zahl eingeben, kurz danach erscheint die Antwort.

Variablen:

F Logarithmus des Ergebnisses

N Eingabe

X Schleife

O. Völkers, Osnabrück

```

10 PRINT "FAKULTAET"
20 PRINT
30 PRINT "VON "
40 INPUT N
50 PRINT N;"="
60 LET F=0
70 FOR X=1 TO N
80 LET F=F+LN X
90 NEXT X
100 LET F=F/LN 10
110 IF F<37 THEN PRINT INT (10*
    *F+.5)
120 IF F>37 THEN PRINT 10**(F-I
    NT F);"E+";INT F
130 GOTO 20
FAKULTAET
VON 0= 1
VON 1= 1
VON 3= 6
VON 5= 120
VON 10= 3628800
VON 17= 3.5568744E+14
VON 23= 2.5852017E+22
VON 69= 1.7112246E+98
VON 111= 1.7629523E+180
VON 176= 1.9790318E+320
```

## Woche für Woche über 13 Millionen Möglichkeiten



Hier stellen wir jedem ZX-Lottospieler und werdende Millionäre das unglaublich kurze Lottoprogramm unseres Lesers Dr. R. Goldammer vor. Das erste Programm wirft Ihnen 6 Zahlen sortiert nacheinander auf den Bildschirm. Das weitere Programm errechnet Ihnen die Möglichkeiten, wann Sie endlich Millionär werden — es handelt sich nur um die Kleinigkeit von 13 Mio. und mehr ...

```

10 REM LOTTO
20 DIM B(50)
30 FOR I=1 TO 6
40 LET Z=INT (49*RND*1+1)
50 IF B(Z)=1 THEN GOTO 40
60 LET B(Z)=1
70 NEXT I
80 FOR I=1 TO 49
90 IF B(I)=1 THEN PRINT I
100 NEXT I
```

```

10 REM M UEBER N
20 PRINT "OBERE ZAHL";
30 INPUT M
35 PRINT "=";M
40 PRINT "UNTERE ZAHL";
50 INPUT N
55 PRINT "=";N
60 LET G=1
70 FOR K=1 TO N
80 LET G=G*(M-K+1)
90 NEXT K
100 PRINT "ZAEHLER=";G
110 LET L=1
120 FOR K=1 TO N
130 LET L=L*K
140 NEXT K
150 PRINT "NENNER=";L
160 LET H=1
170 LET H=H*G/L
180 PRINT
190 PRINT "M UEBER N=";H
200 PAUSE 4E4
210 CLS
220 RUN
```

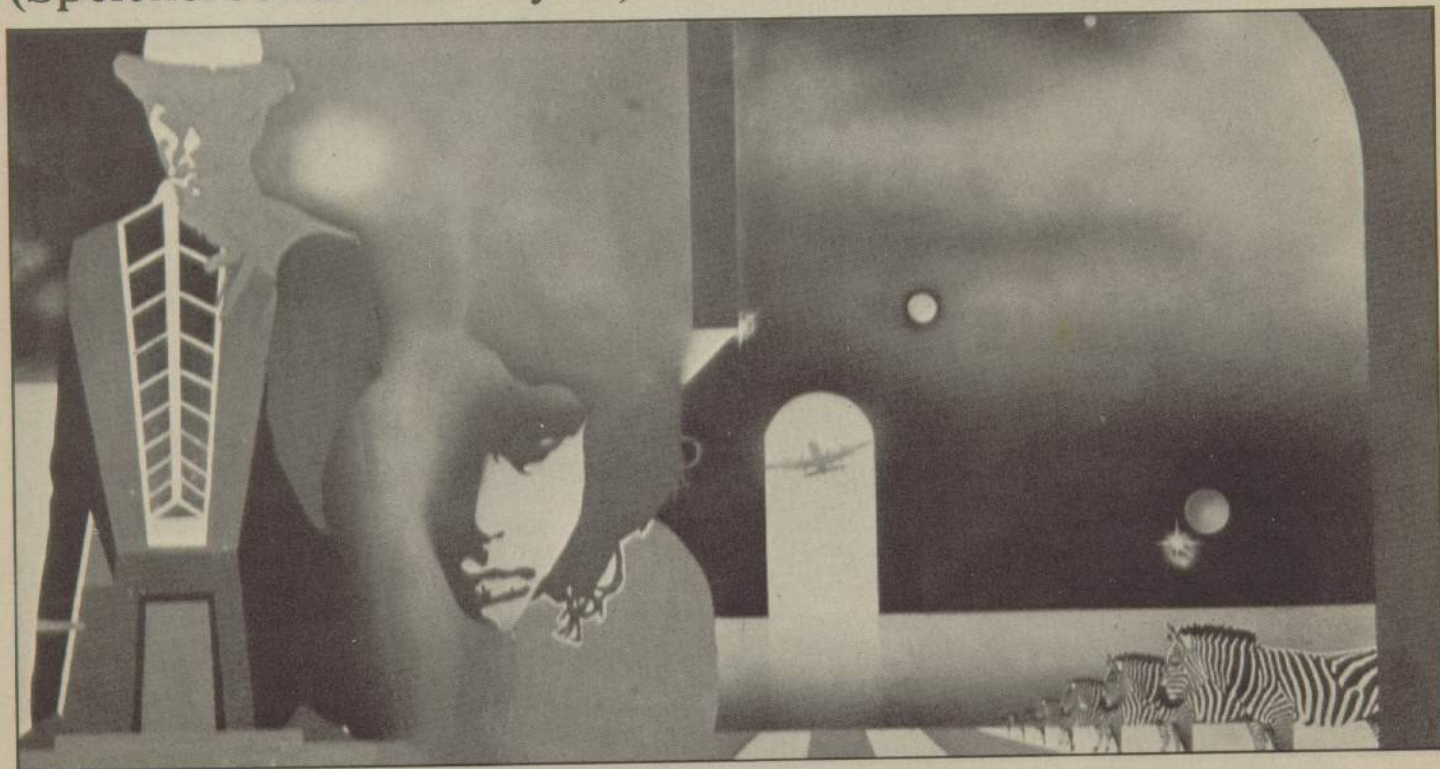
# ZX-USER CLUB



# PROGRAMME

## Siebzehn und vier

(Speicherbedarf 997 Bytes)



Bei diesem Spiel wird nicht gewürfelt, sondern es werden — wie es sich gehört — Karten gezogen. Die Regeln entsprechen in etwa denen des Originals, mit einer Abweichung: Wer zwei Asse zieht, hat hier 22 und damit verloren. Bei Punktgleichheit gewinnt der Computer, der als Bankhalter fungiert.

Um das Programm auch auf der 1K-Version des ZX81 lauffähig zu machen, wurde es in zwei Teile aufgeteilt, von denen das erste von der Kassette geladen werden kann. Dann müssen die Kartennamen und Kartenwerte eingegeben werden:

```
10 DIM A$(8,2)
20 DIM A(8)
30 FOR N=1 TO 8
40 INPUT A$(N)
50 INPUT A(N)
60 NEXT N
```

```
10 LET G=100
20 PRINT G
30 INPUT E
40 LET G=G-E
50 CLS
60 LET SC=0
```

Namen: »10« »9« »8« »7« »B« »D« »K« »A«

Werte: 10 9 8 7 2 3 4 11

Nun müssen die Programmzeilen 10 bis 60 einzeln mit den Zeilen des Spielprogramms überschrieben werden. (Nicht NEW!!! und nicht mit LOAD laden, da sonst die vorher eingegebenen Werte wieder gelöscht werden.) Aus diesem Grund darf das Programm auch nur mit GOTO 10 gestartet werden.

Der Computer zeigt zunächst das Guthaben und fragt nach dem Einsatz. Mit dem Fragezeichen möchte er dann wissen, ob Sie eine Karte wollen (»J«)

```
70 LET R=0
80 LET S=0
90 LET R=R+3
100 PRINT AT 0,R;"?"
110 IF INKEY#<"N" AND INKEY#<"J" THEN GOTO 110
120 PRINT AT 0,R;CHR$(0)
130 IF INKEY#="N" THEN GOTO 100
140 LET K=INT (RND*8)+1
150 LET S=S+A(K)
160 PRINT AT 0,R;A$(K)
170 IF S>21 THEN GOTO 1000
180 GOTO 90
1000 PRINT S
```

oder nicht (»N«). Dann zeigt er die Karte an usw., bis Sie ihm mit »N« zu verstehen geben, daß er seine Karten ziehen kann.

Nun folgt die Abrechnung und wenn Sie noch Lust zum Weiterspielen haben, drücken Sie einfach irgend eine Taste.

**Achtung:** Wenn Sie sich beim nächsten Mal die Eingabe sparen wollen, unterbrechen Sie das Spiel und save es mit dem Namen. Sie müssen nur darauf achten, daß Sie es beim nächsten Mal wieder mit »GOTO 10« starten.

K. Engelhard, Hamburg

```
1010 LET R=0
1020 IF 21-SC<4 THEN GOTO 1500
1030 LET K=INT (RND*8)+1
1040 LET R=R+3
1050 PRINT AT 5,R;A$(K)
1060 LET SC=SC+A(K)
1070 IF SC>21 THEN GOTO 1500
1080 GOTO 1020
1500 PRINT SC
1510 LET S=(S AND S<22)
1520 LET SC=(SC AND SC<22)
1530 LET G=G+(2*S AND S>SC)
1540 PAUSE 4E4
1550 CLS
1560 GOTO 20
```

**ZX-USER CLUB**

1K



## Neu von Sybex:

### MEIN SINCLAIR ZX81

Der Mikrocomputer Sinclair ZX81 wurde in kürzester Zeit zu einem der erfolgreichsten Geräte auf dem Weltmarkt. Das Buch erläutert die vielfältigen Möglichkeiten, z.B. den Anschluß an Fernsehgerät und Kassettenrecorder, gibt Tips zur eigenen Programmgestaltung für Grafiken, Kalkulationen, Spiele. Eine Einführung in die leicht erlernbare Programmiersprache BASIC sowie lauffähige Programme machen dieses Handbuch zum Muß für jeden Sinclair-Besitzer.



Dr. Hergert  
**MEIN SINCLAIR ZX81**  
173 Seiten, 120 Abb.  
Ref.-Nr. 3021 (1983)  
DM 25,-

**Aus dem Inhalt:**  
Der erste Eindruck  
Die Programmeingabe  
Ein kurzer und  
anschaulicher BASIC-Kurs  
Wir können nicht  
nur bis drei zählen  
Strings- und  
Stringfunktionen  
Der BASIC Wortschatz

Sybex-Bücher sind erhältlich  
bei Ihrem Fachhändler.  
Fragen Sie danach!

**Verlagsauslieferung:**  
Berlin: Buchhandlung  
Billig GmbH, Blissestr. 61  
Österreich: Fachbuch-Center  
ERB, Amerlingstr. 1, 1061 Wien  
Schweiz: Versandbuchhandlung  
Thali AG, Industriest. 2,  
6285 Hitzkirch

Direktbestellungen beim Verlag  
gegen Verrechnungsscheck  
(+DM 2,50 Versandkostenanteil)

Fordern Sie ein  
Gesamt-Buch-Verzeichnis an.

Sie finden Sybex auf der  
SYSTEMS 83, Halle 23, Stand 604



**SYBEX-VERLAG** <sup>GM</sup><sub>BH</sub>

Abt. UC 1083 Postfach 120513  
4000 DÜSSELDORF 12  
Tel. 0211/287066, Telex 8588163

## Neu von Sybex:

### SINCLAIR ZX81 BASIC Handbuch

Ihr ZX81 spricht BASIC.  
Sprechen Sie seine Sprache!  
Dieses Buch hilft Ihnen,  
alle Möglichkeiten Ihres  
ZX81 zu beherrschen. Das  
vollständige BASIC  
Vokabular wird anhand  
von praktischen Beispielen  
erläutert und beschrieben.  
Anhand der vielen Pro-  
grammbeispiele lernen Sie  
das Programmieren mit  
Ihrem ZX81.



D. Hergert  
**SINCLAIR ZX81  
BASIC Handbuch**  
184 Seite, 120 Abb.  
Ref.-Nr. 3028 (1983)  
DM 28,-

**Sprechen Sie  
die Sprache  
die Ihr  
ZX81  
versteht!**

Sybex-Bücher sind erhältlich  
bei Ihrem Fachhändler.  
Fragen Sie danach!

**Verlagsauslieferung:**  
Berlin: Buchhandlung  
Billig GmbH, Blissestr. 61  
Österreich: Fachbuch-Center  
ERB, Amerlingstr. 1, 1061 Wien  
Schweiz: Versandbuchhandlung  
Thali AG, Industriest. 2,  
6285 Hitzkirch

Direktbestellungen beim Verlag  
gegen Verrechnungsscheck  
(+DM 2,50 Versandkostenanteil)

Fordern Sie ein  
Gesamt-Buch-Verzeichnis an.

Sie finden Sybex auf der  
SYSTEMS 83, Halle 23, Stand 604

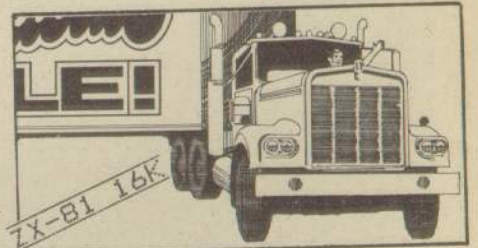


**SYBEX-VERLAG** <sup>GM</sup><sub>BH</sub>

Abt. UC 1083 Postfach 120513  
4000 DÜSSELDORF 12  
Tel. 0211/287066, Telex 8588163

## Endlosschlange

Sie müssen versuchen, mit Ihrer Schlange so lange wie möglich über den Bildschirm zu kriechen, ohne ein Hindernis zu berühren. Gesteuert wird mit 1 und 0 für links und rechts. Ein Druck auf eine der beiden Tasten bewirkt eine Richtungsänderung um 90 Grad. Punkte bekommt man nur, wenn man »Neuland« überfährt. Nach je zwei Durchläufen wird das Spielfeld neu aufgebaut. Newline für Start drücken.



```

1 REM 70 BELIEBIGE ZEICHEN
2 GOSUB 100
3 POKE 16416,0
10 RAND USR 16514
11 RAND
12 FOR N=1 TO 50
14 PRINT AT (INT (RND*20)+1), (
INT (RND*30)+1);CHR$ 128
16 NEXT N
17 FOR I=1 TO 2
18 LET P=0
20 LET A=(PEEK 16396+256*PEEK
16397)+300
22 LET S=0
24 POKE A,23
26 IF INKEY$(">") THEN GOTO 26
27 IF INKEY$=" " THEN GOTO 27
28 LET S=S+(INKEY$="0")-(INKEY$
$="1")
30 IF S>3 OR S<-3 THEN LET S=0
32 LET A=A+(S=1 OR S=-3)-(S=-1
OR S=3)+(ABS S=2)*33)-(S=0)*33
34 LET B=PEEK A
36 IF B=128 THEN GOTO 50
37 POKE A,23
38 IF B<>23 THEN LET P=P+1
40 RAND P
41 PRINT AT 23,10;
42 LET L=USR 16553
44 GOTO 28
50 POKE A,136
52 FOR N=1 TO 10
54 NEXT N
56 IF CODE INKEY$(">")=118 THEN GO
TO 58
60 POKE A,128
61 POKE 16567,23
62 RAND USR 16561
64 PRINT AT 23,10;"0"
65 PRINT AT 23,10;
66 NEXT I
67 POKE 16567,23
68 RAND USR 16561
69 POKE 16567,128
70 RAND USR 16561
71 PRINT AT 0,0;
72 GOTO 3
100 PRINT " E N D L O S - S C H
L A N G E "
105 PRINT "SIE MUESSEN MIT IH
RER SCHLANGE UEBER MOEGLICHT U
IEL FLAECHEN FAHREN OHNE DABEI
GEGEN EIN HINDERNIS ZU FAHRE
N."
110 PRINT "GESTEUERT WIRD MIT >
1< UND 0< FUER LINKS UND RECHT
S.EIN DRUCK AUF EINE DER BEIDEN
TASTEN BE- WIRKT EINE RICHTUNGS
AENDERUNG UM90 GRAD.PUNKTE BEKOM
MT MAN NUR, UENN MAN >NEULAND< U
EBERFAEHRT."
115 PRINT "NACH JE ZUEI DURCHLA
EUFEN WIRD DAS SPIELFELD NEU AU
FGEBAUT."
120 PRINT AT 18,0;"NEW-LINE FUE
R START DRUECKEN"

```



```

125 IF CODE INKEY$(<>118 THEN GO
TO 125
130 CLS
140 RETURN

```

16514	42	2A
16515	12	0C
16516	64	40
16517	6	06
16518	32	20
16519	35	23
16520	54	36
16521	128	80
16522	16	10
16523	251	FB
16524	17	11
16525	33	21
16526	0	00
16527	6	06
16528	21	15
16529	25	19
16530	54	36
16531	128	80
16532	16	10
16533	251	FB
16534	42	2A
16535	12	0C
16536	64	40
16537	35	23
16538	6	06
16539	21	15
16540	25	19
16541	54	36
16542	128	80
16543	16	10
16544	251	FB
16545	6	06
16546	31	1F
16547	35	23
16548	54	36
16549	128	80
16550	16	10
16551	251	FB
16552	201	C9
16553	237	ED
16554	75	4B
16555	50	32
16556	64	40
16557	205	CD
16558	152	98
16559	10	0A
16560	201	C9
16561	42	2A
16562	12	0C
16563	64	40
16564	14	0E
16565	22	16
16566	62	3E
16567	23	17
16568	6	06
16569	32	20
16570	35	23
16571	190	BE
16572	204	CC
16573	201	C9
16574	64	40
16575	16	10
16576	249	F9
16577	35	23
16578	62	3E
16579	0	00
16580	13	0D
16581	185	B9
16582	32	20
16583	238	EE
16584	201	C9

## Neu von Sybex:

### SINCLAIR ZX SPECTRUM Programme zum Lernen und Spielen

Dieses Buch ist zur praktischen Anwendung bestimmt. Die wesentlichen Grundzüge des Programmierens beim SPECTRUM werden dargestellt. Programme aus dem kaufmännischen Bereich, Lehr- und Lernprogramme sowie viele Spiele helfen Ihnen in BASIC mit Ihrem Spectrum zu lernen. Sie erhalten auch direkt anwendbare Programme.



T. Hartnell  
**Sinclair ZX Spectrum  
Programme zum  
Lernen und Spielen**  
224 Seiten, 120 Abb.  
Ref.-Nr. 3022 (1983)  
DM 28,-

**Aus dem Inhalt:**  
Programmieren in Basic/  
Experimente mit den  
Farben des ZX Spectrum /  
Gebrauch des Spectrum im  
kaufmännischen Bereich /  
Der Spectrum als Lehr- und  
Lernmittel / Der Spectrum  
als Spielpartner / Ein Leit-  
faden für besseres  
Programmieren u.a.

Sybex-Bücher sind erhältlich bei Ihrem Fachhändler.  
**Fragen Sie danach!**

**Verlagsauslieferung:**  
Berlin: Buchhandlung  
Billig GmbH, Blissestr. 61  
Österreich: Fachbuch-Center  
ERB, Amerlingstr. 1, 1061 Wien  
Schweiz: Versandbuchhandlung  
Thali AG, Industriest. 2,  
6285 Hitzkirch

Direktbestellungen beim Verlag  
gegen Verrechnungsscheck  
(+DM 2,50 Versandkostenanteil)

**Fordern Sie ein  
Gesamt-Buch-Verzeichnis an.**

Sie finden Sybex auf der  
**SYSTEMS 83, Halle 23, Stand 604**



**SYBEX-VERLAG** <sup>GM</sup><sub>BH</sub>  
Abt. UC 1083 Postfach 120513  
4000 DÜSSELDORF 12  
Tel. 0211/287066, Telex 8588163

## Neu von Sybex:

### SINCLAIR ZX SPECTRUM BASIC HANDBUCH

Für jeden SPECTRUM-Anwender stellt dieses Handbuch eine wichtige Hilfe dar. Es gibt eine Übersicht für alle BASIC-Begriffe, die auf diesem Rechner verwendet werden können. Anhand von Beispielen werden diese Begriffe ausführlich erläutert. Zahlreiche Programme dienen zugleich als Anregung für den Anwender, um fremde Programme besser verstehen und eigene leicht entwickeln zu können.



D. Hergert  
**SINCLAIR  
ZX SPECTRUM  
BASIC HANDBUCH**  
ca. 224 Seiten  
ca. 150 Abbildungen  
Ref.-Nr. 3027 (1983)  
DM 32,-

**Sprechen Sie  
die Sprache  
die Ihr  
SPECTRUM  
versteht!**

Sybex-Bücher sind erhältlich bei Ihrem Fachhändler.  
**Fragen Sie danach!**

**Verlagsauslieferung:**  
Berlin: Buchhandlung  
Billig GmbH, Blissestr. 61  
Österreich: Fachbuch-Center  
ERB, Amerlingstr. 1, 1061 Wien  
Schweiz: Versandbuchhandlung  
Thali AG, Industriest. 2,  
6285 Hitzkirch

Direktbestellungen beim Verlag  
gegen Verrechnungsscheck  
(+DM 2,50 Versandkostenanteil)

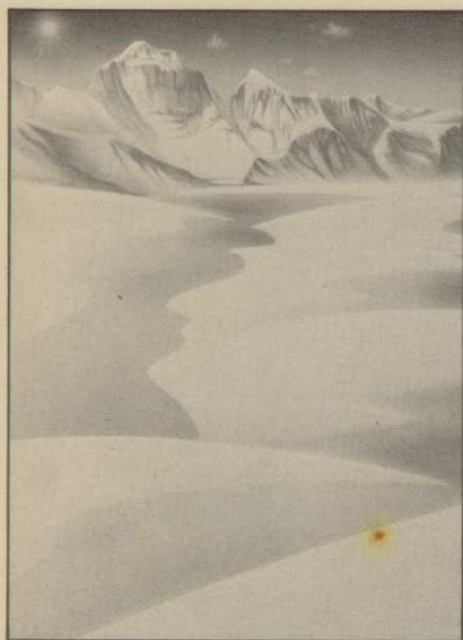
**Fordern Sie ein  
Gesamt-Buch-Verzeichnis an.**

Sie finden Sybex auf der  
**SYSTEMS 83, Halle 23, Stand 604**



**SYBEX-VERLAG** <sup>GM</sup><sub>BH</sub>  
Abt. UC 1083 Postfach 120513  
4000 DÜSSELDORF 12  
Tel. 0211/287066, Telex 8588163





## Gratwanderung

Dieses Spiel erfordert Geschick und starke Nerven und ist nicht das Richtige, wenn Sie gerade mit Ihren Eltern gestritten haben. Versuchen Sie, den Windungen des Gipfelwegs zu folgen. Mit der Taste »A« geht es aufwärts, mit »Z« abwärts. Drücken Sie am Ende eine beliebige Taste, und es entsteht ein neuer Höhenweg für Sie.

```
10 LET Y=10
20 PRINT AT Y,0,"-----"
30 FOR X=5 TO 25
40 PRINT AT Y,X,"-";
50 LET Y=Y+INT (RND*3)-1
60 IF Y<5 THEN LET Y=5
70 IF Y>15 THEN LET Y=15
80 NEXT X
90 PRINT "ENDE"
100 LET Y=10
102 LET X=0
104 PRINT AT Y,X,"-";
110 LET X=X+1
120 LET Y=Y+(INKEY#="Z")-(INKEY#="A")
130 PRINT AT Y,X;
140 LET N=PEEK (PEEK 16398+PEEK 16399*256)
150 PRINT "■"
160 FOR P=3 TO 10
170 NEXT P
180 IF N=CODE "■" THEN GOTO 240
190 IF N=CODE "-" THEN GOTO 104
200 PRINT "ABGESTUERTZT"
210 PAUSE 4E4
220 CLS
230 RUN
240 PRINT "GESCHAFFT"
250 GOTO 210
```

14



## Simon

Sie sollen versuchen, die auf dem Bildschirm erscheinende Buchstabenfolge zu wiederholen. Geben Sie sie zusammenhängend ein und drücken Sie NEWLINE. Das Spiel endet, wenn es Ihnen gelungen ist, 20 Buchstaben richtig einzugeben.

14

```
10 LET A#=CHR# INT (RND*8+38)
20 LET P=0
30 LET P=P+1
40 FOR X=1 TO P
50 PRINT AT 0, CODE A#(X)-38;A#
(X)
60 PAUSE 30
70 CLS
80 NEXT X
90 INPUT B#
100 IF A#<>B# THEN GOTO 160
110 LET A#=A#+CHR# INT (RND*8+38)
120 IF P<20 THEN GOTO 30
130 PRINT "ALLE 20 WIEDERHOLT"
140 PRINT "SEHR GUT"
150 STOP
160 PRINT B#;" IST FALSCH"
170 PRINT "RICHTIG WAERE: ";A#
```



## Mauer einwerfen

In dieser Version des bekannten Automatenspiels bewegen Sie mit den Tasten »M« und »N« Ihren Schläger und halten damit Ihren (einzigen) Ball so lange im Spiel, bis er die Mauer eingeworfen hat. Um das Spiel wieder zu starten, drücken Sie »R«.

```
1 LET C=6
2 LET E=C/C
3 LET B=C-E
4 FOR X=E TO C
5 PRINT "-----"
6 NEXT X
7 FOR X=E TO C-E
8 PRINT "■";TAB 12;"■"
9 NEXT X
10 LET D=E
11 LET F=D
12 PRINT AT 11,B;" "
13 LET B=B+(INKEY#="M")-(INKEY#="N")
14 IF B<ABS E THEN LET B=ABS E
15 IF B>10 THEN LET B=10
16 PRINT AT 11,B;
17 PRINT AT C,D;" "
18 IF C=C-C OR C=10 AND B<>D AND B+1<>D THEN GOTO 30
19 LET C=C+E
20 LET D=D+F
21 IF D<ABS E+ABS F OR D>11-ABS F THEN LET F=-F
22 PRINT AT C,D;
23 IF C=10 THEN GOTO 27
24 IF PEEK (PEEK 16398+PEEK 16399*256)=CODE "■" THEN LET E=ABS E
25 PRINT "."
26 GOTO 12
27 LET E=-E
28 LET F=SGN F*(INT (RND+RND)+ABS E)
29 GOTO 25
30 CLS
31 RUN
```

14



# COMPUTER ACCESSOIRES INT'L



## EPSON FX-80

160 Zeichen pro sec., 136 Schriftarten, Grafik, 4 k-Byte RAM Charaktergenerator mit Interlace u. Kabel für ZX81, Nr. 101 DM 1.998,- EPSON RX-80 m. Interf. u. Kabel Nr. 102 DM 1.398,-



## SHINWA CP 80

80 Zeichen pro sec., 228 ASCII-Zeichen, 4 Schriftarten, Grafik, Nr. 103 DM 945,-



## BROTHER EP 20

Die Super-Schreibmaschine Nr. 104 DM 395,- BROTHER EP-20 INTERFACE für ZX81. Gleichzeitig als Eingabe- und Ausgabegerät verwendbar! Centronics-Interface eingebaut. Nr. 105 DM 578,-



## ALPHACOM 32

32 Zeichen pro Zeile, 100% kompatibel mit ZX81 und SPECTRUM. Alle Grafikzeichen und hochaufl. Grafik kann ausgedruckt werden. Incl. Stromversorgung. Nr. 106 DM 298,-



## FORTH

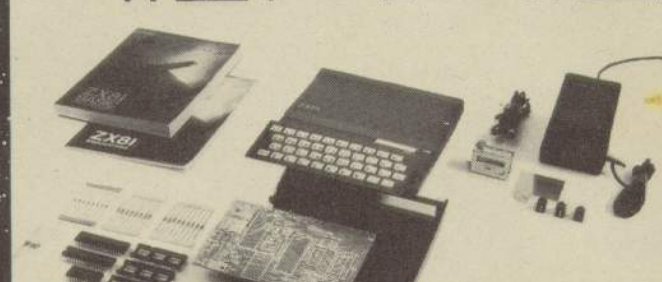
Mindest. 5 mal so schnell wie BASIC, durch den modularen Aufbau sehr flexibel. SPECTRUM 48K-RAM erforderlich. Nr. 021 DM 98,-



## BACKGAMMON

Tolle Auflösung, sehr spielstark Nr. 022 DM 29,80 THE GAUNTLET Ein Weltraumspiel. Nr. 023 DM 24,80

# SINCLAIR

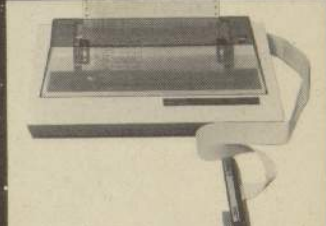


## BAUSATZ ZX81

Preissensation!

Den ZX81 Bausatz mit der ausführlichen Original SINCLAIR-Beschreibung, ausführlicher Bauanleitung für nur DM 129,-. 8K-Byte BASIC ROM, 1K-Byte RAM, Z80A-CPU, komplett mit Netzteil, Anschlußkabel für TV und Kassettensrecorder. Nr. 001 DM 129,-

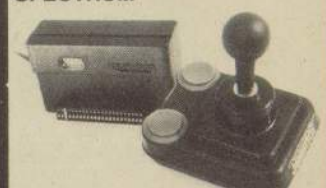
**ZX81  
BAUSATZ  
NUR  
DM 129,-**



## SEIKOSHA GP-100A MARK II

50 Zeichen pro sec., Incl. Centronics Interface für ZX81. Nr. 116 DM 798,-

## SPECTRUM



## KEMPSTON JOYSTICK

Joystick mit Interface, der meistverkaufteste in England, daher sind viele Spiele von Quicksilver PSS, Vision u. a. programmiert. Nr. 118 DM 98,-



## SPECTRUM

3-D-STRATEGY, 4-dimensionale Mühle Nr. 024 DM 39,- SMUGGLER COVE, Schatzsuche, Nr. 025 DM 39,- VELNOR'S LAIR, Abenteuer, Nr. 026 DM 39,- AQUAPLANE, Wasserski gefährlich, Nr. 027 DM 39,- XADOM, versch. Spielerebenen, Nr. 028 DM 39,-



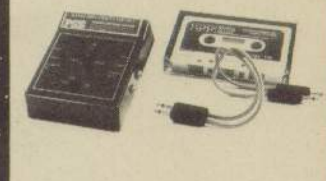
## ZX81 SPRACHSYNTHESIZER

250 deutsche festprogrammierte Begriffe, Lautsprecher, mittels 64 Phonemen eigene Wortschöpfungen leicht selbst zu programmieren. Nr. 107 DM 495,-



## KEMPSTON-CENTRONICS-INTERFACE für SPECTRUM

Per Software auf Cassette Seikosha, Epson, Shinwa und andere Drucker ansteuerbar, hochaufl. Grafik voll ausdrückbar. Nr. 108 DM 195,- mit Kabel



## Q-SAVE VON PSS

Die Übertragungsrate wird von 250 auf 4000 Baud erhöht, 16 mal schneller! Mit Software für 16 und 64K-RAM. Nr. 029 DM 79,-



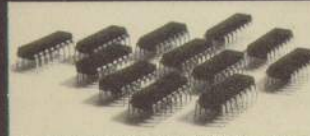
## DCP-SPEECH-PACK

8K-Byte-Rom. Enthält alle Zahlen zwischen 0 und 1 Mio., das gesamte Alphabet und einige Wörter. Erweiterbar durch Word-ROM's. Nr. 109 DM 198,-



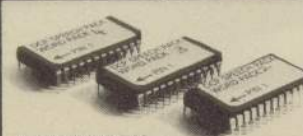
## INTERSPEAK PACK

In- u. Output Ports: 8 Bit TTL kompatibel, schaltbare Eingänge: 4, gepuffert, Relais-Ausgänge: 4, belastbar mit 1 Amp. 24 V. Erweiterungsbus. Nr. 110 DM 198,-



## SPECTRUM-AUFRÜSTSATZ

Durch Einsetzen von 12 IC's rüsten Sie Ihren 16K auf 48K um. (Bitte bei Best. ISSUE TWO oder THREE angeben. Steht auf der Platine rechts unten) Nr. 111 DM 98,-



## WORDPACK-ERWEITERUNGS-ROM'S

Durch diese 3 ROM's stehen Ihnen viele neue Wörter zur Verfügung. Wortliste anfordern! Nr. 112, 113, 114 DM 66,-, alle 3 zusammen Nr. 115 DM 178,-

## BESTELLCOUPON

Hiermit bestelle ich ☐ per Vorausscheck ☐ per Nachnahme (zuzügl. Nachnahmegeb.)

Stück	Art.-Nr.	Preis

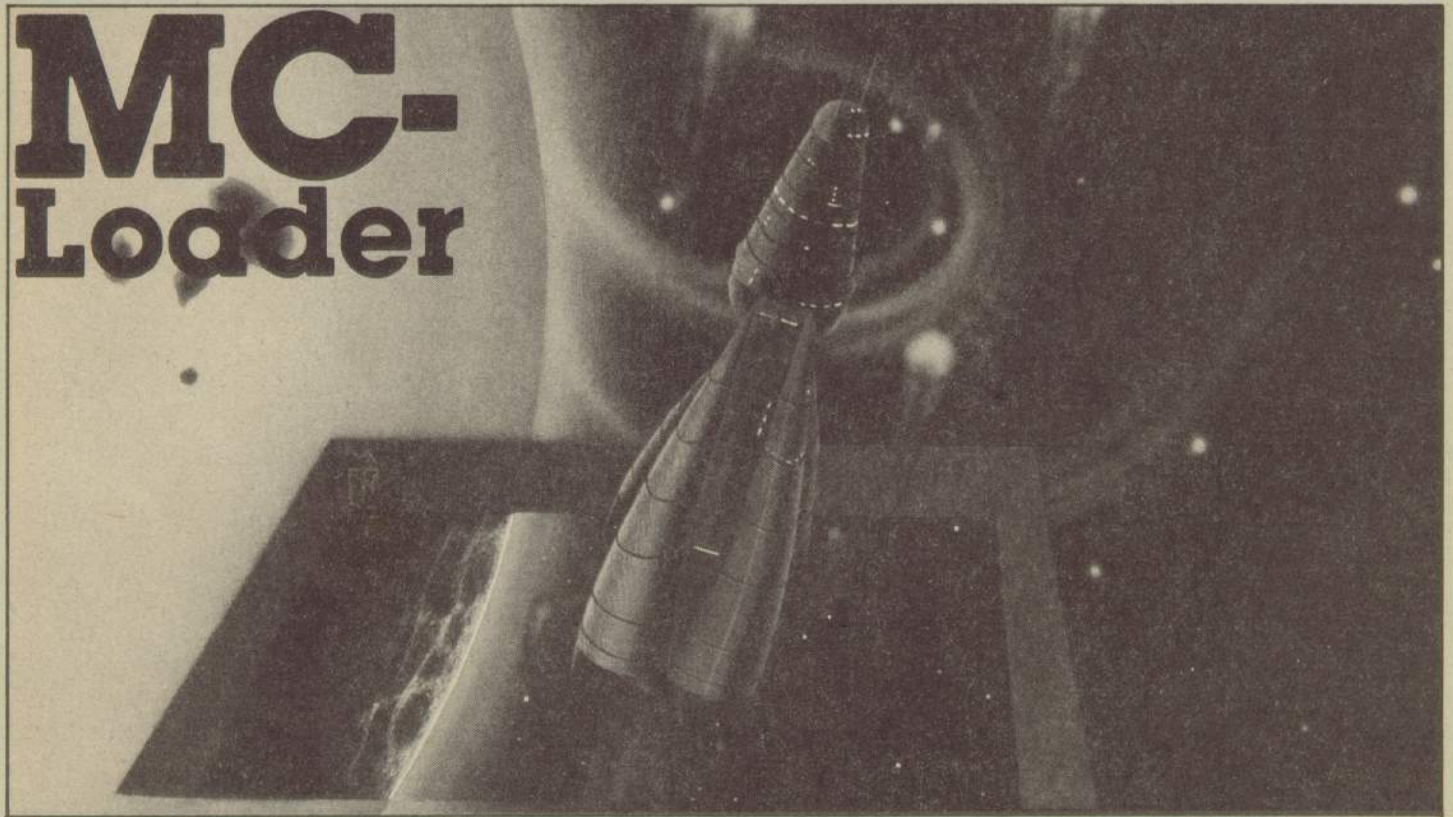
Name \_\_\_\_\_  
 Straße \_\_\_\_\_  
 PLZ/Ort \_\_\_\_\_  
 Datum \_\_\_\_\_ Unterschrift \_\_\_\_\_

Bei Bestellungen unter DM 250,- zuzügl. Versandkosten.

COMPUTER ACCESSOIRES INT'L · Jägerweg 10 · 8012 Ottobrunn



## MC-Loader



Verehrter Leser,  
da wir in dieser Ausgabe MC-Programme behandeln, stellen wir Ihnen 3 verschiedene Möglichkeiten (Hilfsprogramme) vor, um ein MC-Programm an eine bestimmte Adresse zu übertragen (Poken).  
Nehmen wir z. B. untenstehendes Programm, das den freien Speicherplatz ausrechnet:

```
4082 210000 LD HL,0000
4085 39 ADD HL,SP
4086 ED5B1C40 LD DE,(401C)
408A ED52 SBC HL,DE
408C 44 LD B,H
408D 4D LD C,L
408E C9 RET
```

```
16514 33
16515 0
16516 0
16517 57
16518 237
16519 91
16520 28
16521 54
16522 237
16523 92
16524 68
16525 77
16526 201
```

Wir reservieren für unser MC-Programm zuerst einmal freien Platz, indem wir in eine REM-Zeile so viele be-

liebige Zeichen eingeben wie benötigt (in der Regel reserviert man mehr, als gebraucht wird), da es vor dem BASIC-Programm stehen soll. Wir geben also ein:

```
1 REM AAAAAAAAAAAAAA und anschließend das Hilfsprogramm von Zeile 5—40. Starten Sie jetzt das Hilfsprogramm mit RUN oder GOTO 1.
```

Mit Programm Nr. 1 geben Sie Adresse für Adresse die jeweiligen Kode des MC-Programms in Dezimalzahlen ein.

```
5 PRINT "ANZAHL DER BYTES"
6 INPUT A
10 FOR I=16514 TO 16514+A
20 INPUT B
30 POKE I,B
40 NEXT I
```

Beispiel: 33 NEWLINE  
0 NEWLINE  
0 NEWLINE  
, ,  
, ,  
20 1 NEWLINE

Wenn Sie nun wissen wollen, wieviel Speicherplatz noch frei ist, rufen Sie das MC-Programm mit PRINT USR 16514 auf.

Programm 2 läuft im Prinzip wie Programm 1 ab. Nur mit dem Unter-

schied, daß statt Dezimalzahlen jetzt Hexadezimalzahlen eingegeben werden. Hierfür wieder ein Beispiel:

```
5 PRINT "ANZAHL DER BYTES"
6 INPUT A
10 FOR I=16514 TO 16514+A
20 INPUT A$
30 POKE I,(CODE A$(1)-28)*16+CODE A$(2)-28
40 NEXT I
```

```
21 NEWLINE
00 NEWLINE
00 NEWLINE
, ,
, ,
```

C9 NEWLINE

Aufrufe des MC-Programms mit PRINT USR 16514

Programm 3 unterscheidet sich von Programm 2 insofern, daß nicht Schritt für Schritt abgefragt wird, sondern direkt aus einem STRING (A\$) die Hexadezimalzahlen gelesen und umgerechnet und an die jeweilige Adresse gepoket werden.

Aufruf durch PRINT USR 16514

```
10 LET A$="21000039ED5B1C40ED5244DC9"
20 FOR I=1 TO LEN A$ STEP 2
30 POKE I/2+16513,(CODE A$(I)-28)*16+CODE A$(I+1)-28
40 NEXT I
```

# ZX-USER CLUB



# Birkhäuser Computer Shop

**Hervorragende Einführungsbücher  
für Sinclair-Home Computer – die  
notwendigen Grundlagen zum Verstehen,  
Programmieren und Spielen.**

**Ian Stewart  
Robin Jones  
Sinclair ZX 81**  
Programme, Spiele, Graphik  
2. Auflage 1983  
144 Seiten, Broschur  
DM 28.80

**Ian Stewart  
Robin Jones  
Maschinencode  
und besseres BASIC**  
2., verbesserte Auflage 1983  
240 Seiten, Broschur  
DM 32.–

**Ian Stewart  
Robin Jones  
Sinclair ZX Spectrum**  
Programmieren leicht gemacht  
2. Auflage 1983  
192 Seiten, Broschur  
DM 29.80



**Ian Stewart  
Robin Jones  
Weitere Kniffe und  
Programme  
mit dem ZX Spectrum**  
1983. 160 Seiten, Broschur  
ca. DM 29.80

Dieser Folgeband zu "Sinclair ZX Spectrum – Programmieren leicht gemacht" hilft Ihnen dabei, noch mehr aus Ihrem ZX Spectrum herauszuholen. Das Buch präsentiert eine ganz neue Auswahl von Programmen und Anwendungen, die nur einen 16K-RAM-Zusatzspeicher benötigen, also mit beiden Versionen des Spectrum gefahren werden können.

Warum also nicht Ihr Spectrum erweitern?



**Ian Stewart  
Robin Jones  
ZX Spectrum Maschinencode**  
1983. 140 Seiten, Broschur  
ca. DM 27.80  
Lernen Sie Eigenschaften des Spectrum-Betriebssystems kennen, die man mit Maschinencode nutzen kann: Attribut- und Display-Dateien, Systemvariablen und die Struktur des BASIC-Programmiersbereichs. Nutzen Sie die Schnelligkeit von Maschinencoderroutinen in Ihren eigenen Programmen. Mit einem siebenteiligen Anhang, der das Programmieren in Maschinencode erleichtert, ist das alles gar nicht so schwer. Das Buch enthält ausserdem ein komplettes Listing aller Z80 Opcodes in alphabetischer Reihenfolge und ein BASIC-Programm, um Maschinencode aufzubereiten, zu laden, sicherzustellen und zu fahren.

**Martin Wren-Hilton  
Spiele mit dem ZX Spectrum**  
1983. ca. 72 Seiten, Broschur  
ca. DM 15.–

Können Sie  
– Einen Geheimcode knacken?  
– Auf einem Motorrad über neun Autobusse springen?  
– Ein feindliches Schiff in die Luft sprengen?  
Vielleicht sind aber "Gärtner spielen" oder "Regenwurm-Rennen" eher Programme nach Ihrem Geschmack. Alle diese Spiele sind dazu da, um Ihre Fähigkeiten und die Möglichkeiten des ZX Spectrum zu testen. Das Buch enthält ausserdem einige Programme, die Ihnen beim Schreiben von eigenen Spielen helfen sollen. Alle beschriebenen Programme können auf dem normalen ZX Spectrum mit dem 16-K-RAM Speicher gefahren werden.



**Martin Wren-Hilton  
ZX 81 und  
ZX Spectrum Ausbaubuch**  
Möglichkeiten und Grenzen  
mit Zusatzgeräten  
1983. ca. 100 Seiten, Broschur  
ca. DM 19.80

O.K., Sie haben Ihren ZX 81 oder ZX Spectrum bekommen und gelernt, wie man ihn programmiert. Nun wollen Sie aber mal etwas wirklich Nützliches mit ihm anstellen!  
Dieses Buch bietet einen Überblick über die Hardware, die Sie zu Ihrem ZX 81 und ZX Spectrum kaufen können und zusätzlich einige sehr brauchbare Programme, um diese Hardware dann auch zum Laufen zu bringen.

**Ian Stewart  
Robin Jones  
ZX-Spielkiste**  
Spiele und Rätsel mit dem  
ZX 81 oder ZX Spectrum  
1983. 98 Seiten, Broschur  
DM 19.80

Das Rätsel hat eine lange Tradition, die bis aufs alte Babylon und noch weiter zurückgeht. Den Beweis dafür liefert die Tatsache, dass Rätsel nun auch in der Computeraera Anklang finden. Dieses Buch enthält Rätsel in der Form von Computerspielen, die auf die beiden Sinclair-Computer ZX 81 und ZX Spectrum zugeschnitten sind und Ihnen helfen werden, manch verregneten Sonntag kurzweiliger zu gestalten.



## In Vorbereitung:

**Owen Bishop  
Einfache Peripherie-  
geräte im Selbstbau**  
ca. 170 Seiten

**Jan Sinclair  
Programmieren mit  
dem Commodore 64**  
ca. 140 Seiten

Bei allen Daten, Preisen etc. Änderungen vorbehalten.  
Stand September 1983

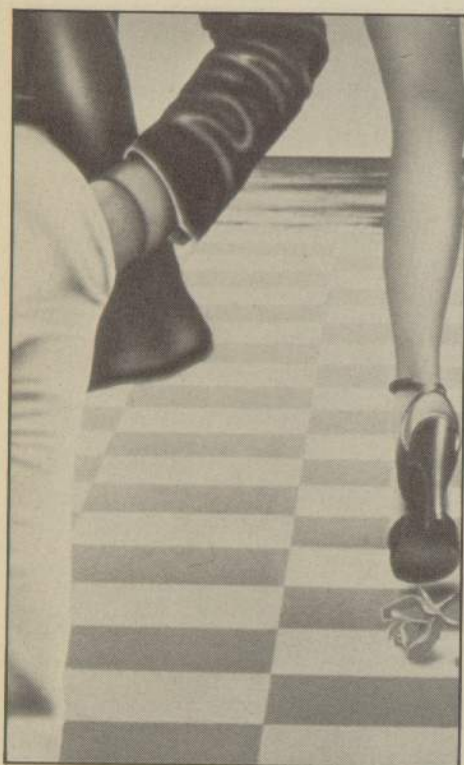
**Birkhäuser  
Verlag**  
Basel · Boston · Stuttgart



Die hier angezeigten  
Bücher sind erhältlich  
im Buchhandel  
und in vielen  
Computer Shops.



# PROGRAMME



## Schach-Uhr

Mit Hilfe dieses Programmes kann der ZX81 als Schach-Uhr verwendet werden. Die kleine Maschinencode-Routine benutzt die Systemvariable FRAMES zur Zeitmessung. Dadurch wird es möglich, die Zeit in 2-Sekunden-Abständen anzuzeigen. Zuerst muß diese Routine in Zeile 1 (17 Punkte) gepoked werden. Dazu dient das Ladeprogramm. Die Eingabe erfolgt dezimal, die entsprechenden Zahlen stehen in Zeile 2 im Hauptprogramm. Nach Eintippen des Hauptprogrammes wird dieses mit RUN 700 auf Band gespeichert. Die Schach-Uhr startet nach dem Laden dann automatisch. Als erstes wird das Zeit-Limit eingegeben, und zwar als dreistellige Zahl (H:MM). Das Minimum ist eine Minute (001),

das Maximum 9:59 (959). Nach NEWLINE beginnt die Zeit für Weiß zu laufen. Wenn Weiß gezogen hat, muß die Taste »W« gedrückt werden, bis der kleine Pfeil umspringt, Schwarz drückt entsprechend »S«. Hat einer der beiden Spieler das Zeit-Limit überschritten, schaltet die Uhr automatisch um und unter der Anzeige des Spielers wird »ZEIT-LIMIT« gedruckt. Die Genauigkeit der Uhr ist von der Netzfrequenz abhängig. Sollte sie zu schnell oder zu langsam laufen, kann die Geschwindigkeit durch POKE 16519,n beeinflusst werden. Bei n>100 wird sie langsamer gehen und bei n<100 schneller.

Dr. W. Steinmüller, Wien

```
16514 0
16515 58
16516 130
16517 64
16518 214
16519 100
16520 50
16521 130
16522 64
16523 33
16524 52
16525 64
16526 86
16527 186
16528 32
16529 252
16530 201
```

```
4082 00 NOP
4083 3A8240 LD A,(4082)
4086 D664 SUB 64
4088 328240 LD(4082),A
408B 213440 LD HL,4034
408E 56 LD D,(HL)
408F BA CP D
4090 20FC JR NZ,408E
4092 C9 RET
```

```
1 REM ...17 BELIEBIGE ZEICHEN
5 REM *****
6 REM * SCHACH UHR *
7 REM * (C) 1982 *
8 REM * W.STEINMUELLER *
9 REM *****
10 LET A=0
20 LET B=A
30 LET C=A
40 LET X=A
50 LET Y=A
```

```
60 LET Z=A
70 LET N=0
80 LET M=N
100 PRINT AT 4,8;"* SCHACH-UHR
*"
110 PRINT AT 2,8;"*****
*" AT 3,8;"*" TAB 21;"*" AT 5,8;
*" TAB 21;"*" AT 6,8;"*****
*****"
120 PRINT AT 8,8;".ZEIT-LIMIT =
." AT 7,8;". " TAB 21;". " AT 9,8;
." TAB 21;". " AT 10,8;". " TAB 2
1;"." AT 11,8;". "
130 INPUT L#
140 IF LEN L#<>3 THEN GOTO 130
150 PRINT AT 10,9;L#(1);".";L#(
2 TO 3);" H:MM"
160 PRINT AT 17,5;"WEISS";TAB 1
9;"SCHWARZ"
170 LET T1=0
180 LET T2=0
200 PRINT AT 15,5;N;".ZUG"
210 PRINT AT 19,3;">";TAB 17;"
"
220 LET N=N+1
230 POKE 16514,32-256*(PEEK 164
36>175)+PEEK 16436
300 PRINT AT 19,5;A;".";B;".";C
;"
310 IF A*100+B>=VAL L# THEN GOT
0 500
320 LET D=USR 16515
330 LET T1=T1+2-43200*(T1>46794
)
340 LET A=INT (T1/3600)
350 LET B=INT (T1/60-60*A)
360 LET C=T1-60*B-3600*A
370 IF INKEY#="W" THEN GOTO 500
380 IF A*100+B>=VAL L# THEN PRI
NT AT 21,5;"ZEIT-LIMIT"
390 GOTO 300
```

```
500 PRINT AT 15,19;M;".ZUG"
510 PRINT AT 19,3;" " TAB 17;">
"
520 LET M=M+1
530 POKE 16514,32-256*(PEEK 164
36>175)+PEEK 16436
600 PRINT AT 19,19;X;".";Y;".";
Z;" "
610 IF X*100+Y>=VAL L# THEN GOT
0 200
620 LET E=USR 16515
630 LET T2=T2+2-43200*(T2>46794
)
640 LET X=INT (T2/3600)
650 LET Y=INT (T2/60-60*X)
660 LET Z=T2-60*Y-3600*X
670 IF INKEY#="S" THEN GOTO 200
680 IF X*100+Y>=VAL L# THEN PRI
NT AT 21,19;"ZEIT-LIMIT"
690 GOTO 600
700 SAVE "SCHACH-UHR"
710 RUN
```

```
*****
* *
* SCHACH-UHR *
* *
*****
. ZEIT-LIMIT =.
. 2:30 H:MM.
.....
```

```
19.ZUG 18.ZUG
WEISS SCHWARZ
> 2:26:14 2:30:0
ZEIT-LIMIT
```

# ZX-USER CLUB



# ... die Zeitschrift mit Durchblick!

Information mit Tiefgang —  
Reports, die leben — Projekte  
ohne Kompromiß — Grundlagen  
glasklar — Tests mit Trenn-  
schärfe — Praxistips, die  
welche sind — Kritiken mit  
Biß — Software, die  
schmeckt — Hardware,  
die bekommt. Und  
dabei so aktuell, wie  
nur irgendwas.  
Kurzum:  
Die Zeitschrift  
mit Durchblick.



**ct** *magazin für  
computer  
technik*

die Herausforderung für Insider,  
der Einstieg für Einsteiger,  
ein neuer Anfang für alle.\*)

\*) Probeheft beim: Verlag Heinz Heise GmbH, Vertrieb **ct**, Postfach 27 46, 3000 Hannover 1



## Eheinstitut »Vergißmeinnicht«

Finden Sie den richtigen Partner



Der Computer produziert verdeckte Spielkarten, von denen Sie jeweils zwei aufdecken können. Haben Sie ein Pärchen gefunden, so zeigt er Ihnen die Partner, sonst verdeckt der Computer die aufgedeckten Karten. Nach Eingabe des Programms startet das Programm mit der Beschreibung.

```

5 GOTO 700
10 RAND
15 DIM B$(1)
18 DIM C$(1)
20 DIM A$(48)
25 LET D$="FALSCH - SCHON BENU
Tzt
28 LET GOES=0
30 LET A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ
TUWXY"
40 LET A$=A$( TO 24)+A$
45 LET T=0
50 FOR N=1 TO 6
60 FOR M=1 TO 10
70 LET A=1+RND*46.9
80 LET B=1+RND*46.9
90 LET C=A$(A)

```

16K

```

100 LET A$(A)=A$(B)
110 LET A$(B)=C$
120 NEXT M
130 PRINT N;"
"
135 PRINT ""
140 NEXT N
150 PRINT " 1 2 3 4 5 6
7 8"
160 PRINT AT 14,0;"WAHLEN SIE
DIE ERSTE KARTE "
170 GOSUB 900
180 LET I=VAL B$
190 LET J=VAL C$
200 LET M=(I-1)*8+J
210 IF A$(M)<>"Z" THEN GOTO 240
220 PRINT AT 16,0;D$
225 FOR O=1 TO 50
226 NEXT O
230 GOTO 160
240 LET I=2*(I-1)
250 LET J=J*3-1
260 PRINT AT I,J;"*"A$(M)
300 PRINT AT 14,0;"WAHLEN SIE
DIE ZWEITE KARTE "
310 GOSUB 900
320 LET K=VAL B$
330 LET L=VAL C$
340 LET N=(K-1)*8+L
350 IF A$(N)<>"Z" AND N<M THEN

```

```

GOTO 380
360 PRINT AT 16,0;D$
365 FOR O=1 TO 80
366 NEXT O
370 GOTO 300
380 LET K=2*(K-1)
390 LET L=L*3-1
400 PRINT AT K,L;"*"A$(N)
410 LET GOES=GOES+1
420 IF A$(N)=A$(M) THEN GOTO 50
0
430 FOR O=1 TO 80
440 NEXT O
450 PRINT AT I,J;" "
460 PRINT AT K,L;" "
470 GOTO 160
500 LET A$(N)="Z"
510 LET A$(M)="Z"
520 LET T=T+1
525 PRINT AT 14,0;"
"
530 PRINT AT 16,0;"ERFOLG - DAS
PAAR "T;" "
540 PRINT AT I,J;" "
550 PRINT AT K,L;" "
555 FOR O=1 TO 80
556 NEXT O
560 IF T<24 THEN GOTO 160
570 PRINT AT 14,0;"GEWONNEN - A
LLE SIND WEG "
580 PRINT "SIE HABEN ES IN "G
OES;" VERSUCHEN","GESCHAFFT."
590 PRINT "DRUECKEN SIE RUN U
ND NEWLINE UM NOCHMAL ZU SPIELEN
"
600 STOP
700 PRINT " FINDEN SIE DEN
PARTNER"
710 PRINT "
"
720 PRINT "IN DIESEM SPIEL MI
SCHT DER COMPUTER EIN SPIEL
KARTEN, DAS"
725 PRINT "AUS JE ZWEI VON 24 B
UCHSTABEN BESTEHT."
730 PRINT "ER WIRD DANN DIE K
ARTEN VEBDECKTZEIGEN."
740 PRINT "SIE MUESSEN ENTDEC
KEN, WELCHE KARTE ZU WELCHER G
EHOERT. SIE"
745 PRINT "WAHLEN EINE KARTE,
INDEM SIE ZUERST IHRE REIHENZA
HL + NEWLINEDRUECKEN, UND DANN I
HRE SPALTEN-ZAHL + NEWLINE EINGE
BEN."
750 PRINT "WENN SIE BEREIT SI
ND ANZUFANGEN,DRUECKEN SIE NEWLI
NE."
760 IF INKEY$="" THEN GOTO 760
770 CLS
780 GOTO 10
900 PRINT AT 16,0;"REIHE NUMMER
?"
910 INPUT B$
930 IF B$>"0" AND B$<"7" THEN G
OTO 960
940 PRINT AT 16,0;"FALSCH - REI
HE NUMMER?"
950 GOTO 910
960 PRINT AT 16,0;"SPALTE NUMME
R?"
970 INPUT C$
990 IF C$>"0" AND C$<"9" THEN R
ETURN
1000 PRINT AT 16,0;"FALSCH - SPA
LTE NUMMER?"
1010 GOTO 970

```



# Sehen Sie mal, was **micro** alles zu bieten hat...

Monat für Monat übersichtlich und leicht lesbar  
alles über Hardware, Software und Peripherie-  
geräte: Marktübersichten · Testberichte ·  
Anwendungsbeispiele aus der Praxis · Programm-  
listings · Computer-News aus aller Welt · Club-  
nachrichten. Last not least: Mit dem Computer  
lernen.

Für 5 Mark erhalten Sie  
**micro** jetzt bei Ihrem Zeit-  
schriftenhändler.

The image shows the cover of the magazine 'micro', issue from September 1983. A cartoon character with a large mustache and a top hat is pointing at the title. The cover features several illustrations of early personal computers, including desktop units and a portable model. Text on the cover includes 'DM 5,-', 'September 83', 'Das Praxismagazin für Mikrocomputer', and a circular badge that says 'HARDWARE TESTS TIPS SOFTWARE'. At the bottom, there are several headlines: 'Software für Profis', 'Lisa im Test', 'Tisch- und Taschenrechner', 'Drucker, Plotter, Datenträger', 'Lernen mit Mikros', and 'Neuer Portable aus USA'. A yellow banner at the bottom right says 'Die neue Bertelsmann Fachzeitschrift'.

DM 5,-

**micro**

September 83

Das Praxismagazin für Mikrocomputer

HARDWARE  
TESTS TIPS  
SOFTWARE

Software für Profis  
Lisa im Test  
Tisch- und Taschenrechner  
Drucker, Plotter,  
Datenträger  
Lernen mit Mikros  
Neuer Portable aus USA

Die neue Bertelsmann  
Fachzeitschrift





# PROGRAMME

## Monaco 1983

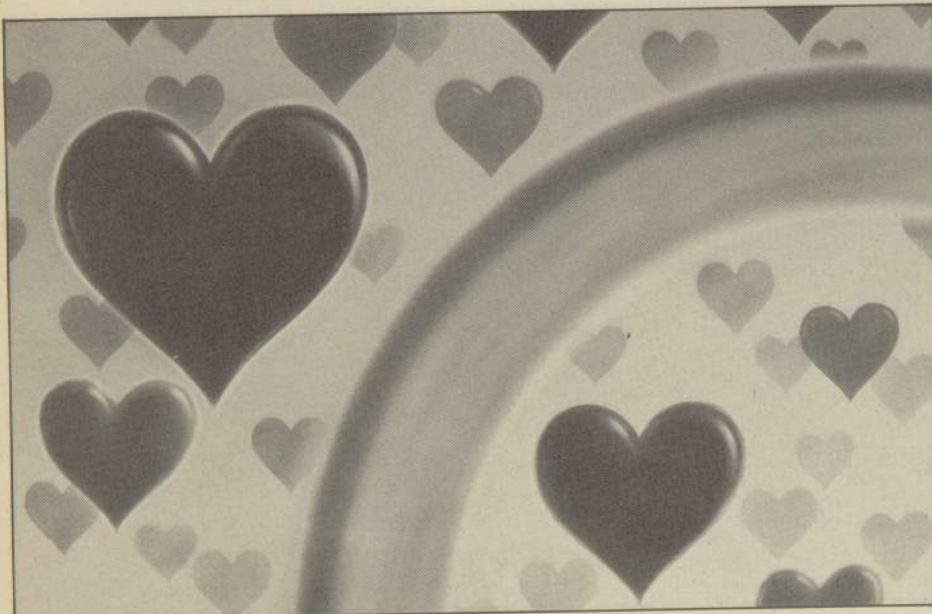
Nikki Lauda wieder im Rennen

40C1 BE  
40C2 C0  
40C3 3617  
40C5 0E00  
40C7 C9

CP(HL)  
RET NZ  
LD(HL),17  
LD C,00  
RET

Dezimal

16514	0
16515	0
16516	58
16517	37
16518	64
16519	254
16520	127
16521	32
16522	22
16523	58
16524	38
16525	64
16526	254
16527	239
16528	32
16529	4
16530	62
16531	255
16532	24
16533	6
16534	254
16535	247
16536	32
16537	7
16538	62
16539	1
16540	33
16541	190
16542	64
16543	134
16544	119
16545	42
16546	12
16547	64
16548	1
16549	182
16550	2
16551	9
16552	34
16553	130
16554	64
16555	1
16556	33
16557	0
16558	9
16559	84
16560	93
16561	42
16562	130
16563	64
16564	1
16565	181
16566	2
16567	237
16568	184
16569	0
16570	33
16571	201
16572	70
16573	14
16574	13
16575	9
16576	175
16577	190
16578	192
16579	54
16580	23
16581	14
16582	0
16583	201



Dieses Programm simuliert eine schnelle Rennstrecke, welche von oben auf das Rennauto (der Stern ganz unten) zukommt. Der Wagen kann durch N (links) und M (rechts) gesteuert werden. Nach einer Kollision wird die bisherige Fahrzeit angezeigt, das Spiel endet und der Wagen blinkt. Durch Drücken einer Taste beginnt ein neues Spiel. Das Programm sollte mit RUN 300 gespeichert werden. Dieses Programm sandte uns Herr Fred Ackermann aus Bruchsal ein.

```
260 IF INKEY#="" THEN GOTO 260
270 CLS
280 PRINT "          ";
290 RUN 100
300 SAVE "MONACO"
310 RUN 100
```

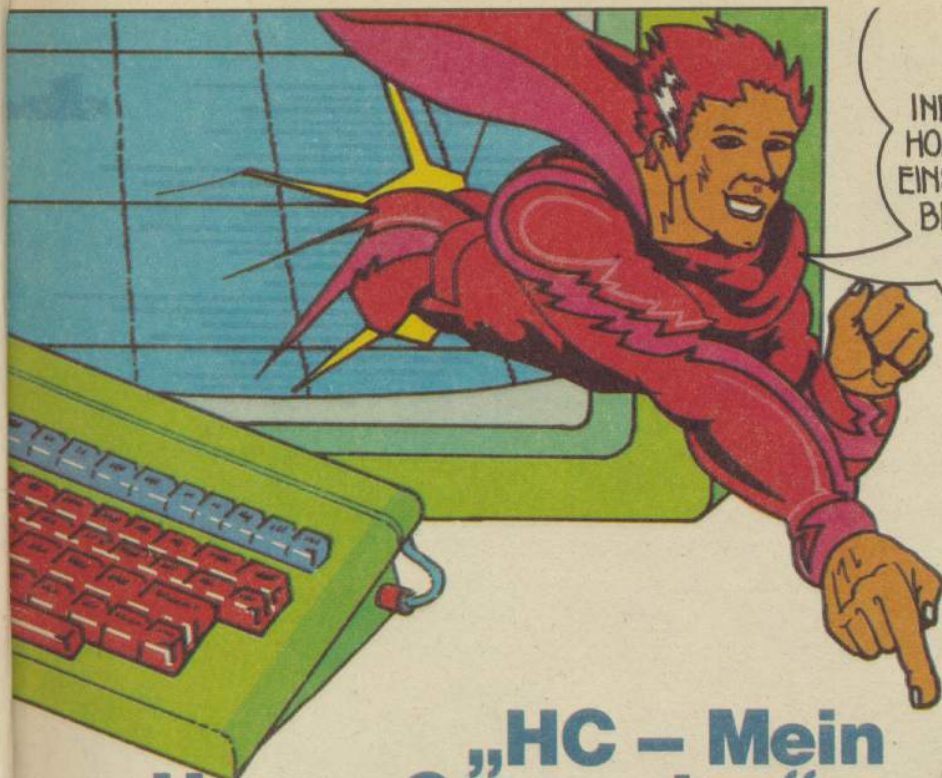
### Hexadezimal

```
4082 00      NOP
4083 00      NOP
4084 3A2540  LD A,(4025)
4087 FE7F    CP 7F
4089 2016    JR NZ,40A1
408B 3A2640  LD A,(4026)
408E FEEF    CP EF
4090 2004    JR NZ,4096
4092 3EFF    LD A,FF
4094 1806    JR 409C
4096 FEF7    CP F7
4098 2007    JR NZ,40A1
409A 3E01    LD A,01
409C 21BE40  LD HL,40BE
409F 86      ADD A,(HL)
40A0 77      LD(HL),A
40A1 2A0C40  LD HL,(400C)
40A4 01B602  LD BC,02B6
40A7 09      ADD HL,BC
40A8 228240  LD(4082),HL
40AB 012100  LD BC,0021
40AE 09      ADD HL,BC
40AF 54      LD D,H
40B0 5D      LD E,L
40B1 2A8240  LD HL,(4082)
40B4 01B502  LD BC,02B5
40B7 EDB8    LDDR
40B9 00      NOP
40BA 21C946  LD HL,46C9
40BD 0E00    LD C,00
40BF 09      ADD HL,BC
40C0 AF      XOR A
```

```
1 REM 73 BELIEB. ZEICHEN
10 LET B=A+INT (RND*3)-1
20 IF B>1 AND B<21 THEN LET A=
B
30 IF USR 16516<>0 THEN GOTO 2
00
40 PRINT AT 0,A;" ■ ■ ■ "
50 GOTO 0
100 LET A=9
105 POKE 16574,A
110 POKE 16436,255
115 POKE 16437,255
120 LET B=694+PEEK 16396+256*PE
EK 16397
125 POKE 16571,B-256*INT (B/256)
)
130 POKE 16572,INT (B/256)
135 GOTO 0
200 LET A=INT ((65535-PEEK 1643
6-256*PEEK 16437)/50)
210 PRINT AT 0,0;"FAHRZEIT: "
INT (A/60);" MIN ";A-INT (A/60)
*60;" SEC"
220 FOR A=0 TO 70
230 PRINT AT 21,PEEK 16574;" "
240 PRINT AT 21,PEEK 16574;"*"
250 NEXT A
```

ZX-USER CLUB





JEDEN MONAT DIE AKTUELLSTEN  
INFORMATIONEN FÜR IHREN  
HOME-COMPUTER - VOM SCHNELLKURSEN FÜR  
EINSTEIGER ÜBER ANWENDUNGSBEISPIELE  
BIS ZU PROGRAMMEN - FÜR WENIGER ALS  
5,-DM - MIT EINEM ABO VON  
»HC-MEIN HOME-COMPUTER«

HC ist neu! Lernen  
Sie es kostenlos  
kennen! Ein  
aktuelles Gratisheft  
ist für Sie  
reserviert!

**NEU!**

## „HC – Mein Home-Computer“ bringt im Dezember

**u.a.:** ● **Report:** Die neuen Computer-Spiele ● **18 Seiten**  
**Programme und Bauanleitungen** für Apple, Atari,  
Commodore, Dragon, Sharp, Sinclair, Tandy und TI  
● **Anwendung:** Printen und Plotten für 500,- DM ● **Test:**  
Was der Commodore 64 wirklich kann ● **Kaufberatung:**  
10 Farb-Computer unter 600,- DM im Vergleich  
● **Programmierkurs:** BASIC ● **Preis ausschreiben:**  
Home-Roboter im Wert von 10.000,- DM zu gewinnen!

Coupon bitte aus-  
schneiden und ein-  
senden an  
HC-Leserservice,  
Vogel-Verlag,  
Postfach 67 40,  
8700 Würzburg 1



Mein Home-Computer

## Coupon für 1 HC- Kennenlernheft

**Ja,** ich möchte  
„HC – Mein Home-Computer“  
kostenlos kennenlernen. Senden  
Sie mir bitte umgehend ein aktuelles Gratisheft.

Wenn mir HC gefällt und ich es im Abonnement  
weiterbeziehen will, brauche ich nichts zu  
unternehmen. Ich erhalte dann HC ab der  
nächsten Ausgabe regelmäßig frei Haus per  
Post und bezahle nur DM 55,- für 12 Hefte  
pro Jahr.

h, ich spare 5,- DM gegenüber dem Einzel-  
verkaufspreis und das Porto zahlt der Verlag.

Kurz nach dem ersten Heft meines  
Abonnements erhalte ich die praktische HC-  
Sammelbox. Sie ist im Preis für die erste  
Bezugsperiode enthalten.

Wenn mir HC nicht gefällt, teile ich dies dem  
Verlag **bis 10 Tage nach Erhalt** des Gratis-  
heftes mit und alles ist für mich erledigt.  
Ich habe also keinerlei Verpflichtung.

### Meine Anschrift:

Vorname Name

Straße

PLZ Ort

Datum Unterschrift

Wenn Sie HC im  
Abonnement  
beziehen,  
profitieren Sie  
außerdem von  
folgenden  
Vorteilen:

1. **Druckfrische  
Frei-Haus-Lieferung  
per Post.** Die  
Versandkosten trägt  
der Verlag.
2. **Ersparnis gegen-  
über dem Einzel-  
verkaufspreis.**  
Sie erhalten 12 Hefte  
zum Preis von 11.
3. **Eine praktische  
Sammelbox** ist im  
Preis für die erste  
Bezugsperiode  
enthalten.
4. **Sie versäumen  
keine Information**

HC hilft jedem, mit einem Home-Computer umzugehen. Auch Einsteiger haben bald ein  
Erfolgserlebnis.  
HC ist aber auch etwas für Fortgeschrittene - wegen seiner interessanten Programme,  
Marktübersichten, Tests aller gängigen Home-Computer und, und, und...



# PROGRAMME

# OTHELLO



**O**thello, auch bekannt unter dem Namen Reversi, ist ein Brettspiel für zwei Personen. Das Spielbrett ist eine Fläche von 8 x 8 Feldern, ähnlich wie bei Schach oder Dame. Die vier zentralen Felder werden vor Spielbeginn besetzt; auf D4 und E5 sind Steine des Spielers (invers \*), auf D5 und E4 Steine des Computers (Ø). Der Spieler hat den ersten Zug. Er muß einen Stein vertikal, horizontal oder diagonal so neben einen gegnerischen Stein legen, daß mindestens ein Stein des Gegners zwischen einem eigenen und dem neu gesetzten liegt. Alle Steine, auf die das zutrifft, dürfen umgedreht werden und haben damit die eigene Farbe. Mögliche Anfangszüge für den Spieler sind also: C5, D6, E3 oder F4. Das Spiel ist zu Ende, wenn alle Felder besetzt sind, keiner mehr einen Stein umdrehen kann, oder einer von beiden keine Steine mehr auf dem Spielfeld hat. Sieger ist derjenige, der bei Spielende die meisten Steine besetzt.

Dieses Programm für den ZX81 hält sich bis auf eine Ausnahme genau an diese Regeln. Ungültige Züge des Spielers werden erkannt und durch blinkende Anzeige deutlich gemacht. Dieses Blinken hält so lange an, bis eine Taste (außer BREAK) gedrückt wird. Dann muß ein neuer Zug

eingegeben werden. Zulässige Züge sind die oben beschriebenen (z.B. C5—NEWLINE) und außerdem Ø—NEWLINE. Die Eingabe Ø bedeutet, daß der Spieler paßt. Hier ist die einzige Möglichkeit zu mogeln, denn eigentlich muß ein Zug gemacht werden, wenn die Möglichkeit dazu besteht. In diesem Programm kann der Spieler jedoch jederzeit passen. Nach Eingabe eines korrekten Zuges wird der ZX81 den Stein an die richtige Stelle setzen, und alle geschlagenen Steine umdrehen. Das dauert ca. 15 sec. Danach berechnet der Computer im FAST-Modus seinen Zug, wofür er bis zu 50 sec. benötigt. Auf der rechten Seite des Bildschirms neben dem Spielfeld wird angezeigt, wie viele Steine jeder gerade hat, wie viele Züge bisher gemacht wurden und wer als nächster am Zug ist, sowie am Ende des Spieles den Sieger. Hat der Computer den letzten Zug gemacht, muß man eventuell noch einmal »passen«, damit der Sieger angezeigt wird.

Wenn man während des Spieles versehentlich die BREAK-Taste gedrückt hat, kann dieses dennoch fortgesetzt werden. Ein »Warmstart« erfolgt mit GOTO 444. Nach CLEAR oder RUN ist allerdings der Spielstand, der in dem Variablenfeld DIM A (10,10) gespeichert wird, gelöscht.

Da dieses Programm vollständig in BASIC geschrieben wurde, ist der ZX81 damit sicher nicht in der Lage, einen »Othello-Experten« zu schlagen. Bei vertretbarer Rechenzeit eignet sich das Programm jedoch, dieses faszinierende Spiel zu erlernen und Gewinnstrategien zu entwickeln.

Spielstarke können für sich ein zusätzliches Handicap einbauen:

RUN

BREAK—nach der Initialisierung

LET A(2,2)=28

GOTO 444

Diese zusätzlichen Eingaben im Direktmodus sichern dem Computer einen der begehrten Eckplätze (A1). Die anderen sind im Variablenfeld auf den Positionen A(2,9)—A(9,2)—A(9,9).

Wird das Programm mit RUN 7500 auf Cassette gespeichert, dann startet es nach dem Laden automatisch.

Dr. W. Steinmüller, Wien

```

1 REM ***** OTHELLO *****
2 REM * (C) 1983 *
3 REM * W. STEINMUELLER *
4 REM *****
10 LET S=151
20 LET C=28
25 FAST
30 GOSUB 7000
35 SLOW
40 DIM A(10,10)
50 FOR X=2 TO 9

```

16K

## ZX-USER CLUB



# PROGRAMME

```

60 FOR Y=2 TO 9
70 LET A(X,Y)=CODE " "
80 NEXT Y
90 NEXT X
100 LET A(5,5)=S
110 LET A(6,6)=S
120 LET A(5,6)=C
130 LET A(6,5)=C
140 LET R=1
150 FOR Z=1 TO 10
160 LET A(1,Z)=R
170 LET A(10,Z)=R
180 LET A(Z,1)=R
190 LET A(Z,10)=R
200 LET Z=R
210 LET J=2
220 GOTO 500
444 GOSUB 7000
450 GOSUB 3000
500 REM **ZUG DES SPIELERS**
510 PRINT AT 20,23;"SPIELER ";C
HR# 133
515 LET O=0
520 INPUT Z#
530 IF Z#="0" THEN LET P=0
540 IF Z#="0" THEN LET M=R
550 IF Z#="0" THEN LET N=R
560 IF Z#="0" THEN GOTO 1000
570 IF LEN Z#>2 THEN GOTO 520
575 IF CODE Z#(1)<38 OR CODE Z#
(1)>45 THEN GOTO 520
580 LET M=CODE Z#(1)-36
590 LET N=CODE Z#(2)-27
595 LET O=R
597 LET U=C
599 LET V=S
600 IF A(M,N)<>CODE " " THEN GO
TO 900
610 LET W=0
620 FOR X=M-R TO M+R STEP 2
630 FOR Y=N-R TO N+R STEP 2
640 IF A(X,Y)<>C AND A(X,N)<>C
AND A(M,Y)<>C AND A(X,Y)<>S AND
A(X,N)<>S AND A(M,Y)<>S THEN LET
W=W+1
650 NEXT Y
660 NEXT X
670 IF W=4 THEN GOTO 900
700 LET P=J
710 GOSUB 5000
720 GOSUB 3000
730 IF P=J AND A(M,N)=S THEN LE
T A(M,N)=CODE " "
740 IF P=J THEN GOTO 900
750 GOTO 1000
900 PRINT AT 20,23;"UNGUELTIG"
910 FOR X=R TO 3
920 NEXT X
930 PRINT AT 20,23;"UNGUELTIG"
940 FOR X=R TO 3
950 NEXT X
960 IF INKEY#="" THEN GOTO 900
970 GOTO 500
1000 REM **ZUG DES COMPUTERS**
1010 PRINT AT 20,23;"COMPUTER"
1020 FOR X=R TO 50
1030 NEXT X
1040 FAST
1050 LET Z=Z+R
1060 LET U=S
1070 LET V=C
1080 LET I=0
1090 FOR X=2 TO 9
1100 FOR Y=2 TO 9
1110 IF A(X,Y)<>CODE " " THEN GO
TO 1340
1120 LET Q=0


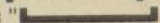
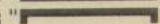
```

```

1130 FOR E=-R TO R
1140 FOR F=-R TO R
1150 LET K=0
1160 LET G=X
1170 LET H=Y
1180 IF A(G+E,H+F)<>U THEN GOTO
1230
1190 LET K=K+R
1200 LET G=G+E
1210 LET H=H+F
1220 GOTO 1180
1230 IF A(G+E,H+F)<>V THEN GOTO
1250
1240 LET Q=Q+K
1250 NEXT F
1260 NEXT E
1270 IF G=2 OR G=9 OR H=2 OR H=9
THEN LET Q=Q*2
1275 IF (G=2 OR G=9) AND (H=2 OR
H=9) THEN LET Q=Q*4
1280 IF G=3 OR G=8 OR H=3 OR H=8
THEN LET Q=Q/2
1290 IF (G=2 OR G=9) AND (H=3 OR
H=8) OR (G=3 OR G=8) AND (H=2 O
R H=9) THEN LET Q=Q/2
1295 IF (G=3 OR G=8) AND (H=3 OR
H=8) THEN LET Q=Q/2
1300 IF Q<I OR Q=0 OR (RND).3 AN
D Q=H) THEN GOTO 1340
1310 LET I=Q
1320 LET M=X
1330 LET N=Y
1340 NEXT Y
1350 NEXT X
1360 IF I=0 AND Q=0 THEN GOTO 60
00
1370 IF I=0 THEN GOTO 1390
1380 GOSUB 5000
1390 SLOW
1400 GOSUB 3000
1410 GOTO 500
3000 REM **STEINE SETZEN**
3010 LET I=0
3020 LET J=0
3030 LET G=2
3040 FOR X=3 TO 17 STEP 2
3050 LET H=2
3060 FOR Y=18 TO 4 STEP -2
3070 PRINT AT Y,X;CHR# A(G,H)
3072 IF A(G,H)=C THEN LET J=J+R
3074 IF A(G,H)=S THEN LET I=I+R
3080 LET H=H+1
3090 NEXT Y
3100 LET G=G+1
3110 NEXT X
3120 PRINT AT 14,23;Z;" "
3130 PRINT AT 4,23;J;" "
3140 PRINT AT 10,23;I;" "
3150 LET L=I
3160 RETURN
5000 REM **STEINE UMDREHEN**
5010 FOR E=-R TO R
5020 FOR F=-R TO R
5030 LET G=M
5040 LET H=N
5050 IF A(G+E,H+F)<>U THEN GOTO
5090
5060 LET G=G+E
5070 LET H=H+F
5080 GOTO 5050
5090 IF A(G+E,H+F)<>V THEN GOTO
5150
5100 LET A(G,H)=V
5110 IF M=G AND N=H THEN GOTO 51
50
5120 LET G=G-E
5130 LET H=H-F

```

```

5140 GOTO 5100
5150 NEXT F
5160 NEXT E
5170 RETURN
6000 IF L>J THEN PRINT AT 20,23;
"SPIELER "
6010 IF L<J THEN PRINT AT 20,23;
"COMPUTER"
6020 IF L=J THEN PRINT AT 20,23;
"REMIS "
6030 PRINT AT 10,23;"SIEGER : "
6040 PAUSE 4E4
6050 CLS
6060 SLOW
6070 RUN
7000 REM **SPIELBRETT**
7010 PRINT "<<<<<<<OTHELLO>>>>>>>
">"
7020 LET T=157
7030 LET Y=8
7040 LET X=128
7050 FOR A=0 TO 20
7060 PRINT AT 1,A;CHR# X;AT 2,A;
CHR# X
7070 PRINT AT 20,A;CHR# X;AT 21,
A;CHR# X
7080 NEXT A
7090 FOR A=3 TO 19
7100 PRINT AT A,0;CHR# X;AT A,1;
CHR# X
7110 PRINT AT A,19;CHR# X;AT A,2
0;CHR# X
7120 NEXT A
7130 FOR A=18 TO 4 STEP -2
7140 PRINT AT A,1;CHR# T;AT A,19
;CHR# T
7150 LET T=T+1
7160 NEXT A
7170 LET T=166
7180 FOR A=3 TO 17 STEP 2
7190 PRINT AT 2,A;CHR# T;AT 20,A
;CHR# T
7200 LET T=T+1
7210 NEXT A
7220 FOR A=3 TO 19 STEP 2
7230 FOR B=2 TO 18
7240 PRINT AT A,B;CHR# Y
7250 PRINT AT B+1,A-1;CHR# Y
7260 NEXT B
7270 NEXT A
7280 LET A=149
7290 PRINT AT 2,1;CHR# A;AT 2,19
;CHR# A
7300 PRINT AT 20,1;CHR# A;AT 20,
19;CHR# A
7310 LET A=23
7320 PRINT AT 2,A;"COMPUTER"
7330 PRINT AT 4,A;"2"
7340 PRINT AT 8,A;"SPIELER"
7350 PRINT AT 10,A;"2"
7360 PRINT AT 14,A;"1. ZUG"
7370 PRINT AT 18,A;"ES ZIEHT"
7380 PRINT AT 20,A;"SPIELER"
7390 PRINT AT 10,9;CHR# C;AT 12,
11;CHR# C
7410 PRINT AT 10,11;CHR# S;AT 12
,9;CHR# S
7420 FOR A=2 TO 20
7430 PRINT AT A,22;CHR# 5;AT A,3
1;CHR# 133
7440 NEXT A
7450 PRINT AT 1,22;""
7460 PRINT AT 21,22;""
7470 PRINT AT 17,22;""
7480 RETURN
7500 SAVE "OTHELLO"
7510 RUN

```



# PROGRAMME

## Wer zuletzt lacht, lacht am besten

Ein Denkspiel für die ganze Familie, nicht nur Winternächte, sondern ganze Sommernächte werden Sie an Ihrem Bildschirm verbringen. Auf dem Bildschirm erscheint eine zufällige Zahl. Sie und der Computer nehmen abwechselnd 1 bis 3 Zahlen von der Menge weg. Wer die letzte nimmt, hat verloren. Der Computer läßt Ihnen natürlich den Vortritt, Sie bestimmen, wie hoch die Zahl sein soll. Dann sagen Sie dem Computer, ob Sie den ersten Zug machen wollen und drücken J für Ja und N für Nein. Dann drücken Sie eine Zahl von 1 bis 3, aber nur wenn Sie am Zug sind, um dem Computer mitzuteilen, wie viel Zahlen Sie vom Haufen nehmen.

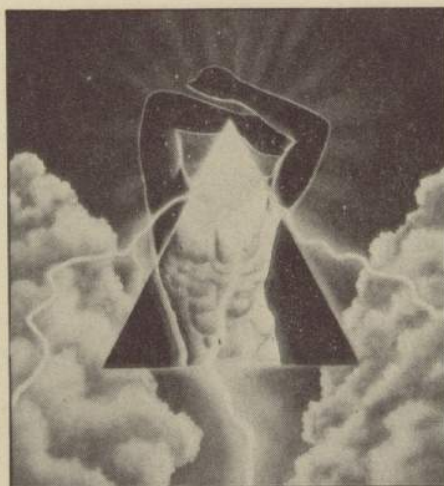


```

10 PRINT "NIMM"
15 PRINT "WIEVIELE IM HAUFEN?"
20 PRINT "WIEVIELE IM HAUFEN?"
30 INPUT N
40 PRINT "WOLLEN SIE ANFANGEN?"

50 IF INKEY$="J" THEN GOTO 100
60 IF INKEY$<>"N" THEN GOTO 50
70 LET X=N-1-INT((N-1)/4)*4
72 IF X=0 THEN LET X=INT(RND*
3+1)
74 IF X>N THEN GOTO 70
76 CLS
80 PRINT "ICH NEHME ";X;" VON
";N;" UND LASSE "
90 LET N=N-X
92 PRINT "N=";N;" "
95 IF N=0 THEN GOTO 200
100 PRINT "WIEVIELE NEHMEN SI
E?"
110 LET X=CODE INKEY$-28
120 IF X<0 OR X>3 OR X>N OR X>I
NT X THEN GOTO 110
125 IF X=0 THEN GOTO 150
130 LET N=N-X
140 IF N>0 THEN GOTO 70
150 PRINT "ICH HABE GEWONNEN"
160 STOP
200 PRINT "SIE HABEN GEWONNEN"

```



## Zahlenspiel

Geben Sie Ihrem Computer eine Zahl von eins bis unendlich ein, der Computer sucht sich in diesem Bereich eine Zahl. Diese Zahl müssen Sie erraten. Machen Sie eine Eingabe Ihrer Schätzung (zwischen 1 und Ihrem Maximum) und drücken Sie Newline. Wir hoffen, Sie werden nicht bis zum Wahnsinn getrieben!

```

5 RAND
10 LET U=1
20 LET A=U-U
30 PRINT "WELCHES MAXIMUM SOLL
ICH NEHMEN?"
40 INPUT F
50 IF F>1000 THEN PRINT "OK. V
IEL GLUCK"
55 LET B=INT(1.25*LN F)
60 PRINT "DRUECKEN SIE NEWLINE"

70 INPUT I$
80 LET H=F
90 LET L=U
100 CLS
130 PRINT "ICH HABE EINE ZAHL
GEWAHLT"
135 LET T=U
140 LET N=INT(RND*(F-.1)+U)
145 PRINT "RATEN SIE"
150 INPUT X
155 PRINT AT U,U-U;"
160 IF X<N THEN GOTO 210
165 PRINT AT U,U;X;" IST ZU HOC
H "
190 LET T=T+U
200 GOTO 145
210 IF X=N THEN GOTO 260
220 PRINT AT U,U;X;" IST ZU WEN
IG"
250 GOTO 190
260 IF T=U THEN PRINT "GENAU"
270 IF T>U THEN PRINT "GESCHAFF
T IN ";T;" VERSUCHEN"
275 IF T<B THEN PRINT "SEHR GUT"
280 PRINT "UM NOCHMAL ZU SPIELE
N,"
320 GOTO 60

```

## Sternschlag im Weltraum

Wie Steinschlag auf der Landstraße spielt sich dieses gefährliche Abenteuer im dunklen Weltraum in der dritten Galaxie ab. Ihr Computer zeigt Ihnen auf dem Bildschirm ein Meteorgebiet und Ihr Raumschiff bewegt sich automatisch von der linken Seite des Schirmes zur rechten. Als geübter Astronaut müssen Sie nun versuchen, Ihr Raumschiff an den Meteoriten mit Hilfe der Tasten Z (unten) und Q (oben) vorbeizusteuern und die rettende rechte Seite zu erreichen. Wie weit Sie gekommen sind, sagt Ihnen Ihr Bordcomputer.



```

10 LET Z=1
20 LET G=0
30 LET X=Z
40 LET Y=0
50 FOR N=0 TO 31
60 FOR M=0 TO 7
70 PRINT AT M,N;" "
80 NEXT M
90 NEXT N
100 FOR N=Z TO 25
110 PRINT AT INT(RND*7),INT(R
ND*27)+5;" "
120 NEXT N
130 PRINT AT X,Y;" "
140 LET G=G+Z
150 LET Y=Y+Z
160 IF Y=31 THEN GOTO 250
170 LET X=X+(INKEY$="Z")-(INKEY
$="Q")
180 LET X=X+(X<0)-(X>6)
190 PRINT AT X,Y;" "
200 PRINT AT X,Y+Z
210 IF PEEK(PEEK 16398+256*PEE
K 16399)<>134 THEN GOTO 130
220 PRINT G
230 PAUSE 150
240 RUN
250 PRINT AT 4,10;"ERFOLG"
260 GOTO 230

```

# ZX-USER CLUB



# PROGRAMME

## ZX 81 ROM-Routinen

Hier endlich eine Zusammenfassung der wichtigsten ROM-Routinen für den ZX 81.

### RAND USR 0:

Löscht den gesamten Rambereich bis zum Maschinenstapel

### RAND USR 681:

Im Fast-Modus angewendet, wartet es, bis eine Taste gedrückt wird und unterdrückt die Meldung von einem ausgeführten Programm

### Beispiel:

```
10 PRINT "P"
20 RAND USR 681
```

### RAND USR 836:

Im Fast-Modus aufgerufen. Nach Laden des Programmes wird mit der Fehlermeldung »C« unterbrochen

### RAND USR 963:

Löscht Rambereich bis Ramtop

### RAND USR 3086:

Scrollt den Bildschirminhalt um eine Zeile nach oben

### RAND USR 3292:

Erfüllt Stopfunktion

### Poke 16389,68:

Organisiert den Bildschirmaufbau wie beim 1 K-Betrieb

### Poke 16418,0:

Schreibt 24 Zeilen auf den Bildschirm

### Poke 16441,34:

Erzeugt eine 34 Spalte auf dem Bildschirm

```
10 POKE 16441,34
20 PRINT "P";
30 GOTO 20
```

### Poke 16510,0:

Ändert die erste Zeilennummer im Programm in eine 0 um, so daß diese Zeile nicht mehr editiert werden kann

## Farbenveränderung

10 OUT 254, (IN 254): GO TO 10

Mit dieser einzigen Zeile können Sie die Borderfarbe durch Tastendruck verändern. Durch Drücken von mehreren Tasten zur gleichen Zeit haben Sie die Möglichkeit, die Farben miteinander zu vermischen. Steht der ZX Spectrum vor einem und betrachtet man die Tasten vertikal, so ergeben sich folgende Farbmöglichkeiten:

Tasten 1 bis Caps Shift	= Gelb
0 bis Space	= Gelb
2 bis Z	= Cyan
9 bis Shift Symb.	= Cyan
3 bis T	= Magenta
8 bis N	= Magenta

### Farbkombinationen:

Gelb und Cyan = Grün  
Magenta und Gelb = Rot  
usw.

Wir wünschen Ihnen viel Spaß!!

**neu**

Ein **Funkschau**-Sonderheft

**Franzis'**

# Zaubern mit dem ZX 81

Das ZX-81-Sonderheft macht diesen Computer transparenter und hilft, die Scheu vor der unbekannten Technik zu nehmen. Wer das Heft liest, entdeckt völlig neue Seiten des ZX 81 und kann ihn besser nutzen.

Ein Teil der Beiträge wurde bereits in der FUNKSCHAU veröffentlicht. Das ZX-81-Sonderheft ist aktiven Hobbyisten auf den Leib geschrieben. Das Programmieren in Maschinensprache – für Einsteiger normalerweise eine unlösbare Aufgabe – wird mit diesem Sonderheft leicht verständlich und durch einen kompletten Lehrgang sehr praxisnah. Bauanleitungen für eine 6 KByte RAM-Erweiterung, für eine Ein-/Ausgabe-Schnittstelle und für eine Schaltung zur Tonerzeugung wenden sich an Leser, die mit dem Lötkolben umzugehen verstehen. Besprechungen käuflich erwerbbarer Software sorgen außerdem dafür, daß niemand die Katze im Sack kaufen muß.



## BEZUGS-MÖGLICHKEITEN

bei allen Bahnhofsbuchhandlungen, beim Elektronik-Fachhandel, bei größeren Zeitschriftenverkaufsstellen, in Buchhandlungen oder direkt beim Franzis-Verlag gegen

- Voreinzahlung von 16,20 DM (14,20 DM + 2,- DM Porto) auf unser Postscheckkonto München Nr. 813 75-809 mit Hinweis „Zaubern mit dem ZX 81“ oder

- Zusendung eines Schecks (16,20 DM)

**Franzis'**

Franzis-Verlag, Karlstraße 37,  
8000 München 2,  
Telefon 0 89/51 17-2 39/-3 80

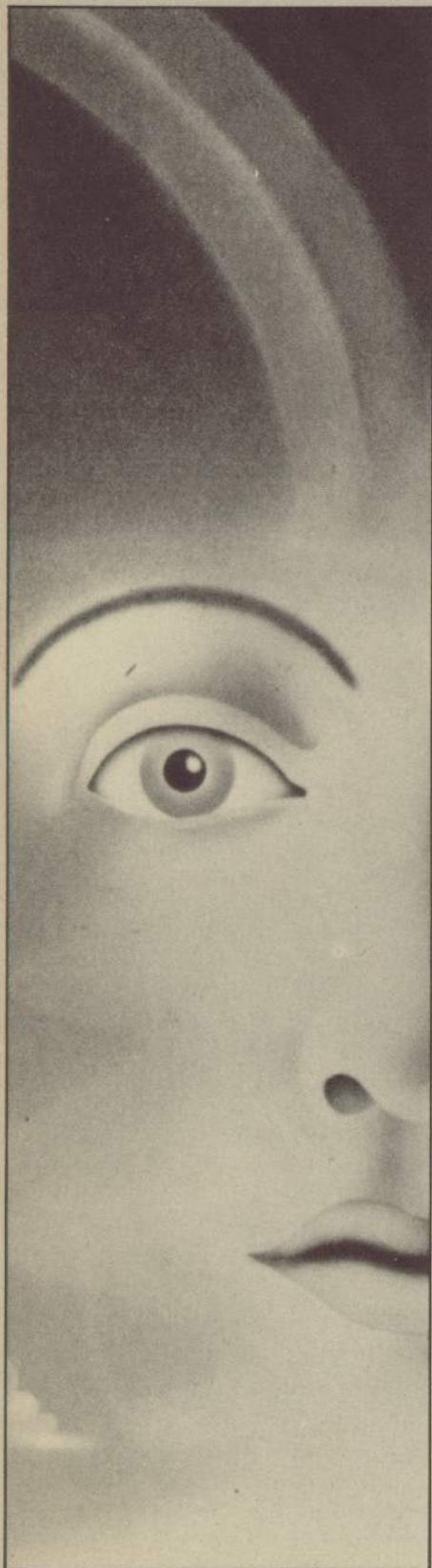
In der Schweiz:  
Verlag Thali AG, CH-6285 Hitzkirch.

In Österreich: Fachbuch Center Erb,  
Amerlingstraße 1, A-1061 Wien.



## Biorhythmus

16K mit Drucker



Durch Beobachtungen wurde festgestellt, daß es im Leben des Menschen verschiedene Zyklen gibt, die einen erheblichen Einfluß auf seine körperliche und geistige Leistungsfähigkeit sowie auf seine seelische Stabilität ausüben. Es wurden bisher 3 solcher Zyklen entdeckt.

1. den seelischen Zyklus mit einer Dauer von 28 Tagen
2. den geistigen Zyklus mit einer Dauer von 33 Tagen
3. den körperlichen Zyklus mit einer Dauer von 23 Tagen

Solange sich die Werte des Zyklus im positiven Bereich bewegen, ist die Leistungsfähigkeit, bzw. die Stabilität hoch, während sie in der Zeit mit negativen Werten etwas nachläßt. Kritisch sind die Tage, in denen die jeweilige Kurve vom positiven in den negativen Bereich wechselt. An diesen Tagen besteht verstärkte Unfallgefahr und bei seelisch ohnehin labilen Menschen häufen sich an diesen Tagen die Kurzschlußreaktionen.

Das nachstehende Programm druckt, unter genauer Beachtung der im gregorianischen Kalender vorgesehenen Schalttage, eine Biorhythmuskurve für

einen beliebigen Zeitraum auf dem ZX-Printer aus. Nach dem Starten durch Eingabe von RUN verlangt das Programm die Eingabe des Geburtstages und von Vornamen und Namen sowie die Angabe des gewünschten Zeitraumes.

Wichtig ist die Beachtung von zwei Punkten:

1. Die Jahreszahlen müssen **immer vierstellig** eingegeben werden (z. B. 1983 und nicht 83!!!)
2. Aus programmtechnischen Gründen sollte das Ende des gewünschten Zeitraumes **nicht der Letzte eines Monats** sein, da sonst der Rechner das Programm nicht beendet.

Beim Fehlen eines Druckers kann man sich die Kurven auch auf dem Bildschirm anzeigen lassen. Hierzu muß lediglich in den Zeilen 695 bis 750, 810, 870, 940, 2100 bis 2235 der Befehl LPRINT durch PRINT ersetzt werden. Vorher empfiehlt sich die Einfügung von jeweils einem SCROLL-Befehl, um ein Aussteigen des Programms mit Fehlermeldung 5 zu vermeiden.

K. Sütterlin, München

```

10 DIM A$(1,32)
20 LET B$=""
30 DIM M$(12,3)
40 LET M$(1)="0"
50 LET M$(2)="31"
60 LET M$(3)="59"
70 LET M$(4)="90"
80 LET M$(5)="120"
90 LET M$(6)="151"
100 LET M$(7)="181"
110 LET M$(8)="212"
120 LET M$(9)="243"
130 LET M$(10)="273"
140 LET M$(11)="304"
150 LET M$(12)="334"
155 PRINT AT 5,8:"BIORHYTHMUS"
160 PRINT AT 7,0:"GEBURTSTAG?<T
T N/L MM N/L JJJJ>"
170 INPUT T
172 LET D=T
180 INPUT M
182 LET E=M
190 INPUT J
192 LET F=J
200 PRINT T;".";M;".";J
205 GOSUB 600
210 GOSUB 450
220 LET A=S
225 GOSUB 2000
410 LET T=D
412 LET M=E
414 LET J=F
416 CLS
418 GOTO 200
420 STOP

```

```

430 LET N=100*SIN (2*PI*(B-A)/L)
440 RETURN
445 REM EINFUEGEN SCHALT TAG
450 LET Z=0
460 IF J/4<>INT (J/4) OR J/4=INT
(J/4) AND J/400=INT (J/400) TH
EN GOTO 490
470 IF M<2 THEN GOTO 490
480 LET Z=1
490 LET Z=Z+(J-1)*365+INT ((J-1)
/4)
500 LET C=VAL (M$(M))
510 LET S=Z+C+T
520 RETURN
600 PRINT "EINGABE VORNAME U. N
AME"
610 INPUT N$
620 RETURN
690 FAST
695 LPRINT TAB 2;"
700 LPRINT TAB 1;" B I O R H
Y T H M U S "
705 LPRINT TAB 2;"
710 LPRINT
715 LPRINT
720 LPRINT "FUER "N$
725 LPRINT
727 LPRINT
730 LPRINT "GEBOREN AM "D;"."
E;".";F
740 LPRINT
750 LPRINT

```



# PROGRAMME

```

760 LET A$(1)(1 TO 18)="FUER DI
E ZEIT VOM "
762 LET C#=STR$ T1
765 IF T1>9 THEN LET A$(1)(19 T
0 20)=C#
770 IF T1<10 THEN LET A$(1)(20)
=C#
775 LET A$(1)(21)="."
780 LET D#=STR$ M1
785 IF M1>9 THEN LET A$(1)(22 T
0 23)=D#
790 IF M1<10 THEN LET A$(1)(23)
=D#
800 LET A$(1)(24)="."
803 LET E#=STR$ J1
805 LET A$(1)(25 TO 28)=E#
810 LPRINT A$(1)
815 LET A$(1)=B#
820 LET A$(1)(1 TO 18)="
BIS "
823 LET F#=STR$ T2
825 IF T2>9 THEN LET A$(1)(19 T
0 20)=F#
830 IF T2<10 THEN LET A$(1)(20)
=F#
835 LET A$(1)(21)="."
837 LET G#=STR$ M2
840 IF M2>9 THEN LET A$(1)(22 T
0 23)=G#
845 IF M2<10 THEN LET A$(1)(23)
=G#
850 LET A$(1)(24)="."
855 LET H#=STR$ J2
860 LET A$(1)(25 TO 28)=H#
870 LPRINT A$(1)
880 LET A$(1)=B#
940 LPRINT "
"
950 RETURN
2000 REM GRAFISCHE DARSTELLUNG
2005 PRINT
2010 PRINT "VOM? (T,M,J) UND JEW
EILS N/L"
2020 INPUT T1
2030 INPUT M1
2040 INPUT J1
2050 PRINT "BIS ? (WIE OBEN)"
2060 INPUT T2
2070 INPUT M2
2080 INPUT J2
2090 GOSUB 690
2100 LPRINT
2110 LPRINT
2117 LPRINT TAB 7;"NEGATIV";TAB
16;"0";TAB 20;"POSITIV"
2120 LPRINT "TAG";TAB 4;"-109876
54321012345678910+"
2125 LPRINT "-----"
2130 GOSUB 3000
2140 SLOW

```

```

2145 LPRINT "-----"
2150 LPRINT "TAG";TAB 4;"-109876
54321012345678910+"
2160 LPRINT
2170 LPRINT
2180 LPRINT
2190 LPRINT "ZEICHENERKLAERUNG:"
2200 LPRINT "-----"
2210 LPRINT "S=SEELISCH"
2220 LPRINT "K=KOERPERLICH"
2230 LPRINT "G=GEISTIG"
2235 LPRINT
2240 STOP
3000 LET T=T1
3010 LET M=M1
3020 LET J=J1
3030 GOSUB 450
3040 LET B=S
3050 LET L=28
3060 GOSUB 430
3070 LET X=INT (N/10+0.5)
3080 LET L=33
3090 GOSUB 430
3100 LET Y=INT (N/10+.5)
3110 LET L=23
3120 GOSUB 430
3130 LET Z=INT (N/10+0.5)
3140 GOSUB 4000
3150 LET T1=T1+1
3160 GOSUB 5000
3170 IF T1=T2+1 AND M1=M2 AND J1
=J2 THEN RETURN
3180 GOTO 3000
4000 LET T#=STR$ T1
4010 IF LEN T#=2 THEN LET A$(1)(
1 TO 2)=T#
4015 IF LEN T#=1 THEN LET A$(1)(
2)=T#
4020 LET A$(1)(3)="."
4024 LET O#=STR$ M1
4026 IF LEN O#=2 THEN LET A$(1)(
4 TO 5)=O#
4028 IF LEN O#=1 THEN LET A$(1)(
5)=O#
4030 LET A$(1)(17)="."
4035 LET A$(1)(6)="."
4040 LET A$(1)(17+X)="S"
4050 LET A$(1)(17+Y)="G"
4060 LET A$(1)(17+Z)="K"
4070 LPRINT A$(1)
4080 LET A$(1)=B#
4090 RETURN
5000 DIM Q(12)
5010 LET Q(1)=31
5015 REM SCHALT TAG
5020 IF J/4=INT (J/4) AND J/400<
>INT (J/400) THEN LET Q(2)=29
5025 IF J/4<>INT (J/4) OR J/400=
INT (J/400) THEN LET Q(2)=28
5030 LET Q(3)=31

```

```

5040 LET Q(4)=30
5050 LET Q(5)=31
5060 LET Q(6)=30
5070 LET Q(7)=31
5080 LET Q(8)=31
5090 LET Q(9)=30
5100 LET Q(10)=31
5110 LET Q(11)=30
5120 LET Q(12)=31
5125 IF T1>Q(M1) THEN GOSUB 5200
5140 RETURN
5200 LET T1=1
5210 LET M1=M1+1
5220 IF M1>12 THEN GOSUB 5300
5230 RETURN
5300 LET M1=1
5310 LET J1=J1+1
5320 RETURN

```

## BIORHYTHMUS

FUER MICHAELA MAY

GEBOREN AM 14.7.1963

FUER DIE ZEIT VOM 1. 1.1993  
BIS 15. 1.1993

NEGATIV 0 POSITIV  
TAG -10987654321012345678910+

1.	1.	S	G	K
2.	1.	GS		K
3.	1.	G	S	K
4.	1.	G	S	K
5.	1.	G		S K
6.	1.	G		K S
7.	1.	G	K	S
8.	1.	G	K	S
9.	1.	G	K	S
10.	1.	G	K	S
11.	1.	G	K	S
12.	1.	GK		S
13.	1.	KG		S
14.	1.	K	G	S
15.	1.	K	G	S

TAG -10987654321012345678910+

## ZEICHENERKLAERUNG:

S=SEELISCH  
K=KOERPERLICH  
G=GEISTIG

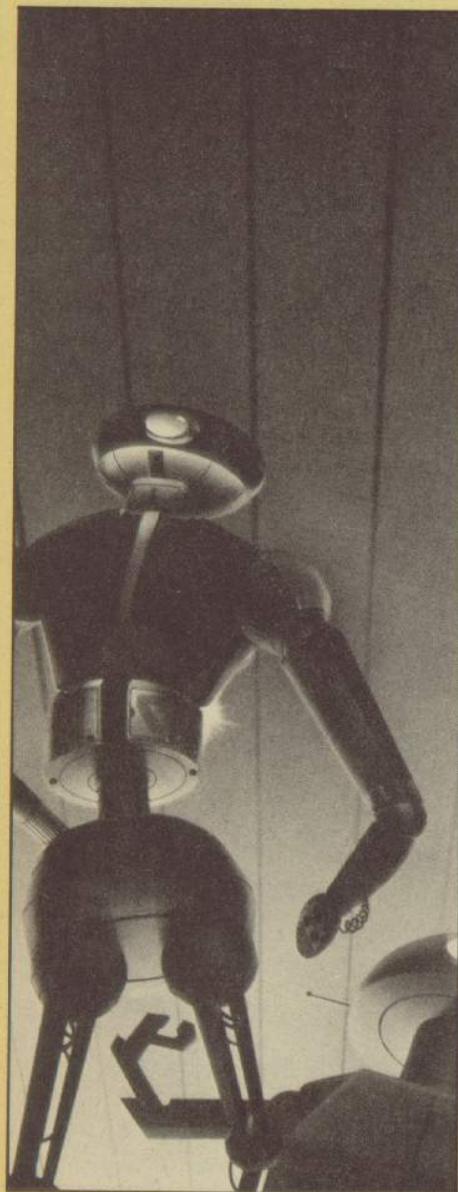
16K



# ZX-USER CLUB



## Adress Verwaltung



Adressverwaltung — mit schnellem Zugriff auf die Daten von bis zu 100 Personen. Außer den normalen Daten wie Name, Adresse und Telefon-Nummer können Sie hier auch zusätzliche Informationen speichern wie etwa Geburtstag, Beruf etc. Sie können die Eintragung nach verschiedenen Kriterien suchen: selbst der Vorname, der Straßennamen oder auch nur die ersten Buchstaben genügen. Eingegebene Daten können korrigiert oder gelöscht, neue Eintragungen hinzugefügt werden.

```

10 REM ** ADRESSVERWALTUNG **
70 SLOW
80 LET M=100
100 DIM A$(M,20)
110 DIM B$(M,22)
120 DIM C$(M,18)
130 DIM D$(M,14)
140 DIM E$(M,10)
150 DIM F$(M,5)
160 DIM G$(M,8)
170 LET T1=29
210 DIM Z$(14)
220 DIM A(7)
230 LET Z$="20221814100508"
240 FOR N=1 TO 7
250 LET A1=VAL (Z$(2*N-1 TO 2*N))
260 LET A(N)=A1
270 NEXT N
500 REM MENUE AUSWAHL
505 CLS
506 PRINT "          ADRESSVERWALTUNG"
510 PRINT "          SIE HABEN FOLGE"
515 PRINT "          NDE AUSWAHL"
520 PRINT "          1 = EINTRAG SUCHEN"
525 PRINT "          2 = EINTRAGEN, LOESCHEN, AENDERN"
535 PRINT "          3 = SPEICHERN AUF CASSETTE"
540 PRINT "          4 = EINGABE DES CODES"
550 INPUT M$
555 IF M$="1" AND M$="3" THEN GOTO 570
560 PRINT AT 21,0;"UNERLAUBTER CODE - WIEDERHOLEN"
565 GOTO 550
570 IF VAL M$<>INT VAL M$ THEN GOTO 560
575 IF VAL M$=3 THEN GOTO 3000
580 GOSUB VAL M$*1000
590 GOTO 500
1000 REM SUCHEN NACH EINTRAEGEN
1005 CLS
1010 PRINT "          INFORMATIONEN ZUM ABRUFEN"
1015 PRINT "          "
1020 PRINT "NAME";TAB T1;"(A)"
1025 PRINT "STRASSE";TAB T1;"(B)"
1030 PRINT "WOHNORT";TAB T1;"(C)"
1035 PRINT "INFO-ZIFFER 1";TAB T1;"(D)"
1040 PRINT "INFO-ZIFFER 2";TAB T1;"(E)"
1045 PRINT "VORWAHL";TAB T1;"(F)"
1050 PRINT "TEL.NUMMER";TAB T1;"(G)"
1055 PRINT "          "
1060 PRINT "ES WIRD MEIST NUR NACH DEM NACH-NAMEN (CODE A) ABRUFEN, ANHANDER TABELLE SEHEN SIE, DASS SIE NACH MEHREREN INDIKATOREN (A-G) ABRUFEN KOENNEN"
1065 PRINT "          "
1070 PRINT AT 18,0;"EINGABE DES SUCHCODES"
1080 INPUT N$
1081 IF N$="" THEN LET N$="A"
1083 PRINT AT 18,23;">>"/N$;"<<"
1085 IF N$="A" AND N$="G" THEN GOTO 1100
1090 PRINT AT 20,5;"UNERLAUBTER CODE " " EINGABE WIEDERHOLEN"
1095 GOTO 1080
1100 PRINT AT 20,0;"NAMEN ODER NUMMER EINGEBEN"
1105 INPUT O$
1106 IF O$="" THEN LET O$=" "
1107 LET L=LEN O$
1108 CLS
1109 PRINT "          SUCHE"
1110 GOSUB 1200
1115 CLS
1120 IF N=0 THEN GOTO 1125
1121 GOSUB 1140
1122 GOTO 1150
1125 PRINT AT 12,0;"EINTRAG NICHT GEFUNDEN"
1130 PRINT AT 14,0;"TASTE NEWLINE ZUR FORTSETZUNG DRUECKEN"
1133 INPUT P$
1136 RETURN
1140 PRINT A(N),B(N),C(N)
1141 PRINT "D(N)";E(N)
1142 PRINT "F(N)";G(N)
1147 RETURN
1150 PRINT "          "
1155 PRINT "          "
1160 PRINT "WEITERSUCHEN";TAB T1;"(A)"
1161 PRINT "ENDE DER SUCHE";TAB T1;"(B)"
1162 PRINT "KOPIEREN";TAB T1;"(C)"
1163 PRINT "          "
1165 INPUT Q$
1166 IF Q$="C" THEN GOSUB 4000
1167 IF Q$="A" THEN RETURN
1170 IF Q$="A" THEN FAST
1171 GOSUB 1210
1172 SLOW
1175 GOTO 1115
1200 FOR N=1 TO M
1202 LET Q=1
1205 GOSUB 1300+10*(CODE N-1)
1206 IF Q=0 THEN RETURN
1210 NEXT N
1211 SLOW
1215 LET N=0
1220 RETURN
1300 LET Q=NOT (A(N)*(1 TO L)=O$)
1305 RETURN
1310 LET Q=NOT (B(N)*(1 TO L)=O$)
1315 RETURN
1320 LET Q=NOT (C(N)*(1 TO L)=O$)
1325 RETURN
1330 LET Q=NOT (D(N)*(1 TO L)=O$)
1335 RETURN
1340 LET Q=NOT (E(N)*(1 TO L)=O$)
1345 RETURN
1350 LET Q=NOT (F(N)*(1 TO L)=O$)
1355 RETURN
1360 LET Q=NOT (G(N)*(1 TO L)=O$)
1365 RETURN
2000 CLS

```



# PROGRAMME

## Gehaltsabrechnung



Der Bildschirm fragt die nächste benötigte Einkommens- und Abzugssumme ab, so daß nichts vergessen werden kann.

Je nach Tarifvertrag kann das Programm natürlich modifiziert werden, und Schichtzulagen usw. berücksichtigt werden. Die drei Sozialversicherungen kann man in einem Betrag zusammenfassen. Mehr Einzelposten dürfen es aber nicht werden, da der Bildschirm voll ist.

Mit dem 16K-RAM ist der ZX81 voll ausgelastet. Mit größerer Kapazität läßt sich die Sache natürlich erweitern. Man könnte die Einzelposten speichern und am Schluß errechnen und ausdrucken lassen, wieviel Lohnsteuer und Sozialversicherung für das ganze Personal insgesamt an Finanzamt und Sozialversicherung überwiesen werden müssen.

Aus Berlin kommt dieses interessante Programm von H. Rothauscher für die Kaufleute unter den ZXern.

Mit diesem Programm kann in einem kleinen Unternehmen die Gehaltsabrechnung für die Mitarbeiter erarbeitet und die Lohnstreifen ausgedruckt werden.

```

2001 PRINT AT 12,5;"J = NEUER EI
NTRAG";AT 14,5;"N = SUCHEN EINTR
AG"
2002 IF INKEY#="J" THEN GOTO 2006
2003 IF INKEY#="N" THEN GOTO 2012
2005 IF INKEY#<>"J" OR INKEY#<>"
N" THEN GOTO 2002
2006 LET N#="A"
2007 LET O#=" "
2008 LET L=LEN O#
2009 GOSUB 1200
2010 SLOW
2011 GOTO 2018
2012 CLS
2015 GOSUB 1010
2018 IF N=0 THEN RETURN
2020 FOR O=0 TO 6
2022 CLS
2023 GOSUB 1140
2025 PRINT "..."
2026 GOSUB 2105+20*O
2030 PRINT "..."
2035 PRINT "J ODER N (X=KEINE
RENDERUNG)"
2040 PAUSE 40000
2041 LET P#="INKEY#"
2042 IF P#="X" THEN GOTO 2062
2045 IF P#<>"J" THEN GOTO 2060
2050 PRINT "NEUE EINGABE BITTE"
2051 PRINT "BIS ZU ";A(O+1);" ST
ELLEN"
2052 INPUT P#
2053 GOSUB 2100+20*O
2060 NEXT O
2062 CLS
2066 GOSUB 1140
2070 PRINT "..."
2075 PRINT "NEWLINE = WEIT
ERMACHEN"
2080 INPUT P#
2085 GOTO 500
2100 LET A#(N)=P#
2105 PRINT "NAME";
2110 RETURN
2120 LET B#(N)=P#
2125 PRINT "ADRESSE"
2130 RETURN
2140 LET C#(N)=P#
2145 PRINT "WOHNORT"
2150 RETURN
2160 LET D#(N)=P#
2165 PRINT "INFO-ZIFFER 1"
2170 RETURN
2180 LET E#(N)=P#
2185 PRINT "INFO-ZIFFER 2"
2190 RETURN
2200 LET F#(N)=P#
2205 PRINT "VORWAHL"
2210 RETURN
2220 LET G#(N)=P#
2225 PRINT "TEL. NUMMER"
2230 RETURN
3000 CLS
3010 PRINT "CASSETTENRECORDER EI
NSCHALTEN. DANN NEWLINE DRUECKE
N"
3015 INPUT R#
3020 LET S#="ADR. VERW."
3025 SAVE S#
3030 GOTO 500
4000 FOR I=9 TO 21
4020 PRINT AT I,0;" "
4030 NEXT I
4090 COPY
4100 RETURN
    
```

```

10 PRINT "GEHALTSABRECHNUNG MA
I 1983"
20 PRINT "NAME/STEUERKLASSE/RE
L/VERS."
30 INPUT A#
40 PRINT A#
45 PRINT "DM DM"
50 PRINT "GRUNDGEHALT"
60 INPUT G
65 LET Y=LEN STR# INT G
70 PRINT AT 4,18-Y;G
80 PRINT "UEBERSTUNDEN"
90 INPUT U
95 LET Y=LEN STR# INT U
100 PRINT AT 5,18-Y;U
110 PRINT "VERM.W.L."
120 LET V=26
130 PRINT AT 6,16;V
140 PRINT "STEUERPFL.,"
145 LET Y=LEN STR# INT (G+U+V)
146 PRINT AT 7,27-Y;G+U+V
150 PRINT "SONN./FEIERTAG"
160 INPUT S
161 LET Y=LEN STR# INT S
162 PRINT AT 8,18-Y;S
165 PRINT "SONST STEUERFR."
170 INPUT T
175 LET Y=LEN STR# INT T
180 PRINT AT 9,18-Y;T
190 PRINT "BRUTTO"
193 LET Y=LEN STR# INT (G+U+V+S
+T)
195 PRINT AT 10,27-Y;G+U+V+S+T
200 PRINT "-----"
205 PRINT "ABZUEGE"
210 PRINT "LOHNSTEUER"
220 INPUT L
225 LET Y=LEN STR# INT L
230 PRINT AT 13,18-Y;L
240 PRINT "KIRCHENSTEUER"
250 INPUT K
    
```

```

255 LET Y=LEN STR# INT K
260 PRINT AT 14,18-Y;K
270 PRINT "ARB.N.ANT. KV"
280 INPUT O
285 LET Y=LEN STR# INT O
290 PRINT AT 15,18-Y;O
300 PRINT "ARB.N.ANT. RV"
310 INPUT R
315 LET Y=LEN STR# INT R
320 PRINT AT 16,18-Y;R
330 PRINT "ARB.N.ANT. AV"
340 INPUT A
345 LET Y=LEN STR# INT A
350 PRINT AT 17,18-Y;A
360 PRINT AT 18,23;"-----"
370 PRINT "AUSZAHLUNG"
375 LET Y=LEN STR# INT (G+U+S+T
-L-K-O-R-A)
380 PRINT AT 19,27-Y;G+U+V+S+T-
L-K-O-R-A
390 PRINT AT 20,23;"=====
    
```

16K

```

GEHALTSABRECHNUNG FEB. 1983
NAME/STEUERKLASSE/REL/VERS.
EMIL SCHULZE/3,2/EV/AOK
DM DM
GRUNDGEHALT 1848.11
UEBERSTUNDEN 0
VERM.W.L. 26
STEUERPFL. (1874.11)
SONN./FEIERTAG 54.5
SONST STEUERFR.100
BRUTTO 2029.61
-----
ABZUEGE
LOHNSTEUER 213.3
KIRCHENSTEUER 7.46
ARB.N.ANT. KV 107.5
ARB.N.ANT. RV 182.54
ARB.N.ANT. AV 20.28
-----
AUSZAHLUNG 1497.53
=====
    
```

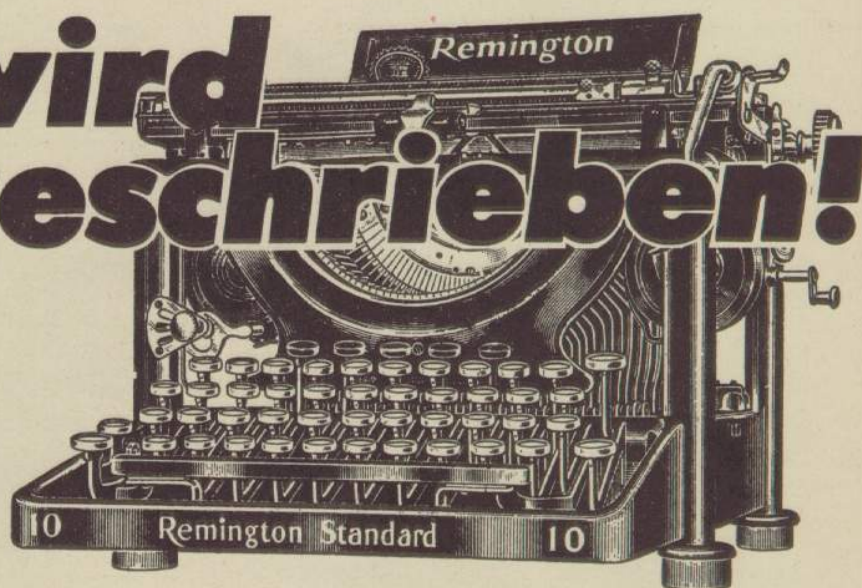
# ZX-USER CLUB



# PROGRAMME

## Jetzt wird Groß-geschrieben!

Nun endlich können Sie Ihren Kunden jedes SONDERANGEBOT per Computer mitteilen. Das Gerät ins Schaufenster oder ins Küchenfenster, der Nachbar wird's schon lesen können, I LOVE YOU endlich in der richtigen Größe. Unser Leser M. Hauser aus Köln hat bestimmt noch ganz andere Anwendungsmöglichkeiten für Großschreiber. Danke!



```
1 REM ....131 BELIEB. ZEICHEN
10 CLEAR
20 DIM A$(40)
30 INPUT A$
40 RAND USR 16514
50 RUN
```

16k

4082 00	NOP	40CD CB03	RLC E	16532	64
4083 2A0C40	LD HL,(400C)	40CF 300D	JR NC,40DE	16533	131
4086 23	INC HL	40D1 CB40	BIT 0,B	16534	95
4087 0600	LD B,00	40D3 2004	JR NZ,40D9	16535	62
4089 0E00	LD C,00	40D5 CBCE	SET 1,(HL)	16536	0
408B 3E38	LD A,38	40D7 1805	JR 40DE	16537	138
408D A0	AND B	40D9 7E	LD A,(HL)	16538	87
408E 81	ADD A,C	40DA 2F	CPL	16539	26
408F C606	ADD A,06	40DB E687	AND 87	16540	95
4091 ED5B1040	LD DE,(4010)	40DD 77	LDX(HL),A	16541	22
4095 83	ADD A,E	40DE 23	INC HL	16542	131
4096 5F	LD E,A	40DF 14	INC D	16543	203
4097 3E00	LD A,00	40E0 CB52	BIT 2,D	16544	35
4099 8A	ADC A,D	40E2 CABB40	JP Z,40BB	16545	203
409A 57	LD D,A	40E5 0C	INC C	16546	18
409B 1A	LD A,(DE)	40E6 CB59	BIT 3,C	16547	203
409C 5F	LD E,A	40E8 CA8B40	JP Z,40BB	16548	18
409D 1683	LD D,83	40EB 23	INC HL	16549	203
409F CB23	SLA E	40EC 04	INC B	16550	35
40A1 CB12	RL D	40ED 78	LD A,B	16551	203
40A3 CB12	RL D	40EE E607	AND 07	16552	35
40A5 CB23	SLA E	40F0 2007	JR NZ,40F9	16553	203
40A7 CB23	SLA E	40F2 78	LD A,B	16554	18
40A9 CB12	RL D	40F3 C6D8	ADD A,D8	16555	120
40AB 78	LD A,B	40F5 C28940	JP NZ,4089	16556	230
40AC E607	AND 07	40F8 C9	RET	16557	7
40AE 83	ADD A,E	40F9 CB40	BIT 0,B	16558	131
40AF 5F	LD E,A	40FB CB40	BIT 0,B	16559	95
40B0 CB52	BIT 2,D	40FB CA8940	JP Z,4089	16560	203
40B2 CBD2	SET 2,D	40FE 11DFFF	LD DE,FFDF	16561	82
40B4 1A	LD A,(DE)	4101 19	ADD HL,DE	16562	203
40B5 2801	JR Z,40BB	4102 C38940	JP 4089	16563	210
40B7 2F	CPL	4105 C9	RET	16564	26
40B8 5F	LD E,A	16514	0	16565	40
40B9 1600	LD D,00	16515	42	16566	1
40BB CB03	RLC E	16516	12	16567	47
40BD CB40	BIT 0,B	16517	64	16568	95
40BF 2002	JR NZ,40C3	16518	35	16569	22
40C1 3600	LD(HL),00	16519	6	16570	0
40C3 3008	JR NC,40CD	16520	0	16571	203
40C5 2004	JR NZ,40CB	16521	14	16572	3
40C7 CBC6	SET 0,(HL)	16522	0	16573	203
40C9 1802	JR 40CD	16523	62	16574	64
40CB CBD6	SET 2,(HL)	16524	56	16575	32
		16525	160	16576	2
		16526	129	16577	54
		16527	198	16578	0
		16528	6	16579	48
		16529	237	16580	8
		16530	91	16581	32
		16531	16	16582	4

### ZX-USER CLUB



# PROGRAMME

16583	203
16584	198
16585	24
16586	2
16587	203
16588	214
16589	203
16590	3
16591	48
16592	13
16593	203
16594	64
16595	32
16596	4
16597	203
16598	206
16599	24
16600	5
16601	126
16602	47
16603	230
16604	135
16605	119
16606	35
16607	20
16608	203
16609	82
16610	202
16611	187
16612	64
16613	12
16614	203
16615	89
16616	202
16617	139
16618	64
16619	35
16620	4
16621	120
16622	230
16623	7
16624	32
16625	7
16626	120
16627	198
16628	216
16629	194
16630	137
16631	64
16632	201
16633	203
16634	64
16635	202
16636	137
16637	64
16638	17
16639	223
16640	255
16641	25
16642	195
16643	137
16644	64
16645	201

```

12345678
90ABCDEF
GHIJKLMN
OPQRSTUU
WXYZ*+<>

```

## BASIC-Renumber-Programm

In der vorherigen Ausgabe haben wir Ihnen ein Renumberprogramm in Maschinensprache vorgestellt. Heute veröffentlichen wir ein BASIC-Renumberprogramm mit GOTO und GOSUB-Änderung, das uns Herr G. Hofmann zuschickte. Dieses Programm ist natürlich um einiges langsamer, soll aber demonstrieren, daß es auch in BASIC seinen Zweck erfüllt.

Das letzte Byte dieses Programmes (das NEWLINE nach STOP) steht an der Maschinenadresse  $16512 + 5 \cdot 256 + 5 = 17797$ .

Dieses Programm numeriert ein nachfolgendes Programm bei Bedarf in ganzen Zehnersprüngen neu, und dabei auch die GOTO-, GOSUB- und LIST-Adressen.

Das Quellprogramm kann ab der Zeilenr. 63 geschrieben werden. Bei RUN wird die Zeilenr. abgefragt, von der an (inclusive) das Programm neu numeriert werden soll. Das Quellprogramm kann immer mit »RUN 63« getestet werden. Die erste neu zu numerierende Zeile wird auf eine ganze Zehnerzahl gesetzt, z. B.: 63 auf 70, 80 auf 80, da im ersten Fall die Zeilen 60, 61, 62 belegt sein können.

```

1 REM
2 DIM W$(21,10)
3 LET A=PEEK 16396+256*PEEK 1
6397-5
4 PRINT "ANFANGSZEILE=?"
5 INPUT B
6 CLS
7 LET C=0
8 LET K=1
9 LET D=B
10 LET B=10*INT (B/10)
11 IF D=B THEN LET B=B-10
12 FOR W=17801 TO A
13 LET E=PEEK W
14 IF E=236 OR E=237 OR E=240
THEN GOSUB 44
15 NEXT W
16 FOR W=17796 TO A
17 IF PEEK W=118 THEN GOSUB 21
18 NEXT W
19 LIST D
20 STOP
21 LET F=256*PEEK (W+1)+PEEK (
W+2)
22 LET W=W+4
23 IF D>F THEN RETURN
24 LET G=B+10
25 POKE W-3,INT (G/256)
26 POKE W-2,G-256*INT (G/256)
27 LET B=G
28 FOR V=2 TO 21
29 IF W$(V,1)=" " THEN GOTO 33
30 LET I=VAL W$(V,6 TO 9)

```

Die Sprungadressen werden folgendermaßen umgewandelt, z. B.:

```

GOTO X+100
→ GOTO X+100
(muß korrigiert werden)
GOTO 100+X*100
→ GOTO 130+X=100
(wird nicht gelesen)
GOSUB 100
→ GOSUB 130
(ohne Problem)

```

Wenn die Zeilenr. sich durch das Renumberprogramm von zwei- auf dreistellige Zahlen, von drei- auf vierstellige Zahlen, oder umgekehrt verändern, z. B.:

```

LIST 990
→ LIST 110

```

Das bedeutet: Es muß eine 0 angehängt werden und die invertierten Zahlen in normale Zahlen zurückgebracht werden.

Alle invertierten Zeichen laufen nicht! Oder: GOSUB 1000 → GOSUB 980 : -Zeichen löschen.

Wenn Sie Ihr Quellprogramm hundertprozentig haben, dann können Sie das Renumberprogramm mit den Befehlen »GOTO 60, NEWLINE, 1, NEWLINE« löschen und nur Ihr Quellprogramm bleibt übrig.

Bei Verändern des Programmes muß Zeile 12, 16, 60, 61 überprüft werden.

```

31 IF F>=I AND C<I THEN GOSUB
35
32 NEXT V
33 LET C=F
34 RETURN
35 LET W$(1,1 TO 4)=STR$ G
36 LET J=VAL W$(V,1 TO 5)
37 LET L=VAL W$(V,10)
38 FOR U=1 TO L
39 LET H=CODE W$(1,U)
40 IF H<8 OR CODE W$(1,L+1)<>0
THEN LET H=H+128
41 POKE J+U,H
42 NEXT U
43 RETURN
44 LET J=PEEK (W+1)
45 IF J>63 OR J<28 THEN RETURN
46 IF J<38 THEN GOTO 49
47 POKE W+1,J+128
48 RETURN
49 LET K=K+1
50 LET W$(K,1 TO 5)=STR$ W
51 FOR U=1 TO 4
52 LET J=PEEK (W+U)-28
53 IF J>9 OR J<0 THEN GOTO 56
54 LET W$(K,U+5)=STR$ J
55 NEXT U
56 LET W$(K,10)=STR$ (U-1)
57 IF VAL W$(K,6 TO 9)<>D THEN
LET K=K-1
58 RETURN
59 POKE 16511,5
60 POKE 16512,5
61 POKE 16512,5
62 STOP

```



# PROGRAMME

## Scrollen ist wirklich ein Freizeitvergnügen und macht viel Spaß



... da aber viele USER-CLUB-Leser mit dem Original SCROLL von Sinclair unzufrieden sind, erdachte sich unser Leser Ch. Götz aus Wertingen ein kleines Maschinencode-Programm. Das lästige und lang andauernde Flimmern des Bildschirms wird radikal abgestellt. Die Bytes werden einfach in eine REM-Zeile eingepokt und der SCROLL-Befehl kann später durch RAND USR 16514 aufgerufen werden. Man muß allerdings die PRINT Position mit AT festlegen. Da das Programm 16 Bytes lang ist, muß man vor dem Einpoken 16 Bytes reservieren. 1 REM ... 16 beliebige Zeichen. Erstes Programm scrollt von unten nach oben und das zweite genau umgekehrt.

### Scroll nach oben

```
4082 2A0C40 LD HL,(400C)
4085 5D LD E,L
4086 54 LD D,H
4087 13 INC DE
4088 012200 LD BC,0022
4089 09 ADD HL,BC
408C 01C002 LD BC,02C0
408F EDB0 LDIR
4091 C9 RET
```

```
16514 42
16515 12
16516 64
16517 93
16518 84
16519 19
16520 1
16521 34
16522 0
16523 9
16524 1
16525 192
16526 2
16527 237
16528 176
16529 201
```

### Scroll nach unten

```
4082 00 NOP
4083 00 NOP
4084 2A0C40 LD HL,(400C)
4087 01B502 LD BC,02B5
408A 09 ADD HL,BC
408B 228240 LD(4082),HL
408E 012100 LD BC,0021
4091 09 ADD HL,BC
4092 54 LD D,H
4093 5D LD E,L
4094 2A0240 LD HL,(4082)
4097 01B502 LD BC,02B5
409A EDB0 LDDR
409C C9 RET
```

```
16514 0
16515 0
16516 42
16517 12
16518 64
16519 1
16520 181
16521 2
16522 9
16523 34
16524 130
16525 64
16526 1
16527 33
16528 0
16529 9
16530 84
16531 93
16532 42
16533 130
16534 64
16535 1
16536 181
16537 2
16538 237
16539 184
16540 201
```

Damit Sie den Vergleich zwischen dem Original-SCROLL und dem MC-Programm SCROLL haben, geben Sie folgende kleine Programme ein.

### ORIGINAL-SCROLL

```
10 DIM A$(15000)
20 FOR N=1 TO 10
30 SCROLL
40 PRINT AT 20,0;N
50 NEXT N
60 CLS
70 LIST
```

### MC-Programm SCROLL

```
10 DIM A$(15000)
20 FOR N=1 TO 10
30 RAND USR 16514
40 PRINT AT 20,0;N
50 NEXT N
60 CLS
70 LIST
```

Und unser Cheftechniker S. Sanver hat in seiner Freizeit und in den Mittagspausen die Scroll-Routinen nach beiden Seiten geschrieben. (s. unten)

### Scroll nach links

```
4082 2A0C40 LD HL,(400C)
4085 23 INC HL
4086 54 LD D,H
4087 5D LD E,L
4088 23 INC HL
4089 3E18 LD A,18
408B 011F00 LD BC,001F
408E EDB0 LDIR
4090 F5 PUSH AF
4091 97 SUB A
4092 12 LD(DE),A
4093 13 INC DE
4094 23 INC HL
4095 13 INC DE
4096 23 INC HL
4097 F1 POP AF
4098 3D DEC A
4099 20F0 JR NZ,408B
409B C9 RET
```

```
16514 42
16515 12
16516 64
16517 35
16518 84
16519 93
16520 35
16521 62
16522 24
16523 1
16524 31
16525 0
16526 237
16527 176
16528 245
16529 151
16530 18
16531 19
16532 35
16533 19
16534 35
16535 241
16536 61
16537 32
16538 240
16539 201
```

### Scroll nach rechts

```
4082 2A1040 LD HL,(4010)
4085 2B DEC HL
4086 2B DEC HL
4087 54 LD D,H
4088 5D LD E,L
4089 2B DEC HL
408A 3E18 LD A,18
408C 011F00 LD BC,001F
408F EDB0 LDDR
4091 F5 PUSH AF
4092 97 SUB A
4093 12 LD(DE),A
4094 2B DEC HL
4095 2B DEC HL
4096 1B DEC DE
4097 1B DEC DE
4098 F1 POP AF
4099 3D DEC A
409A 20F0 JR NZ,408C
409C C9 RET
```

# ZX-USER CLUB



# PROGRAMME

```

16514      42
16515      16
16516      64
16517      43
16518      43
16519      84
16520      93
16521      43
16522      62
16523      24
16524      1
16525      31
16526      0
16527      237
16528      184
16529      245
16530      151
16531      18
16532      43
16533      43
16534      27
16535      27
16536      241
16537      61
16538      32
16539      240
16540      201
    
```

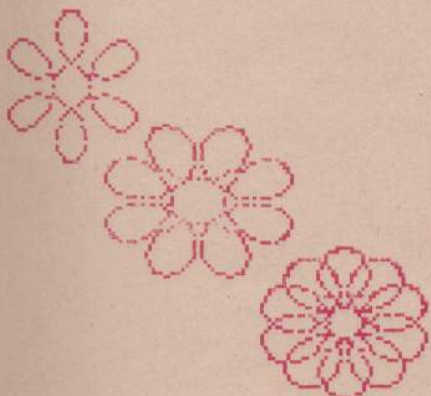
```

1 REM E#RND7??Y/3 GOSUB K
PRINT >K<7 LET X4 LIST TAN
    
```

```

10 FOR I=1 TO 100
20 PRINT AT 10+10*SIN (I/32*PI)
,31;" "AT 10+10*COS (I/32*PI),
31;" "AT 10,31;"--"
25 RAND USR 16514
30 NEXT I
    
```

## Schleifen



ANZAHL SCHLEIFEN: Gewünschte Anzahl eingeben.

SCHLEIFEN NACH INNEN ODER AUSSEN: Entsprechende Eingabe.

MUSTER: Ist die Zahl, die Sie eingeben, größer als die Anzahl der

Schleifen, so entsteht ein kreisförmiges Muster, ansonsten ein sternförmiges Muster.

SCHRITTWEITE: Bestimmt die Dichte der Linien (je größer, desto dichter).

VERZERRUNG: Ist sie negativ, dann wird das Muster verkleinert, ansonsten vergrößert. Mit X und Y kann das Bild in die Breite bzw. in die Höhe gezogen werden.

Musterbeispiele:

```

ANZ.SCHL. 7 7 5 5
INN/AUSS. -1 -1 1 -1
MUSTER 1.5 4 3 1.5
SCHRITTW. 90 100 100 100
VERZ.X 1.5 3 2 3
VERZ.Y 1.5 2 2 3
    
```

Wenn die Eingabe des letzten Durchlaufes beibehalten werden soll, einfach »D« eingeben.

```

1 CLS
10 PRINT "ANZAHL DER SCHLEIFEN"
15 INPUT A$
20 IF A$="D" THEN GOTO 30
25 LET AS=VAL A$
30 PRINT "I=INNEN -I=AUSSEN"
35 INPUT I$
40 IF I$="D" THEN GOTO 50
45 LET IS=VAL I$
50 PRINT "MUSTER"
55 INPUT M$
60 IF M$="D" THEN GOTO 70
65 LET MU=VAL M$
70 PRINT "SCHRITTWEITE"
75 INPUT S$
80 IF S$="D" THEN GOTO 90
85 LET SC=VAL S$
86 PRINT "VERZERRUNG X"
87 INPUT X$
88 IF X$="D" THEN GOTO 96
89 LET XV=VAL X$
96 PRINT "VERZERRUNG Y"
97 INPUT Y$
98 IF Y$="D" THEN GOTO 100
99 LET YV=VAL Y$
100 CLS
101 FOR A=-MU TO 3 STEP .1
110 FOR B=0 TO (2*PI)+PI/SC STEP
PI/SC
120 LET X=(AS+IA)*COS B-A*IA*CO
S ((AS+IA)/IA*B)
130 LET Y=(AS+IA)*SIN B-A*IA*SI
N ((AS+IA)/IA*B)
140 LET X1=X*XV+32
145 LET Y1=Y*YV+21
160 IF X1>62 THEN GOTO 210
161 IF Y1>41 THEN GOTO 210
180 PLOT X1,Y1
200 PAUSE 1
210 NEXT B
220 NEXT A
230 PAUSE 4E4
240 IF INKEY#<>"S" THEN GOTO 1
    
```

## Umwandlung in Hexadezimal



Dieses Programm wandelt alle Zahlen zwischen 0 und 255 in die entsprechende Hex-Zahl um.

```

4082 00      NOP
4083 218240  LD HL,4082
4086 7E      LD A,(HL)
4087 CB1F    RR A
4089 CB1F    RR A
408B CB1F    RR A
408D CB1F    RR A
408F CD9340  CALL 4093
4092 7E      LD A,(HL)
4093 E60F    AND 0F
4095 C61C    ADD A,1C
4097 D7      RST 10H
4098 C9      RET
    
```

```

16514      0
16515      33
16516      130
16517      64
16518      126
16519      203
16520      31
16521      203
16522      31
16523      203
16524      31
16525      203
16526      31
16527      205
16528      147
16529      64
16530      126
16531      230
16532      15
16533      198
16534      28
16535      215
16536      201
    
```

Die umzurechnende Zahl wird in 16514 gePOKED und das Programm mit USR 16515 aufgerufen.

Funktion:

Zuerst wird das linke Nibble in die entsprechende Ziffer gewandelt und ausgegeben und dann das rechte Nibble.



# PROGRAMME



## Pferderennen

Hier können Sie sich nun endlich als Reiter auf Ihrem hohen Roße versuchen. Versuchen Sie in kürzester Zeit den Parcours zu durchreiten. Jedes Hindernis nehmen Sie spielend mit der Taste M. Wir wünschen Ihnen viel Erfolg.

M. Gehrke, Weinstadt.

16K

```
2000 PRINT "PFERDERENNEN"
      "===== "
2010 PRINT
2020 PRINT "SIE REITEN AUF EINER
HINDERNIS- RENNBAHN."
2030 PRINT "UEBERSPRINGEN SIE DA
S HINDERNIS DURCH DRUECKEN DER T
ASTE "M"
2040 PRINT
2050 PRINT "VIEL SPASS."
2060 IF INKEY#="" THEN GOTO 2060
2080 CLS
2090 RETURN
3000 SAVE "PFERDERENNEN"
3010 RUN
```

```
16514      42
16515      12
16516      64
16517      1
16518      84
16519      2
16520      9
16521      68
16522      77
16523      10
16524      84
16525      93
16526      35
16527      1
16528      29
16529      0
16530      237
16531      176
16532      43
16533      119
16534      201
```

```
4082 2A0C40 LD HL,(400C)
4085 015402 LD BC,0254
4088 09 ADD HL,BC
4089 44 LD B,H
```

```
408A 4D LD C,L
408B 0A LD A,(BC)
408C 54 LD D,H
408D 5D LD E,L
408E 23 INC HL
408F 011D00 LD BC,001D
4092 EDB0 LDIR
4094 2B DEC HL
4095 77 LD(HL),A
4096 C9 RET
```

```
1 REM 21 BELIEBIGE ZEICHEN
2 REM *(< PFERDERENNEN >)*
3 REM (C)BY MARTIN GEHRKE
4 GOSUB 2000
5 LET V=0
6 LET N=2
7 LET A$=""
10 FOR Q=0 TO 31
20 PRINT AT 19,Q;" "
30 IF Q=10 OR Q=17 OR Q=23 OR
Q=28 OR Q=30 THEN PRINT AT 18,Q;
"|"
40 NEXT Q
45 FOR W=1 TO 5
50 IF USR 16514 THEN
60 IF INKEY#="M" THEN GOTO 100
65 PRINT AT 18,N;
70 LET A=PEEK (PEEK 16398+256*
PEEK 16399)
75 LET V=0
80 IF A<>0 THEN GOTO 1000
90 PRINT AT 18,N;"R"
95 GOTO 200
100 IF V>3 THEN GOTO 65
105 PRINT AT 17,N;"R";AT 18,N;"
"
110 LET V=V+1
205 PRINT AT 18,(N+1);
206 LET A$=CHR# PEEK (PEEK 1639
8+256*PEEK 16399)
210 PRINT AT 18,N;A$;AT 17,N;"
"
220 NEXT W
230 IF N<27 THEN LET N=N+1
235 IF INT (N/6)=N/6 THEN PRINT
AT 18,29;"|"
240 GOTO 45
1000 PRINT AT 0,0;"STURZ...."
1010 FOR Q=1 TO 100
1020 NEXT Q
1025 CLS
1030 RUN
```

## BOLIDEN



Sie sind Raumkadett und stehen vor der Beförderung zum Leutnant. Die Prüfung ist ein Raumflug durch einen Bolidschwarm. Die Steuerung erfolgt mit 1 — Flug nach oben, SHIFT — Flug nach unten, O — Feuer.

Machen Sie ruhig ausgiebigen Gebrauch von der Bordkanone. Viel Glück.

M. Gehrke, Weinstadt.

16K

```
1 REM 279 BELIEBIGE ZEICHEN
2 REM 22 BELIEBIGE ZEICHEN
10 SLOW
15 GOSUB 1000
20 FOR Q=0 TO 31
30 PRINT AT INT (RND*22),Q;"O"
40 NEXT Q
50 PRINT AT 0,0;"=====
===== "
60 PRINT AT 21,0;"=====
===== "
64 POKE 16436,255
65 POKE 16437,255
67 PRINT AT 10,6;" "
70 IF USR 16514 THEN
72 LET A=(65536-(PEEK 16436+25
6*PEEK 16437))/50
74 IF PEEK 16417=0 THEN GOTO 2
00
75 FOR Q=0 TO 80
76 IF USR 16799 THEN
77 NEXT Q
80 CLS
90 PRINT ,,,,"SIE HABEN IHR RA
UMSCHIFF NACH"
100 PRINT A;TAB 6;"SEKUNDEN DUR
CH KOLLISION";TAB 0;"MIT EINEM B
OLIDEN VERLOREN."
110 PRINT ,,"SCHADE - SIE BLEIB
EN RAUMKADETT"
120 PRINT ,,,,"WOLLEN SIE DIE P
RUEFUNG WIEDER- HOLEN ? SIE BRAU
CHEN NUR NEURNE ZU DRUEC
KEN"
130 FOR Q=0 TO 1000
140 IF INKEY#="CHR# 118 THEN GOT
O 170
150 NEXT Q
155 CLS
160 RUN
170 CLS
```



# PROGRAMME

```

180 GOTO 20
200 CLS
205 PRINT "SIE HABEN DEN BO
LIDENSCHWARM OHNE SCHADEN DUR
CHQUERT. SIE WERDEN HIERN
IT ZUM LEUTNANT BEFOERDERT."
"ICH GRATULIERE"
210 FOR Q=0 TO 500
220 IF INKEY$(">") THEN GOTO 240
230 NEXT Q
240 CLS
250 RUN
300 STOP

```

```

1000 PRINT "**** BOLIDEN ****"
SIE SIND RAUMKADETT
VOR DER BEFOERDERUNG
LEUTNANT. DIE PRUEFU
RAUMFLUG DURCH EINEN
SCHWARM.
"STEUERUNG:
1.....FLUG NAC
SHIFT.....FLUG NAC
H OBEN
H UNTEN "
1020 PRINT "0.....SCHUSS"
1030 PRINT "MACHEN SIE RUHIG A
USGIEBIGEN GEBRAUCH VON DER B

```

```

ORDKANONE."
1040 PRINT "VIEL GLUECK."
1045 POKE 16418,0
1046 PRINT "COPY M
.GEHRKE"
1047 POKE 16418,2
1050 FOR Q=0 TO 4000
1060 IF INKEY$(">") THEN GOTO 110
0
1070 NEXT Q
1100 CLS
1110 RETURN
2000 SAVE "BOLIDEN"
2010 RUN

```

16514	42	2A	16575	35	23	16636	22	16	16697	64	40	16759	25	19
16515	12	0C	16576	62	3E	16637	43	2B	16698	79	4F	16760	54	36
16516	64	40	16577	0	00	16638	43	2B	16699	10	0A	16761	180	B4
16517	17	11	16578	190	BE	16639	43	2B	16700	230	E6	16762	197	C5
16518	80	50	16579	192	C0	16640	16	10	16701	31	1F	16763	225	E1
16519	1	01	16580	42	2A	16641	235	EB	16702	254	FE	16764	24	18
16520	25	19	16581	123	7B	16642	193	C1	16703	31	1F	16765	4	04
16521	34	22	16582	64	40	16643	16	10	16704	32	20	16766	54	36
16522	123	7B	16583	54	36	16644	229	E5	16705	2	02	16767	180	B4
16523	64	40	16584	18	12	16645	42	2A	16706	54	36	16768	197	C5
16524	62	3E	16585	62	3E	16646	123	7B	16707	180	B4	16769	225	E1
16525	254	FE	16586	191	BF	16647	64	40	16708	16	10	16770	35	23
16526	38	26	16587	38	26	16648	54	36	16709	237	ED	16771	193	C1
16527	1	01	16588	1	01	16649	0	00	16710	58	3A	16772	16	10
16528	219	DB	16589	219	DB	16650	42	2A	16711	52	34	16773	216	D8
16529	254	FE	16590	254	FE	16651	12	0C	16712	64	40	16774	35	23
16530	164	A4	16591	164	A4	16652	64	40	16713	230	E6	16775	193	C1
16531	32	20	16592	300	C8	16653	35	23	16714	7	07	16776	16	10
16532	15	0F	16593	62	3E	16654	6	06	16715	254	FE	16777	209	D1
16533	42	2A	16594	239	EF	16655	21	15	16716	7	07	16778	58	3A
16534	123	7B	16595	38	26	16656	197	C5	16717	194	C2	16779	33	21
16535	64	40	16596	1	01	16657	126	7E	16718	140	8C	16780	64	40
16536	17	11	16597	219	DB	16658	229	E5	16719	64	40	16781	254	FE
16537	33	21	16598	254	FE	16659	209	D1	16720	62	3E	16782	0	00
16538	0	00	16599	164	A4	16660	35	23	16721	0	00	16783	200	C8
16539	25	19	16600	32	20	16661	1	01	16722	50	32	16784	195	C3
16540	62	3E	16601	7	07	16662	31	1F	16723	33	21	16785	140	8C
16541	0	00	16602	42	2A	16663	0	00	16724	64	40	16786	64	40
16542	190	BE	16603	123	7B	16664	237	ED	16725	42	2A	16787	0	00
16543	32	20	16604	64	40	16665	176	80	16726	12	0C	16788	0	00
16544	3	03	16605	35	23	16666	43	2B	16727	64	40	16789	0	00
16545	34	22	16606	35	23	16667	119	77	16728	35	23	16790	0	00
16546	123	7B	16607	54	36	16668	35	23	16729	6	06	16791	0	00
16547	64	40	16608	22	16	16669	35	23	16730	21	15	16792	201	C9
16548	62	3E	16609	42	2A	16670	19	13	16731	197	C5	16793	118	76
16549	247	F7	16610	12	0C	16671	19	13	16732	6	06	16794	0	00
16550	38	26	16611	64	40	16672	193	C1	16733	32	20	16795	2	02
16551	1	01	16612	17	11	16673	16	10	16734	197	C5	16796	24	18
16552	219	DB	16613	214	D6	16674	237	ED	16735	229	E5	16797	0	00
16553	254	FE	16614	2	02	16675	58	3A	16736	193	C1	16798	234	EA
16554	164	A4	16615	25	19	16676	53	35	16737	62	3E	16799	42	2A
16555	32	20	16616	6	06	16677	64	40	16738	180	B4	16800	12	0C
16556	15	0F	16617	21	15	16678	254	FE	16739	190	BE	16801	64	40
16557	42	2A	16618	197	C5	16679	249	F9	16740	32	20	16802	35	23
16558	123	7B	16619	6	06	16680	56	38	16741	28	1C	16803	6	06
16559	64	40	16620	33	21	16681	28	1C	16742	50	32	16804	22	16
16560	17	11	16621	62	3E	16682	42	2A	16743	33	21	16805	197	C5
16561	223	DF	16622	22	16	16683	12	0C	16744	64	40	16806	6	06
16562	255	FF	16623	190	BE	16684	64	40	16745	54	36	16807	32	20
16563	25	19	16624	32	20	16685	17	11	16746	0	00	16808	126	7E
16564	62	3E	16625	13	0D	16686	32	20	16747	17	11	16809	22	16
16565	0	00	16626	54	36	16687	0	00	16748	223	DF	16810	128	80
16566	190	BE	16627	0	00	16688	0	00	16749	255	FF	16811	130	82
16567	32	20	16628	35	23	16689	25	19	16750	25	19	16812	119	77
16568	3	03	16629	35	23	16690	6	06	16751	62	3E	16813	35	23
16569	34	22	16630	62	3E	16691	21	15	16752	8	08	16814	16	10
16570	123	7B	16631	118	76	16692	17	11	16753	190	BE	16815	248	F8
16571	64	40	16632	190	BE	16693	33	21	16754	32	20	16816	35	23
16572	42	2A	16633	40	28	16694	0	00	16755	10	0A	16817	193	C1
16573	123	7B	16634	4	04	16695	25	19	16756	17	11	16818	16	10
16574	64	40	16635	54	36	16696	58	3A	16757	148	94	16819	241	F1
							52	34	16758	2	02	16820	201	C9



# PROGRAMME

## BLACK JACK

Auf drei, nicht weit voneinander entfernten Feldern des Bildschirms erscheinen in schnellem Rhythmus verschiedene, immer wiederkehrende Graphiksymbole.

Der Spieler muß nun (wie bei den großen Glücksspielautomaten) diese blinkenden Felder — unter Zuhilfenahme von INKEY\$=»1« — nacheinander 'anhalten' und jedes Feld auf ein bestimmtes Graphiksymbol justieren.

Ziel des Spiels ist es, drei oder zumindest zwei gleiche Zeichen auf dem Bildschirm zum Stehen zu bringen. Gelingt das, werden dem Spieler bestimmte Geldbeträge zu seinem Guthaben addiert, wenn nicht, wird Geld abgezogen.

20 Durchgänge dieses Spiels muß jeder 'Gambler' hinter sich bringen.

Der höchste erreichte Geldbetrag wird gespeichert und bei SAVE auf Tonband mitaufgenommen.

Falls das Programm mit BREAK unterbrochen wird, bitte wieder mit GOTO starten, da RUN die Variable D, die den Rekord speichert, löschen würde.

FAITES VOS JEUX — und viel Spaß!

R. Pecha, Filderstadt



```

1 LET D=0
5 PRINT "BLACK JACK"
10 LET G=0
15 FOR K=1 TO 2
20 LET A=1
30 PRINT AT 2,11;"SPIEL ";K;"
"
50 LET A$="■□●"
60 FOR N=1 TO 3
70 IF A=1 THEN LET B$=A$(N)
75 IF A=2 THEN LET C$=A$(N)
80 IF A=3 THEN LET D$=A$(N)
85 PRINT AT 5,9;B$;" ";C$;"
";D$
90 LET A=A+(INKEY$="1")
100 NEXT N
110 IF A<4 THEN GOTO 60
120 LET G=G+(B$=C$ OR C$=D$ OR B$=D$)*300+(B$=C$ AND B$=D$)*200-200
125 PRINT AT 9,7;"GUTHABEN: ";G;"
";DM
130 NEXT K
150 IF G>D THEN LET D=G
160 PRINT AT 11,9;"REKORD: ";D;"
";DM
180 IF INKEY$="" THEN GOTO 180
190 GOTO 10
    
```

## Kamikaze



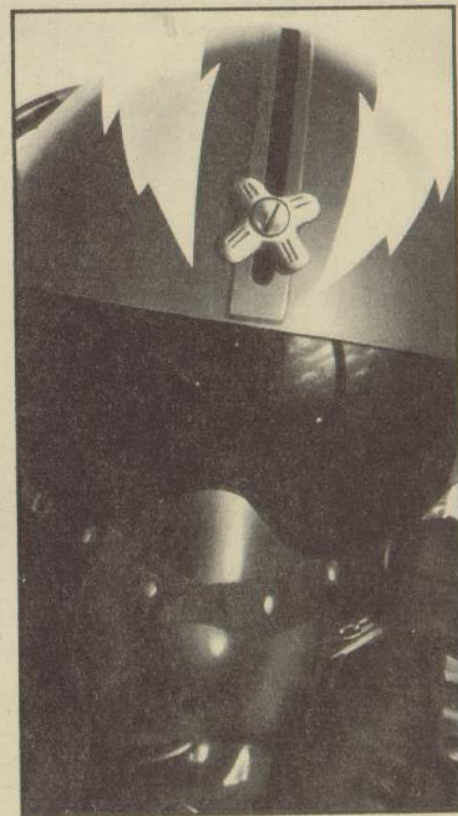
Sie befinden sich auf der linken Seite des Bildschirms und Ihr Angreifer kommt von der rechten Seite des Bildschirms auf Sie zugeflogen. Mit den Tasten Q nach oben und A nach unten und P für Feuer können Sie Ihren Gegner treffen.

```

10 LET A=0
20 LET J=200
30 LET K=10
40 LET G=0
50 LET X=INT (RND*18)+2
60 LET A=A+1
70 IF A=21 THEN GOTO 260
80 LET Y=30
90 PRINT AT K,0;CHR$ 130;CHR$
128;AT X,Y;"X"
100 IF J<0 THEN GOTO 150
110 IF INKEY$="Q" THEN LET K=K-1
120 IF INKEY$="A" THEN LET K=K+1
130 IF INKEY$="P" THEN PRINT AT K,2;"*****"
140 IF INKEY$>"5" THEN LET J=J-1
150 LET Y=Y-1.5
160 IF Y=3 THEN LET G=G+1
170 IF G=5 THEN GOTO 240
180 IF Y=3 THEN GOTO 50
190 IF INKEY$="P" AND K=X AND Y<21 THEN GOTO 220
200 CLS
210 GOTO 90
220 PRINT AT X,Y+1;CHR$ 189
230 GOTO 50
240 PRINT "ZERSTOERT"
250 STOP
260 PRINT "SIE GEWINNEN"
270 PRINT "TANK=";J
    
```

## Reaktionstest mit Zeitangabe

Herr Christian Götz aus Wertingen schickte uns dieses 1K-Programm ein. Dieses Programm erklärt sich selbst und klärt Sie genauestens über Ihren Gemütszustand auf.



```

10 PRINT "ZUM START TASTE DRUECKEN"
20 PRINT "SOBALD AUF DEM MONITOR EIN PUNKTERSCHEINT, SCHNELL EINE TASTE BETÄTIGEN..."
30 IF INKEY$="" THEN GOTO 30
40 CLS
50 FOR N=0 TO RND*600+20
60 NEXT N
70 PRINT AT RND*19,RND*31;"."
90 LET TA=PEEK 16436+256*PEEK 16437
100 IF INKEY$="" THEN GOTO 100
110 LET TE=PEEK 16436+256*PEEK 16437
120 LET T=(ABS TA-ABS TE)/50
125 IF T<.1 THEN GOTO 160
130 PRINT AT 10,0;T;" SEK."
140 IF T<=.25 THEN PRINT "... I ST JA SPITZE"
150 IF T<=.4 AND T>.25 THEN PRINT "...NICHT SCHLECHT"
155 IF T>.4 THEN PRINT "...DU NACHTWAECHESTER"
160 IF T<.1 THEN PRINT "DU VERDAMMTER HELLSEHER...TASTE NICHT SCHON VORHER DRUECKEN"
165 IF INKEY$<">" THEN GOTO 165
170 RUN
    
```



# PROGRAMME

## ALBRECHT DÜRER

Es handelt sich hier um das bekannte Spiel, bei dem 15 (von 1—15 nummeriert) Klötzchen in einem Rahmen so lange hin und her geschoben werden müssen, bis sie in einer Folge liegen. Das Programm läuft auf dem ZX81 in der 1K-Version. Herr Alexander Schaut aus Winsen schickte uns dieses interessante Programm ein.

Nach Eingabe des Programms starten Sie durch RUN und warten ca. zwei Minuten. CPU benötigt die Zeit, um die interne Tabelle aufzubauen. Eine Matrix 4 mal 4 muß von oben links in folgender Reihenfolge geordnet werden:

1 2 3 4 5 6 7 8 9 A B C D E F. Das Verschieben geschieht durch Eingabe der Steinbezeichnung (1—F) auf der Tastatur. Eine Eingabe der Bewegungsrichtung entfällt, da es nur eine mögliche Richtung gibt, die der Computer selber ermittelt. Die Anzahl der Verschiebungen wird angezeigt. Bitte nur gültige Werte eingeben, da wegen Speichermangels keine Fehlerbedingungen abgefangen werden können.

```

1 LET N=SGN PI
3 LET V=VAL "5"
10 DIM A$(V+N,V+N)
100 FOR Z=N TO VAL "15"
120 LET Z#=CHR# INT (RND*VAL "1
5"+VAL "29")
150 FOR A=N+N TO V
152 LET A$(A,V+N)="I"
154 LET A$(A,N)="I"
160 FOR B=N+N TO V
162 LET A$(N,B)="I"
164 LET A$(V+N,B)="I"
170 IF A$(A,B)=Z# THEN GOTO VAL
"120"
175 IF A$(A,B)=" " THEN GOTO VA
L "320"
180 NEXT B
190 NEXT A
320 LET A$(A,B)=Z#
340 NEXT Z
350 FOR A=SGN PI TO VAL "6"
360 PRINT AT A,NOT PI;A$(A)
370 NEXT A
380 PRINT AT NOT PI,NOT PI;Z-VA
L "15"
400 INPUT Q#
405 IF Q#>"F" OR Q#<"1" THEN GO
TO VAL "400"
410 FOR A=N+N TO V
420 FOR B=N+N TO V
430 IF A$(A,B)=Q# THEN GOTO VAL
"510"
440 NEXT B
450 NEXT A
510 IF A$(A+N,B)=" " THEN LET A
$(A+N,B)=Q#
520 IF A$(A-N,B)=" " THEN LET A
$(A-N,B)=Q#
530 IF A$(A,B-N)=" " THEN LET A
$(A,B-N)=Q#
540 IF A$(A,B+N)=" " THEN LET A
$(A,B+N)=Q#
550 LET A$(A,B)=" "
610 LET Z=Z+N
710 GOTO VAL "350"

```

## Bildschirminvertierer

1K/16K



Mit Hilfe dieses NC-Programms können Sie den gesamten Bildschirminhalt invertieren. Poken Sie mit Hilfe des NC-Loader (siehe Heft März/April) die Dezimal- oder Hexadezimalzahlen. Anschließend starten Sie bitte das Programm mit RUN. Geben Sie nun Ihren Text ein. Durch Drücken der SHIFT- und S-Taste zu gleicher Zeit rufen Sie das MC-Programm auf. Überraschung: Ändern Sie die Zeile 30 wie folgt und fügen Sie die Zahlen 100 bis 130 zu. Sie werden staunen.

```

1 REM 26 BELIEBIGE ZEICHEN
10 IF INKEY#<" " THEN GOTO 10
20 IF INKEY#="" THEN GOTO 20
30 IF INKEY#=CHR# 225 THEN RAN
D USR 16514
40 IF CODE INKEY#>63 THEN GOTO
10
50 PRINT INKEY#;
60 GOTO 10

30 IF INKEY#=CHR# 225 THEN GOS
UB 100

100 FOR I=1 TO 10
110 RAND USR 16514
120 NEXT I
130 RETURN

```

### Hexadezimal

```

4082 2A0C40 LD HL,(400C)
4085 23 INC HL
4086 ED5B1040 LD DE,(4010)
408A A7 AND A
408B ED52 SBC HL,DE

```

```

408D 19 ADD HL,DE
408E D0 RET NC
408F 46 LD B,(HL)
4090 78 LD A,B
4091 07 RLCA
4092 07 RLCA
4093 36F0 JR C,4085
4095 0F RRCA
4096 0F RRCA
4097 EE00 XOR 80
4099 77 LD(HL),A
409A 18E9 JR 4085

```

### Dezimal

```

16514 42
16515 12
16516 64
16517 35
16518 237
16519 91
16520 16
16521 64
16522 167
16523 237
16524 82
16525 25
16526 208
16527 70
16528 120
16529 7
16530 7
16531 56
16532 240
16533 15
16534 15
16535 238
16536 128
16537 119
16538 24
16539 233

```

# ZX-USER CLUB



4/6/7/14



## Australische Landung

Sind Sie ein guter Känguruhvater?

Im abenteuerlichen australischen Busch steht Mutter Känguruh (leicht zu erkennen an ihrer Figur) und hat ihr kleines Baby (»J«-invers) am Boden verloren. Versuchen Sie nun als Känguruhvater durch Eingabe der Wurfweite (Zahlen 1 bis 19) Ihr Baby in den schützenden Kängurhsack der Mutter zu manövrieren. Sie haben drei Versuche.

```

10 PRINT TAB 15; "■", TAB 14; "■"
20 PRINT TAB 16; "■", TAB 15; "■"
30 PRINT TAB 13; "■", TAB 14; "■"
40 PRINT TAB 15; "■", TAB 13; "■"
50 LET F=1
60 LET G=F+F
70 LET E=F-F
80 LET H=7
90 FOR N=F TO 3
100 LET I=RND*6
110 PRINT AT E,E;"SPRUNG ";N
120 PRINT AT H,I;"■"
130 INPUT K
140 LET F=1
150 LET X=H
160 FOR Y=I TO 12
170 PRINT AT X,Y;"■"
180 LET Z=X-F
190 IF Y=K+I THEN LET F=-F
200 IF Z<E THEN LET Z=E
210 IF Z>H THEN LET Z=H
220 PRINT AT X,Y;" "
230 LET X=Z
240 NEXT Y
250 IF X>4 THEN GOTO 280
260 PRINT AT 3,4;"ERFOLG ■"
270 STOP
280 NEXT N
290 PRINT AT 6,0;"ZU ENDE"

```

## Turm von Babel

Hier können Sie sich genötigen in Ihren Sessel zurücklehnen und zuschauen, wie Ihr Computer den Turm von Babel wieder aufbaut. Der Turm von Babel ist eine Pyramide, die Schicht für Schicht von einem Ort zu einem anderen versetzt wird. Die Form darf natürlich nicht verändert werden. Außerdem gibt es nur eine Stelle, wo Sie Teile der Pyramide für kurze Zeit lagern können. Der Computer fragt Sie nach der gewünschten Anzahl der Schichten, die Sie zwischen 1 und 9 wählen können. Drücken Sie dann Newline. Dann bestimmen Sie, in welche Position die Pyramide am Anfang sein soll (1, 2 oder 3) und wann die Pyramide am Ende sein soll (1, 2 oder 3). Die Pyramide wird nun von Schicht für Schicht vom Computer in der kleinsten Anzahl von Schritten versetzt werden. Versuchen Sie einmal vor jedem Zug vorherzusagen, welchen Zug der Computer als nächstes tun wird.

```

10 LET T=INT PI
20 LET Q=T*T
30 LET U=T/T
50 PRINT "ZAHL"
60 INPUT N
70 PRINT "VON"
80 INPUT A
90 PRINT "ZU"
100 INPUT B
110 CLS
120 LET X=U
130 DIM Z$(N)
140 DIM K(T)
150 DIM H$(T,Q)
160 LET K(A)=N
170 FOR C=U TO N
180 LET H$(A,C)=STR$(N+U-C)
190 PRINT AT Q-C,A+A-U-U;N+U-C;
    AT Q,T-T;" "
200 NEXT C
240 GOSUB Q**T
250 PRINT AT Q+U,T-T;"FERTIG"
260 STOP
730 IF N<U THEN RETURN
740 LET Z$(X)=STR$(T*A-T+B)
750 LET B=T+T-A-B
760 LET X=X+U
770 LET N=N-U
780 GOSUB Q**T
790 LET X=X-U
800 LET A=INT ((VAL Z$(X)+U+U)/
    T)
810 LET B=VAL Z$(X)-T*A+T
820 LET K(B)=K(B)+U
830 PRINT AT Q-K(A),A+A-U-U;" "
    AT Q-K(B),B+B-U-U;H$(A,K(A))
840 LET H$(B,K(B))=H$(A,K(A))
850 LET K(A)=K(A)-U
860 LET A=T+T-A-B
870 GOSUB Q**T
880 LET N=N+U
890 RETURN

```



## Bowling

Beim Kegeln heißt es »alle Neune« — beim Bowling »gute Bowle«. Ihr Computer setzt 30 Pins und Sie sollten natürlich versuchen, mit 6 Bowlingkugeln einen Strike zu werfen. Mit den Tasten Z (unten) und Q (oben) können Sie die Kugel in die richtige Wurfposition bringen. Wenn Sie dann bereit sind, drücken Sie P und die Bowlingkugel nimmt ihren Lauf.

```

10 RAND
20 LET Q=VAL "5"
30 DIM B$(Q,6)
40 LET R=Q/Q
50 LET S=Q-Q
60 FOR N=R TO Q
70 LET B$(N)="000000"
80 PRINT AT N,R;B$(N)
100 NEXT N
150 LET A$="Q"
160 LET X=S
170 PRINT AT X,15;A$
180 PRINT AT X,15;" "
200 LET X=X+(INKEY$="Z")-(INKEY$="Q")
210 LET X=X+(X<S)-(X>6)
230 IF INKEY$<>"P" THEN GOTO 165
290 LET C=X
300 FOR Y=15 TO R STEP -1
310 PRINT AT X,Y;A$
320 IF Y>6 OR X=S OR X=6 THEN GOTO 530
340 IF B$(X,Y)<>"0" THEN GOTO 530
350 LET B$(X,Y)=" "
370 LET C=X+(RND*.6)-(RND*.4)
530 PRINT AT X,Y;" "
540 LET X=C
550 NEXT Y
560 LET A$=CHR$(CODE A$-R)
570 IF A$<>"Q" THEN GO TO 160
580 PRINT AT S,S;"DAS SPIEL IST ZU ENDE"

```



# PROGRAMME

## Lineare Regressions- und Korrelationsanalyse



Nach Eingabe des Programms ist das Programm selbsterklärend und druckt Ihnen die Ergebnisse der X- und Y-Werte aus.  
Martin Gehrke, Weinstadt.

16K

```
10 REM LINEARE REGRESSIONS-UND
    KORRELATIONSANALYSE
20 REM (C)BY MARTIN GEHRKE
30 PRINT "LINEARE REGRESSIONS-
    UND
    KORRELATIONSANALYSE
    =====
```

```
=====
31 LET B=0
32 LET X=0
33 LET Y=0
34 LET X1=0
35 LET Y1=0
36 LET X2=0
37 LET Y2=0
38 LET Z=0
39 LET S1=0
40 LET R=0
45 PRINT AT 5,0;"EINGABE: ANZAHL
    DER ZAHLENPAARE "
50 INPUT N1
60 IF N1>=3 THEN GOTO 100
70 PRINT AT 5,0;"ES MUESSEN ME
    HR ALS 2 PAARE SEIN"
80 FOR Q=1 TO 150
85 NEXT Q
90 GOTO 40
100 DIM L(2,N1)
110 PRINT AT 5,8;"ZAHLENPAARE (
    X,Y) "
120 FOR N=1 TO N1
130 FOR Q=1 TO 2
140 IF Q=2 THEN PRINT AT 21,0;"
    Y:"
150 IF Q=1 THEN PRINT AT 21,0;"
    X:"
160 INPUT L(Q,N)
170 NEXT Q
180 NEXT N
190 PRINT AT 21,0;" "
200 FOR N=1 TO N1
```

```
210 LET X=X+L(1,N)
220 LET Y=Y+L(2,N)
230 LET X2=X2+L(1,N)*L(1,N)
240 LET Y2=Y2+L(2,N)*L(2,N)
250 LET Z=Z+L(1,N)*L(2,N)
260 NEXT N
270 LET X1=X/N1
280 LET Y1=Y/N1
285 IF N1*X2-X*X=0 THEN GOTO 30
290 LET B=(N1*Z-X*X*Y)/(N1*X2-X*X
    )
300 LET A=Y1-B*X1
305 IF N1-2=0 THEN GOTO 320
310 LET S1=SQR ((Y2-A*Y-B*Z)/(N
    1-2))
320 LET X0=SQR (N1*X2-X*X)
330 LET Y0=SQR (N1*Y2-Y*Y)
335 IF X0*Y0=0 THEN GOTO 350
340 LET R=(N1*Z-X*X*Y)/(X0*Y0)
350 PRINT AT 5,0;"ERGEBNIS DER
    ANALYSE: "
380 PRINT AT 6,0;"X-WERTE";TAB
    15;"Y-WERTE"
390 PRINT "-----";TAB 15;"-----"
395 LET Q=1
400 FOR N=1 TO N1
410 PRINT L(1,N);TAB 15;L(2,N)
420 IF Q>13 THEN GOTO 450
430 LET Q=Q+1
440 NEXT N
450 LET A#="WEITER ? NEWLINE DR
    UECKEN "
460 FOR J=1 TO 30
470 POKE (PEEK 16396+256*PEEK 1
    6397)+726+J,CODE A#(J)
480 NEXT J
490 LET Q=1
500 FOR J=1 TO 5000
510 IF INKEY$=CHR$ 118 THEN GOT
    O 530
520 NEXT J
530 FOR J=1 TO 30
540 IF J<31 THEN POKE (PEEK 163
    96+256*PEEK 16397)+726+J,0
```

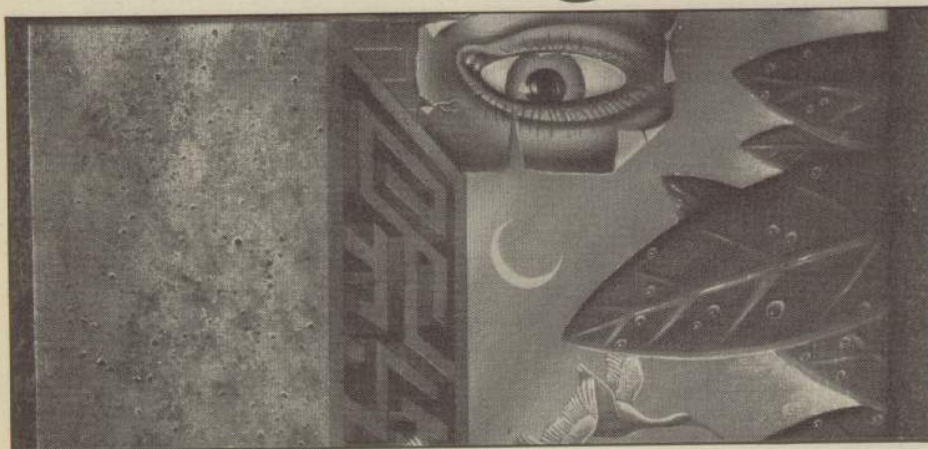
```
545 NEXT J
550 FOR J=8 TO 21
560 PRINT AT J,0;" "
570 NEXT J
580 PRINT AT 8,0;
590 NEXT N
600 PRINT "ANZAHL DER PAARE:";TAB
    AB 20;N1
610 PRINT "SUMME DER X-WERTE:";
    TAB 20;X
620 PRINT "SUMME DER Y-WERTE:";
    TAB 20;Y
630 PRINT "SUMME VON X*Y:";TAB
    20;Z
640 PRINT "SUMME DER X-QUADR.:"
    TAB 20;X2
650 PRINT "SUMME DER Y-QUADR.:"
    TAB 20;Y2
660 PRINT "MITTELWERT VON X:";TAB
    AB 20;X1
670 PRINT "MITTELWERT VON Y:";TAB
    AB 20;Y1
680 PRINT "STEIGUNG: GERADE :";
    TAB 20;B
690 PRINT "SCHNITTP. Y-ACHSE:";
    TAB 20;A
700 PRINT "STANDARTFEHLER:";TAB
    20;S1
710 PRINT "KORRELATIONSKOEFFIZI
    ENT:";TAB 20;R
720 LET A#="FUER <COPY> Z DRUECK
    EN SONST X"
730 FOR Q=1 TO 30
735 POKE (PEEK 16396+256*PEEK 1
    6397)+726+Q,CODE A#(Q)
740 NEXT Q
800 FOR Q=1 TO 5000
810 IF INKEY$="Z" THEN COPY
815 IF INKEY$="X" THEN GOTO 830
820 NEXT Q
830 CLS
840 RUN
1000 SAVE "LRKA"
1010 RUN
```

ZX-USER CLUB



# PROGRAMME

## Balkendiagramm



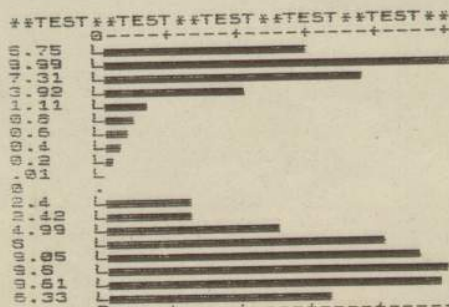
Dieses Programm ermöglicht das Erstellen von Balkendiagrammen mit gleichzeitiger Anzeige des Zahlenwertes. Es können maximal 20 Balken in einem Diagramm gezeichnet werden. Sind es genau 20, dann fällt allerdings die untere Begrenzung weg. Die Auflösung beträgt 50 Schritte. Die rechte Begrenzung des Diagramms kann auf zwei Arten vorgenommen werden. Nach der Dateneingabe wird nach dem Maximalwert gefragt. Bei Eingabe einer Zahl entspricht der rechte Rand des Diagramms diesem Wert; bei Eingabe von D wird der höchste Wert die ganze

Skalenbreite einnehmen und die Länge der übrigen Balken wird in Relation dazu berechnet. Ist ein Wert 0, wird das extra durch einen Punkt am linken Rand angezeigt. Die eingegebenen Zahlen sollten inklusive Dezimalpunkt nicht länger als 6 Stellen sein, da sie sonst teilweise überschrieben werden. Das Diagramm kann mit einer Überschrift von maximal 32 Zeichen versehen werden. In dieser Form ist das Balkendiagramm nur zur Anzeige von positiven Zahlen geeignet.

Dr. W. Steinmüller, Wien

```
10
1 REM (C) 1982 W.STEINMUELLER
10 PRINT "BALKENDIAGRAMM"
20 PRINT "===== "
30 PRINT "DATENMENGE =", " (M
AX=20)"
40 INPUT N
50 PRINT AT 2,13;N
60 PRINT "AUFLOESUNG = 50 SC
HRITTE"
70 PRINT "DATENEINGABE"
80 DIM D(N)
90 LET MAX=0
100 FOR I=1 TO N
110 INPUT D(I)
120 PRINT D(I)
130 IF D(I)>MAX THEN LET MAX=D
(I)
140 NEXT I
150 PRINT AT 8+N/2,0;"MAXIMALWE
RT ="
160 LET D=MAX
170 INPUT MAX
180 PRINT MAX
190 PRINT
200 PRINT "BESCHRIFTUNG BALKEND
IAGRAMM : "
210 DIM A$(32)
220 INPUT A$
230 PRINT A$
```

```
240 PAUSE 250
250 CLS
260 PRINT A$
270 PRINT AT 1,6;"0-----+---
+-----+
280 IF N<20 THEN PRINT AT N+2,6
;"0-----+-----+-----+
290 FOR I=1 TO N
300 PRINT AT 1+I,0;D(I);TAB 6;"
L"
310 IF D(I)=0 THEN PRINT AT 1+I
,6;"."
320 LET D(I)=(((D(I)/MAX)*50)+.
5)
330 FOR J=0 TO D(I)-1
340 PLOT J+14,40-I*2
350 NEXT J
360 NEXT I
```



## Umwandlung in die 2er-Basis

Dieses Programm wandelt alle Zahlen zwischen 0 und 255 in die entsprechende Binärzahl um und gibt diese auf dem Bildschirm aus.

Hier das Z80-Listing:

```
4082 00 NOP
4083 218240 LD HL,4082
4086 4E LD C,(HL)
4087 0609 LD B,09
4089 CB19 RR C
408B 1001 DJNZ 408E
408D C9 RET
408E CB11 RL C
4090 CB79 BIT 7,C
4092 2005 JR NZ,4099
4094 3E10 LD A,10
4096 D7 RST 10H
4097 18F2 JR 408B
4099 3E10 LD A,10
409B D7 RST 10H
409C 18ED JR 408B
```

```
16514 0
16515 33
16516 130
16517 64
16518 78
16519 6
16520 9
16521 203
16522 25
16523 16
16524 1
16525 201
16526 203
16527 17
16528 203
16529 121
16530 32
16531 5
16532 62
16533 28
16534 215
16535 24
16536 242
16537 62
16538 29
16539 215
16540 24
16541 237
```

Die umzurechnende Zahl wird in 16514 gePOKED und das Programm mit USR 16515 aufgerufen.

Die Funktionsweise ist einfach; es werden nacheinander Bit 7 bis 0 abgefragt und ausgegeben.

# ZX-USER CLUB



## Umrechnung

Umrechnung von Yard/Fuß/  
Zoll und Meter/Centimeter

Dieses Programm ist gerade 1983 unentbehrlich. Sie können in Ihren Besuchen drüben in England blitzschnell alles in Yard, Fuß oder Zoll umrechnen. Das gilt natürlich auch umgekehrt. Um Meter in Yard umzurechnen, benutzen Sie m und Newline. Um Yard in Meter umzurechnen, benutzen Sie y und Newline. Dann den umzurechnenden Betrag eingeben, der Computer wird Ihnen das Ergebnis mitteilen.



```

20 LET B$="ANZAHL: "
30 PRINT "TIPPEN SIE M (METE
R IN YARD), Y (YARD IN M), S (Z
U BEENDEN)"
50 LET A$=INKEY$
60 IF A$="M" THEN GOTO 300
70 IF A$="Y" THEN GOTO 300
75 IF A$="S" THEN STOP
90 GOTO 50
100 CLS
105 PRINT B$;"YARD ";
110 INPUT Y
120 PRINT Y,B$;"FUSS ";
130 INPUT F
140 PRINT F,B$;"ZOLL ";
150 INPUT I
160 LET C=INT ((Y*36+F*12+I)*2.
54+.5)
170 LET K=INT (C/100)
180 PRINT I,K;" M ";C-100*K;" C
M"
190 GOTO 30
300 CLS
305 PRINT B$;"M ";
310 INPUT K
320 PRINT K,B$;"CM ";
330 INPUT C
340 LET I=INT ((100*K+C)*.3937+
.5)
350 LET Y=INT (I/36)
360 LET I=I-36*Y
370 LET F=INT (I/12)
380 PRINT C,Y;" YARD ";F;" FUSS
";I-12*F;" ZOLL"
390 GOTO 30
    
```



## Der Spion, der aus der Kälte kam

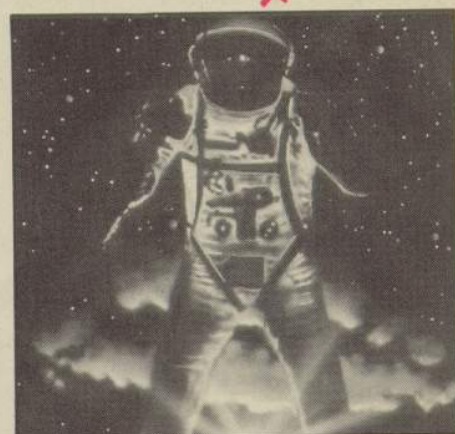
Wie schnell sind Sie, einen Code zu entziffern, den Ihr ZX81 vorgibt? Versuche des Code zu dechiffrieren. Der Computer denkt sich eine vierstellige Zahl aus von eins bis sechs. Keine Ziffer erscheint doppelt. Es dauert ein bißchen, bis Ihr Computer seinen Code zusammengesetzt hat. Geben Sie nur Zahlen ein, Buchstaben bedeuten das Ende des Programms. Der Computer setzt auf jede korrekte Zahl an der richtigen Stelle ein x, ein O für eine richtige Zahl, aber nicht an der richtigen Stelle.

```

10 RAND
20 DIM A$(4)
30 DIM B$(4)
60 FOR N=1 TO 4
70 LET A$(N)=STR$(1+INT (6*RND
D))
80 FOR P=1 TO N
90 IF A$(P)=A$(N) AND N<>P THE
N GOTO 70
100 NEXT P
110 NEXT N
200 PRINT "CODE RESULTAT"
210 FOR X=1 TO 10
220 INPUT B$
230 PRINT "B$;" ";
240 IF B$=A$ THEN GOTO 600
250 FOR N=1 TO 4
260 IF B$(N)<>A$(N) THEN GOTO 2
90
270 PRINT "X";
290 NEXT N
300 FOR N=1 TO 4
310 FOR P=1 TO 4
320 IF N=P OR A$(P)<>B$(N) THEN
GOTO 350
330 PRINT "O";
350 NEXT P
360 NEXT N
370 NEXT X
380 PRINT "DER COMPUTER GEW
INNT: ";A$
390 STOP
600 PRINT "RICHTIG"
    
```

## Astronauten- tests

Als zukünftiger Astronaut können Sie hier Ihre Reaktionsgeschwindigkeit testen. Ihr Computer produziert in wahlloser Reihenfolge Buchstaben auf dem Bildschirm. Drücken Sie den Buchstaben, der auf dem Bildschirm erscheint und halten Sie ihn, bis der Computer antwortet. Haben Sie den gleichen Buchstaben gedrückt, reagiert der Computer schneller. Sonst gewöhnt er sich an Ihre Geschwindigkeit.



```

5 RAND
10 PRINT "PRUEFEN SIE IHRE REA
KTION"
20 PRINT "TIPPEN SIE JEDEN B
UCHSTABEN IN"
25 PRINT "DER VERFUEGBAREN ZEI
T"
30 PRINT "DRUECKEN SIE O UM
ZU ENDEN"
40 PRINT "DRUECKEN SIE NEWLI
NE UM ZU"
45 PRINT "BEGINNEN"
50 IF INKEY$="" THEN GOTO 50
90 LET P=100
100 CLS
110 LET A$=CHR$(CODE "A"+INT (
RND*25))
120 PRINT AT 5,5;A$
130 FOR N=1 TO P
140 NEXT N
150 LET B$=INKEY$
160 IF B$="O" THEN GOTO 300
170 IF A$=B$ THEN GOTO 250
180 IF B$="" THEN PRINT "ZU LAN
GSAM"
190 IF B$<>"" THEN PRINT "FALSC
HER BUCHSTABE"
200 LET P=P+5
210 FOR N=1 TO P
220 NEXT N
230 GOTO 100
250 PRINT "GUT"
260 LET P=P-5
270 GOTO 210
300 PRINT "SIE HABEN ";(400-P)/
4;" PUNKTE"
310 PRINT "ERREICHT."
    
```



# PROGRAMME

## Das Coingame



Nachdem Sie das Programm eingegeben und mit RUN gestartet haben, erklärt sich das Programm von selbst. Der Superspielautomat auf Ihrem Bildschirm.

Axel Blume, Düsseldorf

ZX-81 16K

```

1 REM COINGAME
2 CLS
3 PRINT "COINGAME AXEL BLUME
SEPT.83"
4 PRINT "GESPIELT WIRD JEWEIL
5 MIT 10 CENTS"
10 SLOW
60 LET GE=0
61 LET ANZ=0
62 LET U=0
65 PRINT
70 PRINT TAB 24;"SPIEL NR"
75 PRINT AT 7,9;" "
76 PRINT AT 8,9;" "
77 PRINT AT 10,14;" "
78 PRINT AT 12,9;" "
80 PRINT AT 9,9;" "
85 PRINT AT 11,9;" "
90 PRINT AT 13,9;" "
100 GOSUB 2000
110 REM SCHLEIFE
115 GOSUB 1200

```

```

216 IF U<=0 THEN GOTO 100
220 PRINT AT 4,0;"RESERVE $"
317 LET U=U-1
320 GOSUB 3000
510 LET G=0
515 LET ANZ=ANZ+1
517 PRINT AT 5,25;ANZ
520 GOSUB 1000
529 IF A=B AND A=C AND A=D AND
A=E THEN LET G=10+C
530 IF A=C AND A=B THEN LET G=C
540 IF D=C AND D=E THEN LET G=C
550 IF D=C AND D=B THEN LET G=C
560 IF A=C AND A=E THEN LET G=C
590 IF C>9 AND (A=B OR A=E) THE
N LET G=A
595 IF C>9 AND (D=B OR D=E) THE
N LET G=D
596 IF C>9 AND A=B AND A=D AND
A=E THEN LET G=5+C
650 GOSUB 1100
660 LET U=U+G
672 IF G=0 THEN GOTO 800
673 PRINT AT 15,9;" "
674 PRINT AT 16,9;"SIE BEKOMMEN
"
675 PRINT AT 17,9;" "
676 PRINT AT 16,19;INT (G/10);"
";(G-INT (G/10));"0"
677 GOSUB 3000
820 PRINT AT 21,0;"DRUECKEN SIE
EINE TASTE"
850 IF INKEY$="" THEN GOTO 850
855 IF U=0 THEN GOTO 4000
860 PRINT AT 21,0;" "
900 GOTO 110
1000 REM ABCDE
1020 LET A=INT (9*RND)+1
1030 LET B=INT (9*RND)+1
1040 LET C=INT (9*RND)+1
1050 LET D=INT (9*RND)+1
1060 LET E=INT (9*RND)+1
1070 RETURN
1100 REM PRINT

```

```

1110 PRINT AT 8,10;10*A;AT 8,20;
10*B
1130 PRINT AT 12,10;10*D;AT 12,2
0;10*E
1150 IF C>9 THEN GOTO 1170
1160 PRINT AT 10,15;10+C
1165 RETURN
1170 PRINT AT 10,15;"*"
1180 RETURN
1200 PRINT AT 8,10;" ";AT 8,20;
"
1210 PRINT AT 10,15;" "
1220 PRINT AT 12,10;" ";AT 12,2
0;" "
1230 PRINT AT 15,9;" "
1240 PRINT AT 16,9;" "
1250 PRINT AT 17,9;" "
1260 PRINT AT 4,11;" "
1290 RETURN
2000 PRINT AT 21,0;"WIEVIEL CENT
STARTKAPITAL?"
2010 INPUT U
2015 IF U<=0 THEN GOTO 2010
2020 PRINT AT 21,0;" "
2040 LET GE=GE+U
2050 LET GED=INT (GE/10)
2055 LET GEC=(GE-10*GED)
2056 PRINT AT 3,0;"STARTKAP. $"
2050 PRINT AT 3,12;GED;" ";GEC;"
0 "
2100 RETURN
3000 LET UD=INT (U/10)
3010 LET UC=(U-10*UD)
3030 PRINT AT 4,12;UD;" ";UC;"0
"
3040 RETURN
4000 CLS
4005 PRINT ,,,TAB 10;"GAME OVER
"
4010 PRINT ,,,TAB 6;"NEUES SPIEL?
(J/N)"
4020 INPUT A$
4030 IF A$(1)="J" THEN RUN

```

ZX-USER CLUB



# PROGRAMME

Hunde, wollt Ihr ewig leben

ZX-81 16K



Ein nicht ganz ernstzunehmendes Programm, aber doch sehr unterhaltend. Sie geben alle Ihre Werte ein, z. B. wieviel Zigaretten Sie täglich rauchen, Ihren Alkoholkonsum, ob Sie männlich oder weiblich sind ect. ect. und der ZX 81 stellt eine Ferndiagnose und sagt Ihnen, wie hoch Ihre Lebenserwartung ist.

```

10 LET H$="M"
20 LET N$="F"
30 RAND
40 PRINT "SCHAUEN WIR MAL
WIE IHRE"
50 PRINT TAB (8); "LEBENSERWART
UNG IST"
60 PRINT
90 PRINT "ZUERST BITTE IH
REN NAMEN"
100 INPUT T$
110 CLS
120 PRINT
140 PRINT "OK ";T$; ", WIE LAUT
ET IHR"
150 PRINT "GEBURTSJAHR? (ANTWOR
T Z.B. 1964)"
160 INPUT A
170 GOSUB 2000
180 PRINT "MAENNLICH (1) ODER WE
IBLICH (2)?"
190 INPUT B
200 GOSUB 2000
210 PRINT "IN WELCHER ALTERSGRU
PPE SIND SIE"
220 PRINT "GEBEN SIE EINEN BUCH
STABEN EIN"

```

```

230 PRINT " 5 BIS 25 - A", "26 B
IS 40 - B", "41 BIS 50 - C", "51 B
IS 60 - D", "61 BIS 65 - E", "66 B
IS 70 - F", "71 BIS 75 - G", "UEB
ER 75 - H"
240 INPUT A$
250 GOSUB 2000
260 PRINT "IST/WURDE IHR VATER
UEBER 70? (J/N)"
270 INPUT B$
280 GOSUB 2000
290 PRINT "IST/WURDE IHRE MUTTE
R UEBER 70? (J/N)"
300 INPUT C$
310 GOSUB 2000
320 PRINT "SIND SIE VERHEIRATET
(J/N)?"
330 INPUT D$
340 GOSUB 2000
350 PRINT TAB (8); "WO WOHNEN SI
E?", "KLEINSTADT - A", "
GROSSTADT - B"
360 INPUT E$
370 GOSUB 2000
380 PRINT "WAREN SIE IM LEBEN M
EIST REICH (J) ODER ARM (
N)?"
390 INPUT F$
400 IF CODE (A$) < 40 THEN GOTO 4
70
410 GOSUB 2000
420 PRINT "HABEN SIE UEBERGEWIC
HT (J/N)"
430 INPUT G$
435 PRINT
440 IF CODE (G$) < 47 THEN GOTO
470
450 PRINT TAB (8); "EIN WENIG -
A"
452 PRINT TAB (8); "MITTLERES -
B"
454 PRINT TAB (8); "HOHES - C"
460 INPUT H$
470 GOSUB 2000
480 PRINT "BEWEGUNG..."
485 PRINT "*****"
490 PRINT
500 PRINT
510 PRINT
520 PRINT "WIEVIEL BEWEGUNG HAB
EN SIE?"
522 PRINT TAB (8); "SEHR WENIG -
A"
524 PRINT TAB (8); "MITTEL - B"
526 PRINT TAB (8); "VIEL - C"
530 INPUT J$
540 GOSUB 2000
550 PRINT "SIND SIE GEWOEHNLICH
:"
552 PRINT TAB (5); "GUTHUETIG UN
D GELASSEN - A"
554 PRINT TAB (5); "ANGESpannt U
ND NERVUES - B"
556 PRINT TAB (5); "DAZWISCHEN -
C"
570 INPUT K$
580 GOSUB 2000
585 PRINT "
590 PRINT "TRINKEN"
592 PRINT "
600 PRINT
610 PRINT "WIE HAEUFIG TRINKEN
SIE:"
611 PRINT "KAUM ODER NIE -
A"
612 PRINT "GELEGENTLICH - B"
613 PRINT "REGELMAESSIG (MA
ESSIG) - C"
614 PRINT "REGELMAESSIG (VI
EL) - D"
615 PRINT "REGELMAESSIG (SE
HR VIEL) - E"
620 INPUT L$
630 GOSUB 2000
640 PRINT "RAUCHEN SIE? (J ODE
R N)"
650 INPUT M$
660 CLS
670 IF CODE (M$) < 47 THEN GOTO
730
675 PRINT
676 PRINT
677 PRINT
680 PRINT "10 - 20 ZIGARETTEN
PRO TAG - A"
690 PRINT "20 - 30 ZIGARETTEN
PRO TAG - B"
700 PRINT "UEBER 30 PRO TA
G"

```

```

710 PRINT "PFEIFE ODER ZIG
ARRE?"
720 INPUT N$
730 GOSUB 2000
740 PRINT "GEHEN SIE HINDERSTE
NS ZUEIMAL"
750 PRINT "PRO JAHR ZUM ZAHNAR
ZT (J/N)"
755 INPUT P$
760 GOSUB 2000
770 PRINT "LASSEN SIE SICH REGE
LMAESSIG AERZLICH UNTERSUCHEN
(J/N)"
780 INPUT Q$
790 GOSUB 2000
800 PRINT "SIND SIE OFT KRANK?
(J/N)"
810 INPUT R$
820 LET L=48*(A<1911)+52*(A>191
0 AND A<1921)+59*(A>1920 AND A<1
931)+61*(A>1930 AND A<1941)+65*(
A>1940 AND A<1951)+67*(A>1950 AN
D A<1961)+68*(A>1960)
830 IF B=2 THEN LET L=51*(L=48)
+56*(L=52)-62*(L=59)+67*(L=61)+7
1*(L=65)+74*(L=67)+75*(L=68)
835 LET L4=1990-A
840 LET V=CODE (A$)-37
850 LET L1=3*(V=1)+5*(V=3 OR V=
6)+7*(V=4)+8*(V=5)+10*(V=6)+12*(
V=7)
860 LET L=L+L1
870 LET L=L+(CODE (B$)=47)
880 LET L=L+(CODE (C$)=47)
890 LET L=L+3*(CODE (D$)=47)
900 LET L=L+4*(CODE (E$)=38)-2*(
CODE (E$)=39)
910 LET L=L-3*(CODE (F$)=47)
920 LET L=L-(CODE (H$)=38)-3*(C
ODE (H$)=39)-5*(CODE (H$)=40)
930 LET L=L+3*(CODE (J$)=39)+5*(
CODE (J$)=40)
940 LET L=L+3*(CODE (K$)=38)-2*(
CODE (K$)=39)
950 LET L=L+3*(CODE (L$)=40)-5*(
CODE (L$)=41)-10*(CODE (L$)=42)
960 LET L=L-3*(CODE (N$)=38)-5*(
CODE (N$)=41)
965 IF L<L4 THEN LET L=L4
970 IF P$="J" THEN LET L=L+1
980 IF Q$="J" THEN LET L=L+1
990 IF R$="J" THEN LET L=L-1
1000 CLS
1010 FOR Z=1 TO 5
1020 PRINT
1030 NEXT Z
1040 PRINT "STATISTISCH GES
EHEN"
1050 PRINT
1060 PRINT TAB (8);T$; ", IST ";L
1070 PRINT
1080 PRINT "IHR VORAUSSICHTL
ICHES"
1085 PRINT "STERBEALTER"
1090 PRINT
1100 PRINT
1110 PRINT "
1999 STOP
2000 CLS
2010 FOR Z=1 TO INT (RND*8)
2020 PRINT
2030 NEXT Z
2040 GOSUB 3000
2070 RETURN
3000 LET Z=INT (RND*10)
3005 PRINT TAB (2);
3010 GOSUB 3000+20*Z
3015 RETURN
3020 PRINT "AHA"
3030 RETURN
3040 PRINT "SOSO"
3050 RETURN
3060 PRINT "GUT"
3070 RETURN
3080 PRINT "OK"
3090 RETURN
3100 PRINT "SCHOEN, ";T$;
3110 RETURN
3120 PRINT "DANKESCHOEN, ";T$; ",
50..."
3130 RETURN
3140 PRINT "NICHT MEHR BESONDERS
LANGE HIN"
3150 RETURN
3160 PRINT "MMM..."
3170 RETURN
3180 PRINT "DANKE"
3190 RETURN
3200 PRINT "ALSO DANN"
3210 RETURN

```

ZX-USER CLUB



# PROGRAMME

## Frogger wie Frosch



Wir haben das Jahr 3000. Die Erde steht noch, aber die Umweltverschmutzung ist verheerend. Da der Fluß in diesem Spiel auch verseucht ist und die Straße voller Autos ist, müssen Sie den Frosch zu seiner Höhle bringen. Sie dürfen nicht in ein Auto »I« Invers rennen, nicht in den Fluß fallen und auch nicht ein »Graphikzeichen« oder ein »+« als Transportmittel benutzen. An der Uferzone läuft eine gefräßige Schlange herum, die sich die Zeit mit Fröschevertilgen vertreibt. Dieses abendfüllende Programm schickte uns Herr Martin Lange, Preetz.

```

1 REM 325 BELIEBIGE ZEICHEN
2 REM (C)BY M.LANGE 1983
2300 PRETZ,WILH.-RAABE-STR.92 A
ABDRUCK NUR MIT SCHRIFTLICHER
ERLAUBNIS DES AUTORS
10 GOSUB 8000
15 LET HS=0
20 GOSUB 7000
40 LET S=0
45 LET R=1
50 LET B=INT (RND*15)+5
60 LET A=22
90 LET F=5-R
91 IF F<=0 THEN LET F=1
95 LET FR=0
100 DIM Z(5)
105 LET T=20
110 POKE 16417,138
115 LET OS=" "
120 LET T=20
125 GOSUB 6000
130 RAND USR 16762
202 LET T=T-.05
203 IF T<=0 THEN GOTO 700
205 PRINT AT 23,5;0$( TO T);" "
220 LET I$=INKEY$
225 PRINT AT A,B;(CHR$ PEEK 164
17 AND (I$<>" " OR A>14))
226 RAND USR 16582
230 LET A=A+(2 AND I$="6")-(2 A
ND I$="7")
231 IF A=24 THEN LET A=22
235 LET B=B+(2 AND I$="9" AND B
30)+(B<0 OR I$="8")-(B>30 OR I$
"5")
236 RAND USR 16641
237 LET B=B+(A=14 OR A=10)-(A=8
OR A=12)
238 RAND USR 16787
240 PRINT AT A,B;
245 LET L=USR 16773
250 PRINT AT A,B;CHR$ 189
251 IF A=6 THEN GOTO 600
255 IF L=150 OR L=0 OR L=174 OR
L=131 OR L=21 THEN GOTO 500
260 IF T<=0 THEN GOTO 700
270 GOTO 200

```

```

500 FOR I=1 TO 5
505 PRINT AT A,B;"X"
510 LET X=SGN PI
515 PRINT AT A,B;"X"
520 LET X=SGN PI
525 NEXT I
530 PRINT AT A,B;CHR$ PEEK 1641
7
535 LET F=F-1
540 LET A=22
545 LET B=INT (RND*15)+5
550 GOSUB 6000
555 IF F=0 THEN GOTO 700
556 POKE 16417,138
560 GOTO 200
600 LET N=0
605 LET N=N+(1 AND B=3)+(2 AND
B=9)+(3 AND B=15)+(4 AND B=21)+(
5 AND B=27)
610 IF N=0 THEN GOTO 500
611 PRINT AT A,B;" "
615 IF Z(N)=1 THEN GOTO 500
616 PRINT AT 3,B-1;"X";AT 4,B
-1;"X";AT 5,B-1;"X";AT 2,B-1
;R=100
620 LET Z(N)=1
625 LET S=S+R*100
630 LET FR=FR+1
635 IF FR=5 THEN GOTO 800
640 GOTO 540
700 PRINT AT 10,10;" GAME OVER

```

```

705 IF S>HS THEN LET HS=S
706 GOSUB 6000
710 PRINT AT 12,10;"NOCHMAL ?";
TAB 10;"TASTE DRUECKEN"
715 IF INKEY$="" THEN GOTO 715
720 GOTO 20
800 GOSUB 7000
805 LET R=R+1
810 GOTO 50
2000 LET A=16514
2005 LPRINT A,PEEK A
2010 LET A=A+1
2015 IF A=16839 THEN STOP
2020 GOTO 2005
3000 IF INKEY$="" THEN GOTO 3000
3010 RAND USR 16762
3020 GOTO 3000
6000 PRINT AT 0,7;S;AT 0,19;HS;A
T 0,30;F
6005 PRINT AT 3,30;R
6010 RETURN
7000 PRINT AT 0,0;
7001 POKE 16418,0
7005 PRINT "SCORE> HIGH>
FROG> "
7010 PRINT "
7015 PRINT "
7020 PRINT "
7025 PRINT "
7030 PRINT "
7035 PRINT "
7040 PRINT "
7045 FOR N=8 TO 15
7050 PRINT AT N,0;"

```

```

7055 NEXT N
7060 PRINT "
7065 FOR N=17 TO 21
7070 PRINT "
7075 NEXT N
7080 PRINT "
7085 PRINT AT 23,0;"TIME
7090 RETURN
8000 PRINT " F R O G G
E R
8005 PRINT "WIR HABEN DAS JAHR 3
000.DIE ERDESTEHT NOCH,ABER DIE
UMWELTVER- SCHMUTZUNG IST SCHLI
MH.DA DER FLUSS IN DIESEM SPIE
L AUCH VER- SEUCHT IST UND DIE S
TRASSE VOLL AUTOS MUESSEN SIE DE
N FROSCH ZU SEINER HOEHLE BRI
NGEN."
8010 PRINT "SIE DUERFEN NICHT IN
EIN AUTO RENNEN,NICHT IN DEN
DRECKFLUSS FALLEN UND AUCH NICH
T EIN - ODER+ ALS TRANSPORTMITTE
L BENUTZEN"

```

```

8015 PRINT "AUF DER UFERZONE LAE
UFT EINE GE-FRAESSIGE SCHLANGE R
UM,DIE SICH DIE ZEIT MIT FROESCH
EVERTILGEN VERTREIBT...."
8020 PRINT "STEUERUNG:"
8025 PRINT "5"" 1 LINKS","""
6"" 2 RUNTER","""7"" 2 HOCH",
,"""8"" 1 RECHTS","""9"" 2 RE
CHTS"
8030 PRINT AT 20,0;"TASTE FUER":
AT 21,0;"START DRUECKEN"
8040 IF INKEY$="" THEN GOTO 8040
8050 RETURN
9000 SAVE "FROGGE"
9005 RUN

```

16514	0	00
16515	21	15
16516	1	01
16517	131	83
16518	1	01
16519	52	34
16520	1	01
16521	136	88
16522	237	ED
16523	91	5B
16524	52	34
16525	64	40
16526	203	CB
16527	178	B2
16528	203	CB
16529	186	BA
16530	133	21
16531	130	82
16532	64	40
16533	6	06
16534	8	08
16535	26	1A
16536	160	AO
16537	184	B8
16538	40	28
16539	4	04
16540	54	36
16541	0	00
16542	24	18
16543	2	02
16544	54	36
16545	1	01
16546	35	23
16547	35	23
16548	19	13
16549	5	05
16550	16	10
16551	239	EF
16552	201	C9
16553	205	CD
16554	138	8A
16555	64	40
16556	6	06
16557	3	03
16558	30	21
16559	130	82
16560	64	40
16561	126	7E
16562	254	FE
16563	0	00
16564	40	28
16565	11	0B
16566	35	23
16567	126	7E
16568	79	4F
16569	35	23
16570	35	23
16571	126	7E
16572	113	71
16573	43	2B
16574	43	2B
16575	119	77
16576	43	2B
16577	35	23
16578	35	23
16579	16	10
16580	236	EC
16581	201	C9
16582	205	CD
16583	138	8A
16584	64	40



# PROGRAMME

165885	42	2A	16670	119	77	16755	24	18
165886	12	0C	16671	24	18	16756	2	02
165887	64	40	16672	20	00	16757	54	36
165888	1	01	16673	54	36	16758	0	00
165889	83	53	16674	0	00	16759	201	09
165890	2	02	16675	1	01	16760	15	0F
165891	9	09	16676	66	42	16761	82	52
165892	84	54	16677	0	00	16762	42	2A
165893	93	5D	16678	9	09	16763	12	0C
165894	35	23	16679	64	54	16764	64	40
165895	1	01	16680	93	5D	16765	1	01
165896	31	1F	16681	43	2B	16766	32	20
165897	0	00	16682	1	01	16767	0	02
165898	237	ED	16683	31	1F	16768	9	09
165899	176	B0	16684	0	00	16769	34	22
166000	43	2B	16685	237	ED	16770	120	78
166001	58	3A	16686	184	B8	16771	65	41
166002	130	82	16687	35	23	16772	201	09
166003	64	40	16688	58	3A	16773	42	2A
166004	254	FE	16689	132	84	16774	14	0E
166005	1	01	16690	64	40	16775	64	40
166006	40	28	16691	254	FE	16776	126	7E
166007	4	04	16692	1	01	16777	254	FE
166008	54	36	16693	32	20	16778	189	BD
166009	128	80	16694	6	06	16779	200	08
166010	24	18	16695	58	3A	16780	6	06
166011	2	02	16696	133	85	16781	0	00
166012	54	36	16697	64	40	16782	79	4F
166013	174	DE	16698	119	77	16783	50	32
166014	1	01	16699	24	18	16784	33	21
166015	66	42	16700	2	02	16785	64	40
166016	0	00	16701	54	36	16786	201	09
166017	9	09	16702	0	00	16787	205	0D
166018	84	54	16703	1	01	16788	138	8A
166019	93	5D	16704	66	42	16789	64	40
166020	43	2B	16705	0	00	16790	42	2A
166021	1	01	16706	9	09	16791	120	78
166022	31	1F	16707	64	54	16792	65	41
166023	0	00	16708	93	5D	16793	209	E5
166024	237	ED	16709	35	23	16794	58	3A
166025	184	B8	16710	1	01	16795	132	84
166026	35	23	16711	31	1F	16796	64	40
166027	58	3A	16712	0	00	16797	254	FE
166028	132	84	16713	237	ED	16798	0	00
166029	64	40	16714	176	B0	16799	40	28
166030	254	FE	16715	43	2B	16800	3	03
166031	1	01	16716	58	3A	16801	35	23
166032	40	28	16717	134	86	16802	24	18
166033	4	04	16718	64	40	16803	1	01
166034	54	36	16719	254	FE	16804	43	2B
166035	128	80	16720	1	01	16805	126	7E
166036	24	18	16721	32	20	16806	254	FE
166037	2	02	16722	66	42	16807	118	76
166038	54	36	16723	58	3A	16808	40	28
166039	174	DE	16724	135	87	16809	9	09
166040	201	09	16725	64	40	16810	54	36
166041	205	0D	16726	119	77	16811	150	96
166042	169	A9	16727	24	18	16812	34	22
166043	64	40	16728	2	02	16813	120	78
166044	42	2A	16729	54	36	16814	65	41
166045	12	0C	16730	0	00	16815	205	E1
166046	64	40	16731	1	01	16816	54	36
166047	1	01	16732	66	42	16817	137	89
166048	9	09	16733	0	00	16818	201	09
166049	1	01	16734	9	09	16819	205	E1
166050	9	09	16735	84	54	16820	34	22
166051	84	54	16736	93	5D	16821	120	78
166052	93	5D	16737	43	2B	16822	64	40
166053	35	23	16738	1	01	16823	58	3A
166054	1	01	16739	31	1F	16824	158	9E
166055	31	1F	16740	0	00	16825	65	41
166056	0	00	16741	237	ED	16826	254	FE
166057	237	ED	16742	184	B8	16827	1	01
166058	176	B0	16743	35	23	16828	32	20
166059	43	2B	16744	58	3A	16829	4	04
166060	58	3A	16745	136	88	16830	62	3E
166061	130	82	16746	254	40	16831	0	00
166062	64	40	16747	254	FE	16832	24	18
166063	254	FE	16748	1	01	16833	2	02
166064	1	01	16749	32	20	16834	62	3E
166065	32	20	16750	66	42	16835	1	01
166066	6	06	16751	58	3A	16836	50	32
166067	58	3A	16752	137	89	16837	158	9E
166068	131	83	16753	64	40	16838	65	41
166069	64	40	16754	119	77	16839	201	09



# DIE FASZINATION DES FORTSCHRITTS

# SINCLAIR

## Das sind Profi-Leistungen:

- 8 Farben für Vorder- und Hintergrund, wahlbar für Border, Paper und Ink, dazu Flash, Bright, Over und inverse. Farben können im Programm beliebig eingesetzt werden, ohne das Programm zu beeinflussen.
- Ton: BEEP-Kommando auf eingebauten Lautsprecher oder Interface über drei Oktaven und in frei wählbarer Länge.
- Bedienungsfreundliche, großformatige Tastatur mit 40 beweglichen Tippasten und bis zu 6 Funktionen pro Taste.
- Kompletter ASCII-Zeichensatz mit Groß- und Kleinschreibung.
- Hochauflösende Grafik: durch 256 Punkte waagrecht und 192 senkrecht, gestochen scharf. Feste Kommandos zum Zeichnen von Punkt, Linie, Kreis und Bogen. Grafik und Schrift kann beliebig gemischt werden.
- LOAD/SAVE in Hochgeschwindigkeit. 16K Übertragung von und auf Kassetten in unglaublichen 100 Sekunden. Durch VERIFY kann das Abspeichern sofort kontrolliert werden.
- Programmier-Sprachen: BASIC und über die PEEK + POKE-Taste in Maschinensprache (Z80 A) programmierbar.
- Mathematische Funktionen und Operationen wie im Großrechner. Neben den Rechenfunktionen Sinus, Cosinus, Tangens und ihre Umkehrfunktionen, natürlicher Logarithmus, Exponentialfunktionen, sign, abs, Integerfunktion, Quadratwurzel, Zufallsgenerator, Pi. Zahlenbereich von  $+3 \times 10^{-39}$  bis  $+7 \times 10^{38}$ . Binäre Zahlen können direkt im BIN-Modus eingegeben werden. Logische Operationen AND, OR, NOT bringen

Boolesche Resultate. Mit DEF FN sind Funktionen frei programmierbar, mit FN werden sie abgerufen. Bis zu 26 numerische und 26 alphanumerische Funktionen können definiert werden.

- Völlig neu gestaltetes Kassetten-Interface. Vor den Informationen wird eine Tonführung aufgenommen, um automatischen Schwund aufzufangen. Nebengeräusche werden mit einem Schmitt-Trigger ausgefiltert. Alle Nachrichten beginnen mit einer Kopfzeile, die Informationen über Art, Titel und Länge der Adresse enthält. Programme, Bildschirmhalte, Daten und Tabellen können jeweils separat abgespeichert werden. Jede Eingabe läßt sich vor dem Löschen aus dem Computer mit VERIFY überprüfen. — Fehler bei der Übertragung auf und Rückspiegelung von der Kassette gehören damit endgültig der Vergangenheit an.

... zum Preis eines Personal-Computers.

Der neue SINCLAIR ZX-SPECTRUM kostet mit 2 deutschen Handbüchern betriebsfertig DM 498,- (mit dem 16K RAM-Speicher), die 48K RAM-Version DM 698,-. Dafür erhalten Sie nicht nur einen Computer mit Zukunft, sondern auch eine Anlage mit fast unbegrenzten Ausbaumöglichkeiten.

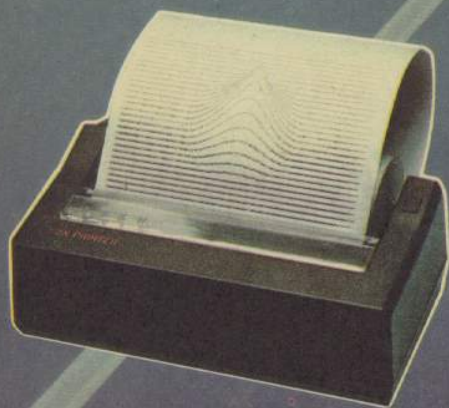
## Die Zukunft ist schon vorprogrammiert.

- ZX-Erweiterungsmodul, mit dem bis zu 8 Mikrolaufwerke oder eine ganze Anzahl von Druckern gesteuert werden können. Sie können endlich mit anderen Computern kommunizieren und eröffnen sich damit eine völlig neue Dimension der Programmgestaltung und -benutzung. (Ab Ende 1983 lieferbar).

- ZX-Mikrodrive: ein speziell für den SPECTRUM entwickeltes Mikrolaufwerk für die Speicherung bis zu 100K-Bytes, mit einer Übertragungsrate von 16K-Bytes pro Sekunde und einer Zugriffsgeschwindigkeit von ca. 3,5 Sek. (Ab Ende 1983 lieferbar).

- ZX-Printer zum Freundschaftspreis von DM 249,-, der 50 Zeichen pro Sekunde, 32 Zeichen pro Zeile, hochauflösende Grafik etc. druckt.

- Und damit Sie auch ohne zeitraubendes Programmieren und umfangreiches Programmierfachwissen alle Möglichkeiten des ZX Spectrum voll nutzen können, bringt Sinclair ein Software-Paket, das keine Wünsche offen läßt. Von unglaublichen Spielen mit bewegter Grafik (z.B. Flug-simulation, Welt-raum-spiele, Schach) bis zu ausgetüftelten Dateien, von fantastischen Grafikmodellen bis zu kompliziertesten wissenschaftlichen Mathematikprogrammen ist alles geboten. — Wir informieren Sie gerne mit ausführlichem Prospektmaterial.



ZX-Printer für nur DM 249,-



ZX-Mikrodrive: sagenhafte 100K-Bytes auf einem Speichermedium (Ende 1983 lieferbar).



ZX-SPECTRUM-Handbücher in Deutsch liegen jedem Computer gratis bei.

Wenn Sie in München wohnen: Unser Computershop, Aventinstraße 6 (5 Minuten vom Marienplatz), 8000 München 5



# SINCLAIR ZX SPECTRUM



## BON So bekommen Sie Ihren ZX-SPECTRUM

Hiermit bestelle ich  
☐ per Nachnahme ☐ per Vorausscheck  
 Preise incl. MwSt., zuzügl. DM 7,50  
 Porto und Verpackung bzw.  
 Nachnahmegebühren. Ich habe  
 6 Monate Original-SINCLAIR-Garantie  
 und bezahle wie oben angekreuzt.

General-Importeur Jürgen Schumpich · Internat. Industrievertretungen GmbH  
 Postfach 6352, 8012 Ottobrunn

Anzahl	Artikel	
<input type="text"/>	Sinclair SPECTRUM 16K-RAM-Version	DM 498,-
<input type="text"/>	Sinclair SPECTRUM 48K-RAM-Version	DM 698,-
<input type="text"/>	Sinclair ZX-Drucker	DM 249,-
<input type="text"/>	Ausführlicher Prospekt	gratis

Name   
 Straße   
 PLZ/Ort   
 Datum   
 Unterschrift

Abt.



# PROGRAMME

## SHIFT



Versuchen Sie über die sich quer verschiebenden Bänder das Spielfeld heraufzuklettern. Die Steuerung ist: nach oben: 7, nach unten: 6. Aber Vorsicht, fallen Sie nicht vom Band... Viel Spaß.

M. Gehrke, Weinstadt.

ZX-B1 16k

```

1 REM 42 BELIEBIGE ZEICHEN
10 REM =====SHIFT=====
20 REM (C)BY MARTIN GEHRKE
23 GOSUB 5000
25 RAND
30 GOSUB 600
60 LET M=INT (RND*15)+7
62 LET N=20
64 LET X1=INT (RND*2)
65 LET X2=INT (RND*2)
66 IF X1=X2 THEN GOTO 65
67 LET X3=INT (RND*2)
70 FOR S=1 TO 10
80 PRINT AT N,M;"*"
90 GOTO 130
100 LET A=INKEY#
110 IF A="7" THEN GOSUB 1000
120 IF A="6" THEN GOSUB 2000
123 RETURN
130 LET A=N-1
140 GOSUB X1*100+400
145 GOSUB 100
150 LET A=N
160 GOSUB X2*100+400
165 GOSUB 100
170 LET A=N+1
240 GOSUB X3*100+400
    
```

```

250 GOSUB 100
260 NEXT S
270 GOTO 64
400 LET F=(33*A)+2
410 POKE 16539,((F/256)-INT (F/
256))*256
420 POKE 16540,INT (F/256)
430 RAND USR 16535
435 IF N=A THEN LET M=M-1
436 IF M<=1 OR M>=30 THEN GOTO
4000
440 RETURN
500 LET F=(33*A)+31
510 POKE 16518,((F/256)-INT (F/
256))*256
520 POKE 16519,INT (F/256)
530 RAND USR 16514
535 IF N=A THEN LET M=M+1
536 IF M>=30 OR M<=1 THEN GOTO
4000
540 RETURN
600 FAST
610 FOR Q=1 TO 704
620 PRINT "■";
630 NEXT Q
640 FOR Q=0 TO 21
650 PRINT AT Q,INT (RND*30)+1;"
"
660 PRINT AT Q,INT (RND*30)+1;"
"
670 NEXT Q
675 SLOW
680 RETURN
1000 REM 7
1010 IF N<=1 THEN GOTO 3000
    
```

```

1020 PRINT AT N-1,M)
1025 LET N1=N
1030 IF PEEK (PEEK 16398+256*PEE
K 16399)=0 THEN LET N=N-1
1035 IF N1=N THEN GOTO 1050
1040 PRINT AT N,M)"*";AT N+1,M)"
"
1044 IF N1=N THEN GOTO 1050
1045 LET X2=X1
1046 LET X3=X2
1047 LET X1=X3
1050 RETURN
2000 REM 6
2010 IF N>20 THEN GOTO 4000
2020 PRINT AT N+1,M)
2025 LET N1=N
2030 IF PEEK (PEEK 16398+256*PEE
K 16399)=0 THEN LET N=N+1
2035 IF N1=N THEN GOTO 2050
2040 PRINT AT N,M)"*";AT N-1,M)"
"
2044 LET X3=X1
2045 LET X1=X2
2046 LET X2=X3
2050 RETURN
3000 PRINT AT 21,0;"GESCHAFFT..."
"
3010 FOR Q=1 TO 100
3020 NEXT Q
3030 CLS
3100 GOTO 25
4000 PRINT AT 21,0;"HERUNTERGEFA
LLEN..."
4010 FOR Q=1 TO 100
4015 NEXT Q
4020 RUN
5000 REM ERKL.
5005 CLS
5010 PRINT "=====SHIFT====="
5020 PRINT
5030 PRINT "VERSUCHEN SIE UEBER
DIE SICH QUER VERSCHIEBENDEN
BAENDER DAS SPIELFELD HERAUFZUKE
TTERN."
5040 PRINT
5050 PRINT "DIE STEUERUNG IST:"
5060 PRINT "NACH OBEN: ""7""
5070 PRINT "NACH UNTEN: ""6""
5080 PRINT
5090 PRINT "=====
=====
5100 PRINT
5110 PRINT "FALLEN SIE NICHT VOM
BAND...."
5120 PRINT
5130 PRINT "VIEL SPASS - "
5140 FOR Q=1 TO 400
5144 IF INKEY#("<") THEN GOTO 514
7
5145 NEXT Q
5147 CLS
5150 RETURN
6000 SAVE "SHIFT"
6010 RUN
    
```

16514	42
16515	12
16516	64
16517	1
16518	47
16519	2
16520	9
16521	68

## ZX-USER CLUB



# PROGRAMME

```

16522      77
16523      10
16524      84
16525      93
16526      43
16527      1
16528      29
16529      0
16530      237
16531      184
16532      35
16533      119
16534      201
16535      42
16536      12
16537      64
16538      1
16539      84
16540      2
16541      9
16542      68
16543      77
16544      10
16545      84
16546      93
16547      35
16548      1
16549      29
16550      0
16551      237
16552      176
16553      43
16554      119
16555      201

```

```

4082  2A0C40  LD HL,(400C)
4083  012F02  LD BC,022F
4088  09      ADD HL,BC
4089  44      LD B,H
408A  4D      LD C,L
408B  0A      LD A,(BC)
408C  54      LD D,H
408D  5D      LD E,L
408E  2B      DEC HL
408F  011D00  LD BC,001D
4092  EDB8    LDDR
4094  23      INC HL
4095  77      LD(HL),A
4096  C9      RET
4097  2A0C40  LD HL,(400C)
409A  015402  LD BC,0254
409D  09      ADD HL,BC
409E  44      LD B,H
409F  4D      LD C,L
40A0  0A      LD A,(BC)
40A1  54      LD D,H
40A2  5D      LD E,L
40A3  23      INC HL
40A4  011D00  LD BC,001D
40A7  EDB0    LDIR
40A9  2B      DEC HL
40AA  77      LD(HL),A
40AB  C9      RET

```

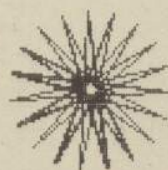
## Muster

Dieses kleine Programm zeichnet lustige Muster in Sternenform auf Ihrem Bildschirm, die Sie natürlich auch ausdrucken können.

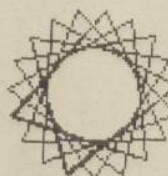
```

10 INPUT a,b,c
20 PLOT 88,122: DRAW a,b,c
30 PRINT AT 21,0;"o.k."
40 PAUSE 0
50 RUN

```



Werte: -40,40,750



Werte: 45,45,500



## Die Sensation – Ein I-K Maschinenprogramm

R. Pecha, Fildestadt

Wichtig: Zuerst tippen Sie bitte das Listing 1 ein und geben dann die Maschinencode hintereinander ein. Anschließend tippen Sie Listing 3 ein.

Sie befinden sich im Weltraum und Ihr Raumschiff (Raumschiff-Stern, wenn getroffen negativ) wird von feindlichen pinball-visitors umzingelt. Sie haben keine Möglichkeit, die feindlichen Angreifer zu bekämpfen, sondern müssen ihnen ausweichen mit den Tasten 5 und 8. Eine Leistungsangabe kann natürlich nicht angezeigt werden, da dieses Programm volle 1 K ausnutzt. Erstaunlich!

### Listing 1

```

1 REM 224 BELIEBIGE ZEICHEN
10 FOR A=16544 TO 16736
20 INPUT B
30 PRINT (STR$(B+1000))$(2 TO
40 IF PEEK 16442<=2 THEN SCROL
L
50 POKE A,B
60 NEXT A

```

### Listing 3

```

8 RAND
9 FAST
10 FOR Z=NOT PI TO VAL "27" ST
EP VAL "3"
20 POKE VAL "16514"+Z,INT (RND
*VAL "3")+VAL "4"
21 POKE VAL "16515"+Z,INT (RND
*VAL "8")+VAL "4"
22 POKE VAL "16516"+Z,INT (RND
*VAL "4")
30 NEXT Z
40 POKE VAL "16417",NOT PI
45 SLOW
50 RAND USR VAL "16544"
60 PRINT "■"
70 RAND RND**RND**RND

```

### Listing 2

16544	6	06
16545	16	10
16546	197	C5
16547	14	0E
16548	0	00
16549	205	CD
16550	245	F5
16551	8	08
16552	62	3E
16553	8	08
16554	215	D7
16555	193	C1
16556	197	C5
16557	14	3E
16558	16	10
16559	205	CD
16560	245	F5
16561	8	08
16562	62	3E
16563	8	08
16564	215	D7
16565	193	C1
16566	16	10
16567	234	EA
16568	14	0E
16569	16	10
16570	197	C5
16571	6	06
16572	0	00
16573	205	CD
16574	245	F5
16575	8	08
16576	62	3E



# PROGRAMME

16577	8	08	16634	35	23	16690	14	0E
16578	215	D7	16635	121	79	16691	64	40
16579	193	C1	16636	119	77	16692	126	7E
16580	197	C5	16637	35	23	16693	254	FE
16581	6	06	16638	254	FE	16694	52	34
16582	16	10	16639	1	01	16695	193	C1
16583	205	CD	16640	32	20	16696	200	C8
16584	245	F5	16641	2	02	16697	54	36
16585	8	08	16642	203	CB	16698	0	00
16586	62	3E	16643	142	8E	16699	58	3A
16587	8	08	16644	254	FE	16700	37	25
16588	215	D7	16645	15	0F	16701	64	40
16589	193	C1	16646	32	20	16702	254	FE
16590	13	0D	16647	2	02	16703	247	F7
16591	121	79	16648	203	CB	16704	32	20
16592	254	FE	16649	206	CE	16705	4	04
16593	255	FF	16650	120	78	16706	13	0D
16594	32	20	16651	254	FE	16707	32	20
16595	230	E6	16652	1	01	16708	1	01
16596	33	21	16653	32	20	16709	12	0C
16597	130	82	16654	2	02	16710	254	FE
16598	64	40	16655	203	CB	16711	239	EF
16599	126	7E	16656	134	86	16712	32	20
16600	71	47	16657	254	FE	16713	9	09
16601	35	23	16658	15	0F	16714	12	0C
16602	126	7E	16659	32	20	16715	62	3E
16603	79	4F	16660	2	02	16716	16	10
16604	197	C5	16661	203	CB	16717	185	B9
16605	229	E5	16662	198	C6	16718	32	20
16606	205	CD	16663	229	E5	16719	1	01
16607	245	F5	16664	205	CD	16720	13	0D
16608	8	08	16665	245	F5	16721	0	00
16609	62	3E	16666	8	08	16722	0	00
16610	0	00	16667	62	3E	16723	121	79
16611	215	D7	16668	52	34	16724	61	3D
16612	225	E1	16669	215	D7	16725	50	32
16613	193	C1	16670	225	E1	16726	33	21
16614	35	23	16670	225	A0	16727	64	40
16615	126	7E	16671	35	23	16728	205	CD
16616	203	CB	16672	62	3E	16729	245	F5
16617	71	47	16673	160	A0	16730	8	08
16618	40	28	16674	189	BD	16731	62	3E
16619	2	02	16675	194	C2	16732	23	17
16620	5	05	16676	215	D7	16733	215	D7
16621	5	05	16677	64	40	16734	195	C3
16622	4	04	16678	58	3A	16735	212	D4
16623	203	CB	16679	33	21	16736	64	40
16624	79	4F	16680	64	40			
16625	40	28	16681	60	3C			
16626	2	02	16682	79	4F			
16627	13	0D	16683	6	06			
16628	13	0D	16684	8	08			
16629	12	0C	16685	197	08			
16630	43	2B	16686	205	CD			
16631	43	2B	16687	245	F5			
16632	120	78	16688	8	08			
16633	119	77	16689	42	2A			

ZX-81 1K

## Hexadezimalumwandlung

Das kürzeste Programm, das dezimale Werte in hexadezimale Form umrechnet.

```

10 INPUT X
20 PRINT CHR$(INT (X/16)+28);
CHR$(X-INT (X/16)*16+28)
30 GO TO 10

```

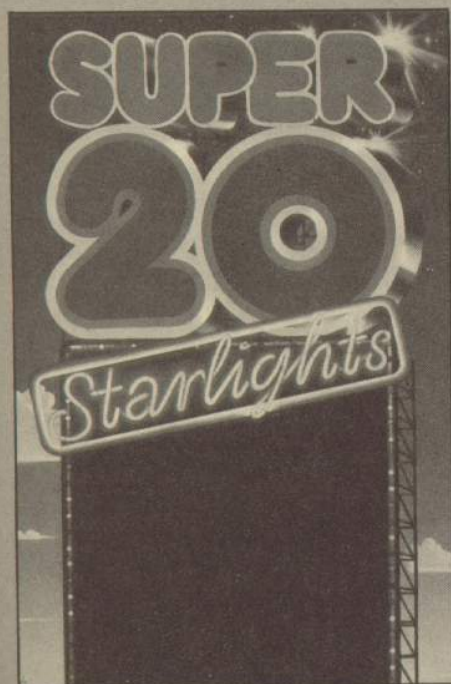
**ZX-USER CLUB**



# PROGRAMME

## Vier gewinnt

Waagrecht, senkrecht oder diagonal



Geben Sie dieses Programm ein. Start mit RUN 2000, das Programm ist selbsterklärend. Sie müssen versuchen, in einem magischen Quadrat senkrecht, waagrecht oder diagonal eine Reihe aufzubauen.

```

1 DIM B(7,7)
6 CLS
7 FAST
8 LET L=0
9 LET Z=19
10 FOR N=0 TO 24
20 PRINT AT 21,N;CHR# 128;AT 1
7,N;CHR# 22;AT 13,N;CHR# 22;AT 9
,N;CHR# 22;AT 5,N;CHR# 22;AT 1,N
;CHR# 22
40 NEXT N
50 FOR F=20 TO 1 STEP -1
60 PRINT AT F,0;CHR# 128;AT F,
4;CHR# 14;AT F,8;CHR# 14;AT F,12
;CHR# 14;AT F,16;CHR# 14;AT F,20
;CHR# 14;AT F,24;CHR# 128
70 NEXT F
80 LET K=14
82 SLOW
83 PRINT AT 3,25;"WEISS"
84 PRINT AT 5,25;"BEGINNT"
90 PRINT AT 0,K;CHR# 21
100 IF INKEY#="8" THEN LET K=K+
4
102 IF K=2 THEN GOTO 110
103 PRINT AT 0,K-4;" "
105 IF K<2 THEN LET K=2
110 IF INKEY#="5" THEN LET K=K-
4
111 PRINT AT 0,K+4;" "
115 IF K>22 THEN LET K=22
120 IF INKEY#="6" OR INKEY#="7"

```

16K

```

THEN GOTO 170
130 GOTO 90
170 PRINT AT 3,25;" "
172 PRINT AT 5,25;" "
180 IF K=2 THEN LET H=1
181 IF K=6 THEN LET H=2
182 IF K=10 THEN LET H=3
183 IF K=14 THEN LET H=4
184 IF K=18 THEN LET H=5
185 IF K=22 THEN LET H=6
187 IF Z=19 THEN LET Y=1
189 IF Z=15 THEN LET Y=2
190 IF Z=11 THEN LET Y=3
191 IF Z=7 THEN LET Y=4
192 IF Z=3 THEN LET Y=5
230 IF L=1 THEN GOTO 275
240 IF B(Y,H)<>0 THEN LET Z=Z-4
241 LET Y=Y+1
242 IF Z<3 THEN GOTO 600
245 IF B(Y,H)<>0 THEN GOTO 240
250 PRINT AT Z,K;CHR# 8
256 LET B(Y,H)=1
257 LET L=1
260 GOSUB 1000
269 LET Z=19
270 PRINT AT 3,25;"SCHWARZ"
271 PRINT AT 5,25;" AM ZUG"
274 GOTO 90
275 IF B(Y,H)<>0 THEN LET Z=Z-4
276 LET Y=Y+1
278 IF Z<3 THEN GOTO 600
279 IF B(Y,H)<>0 THEN GOTO 275
280 PRINT AT Z,K;CHR# 128
283 LET B(Y,H)=2
285 LET L=0
290 GOSUB 1000
340 LET Z=19
342 PRINT AT 3,25;" WEISS"
345 PRINT AT 5,25;" AM ZUG"
350 GOTO 90
500 CLS
501 PRINT
502 PRINT
503 PRINT "VIER GEWINNT"
504 PRINT "===== "
505 REM (C)1983 R.ZENKER/FUESS.
506 PRINT
507 PRINT "VERSUCHE DIE SPIELS
TEINE SO ZU WERFEN ,DASS SICH EN
TWEDER WAAG-RECHTE,SENKRECHTE OD
ER DIAGONALEVIERERREIHEN ERGEBEN
"
508 PRINT
509 PRINT "DER KURSOR WIRD MIT
DEN TASTEN 5BZW. 8 GESTEUERT.MIT
DEN TASTEN 6 ODER 7 WIRD DER AB
WURF DER STEINE AUSGELOEST."
510 PRINT
513 PRINT "WARTE NUN EIN PAAR S
EKUNDEN BIS DAS SPIELFELD AUFGE
AUT IST"
515 PAUSE 1500
520 CLS
530 RUN
600 PRINT AT 9,25;"TOELPEL"
610 PRINT AT 8,25;"DU"
613 FOR N=0 TO 20
614 NEXT N
617 PRINT AT 9,25;" "
618 PRINT AT 8,25;" "
630 IF L=0 THEN GOTO 340
640 GOTO 269
800 IF L=0 THEN GOTO 900
850 PRINT "WEISS GEWINNT"
860 GOTO 930
870 PRINT "IHR HABT UNENTSCHIED
N GESPIELT"
880 GOTO 930

```

```

899 STOP
900 PRINT "SCHWARZ GEWINNT"
930 PRINT AT 10,25;" NEUES"
934 PRINT AT 12,25;" SPIEL"
935 LET L=0
936 PRINT AT 14,25;" (J/N)"
940 IF INKEY#="J" THEN RUN
945 IF INKEY#="N" THEN GOTO 200
0
950 GOTO 940
999 STOP
1000 LET C=0
1001 LET U=1
1002 IF L=0 THEN LET U=2
1003 LET S=H
1050 LET H=H+1
1060 IF B(Y,H)=U THEN GOTO 1050
1080 LET H=H-1
1090 LET C=C+1
1100 IF C=5 THEN GOTO 800
1110 IF H<1 THEN GOTO 1200
1120 IF B(Y,H)=U THEN GOTO 1080
1200 LET C=0
1203 LET Q=Y
1205 LET H=S
1210 LET Y=Y+1
1220 IF B(Y,H)=U THEN GOTO 1210
1230 LET Y=Y-1
1240 LET C=C+1
1250 IF C=5 THEN GOTO 800
1260 IF B(Y,H)=U THEN GOTO 1230
1300 LET Y=Q
1305 LET C=0
1310 LET Y=Y+1
1320 LET H=H+1
1330 IF B(Y,H)=U THEN GOTO 1310
1340 LET Y=Y-1
1350 LET H=H-1
1360 LET C=C+1
1370 IF C=5 THEN GOTO 800
1380 IF H<1 THEN GOTO 1400
1390 IF B(Y,H)=U THEN GOTO 1340
1400 LET Y=Q
1410 LET H=S
1420 LET C=0
1430 LET Y=Y+1
1440 LET H=H-1
1445 IF H<1 THEN GOTO 1455
1450 IF B(Y,H)=U THEN GOTO 1430
1455 LET Y=Y-1
1460 LET H=H+1
1466 LET C=C+1
1470 IF C=5 THEN GOTO 800
1475 IF H<1 THEN GOTO 1500
1480 IF B(Y,H)=U THEN GOTO 1455
1500 IF B(6,1)<>0 AND B(6,2)<>0
AND B(6,3)<>0 AND B(6,4)<>0 AND
B(6,5)<>0 AND B(6,6)<>0 THEN GOT
O 870
1501 RETURN
1998 SAVE "VIER"
2000 CLS
2005 PRINT AT 10,8;"* VIER GEWIN
NT *"
2006 PRINT AT 8,8;"*****
***";AT 9,8;"*";TAB 23;"*";AT 11
,8;"*";TAB 23;"*";AT 12,8;"*****
*****"
2010 PRINT AT 16,2;"BRAUCHEN SIE
EINE ERKLAERUNG ?"
2012 PRINT AT 20,4;"<J=JA,N=NEIN
>"
2020 IF INKEY#="J" THEN GOTO 500
2030 IF INKEY#="N" THEN RUN
2170 GOTO 2020

```

**ZX-USER CLUB**



## Stop

Jedem Programmierer ist es bei der Eingabe von längeren Programmen schon aufgefallen, daß beim Einfügen einer neuen Zeile das komplette Listing durchgeblättert wird, was natürlich sehr störend und zeitraubend ist.

Das folgende kurze Basic-Programm schafft Abhilfe.

Sie geben die 5 Zeilen Nr. am Anfang Ihres Programmes ein. Immer dann, wenn der Bildschirm mit Programmzeilen gefüllt ist, starten Sie das Programm mit RUN oder GO TO 1.

Resultat: die zuletzt eingegebene Zeile wird auf dem Bildschirm als oberste angezeigt. **AND DON'T FORGET!** zum Schluß die Zeilen 1—5 wieder löschen.

```
1 LET ZNR=PEEK 16394+256*PEEK
16395
2 POKE 16419,ZNR-256*INT (ZNR
/256)
3 POKE 16420,INT (ZNR/256)
4 LIST ZNR
5 STOP
```

## Fließkomma

Ein Problem bei Fließkomma-Rechnungen ist die Wahl des Ausgabeformats. Da der ZX 81 eine unformatierte Zeichenkette ausgibt, ist es schwierig, eine Tabelle von Ergebnissen zusammenzustellen, in der alle Zahlen in gleichmäßigen Spalten stehen.

Das oben erwähnte Problem wird besonders deutlich, wenn Rechnungen mit Geld in einem Programm auftreten — zwei Dezimalstellen werden auch dann benötigt, wenn deren Wert Null ist.

Das folgende Unterprogramm wandelt Zahlen in ein festes Format und gibt sie bei einer bestimmten Spalten-Nummer aus. Dadurch wird es wesentlich leichter, einen Bildschirm voll Zahlenergebnisse auszuwerten. Wenn Sie einen Drucker verwenden, so werden diese Ergebnisse dadurch wesentlich schöner aussehen.

Alle Ergebnisse werden auf zwei Dezimalstellen gerundet, so wird z. B. der Wert 3.245 als 3.25 ausgegeben. Beachten Sie jedoch, daß Bruchteile in Ihren Zahlen mit der Zeit Ganzzahlen um einen Pfennig verfälschen.

```
10 INPUT V
20 GOSUB 9500
30 PRINT TAB 5;V;TAB 25-LEN Z#
+1;Z#
40 RUN
9500 REM ZAHL IN V WIRD GERUNDET
UND IN STRING Z# GELADEN
9510 LET XL=INT (ABS V+.005)*SGN
V
9520 LET XP=INT ((ABS (V-XL)*100
)+.5)
9530 LET Z#=STR# XP
9540 LET Z#=STR# XL+"."+"0"+Z#)
(LEN Z# TO )
9550 RETURN
```

EINGABE	AUSGABE
0	0.00
.01	0.01
0.1	0.10
10	10.00
12.5	12.50
18.99	18.99
19.9944	19.99
19.9999	20.00
200.01	200.01

## Unendlich lange Pause

Die erste Idee hält der Autor für besonders raffiniert, und sie wird, wie Sie vielleicht schon festgestellt haben, in zahlreichen Programmen dieses Buches verwendet. Aus dem Sinclair-Programmier-Handbuch ist ersichtlich, daß ein PAUSE-Befehl mit einem Wert größer als 32767 zu einer »unendlich langen Pause« führt. Der Zustand kann in einem Programm nur durch Drücken einer Taste beendet werden.

Diese Möglichkeit ist besonders für ZX 80-Besitzer, welche zusätzlich die 8 K ROM-Erweiterung angeschafft haben, interessant. Leider ist beim ZX 80 während des Programmlaufs keine Bildschirm-Darstellung möglich; daher können die Möglichkeiten der INKEY\$-Funktion nicht völlig ausgenutzt werden. Das eben Gesagte gilt allerdings nicht für den ZX 81 — mit Ausnahme der schnellen Betriebsart dieses Gerätes.

In einem Programm kann man daher schreiben:

```
200 PAUSE 33000
```

Der Autor benutzt als Gedächtnisstütze jedoch folgenden Ausdruck:

```
200 PAUSE 4E4
```

... aus zweierlei Gründen: Diese Zeile benötigt nicht nur geringfügig weniger Speicherplatz als die vorhergehende, sondern

klings bei der Aussprache »Pause for-ee-for« auch noch wie der englische Ausdruck »Pause forever«.

Wie Sie bereits wissen, sind beim ZX 81 technisch-wissenschaftliche Zahlendarstellungen durchaus möglich. Der hier verwendete Wert 4E4 entspricht der Zahl 40000, und dies ist größer als der oben erwähnte Wert 32767.

## JOJO oder die optische Täuschung des ZX 81

Dieses kleine Programm von L. Selle aus Bad Berleburg erzeugt ein bewegtes Bild mit scheinbar gleichzeitig vor- und rückwärts rollendem Bild. Eine optische Täuschung?

... eine kleine Modifikation und nun rollt alles über den ganzen Bildschirm.

```
10 LET S=1
20 FOR N=0 TO 21
30 SCROLL
40 PRINT AT N,S; "I"
50 NEXT N
60 LET S=2/S
70 GOTO 20
```

```
5 LET Z=0
10 LET S=1
20 FOR N=0 TO 21
30 SCROLL
40 PRINT AT N,S; "I"
50 NEXT N
60 LET S=S-PI
70 LET Z=Z+1
80 IF Z<10 THEN GOTO 20
90 RUN
```

Hier stellen wir Ihnen wichtige ZX-Abfragen vor.

### Programmlänge

```
PRINT (PEEK 16396+256*PEEK
16397)-16509
```

### Programmlänge + Bildspeicher + Datenbereich

```
PRINT (PEEK 16404+256*PEEK
16405)-16509
```

Länge des Bildspeichers (bei 16K Anwendung immer 792 Bytes) also 24 x 33

```
PRINT (PEEK 16400+256*PEEK
16401)-(PEEK 16396+256*PEEK 16397)+1
```

### Länge des Datenbereichs

```
PRINT (PEEK 16404+256*PEEK
16405)-(PEEK 16400+256*PEEK 16401)
```

### Freier Speicherplatz

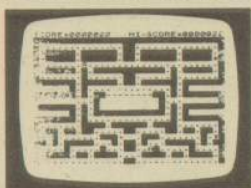
```
PRINT (PEEK 16386+256*PEEK
16387)-(PEEK 16412+256*PEEK 16413)
```



# SOFTWARE AUF CASSETTE

Für den Sinclair ZX81-Microcomputer.

## CASSETTE 19



### Pack-Man

Das beliebte und weltbekannte Freß-Mich — Freß-Dich-Spiel nun auch endlich für Ihren ZX81 zuhause auf dem Bildschirm. In superschneller Maschinensprache und irrer Grafik Ihre private Spielhölleparty kann beginnen.

Art-Nr. 1019, DM / Fr. 29.80

## CASSETTE 20

### Space-Invaders

... schon wieder ein weltbekanntes und faszinierendes Spiel aus der Welt der Spielcasinos. Klar, in der superschnellen-Maschinensprache und echt guter Grafik. 100 Schwierigkeitsstufen! Wird die UFO-Flotte Mutter Erde erreichen?

Art-Nr. 1020, DM / Fr. 29.80

## CASSETTE 21

### Centipedes

Las Vegas erblaßt! Centipedes, die Schlange, die vom Himmel kommt und sich bei Tretter in eine Pilz verwandelt. Die Geschwindigkeit bestimmen Sie. Ihre, 100 Schwierigkeitsstufen! Soll Las Vegas einmal nachmachen!

Art-Nr. 1021, DM / Fr. 29.80

## CASSETTE 22

### Mazogs

Der Ritter Kunibert rast durch den Garten, holt den Schatz, rast zurück, doch... unterwegs hat er gegen feuerspeiende Drachen und Ungeheuer zu kämpfen, schaffst du es, den Schatz gut nachhause zu bringen? Tolle Grafik, superschnell, Maschinensprache!

Art-Nr. 1022, DM / Fr. 29.80

## CASSETTE 23

### Roulette

Monaco, Glanz, Glitter, Champagner und Roulette, die Welt, die Sie mit dieser faszinierenden Cassette auf Ihren Bildschirm zaubern können. Sehr hübsche Grafik, rien ne va plus.

### Unwetter

Regen, Hagel, Wind und Gewitter können unseren 2 Freunden nichts anhaben, aber jeder versucht, das Einfamilienhaus des anderen dem Erdboden durch ein Unwetter gleich zu machen.

## Labyrinth

Fanatisch fantastisch, dreidimensional können Sie nun endlich durch das Labyrinth durch Knossos irren. Finden Sie den Ausgang oder verhungern und verdursten Sie? 3 Spiele auf 1 Cassette.

Art-Nr. 1023, DM / Fr. 29.80

## CASSETTE 24

### Regierung

Zurück zur Monarchie! Sie als König eines Kleinstates mit 1000 Erdenbürgern regieren über Wohlstand und Untergang Ihres Landes.

Art-Nr. 1024, DM / Fr. 19.80

## CASSETTE 25

### Schiffe versenken

Die Knilche von der letzten Bank und viele Generationen davor und danach haben dieses Spiel nicht nur im Mathe-Unterricht gespielt, jetzt endlich für Ihren ZX81.

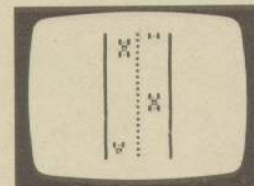


### Schlange

Sie als Schlange müssen auf dem Bildschirm verschiedene Kostlichkeiten (Zahlen) verkostigen und werden als Punktesieger den Bildschirm verlassen, aber Vorsicht!

### Makler-Spiel

Das Spiel für jung und alt. Kaufen und verkaufen Sie Straßen, Häuser, Hotels und ganze Straßenzüge. Wer ist der bessere Kapitalanleger?



### Indianapolis

Wir sind sicher, Sie werden nach dem 321. Rennen die gefährlichste Rennstrecke der Welt gegen die weltbesten Fahrer im Schlat beherrschen.

4 Spiele auf einer Cassette.

Art-Nr. 1025, DM / Fr. 29.80

## CASSETTE 26

### Einmauern

Ihren Todfeind können Sie einmauern, ein leicht makaberes Vergnügen. Wir drücken Ihnen alle Daumen, daß Sie nicht selbst dran glauben müssen. Jede Menge Geschicklichkeit ist erforderlich.

## Seekrieg

Tick, tick, tick... das U-Boot, in dem Sie sitzen, wird von Flugzeugen und Zerstörern bombardiert.

### Astro-Fighter

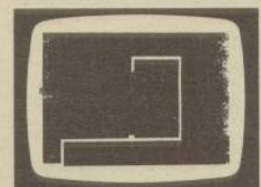
Ihr superschneller Astro-Fighter fliegt durch die Galaxien und trifft auf feindliche Raumschiffe, die Ihnen nichts Gutes wollen.

## Städte-Verteidigung

Ein interessantes Weltraumspiel mit dynamischer Grafik. Verteidigen Sie 2 Städte gegen außerirdische Angreifer. Sie haben Waffen, Schutzschilde und jede Menge anderer Möglichkeiten sich zu verteidigen.

Art-Nr. 1026, DM / Fr. 29.80

## CASSETTE 27



### Graben

Jack McNugget, der wildeste Goldgräber unserer Zeit, grabt weiter nach Gold. Aber Vorsicht, nach jedem Fund sollte man schleunigst zur städtischen Sparkasse gehen, um den Goldfund in Sicherheit zu bringen.

Art-Nr. 1027, DM / Fr. 19.80

## CASSETTE 28

### Tages-Programm

Wir tun was für den Familienfrieden... denn nun weiß jeder, wann und was er im Fernsehen sehen will. Sie können auf Wochen voraus alle Fernsehsendungen für Ihre Familienmitglieder speichern und jederzeit abrufen. Ein Wunder der Technik, meint unser Cheftechniker Brandl.

Art-Nr. 1028, DM / Fr. 19.80

## CASSETTE 29

### Kombinatorik

Unsere Hommage an alle Einsteins... dieses Programm berechnet in Sekunden die verschiedenen Kombinationsmöglichkeiten z. B. eines einfachen Lottoscheins.

Art-Nr. 1029, DM / Fr. 19.80

## CASSETTE 30

### Logbuch

Speziell für unsere Amateurfunker zwischen Äquator und Südpol. Speichern Sie einfach alle Daten, wie Uhrzeit, Datum, Ruf, Frequenz, QTH und QSL usw.

## Entfernung

Durch Eingabe des Längen- und Breitengrades zweier Orte definiert der Computer die kürzeste Strecke zwischen diesen 2 Punkten.

### Weltzeit

Wenn Ihre Tante in Honolulu oder Wanne Eickel aufsteht, erfahren Sie endlich nur durch die Eingabe Ort und Längengrad und schon haben Sie die Ortszeit.

3 Programme auf 1 Cassette.

Art-Nr. 1030, DM / Fr. 29.80

## CASSETTE 31

### Finanzbuchhaltung

Das Programm benötigt mindestens 16 K, kann aber auch bis zu 64 K verwendet werden, da das Programm den Speicherplatz automatisch abfragt. Ein umfangreiches Programm mit z. B. Kontokorrentliste, Fibu-Anschluß und Bilanz etc.

Art-Nr. 1031, DM / Fr. 39.80

## CASSETTE 32

### Maschinencode-Tester

Ermöglicht das Entwickeln und Experimentieren mit Programmen in Maschinencode. Sehr umfangreich mit z. B. Sprungweitenberechnung, Übersichtsauflistung, und Decrementieren der laufenden Zeile etc.

Art-Nr. 1032, DM / Fr. 29.80

## CASSETTE 33

### Editor

Machen Sie Ihren Computer transparenter. Aktuelle Auflistung der Speichereinhalte und der Z80-Register. Sehr wichtig für das Arbeiten in Maschinensprache.

### Hexlist

Das Programm erzeugt ein nach Z80-Befehlen geordnetes Hex-Listing mit Berechnung der absoluten Zieladressen von relativen Sprüngen.

2 Programme auf 1 Cassette.

Art-Nr. 1033, DM / Fr. 29.80

## CASSETTE 35

### Textverarbeitung

Ein Maschinensprache-Programm. Das totale Textverarbeitungsprogramm. Sie speichern seitenlang, fügen Sätze ein, löschen ganze Absätze und drucken es auf Ihrem Normalpapierdrucker als Manuskript aus etc.

Art-Nr. 1035, DM / Fr. 29.80

## CASSETTE 36

### Texteditor

Das Programm mit den unendlichen Möglichkeiten, z. B. Spalten versetzen, Absätze versetzen, das Verschieben und Einfügen von Wörtern und Buchstaben, automatischer Umbruch etc.

Art-Nr. 1036, DM / Fr. 29.80

## CASSETTE 37



### Druckgrafik

Nun haben Sie endlich die Möglichkeit, eine hochauflösende Grafik per Software auf Ihrem ZX-Drucker darzustellen. Auflösung auf ca. 47.000 Einzelpunkte.

Art-Nr. 1037, DM / Fr. 19.80

## CASSETTE 38

### Groß- und Kleinschrift

Ihr ZX81 ist nun endgültig aus den Kinderschuhen entwichen... mit diesem Programm können Sie nun auf dem ZX-Drucker groß oder klein schreiben.



### Schreibschrift

Sogar eine Sonntags-nachmittags-Schönschrift und exclusive Schreibschrift für den besonderen Anlaß, Heiratsantrag, Liebesbriefe und Computerbestellungen erzeugt dieses Programm auf dem ZX-Drucker.

2 Programme auf 1 Cassette.

Art-Nr. 1038, DM / Fr. 29.80

## CASSETTE 39

### BASIC-COMPILER

Mit dem BASIC-COMPILER schreiben Sie ohne Maschinensprachekenntnisse superschnelle Programme, denn er übersetzt Ihre Basicprogramme in den Maschinencode. Es können damit Verarbeitungsgeschwindigkeiten erzielt werden, die bis zu 100 mal schneller sind als in BASIC. Sie werden Ihren Augen nicht trauen.

Art-Nr. 1039, DM / Fr. 29.80

Bestellen Sie im Innenteil auf Seite 60



## Einführung in die Maschinensprache

Grundlagenkurs Maschinensprache der ZX80  
am Beispiel des ZX81

Von Martin Gehrke

**N**ach vor nicht allzu langer Zeit zuckte ich ratlos die Achseln, wenn das Zauberwort Maschinensprache fiel.

Was war das, über das man nur hinter vorgehaltener Hand sprach? Was bedeuteten die geheimnisvollen Zeichen im ZX-Handbuch? Nun, es ist viel einfacher und überhaupt viel logischer als man zu Anfang denkt. Diese Vielzahl von Befehlen ist gar nicht so unüberschaubar wie es scheint. Wir wollen beim wichtigsten und wohl auch einfachsten Befehl anfangen — dem Ladebefehl, abgekürzt ld. Dazu muß man wissen, daß die CPU — das Gehirn des Computers — verschiedene Register hat. Die wichtigsten sind:

- Das a-Register oder Akkumulator. Es wird in erster Linie zur Speicherung aktueller Werte verwendet, d.h. für Werte, die gerade behandelt werden. Das kommt daher, weil man mit dem a-Register sehr viel operieren kann.
- Dann gibt es das bc-Register oder besser Registerpaar. Man kann das b-Register wie auch das c-Register einzeln behandeln, doch oft kombiniert man beide, so daß man ein 16-Bit-Register erhält. bc steht für Byte-Counter, weil es oft für Zählaufgaben benutzt wird. Beim ZX hat das bc-Register noch eine besondere Aufgabe. Wenn man ein Maschinenprogramm mit USR aufruft, so kann man es in der Form PRINT USRn oder LET A = USRn tun. Im ersten Fall wird der Wert, der im bc-Registerpaar als 16-Bit-Zahl abgespeichert ist, direkt auf den Bildschirm ausgegeben, im zweiten Fall wird dieser Wert der Variablen A zugewiesen. Die anderen Register können nicht direkt ins BASIC ausgegeben werden.
- Als drittes gibt es das hl oder high-low Registerpaar. Hier gilt das gleiche wie für das bc-Registerpaar: man kann jedes Register einzeln ansprechen, doch man nutzt es sinnvollerweise als 16-Bit-Register, denn die CPU bietet dieses Register geradezu an für die Speicherung von Adressen, die ja bekanntlich mit 16-Bits dargestellt werden. Ansonsten wäre ja auch eine Adressierung bis 65535 gar nicht möglich.
- Als letztes wichtiges Registerpaar

möchte ich hier das de-Register ansprechen. Können in der Maschinensprache werden jetzt fragen: was ist denn mit dem Stackpointer oder mit dem Register ix und iy? Nun für solche Leute ist dieser Artikel nicht geschrieben, aber um die nun neugierig gewordenen anderen Leser zu beruhigen, sage ich an dieser Stelle, wir werden auch diese Register später noch kurz behandeln. Das de-Registerpaar ist ein Hilfsregister, das dazu dient, eine 16-Bit-Zahl zu speichern, die in den anderen Registerpaaren nicht sinnvoll wäre oder wenn diese Register gelegt sind, woanders gespeichert werden muß, oder auch ganz einfach deswegen, weil manche Befehle zu ihrer Ausführung drei Registerpaare benötigen.

Nun zu den Ladebefehlen:

Wenn man sich diese Einteilung der CPU so anschaut, wird man sich fragen: wie kann man Daten bzw. Zahlen vom Speicher in das eine oder andere Register packen oder auch von einem weniger geeigneten in ein mehr geeignetes Register umschreiben. Dazu gibt es ein ganzes Sortiment von Ladebefehlen, z.B. ld a,c bewirkt, daß der Inhalt des c-Registers ins a-Register übertragen wird. Dabei bleibt der Inhalt des c-Registers unverändert. Der alte Inhalt des a-Registers wird dabei überschrieben. Das hier gesagte gilt so für alle gleich strukturierten Ladebefehle.

z.B. Um aber Daten innerhalb der CPU umherschoben zu können, müssen sie erst einmal hineinkommen. Das geht wieder mit einem Ladebefehl. Er hat die Struktur ld a,N wobei das N für die Zahl steht, die in dem Byte des Speichers steht, welches unmittelbar auf das Byte folgt, in welchem der Ladebefehl steht.

ld a,b; ld a,h; ld a,l; ld a,d;  
ld a,e; ld b,c; ld c,d; ld c,e;  
usw.

z.B. die Adresse 40 82 (hexadezimal) enthalte 3E, was soviel bedeutet wie ld a,N, so wird der Inhalt der Adresse 40 83 in den Akku (so wollen wir das a-Register jetzt nennen) übertragen.

Somit können wir fest im Programm gespeicherte Werte in die CPU übernehmen. Das Programm sähe wie folgt aus:

Adresse (Hexadez.)	Inhalt (Hexadez.)	Bedeutung
40 82	3E	ld a,N
40 83	z.B. FF	

Hier würde die hexadezimale Zahl FF (die der dezimalen Zahl 255 entspricht) in dem Akku geladen.

Genauso ist es möglich, ein Registerpaar mit einer Zahl zu laden:

Adresse (Hexadez.)	Inhalt (Hexadez.)	Bedeutung
40 82	01	ld bc,NN
40 83	z.B. D8	
40 84	14	

Hier wird das b-Register mit 14 geladen und das c-Register mit D8. Darunter kann man sich aber noch nicht viel vorstellen. Welchen Sinn hat das ganze?

Nun, zu Anfang hatten wir festgestellt, daß man mit einem Registerpaar Zahlen bis 65535 darstellen kann. In unserem Beispiel hieße die Zahl 14 D8. Wieviel Dezimal dies ist, läßt sich leicht errechnen. 14 ist das höherwertige Byte, D8 das niederwertige. Schauen wir nach, was im Handbuch in der Zeile in der unter Hex 14, unter CODE steht. Dort steht 20. Bei D8 finden wir 216. Nun müssen wir nur noch das höherwertige Byte (in unserem Fall 20) mit 256 multiplizieren und das niederwertige Byte hinzuzählen. Schon haben wir die fertige dezimale Zahl. Sie lautet in unserem Fall  $20 * 256 = 5120$

+ 216

5336

Damit enthält das bc-Register nach Ablauf dieses Programms 5336.

Um dies nun auch praktisch ausprobieren zu können, benötigen wir noch einen kleinen, aber sehr wichtigen Befehl. Es ist der ret-Befehl. Er veranlaßt den Computer, der sich »im« Maschinenprogramm befindet, ins BASIC zurückzukehren. Würde man diesen Befehl am Ende eines Maschinenprogramms weglassen, würde der Computer Byte für Byte im Speicher weitergehen, um nach neuen Anweisungen in Maschinensprache zu suchen. Irgendwann ist er am Ende des Speicherbereiches und fängt von vorne an, oder er trifft früher oder später auf Teile im Speicher, die mit BASIC beschrieben sind und produziert dann einen ganzen Haufen Müll — d.h. er »steigt aus«. Soweit wollen wir es jedoch gar nicht kommen lassen.

Nun müßten wir uns einen geeigneten Platz aussuchen, an welchem wir unser kleines Programm unterbringen wollen. Prinzipiell kann man es an jeden Platz des Speichers schreiben, man muß jedoch aufpassen, daß man mit dem BASIC, dem Bildschirmspeicher, dem variablen Spei-



# TIPS & TRICKS

cher und allen anderen vom Computer benutzten Speicherbereichen nicht in Konflikt kommt. Das hätte dann zur Folge, daß entweder das Maschinenprogramm überschrieben wird oder der Computer aussteigt. Also wählen wir einen sicheren Platz.

Es gibt deren zwei. Der eine ist dort, wo alles BASIC und alle sonstigen Speicherbereiche aufhören, nämlich hinter dem zu diesem Zweck herabgesetzten RAMTOP. Normalerweise ist das RAMTOP das letzte Byte überhaupt im Speicher. Man kann es jedoch um soviel herabsetzen, wie man Platz dahinter braucht. Dieser Platz kann vom BASIC nicht mehr überschrieben werden, wird aber beim SAVE-Befehl nicht mit auf die Kassette übertragen — ein schwerer Nachteil.

Deshalb wählen wir die Höhle des Löwen, um darin einen sicheren Platz für unser Maschinenprogramm zu finden. Wir wählen die erste Zeile im BASIC-Programm und nehmen dafür den Befehl REM.

Wenn davor keine andere Zeile ist, können wir sicher sein, daß die Adresse des ersten Bytes nach dem REM-Befehl 16514 heißt. Weil unser ZX von sich aus keinen Hexadezimalcode schluckt, wollen wir an dieser Stelle dazu übergehen, uns in dezimalen Zahlen auszudrücken — außerdem ist es am Anfang übersichtlicher.

Nun ist es einfach, unser Programm einzugeben und es laufen zu lassen. Dazu übersetzen wir es in Dezimalcodes, und was noch wichtiger ist, in die ZX-Charakter, denen ja auch jeweils ein Maschinenbefehl oder eine Zahl zugeordnet werden kann. Das sieht dann so aus:

Adresse (dezimal)	Befehl (dezimal)	ZX-Charakter
16514	1	*
16515	216	**
16516	20	=

Also sieht die REM-Zeile so aus:

```
1 REM * ** = TAN
```

Das TAN ist der ZX-Charakter, für den Rücksprung ret (Code 201), den wir an das Ende eines jeden Maschinenprogramms hängen müssen.

Achtung! TAN und \*\* dürfen nicht als einzelne Zeichen eingegeben werden, da sie sonst eine andere Bedeutung hätten. Sie müssen als Shift-Newline E und Shift-H eingegeben werden!

Nun schreiben wir:  
10 PRING USR 16514 und starten mit RUN

Als bald wird links oben auf dem Bildschirm die Zahl 5336 stehen — es hat geklappt. Unser erstes Maschinenprogramm läuft.

Nun bietet die Z80 CPU aber noch eine andere Möglichkeit, Register oder auch Speichstellen zu laden. Es ist die Möglichkeit der indirekten Adressierung. Es fällt beim Durchschauen der Maschinenbefehle auf, daß es auch Ladebefehle gibt, deren einer Term in Klammern steht. Was dies bedeutet, wollen wir anhand eines kleinen Programms sehen:

Adresse	Befehl	ZX-Charakter
16514	ld a, (NN)	U
16515	137	■
16516	64	RND
16517	ld b, N	■
16518	0	?
16519	ld c, a	TAN
16520	ret	COPY
16521	255	

Das a-Register wird mit dem Inhalt der Speicherstelle geladen, die die Adresse hat, die in den folgenden beiden Bytes gespeichert ist. In unserem Fall ist dies die Adresse 16521 ( $64 \cdot 256 + 137 = 16521$ ). In dieser Speicherstelle ist die Zahl 255 gespeichert. Also müßte das a-Register nach Ausführung dieses Befehls den Wert 255 enthalten. Nun laden wir das b-Register mit 0 (damit das Ergebnis nicht größer als 255 sein kann) und das c-Register mit a, so daß wir über das bc-Register den Wert a ins BASIC übernehmen können. Danach kommt der bekannte Rücksprungbefehl. Die nun folgende Adresse wird vom Maschinenprogramm direkt nicht mehr berührt, sondern nur als Datenspeicher vom a-Register abgefragt.

Wie bringen wir aber nun den Befehl ld c, a in die REM-Zeile? Es gibt ja keinen Charakter für den Code 79!

Ganz einfach. Wir schreiben zuerst:

```
1 REM U ■ RND ■ ? TAN COPY
```

Da Copy nur im -Modus verfügbar ist, müssen wir erst THEN COPY eingeben, dann mit dem Cursor zurückspringen, und das THEN ausradieren. Wir lassen nach dem Space eine Stelle in der REM-Zeile frei (mit Punkt gekennzeichnet). Nun können wir sie mit POKE 16519, 79 mit dem gewünschten Code beschreiben. Allgemein gilt, daß beim POKE in eine REM-Zeile die Codes vom Computer als Fragezeichen angezeigt werden, weil kein Charakter dafür existiert. Es gibt ein paar spezielle Code, die etwas Sonderbare Erscheinungen hervorrufen. Der Code 126 läßt die 5 folgenden Bytes auf dem Bildschirm verschwinden. Im

Speicher sind sie aber noch vorhanden! Eine Zeile, die diesen Code enthält, darf nicht editiert werden, da sonst diese Bytes tatsächlich verschwinden. Code 118 läßt den gesamten nachstehenden Bildschirminhalt verschwinden. Die folgenden Bytes sind jedoch noch vorhanden. Code 127 kann zwar editiert werden, aber man kann diese Stelle mit dem Cursor nicht überqueren. Mit Codes 112—127 sollte man generell vorsichtig sein. Nach Eingabe der Zeile

10 PRINT USR 16514 und RUN

erscheint links oben die Zahl 255 — der Inhalt der Adresse 16521. Dies können wir nachprüfen, indem wir diese Adresse mit einer anderen Zahl vollschreiben; sie mußte dann auf dem Bildschirm erscheinen.

Mit der Möglichkeit, indirekt zu adressieren, können wir nicht nur bestimmte Speicherstellen auslesen, sondern auch beschreiben. Sinnvollerweise nehmen wir hierzu eine Speicherstelle, die im Bildschirmspeicherbereich liegt, damit wir das Ergebnis gleich sehen. Dieser Versuch klappt nur in der 16K Version des ZX81, weil bei nur 1K-Speicherkapazität der Bildschirm anders organisiert ist. Wo aber fängt der Bildschirmspeicherbereich an? Ganz einfach. Bei der Speicherstelle, deren Adresse im sogenannten D-FILE — einer Systemvariablen — gespeichert ist. Der Bildschirmspeicher ist wie folgt aufgebaut:

In dem Byte, auf daß das D-FILE zeigt, steht immer ein Newline — Code 118. Dann kommen die 32 Zeichen, die wir auf dem Schirm sehen, danach wieder eine 118 usw. 24 mal das ganze — das letzte Zeichen im Bildschirmspeicher ist wieder ein Newline.

● Aber Vorsicht beim Beschreiben des Bildschirmspeichers. Die Newlines dürfen nicht überschrieben werden, sonst steigt der Computer aus.

Um also ein Zeichen direkt auf dem Bildschirm darstellen zu können, müssen wir erst einmal wissen, wo er anfängt.

Dazu schreiben wir folgendes Programm:

16514	ld hl, (NN)	E
16515	12	E Systemvariable
16516	64	RND D-FILE
16517	inc hl	7
16518	ld a, N	Y
16519	128	■
16520	ld (hl), a	?
16521	ret	TAN

Ja was soll denn das? Gleich zwei neue Befehle? ld hl, (NN) ist im Prinzip nichts

## ZX-USER CLUB



anderes als der bekannte `ld a, (NN)`, nur daß jetzt das `l`-Register mit dem Inhalt der durch die beiden nachfolgenden Bytes adressierten Speicherstelle, und das `h`-Register mit dem der nächsten Speicherstelle gefüllt wird. Das heißt, das im `hl`-Register nun der Inhalt zweier Speicherstellen, nämlich in unserem Fall von 16396 und 16397 gespeichert ist. Und was bedeutet `inc hl`? Nichts anderes als daß das `hl`-Registerpaar um 1 erhöht wird. Dort ist nämlich jetzt die 1. Adresse des Bildschirmspeichers gespeichert und die enthält ja ein Newline, welches nicht überschrieben werden darf. Also erhöhen wir diese Adresse um 1 und haben nun die Adresse des 1. Zeichens auf dem Bildschirm im `hl`-Registerpaar gespeichert. Nun laden wir den Akku mit dem Code des Zeichens, das wir auf den Bildschirm bringen wollen (in unserem Fall das invertierte SPACE) und laden den Akku in die Adresse, die im `hl`-Registerpaar gespeichert ist — also in den 1. Platz auf dem Bildschirm. Dann springen wir ins BASIC zurück. Auf dem Bildschirm sehen wir nun links oben ein schwarzes Quadrat.

In 16520 müssen wir vorher den Code für `ld (hl),a`, die 119 POKEn. Auch dürfen wir das Programm nicht wie bisher mit `PRINT USR 16514` starten, sondern sollten eine Möglichkeit suchen, bei der keine Ausgabe des `bc`-Registers erfolgt. In unserem Fall würde auf dem Schirm 16514 stehen, weil wir das `bc`-Register nicht beeinflußt haben und dieses dann die Startadresse des Maschinenprogramms enthält. Wir können den `USR`-Befehl hier z.B. mit `IF USR 16514 THEN`, mit `LET X=USR 16514` (wobei die Variable `X` nicht gebraucht wird) oder mit `RAND USR 16514` starten. `Rand USR n` hat dabei den Vorteil, daß es am schnellsten getippt ist und sonst keine nachteiligen Folgen hat. Daß der Zufallszahlengenerator im BASIC dabei initialisiert wird, spielt keine Rolle.

Das gesamte Programm sollte also so aussehen:

```
1 REM E&RND 7Y?PTAN
10 RAND USR 16514
```

In diesem Programm haben wir erstmals den Befehl `inc` verwendet. Es gibt selbstverständlich auch den Komplementären Befehl `dec`, der das entsprechende Register um 1 erniedrigt.

Es gibt zwei Sorten von Befehlen:

- Die eine erhöht ein einzelnes Register (bei entsprechend häufiger Durchführung) bis 255 und beginnt dann wieder bei 0. (Das gilt auch — nur umgekehrt — für `dec`).

— Die andere Sorte erhöht ein Registerpaar; und zwar solange bis dieses Paar die Zahl 65535 enthält. Das heißt, es gibt immer dann einen Übertrag von niederwertigen auf das höherwertige Register, wenn das Niederwertige von 255 auf 0 umspringt (das gleiche gilt für den `dec`-Befehl, nur umgekehrt). Nach 65535 springt das Registerpaar auf 0 um.

Es gibt auch einen indirekten `inc/dec` Befehl. `inc (hl)` und `dec (hl)`, was soviel heißt, wie erhöhe/erniedrige die Speicherzelle, deren Adresse im `hl`-Register gespeichert ist, um 1.

Nun wollen wir aber auch wissen, wie so ein Computer überhaupt rechnen kann. Dazu gibt es die Befehle `add`, `sub`, `adc` und `sbc`. Der Befehl `add, a, c` z.B. addiert zum alten Inhalt des Akkus den Inhalt des `c`-Registers und legt das Ergebnis im Akku ab. Das gleiche gilt für alle gleich strukturierten Befehle. `sub c` nimmt den Inhalt des Akkus, subtrahiert davon den Inhalt vom `c`-Register und legt das Ergebnis im Akku ab. `adc` und `sbc` funktionieren im Prinzip gleich, setzen jedoch, wenn das Ergebnis größer als 255 ist, das sogenannte carry flag oder `c-bit`, wenn das Ergebnis kleiner als 0 ist, wird das carry flag zurückgesetzt, d.h. es wird 0. Das carry flag ist nur ein einziges Bit und hat mit dem `c`-Register nichts zu tun. Wir werden später noch sehen, wie man den Zustand des carry flags abfragen und sinnvoll einsetzen kann. Da wäre z.B. eine Subtraktion oder Addition mit mehr als einem Byte; da nimmt man das carry flag als Übertrag, damit das zweite Byte, wie bei der schriftlichen Addition oder Subtraktion den Rest des ersten Bytes angerechnet bekommt. So wird bei `adc` und `sbc` das carry flag zu Akku addiert. Es gibt auch zwei Befehle, die das carry flag direkt setzen oder invertieren können. Es sind die Befehle `scf` (set-carry-flag) und `ccf` (change-carry-flag). Mit `scf` wird das `c-flag` auf 1 gesetzt, mit `ccf` wird es invertiert, d.h. wenn es 1 ist, wird es 0 und umgekehrt.

Wir wollen nun folgendes Programm ausprobieren:

```
16514 ld a,20 y
16515 20 =
16516 ld d,12 -
16517 12 =
16518 add a,d
16519 ld b,0
16520 0
16521 ld c,a ? (Code: 79)
16522 ret TAN
```

Dieses Programm addiert das `d`-Register zum `a`-Register. Zur Ausgabe wird es ins `bc`-Register umgeladen.

Nach Starten mit `Print USR 16514` steht auf dem Bildschirm 32 ( $20 + 12 = 32$ ). Ersetzen wir nun den Befehl `add a, d` durch `sub d`, so erscheint die Zahl 8 als Ergebnis von  $20 - 12$ . `sub d` hat dieses

Grafik-Zeichen. Was passiert aber, wenn das Ergebnis einer Subtraktion kleiner ist als 0? Probieren wir es aus. Laden wir das `d`-Register einfach mit 21 und lassen es laufen (das muß durch ein `+` ersetzt werden). Das Ergebnis ist überraschend 255?! Nun der Subtrahierer in der CPU macht, wenn das Ergebnis 0 überschreitet, einfach weiter mit 255. Wenn wir das `d`-Register mit 22 laden, so erscheint der Wert 254. Nun aber wieder zurück zu unserem carry flag. Wir laden das `d`-Register wieder wie ursprünglich mit 12 und ersetzen den `sub`-Befehl durch `adc`. Wenn wir vor den `adc`-Befehl nun noch den Befehl `scf` setzen (Charakter `R`), dann können wir sicher sein, daß nach Ausführung des Programms die Zahl 33 erscheint ( $20 + 12 + 1$  vom `c-flag`!). Fügen wir zwischen `scf` und `adc` nun `ccf` ein, so ist das `c-flag` 0 und das Ergebnis 7 ( $20 - 12 - 1 = 7$ ).

Nun wollen wir eine Mehr-Byte-Addition ausführen lassen. Dazu müssen wir uns vergegenwärtigen, daß der Befehl `adc` das `c-flag` nicht nur mit verarbeitet, sondern auch bei einem Ergebnis größer als 255 setzt. Dazu müssen wir aber erst noch eine andere Verwendungsmöglichkeit des `c-flags` besprechen.

Man kann, ähnlich wie im BASIC, auch im Maschinenprogramm Sprünge an eine andere Stelle ausführen. Hier gibt es 2 Möglichkeiten:

- Erstens den direkten Sprung an eine bestimmte Adresse.
- Zweitens, und das ist für uns jetzt wichtig, den relativen Sprung um `n` Bytes vorwärts oder rückwärts. Dieser Sprung heißt `jump relative`, kurz `jr`.

Das Sinnvolle an diesem Befehl ist, daß man ihn mit dem Zustand verschiedener Status-bits, wie unser `c-flag` auch eines ist, bedingen kann. Es gibt ihn in der Form `jr c DIS` und `jr nc DIS`, bei `jr c DIS` wird der Sprung nur ausgeführt, wenn das `c-flag` 1 ist, bei `jr nc DIS` wird nur gesprungen, wenn das `c-flag` 0 ist. Natürlich gibt es auch den bedingungslosen Sprung `jr DIS`. Was dieses mysteriöse `DIS` nun bedeutet, wollen wir auch klären. Grundsätzlich ist es möglich, vorwärts wie auch rückwärts zu springen. Ist das folgende Byte nach dem `jr`-Befehl größer als 127, so wird rückwärts gesprungen; ist es kleiner oder gleich 127, so springt der Computer vorwärts, und zwar nach folgendem Schema:



# TIPS & TRICKS

Adresse	Befehl	Sprungzahl	DIS
x-3		251	
x-2		252	
x-1		253	
x	jr	254	
x+1	DIS	255	
x+2		0	
x+3		1	
x+4		2	
x+5		3	USW.

In unserem Programm benötigen wir den Befehl jr nc DIS. Er soll den Programmteil überspringen, indem das b-Register mit 1 geladen wird, falls das c-flag nicht gesetzt ist. Dann nämlich muß b=0 sein.

16514	ld a,254	Y
16515	254	RETURN
16516	ld a,243	
16517	243	NEXT
16518	adc a,d	
16519	ld c,a	? (CODE 79)
16520	ld b,0	
16521	0	
16522	jr nc,DIS	K
16523	2	
16524	ld b,1	
16525	1	
16526	ret	TAN

Als Ergebnis erscheint 497 (= 254 + 243). Wir haben hier 2 8-Bit Zahlen addiert und als Ergebnis eine 2 Byte oder 16-Bit-Zahl erhalten (wenn auch das Ergebnis nie größer als 510 sein kann; 254 + 255 = 510). Wenn wir das a-Register beispielsweise mit 50 (M) und das d-Register mit 58 (U) laden, so erhalten wir das korrekte Ergebnis 108. Das zeigt zum einen, daß der adc-Befehl sein carry flag richtig gesetzt bzw. zurückgesetzt hat und zum anderen, daß der jr nc DIS-Befehl richtig arbeitet.

Außer diesen 1-Byte-Rechenbefehlen gibt es noch einige 2-Byte-Befehle zum Addieren und Subtrahieren zweier Register, z.B. add hl, de; add hl, bc; aber auch adc hl, bc; oder sbc hl, bc.

Die beiden letzten Befehle finden wir nicht in der Spalte »Z80-assembler«, sondern in der Spalte »—after EDh«. Was soll denn das nun bedeuten? Wie immer ganz einfach. Um diese Befehle abrufen zu können, müssen wir vor den Code des Befehls, den Code der Hexadezimalzahl ED setzen (das kleine h bedeutet hexadezimal). Folglich sieht der Befehl sbc hl, bc im Speicher wie folgt aus:

Adresse	Code	Befehl	Charakter
x	237	EDh	GOSUB
x+1	66	sbc hl, bc	PI

Würden wir CBh vorsetzen, würden wir den Befehl bit 0, d, den wir später noch besprechen werden, erhalten. Setzen wir

gar nichts vor, so gilt der Befehl ld b, d, der ja bekannt sein dürfte. Da die 2-Byte-Rechenbefehle im Prinzip genauso funktionieren wie die Rechenbefehle mit einem Byte, bleibt es dem Leser überlassen, weitere Versuche damit anzustellen. Wir wollen uns nun einem anderen wichtigen Befehl zuwenden — dem compare-Befehl. Mit dem compare — oder kurz cp-Befehl — können Vergleiche ausgeführt werden. Dazu ist noch ein anderes flag wichtig; das z — oder zero-bit. Es wird unter anderem vom cp-Befehl gesetzt oder rückgesetzt. Gesetzt wird es, wenn der Inhalt der beiden zu vergleichenden Register, oder bei indirekter Adressierung, auch Speicherstellen, gleich ist. Rückgesetzt, d.h. es ist dann logisch 0, wird es, wenn der Vergleich negativ ist, d.h. beide Inhalte ungleich sind. Auch für dieses Bit gibt es abhängige jr-Befehle.

Der cp-Befehl beeinflusst auch das c-flag. Und zwar wird es 0, wenn die zum Vergleich anstehende Zahl kleiner oder gleich dem Akku ist. 1 wird es, wenn die Zahl größer ist als der Akku.

Den cp-Befehl wollen wir gleich in der Praxis kennenlernen. Dazu folgendes Programm:

16514	ld bc,00	
16515	0	
16516	0	
16517	ld a,38	Y
16518	38	A
16519	cp N	RETURN
16520	38	A
16521	jr nz,DIS	4
16522	3	
16523	ld bc,01	
16524	1	
16525	0	
16526	ret	TAN

Dieses Programm gibt auf dem Bildschirm, nachdem es mit PRINT USR gestartet wurde, ein 1 aus, wenn der Inhalt des Akkus gleich der Zahl ist, die noch dem cp-Befehl kommt, wenn die Zahlen ungleich sind, wird eine 0 ausgegeben. Das können wir ausprobieren, wenn wir den Akku mit 39 laden oder die 38 hinter dem cp-Befehl durch eine 39 ersetzen. Gleichzeitig haben wir auch einen neuen jr-Befehl in seiner Funktion nachgewiesen.

Der cp-Befehl vergleicht immer mit dem Akku, d.h. bei cp c beispielsweise werden Akku und c-Register miteinander verglichen, bei cp (hl) werden Akku und der Inhalt der im hl-Register gespeicherten Adresse miteinander verglichen. Von z- und c-Bit kann man außer Relativsprüngen auch Direktsprünge und den ret-Befehl abhängig machen, z.B. ret z oder jp z NN.

Den Direktsprung wollen wir uns auch noch einmal kurz anschauen. Wir ersetzen in unserem Programm den Befehl jr nz DIS durch jp nz NN. Ebenso ersetzen wir die 3 durch das Niederwertige Byte der Adresse 16527 und fügen das höherwertige Byte ein (dadurch wird der ret-Befehl von 16526 nach 16527 geschoben). Das sieht folgendermaßen aus:

16521	jp nz,NN	TAB
16522	143	
16523	64	RND

Das Ergebnis bleibt gleich.

Wie man prinzipiell eine 16-Bit-Zahl in höherwertiges oder niederwertiges Byte zerlegt, ist klar oder? Einfach die Zahl durch 256 teilen, das Ergebnis vor dem Komma ist das höherwertige Byte. Das hinter dem Komma wird mit 256 multipliziert und ergibt dann das niederwertige Byte. Das können wir nachprüfen, denn  $64 * 256 + 143 = 16527$

Nun wollen wir uns etwas anderem, viel logischerem zuwenden — den logischen Verknüpfungen. Dazu müssen wir ein paar Grundlagen der Digitaltechnik über uns ergehen lassen, oder wenn diese bekannt sind, diesen Teil überspringen. Grundsätzlich besteht ein Byte aus 8 Bits. Jedes Bit kann logisch 1 oder logisch 0 sein. Nun können wir 2 Bits logisch miteinander verknüpfen und zwar folgender Arten:

Bit 1	Bit 2	1 and 2	1 or 2	1 xor 2
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Daraus ergibt sich die Möglichkeit, auch 2 Bytes, also 2 x 8 Bits miteinander zu verknüpfen. Das resultierende Byte ist dann die Summe der Verknüpfungen aus den Bits der ersten beiden Bytes.

z.B.	01001101
and	10101011
	00001001

Das kann man mit Hilfe der Spalte 1 and 2 für jedes Bit einzeln nachvollziehen. Genauso läßt sich die or und die xor Verknüpfung auf das komplette Byte übertragen.

Der Computer kennt hierfür 3 Befehle and, or und xor — wie könnte es anders sein...

Auch hier ist wieder der Akku das 1. benutzte Register. Das zweite wird beim Befehl angegeben: z.B. and d.



# TIPS & TRICKS



Jetzt wollen wir den Versuch wagen, zwei Bytes zu undieren, wie man im Fachjargon zur and-Verknüpfung sagt. Wir nehmen unsere beiden Bytes von eben, nur müssen wir sie in Dezimalzahlen umrechnen. Das geschieht folgendermaßen:

Wir nehmen das MSB, das Most-Signifikant-Bit ganz lin's mal 128, d.h. wenn es 0 ist, dann ist das Ergebnis 0. So verfahren wir weiter mit den restlichen Bits und multiplizieren sie mit 64, 32, 16, 8, 4, 2, 1. Jedes Bit mit der dazugehörigen Zahl. Für das 1. Byte erhalten wir:

$$64 + 8 + 4 + 1 = 77$$

für das 2. Byte:

$$128 + 32 + 8 + 2 + 1 = 171$$

für das Ergebnis:  $8 + 1 = 9$

Das Versuchsprogramm bestätigt unser errechnetes Ergebnis.

```
ld a,77
ld d,171
and d
ld b,0
ld c,a
ret
```

(Ab hier möchte ich es dem Leser überlassen, den Z80 Assembler-Code in ZX-verständliche Zeichen zu übersetzen und die entsprechenden Adressen einzutragen.)

Der gleiche Versuch läßt sich auch mit den Befehlen or d und xor d durchführen. Das Ergebnis wird nach bekanntem Schema, aber mit Hilfe der Spalte 1 or 2 und 1 xor 2 der Tabelle errechnet.

Wichtig für jedes umfangreichere Maschinenprogramm sind außerdem die Befehle push und pop. Um diesen Befehlen gerecht zu werden, ist es nötig, den Begriff »Stackpointer« und »Maschinenstapel« zu klären. Der Anwender braucht manchmal Platz, bearbeitet Daten mittelfristig zu speichern. Diesen Wunsch kann der Mikroprozessor mit Hilfe seines Stacks erfüllen.

Dies ist ein Speicherbereich, den die CPU direkt verwaltet. Er kann unterschiedlich lang sein und auch an verschiedenen Stellen im Speicher liegen. Sein Anfang wird durch das dafür vorgesehene sp-

Register, den Stackpointer, festgelegt. Hier kann die CPU Daten speichern und zwar nach dem Stapel-Verfahren. Das, was zuletzt angekommen ist, liegt ganz unten. Dementsprechend muß die CPU diese Daten auch vor den davor abgelegten Daten abarbeiten. Mit den Befehlen push und pop kann man die Inhalte verschiedener Register auf diesen Stapel ablegen. push bc beispielsweise legt den Inhalt des bc-Registerpaares (2 Byte!) im Stack ab. Der alte Wert bleibt aber, bis das bc-Register überschrieben wird, dort erhalten. Nun kann man mit dem bc-Register andere Probleme bearbeiten. Wenn die CPU fertig ist mit der Lösung des Problems, kann sie den ursprünglichen Inhalt des bc-Registerpaares durch pop bc wieder zurückrufen. Dies kann ineinander so tief geschachtelt werden, wie der Speicher Platz bietet.

Dazu wollen wir erstmals ein sinnvolles Programm entwerfen — ein Warteschleifenprogramm.

Doch vorher ist es noch nötig, den Befehl dj nz kurz zu erklären. Dies ist ein jump-Befehl wie jr, doch für ihn muß die Bedingung erfüllt sein, daß das b-Register ungleich 0 ist. Wenn das b-Register = 0 ist, wird der Befehl ignoriert. Zusätzlich noch zum Vergleich decremientiert, also erniedrigt der Befehl noch das b-Register um 1. So wird der Befehl, wenn er mit einem Rücksprung programmiert ist, solange immer wieder eine Schleife durchlaufen, bis das b-Register gleich 0 ist. Er heißt: decrem and jump if not zero. Mit seiner Hilfe lassen sich beliebige Schleifen, ähnlich der FOR'NEXT-Schleifen, im BASIC realisieren.

Nun das Warteschleifenprogramm:

```
ld b,255
push bc
ld b,255
djnz 254
pop bc
djnz 248
ld c,0
ret
```

Nach dem Start mit PRINT USR 16514 passiert erst eine ganze Weile lang nichts, dann erscheint eine 0 auf dem Bildschirm — ein Beweis (neben dem Funktionsbeweis), daß der dj nz-Befehl und somit auch push und pop funktioniert haben. Andernfalls enthielte das b-Register noch 255, was zur Folge gehabt hätte, daß der Bildschirm 65280 anzeigen würde. Das alles gilt selbstverständlich auch wieder für alle anderen zu push enden Register. Einen Trick noch kann man mit push und pop anwenden; um das de-Register beispielsweise unkompliziert ins hl-Register

zu bekommen, kann man statt den Befehlen

```
ld h,d
ld l,e
```

welche für den ZX sehr ungünstig, weil zu POKende Codes haben,

Schreiben push de

pop hl

d.h. bei pop wird der Stack ausgelesen, egal aus welchem Register der Inhalt kam, und in das angesprochene Register übertragen. Danach wird der Stackpointer um 2 (weil 2 Bytes) erhöht und die eben gepopten Daten sind auch aus dem Stack verschwunden.

Pusht man danach, wird der Stack wieder herabgesetzt und die Speicherstellen werden neu beschrieben.

Als weiteren wichtigen Befehl möchte ich hier den call-Befehl behandeln. Er ist das Maschinensprache äquivalent zum BASIC-Befehl GOSUB. Von der Programmierung her wird er genauso behandelt wie der Befehl jp NN — die beiden folgenden Bytes bezeichnen die Adresse, an der das Unterprogramm anfängt. Der Rücksprung erfolgt durch den Befehl ret. Dabei muß darauf geachtet werden, daß im Unterprogramm für jeden push-Befehl ein pop-Befehl steht, denn sonst stimmt die Rücksprungadresse, die auch im Stack gestapelt ist, nicht mehr. Die Folge wäre, daß der Computer vom Unterprogramm aus zu einer völlig anderen Adresse zurückspringen würde, als zu der Adresse, nach dem call NN-Befehl. Das gilt übrigens auch für den Rücksprung ins BASIC, der ja auch mit einem ret-Befehl durchgeführt wird. Grundsätzlich muß immer für jedes push ein pop-Befehl ausgeführt werden. Zum call-Befehl folgendes kleines Programm:

```
16514 ld bc,00
      call 16526
      ret
      hier 6 SPACE
      eingeben!

16526 ld bc,01
      ret
```

Zuerst wird das bc-Register mit 0 geladen, dann erfolgt der Sprung ins Unterprogramm, wo bc mit 1 geladen wird und danach springt der Computer ins Hauptprogramm zurück, von wo aus er ins BASIC geht.

Die 6 SPACE sind nicht unbedingt nötig, sie dienen nur dazu, das Unterprogramm vom Hauptprogramm zu trennen.

Wenn der call-Befehl nicht funktionieren würde, würde auf dem Bildschirm 0 erscheinen. Funktioniert er, erscheint eine 1. Dem call-Befehle gibt es auch mit allen möglichen Bedingungen, wie z.B. ca-1 z NN oder call nc NN usw.

## ZX-USER CLUB



# TIPS & TRICKS

Jetzt wollen wir noch kurz die Befehle zur Einzelbitverarbeitung besprechen. Es handelt sich hier um Befehle: bit

set  
res.

Mit dem Befehl bit kann man ein einzelnes, durch den Befehl genau beschriebenes Bit eines bestimmten Registers abfragen. Der Zustand dieses Bits wird in invertierter Form von z-flag wiedergegeben, welches man danach durch jr, jp, call oder auch ret abfragen kann. »Invertierte Form« bedeutet, daß das z-flag genau das Gegenteil von dem getesteten Bit enthält. Ist dieses logisch 1, so ist das z-flag logisch 0 und umgekehrt.

Der set-Befehl setzt ein bestimmtes Bit des gewählten Registers auf 1, der res-Befehl setzt es auf 0.

Es gibt bei der Z80 CPU auch Befehle, mit denen ganze Blöcke von Bytes transportiert und umgeladen oder auch verglichen werden können. Das sind die sogenannten Blockbefehle. Fangen wir mit dem Blockladebefehl an. ldir faßt den Inhalt des hl-Registers als Anfangsadresse des Blocks auf. Der Inhalt des de-Registers wird als Zieladresse für das 1. Byte des Blocks gespeichert. Der Befehl ldir nimmt nun das 1. Byte des Blocks, kopiert es an die Zieladresse, decreментиert das bc-Register, und incrementiert das hl- und das de-Register, so daß sie auf das zweite Byte im Zielbereich zeigen. Nun fängt das Ganze wieder von vorne an, solange bis das bc-Register gleich 0, d.h. bis der Block abgearbeitet ist. Das sieht dann so aus:

```
(DE) ← (HL) ; (DE) ← DE + 1
HL ← HL + 1 ; BC ← BC - 1
```

Der Befehl lddr funktioniert im Prinzip gleich, nur das die Register de und hl decreментиert werden.

```
(DE) ← (HL) ; DE ← DE - 1
HL ← HL - 1 ; BC ← BC - 1
```

ldir heißt übrigens load, increment and repeat; ldd heißt load decrement and repeat. cpir ist ein Blockvergleichsbefehl. Er sucht einen Block, dessen Länge vom bc-Register bestimmt wird und dessen 1. Adresse im hl-Register steht, nach dem Zeichen ab, welches im Akku gespeichert ist. Ist das Zeichen gefunden, stoppt der Befehl und der nächste Befehl wird angenommen. cpdr funktioniert genauso, nur wird diesmal das hl-Register decreментиert. Es gibt noch die Blockbefehle inir und otir, doch dies sind Befehle, die nur zur Steuerung von Peripherie verwendet

werden, und sollen hier nicht besprochen werden.

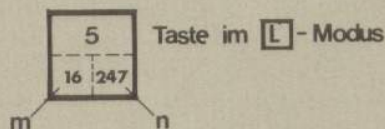
Kurz möchte ich auch noch auf den exchange-Befehl eingehen, der zwei einfache Register austauscht. ex de, hl vertauscht z.B. die Inhalte von de- und hl-Register. Doch Vorsicht! Nicht alle Register kann man ungestraft exchange. Das af-Register z.B. das aus dem Akku und dem flag-Register (es gibt noch mehr flags als nur z- und c-flags) besteht, nimmt es einem übel, wenn man es gegen das af-Registerpaar austauscht. Der Computer steigt aus.

Für den Gebrauch im Spielprogramm ist es immer nützlich, wenn man Zufallszahlen zur Verfügung hat. Zu diesem Zweck kann man 1. die Systemvariable 16436 und 16437 abfragen. Das aber hat den Nachteil, daß eine gewisse Regelmäßigkeit vorliegt und vor allem ändert sich diese Systemvariable für manche Verwendungen nicht schnell genug. 2. gibt es die Möglichkeit, das r-Register, das Refresh-Register abzufragen. Dieses Register verändert seinen Inhalt mit jedem Refresh-

Impuls, den der Computer an die dynamische RAM-Speicher abgibt. Diese benötigen den Impuls, um ihre Informationen nicht zu verlieren. Das r-Register läßt sich mit ld a,r abfragen (after EDh, Code 95). Man kann dieses Register auch verändern mit ld r, a, jedoch ist dies weder sinnvoll noch ratsam, weil im ungünstigen Fall der Computer dabei aussteigen kann.

Wenn man ein Spielprogramm schreiben will, welches während seines Ablaufs nicht ins BASIC zurückkehrt, ist es nötig, eine Tastaturabfrage einbauen. Die Tastatur ist als Port mit der Systemadresse 254 an die Z80 A angeschlossen. Eine Systemadresse ist keine Adresse im Speicherbereich, sondern sie gehört einem Port, dessen Inhalt man mit dem Befehl in auslesen kann oder an dessen Adresse man mit out Daten senden kann. Im Fall der ZX-Tastatur kann man den Port nur mit in abfragen. Da ich hier nicht weiter auf die Hardware eingehen will, möchte ich nur das Programm und die Codes-Tabelle anführen: Zu jeder Taste gehören 2 Codes, m und n; sie sind aus der Tabelle zu entnehmen:

1	2	3	4	5	6	7	8	9	0
1 247	2 247	4 247	8 247	16 247	16 239	8 239	4 239	2 239	1 239
Q	W	E	R	T	Y	U	I	O	P
1 251	2 251	4 251	8 251	16 251	16 223	8 223	4 223	2 223	1 223
A	S	D	F	G	H	J	K	L	NL
1 253	2 253	4 253	8 253	16 253	16 191	8 191	4 191	2 191	1 191
SHIFT	Z	X	C	V	B	N	M	.	SPACE
1 254	2 254	4 254	8 254	16 254	16 127	8 127	4 127	2 127	1 127



Programm: ld a,n  
ld h,m  
in a,254  
and h

Wenn die abgefragte Taste gedrückt ist, ist das z-flag gesetzt, d.h. es ist 1. Zu erwähnen wäre dann zum Schluß noch das pc-Register. pc heißt Programm-Counter. Das Register enthält die Adresse des gerade ausgeführten Befehls. Der Befehl jp NN entspricht also dem imaginären Befehl ld pc,NN. Imaginär deswegen, weil das pc-Register nicht direkt ansprechbar ist.

Viele Befehle habe ich hier nicht aufge-

führt, da dies für einen Grundlagenkurs zu weit führen würde und auch weil manche Befehle auf dem, wegen seiner Konstruktion dafür nicht geeigneten ZX 81, nicht eingesetzt werden können. Ich hoffe aber, daß diejenigen, die sich mit der Maschinensprache angefreundet haben, in Zukunft noch mehr Spaß an ihrem ZX81 haben werden und wünsche viel Erfolg.

Martin Gehrke

## ZX-USER CLUB



# PROGRAMME

## Knobel mal wieder



Dieses auf der ganzen Welt verbreitete Spiel mit Brunnen, Schere und Papier können Sie nun endlich mit Ihrem ZX81 nachvollziehen. Wer als erster fünf Punkte erreicht, hat gewonnen.

```

5 LET U=VAL "1"
8 LET U=VAL "3"
10 DIM A$(U,7)
15 LET A$(U)="BRUNNEN"
20 LET A$(U+W)="SCHERE"
25 LET A$(U)="PAPIER"
30 LET Z=U-W
35 LET D=Z
40 LET C=INT (RND*U)+W
50 FOR J=U TO U
60 PRINT J;";";A$(J)
70 NEXT J
80 INPUT Y
85 IF Y<W OR Y>U THEN GOTO 80
90 GOSUB 300
95 PRINT
100 IF C=Y THEN GOTO 160
110 IF C=U THEN LET C=U-W
120 IF C+W=Y THEN GOTO 180
130 PRINT "SIE GEWINNEN"
140 LET Z=Z+W
150 GOTO 200
160 PRINT "UNENTSCHIEDEN"
170 GOTO 200
180 PRINT "ICH GEWINNE"
190 LET D=D+W
200 IF D=VAL "5" OR Z=VAL "5" THEN
HEN GOTO 240
210 PAUSE VAL "100"
220 CLS
230 GOTO 40
240 PRINT
250 PRINT "PUNKTE:","IHRE PUNK
TE:","Z;"; MEINE PUNKTE:";D
290 STOP
300 CLS
305 PRINT "IHRE WAHL-";A$(Y)
315 IF C=U-W THEN LET C=U
320 PRINT "MEINE WAHL-";A$(C)
330 PRINT
340 RETURN

```

## Berechnung des Ostersonntages bis zum Jahre 1991



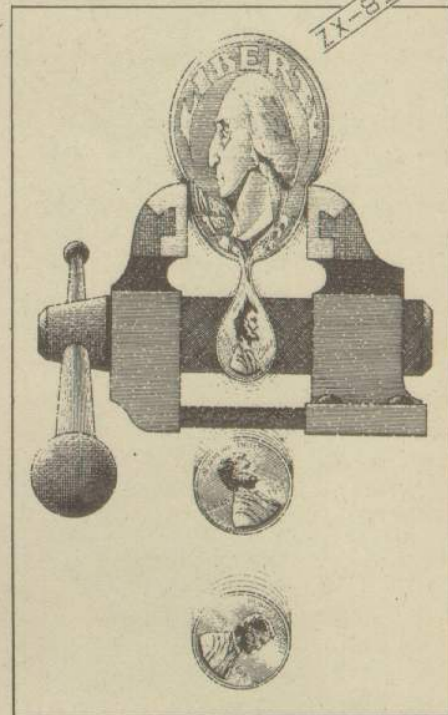
G. Papendieck, Hamburg, schickte uns sein entwickeltes Programm für die Berechnung des Datums, auf das in jedem gewünschten Jahr der Ostersonntag fällt. Das Programm arbeitet absolut genau, und ist für die Jahre 1939 und 1968 bis 1991 geprüft. Leider standen andere Osterdaten nicht zur Verfügung. Das Programm wird mit »RUN 300« gesaved und nach load »Ostersonntag« läuft es sofort an.

```

2 PRINT AT 0,2;"BERECHNUNG DE
S OSTERSONNTAGS"
3 PRINT TAB 2;";-----"
4 PRINT AT 4,0;"GEBEN SIE BIT
TE DAS JAHR EIN:"
10 INPUT Y
20 LET A=Y-(19*INT (Y/19))
30 LET B=INT (Y/100)
40 LET C=Y-(B*100)
50 LET D=INT (B/4)
55 LET E=B-(D*4)
60 LET F=INT ((B+8)/25)
70 LET G=INT ((B-F+1)/3)
80 LET H=(19*A+B-D-G+15)-INT (
(19*A+B-D-G+15)/30)*30
90 LET J=INT (C/4)
95 LET K=C-(J*4)
100 LET L=(32+2*E+2*J-H-K)-INT
((32+2*E+2*J-H-K)/7)*7
110 LET M=INT ((A+11*H+22*L)/45
1)
120 LET N=INT ((H+L-7*M+114)/31
)
130 LET P=(H+L-7*M+114)-N*31
140 LET O=P+1
150 CLS
160 PRINT AT 6,0;"DER OSTERSONN
TAG FAEHLT IM JAHR";Y;" AUF DEN
";AT 9,9;"***";O;"N;";"****"
170 PRINT AT 16,0;"DER LIEBE KL
EINE ZX81 WUENSCHT ""FROELICHE
OSTERN"";BZW. SOLCHE GEHABT ZU
HABEN ***"
180 STOP
300 SAVE "OSTERSONNTAG"
310 RUN

```

## Banko



Bei diesem Spiel müssen Sie gegen den Computer durch geschicktes Spielen und Taktieren versuchen zu gewinnen. Sie dürfen maximal neun Punkte erreichen. Erreichen Sie zehn Punkte, werden Ihnen zehn Punkte abgezogen. Erreichen Sie mehr als zwanzig Punkte, werden Ihnen zwanzig Punkte abgezogen. Also ist der höchste Punktestand, den Sie erreichen können, neun.

```

5 LET Q=9
8 LET T=5
10 LET P=Q/Q
12 LET A$="KARTE "
15 DIM X(Q-P)
20 FOR J=P TO Q-P
30 LET X(J)=INT (RND*(Q+T))
40 IF X(J)>Q THEN LET X(J)=J-J
50 NEXT J
60 PRINT A$;"1- ";X(P)
70 PRINT
75 LET X=X(P)
80 LET R=X(T)
90 FOR J=P+P TO T-P
100 PRINT A$;J;"- ";X(J)
110 LET X=X+X(J)
115 IF X>Q THEN LET X=X-Q-P
120 IF J=4 THEN GOTO 140
125 PRINT "2"
130 GOTO 130+Q*(INKEY$="J")+(Q+
T)*(INKEY$="N")
140 NEXT J
150 PRINT
160 PRINT "SIE HABEN ";X
170 PRINT
180 PRINT A$;"1- ";X(5)
190 PRINT
195 FOR J=T+P TO Q-P
200 PRINT A$;J-T+P;"- ";X(J)
205 PRINT
210 LET R=R+X(J)
220 IF R>Q THEN LET R=R-Q-P
230 IF R>T+P THEN GOTO 250
240 NEXT J
250 PRINT "ICH HABE ";R
260 PRINT
260 PAUSE 150
290 CLS
300 GOTO T

```



# PROGRAMME

## Data für ZX81

Dieses Programm ermöglicht die Verwendung der Befehle READ und RESTORE für den ZX81. Es ist auf 16K geschrieben, kann aber leicht so umgewandelt werden, daß es mit 1K-RAM genutzt werden kann. Das Programm soll hinter RAM-TOP gelegt werden, damit es nicht immerzu neu eingegeben werden muß. Dazu muß zuerst der RAMTOP um 66 Speicherplätze verschoben werden. Das geschieht mit folgendem Programm:

```
10 POKE 16388,PEEK 16388-66
20 POKE 16389,PEEK 16389-1
30 NEW
```

Zur Eingabe des Maschinenprogramms bietet sich folgendes an:

```
10 LET P=PEEK 16388+PEEK 16389
*256
20 FOR N=P TO P+66
30 INPUT A#
40 SCROLL
50 PRINT A#
60 POKE N,(CODE A#(1)-28)*16+CODE A#(2)-28
70 NEXT N
```

Starten Sie nun dieses Programm und geben Sie nacheinander die Befehlscodes ein.

```
7FBE 118140 LD DE,4081
7FC1 ED53FE7F LD(7FFE),DE
7FC5 C9 RET
7FC6 ED58FE7F LD DE,(7FFE)
7FCA 2A1040 LD HL,(4010)
7FCD 7E LD A,(HL)
7FCE FE46 CP 46
7FD0 2803 JR Z,7FD5
7FD2 23 INC HL
7FD3 18F8 JR 7FCD
7FD5 010300 LD BC,0003
7FD8 09 ADD HL,BC
7FD9 22FE7F LD(7FFE),HL
7FDC 0600 LD B,00
7FDE 3E00 LD A,00
7FE0 77 LD(HL),A
7FE1 23 INC HL
7FE2 10FC DJNZ 7FE0
7FE4 2AFE7F LD HL,(7FFE)
7FE7 010000 LD BC,0000
7FEA 13 INC DE
7FEB ED53FE7F LD(7FFE),DE
7FEF 1A LD A,(DE)
7FF0 FE1A CP 1A
7FF2 C8 RET Z
7FF3 FE76 CP 76
7FF5 28C7 JR Z,7FBE
7FF7 03 INC BC
7FF8 77 LD(HL),A
7FF9 23 INC HL
```

```
7FFA 18EE
7FFC 00
7FFD 00
7FFE 81
7FFF 40
```

Befehlscode

```
32702 11
32703 81
32704 40
32705 ED
32706 53
32707 FE
32708 7F
32709 C9
32710 ED
32711 5B
32712 FE
32713 7F
32714 2A
32715 10
32716 40
32717 7E
32718 FE
32719 46
32720 28
32721 03
32722 23
32723 18
32724 F8
32725 01
32726 03
32727 00
32728 09
32729 22
32730 FE
32731 7F
32732 06
32733 00
32734 3E
32735 00
32736 77
32737 23
32738 10
32739 FC
32740 2A
32741 FE
32742 7F
32743 01
32744 00
32745 00
32746 13
32747 ED
32748 53
32749 FE
32750 7F
32751 1A
32752 FE
32753 1A
32754 C8
32755 FE
32756 76
32757 28
32758 C7
32759 03
32760 77
32761 23
32762 18
32763 EE
32764 00
32765 00
32766 B1
32767 40
```

```
JR 7FEA
NOP
NOP
ADD A,C
LD B,B
```

```
RESTORE = RAND USR 32702
READ A# = RAND USR 32710
```

Aufbau einer Zeichenkette:

Wenn als Variablenamen also 46h.(siehe 7fce) steht, entspricht dies nun der Zeichenkette:

A\$=»ABD«

Beispielprogramm:

A\$ kann an beliebiger Stelle im Programm definiert werden. Dies sollte jedoch im günstigsten Falle die Zeile vor READ sein. Falls Sie verschiedene Wortlängen verwenden, sollte READ durch:

```
LET A#=A$(1 TO USR 32710)
```

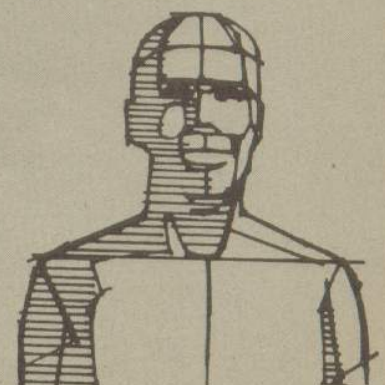
aufgerufen werden.

```
10 REM JAN,FEB,MRZ,APR,MAI,JUN
JUL,AUG,SEP,OKT,NOV,DEZ
20 POKE 32733,3
30 RAND USR 32702
40 FOR N=1 TO 12
50 LET A$=" "
60 RAND USR 32710
70 PRINT A$
80 NEXT N
```

```
10 REM JANUAR,FEBRUAR,MAERZ,APRIL,MAI,JUNI,JULI,AUGUST,SEPTEMBER,OKTOBER,NOVEMBER,DEZEMBER
20 POKE 32733,9
30 RAND USR 32702
40 FOR N=1 TO 12
50 LET A$=" "
60 LET A#=A$(1 TO USR 32710)
70 PRINT A$
80 NEXT N
```

In POKE 32733 steht die Anzahl des längsten Data-Wortes.

Georg W. Seefried, Neunhof.



Im Basicprogramm werden READ und RESTORE wie folgt aufgerufen:

# ZX-USER CLUB



Hardware,  
Bestoff und Informa  
ausfüllen.

Senden Sie heute noch Ihre Bestellung an  
Fa.  
Jürgen Schumpich GmbH  
und Software Service

Sie heute noch mit  
Fa.  
Jürgen Schumpich GmbH  
Individual Software Service  
Postfach 63 52,  
8012 Ottobrunn  
Absender nicht ver

☐ per Nachnahme ☐ per Vorausscheck  
(bitte Zutreffendes ankreuzen)

Preise incl. MwSt., zuzügl. Nachnahmegebühren, bei Lieferungen unter DM 250,- zuzügl. Porto und Verpackung.

Datum \_\_\_\_\_ Unterschrift \_\_\_\_\_  
Vergessen Sie bitte nicht auf der Rückseite Ihre Adresse!

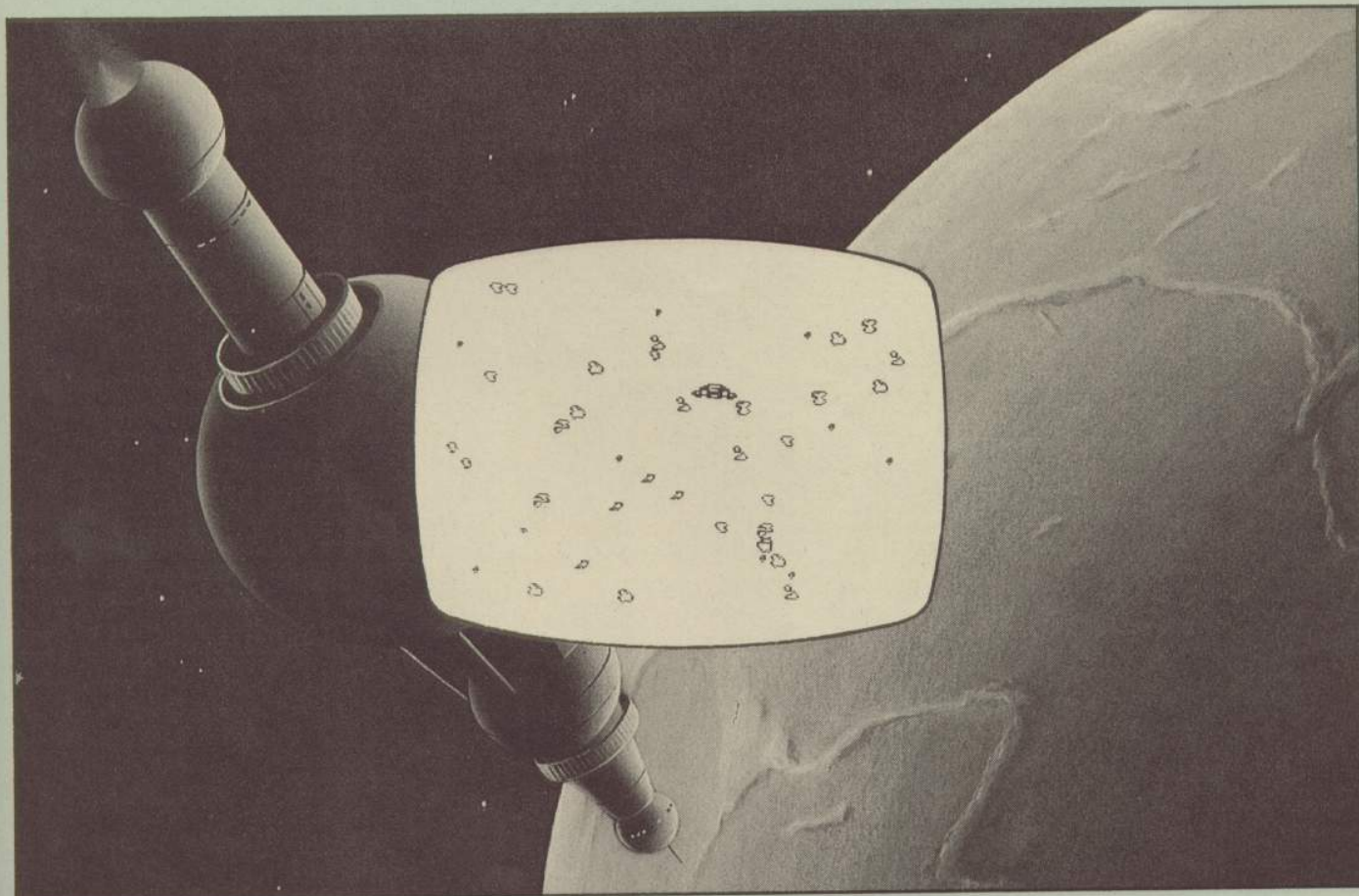
Herr Thomas Rogmann aus Goch schickte uns diesen Mini-Packman für 1 K-Besitzer zu. Natürlich kann man von 1 Kilo-byte nicht sonderlich viel erwarten, aber die Idee finden wir sehr interessant. Nach dem Starten wird zunächst das Spielfeld aufgebaut. Dann erscheint in der oberen linken Ecke ein graues Quadrat als Markierung des Startpunktes. Jetzt haben Sie eine kleine Wartepause (Zeile 18: Pause 50) zum Loslaufen. Startet man zu spät, ist das Spiel aus. Mit den Cursor-Tasten bewegen Sie sich in alle vier Richtungen. Das Ziel ist viele Punkte (».«) und vor allem (»\*)«) zu fressen, ohne an die Geister inverses »A« zu stoßen. Für einen Punkt gibt es 12 und für ein Sternchen 144 Punkte. Nachdem man losgelaufen ist, darf man auch nicht mehr stehenbleiben oder auf ein freies Feld treten.

```

1 LET A=CODE "£"
2 LET B=A-A
3 FOR X=B TO A STEP CODE "■"
4 FOR Y=B TO A
5 PRINT AT X,Y;".";AT Y,X;"."
6 NEXT Y
7 NEXT X
8 FOR X=B TO A
9 PRINT AT RAND*A,RND*A;
10 GOSUB 70
11 IF P<>CODE "." THEN GOTO CODE "."
12 PRINT "£" AND X<=CODE "■";"
   " AND X>CODE "■"
13 NEXT X
14 LET X=B
15 LET Y=B
16 LET L=B
17 PRINT AT B,B;"■"
18 PAUSE VAL "50"
19 PRINT AT Y,X;" "
20 LET A$=INKEY$
21 LET X=X+(A$="8")-(A$="5")
22 LET Y=Y+(A$="6")-(A$="7")
23 PRINT AT Y,X;
24 GOSUB 70
25 PRINT "■"
27 IF P=CODE "■" OR P=CODE " "
THEN GOTO 50
28 LET L=L+(A AND P=CODE ".")+
(A AND P=CODE "£")
29 GOTO CODE "<"
50 PRINT AT B,B;L;"SUBT TC "
52 PAUSE 4E4
53 CLS
54 RUN
70 LET P=PEEK (PEEK 16396+PEEK
16399*255)
71 RETURN

```





## Asteroids

Sie befinden sich im Weltraum mit Ihrem Raumschiff, müssen nun den immer zahlreicher werdenden Asteroiden versuchen auszuweichen. Das erfolgt mit den Tasten 5 und 8 und mit der Taste 0 können Sie einen Schutzschirm aufbauen. Aber Achtung, Ihnen stehen nur begrenzte Energieeinheiten zur Verfügung! Geben Sie bitte die unterstrichenen Buchstaben im Programm im Graphikmodus ein.

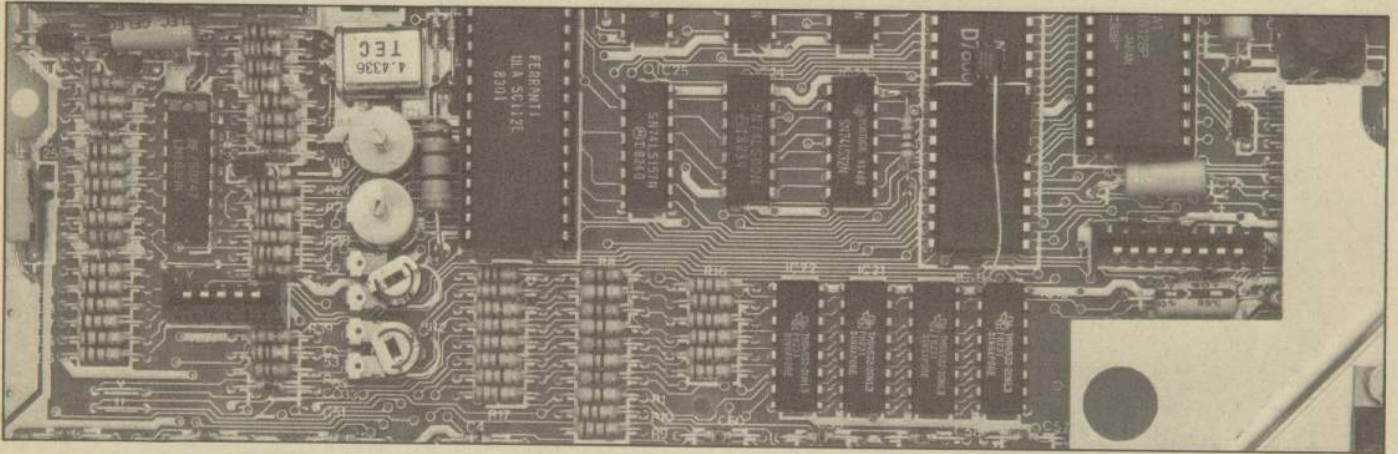
```
1 BORDER 1: PAPER 1: CLS : IN
K 7: PRINT AT 0,8: FLASH 1:"Aste
roiden" FLASH 0:"Sie fliegen m
it Ihrem Raumschiff mitten durch
einen Asteroiden- schwarm. Mit
den Tasten 5 und 8 muessen Sie v
ersuchen diesem auszuweichen.
Mit der Taste 0 koennen Sie b
is zu 5mal einen Schutzschirm
aufbauen" FLASH 1:"Sie haben n
ur fuenf Raumschiffe" "Druecken
Sie eine Taste"
2 GO SUB 9000: PAUSE 0: LET b
=0: LET e=0
10 INK 0: CLS : LET t=0: LET p
=5: LET l=15: LET ll=1: LET f=6
50 POKE 23692,255: LET l=1+(IN
KEY$="8" AND 1<29)-(INKEY$="5" A
ND 1>0): IF INKEY$="0" THEN LET
f=f-1: IF f>0 THEN PRINT AT 8,1:
INK 6:"PQR": GO TO 64
62 IF INKEY$="0" THEN LET f=f-
1: IF f>0 THEN PRINT AT 8,1: INK
6:"PQR"
64 IF ATTR (8,1+2)=15 OR ATTR
(8,1)=15 THEN LET p=p-1
```

```
66 IF ATTR (8,1+1)=15 THEN LET
p=p-2
67 IF p<1 THEN GO TO 400
99 BEEP 0.009,1
100 PRINT AT 7,11:" " AT 21,3
1:"": PRINT : PRINT AT 7,1: INK
6:"S": INVERSE 1:p: INVERSE 0:"T
": LET t=t+1: LET ll=1: PRINT AT
21,RND*31: INK 7:CHR$ (RND*11+1
44): AT 21,RND*31:CHR$ (RND*11+14
4): GO TO 50
400 FOR o=0 TO 1: FOR i=1 TO 15
: BEEP 0.009,i: NEXT i: NEXT o
401 FOR n=1 TO 3: PRINT AT 6,11
:" " AT 7,11: INK 2:" M " AT 8
,11:" " : PAUSE 2: PRINT AT 6,1
1: INK 6:"OQU": AT 7,11:"N N": AT
8,11:"PQR": PAUSE 2: NEXT n: FOR
n=6 TO 8: PRINT AT n,11:" " :
NEXT n
402 CLS : IF b<t-13 THEN LET b=
t-13
403 INK 7: IF e<b THEN LET e=b:
PRINT AT 6,0:"Sie haben neuen R
ekord aufge- stellt" "Geben Si
e Ihre Initialen ein": LET n$=""
: FOR n=1 TO 2: PAUSE 0: LET n$=
n$+INKEY$+"": NEXT n: CLS
405 PRINT : Punkte="t-13,"R
ekord="b" Hoechstrekord="":
PRINT FLASH 1:e:" von "n$":
FOR n=1 TO 150: NEXT n: PRINT F
LASH 0:"Druecken Sie ""p"" fuer
ein neues Spiel""
s zum abspeichern""
f fuer Ende": PAUSE 0: GO TO (5
AND INKEY$="p")+(430 AND INKEY$=
"s")+ (450 AND INKEY$="f")
430 SAVE "asteroids" LINE 1
450 PRINT : "Danke fuer das
Spiel " "Bye""
460 GO TO 460
9020 RESTORE 9070: FOR m=1 TO 21
9030 READ p$
```

```
9040 FOR n=0 TO 7
9050 READ a: POKE USR p$+n,a
9060 NEXT n: NEXT m: RETURN
9070 DATA "a",62,33,255,129,129,
65,37,26
9080 DATA "b",112,148,138,65,129
,129,94,32
9090 DATA "c",0,0,24,36,36,56,0,
0
9100 DATA "d",0,0,48,40,48,32,0,
0
9110 DATA "e",0,24,40,40,16,0,0,
0
9120 DATA "f",112,136,136,68,40,
48,0,0
9130 DATA "g",0,0,0,56,44,16,0,0
9140 DATA "h",0,0,30,18,34,84,23
2,0
9150 DATA "i",94,177,130,228,34,
65,66,60
9160 DATA "j",0,62,65,66,34,17,1
4,0
9170 DATA "k",96,144,144,116,10,
49,65,126
9180 DATA "l",62,65,242,9,247,13
6,144,98
9190 DATA "m",145,82,16,7,244,8,
74,137
9200 DATA "n",100,24,198,0,222,0
,24,102
9210 DATA "o",0,4,34,18,1,68,50,
9
9220 DATA "p",9,50,68,1,18,34,4,
0
9230 DATA "q",0,8,170,42,73,65,1
37,137
9240 DATA "r",144,76,34,128,72,6
8,32,0
9250 DATA "s",0,3,15,15,120,200,
255,56
9260 DATA "t",0,192,240,240,30,1
9,255,28
9270 DATA "u",0,32,68,72,128,36,
76,144
```



## Rund um den Spectrum



### WARUM SPECTRUM?

Diese Frage werden sich bestimmt schon viele gestellt haben. Warum heißt der neue Rechner von Sinclair Spectrum und nicht ZX82. Er ist doch eine verbesserte und zugegeben, in vielen Punkten völlig neue Version des Erfolgscomputers ZX81. Vielleicht weil der Spectrum ein wesentlich breiteres Spektrum an Möglichkeiten bietet, die der ZX81 nur über Zusatzmodule hat oder überhaupt nicht.

Die wohl augenfälligste Änderung ist die Tastatur. Es wurde, so meint man, keine Folientastatur mehr verwendet, sondern Gummitasten. Bei jedem Tastendruck spürt man jetzt, daß der Finger die Taste trifft und nicht den Steg dazwischen. Unter den Gummitasten sitzt allerdings wieder die altbewährte Folientastatur. Verständlich, die Folie hat sich beim ZX81 über eine Million mal bewährt. Warum nicht beim Spectrum darauf zurückgreifen. Das Betriebssystem des Spectrum wurde stark erweitert. Umfaßte es beim ZX81 8K-Byte, so belegt es jetzt das Doppelte. Eine Anzahl neuer Befehle stehen zur Verfügung, die einem das Spiel oder die Arbeit mit dem Computer erleichtern.

Am aufwendigsten wurde die Displaymöglichkeit verstärkt. 8 Farben können dargestellt werden und diese auch aufgehellt werden, so daß eigentlich nochmal 8 Abstufungen entstehen. Der Bildschirm wurde unterteilt in den Rand, den Hintergrund und den Vordergrund. Benannt wurde das von Sinclair sehr lustig, aber für jeden sofort verständlich mit Rand, Papier und Tinte. Normalerweise läßt sich die Farbe nur in der Matrixgröße 8 x 8 verändern, aber mit kleinen Tricks läßt sich da einiges machen. Weitere Bildschirm-

gags können mit »Flash« und »Invers« gemacht werden. »Invers« läßt das dargestellte Bild oder Teile davon invertiert erscheinen. »Flash« ist die Steigerung davon. Der Bildschirm schaltet in schnellen Rhythmus von invertierte in normale Darstellung. Das für den ZX81 erhältliche High-Resolution-Graphics-Modul ist beim Spectrum bereits in dem neuen Rom enthalten. Jeder Bildschirmpunkt kann einzeln angesprochen werden. Durch den Befehl »Draw« wird ein zweiter Punkt definiert und zwischen beiden eine Linie gezogen. Aber nicht nur für Linien gibt es einen speziellen Befehl, sondern auch um Kreise zu zeichnen.

Mit »Circle« wird der Mittelpunkt angegeben und der Radius des Kreises und der Spectrum zeichnet Kreise in jeder Größe in einer tollen Geschwindigkeit. Speziell beim Gebrauch der Graphik wird deutlich, daß der Spectrum wesentlich schneller ist im Vergleich zum ZX81.

Jedem, der den ZX81 kennt, wird auffallen, daß der Spectrum weniger Graphiksymbole auf der Tastatur hat. Allerdings können die Symbole mit Shift in die invertierte Darstellung umgeschaltet werden. Eine der tollsten Möglichkeiten, die der neue Sinclair-Computer offenbart, ist das freie definieren von 21 Zeichen. Jeder Punkt in einer Matrix von 8 x 8 kann belegt werden.

So kann jeder sein Ufo, Männchen oder sogar komplette Schriftarten entwerfen. Mittels eines Beep-Kommandos lassen sich dem Spectrum auch Töne entlocken. Die Skala umfaßt 10 Oktaven, die in 130 Halbschritte unterteilt werden.

Die wesentlichste Verbesserung fand bei der Load- und Save-Routine statt. Nicht nur, daß der Spectrum 6 mal so schnell ist wie sein kleiner Bruder, er ist auch bei wei-

ten nicht so empfindlich bezüglich der Lautstärke. Beim Laden wird der Programmname auf den Bildschirm geschrieben. Man merkt also nicht erst nach 6 Minuten Laden, daß es ja gar nicht das richtige Programm ist. Es lassen sich jetzt auch Daten allein speichern und wieder laden.

Der Vorteil, von einem Programm in ein anderes Daten übernehmen zu können, ist gerade bei großen Datenmengen sehr hilfreich. Aber nicht nur Daten können ohne Programm gesaved werden, auch der Bildschirminhalt.

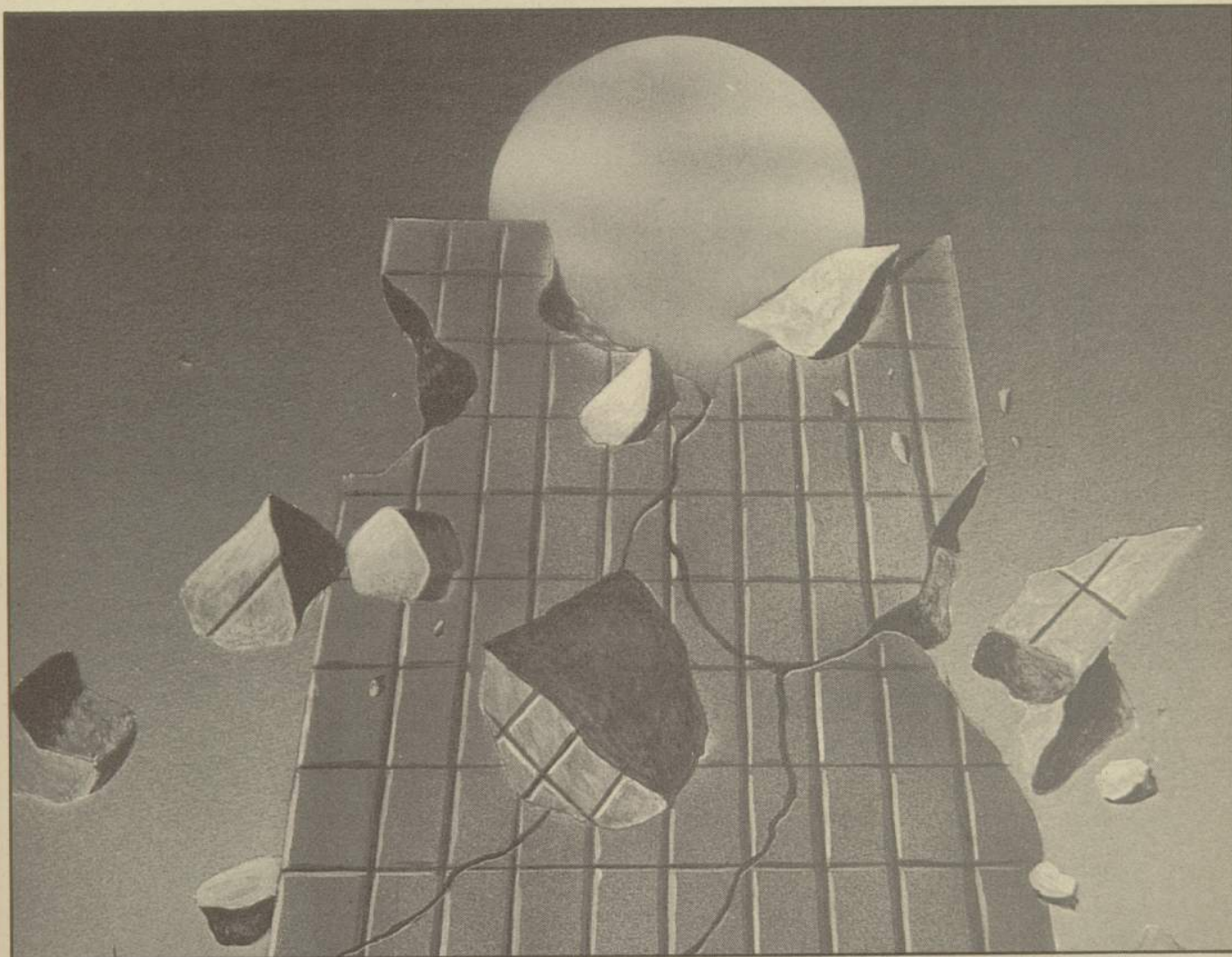
Für Maschinensprachekenner sei gesagt, daß jeder Teil des Speichers gesaved und an jede Stelle des Speichers wieder geladen werden kann. Blocktransfer per Cassettenrecorder! Nicht nur Daten, sondern auch Programme können gemischt werden. Wir haben einen Test gemacht, indem wir ein Programm mit geraden und eines mit ungeraden Zeilennummern zusammengeladen haben. Der Spectrum macht das einwandfrei! Selbst wenn manche Zeilennummern doppelt auftauchen, bricht er den Ladevorgang nicht ab, sondern überschreibt die vorhandene Zeilennummer.

Eine Anzahl von Befehlen, die der Spectrum bereits auf der Tastatur hat, kann noch nicht benutzt werden, da die Microdrives noch nicht auf dem Markt sind. Was aber verwendet werden kann, ist der ZX-Printer. Alle Zeichen, die der Spectrum darstellen kann, werden auf dem Drucker auch ausgedruckt.

Alle Möglichkeiten, die der Spectrum bietet, aufzuzeigen ist in kurzer Form gar nicht möglich. Genauso wie der ZX81 hat der Spectrum noch eine Vielzahl von kleinen Tricks, die jeder selbst entdecken sollte.



# FESTUNG



»Festung« ist mehr ein intelligentes Puzzle als ein Spiel im üblichen Sinn. Es ist sehr einfach, wenn man das Geheimnis kennt, und praktisch unmöglich zu gewinnen, wenn man es nicht kennt. Der Spectrum jedenfalls kennt es und spielt daher sehr gut!

Die beiden Programme sollten wie folgt eingegeben werden:

Geben Sie »FORT-DATA« ein.

RUN. Dadurch werden die Arrays aufgebaut, aber außer der STOP-Meldung erscheint nichts auf dem Schirm.

Geben Sie CONT ein, um eine Kopie von FORT-DATA zusammen mit seinen Arrays auf Band zu speichern.

Geben Sie **NICHT** NEW ein!

Tippen Sie einfach »FESTUNG« ein und überschreiben damit die Zeilen von »FORT-DATA«.

Geben Sie **NICHT** RUN ein! Benutzen Sie GOTO 9000, um eine Kopie von FESTUNG zusammen mit den Arrays

von FORT-DATA abzuspeichern. Diese Kopie läuft automatisch ab, wenn sie ohne CLEAR geladen wird. Sie können das Programm auch sofort starten, jedoch **NICHT MIT** RUN, sondern durch GOTO 0.

Haben Sie irrtümlich RUN (oder CLEAR) beim Eintippen von »FESTUNG« eingegeben, so ist das auch kein Beinbruch. Geben Sie einfach das Programm weiter ein, dann:

Geben Sie GOTO 9000 zum Speichern ein

Geben Sie LOAD »FORT-DATA« ein

Laden Sie »FORT-DATA« vom Band, und es läuft ab und zeigt die STOP-Meldung wie vorher.

Jetzt geben Sie MERGE »FESTUNG« ein.

Schließlich geben Sie GOTO 9000 ein, um eine weitere Kopie von »FESTUNG« abzuspeichern — dieses Mal mit einwandfreien Arrays.



Warum das alles? Warum werden nicht beide Programme zu einem zusammengefaßt?

Der Grund besteht NICHT darin, daß Speicherplatz gespart werden soll, obwohl das durch diese Technik geschieht. Sie sollten sich auch daran erinnern, wenn Sie einmal längere Programme schreiben. Vielmehr sind manche Leute heutzutage so 'computer-gelehrt', daß traditionelle Problemlösungsmethoden einfach abgelehnt werden. Mit einem 'Quiz-Programm' konfrontiert würden sich viele Leute nicht die Mühe machen, die Fragen auf die übliche Art zu beantworten, sondern einfach das Programm BREAKen und im Listing nach den gesuchten Antworten schauen. Ich habe viele kommerzielle Quiz-Programme gesehen, wo Fragen und Antworten tatsächlich in der gleichen Zeile des Programms stehen!

Die Moral davon ist, daß es nicht mehr länger genügt, die Programme 'idiotensicher' zu machen, sie müssen außerdem auch noch 'geniesicher' sein!

In »FESTUNG« ist die Lösung nicht aus dem Listing ersichtlich, sondern der Schlüssel ist in den Zahlen enthalten, die im Array c() abgelegt sind. Bei Anwendung dieser Zwei-Programm-Technik ist es möglich, auf dieses Array zuzugreifen, ohne daß sein Inhalt im Listing offenbart wird.

## SPIELREGELN

FESTUNG kann von zwei Spielern oder einem Spieler und dem Computer gespielt werden. Die beiden Spieler sind der ANGREIFER und der VERTEIDIGER. Das Spielfeld stellt ein Schloß dar mit 18 Räumen, die mit den Buchstaben A bis R markiert sind. Jeder Raum hat 4 Ausgänge, von denen jeder zu einem angrenzenden Zimmer führt. (Ein Plan des Schlosses wird am Bildschirm gezeigt, um es näher zu erklären.) Das Ziel eines jeden Spielers ist es, den Gegner durch Betreten des Raumes, in dem sich dieser befindet, gefangenzunehmen. Es gibt gewisse Beschränkungen bei den Bewegungen der beiden Spieler.

Der Verteidiger darf den 'Hauptturm', d.h. das Dreieck, das durch die Räume A, B und D gebildet wird, nicht verlassen.

Der Angreifer darf jeden Raum betreten, aber er kann jeden Durchgang nur einmal benutzen (die Wege werden vom Schirm gelöscht, sobald sie einmal begangen wurden.) Es ist daher möglich, daß der Angreifer das Spiel verspielt, weil er in einem Raum gefangen ist, dessen vier Ausgänge bereits benutzt worden waren. Der Angreifer muß IMMER zuerst ziehen.

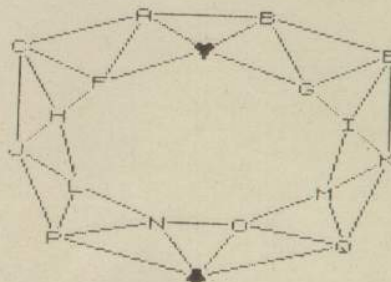
So betrachtet, hat der Verteidiger eine ziemlich langweilige Rolle. Er kann nur in den drei Räumen des Hauptturms umhergehen und darauf warten, daß der Angreifer in seine Reichweite gelangt. Versuchen Sie, ein paar Spiele gegen einen Freund zu spielen (lassen Sie ihn den Verteidiger spielen!), und Sie werden ihn vermutlich leicht schlagen. Recht bald wird er die Rolle des Angreifers übernehmen wollen. Lassen Sie ihn, aber lassen Sie den Computer verteidigen. Ich kann GARANTIEREN, daß der Computer gewinnen wird!

Die Verteidigung des Computers bei »FESTUNG« ist unschlagbar, und sein Angriff, obwohl abzuwehren, ist sehr gut. Um gewinnen zu können, müssen Sie den Verteidiger spielen, und Sie müssen entweder eine Menge Glück haben oder die gleiche Strategie übernehmen, die der Computer bei der Verteidigung anwendet. Ich werde Ihnen NICHT erzählen, worin diese Strategie besteht, aber ich habe Ihnen bereits gesagt, wo der Schlüssel zu finden ist, und der Rest hängt von Ihnen ab.

```

10 DIM p(18,2): DIM q(18,2)
20 RESTORE : FOR J=1 TO 18: RE
AD p(J,1),p(J,2)
30 LET q(J,1)=4+p(J,2)*8: LET
q(J,2)=172-p(J,1)*8
40 NEXT J
50 DIM c(18): FOR J=1 TO 18: R
EAD c(J): NEXT J
60 DIM l$(18,4): FOR J=1 TO 18
READ l$(J): NEXT J
70 STOP
1000 DATA 1,12,1,18,3,6,3,15,3,2
4,5,10,5,20,7,8,7,22,9,6,9,24,11
,9,11,21,13,13,13,17,14,8,14,22,
16,15
2000 DATA 1,2,3,3,3,2,1,1,2,2,1,
3,3,2,1,1,2,3
3000 DATA "BCDF","ADEG","AFHJ","
ABFG","BGIK","ADCH","BDEI","CFJL
","GEKM","CHLP","EIMQ","HJPN","I
KQO","LOPR","MNOR","JLNR","MKOR"
,"NOPQ"
9000 SAVE "FORT-DATA" LINE 0

```



```

1 GO TO 100
5 CLS : FOR J=1 TO 18
10 FOR K=1 TO 4
15 PLOT INK 4;q(J,1),q(J,2)
20 IF l$(J,K)>CHR$(J+64) THEN
GO TO 35
25 DRAW INK 4;q(CODE l$(J,K)-6
4,1)-q(J,1),q(CODE l$(J,K)-64,2)
-q(J,2)
30 NEXT K
35 NEXT J
40 FOR J=1 TO 18: PRINT AT p(J
,1),p(J,2);CHR$(J+64): NEXT J
50 RETURN
60 PRINT AT 21,0;e$: FOR J=1 T
O 100: NEXT J: PRINT AT 21,0; OU
ER 1;e$: RETURN
70 PRINT INK 2;AT p(ap,1),p(ap
,2);CHR$ 144: RETURN
75 PRINT INK 1;AT p(dp,1),p(dp
,2);CHR$ 145: RETURN
100 RESTORE : FOR J=USR CHR$ 14
4 TO USR CHR$ 145+7: READ a: POK
E J,a: NEXT J
150 BORDER 6: PAPER 6: BRIGHT 1
: INK 9: CLS
160 DIM m$(18,4): FOR J=1 TO 18
LET m$(J)=l$(J): NEXT J
170 LET ap=18: LET dp=4
180 DIM e$(20)
190 LET m$="": LET g=0: RANDOMI
ZE
200 CLS : PRINT TAB 11;"FESTUNG
": AT 5,0;"1: 1 Spieler""2: 2 S
pieler"
210 LET i$=INKEY$: IF i$<"1" OR
i$>"2" THEN GO TO 210
220 LET no=0: IF i$="1" THEN GO
TO 3000
230 GO SUB 5: GO SUB 70: GO SUB
500
240 IF no<>100 THEN GO TO 1000
250 GO TO 8500
260 IF no<>97 THEN GO TO 2000
270 GO TO 8000
1000 PRINT AT 21,0;"Angreifer am
Zug (" : INK 2;CHR$ 144: INK 9;")

```



# ZX-SPECTRUM PROGRAMME

```

1010 IF m$(ap)="" THEN LET w$="VERTEIDIGER": GO TO 7000
1020 LET i$=INKEY$
1030 IF i$<"a" OR i$>"r" THEN GO TO 1020
1040 LET e$="KEIN DURCHGANG"
1050 FOR j=1 TO 4
1060 IF m$(ap,j)=CHR$(CODE i$-32) THEN GO TO 1100
1070 IF m$(ap,j)=CHR$(42 AND i$(ap,j)=CHR$(CODE i$-32) THEN LET e$="DURCHGANG BENUTZT": LET j=4
1080 NEXT j
1090 GO SUB 60: GO TO 1000
1100 LET nap=CODE i$-96
1110 LET m$(ap,j)=CHR$(42
1120 FOR j=1 TO 4
1130 IF m$(nap,j)=CHR$(ap+64) THEN LET m$(nap,j)=CHR$(42

```

```

1140 NEXT j
1150 IF nap=dp THEN LET w$="ANGREIFER"
1160 LET x=ap: LET y=nap: IF nap>ap THEN LET x=nap: LET y=ap
1170 PLOT q(x,1),q(x,2)
1180 DRAW OVER 1: INK 4: q(y,1)-q(x,1),q(y,2)-q(x,2)
1190 PRINT AT p(ap,1),p(ap,2):CHR$(ap+64)
1200 LET ap=nap
1210 GO SUB 70
1220 IF INKEY$=i$ THEN GO TO 1220
1230 IF w$="" THEN GO TO 520
1240 GO TO 7000
2000 PRINT AT 21,0:"Verteidiger zieht(," INK 1,CHR$(145): INK 9:

```

```

2010 LET i$=INKEY$
2020 IF i$="" OR i$=CHR$(13) THEN GO TO 2010
2030 IF i$="a" OR i$="b" OR i$="d" THEN GO TO 2070
2040 LET e$="BLEIBE IM BEREICH!"
2050 GO SUB 60
2060 GO TO 2000
2070 LET ndp=CODE i$-96
2080 IF ndp=dp THEN GO TO 2010
2090 IF ndp=ap THEN LET w$="VERT EIDIGER"

```

```

2100 PRINT AT p(dp,1),p(dp,2):CHR$(dp+64)
2110 LET dp=ndp: GO SUB 75
2120 IF INKEY$=i$ THEN GO TO 2120
2130 IF w$="" THEN GO TO 500
2140 GO TO 7000
3000 PRINT "A:Du bist der ANGREIFER""D:Du bist der VERTEIDIGER"
3010 LET i$=INKEY$: IF i$<>"a" AND i$<>"d" THEN GO TO 3010
3020 LET no=CODE i$: RETURN
7000 IF w$(1)="D" AND no=100 THEN LET w$="Du hast"
7010 IF no=97 OR (no=100 AND w$(1)="A") THEN LET w$="Ich habe"
7020 PRINT AT 21,0:w$:" gewonnen"
7030 IF no=2 THEN PRINT "S":
7030 PRINT "": REM 13 spaces
7040 INPUT "P= Neues Spiel :S= stop :i$
7050 IF i$<>"p" AND i$<>"s" THEN GO TO 7040
7060 IF i$="s" THEN GO TO 9999
7070 GO TO 150
8000 LET i$=CHR$(c(ap)+64+(c(ap)-3))
8010 PRINT AT 21,0:"MEIN ZUG "i$:" (Druecke ENTER)"
8020 IF INKEY$<>CHR$(13) THEN GO TO 8020

```

```

8030 PRINT AT 21,17:"": RE M 6 spaces
8040 LET i$=CHR$(CODE i$+32)
8050 GO TO 1050+(1020 AND no=97)
8500 LET g=g+1: IF g=1 THEN LET i$=m$(ap,INT (RND*4)+1): GO TO 8010
8510 DIM g(4): FOR j=1 TO 4
8520 LET pm=CODE m$(ap,j)-64
8530 IF pm=-22 THEN LET g(j)=-30: GO TO 8600
8540 IF c(pm)=c(dp) THEN LET g(j)=15: GO TO 8600
8550 IF pm=dp THEN LET g(j)=30: GO TO 8600
8560 IF pm<3 OR pm=4 THEN LET g(j)=-10: GO TO 8600
8570 FOR k=1 TO 4
8580 IF m$(pm,k)=CHR$(42) THEN LET g(j)=g(j)-1
8590 NEXT k
8600 NEXT j
8610 LET k=g(1)
8620 FOR j=2 TO 4: IF g(j)>k THEN LET k=g(j)
8630 NEXT j
8640 IF k=-30 THEN LET w$="VERTEIDIGER": GO TO 7000
8650 FOR j=1 TO 4: IF g(j)<>k THEN NEXT j
8660 LET i$=m$(ap,j): GO TO 8010
9000 SAVE "FESTUNG" LINE 0
9500 DATA 60,60,60,126,255,255,60,60,102,255,255,126,60,60,24,24
9999 STOP

```

## ANMERKUNGEN ZUM PROGRAMM

Die durch »FORT-DATA« aufgebauten Arrays sind folgende:  
p (18,2): DATA Zeile 1000: Die Koordinaten auf dem Schirm für die 18 Räume des Schlosses.

q (18,2): Die gleichen Bildschirmpositionen, umgewandelt in PLOT-Koordinaten. (Array p wird benötigt, um die Buchstaben der Zimmerkennzeichnung und die Position der Spieler anzuzeigen. Mit Hilfe von Array q werden die Verbindungswege gezeichnet (DRAW).

c (18): DATA Zeile 2000: das können Sie selbst herausfinden!  
i\$ (18,4): DATA Zeile 3000: die »erlaubten Züge« von jedem Raum zu Beginn des Spiels. Dieses Array wird später in ein anderes übertragen (m\$(18,4)), und wenn Durchgänge benutzt wurden, werden Elemente von m\$ mit Sternchen (CHR\$(42)) überschrieben. i\$ bleibt unverändert und bereit fürs nächste Spiel.

»FESTUNG« beginnt (bei Zeile 100) mit der Definition von zwei Graphikzeichen und zwar CHR\$(144) für den Angreifer und CHR\$(145) für den Verteidiger. Die davor liegenden Zeilen enthalten vier Unterprogramme zum Zeichnen des 'Spielfeldes' (5 bis 50), Aufgabe einer Fehlermeldung (60), Ausdrucken des Angreifers (70) und Ausdrucken des Verteidigers (75).

Die vier Hauptteile des Programms sind:

- 1000-1240: Zug Angreifer (Eingabe durch menschlichen Spieler)
- 2000-2140: Zug Verteidiger (Eingabe durch menschlichen Spieler)
- 8000-8050: Zug Verteidiger (Computer)
- 8500-8650: Zug Angreifer (Computer)

Die vier Zeilen 500 bis 530 kontrollieren die Reihenfolge, in der diese Programmteile (nur zwei davon werden in jedem Spiel benötigt), aufgerufen werden.



# PROGRAMME

## Scroll in alle Richtungen Für 16 K und 48 K-Spectrum

Dieses Demonstrationsprogramm ermöglicht Ihnen das Scrollen des Bildschirmhaltes in jede Richtung. In der Variable »Spalten« steht die Anzahl der durchzuführenden Scrolls.

```

10 REM SCROLL
20 CLEAR PEEK 23732+256*PEEK 2
3733-168-256
30 LET ramtop=PEEK 23730+256*P
EEK 23731+1
40 GO SUB 500
60 LET links=ramtop+67
70 LET rechts=ramtop+100
80 LET oben=ramtop+153
90 LET unten=ramtop+200
100 LET spalten=23681
110 POKE spalten,8
120 CLS : PRINT AT 10,11;"Scrolls"
130 FOR x=1 TO 2: RANDOMIZE USR
links: RANDOMIZE USR rechts: NE
XT x
140 PRINT #0;AT 0,0;"Druecken S
ie eine Taste": PAUSE 0
150 CLS
160 PRINT AT 2,3;"Dieses Demons
tationspro-gramm ermoeeglic
ht Ihnen das Scrollen des B
ild-"schirminhaltes in jede
"Richtung. In der Variable"
"spalten" steht die Anzahl
der durchzufuehrenden"
Scrolls."
170 PRINT #0;AT 0,0;"Druecken S
ie eine Taste und sehen Sie
zu!": PAUSE 4e4
180 POKE spalten,16
190 FOR x=1 TO 3
200 RANDOMIZE USR links
210 RANDOMIZE USR oben
220 RANDOMIZE USR rechts: RANDO
MIZE USR rechts
230 RANDOMIZE USR unten: RANDO
MIZE USR unten
240 RANDOMIZE USR links
250 RANDOMIZE USR oben
260 NEXT x
270 CLS
280 PRINT AT 10,0;
290 FOR x=0 TO 31: PRINT PAPER
x-INT (x/8)*8;" ";: NEXT x
300 PRINT AT 10,2; PAPER 8; INK
9;"<Farben werden nicht bewegt>"
310 PAUSE 100
320 POKE spalten,24
330 RANDOMIZE USR rechts: RANDO
MIZE USR links
340 RANDOMIZE USR links: RANDO
MIZE USR rechts
350 POKE spalten,8: PRINT AT 14
,0;
360 FOR x=0 TO 31: PRINT PAPER
7*(x-INT (x/2)*2);" ";: NEXT x
370 PRINT AT 14,3; PAPER 8; INK
0;" Z X - S P E C T R U M"
380 FOR x=1 TO 3: PAUSE 100: RA
NDOMIZE USR links: PAUSE 100: RA
NDOMIZE USR rechts: NEXT x
390 RANDOMIZE USR links
400 PRINT AT 21,10;"VIEL SPASS!"
410 POKE spalten,176

```

```

420 RANDOMIZE USR oben
430 CLS
440 STOP
500 FOR i=ramtop+67 TO ramtop+2
50: READ a: POKE i,a: NEXT i: RE
TURN
510 REM Scroll nach links
520 DATA 58,129,92,79,33,255,63
,6,192,17,32,0,25,229,175,126,23
,119,43,29,32,249,225,16,240,13,
32,232,201,0,0,0
530 REM Scroll nach rechts
540 DATA 58,129,92,79,33,0,64,6
,192,30,32,175,126,31,119,35,29,
32,249,16,244,13,32,236,201,59,5
9,227,35,35,227,251,245,245,120,
230,24,246,64,87,241,130,87,120,
230,7,15,15,15,129,95,241,201
550 REM Scroll nach oben
560 DATA 58,129,92,71,197,175,7
1,79,243,205,82,0,24,214,60,254,
8,56,7,4,120,254,24,48,18,175,23
5,243,205,82,0,24,195,235,197,1,
32,0,237,176,193,24,221,193,16,2
14,201
570 REM Scroll nach unten
580 DATA 58,129,92,71,197,62,7,
6,23,14,0,243,205,82,0,24,164,23
5,61,254,255,32,8,5,120,254,255,
40,18,62,7,243,205,82,0,24,144,2
35,197,1,32,0,237,176,193,24,220
,193,16,210,201

```

### nach links

```

FE9B 3A815C LD A,(5C81)
FE9E 4F LD C,A
FE9F 21FF3F LD HL,3FFF
FEA2 06C0 LD B,C0
FEA4 112000 LD DE,0020
FEA7 19 ADD HL,DE
FEA8 E5 PUSH HL
FEA9 AF XOR A
FEAA 7E LD A,(HL)
FEAB 17 RLA
FEAC 77 LD (HL),A
FEAD 2B DEC HL
FEAE 1D DEC E
FEAF 20F9 JR NZ,-7FEAA
FEB1 E1 POP HL
FEB2 10F0 DJNZ -16FEA4
FEB4 0D DEC C
FEB5 20E8 JR NZ,-24FE9F
FEB7 C9 RET
FEB8 00 NOP
FEB9 00 NOP
FEBA 00 NOP
FEBB 00 NOP

```

### nach rechts

```

FEBB 3A815C LD A,(5C81)
FEBF 4F LD C,A
FEC0 210040 LD HL,4000
FEC3 06C0 LD B,C0
FEC5 1E20 LD E,20
FEC7 AF XOR A
FEC8 7E LD A,(HL)
FEC9 1F RRA
FECA 77 LD (HL),A
FECB 23 INC HL
FECC 1D DEC E
FECD 20F9 JR NZ,-7FECE8
FECF 10F4 DJNZ -12FECE5
FED1 0D DEC C
FED2 20EC JR NZ,-20FECC
FED4 C9 RET
FED5 3B DEC SP
FED6 3B DEC SP
FED7 E3 EX HL,(SP)
FED8 23 INC HL
FED9 23 INC HL
FEDE E3 EX HL,(SP)
FEDB FB EI
FEDC F5 PUSH AF
FEDD F5 PUSH AF
FEDE 78 LD A,B
FEDF E618 AND 18
FEE1 F640 OR 40

```

```

FEE3 57 LD D,A
FEE4 F1 POP AF
FEE5 82 ADD A,D
FEE6 57 LD D,A
FEE7 78 LD A,B
FEE8 E607 AND 07
FEEA 0F RRCA
FEEB 0F RRCA
FEEC 0F RRCA
FEED 81 ADD A,C
FEEF 5F LD E,A
FEFF F1 POP AF
FEF0 C9 RET

```

### nach oben

```

FEF1 3A815C LD A,(5C81)
FEF4 47 LD B,A
FEF5 C5 PUSH BC
FEF6 AF XOR A
FEF7 47 LD B,A
FEF8 4F LD C,A
FEF9 F3 DI
FEFA CD5200 CALL 0052
FEFD 18D6 JR -42FEF5
FEFF 3C INC A
FF00 FE08 CF 08
FF02 3807 JR C,+7FF0B
FF04 04 INC B
FF05 78 LD A,B
FF06 FE18 CP 18
FF0B 3012 JR NC,+18FF1C
FF0A AF XOR A
FF0B EB EX DE,HL
FF0C F3 DI
FF0D CD5200 CALL 0052
FF10 18C3 JR -61FEF5
FF12 EB EX DE,HL
FF13 C5 PUSH BC
FF14 012000 LD BC,0020
FF17 EDB0 LDIR
FF19 C1 POP BC
FF1A 18DD JR -35FEF9
FF1C C1 POP BC
FF1D 10D6 DJNZ -42FEF5
FF1F C9 RET

```

### nach unten

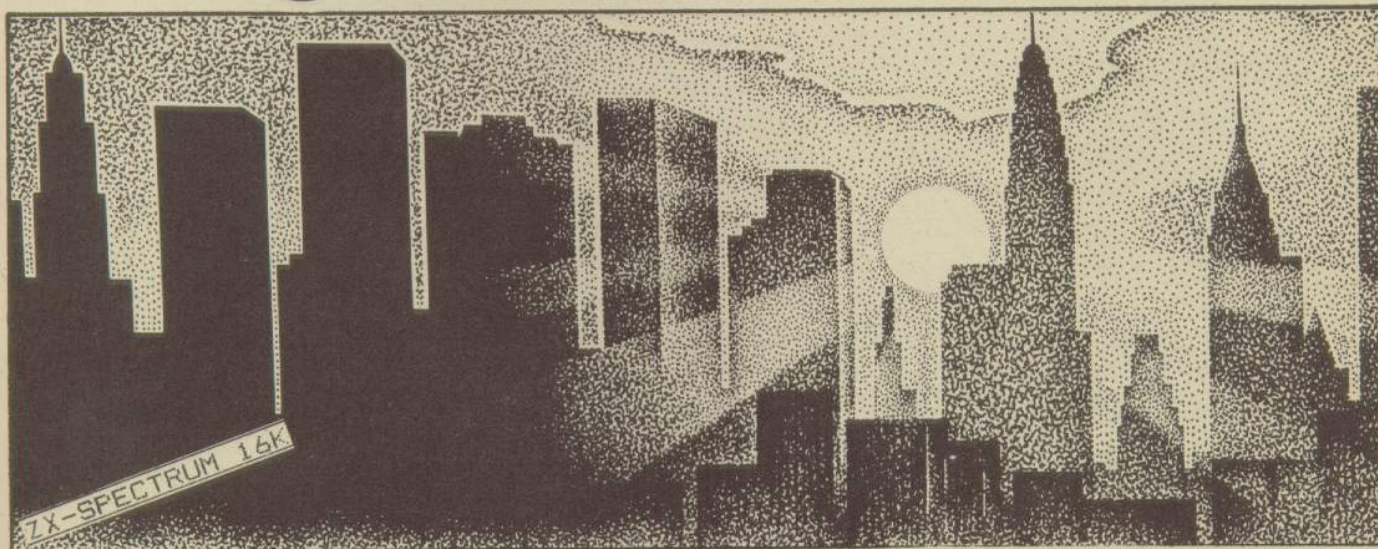
```

FF20 3A815C LD A,(5C81)
FF23 47 LD B,A
FF24 C5 PUSH BC
FF25 3E07 LD A,07
FF27 0617 LD B,17
FF29 0E00 LD C,00
FF2B F3 DI
FF2C CD5200 CALL 0052
FF2F 18A4 JR -92FEF5
FF31 EB EX DE,HL
FF32 3D DEC A
FF33 FEFF CF FF
FF35 2008 JR NZ,+8FF3F
FF37 05 DEC B
FF38 78 LD A,B
FF39 FEFF CF FF
FF3B 2812 JR Z,+18FF4F
FF3D 3E07 LD A,07
FF3F F3 DI
FF40 CD5200 CALL 0052
FF43 1890 JR -112FEF5
FF45 EB EX DE,HL
FF46 C5 PUSH BC
FF47 012000 LD BC,0020
FF4A EDB0 LDIR
FF4C C1 POP BC
FF4D 18DC JR -36FF2B
FF4F C1 POP BC
FF50 10D2 DJNZ -46FF24
FF52 C9 RET

```



## Farbig durch den Winter



Mit diesem herrlichen Skiprogramm können Sie leider nur Ihre Finger trainieren. Sie fahren durch Wälder und haben viele Hindernisse zu umfahren. In voller Geschwindigkeit frontal gegen einen Tannenbaum zu fahren, bedeutet das Ende für beide.

```

10 CLS : BORDER 2
20 LET h=0
30 DATA "c",60,60,60,24,255,60
36,102
40 DATA "d",16,16,56,16,124,16
,255,16
50 DATA "e",66,126,24,90,90,21
9,24,24
60 LET u=PEEK 23675+256*PEEK 2
3676
70 FOR i=1 TO 3
80 READ q$
90 FOR r=0 TO 7
100 READ j
110 POKE USR q$+r,j
120 NEXT r
130 NEXT i
140 CLS
150-PRINT INK 1; PAPER 6;AT 2,1
2;"Ski Rennen"
160 PRINT AT 6,2;"Taste '8' fah
rt nach rechts"
170 PRINT AT 8,2;"Taste '5' fah
rt nach links"
180 PRINT AT 12,4;"Leichte Abfa
hrt 'L'"
190 PRINT AT 14,4;"Schwere Abfa
hrt 'S'"
200 PRINT ""Hindernisse 8,4,7
und ""
210 IF INKEY$="1" THEN LET w=10
220 IF INKEY$="1" THEN GO TO 26
0
230 IF INKEY$="s" THEN LET w=7
240 IF INKEY$="s" THEN GO TO 26
0
250 GO TO 150
260 CLS
270 LET t=0
280 LET a=10
290 LET b=0
300 LET x=13
310 LET y=12
320 LET k=INT (RND*2)
330 LET l=INT (RND*3)

```

```

340 LET a=a-(k=1 AND a>1)+(k=0
AND a<21)
350 LET b=b-(l=2 AND b>1)+(l=0
AND b<10)
360 PRINT AT y,x-1; INK 1;"8"
370 PRINT AT y-1,x-2;" ";AT y
-2,x-2;" "
380 PRINT AT 20,a; INK 4;"*";TA
B a+w;"*"
390 IF b=2 THEN PRINT INK 2;AT
20,a+INT (w/2)-2;"4"; INK 7;AT 2
1,a+(w/2)-2;" "
400 IF b=7 THEN PRINT INK 3;AT
20,a+INT (w/2)-2;"8"; INK 7;AT 2
1,a+(w/2)-2;" "
410 IF b=3 THEN PRINT INK 4;AT
20,a+INT (w/2)-2;"4"; INK 0;AT 2
0,a+(w/2)-2;"4"; INK 7;AT 21,a+I
NT (w/2)+1;" "
420 IF b=8 THEN PRINT INK 2;AT
20,a+INT (w/2)-2;"8"; INK 7;AT 2
1,a+(w/2)-2;" "
430 IF b=5 THEN PRINT INK 0;AT
20,a+INT (w/2)-2;"7"; INK 7;AT 2
1,a+(w/2)-2;" "
440 PRINT
450 POKE 23692,-1; PRINT
460 PRINT INK 0; PAPER 5;AT 0,1
;"PUNKTE ";t;" "
470 PRINT INK 0; PAPER 5;AT 0,1
4;"HIGH SCORE IS ";h;" "
480 IF SCREEN$ (y,x-1)="*" THEN
GO TO 540
490 IF SCREEN$ (y,x-1)="4" THEN
GO TO 540
500 IF SCREEN$ (y,x-1)=" " THEN
GO TO 540
510 LET x=x-(INKEY$="5")+(INKEY
$="8")
520 LET t=t+1
530 GO TO 320
540 CLS
550 IF t>h THEN LET h=t
560 PRINT AT 3,5; FLASH 1; BRIG
HT 1;" Sie sind gestuerzt ! "
570 PRINT AT 8,6; FLASH 1; INK
RND*7; PAPER 9;" Ihre Punktzahl
";t;" "
580 PRINT AT 13,6; FLASH 1; INK
RND*3; PAPER 9;" HIGH SCORE is
t ";h;" "
590 PRINT AT 18,0;"Fuer neues S
piel 'N' druecken"
600 BEEP .01,RND*20-RND*20
610 IF INKEY$="n" THEN GO TO 14
0

```

```

620 GO TO 560
630 SAVE "ski run" LINE 10
640 STOP

```

Ski Rennen

Taste '8' fährt nach rechts

Taste '5' fährt nach links

Leichte Abfahrt 'L'

Schwere Abfahrt 'S'

Hindernisse 8,4,7 und "

PUNKTE 27 HIGH SCORE IS 0

```

* 8 *
* 4 *
* 7 *
*  " *

```

8=Graphic 'C'  
4=Graphic 'D'  
7=Graphic 'E'



# P ZX-SPECTRUM PROGRAMME

## Wer anderen eine Grube gräbt, fällt selbst hinein

Lassen Sie sich nicht einkreisen! Sie können sich mit den Cursor-Tasten bewegen. Jeder Schritt bringt einen Punkt. Für jedes überfahrene X gibt es Extra-Punkte.

A. Stenger, Nürnberg.

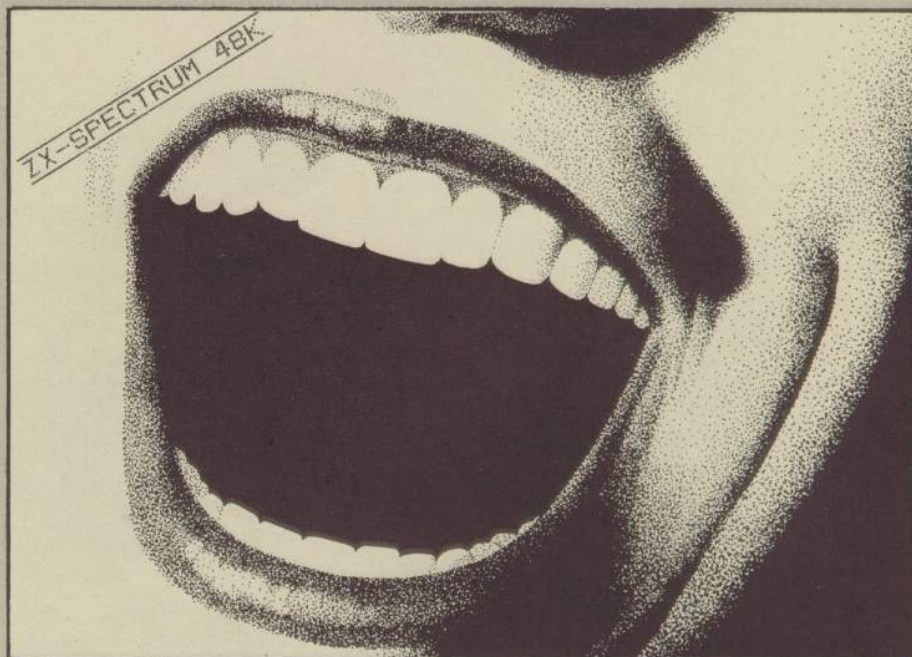
■=Graphic 'A'  
□=Graphic 'B'  
○=Graphic 'C'

```
5 RESTORE 800: FOR n=USR "a"
TO USR "a"+23: READ a: POKE n,a:
NEXT n: RESTORE
10 PAPER 0: BRIGHT 1: BORDER 1
: INK 7: CLS
20 FOR m=0 TO 24 STEP 12: FOR
n=0 TO 7: PRINT AT 8,9: INK n;"H
INTERHALT": BEEP .02,m+n: NEXT n
: NEXT m
30 PRINT "Lassen Sie sich n
icht einkreisen! Sie können
sich mit den Kursortasten bewe
gen. Jeder Schritt bringt ei
nen Punkt. Fuer jedes ueberf
ahrene x gibt es Extrapunkte. D
ruecken Sie eine Taste um anz
ufangen!"
```

```
40 PAUSE 0
50 LET h=0
60 RESTORE: LET r=0: LET m=80
: LET s=0: LET x=16: LET y=6
70 PAPER 1: BORDER 1: INK 5: B
RIGHT 1: CLS: PLOT 11,11: FOR n
=0 TO 7: READ a,b: DRAW a,0: DRA
W 0,b: NEXT n
80 FOR n=1 TO 10: PRINT INK 7:
AT RND*14+3,RND*22+5;"x": NEXT n
90 INK 7: PRINT AT y,x;" "
100 IF INKEY$=CHR$ 56 THEN LET
x=x+1: LET s=s+1
110 IF INKEY$=CHR$ 53 THEN LET
x=x-1: LET s=s+1
120 IF INKEY$=CHR$ 55 THEN LET
y=y-1: LET s=s+1
130 IF INKEY$=CHR$ 54 THEN LET
y=y+1: LET s=s+1
140 IF CODE SCREEN$ (y,x)=0 THE
N GO TO 400
150 IF SCREEN$ (y,x)="x" THEN L
ET s=s+10-INT (s/15): BEEP .05,1
2: LET m=INT (m/1.2)
160 PRINT AT y,x;"o"
170 LET b=INT (RND*4)
180 IF b=8 THEN PRINT AT y,x;"■"
": GO TO 400
190 LET a=4*(b/4-INT (b/4))
200 LET u=x+(a=2)-(a=1): LET v=
y+(a=0)-(a=3): IF CODE SCREEN$ (
v,u)<>32 THEN LET b=b+1: GO TO 1
80
```

```
210 PRINT AT v,u: INK 6;"■"
220 LET r=r+1: BEEP .01,0
230 PRINT AT 21,2:"SCORE ";s;"
HIGH ";h;" x=";10-INT (r/1
5);" ": FOR n=0 TO m: NEXT n: GO
TO 90
400 FOR n=0 TO 40: PRINT AT y,x
: OVER 1;"o": BEEP .04,24: NEXT
n: PRINT AT y,x: OVER 1;"a"
410 FOR n=0 TO 200
420 LET x=13: PAPER 2: INK 7: I
F y>11 THEN LET x=5
430 PRINT AT x+2,9: FLASH 1;"
GAME OVER "
440 IF h<s THEN PRINT AT x+6,24
;" ";AT x+4,3;" Sie haben den
High-Score";AT x+6,3;" um ";s-h
;" Punkte verbessert": LET h=s
460 PAUSE 1: PAUSE 0: GO TO 60
600 DATA 234,154,-1,-154,-1,154
,-232,-154,1,154,1,-153,229,152,
-228,-152
700 SAVE "Hinterhalt" LINE 5
800 DATA 0,126,126,126,126,126,
126,0,0,126,66,66,66,126,0,0,
0,60,36,36,60,0,0
```

## Sprachausgabe auf dem ZX Spectrum



Mit diesem Programm wandelt der ZX Spectrum Töne, Sprache und Sonstiges, die vom Cassettenrekorder in den Computer eingespielt werden, um und gibt diese dann auf Tastendruck über seinen eingebauten Lautsprecher wieder. Natürlich benötigt dieses Programm sehr viel Speicherplatz, somit läuft es auch nur auf einem Spectrum mit 48 K. Die Länge der Eingabe ist auf 5 bis 6 Sekunden beschränkt. Sie haben also nur wenige Sekunden Zeit, Ihre Information in eine synthetische Stimme des Spectrums umwandeln zu lassen.

```
2 CLEAR 32767
5 GO SUB 1000
10 PRINT AT 6,9;"Sprachausgabe"
20 PRINT AT 12,10;"1= Einlesen"
30 PRINT AT 14,10;"2= Ausgabe"
40 INPUT ein
50 IF ein=1 THEN GO TO 100
60 IF ein=2 THEN RANDOMIZE USR
65308
70 IF ein<>1 OR ein<>2 THEN GO
TO 40
100 PRINT AT 21,0;"Rekorder ein
schalten und eine Taste Druec
ken"
110 PAUSE 0
115 PRINT FLASH 1;"Aufnahme";
FLASH 0
120 RANDOMIZE USR 65280
130 RUN 10
1000 RESTORE
1010 FOR i=65280 TO 65339
1020 READ a
1030 POKE i,a
1040 NEXT i
1050 RETURN
```

```
2000 DATA 243,33,0,128,6,8,219,2
54,203,119,32,2,203,254,203,62,1
6,244,203,14,35,124,254,254,32,2
34,251,201
2010 DATA 243,33,0,128,6,8,203,7
0,40,4,62,0,211,254,62,255,211,2
54,203,6,16,240,203,6,35,124,254
,254,32,230,251,201
```

Addr Hex OP OPERAND NOTES

### EINLESEN

FF00 F3	DI	
FF01 210080	LD	HL,8000
FF04 0608	LD	B,08
FF06 DBFE	IN	A,(FE)
FF08 CB77	BIT	6,A
FF0A 2002	JR	NZ,+2FF0E
FF0C CBFE	SET	7,(HL)
FF0E CB3E	SRL	(HL)
FF10 10F4	DJNZ	-12FF06
FF12 CB0E	RRC	(HL)
FF14 23	INC	HL
FF15 7C	LD	A,H
FF16 FEFE	CP	FE
FF18 20EA	JR	NZ,-22FF04
FF1A FB	EI	
FF1B C9	RET	

### AUSGABE

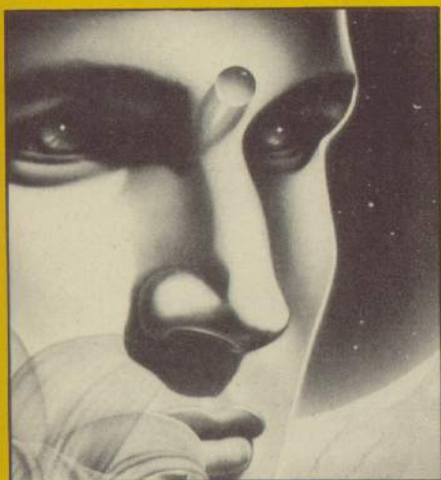
FF1C F3	DI	
FF1D 210080	LD	HL,8000
FF20 0608	LD	B,08
FF22 CB46	BIT	0,(HL)
FF24 2804	JR	Z,+4FF2A
FF26 3E00	LD	A,00
FF28 D3FE	OUT	(FE),A
FF2A 3EFF	LD	A,FF
FF2C D3FE	OUT	(FE),A
FF2E CB06	RLC	(HL)
FF30 10F0	DJNZ	-16FF22
FF32 CB06	RLC	(HL)
FF34 23	INC	HL
FF35 7C	LD	A,H
FF36 FEFE	CP	FE
FF38 20E6	JR	NZ,-26FF20
FF3A FB	EI	
FF3B C9	RET	



## Picasso

Mit diesem Programm werden Sie einige Meisterwerke auf Ihrem Bildschirm zaubern. Die Richtung bestimmen Sie mit den Tasten »5«—»8«. Sie ändern den Typ, wenn Sie »H« drücken und dann den neuen Typ eingeben.

Sie können die Farbe ändern, indem Sie »C« und dann die neue Farbe eintippen. Wollen Sie es etwas blitzen lassen, drücken Sie »F«. Soll das aufhören, tippen Sie »N«.

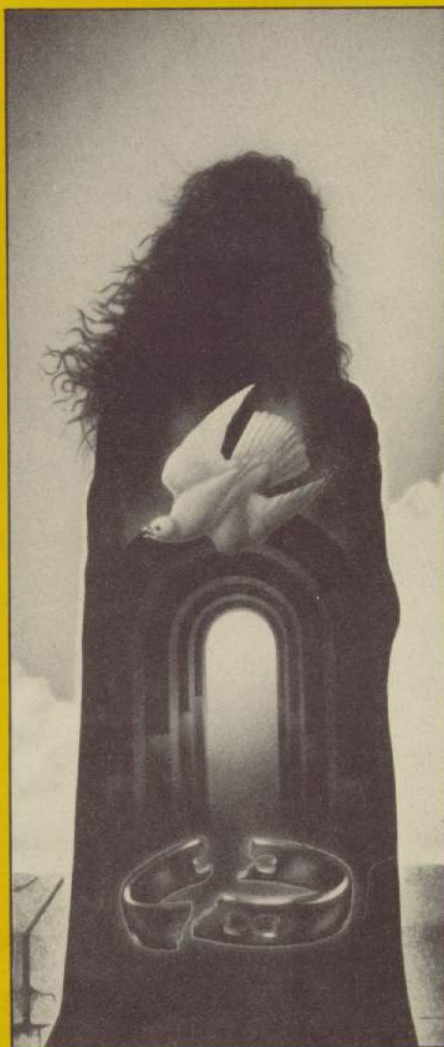


```

10 REM PICASSO
30 PRINT INK 1;TAB 6;"Picasso"
40 PRINT " INK RND*5;"Bewegen
  Sie sich mit den Cursor Tasten"
:
50 PRINT " INK RND*5;"Wenn Si
  e eine neue Farbe wollen"
60 PRINT " INK RND*5;"tippen
  Sie "; FLASH 1;"C"; FLASH 0;" un
  d die neue Farbe"
70 PRINT " INK RND*5;"Wenn Si
  e einen neuen Charakter "
80 PRINT INK RND*5;"wollen, ti
  ppen Sie "; FLASH 1;"H"; FLASH 0
  ;" und den Charakter"
85 PRINT " INK RND*5;"Wenn es
  FLASH'en soll tippen Sie"
86 PRINT INK RND*5; FLASH 1;"F
  "; FLASH 0;" (FLASH'en zurueck m
  it "; FLASH 1;"N"; FLASH 0;"")
90 PRINT " INK 1; FLASH 1;"PR
  ESS ANY KEY TO CONTINUE"
100 PAUSE 4E4:CLS
110 LET X=10: LET Y=X: LET B=0
120 LET C=2: LET H$=""
130 PRINT AT X,Y: INK C; FLASH
  B;H$
140 LET Y=Y+(INKEY$="8")-(INKEY
  $="5")
145 IF Y=-1 THEN LET Y=0
146 IF Y=32 THEN LET Y=31
150 LET X=X+(INKEY$="6")-(INKEY
  $="7")
155 IF X=-1 THEN LET X=0
156 IF X=22 THEN LET Y=21
160 IF INKEY$="C" THEN GO SUB 2
00
170 IF INKEY$="H" THEN GO SUB 3
00
175 IF INKEY$="F" THEN LET B=1
177 IF INKEY$="N" THEN LET B=0
180 GO TO 130
200 INPUT "Type in new colour "
: C
210 RETURN
300 INPUT "Type in new characte
  r ";H$
310 RETURN
  
```

## Dezimal-, Hexadezimal- Wandlung für den ZX Spectrum

Die Zahl a « 65536 wird in a\$ mit 4 Cha-  
raktern konvertiert. Dieses kleine Pro-  
gramm schickte uns W. Golda, Erlangen.



```

10 INPUT "Dezimalzahl= ";a
20 CLS : PRINT "Dez ";a: PRINT

30 DIM a$(4)
40 LET t=4096: LET k=a
50 FOR z=1 TO 4
60 LET k=INT (a/t)
70 LET a=a-k*t
80 IF k>9 THEN LET k=k+7
90 LET a$(z)=CHR$ (k+48)
100 LET t=t/16
110 NEXT z
120 PRINT "Hex ";a$: GO TO 10
  
```

## Spektral Henker

(entnommen aus dem Buch SPASS & PROFIT &  
SPECTRUM, erschienen im Verlag Cooperation,  
München)



Ein relativ einfaches Spiel, bei dem der  
Computer ein Wort aus seinem Vokabu-  
lar wählt, das Sie in wenigen Versuchen  
(deren Zahl sich nach der Länge der  
Worte richtet) herausfinden müssen. Das  
Vokabular kann leicht verändert oder  
erweitert werden. SPEKTRAL HENKER  
wurde von Ken Mahogany geschrieben.

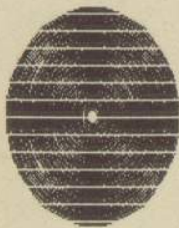
```

10 REM spectral hangman
20 REM © K Mahogany, 1982
30 FOR q=1 TO RND*25+1
40 READ a$
50 NEXT q
60 LET n=LEN a$
70 DIM b(n): DIM d(n)
80 FOR q=1 TO n
90 LET b(q)=CODE a$(q)
100 LET d(q)=b(q)
110 NEXT q
120 FOR j=1 TO n+1/3
140 GO SUB 410
150 IF h=n THEN GO TO 300
190 PRINT " INK RND*6;"Ihr "
: j: " Versuch"
200 INPUT c$
210 LET f=CODE c$
220 FOR q=1 TO n
230 IF d(q)=f THEN LET d(q)=0
240 NEXT q
260 NEXT j
265 GO SUB 410
270 PRINT " INK 3;"Schade, die
  Zeit ist um."
280 GO TO 330
300 REM **** gewonnen ****
310 PRINT " INK 4;TAB 4;"Gut ge
  macht"
320 PRINT " INK 4;TAB 4;"Sie ha
  ben das Wort in ",TAB 4;j-1;" Ve
  rsuchen erraten"
330 PRINT " INK 2;"Es hiess ";
  a$
335 PRINT " INK RND*5;"Fuer ne
  ues Spiel Taste druecken"
340 PAUSE 4E4
350 RUN
360 DATA "merkmal","spectrum","
  cambridge","zufall","melone"
370 DATA "frage","quiz","gesich
  t","onkel","rekorder"
380 DATA "basic","formel","freu
  ndlich","quelle","besser"
390 DATA "butter","erdbeere","d
  umm","atom","zauberer"
400 DATA "magier","verhext","ne
  id","mutwillig","trompete"
410 LET h=0
415 FOR e=1 TO n
420 IF b(e)=d(e) THEN PRINT INK
  RND*6;"-";
430 IF b(e)<>d(e) THEN PRINT IN
  K RND*6;CHR$ (b(e));: LET h=h+1:
  BEEP .25,h
440 NEXT e
450 BORDER RND*6
460 PRINT " INK 0; PAPER 6;"Sie
  haben "; INK RND*5;h: INK 0;" B
  uchstaben"
470 PRINT
480 RETURN
  
```

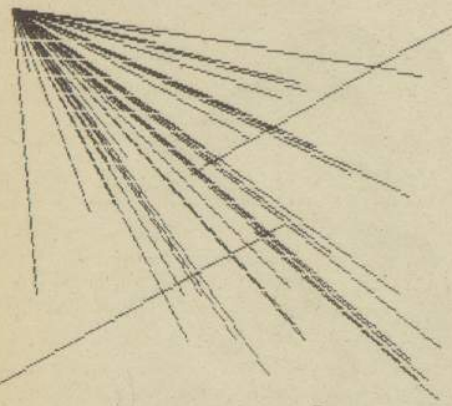


## Plotten und Zeichnen

Die nächsten Programme zeigen die Möglichkeiten des Spectrum, zu plotten und zu zeichnen. Das erste Programm zeichnet eine Reihe von Kreisen und benutzt den Kreis-Befehl, um einen Kegel zu konstruieren.

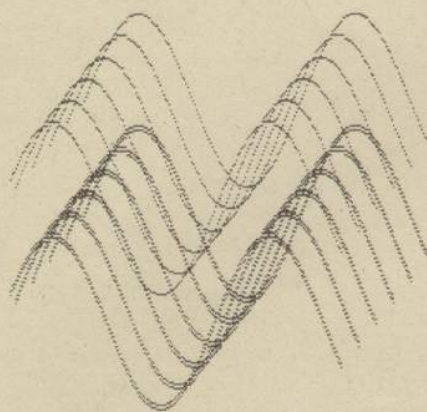


```
1 REM circle
10 INPUT "x-Koordinate ? ":x
11 INPUT "y-Koordinate ? ":y
12 INPUT "Start ( erster radiu
s ) ? ":s
13 INPUT "Ende ( letzter radiu
s ) ? ":f
14 INPUT "Colour ":c
40 FOR z=s TO f
50 CIRCLE INK c;x,y,z
60 NEXT z
70 GO TO 10
```



```
1 REM graph1
10 PLOT 0,0: DRAW INK RND*6;RN
D*255,RND*175
20 PLOT 355,175: DRAW INK RND*
6,-RND*355,-RND*175
30 PLOT 255,0: DRAW INK RND*5,
-RND*255,RND*175
40 PLOT 0,175: DRAW INK RND*6,
RND*255,-RND*175
50 GO TO 40
60 GO TO 10
```

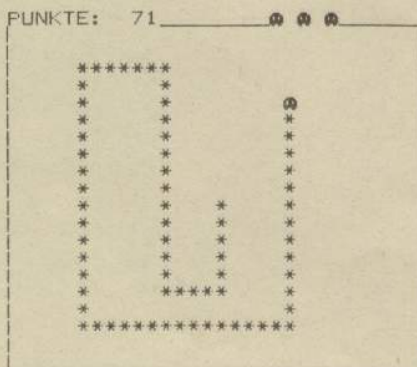
```
1 REM PLOTTEN UND ZEICHNEN
10 INPUT "Farbe? ":c
20 BORDER 7: PAPER 7: INK c: C
LS
30 INPUT "Geben Sie die Breite
ein 1-25 (1=fein,25=brei
t) ":b
40 LET a=0
50 PLOT 0,175: DRAW 0,-175
60 PLOT 0,0: DRAW 175,0
70 FOR t=175 TO 0 STEP -b
80 PLOT 0,t: DRAW a,-t
90 LET a=a+b
100 NEXT t
```



```
1 REM graph3
10 LET x=RND*9: PAPER x: INK 9
: BORDER x: CLS
15 FOR d=1 TO 2
20 FOR a=0 TO 50 STEP 10
30 FOR x=0 TO 10 STEP .05
40 IF d=1 THEN PLOT 20*x+a,40*
SIN x+40+a
50 IF d=2 THEN PLOT OVER 1;20*
x+a,40*SIN x+40+a
60 NEXT x: NEXT a: NEXT d
```

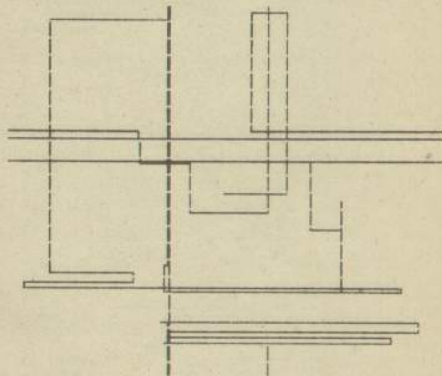
## Trailer

TRAILER basiert auf dem Arcade-Spiel SCHLANGE. Hier müssen Sie Ihr Männchen über den Bildschirm bringen, ohne daß es seine eigene Spur kreuzt. Sie haben drei Leben, und Ihr Ergebnis und Ihre übrigen Männlein werden an der obersten Zeile gezeigt. Sie können zwischen den Geschwindigkeiten 0—10 wählen. 0 ist schnell, 10 langsam. Ihre Richtung steuern Sie durch die Tasten »5«, »6«, »7«, »8«.



```
1 REM Trailer von D Harwood
5 FOR t=1 TO 7: READ n
6 POKE USR "k"+t,n: NEXT t
7 DATA 60,126,219,219,255,255
,102,0
8 REM um * zu bekommen, tippe
n Sie "k" im Graphic-Mode ein
9 LET p$="* * *"
10 LET s=0: LET c=3
11 BORDER 7: PAPER 7: INK 0: C
LS
```

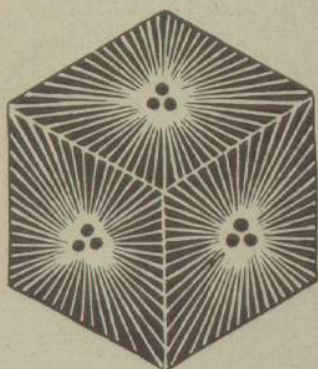
```
12 REM Instruktionen
15 PRINT TAB 8; INK 1;"T ": P
AUSE 10: PRINT INK 2;"r ": PAUS
E 10: PRINT INK 3;"a ": PAUSE 1
0
20 PRINT INK 4;"i ": PAUSE 10
: PRINT INK 1;"1 ": PAUSE 10: P
RINT INK 2;"e ": PAUSE 10: PRIN
T INK 3;"r "
30 PRINT "Ein Spiel fuer Exp
erten. Ihr "
40 PRINT "Mann jagt ueber den
Bildschirm und "
50 PRINT "Sie sollen verhinder
n, dass er seine "
60 PRINT "eigene Spur kreuzt.
Siehaben drei Leben"
65 INPUT "Tempo (0-10) ",(0 s
chnell 10 langsam) ":sp
70 PRINT INK 2; FLASH 1;AT 10,
10;"Viel Spass"
80 PAUSE 300: CLS
85 CLS : PLOT 248,168: DRAW -2
41,0: DRAW 0,-161: DRAW 241,0: D
RAW 0,160
90 LET a=11: LET b=16: LET a$=
STR$ (INT (RND*4)+5):
95 PRINT AT 0,20;p$
100 PRINT AT a,b;" "
105 FOR z=1 TO sp*2: NEXT z
110 IF INKEY$<>" " THEN LET a$=I
NKEY$
115 PRINT AT a,b;"*"
120 LET a=a+(a$="6")-(a$="7")
130 LET b=b+(a$="8")-(a$="5")
140 IF a=21 OR a=0 THEN LET a=A
BS (a-20)
150 IF b=31 OR b=0 THEN LET b=A
BS (b-30)
160 BEEP .01,RND*20
170 LET s=s+1: PRINT AT 0,1;"PU
NKTE: ":s
180 IF SCREEN$ (a,b)="/" THEN G
O TO 200
190 GO TO 100
200 CLS : PRINT PAPER 1: INK 7:
FLASH 1;AT 11,0;"!!!!!!!!!!!!!!B
ANG!!!!!!!!!!!!!!"
210 LET c=c-1
220 IF c=0 THEN GO TO 270
230 LET p$="": FOR i=1 TO c: LE
T p$=p$+" " : NEXT i
240 PRINT "Nur noch ";c;" Leb
en"
250 PRINT "PASSEN SIE AUF "
260 GO TO 80
270 PRINT AT 13,8; FLASH 1;"SPI
EL BEENDET"
```



```
2 LET a$="": LET a=127: LET b
=87
10 IF INKEY$<>" " THEN LET a$=I
NKEY$
20 LET b=b+(a$="7")-(a$="6")
21 LET a=a+(a$="8")-(a$="5")
40 IF a=256 OR a=0 THEN LET a=
ABS (a-255)
41 IF b=176 OR b=0 THEN LET b=
ABS (b-175)
60 PLOT a,b: GO TO 10
```

(entnommen aus dem Buch SPASS & PROFIT & SPECTRUM, erschienen im Verlag Cooperation, München)





## READER 1

(entnommen aus dem Buch SPECTRUM SPEKTAKULÄR, erschienen im Verlag Cooperation, München)

Wir wissen, wo im Speicher des Spectrum ein Programm gespeichert ist, im Bereich zwischen PROG und VARS, und wir können die echten Adressen von PROG und VARS finden, indem wir die entsprechenden Systemvariablen ein-PEEKen. Wenn wir also alle Adressen zwischen PROG und VARS PEEKen, dann sollten wir in der Lage sein, unser Programm zu finden. READER 1 ist ein allgemeines Allzweck-»PEEK-Programm«, das, wenn ihm eine Start- und Endadresse gegeben wird (in Zeilen 10 und 15), alle Adressen dazwischen liest. Nun ist der Programmbereich ein logischer Ort, um mit PEEK zu lesen, weil wir zumindestens wissen, was wir hier zu finden erwarten. Sie würden doch nach allem, was Sie bisher erfahren haben, erwarten, daß das Programm in genau der gleichen Weise gespeichert ist, wie es eingegeben wurde. (Oder etwa nicht?)

Starten Sie das Programm und schauen Sie, was passiert. (Alle Programme in diesem Kapitel arbeiten mit sich selbst, außer RENUMBER, oder mit irgend einem anderen Programm, das gleichzeitig mit ihnen im Speicher ist. Wenn es Ihnen also langweilig wird, READER 1 zu »lesen«, können Sie ein anderes Programm mit MERGE dazuladen und sich auch durch dieses lesen. Achtung, daß keine Zeilen miteinander kollidieren.)

Die Ausgabe des Programms erfolgt in drei Spalten: Spalte 1 listet jede Adresse zwischen PROG und VARS auf, Spalte 2 gibt den »gelesenen« Inhalt dieser Adressen wieder, und Spalte 3 gibt das Zeichen an, das durch die gePEEKten Werte repräsentiert wird.

Für jeden anderen Bereich des Speichers wäre Spalte 3 nicht sehr informativ, aber was den Programmbereich betrifft, so sollte in Spalte 3 das Programm genauso erscheinen wie es eingegeben wurde.

Natürlich geschieht das nicht. Aber ob-

wohl es einige geheimnisvolle Zeichen und ein oder zwei »nicht abdruckbare« Dinge gibt (das sind keine schmutzigen Wörter, sondern Farb-Kontrollzeichen, die eine Fehlermeldung zur Folge hätten, wenn man sie zu drucken versuchte), so bleibt doch genug Gleiches zwischen Spalte 3 und Ihrem Originalprogramm, daß es klar zu erkennen ist. Einiges davon wurde genau in der gleichen Weise gespeichert wie Sie es eingetippt haben, aber andere Dinge wurden auf recht seltsame Art gespeichert. Versuchen Sie herauszufinden, was passiert ist, und erklären Sie die seltsamen Zeichen, bevor Sie READER 2 eingeben.

```

10 LET P=PEEK 23635+256*PEEK 2
3636
15 LET V=PEEK 23627+256*PEEK 2
3628
20 LET C=PEEK P
25 PRINT P;TAB 8; IF C<15 OR
C>23 THEN PRINT TAB 16;CHR$ C:G
O TO 35
30 PRINT TAB 16;"(undruckbar)"
35 LET P=P+1: IF P=V THEN STOP
40 GO TO 20
9000 SAVE "READER 1" LINE 0

```

## READER 2

Das ist eine verbesserte Version des vorhergehenden Programms, bei der Ihnen Spalte 3 das genaue Programmlisting wiedergibt. (Es können immer noch einige nicht abdruckbare Zeichen erscheinen, falls Sie Farb-Kontrollzeichen benutzen, um Ihr Programm in prächtigen Farben aufzulisten.)

»Reader 3«, falls Sie das für sich selbst schreiben wollen, könnte dies ebenfalls erklären.

Wie Sie bei READER 1 gesehen haben, besteht der Hauptunterschied zwischen einem Programm im Speicher und einem aufgelisteten Programm in der Art, wie Zahlen gespeichert werden. Der Spectrum unterscheidet zwischen Zeilennummern und allen anderen Zeichen, und speichert sie auf völlig verschiedene Art.

### ZEILENNUMMERN

Zeilennummern (d.h. Zahlen, die vor jeder Programmzeile stehen. Zeilennummern, die einem GOTO oder GOSUB folgen, werden auf die gleiche Weise wie andere Zahlen gespeichert).

Zeilennummern werden als zwei Bytes große Hexadezimalzahlen gespeichert. Zahlen, die weniger als 2 Bytes benötigen, d.h. Zahlen unter 256 dezimal, werden mit einer führenden Null gespeichert.

Deshalb besteht die erste Änderung von READER 2 darin, daß, wann immer es sich um eine Zeilennummer handelt, 2 Bytes aus dem Speicher in eine einfache Dezimalzahl umgewandelt werden (Zeilen 20 und 25).

Die beiden Bytes, die der Zeilennummer folgen, sind KEIN Teil des Programms. Sie werden durch den Computer eingefügt, um die Länge der Programmzeile (in Bytes) anzuzeigen, und dienen als ZEIGER auf die nächste Adresse, die als Zeilennummer behandelt wird.

Auch diese beiden Bytes stellen eine Hexadezimalzahl dar, aber sie sind in umgekehrter Reihenfolge gespeichert. Das »HIGH ORDER«-Byte (d.h. dasjenige, dessen Wert mit 256 multipliziert werden muß, wenn man es in eine Dezimalzahl umwandelt) ist hinter dem »LOW ORDER«-Byte abgespeichert. Das klingt vielleicht unlogisch, aber Sie werden feststellen, wenn Sie tiefer in das »Gedächtnis« des Spectrum eingedrungen sind, daß dies die normale Art bei einem Computer ist, 2 Byte große Zahlen zu speichern; und in der Tat sind es die Zeilennummern, die seltsam abgespeichert werden.

### ANDERE ZAHLEN

Alle anderen Zahlen werden im Programmbereich **zweimal** gespeichert. Zuerst werden sie einfach als Zeichen gespeichert (d.h. 1.2 wird als CHR\$ 49; CHR\$ 46; CHR\$ 50 gespeichert). Dann wird ein CHR\$ 14 dahintergespeichert, um anzuzeigen, daß diese Zeichen tatsächlich eine Zahl bilden, und dann wird der Wert der Zahl in den nächsten fünf Bytes gespeichert, indem entweder Ganzzahl- oder Fließkommaformat verwendet wird. Wir werden den Unterschied zwischen den beiden Formaten im Programm »VARIABLE« sehen, aber für den Augenblick genügt es zu sagen, daß, wo auch immer das Zeichen CHR\$ 14 im Programmbereich auftaucht, wir und der Computer wissen, daß die nächsten fünf Bytes eine Zahl repräsentieren. (Siehe Zeile 50.)

```

10 LET P=PEEK 23635+256*PEEK 2
3636
15 LET V=PEEK 23627+256*PEEK 2
3628
20 LET N=256*PEEK P+PEEK (P+1)
25 GO SUB 100: PRINT N;TAB 21;
"( ZEILEN": LET P=P+1: GO SUB 10
0: PRINT TAB 21;"( NUMBER": LET
P=P+1
30 LET L=PEEK P+256*PEEK (P+1)
35 GO SUB 100: PRINT L;TAB 21
;"( ZEILEN": LET P=P+1: GO SUB 1
00: PRINT TAB 21;"( LAENGE": LET
P=P+1
40 LET L=P+L
45 LET C=PEEK P
50 GO SUB 100: IF C<14 THEN RE
STORE: PRINT TAB 21;"ZAHL ":F
OR J=1 TO 5: LET P=P+1: GO SUB 1
00: READ N$: PRINT TAB 21;N$: NE
XT J: GO TO 65
55 IF C<15 OR C>23 THEN PRINT
TAB 16;CHR$ C: GO TO 65
60 PRINT TAB 16;"(undruckbar)"
65 LET P=P+1: IF P=V THEN STOP
70 IF P=L THEN GO TO 20
75 GO TO 45
100 PRINT P;TAB 8;PEEK P;TAB 16
: RETURN
200 DATA "FUENF","BYTES","WERT"
,"DER","ZAHL"
9000 SAVE "READER 2" LINE 0

```



## Sortieren mit dem ZX Spectrum

Wer viel mit Namensdateien arbeitet, wird mitunter den alphabetisch geordneten Ausdruck der gespeicherten Namen wünschen. Zum Beispiel zur Herstellung von Mitglieder-

Telefonverzeichnissen oder dergleichen. Wer seine Software selbst schreibt, muß dann in sein Programm eine Routine für die sogenannte Alpha-Sortierung einbauen. Es sind mehrere Sortierv Verfahren be-

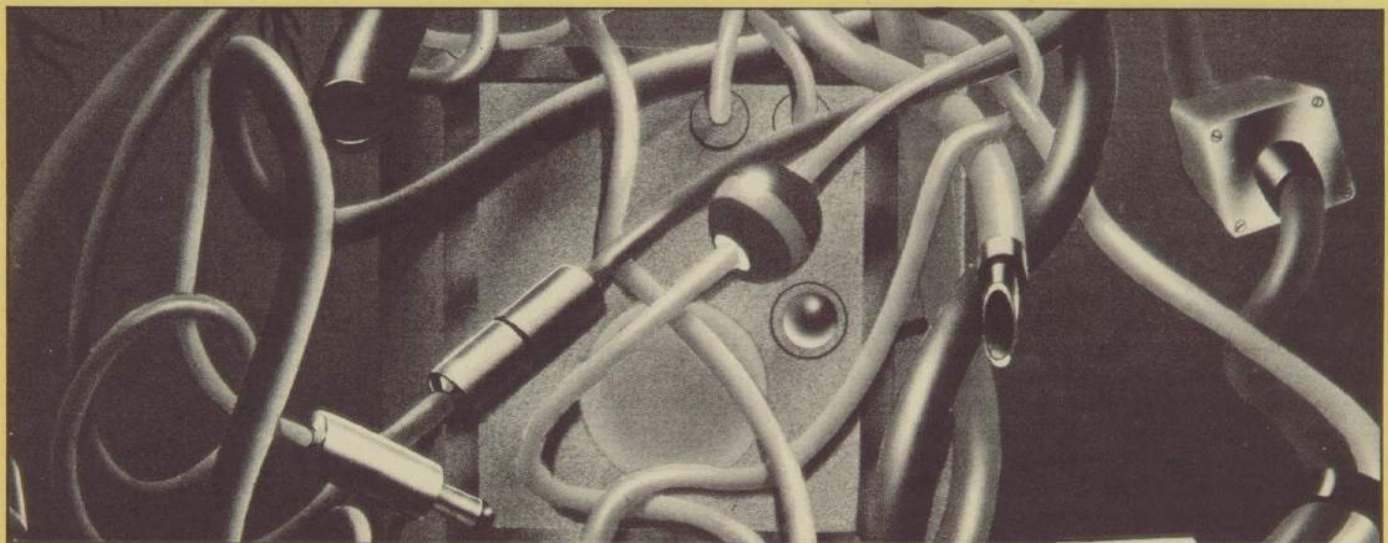
kannt, die man verwenden kann. Ein recht brauchbares, relativ schnelles Sortierv Verfahren, welches nur wenige Programmzeilen belegt, ist der Shell-Metzner-Sort.

```
10 REM "SORT DEMO"
20 DIM a$(21,10)
40 LET c=0
100 CLS
110 PRINT AT 0,0;"Eingabe:";TAB
11;"SORT I:";TAB 22;"SORT II:"
120 INPUT b$
121 LET c=c+1
125 IF CODE b$=32 THEN GO TO 49
130 LET a$(c)=b$
170 PRINT AT c+1,0;a$(c)
190 GO TO 120
499 LET c=c-1
500 REM shell-metzner sort einf
510 LET m=c
520 LET m=INT (m/2)
530 IF m=0 THEN GO TO 670
540 LET j=1
550 LET k=c-m
560 LET i=j
570 LET l=i+m
580 IF a$(i)<=a$(l) THEN GO TO
590 LET h=a$(i)
600 LET a$(i)=a$(l)
610 LET a$(l)=h$
620 LET i=i-m
630 IF i<1 THEN GO TO 650
640 GO TO 570
650 LET j=j+1
```

```
660 IF j>k THEN GO TO 520
665 GO TO 560
670 REM ausdruck sort einf.
680 FOR e=1 TO c
690 PRINT AT e+1,11;a$(e)
700 NEXT e
900 GO SUB 1000
910 FOR e=1 TO c
920 PRINT AT e+1,22;a$(e)
930 NEXT e
940 STOP
950 FOR e=2 TO 21
960 PRINT AT e,11;"
"
970 NEXT e
980 GO TO 120
1000 REM shell-metzner sort erw.
1010 LET m=c
1020 LET m=INT (m/2)
1030 IF m=0 THEN RETURN
1040 LET j=1
1045 LET k=c-m
1050 LET i=j
1060 LET l=i+m
1061 LET r=a$(i)
1062 GO SUB 1150
1063 LET p=r$
1064 LET r=a$(l)
1065 GO SUB 1150
1066 LET q=r$
1070 IF p<=q$ THEN GO TO 1120
1080 LET h=a$(i)
```

```
1083 LET a$(i)=a$(l)
1086 LET a$(l)=h$
1090 LET i=i-m
1100 IF i<1 THEN GO TO 1120
1110 GO TO 1060
1120 LET j=j+1
1130 IF j>k THEN GO TO 1020
1140 GO TO 1050
1150 FOR s=1 TO 10
1160 IF CODE r$(s)>90 THEN LET r
$(s)=CHR$ (CODE r$(s)-32)
1170 NEXT s
1180 RETURN
1190 SAVE "SORT DEMO" LINE 10
```

Eingabe:	SORT I:	SORT II:
c	A	A
d	B	B
w	D	c
e	E	D
r	F	d
A	O	e
E	c	E
D	d	F
F	e	f
B	f	O
O	o	o
z	p	p
f	r	r
o	w	w
p	z	z





# **P**ROGRAMME

**ZX-SPECTRUM**

Dieses Sortiervorgang ist zum Beispiel in Floegel »Programmieren in Basic und Maschinencode mit dem ZX81«, Holzkirchen, 1982, auf den Seiten 56-58 beschrieben. Allerdings stecken auch dort zwei dieser kleinen Fehler drin, welche daran zweifeln lassen, ob das Programm so je gelaufen ist. Hat man nämlich eine Liste von Worten eingegeben und sortiert und will man dann noch weitere Worte eingeben, so wirft das Programm aus der ursprünglichen Liste ein Wort hinaus.

Das muß ja nicht so sein. Man sollte deshalb in dem Programm die Zeile 20 ändern und LET N=O und eine neue Zeile 105 LET N=N+1 einfügen. Außerdem sollte man nicht vergessen, den Fehler in Zeile 170 zu korrigieren, wo statt »kleiner als« besser »größer als« stehen sollte. Nur mit dieser Änderung nähert sich nämlich das Bildschirm-Layout der Vorstellung, die der Autor im Sinn gehabt haben dürfte. Mit den Korrekturen läuft dann das Sortierprogramm ganz erstklassig.

Gibt man es in den neuen ZX Spectrum ein, so läuft das Programm immer noch einwandfrei, solange man die Dateiworte nur in Großbuchstaben oder nur in Kleinbuchstaben eintippt. Aber das ist ja nicht der Sinn der Groß-Kleinschreibung, die der ZX Spectrum anbietet. Vielmehr will man doch Worte jetzt so eingeben, wie sie normalerweise geschrieben werden, etwa in der Form »VAL«, »Computer« und »sinclair«.

Wer so eingegebene Worte mit dem vorerwähnten Programm sortiert, der wird sich erst einmal wundern. Schließlich erwartet man ja die Reihenfolge »Computer«, »sinclair« und »VAL«.

Tatsächlich steht »sinclair« aber ganz am Schluß.

Der Grund dafür liegt darin, daß die mit dem »kleiner oder gleich«-Operator arbeitenden Sortiervorgänge die Sortierung nach den Dezimalcodes der zu sortierenden Buchstaben vornehmen. So hat beim ZX Spectrum »X« den Dezimalcode 88 und »Y« den Dezimalcode 89. Also kommt beim Sortieren »X« vor »Y«. So weit, so gut. Da nun aber jedes Zeichen des Zeichenvorrats einen ganz bestimmten Dezimalcode hat, liegt es auf der Hand, daß auch »Y« und »y« verschiedene Dezimalcodes haben müssen. Beim ZX Spectrum haben die Leute, welche das ROM entworfen haben, ihre Entscheidung dahin getroffen, daß alle kleinen Buchstaben einen um 32 höheren Dezimalcode als die ihnen entsprechenden großen Buchstaben

haben. Da »Y« den Dezimalcode 89 hat, ist dann für »y« der Dezimalcode 121. Ganz klar, daß dann beim Sortieren »y« erst nach »Y« kommt, und »sinclair« erst nach »VAL«.

Man sollte darin keinen Nachteil des Systems sehen. Man kommt, wenn man überhaupt Kleinbuchstaben haben will, nicht darum herum, sie von Großbuchstaben unterscheidbar zu machen, also für Klein- und Großbuchstaben verschiedene Dezimalcodes zu verwenden, obwohl die zwei unterschiedlichen Buchstaben eines Groß-Klein-Paares in der alphabetischen Folge beide nur einen einzigen, nämlich denselben Rangplatz einnehmen.

Was technisch nicht zu ändern ist, muß man akzeptieren und, soweit es bei bestimmten Aufgabenstellungen nachteilig ist, durch besondere Tricks überwinden. Bei der Alpha-Sortierung mit dem ZX Spectrum besteht der Trick darin, daß man sich entschließt, die eigentliche Sortierung grundsätzlich nur mit Großbuchstaben oder nur mit Kleinbuchstaben durchzuführen. Dazu bieten sich zwei Wege an.

Der eine Weg besteht darin, daß man parallel zur Namensdatei noch eine gleich große Hilfsdatei einrichtet, in der die Namen nur in Großbuchstaben oder nur in Kleinbuchstaben stehen. Die automatische Umsetzung in ausschließlich große oder kleine Buchstaben läßt sich leicht durch eine spezielle Programmroutine einrichten, entweder regelmäßig gleich bei der Eingabe der Namen in den Computer, oder jeweils zu Beginn einer Sortierung. Die Sortierung findet dann ziemlich schnell in der Hilfsdatei statt. Der Nachteil dieser Methode ist, daß die Hilfsdatei genau so viel Speicherplatz wie die Namensdatei braucht, daß also im Ergebnis nur halb so viele Namen gespeichert werden können, als es ohne die Hilfsdatei möglich wäre. Der andere Weg besteht darin, daß man fallweise jeweils dann, wenn im Sortierprogramm zwei Namen zum Vergleich anstehen, diese beiden Namen einheitlich in Großbuchstaben oder in Kleinbuchstaben transformiert. Auch diese Umsetzung läßt sich durch eine spezielle Programmroutine leicht einrichten. Der Nachteil dieser Methode ist, daß schon deshalb, weil für jeden Vergleich zwei Namen in die einheitliche Schreibweise transformiert werden müssen, die Anzahl der Umsetzungen erheblich größer als beim vorher beschriebenen Weg ist, bei dem jeder Name nur einmal transformiert werden muß. Die Sortierung dauert deshalb auch erheblich länger. Dieser Nachteil dürfte aber für die meisten

Anwender geringer wiegen gegenüber dem Vorteil, daß erheblich mehr Namen gespeichert werden können. Für normale Anwenderprobleme erscheint deshalb der zweite Weg empfehlenswert.

Das hier abgedruckte Demonstrationsprogramm für den ZX Spectrum gibt unter »Eingabe:« die eingegebenen Namen aus, unter »SORT I:« die Sortierung, wie sie sich mit dem unveränderten Shell-Metzner-Sort ergibt, und unter »SORT II:« die Sortierung mit dem erweiterten Shell-Metzner-Sort, wonach alle Worte, unabhängig von ihrer Schreibweise mit großen oder kleinen Buchstaben, in der richtigen alphabetischen Reihenfolge erscheinen. Das Programm wird mit RUN und ENTER gestartet. Sodann werden die Namen eingegeben, jeweils mit Namenseingabe und ENTER. Ist die Eingabe beendet, so beginnt nach SPACE und ENTER die Sortierung. SORT I ist sehr schnell fertig, aber unbefriedigend, was die alphabetische Ordnung angeht. SORT II braucht einige Zeit, die man in Ruhe abwarten sollte. Dafür ist das Ergebnis dann voll zufriedenstellend. Wer nach solch einer Sortierung die Eingabe fortsetzen will, drückt CONT und ENTER. Wer nach einer Sortierung mit einer völlig neuen Eingabe beginnen will, drückt RUN und ENTER.

Wer diesen erweiterten Shell-Metzner-Sort in eigene Programme einbinden will, beachte dies: Wenn die Namensdatei in einem durch DIM A\$(100,10) reservierten Feld und der zuletzt eingegebene Name in A\$(c) steht, dann kann mit den Programmzeilen 499 bis 665 die Sortierung durchgeführt werden. Wird diese Sortieroutine aus dem selbst geschriebenen Hauptprogramm mit GOSUB aufgerufen, dann bleibt in Zeile 1030 die RETURN-Anweisung stehen. Wird jedoch die ganze Sortieroutine mitten in das selbst geschriebene Programm hinein geschrieben, dann muß in Zeile 1030 die RETURN-Anweisung durch eine GOTO-Anweisung auf die erste Zeile hinter der Sortieroutine ersetzt werden, wie dies zur Demonstration beim Programmteil für SORT I in Zeile .... gezeigt ist.

Dipl.-Ing. Rudolf W. Ihmig



ZX-SPECTRUM 16K

# Mondlandung



```

speed 145 m/s      height 1680 m
fuel 6280 t         power 900 kp
    
```

Geben Sie folgendes Programm ein und starten Sie es mit RUN. Anschließend geben Sie bitte die MC-Werte ein. Zum Abspeichern drücken Sie die Break-Taste und anschließend GOTO 4010. Das Programm ist selbsterklärend. Die unterstrichenen Buchstaben bitte im Graphikmodus eingeben.

Alexander Stenger, Nürnberg.

```

5 GO SUB 2000
7 IF PEEK 32300<>112 THEN GO
SUB 3000
10 BORDER 2: PAPER 6: INK 9: C
LS : PRINT BRIGHT 1;AT 10,7;" L
UNAR LANDER "
20 FOR n=32300 TO 32331: BEEP
.1,PEEK n-100: NEXT n: BEEP .2,1
2
30 LET y=0: CLS : PRINT AT 4,2
;"Kennun Sie die Spielregeln?";
AT 6,8;"(J)A / (N)EIN"
40 IF INKEY$="j" OR INKEY$="J"
THEN GO TO 60
45 IF INKEY$<>"n" AND INKEY$<>
"N" THEN GO TO 40
50 PRINT AT 8,1;"Versuchen Sie
,moglichst sanuft";AT 10,1;"auf
dem Mond zu landen. Mit";AT 12,1
;"den Tasten 1-9 konnen Sie die"
55 PRINT AT 14,1;"Schubkraft e
installen, mit 0";AT 16,1;"schal
ten Sie Ihre Triebwerke";AT 18,1
;"aus. Drumcken Sie irgendeine";A
T 20,1;"Taste, um anzufangen!";
PAUSE 1: PAUSE 0
60 LET c=700: LET f=7000: LET
a=0: LET e=0
70 BORDER 1: PAPER 7: INK 0: B
RIGHT 0: CLS
    
```

```

80 PRINT AT 17,0;"GJHLG HLFHEE
EELKLGIKEFJ JIK EL"
90 PRINT AT 6,11;"AC";AT 7,11;
"BD"
100 POKE 23296,235: POKE 23297,
71: INVERSE 1: BRIGHT 1
110 PRINT AT 20,0;"speed
height fuel
power 000 kp "
120 FOR n=23232 TO 23295: POKE
n,0: NEXT n
130 LET m=0: LET p=0: LET s=200
: LET h=3990
140 IF f>0 THEN LET a=CODE INKE
Y$: IF a<58 AND a>47 THEN LET p=
a-48
150 LET s=s+(6000-h)/c: IF h<60
0 THEN LET s=s+(600-h)/100
160 LET f=f-p*2: LET s=s-p*2
170 LET i=h-s/12
180 IF f<0 THEN LET f=0: LET p
=0
190 LET m=m+i-h
200 IF m>52 THEN RANDOMIZE USR
32440: LET m=m-52: GO TO 200
210 IF m<-52 THEN RANDOMIZE USR
32400: LET m=m+52: GO TO 210
220 PRINT AT 20,6;INT (s+3);" m
/s ";AT 20,23;INT h+30;" m ";A
T 21,6;f;" t ";AT 21,23;p
240 LET h=i
250 IF h<=-29 THEN GO TO 300
260 IF h>6200 THEN GO TO 1300
290 GO TO 140
300 LET s=s+2.5: PRINT AT 20,23
;"0 m ";INVERSE 0: BRIGHT 0: B
ORDER 5
305 IF s>100 THEN GO TO 800
310 IF s>30 THEN GO TO 700
320 PRINT AT 5,1;"Landegeschwin
digkeit: ";INT (s*100)/100;" m/s
    
```

```

330 PRINT AT 7,1;"Restlicher Tr
eibstoff: ";f;" t"
340 IF s>15 THEN GO TO 600
350 IF s>6 THEN GO TO 500
410 RESTORE 420: FOR n=0 TO 7:
READ a: BEEP .1,a: NEXT n
420 DATA 12,12,16,12,16,19,19,1
6
430 LET a$="+ + landung gelunge
n + +": LET b=540: GO SUB 1000
440 PRINT AT 10,12;"SUPER !"
450 LET c=c-100
460 LET f=f+2000: PRINT AT 7,24
;f
470 LET a$="2000 t Treibstoffbo
nus": LET b=270: GO SUB 1000
480 PRINT AT 12,2;"Sie durfen a
uchmal absturzen.";AT 14,2;"Prob
ieren es nochmal....."
490 PAUSE 1: PAUSE 200: LET e=I
NT (e+2000/s): GO TO 70
500 LET m=30: GO SUB 900
510 PRINT AT 10,2;"Die Landung
hat Sie zwar ganz";AT 12,2;"schon
mitgenommen, aber Sie";AT 14,2
;"durfen es nochmal versuchen."
520 RESTORE 530: FOR n=0 TO 7:
READ a: BEEP .1,a: NEXT n
530 DATA 12,11,12,7,4,2,4,0
540 LET a$="+ + landung gelunge
n + +": LET b=270: GO SUB 1000
560 LET f=f+1500: PRINT AT 7,24
;f
570 LET a$="1500 t Treibstoffbo
nus": LET b=270: GO SUB 1000
590 LET e=INT (e+2000/s): GO TO
70
600 LET m=60: GO SUB 900
610 OVER 1: PLOT 93,44: DRAW 8,
-7: OVER 0
620 PRINT AT 7,1;" Sie leben zw
ar noch, aber ";AT 9,2;"auf de
m Mond. Und von da";AT 11,2;"kom
    
```



# PROGRAMME

```

men Sie ohne Raumschiff"; AT 13,2
;"leider nicht mehr weg."
630 FOR n=0 TO 2: BEEP .3,4: NE
XT n: BEEP 1,2,0
640 LET a$="Raumschiff zerstört
rt +": LET b=540: GO SUB 1000
690 GO TO 1100
700 BORDER 2: PRINT AT 15,11;"
": LET m=8: GO SUB 900: PRINT 0
VER 1; AT 16,11;"8/"; AT 16,11;"--
"
710 POKE 23297,80: FOR n=0 TO 3
: RANDOMIZE USR 32400: NEXT n
720 PRINT AT 5,6;"Geschwindigke
it beim"; AT 7,6;"Aufprall : "; IN
T (s*10)/10;" m/s"; AT 9,6;"Von I
nnen sind nur"; AT 11,6;"noch TrM
mmer Mbrig !"
730 LET a$="landung missglMck
t -": LET b=0: GO SUB 1000
740 PLOT 72,38: DRAW 0,30: PLOT
62,58: DRAW 20,0
750 DATA -5,-7,-9,-10
760 RESTORE 750: FOR n=0 TO 3:
READ a: BEEP .6,a: NEXT n: BEEP
1,8,-12
770 GO TO 1100
800 BORDER 3: PRINT AT 15,11;"
": LET m=8: GO SUB 900: PRINT 0
VER 1; AT 16,11;"8/"; AT 16,11;"mm
"
810 POKE 23297,80: FOR n=0 TO 1
2: RANDOMIZE USR 32400: NEXT n:
PAUSE 5
820 PRINT AT 17,10;"L K"; AT 18
,11;"HF"
830 PRINT AT 5,6;"Geschwindigke
it beim"; AT 7,6;"Aufprall : "; IN
T (s*10)/10;" m/s"; AT 9,6;"Sie h
aben einen neuen"; AT 11,6;"Krate
r geschaffen !"
840 LET a$="landung missglMck
t -": LET b=270: GO SUB 1000
850 PLOT 72,38: DRAW 0,30: PLOT
62,58: DRAW 20,0
860 RESTORE 750: FOR n=0 TO 3:
READ a: BEEP .6,a: NEXT n: BEEP
1,8,-12
870 GO TO 1100
900 FOR n=0 TO m: RANDOMIZE USR
32440: RANDOMIZE USR 32400: NEX
T n: RETURN
1000 FOR n=1 TO LEN a$: PRINT AT
2,31;a$(n)
1010 FOR m=0 TO 7: PAUSE 1: RAND
OMIZE USR 32500
1020 NEXT m: NEXT n
1030 FOR n=0 TO b: PAUSE 1: RAND
OMIZE USR 32500: NEXT n
1040 FOR n=0 TO 255: PRINT AT 2,
0;" ": RANDOMIZE USR 32500: NEXT
n
1050 RETURN
1100 PAUSE 300: LET a=INT (2000/
s)+e
1110 IF s<30 THEN LET a=a+INT (f
/40)
1120 PAPER 5: BORDER 1: CLS
1130 IF a>y THEN PRINT AT 8,4;"S
ie haben den High Score"; AT 10,4
;"um "; a-y;" Punkte Mberboten !"
: GO TO 1160
1140 PRINT AT 8,4;"Sie haben den
High Score"; AT 10,4;"nicht verb
essert. Den"; AT 12,4;"hat nDmlic
h immer noch"; AT 14,4;b$;" mit "
;y;" Punkten": GO SUB 1200
1150 GO TO 60
1160 PRINT AT 12,4;"Zur Belohnun
g dMrfen Sie"; AT 14,4;"jetzt Ihr
en Namen einge-"; AT 16,4;"ben ."
1170 GO SUB 1200
1180 BEEP .05,16: LET y=a: BEEP
.1,16: INPUT b$: BEEP .1,4
1190 GO TO 60
1200 LET a$="* * "+STR$ a+" Punk
te * *": LET b=320: GO SUB 1000
1210 RETURN
1300 PRINT AT 9,0: INVERSE 0: BR
IGHT 0;" So kommen Sie nie zum
Mond ! Damit sowas nicht noch

```

```

mal passiert, werde ich Si
e ein wenig anschieben...100
0 m/s"
1310 LET s=1000: PRINT AT 20,6;"
1000 m/s h": FOR n=0 TO 20: RAN
DOMIZE USR 32400: NEXT n: LET h=
h-1040
1320 PAUSE 1: PAUSE 500
1330 PRINT AT 9,0;" FOR n=0 TO 1
5: PRINT BRIGHT 0;" "": N
EXT n: GO TO 140
2000 RESTORE 2100
2010 FOR i=1 TO 14
2020 READ p$
2030 FOR g=0 TO 7
2040 READ a
2045 POKE USR p$+g,a
2050 NEXT g
2060 NEXT i
2070 RETURN
2100 DATA "a",0,0,1,1,3,15,63,25
1
2110 DATA "b",227,239,127,63,19,
36,64,224
2120 DATA "c",0,0,128,128,192,24
0,252,223
2130 DATA "d",199,247,254,252,20
0,36,2,7
2140 DATA "e",0,0,0,0,0,255,255,
255
2150 DATA "f",1,3,7,31,63,255,25
5,255
2160 DATA "g",0,0,1,3,15,63,255,
255
2170 DATA "h",128,192,224,248,25
2,255,255,255
2180 DATA "i",0,0,128,192,240,25
2,255,255
2190 DATA "j",0,0,129,195,255,25
5,255,255
2200 DATA "k",0,0,0,0,0,3,31,255
2210 DATA "l",0,0,0,0,0,192,248,
255
2220 DATA "m",68,0,68,68,68,68,5
6,0
2230 DATA "n",68,0,56,68,68,68,5
6,0
2240 DATA "o",36,0,56,4,60,68,60
,0
3000 CLEAR 32299
3010 PRINT "Eingabe des Maschine
ncodes in Dezimalwerten."
3020 FOR i=32300 TO 32592
3030 INPUT a: PRINT i,a: POKE i,
a: NEXT i
3040 RETURN
4000 CLEAR 32299: LOAD ""CODE :
RUN
4010 SAVE "Lander" LINE 4000: SA
VE "Land.mc."CODE 32300,292
32300 112
32301 115
32302 119
32303 115
32304 100
32305 115
32306 119
32307 115
32308 107
32309 110
32310 114
32311 110
32312 95
32313 110
32314 114
32315 110
32316 105
32317 108
32318 112
32319 108
32320 93

```

```

32321 108
32322 112
32323 108
32324 103
32325 112
32326 114
32327 112
32328 114
32329 115
32330 115
32331 114
32332 0
32333 0
32334 208
32335 87
32336 243
32337 13
32338 206
32339 11
32340 228
32341 80
32342 206
32343 11
32344 229
32345 80
32346 28
32347 23
32348 220
32349 10
32350 206
32351 11
32352 235
32353 80
32354 22
32355 23
32356 220
32357 10
32358 215
32359 24
32360 177
32361 51
32362 37
32363 109
32364 5
32365 0
32366 219
32367 2
32368 77
32369 0
32370 47
32371 85
32372 209
32373 0
32374 219
32375 2
32376 77
32377 0
32378 0
32379 0
32380 33
32381 24
32382 192
32383 87
32384 113
32385 14
32386 243
32387 13
32388 33
--

```

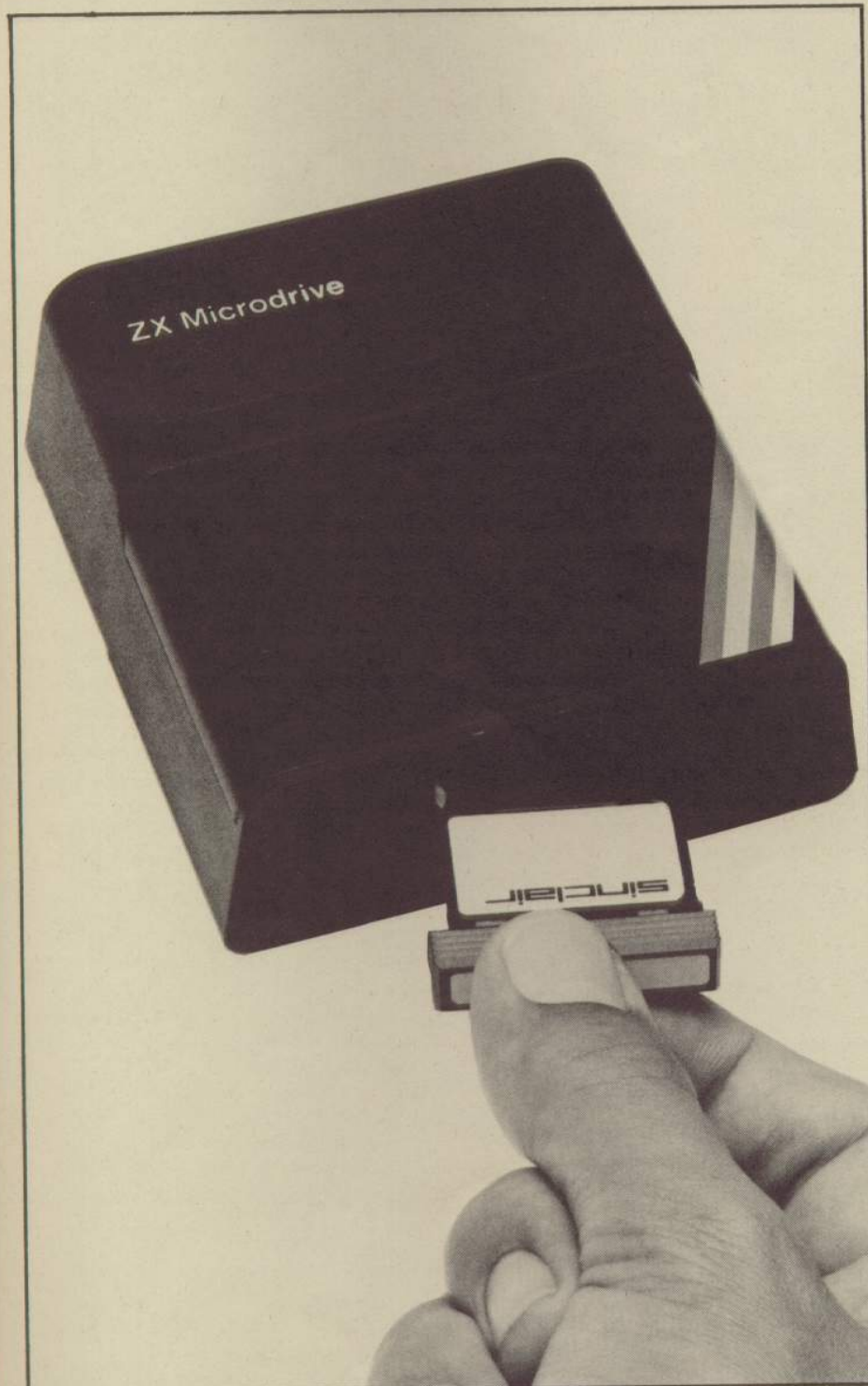


# PROGRAMME

32389	23	32457	249	32525	2
32390	198	32458	14	32526	219
32391	30	32459	2	32527	2
32392	43	32460	229	32528	77
32393	126	32461	6	32529	0
32394	118	32462	17	32530	210
32395	27	32463	84	32531	86
32396	3	32464	93	32532	0
32397	19	32465	205	32533	0
32398	0	32466	63	32534	174
32399	62	32467	127	32535	62
32400	42	32468	126	32536	243
32401	0	32469	18	32537	13
32402	91	32470	16	32538	206
32403	205	32471	247	32539	11
32404	63	32472	225	32540	228
32405	127	32473	35	32541	80
32406	34	32474	13	32542	206
32407	0	32475	32	32543	11
32408	91	32476	239	32544	229
32409	14	32477	201	32545	80
32410	2	32478	0	32546	28
32411	229	32479	0	32547	23
32412	6	32480	33	32548	220
32413	17	32481	0	32549	10
32414	84	32482	88	32550	62
32415	93	32483	14	32551	7
32416	197	32484	3	32552	164
32417	205	32485	6	32553	32
32418	38	32486	0	32554	15
32419	127	32487	54	32555	1
32420	193	32488	112	32556	224
32421	126	32489	35	32557	7
32422	18	32490	16	32558	9
32423	16	32491	251	32559	62
32424	245	32492	13	32560	224
32425	225	32493	32	32561	165
32426	35	32494	246	32562	254
32427	13	32495	201	32563	224
32428	32	32496	0	32564	32
32429	237	32497	0	32565	4
32430	201	32498	0	32566	1
32431	0	32499	0	32567	0
32432	0	32500	33	32568	249
32433	0	32501	64	32569	9
32434	0	32502	64	32570	37
32435	0	32503	17	32571	201
32436	0	32504	31	32572	0
32437	0	32505	0	32573	0
32438	0	32506	14	32574	0
32439	0	32507	8	32575	36
32440	42	32508	229	32576	62
32441	0	32509	126	32577	7
32442	91	32510	25	32578	164
32443	205	32511	23	32579	192
32444	38	32512	6	32580	62
32445	127	32513	32	32581	32
32446	34	32514	203	32582	133
32447	0	32515	22	32583	111
32448	91	32516	43	32584	62
32449	6	32517	16	32585	224
32450	15	32518	251	32586	165
32451	197	32519	225	32587	200
32452	205	32520	36	32588	62
32453	38	32521	13	32589	248
32454	127	32522	32	32590	132
32455	193	32523	240	32591	103
32456	16	32524	201	32592	201



## SPECTRUM MICRODRIVE



Der neue Sinclair ZX Microdrive erschließt erstmals den Anwendern zu Hause, im Unternehmen und in Schulen die Vorteile der für schnellen Datenzugriff ausgelegten Massenspeicherung zum besonders günstigen Preis. Er bietet eine Speicherkapazität von rund 85 K-Byte.

Mit dem Microdrive ist es Sinclair wieder einmal gelungen, eine Alternative zu teuren Diskettenlaufwerken und zum langsamen Cassetteninterface zu schaffen.

Massenspeicher waren von jeher das Sorgenkind im Bereich der Mikrocomputer, die in der Preisklasse unter 1000,— DM angesiedelt sind. Bislang gab es keine brauchbare Alternative zu den Cassettenrekordern, außer Diskettenlaufwerke. Da letztere die Systemkosten aber beträchtlich über die 1000-DM-Schwelle treiben, kann man sie aber nicht als echte Alternative zum Cassettenrekorder betrachten.

Das ZX-Microdrive von Sinclair ist das fehlende Bindeglied zwischen den beiden herkömmlichen Speichertechniken. Es bietet entschieden kürzere Zugriffszeiten und wesentlich mehr Funktionen für die Dateiverwaltung als das umständliche Tonband. Dabei bewegt sich der Preis im Rahmen des Computers, für den das Laufwerk konzipiert wurde.

### Das Laufwerk

Das Microdrive selbst besitzt in etwa die Größe einer geballten Faust. Die eigentliche Überraschung aber ist das »Interface 1«. Es befindet sich zu seinem größten Teil unter, zu einem kleinen Teil hinter dem Spectrum. Zum einen verbindet das Interface das Microdrive über ein kurzes Flachbandkabel mit dem Computer, zum anderen ermöglicht es den Aufbau eines Computer-Verbundnetzes und den Datentransfer von einem Rechner zum anderen. Durch Schrauben, mit denen das Interface mit dem Gehäuse des Spectrum fest verbunden werden kann, hat Sinclair Vorsorge gegen wackelige, unzuverlässige Aufbauten getroffen. Die gefürchteten »Abstürze« des Systems sind somit nicht zu erwarten!

Auch die Microdrives lassen sich mechanisch miteinander verbinden. Es kann eine Kette aus bis zu 8 Laufwerken gebildet werden; jedes Laufwerk kann jedoch nur einzeln benutzt werden, und muß durch eine längere Befehlsfolge aufgerufen werden.

Die Laufwerke und das Interface besitzen nur ein geringes Gewicht. Es ist leicht vorstellbar, daß dafür ein passender Tragkoffer auf den Markt kommen wird. Bei den Möglichkeiten, die das Interface 1 bietet, kann man mit einer wahren Flut an interaktiven Spielen und anderer Software



rechnen. Einem unterhaltsamen Abend mit dem Spectrum-System steht somit nichts mehr im Wege.

## Die Kassette

Das Microdrive verwendet als Speichermedium eine Kassette, die ein 2 mm breites Magnetband in Form einer Endloschleife enthält. Die Kassette hat etwa die Größe eines ROM-Speicher-IC's und faßt ca. 5 m Tonband, das den Tonkopf mit einer Geschwindigkeit von etwa 75 cm/s passiert. Messungen ergaben für die Umlaufzeit einen Wert von ca. 10 Sekunden. Sinclair gibt als Zugriffs-/Übertragungszeit für einen 48-kB-Datenbereich 3,5 Sekunden an. Das setzt voraus, daß der Beginn der Aufzeichnung bei Beginn der Übertragung bereits richtig positioniert ist. Doch selbst im ungünstigsten Fall, wenn die Daten erst nach einem Umlauf des Bandes geladen werden können, stellt eine Wartezeit von 10 Sekunden eine erhebliche Verbesserung gegenüber der normalen Compactcassette dar.

Die Kassetten für 100-kB werden unformatiert geliefert. Für die Formatierung sind etwa 30 Sekunden erforderlich. Um den Vorspann und die Prüfsumme zu schreiben und beides mehrmals zu überprüfen, muß das Band einige Male umlaufen. Dabei werden mangelhafte Bandstellen markiert und anschließend blockiert. Laut Sinclair besitzt keine Kassette weniger als 85 kB verfügbaren Speicherplatz — der tatsächliche Wert wird nach Aufruf von »cat« = catalogue angezeigt. Unsere Testgeräte hatten alle über 90 K-Byte!

Einmal formatiert, kann jede Kassette bis zu 50 Files (Datenfelder) aufnehmen. Am einfachsten lassen sich diese Files in katalogisierter Form auf einer Kassette speichern. Dazu sind nur ein paar Eingaben erforderlich und die Wartezeit beläuft sich auf ca. 7—10 s.

Das um die Microdrives erweiterte Betriebssystem stellt für den Benutzer eine Reihe von Kanälen und Strömen dar. Einfach gesagt, sind Kanäle und Ströme ein Grundkonzept für den Datentransfer innerhalb des Systems.

Es gibt Kanäle für den Bildschirm, für den ZX-Printer, für die Microdrive-Dateien, andere Spectrum-Computer (mit dem Interface 1) und über die RS 232-Schnittstelle zu anderen Peripheriegeräten. Das alles sind Ausgangskanäle.

Die Tastatur, Dateien vom Microdrive, andere Spectrum-Computer, sowie die RS 232-Schnittstelle erfordern dementsprechend Eingangskanäle.



**Im Bild der Bestseller von Sinclair, der Personalcomputer ZX Spectrum — bisher wurden weltweit mehr als 700.000 Stückzahlen verkauft — zusammen mit den neuen externen Massenspeichern ZX Microdrive und der Schnittstelle ZX Interface. Der außerordentlich kompakte Abmessungen aufweisende Massenspeicher ZX Microdrive verfügt über eine Kapazität von rund 85 K-Byte auf Magnetkassette. Die Schnittstelle ZX Interface 1 ist Steuereinheit für den Microdrivespeicher und verfügt zusätzlich über eine RS-232-Schnittstelle. Damit kann ein Spectrum-Personalcomputer auch mit anderen Computern und peripheren Geräten verbunden sowie an lokale Netzwerke mit bis zu 64 Teilnehmersystemen angeschlossen werden.**

Die »Ströme« regeln den Verlauf des Datenverkehrs; es gibt insgesamt 16, von 0—15. Die ersten 4 sind bestimmten Kanälen zugeordnet, doch die übrigen 12 können durch gewisse Befehle zu mehr Flexibilität beim Programmieren verhel-

tenübermittlung einzelne Geräte nicht ein- oder ausgeschaltet werden dürfen.

Um sich einem Netz anzuschließen, müssen Sie Ihrem Computer eine Kenn-Nummer geben, sind es nur 2 Computer, so kann jeder die Ziffer 1 ohne zusätzliche Kennung erhalten. Das Senden und Empfangen von Daten ist mit recht einfachen Befehlsfolgen zu bewerkstelligen. Ein Programm wird in den Computer geladen und an den Bestimmungsort via »SAVE« übertragen.

Die Übertragung eines Programmes oder von Daten erfordert die sorgfältige Koordination der beteiligten Partner. Sofern die Distanz die Rufweite überschreitet, müssen die Teilnehmer über Telefonverbindung verfügen, um zum Erfolg zu kommen. Beispielsweise besteht keine Möglichkeit, in ein laufendes Programm eines anderen Computers einzugreifen, um eine bevorstehende Datenübertragung anzukündigen.

Um ein Programm zu senden, lautet die Eingabe:

SAVE \* »n«;2

Der Empfänger muß sein Gerät mit folgenden Anweisungen für die Daten vorbereiten:

FORMAT »n«;2

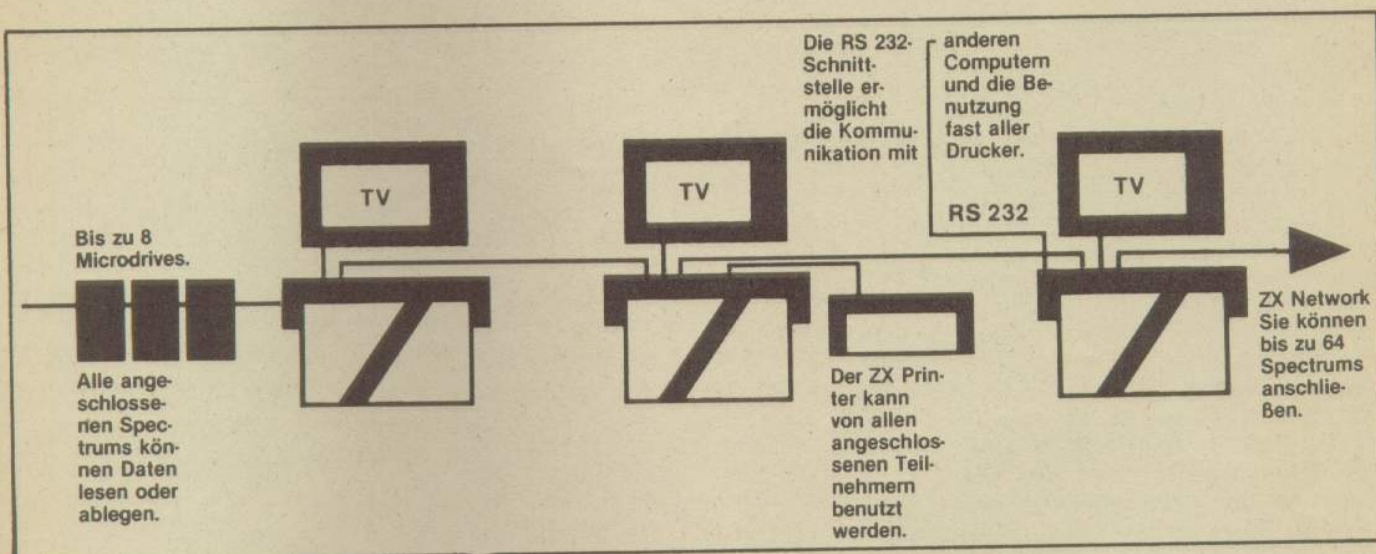
## Computernetze

Die größte Überraschung bei der Vorstellung des Microdrives war die Möglichkeit, Computer in einem örtlichen Netz zu einem Verbund zusammenzuschließen. Es lassen sich bis zu 64 Spectrum-Rechner im Umkreis von 100 m verbinden. Im Gegensatz zu manchen anderen einfachen Verbundsystemen werden dabei keine zusätzlichen Komponenten (Priority Controller) für die Interruptbehandlung benötigt.

Mit jedem Interface 1 wird ein Verbindungskabel zum Anschluß an ein Verbundnetz geliefert. Um miteinander in Verbindung zu treten, müssen lt. Sinclair nicht sämtliche Computer im Verbund eingeschaltet sein. Die einzige Einschränkung besteht darin, daß im Laufe der Da-

# ZX-USER CLUB





**LOAD \* »n«;1**

Danach muß der Empfänger die einwandfreie Übertragung mit einem anderen Befehl überprüfen.

Es besteht auch die Möglichkeit, mehrere Geräte von einem zentralen Spectrum aus mit Daten zu versorgen, beispielsweise in einer Schulklasse wäre dies anwendbar. Die Krönung der Interface-Eigenschaften ist aber die RS 232-Steckverbindung. Damit stehen dem Anwender eine Reihe von Möglichkeiten offen, vom Anschluß eines Matrixdruckers bis zum Telefonmodem. Der Anschluß eines Druckers ist eine naheliegende Sache und bald werden Textverarbeitungsprogramme den Nutzen daraus ziehen.

für kommerzielle Anwendungen und für Spiele, nicht nur für selbstgeschriebene Programme, nutzen. Dabei ergaben sich bisher Probleme, wenn die Programme vor dem Auflisten geschützt waren, und damit eine Überspielung auf ein anderes Speichermedium nicht möglich war. Mit der Unterstützung durch die Firma Sinclair werden die Software-Hersteller sicher eine Reihe von derart geschützten Programmen auf Microdrive-Kassetten herausgeben.

Noch bedeutsamer ist vielleicht die Tatsache, daß mit dem Microdrive der Anwendungsbereich des Spectrums erweitert wird, so daß benutzerfreundliche Programme von bislang ungeahnter Qualität möglich werden.

Bestimmt die Baudrate des Interface. (x kann 50, 110, 300, 600, 1200, 2400, 4800, 9600 oder 19200 sein)

**LOAD\*»m«;1;»Name«**

Lädt eine Datei unter dem angegebenen Namen vom Microdrive 1

**SAVE\*»M«;1;»NAME«**

Speichert eine Datei unter dem Namen auf Microdrive 1

Die Steuerbefehle für das Microdrive und den Betrieb im Verbundsystem sind im Befehlssatz des Spectrum bereits enthalten. Wie aus dem vorhergehenden Auszug aus dem Befehlssatz hervorgeht, kann dies die Befehlsfolge für ganz einfache Operationen u.U. komplizieren. Beispielsweise erfordert es zwischen 11 und 21 Tasteneingaben für das häufig notwendige Laden einer Datei. Die Mehrzahl der Eingaben muß außerdem mit wechselnden Shift-Befehlen erfolgen.

Als Sinclair den Spectrum im April 1982 vorstellte, gab es viel Rummel um das zu erwartende billige Diskettenlaufwerk. Wie sich nun herausstellt, dauerte es doch geraume Zeit, bis es erhältlich ist, und daneben arbeitet es auch nicht mit Disketten. Es verwendet als Speichermedium ein Band und bietet damit nur seriellen Zugriff auf Daten anstelle des wahlfreien Speicherzugriffs, den jedermann erwartet hatte. Es ist aber sicherlich nicht Clive Sinclairs Schuld, wenn nach dem ZX 81 und dem Spectrum viele Unmögliches von ihm erwarteten: Laufwerke mit wahlfreiem Zugriff für weniger als 400,— DM. Eine weitere Vermutung, daß das Betriebssystem des Spectrum bereits die CAT, CLOSE, DELETE, ERASE, FORMAT, MOVE und OPEN-Befehle bearbeiten würde, trog. Vielmehr enthält das Microdrive-Interface das Betriebssystem

## Zusammenfassung

Das Microdrive erschließt keine neue technische Dimension — derartige »Stringy Floppies« gibt es bereits seit geraumer Zeit. Seine Bedeutsamkeit erhält es vielmehr dadurch, daß Sinclair hinter diesem Produkt steht.

Um wirklich sinnvoll zu sein, muß ein Speichermedium mit der entsprechenden Software versehen werden. Dazu muß der externe Speicher von den Softwarehäusern als ein möglicher »Standard« angesehen werden, so daß für die Programme genügend Nachfrage existiert, ehe die entsprechenden Programme geschrieben werden. Sobald die ganze Sache aber erst einmal in Fluß gekommen ist, wird es sehr schwer werden, die Microdrives durch etwas anderes zu ersetzen. Somit ist es sehr wahrscheinlich, daß die Microdrives etliche Jahre Sinclair-Produkte als Zusatzspeicher begleiten werden.

Der durchschnittliche Spectrum-Besitzer will den zusätzlichen Speicherraum auch

## Die Steuerbefehle

**CAT 1**

Listet alle Dateien auf einer Kassette des gewählten Microdrives (hier Nr. 1) auf.

**CAT #z;1**

Sendet den Katalog des Microdrives 1 (oder 2, 3, usw.) an Strom z

**CLOSE #1**

Trennt einen Kanal vom angegebenen Strom

**ERASE »m«;1;»Name«**

Löscht die Datei mit dem angegebenen Namen von der Kassette in Microdrive 1

**FORMAT»m«;1;»Name«**

Bereitet eine leere Kassette vor und gibt ihr einen Namen

**FORMAT»n«;x**

Gibt der Station im Computer-Verbund die Nummer x

**FORMAT»t«;x**

# ZX-USER CLUB



und daher ist es für den Speicherausbau unerlässlich.

Die Firma Sinclair Research war von jeher etwas zurückhaltender (im nachhinein betrachtet), was die Angaben zu dem Microdrive angeht. Das lag mit Sicherheit auch daran, daß die Laufwerke während ihrer Entwicklung mehrere kleinere Abänderungen erforderten. Sinclair versprach, daß jedes Microdrive bis zu 100 kB auf einer einzelnen, auswechselbaren »Microfloppy« speichern könne, und daß alle Steuerwörter für das Laufwerk auf dem Spectrum zu finden seien.

Es blieb die Frage, was war mit »Microfloppy« gemeint? Es gab viele Spekulationen, oft wurde auch richtig geschlossen, daß ein echtes Floppy-Disk-Laufwerk zu diesem Preis nicht realisiert werden könne, daß es sich folglich um ein Bandlaufwerk handeln müsse.

Die eigentliche Überraschung ist daher nicht das Laufwerk selbst, sondern eher das Erweiterungsmodul »Interface 1«, die Schnittstelle zwischen Spectrum und Microdrive. Das Erweiterungsmodul bietet einen Ausgang für die Motorsteuerung eines Cassettenrekorders, einen Steckkontakt zum Anschluß an ein Verbundnetz, sowie eine RS 232-Schnittstelle.

Es besteht die Möglichkeit, ein Netz aus max. 64 Spectrum zu errichten, die mit Baudraten von 100 K-Baud kommunizieren können.

Der Erfahrungsbericht eines Sinclair-Mitarbeiters, der bereits mit dem Microdrive arbeiten konnte, besagt, daß nach anfänglichen Entwicklungsproblemen das Interface eine bedeutsame Erweiterung des Microdrives ist, wenn die Zuverlässigkeit auf lange Sicht gewährleistet wird. Vorstellbar ist beispielsweise ein kleinerer Gewerbebetrieb, der von Spectrum mit einem Textverarbeitungsprogramm oder erweiterten Betriebssystem in einer ROM-Kassette, wie sie von Sinclair beabsichtigt ist, einsetzt. In Verbindung mit zwei Microdrives als Massenspeicher, Verbindungen zu anderen Computern an weiteren Arbeitsplätzen und einem Matrix- oder vielleicht sogar Typenraddrucker ist somit Datenverwaltung in kleinerem Umfang möglich. Sinclair hat des weiteren die baldige Veröffentlichung von Software für die Microdrives angekündigt, zusätzlich zu den Programmen, die es bereits auf Cassette gibt, und die als ROM-Module erscheinen sollen. Handelsübliche Cassetten werden nach wie vor das preiswerteste Speichermedium bleiben, aber kleinere Programme lassen sich in Steckmodulen unterbringen, und umfangreiche Programme können vorteilhaft auf den Microdrives

mit ihrer Kapazität von ca. 90 kB gespeichert werden.

## Dokumentation

Sehr bald wurde bei der Arbeit mit dem Microdrive klar, daß die Betriebsanleitung zumindest unzureichend ist.

Sie besteht aus 57 Seiten, in denen alle Eigenschaften des Microdrives und auch des Interface mit abgehandelt werden. Die fehlenden oder nicht optimal erklärten Teile könnten bei der Übersetzung verbessert werden.

Ein grundlegendes Konzept, dessen man sich im Zusammenhang mit dem Interface 1 bemächtigen muß, sind die Kanäle und Ströme. Diesem Gedanken werden lediglich zwei Seiten gewidmet, obwohl er auf den folgenden Seiten unablässig verwendet wird. Die Kanäle sind die Plätze, zu welchen Sie und der Computer Daten übersenden — an den Bildschirm, den Drucker, das Microdrive, die RS 232-Schnittstelle oder andere Spectrum im Verbundnetz.

Die Ströme sind die Bahnen, über die der Informationsfluß zu den entsprechenden Kanälen erfolgt. Die ersten vier sind bestimmten Komponenten zugeordnet, den Bildschirmhälften, der Tastatur und dem Drucker. Damit stehen für den Datentransfer von und zum Microdrive und mit anderen Computern die Ströme 4—15 zur Verfügung.

Will man die Möglichkeit eines Verbundsystems ausnutzen, muß man zwischen Programmen und Daten klar unterscheiden. Während ein Programm in einem recht einfachen Vorgang übertragen werden kann, erfordert die Übergabe von Daten eine etwas umfangreichere Vorbereitung. Da das Verbundnetz mit einem 255 Byte langem Pufferspeicher operiert, erfordert der Abschluß der Datenübertragung einen »CLOSE STREAM«-Befehl, damit der Puffer geleert wird, selbst wenn er nicht vollständig gefüllt ist. Dasselbe Konzept liegt der Manipulation von Daten mit dem Microdrive zugrunde. Man muß dazu einen Strom definieren und die Datei mit ihrem Namen aufrufen. Die betreffende Kassette wird dann danach durchsucht, und, abhängig davon, ob eine Datei bereits existiert oder nicht, wird sie für den Lesevorgang vorbereitet, oder es wird eine entsprechende Datei angelegt.

Alles schön und gut; daraus ergeben sich noch keine unlösbaren Probleme. Doch das Kapitel »Daten und Computerverbund« ist beispielsweise lediglich 1 1/2 Seiten lang. Es gibt nur ein kurzes Demo-Programm, und die kurzgefaßte Erläute-

rung läßt auf Verfasser schließen, die aufgrund langer Beschäftigung mit Bits und Bytes vergessen haben, daß der normale Anwender nicht mit derselben Präzision Informationen speichern kann, wie das Microdrive selbst.

Oder kann er es doch? So sehr ich ein scheinbar umwälzend neues Produkt mit überschäumender Begeisterung begrüßen möchte, muß ich doch hinzufügen, daß ich an der Zuverlässigkeit gewisse Zweifel hege. Sicher muß man Sinclair Zeit zum Nachweis der Betriebssicherheit zugestehen. Solange noch nicht große Stückzahlen öffentlich im Umlauf sind, läßt sich über die Ausfallraten und die Begleitumstände, die zu Fehlern führen, nichts Genaueres sagen.

## Schnittstellen

Ein anderes Problem ergab sich aus der Verbindung der RS 232-Schnittstelle mit einem Drucker. Einfach gesagt: — es ging nicht. (Ja, ich bin nach dem Handbuch vorgegangen!) Ich bin sicher, daß es irgendwie geht, da ich es bei Sinclairs Präsentation gesehen habe. Aber wenn es ein Anwender selbst versucht, stößt er auf Probleme. Schließlich ist Sinclair kaum in der Lage, einer potentiellen Käuferschicht, die nach Hunderttausenden gezählt werden muß, detailliert auf Nachfrage zu antworten.

Das ist schade, denn die meisten Probleme resultieren wahrscheinlich aus dem zu kurzen Handbuch. Man möchte fast vermuten, daß es sich bei der Bedienungsanleitung um den Teil einer Verschwörung handelt, bei der es um die Verkaufsvorbereitung eines Buches »Wie man das Microdrive in den Griff bekommt« geht. Dieses Buch müßte allerdings erst noch geschrieben werden. Der Markt dafür und die Nachfrage ist schon da ...

## Peripherie

Für den Spectrum muß es mehr spezielle Peripheriegeräte geben, als für irgendeinen anderen Mikro-, Mini- oder E-n-platinencomputer auf dem Markt. Man kann den Speicherbereich erweitern, es kann Ton- und Sprachausgabe hinzugefügt werden, es gibt Grafikzusätze, Farbdrucker und externe Tastaturen.

Wollen Sie Ihr System um den Spectrum herum erweitern, brauchen Sie sich keine Sorgen zu machen. Sie haben nur die Qual der Wahl, Ihren Geldbeutel wird es nicht allzusehr belasten. Zusatzgeräte für den Spectrum gibt es zu ganz vernünftigen Preisen.



## Akustische Eingabekontrolle für den ZX 81

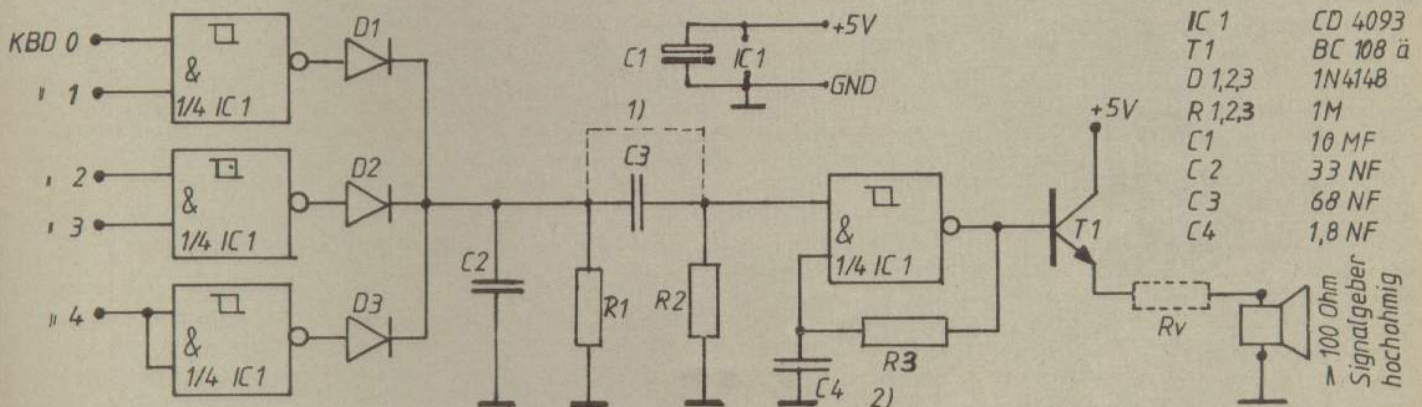
Die Folientastatur des ZX 81 erlaubt wegen des geringen Tastenhubs und des fehlenden Druckpunkts keine exakte Eingabekontrolle beim Eintasten. Die Richtigkeit der Eingabe kann erst am Bildschirm geprüft werden. Eine akustische Eingabekontrolle ist daher für den ZX 81 besonders zweckmäßig.

Diese Verbesserung ist bereits mit einer

sehr einfachen Schaltung realisierbar (siehe Anlage). Die erforderliche kleine Zusatzplatine läßt sich problemlos im ZX 81-Gehäuse unterbringen. Die Kosten für diese Zusatzanordnung liegen bei ca. 10,— DM.

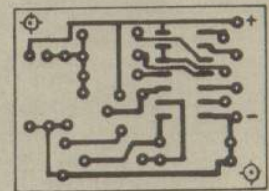
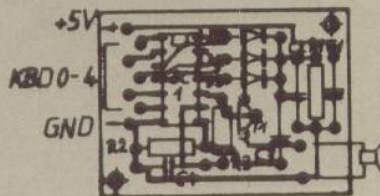
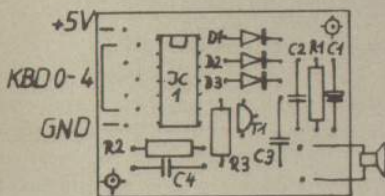
Ich hoffe, daß dieser Vorschlag für ZX 81-Benutzer von Nutzen ist.

M. Himmelstoß, Regensburg



1) durch Einlegen der Brücke ertönt das Signal solange die Taste gedrückt ist

2) die Tonhöhe ist durch R3, C4 veränderbar



## RESET-Taste

Nach genauem Überprüfen habe ich festgestellt, daß darin Programme für ZX 81 mit 1K-RAM-Ausführung enthalten sind und nicht nur für 16K-RAM.

Vermißt habe ich aber einfache Hardware-Erweiterungen. Daher möchte ich Ihnen eine Möglichkeit unterbreiten, mit einfachen Mitteln und preiswert die Speicherkapazität von 1K-RAM zu verdoppeln.

Auch wäre es oftmals sehr nützlich, eine Reset-Taste zu besitzen, was recht einfach zu verwirklichen ist.

Der Vorteil einer Reset-Taste: Nach eventuellen Abstürzen des ZX 81 muß nicht immer die Netzspannung abgeschaltet werden.

Anbei meine beiden Hardware-Änderungsvorschläge, mit denen mein eigener ZX 81 schon seit längerer Zeit ausgerüstet ist und störungsfrei läuft.

Zur Speichererweiterung von 1K-RAM-Grundausstattung auf 2K-RAM.

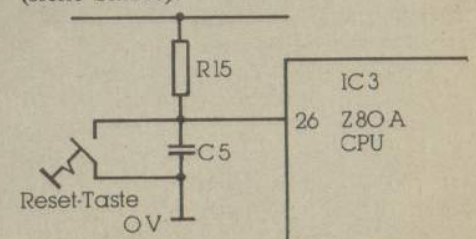
Auf der Leiterplatte des ZX 81 befinden sich IC-Anschlüsse für IC 4 oder alternative IC 4a und IC 4b. Zur Erweiterung auf 2K-RAM müssen lediglich die zwei RAM's IC 4a und IC 4b (uPD 2114 LC) oder wenn mit diesem bestückt, das IC 4 (MK 4118) aus ihren Sockeln entfernt werden.

In den Sockel des IC 4 (24 PIN) kann dann ein statisches RAM 2k x 8 Typ 6116 (Preis ca. DM 20,—) eingesetzt werden.

Außerdem muß dann die ebenfalls auf der Leiterplatte vorgesehene Drahtbrücke L2 eingelötet werden. Ist der beschriebene Umbau beendet, stehen 2K-RAM Speicherkapazität zur Verfügung.

Zur Reset-Taste:

Der Reset-Taster kann einfach mit zwei dünnen Leitungen an die Anschlüsse des Kondensators C5 auf der Leiterplatte des ZX 81 angeschlossen werden (siehe Skizze).



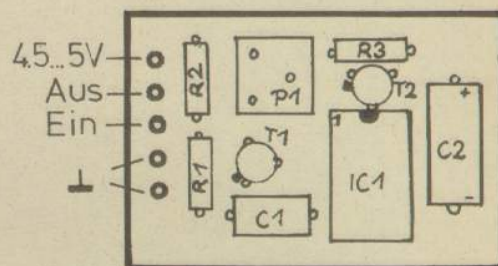
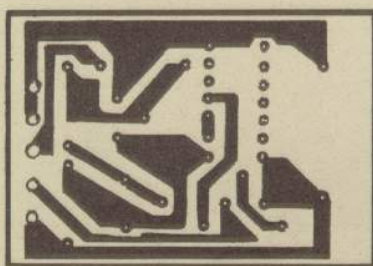
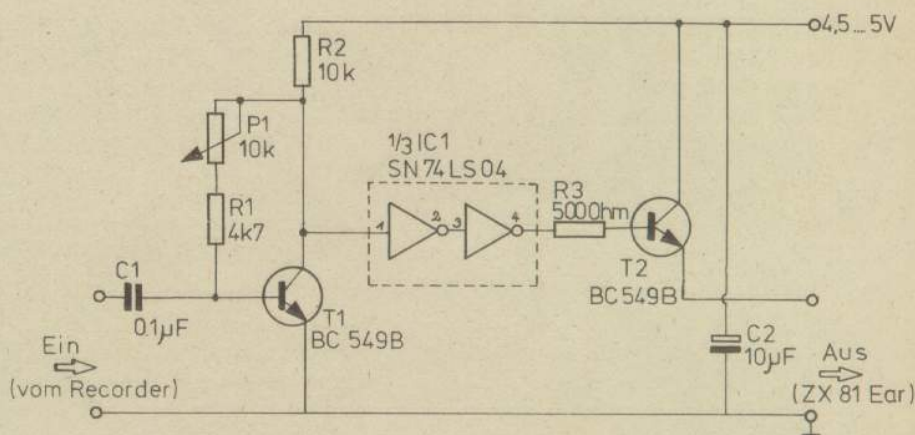


## Aufnahmeverstärker

Herr Christian Götz aus Wertingen schickte uns folgendes Schaltbild:

Dieser Aufnahmeverstärker ist besonders günstig für Recorder, die für den ZX81 zu leise sind. Die Schaltung ist äußerst stromsparend (4 mA ohne Signal vom Cassetten-Recorder, 13 mA mit Signal), so daß eine 4,5-V-Flachbatterie eine sehr lange Betriebsdauer garantiert! Außerdem ist die Verstärkung an P1 regelbar. Wichtig: 5 V ist die maximale Betriebsspannung! Der IC wird sonst zerstört! Am besten benutzen Sie eine 4,5-V-Flachbatterie oder ein stabilisiertes Netzgerät.

Sie können das Gerät auch betriebsfertig direkt bei Herrn Götz erhalten. Anfragen richten Sie bitte an Herrn Christian Götz, Gerh.-Hauptmann-Ring 4, 8857 Wertingen.



## Änderung des Super Moving Keyboard

Gerhard Sann, Garching.

Als Besitzer eines »Super Moving Keyboards« benutzte ich den rechten Zifferntastenblock praktisch nie. Dieser bot sich daher für die meistbenutzten Zweitfunktionen wie

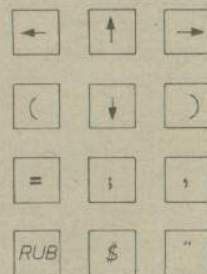
() = ; , ' \$ Rechts Links Auf Ab Rubout und Edit

an. Nach Auftrennen der Leiterbahnen (eine Leiterbahn — Taste 7 — ist durchgeschleift und muß nach Abtrennen der Taste überbrückt werden) werden die Tasten folgendermaßen beschaltet. Die Transistoren wurden direkt unter die Erstfunktions-Tasten gelötet. Die Dioden fanden unter dem Zifferntastenblock — nun Zweitfunktions-Tasten — ein Plätzchen. WICHTIG: Alle Dioden für den SHIFT-Transistor müssen Germanium-Dioden, alle für die Erstfunktions-Transistoren Silizium-Dioden sein!!

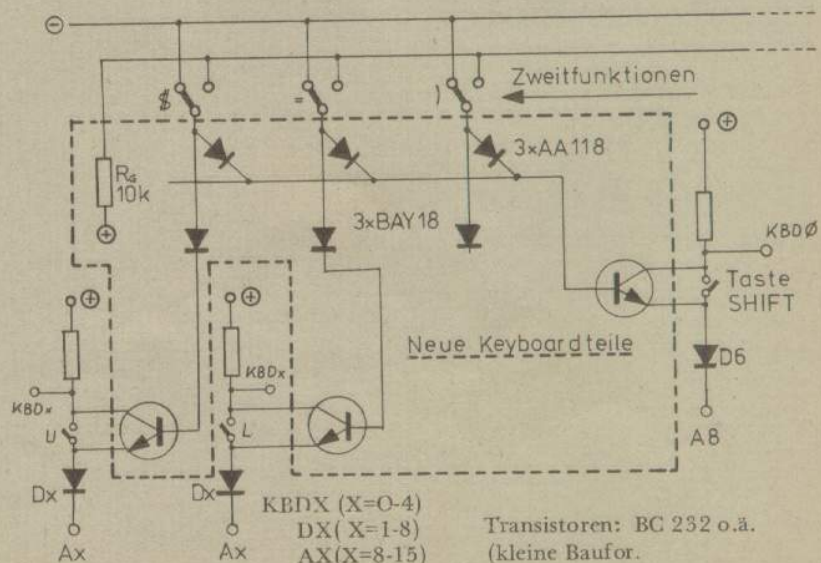
Falls andere als vorgegebene Transistoren verwendet werden, muß  $R_B$  neu dimensioniert werden. Funktion ; muß dabei gerade noch erkannt werden (einziger Fall, wo zwei Transistoren gleichzeitig leitend sind). Aus Bauteilersparnis wurde auf Basis-Entladewiderstände verzichtet. Der Leckstrom über die Dioden nach Minus reicht aus, um die Transistoren sicher zu sperren.

Die freie Taste (auf dem Keyboard links) wurde für EDIT ausgenutzt. Als letztes mußten die Beschriftungen der Tasten aktualisiert werden. Die um etwa fünf Millimeter dickere Tastatur paßt nach wie vor in das Gehäuse.

Das so geänderte USER-optimale Keyboard erfüllt seit geraumer Zeit seine Aufgaben störungs- und fehlerfrei.



Zweitfunktionen



Transistoren: BC 232 o.ä.  
(kleine Bauför.)



# SPECTRUM SOFTWARE

Alle Programme auf Kassette mit ausführlicher englischer Beschreibung.



Diese Programme nur spielbar bei sehr guten Englischkenntnissen!

Bestellen Sie im Innenteil auf Seite 6



# ALLE WEGE FÜHREN ZU SINCLAIR — ABER...

... der sicherste Weg, einen SINCLAIR ZX81 oder SINCLAIR ZX SPECTRUM zu kaufen und dann auch wirklich zu genießen ist der Weg zum Sinclair-Fachhändler, von denen es über 300 in der Bundesrepublik gibt. Nur da bekommen Sie SINCLAIR mit allem: fachkundige Beratung, Zubehör, Software und die Original-Garantie. Nur beim SINCLAIR Fachhändler sind Sie gut aufgehoben — garantiert!

Sinclair ZX Spectrum — der Heimcomputer, der das Wunderbare leistet: ein Kleinrechner, der es auf seine Weise mit einem richtigen Großrechner aufnehmen kann. Mathematische Funktionen und Operationen wie bei großen Profi-Geräten. 8-beliebig einsetzbare Farben für Vorder- und Hintergrund. BEEP-Kommandos über

drei Oktaven. Bedienungsfreundliche Tastatur mit 40 Tipptasten. Hochauflösende Grafik durch 256 Punkte waagrecht und 192 senkrecht und beliebiger Mischung von Grafik und Schrift. LOAD/SAVE in Hochgeschwindigkeit. Programmierbar in BASIC und Maschinensprache (Z80A). Völlig neu gestaltetes Kassetten-Interface.

Sie erhalten auch nur bei unseren autorisierten Fachhändlern die 2 unentbehrlichen deutschen Original-Handbücher zu jedem Gerät. Und Sie erhalten nicht nur einen Computer mit Zukunft, sondern auch eine Anlage mit fast unbegrenzten Ausbaumöglichkeiten. Der ZX SPECTRUM ist ein Gerät, das seinem Namen Ehre macht: das Spektrum von Spectrum reicht unendlich weit!

SINCLAIR ZX 81 — einer der erfolgreichsten Personal-Computer der Welt, jetzt für einen Bruchteil der Summe zu haben, die vergleichbare Computer kosten. Die Massenverbreitung dieses weltweit beliebten Gerätes (bisher über 1 Million Käufer) macht einen sensationellen Preis möglich, inklusive ein 212-seitiges Handbuch, Netzteil und alle Anschlüsse. Dabei wird am Gerät selbst an nichts gespart: Assembler über die USER-Taste. Eingebauter Syntax-Check mit Cursor. Keyboard mit 40 Tipptasten, für Grafik, Symbole und Zeichen. Wie den SINCLAIR ZX SPECTRUM gibt es den ZX81 mit der Original-Sinclair-Garantie nur von uns. Und direkt bei unseren autorisierten Fachhändlern.

**sinclair**



Wo Sie Ihren Original-Sinclair bekommen, sagt Ihnen unser Händlernachweis. Schreiben Sie uns, Sinclair-Generalvertretung Deutschland, Postfach 6352 8012 Ottobrunn.