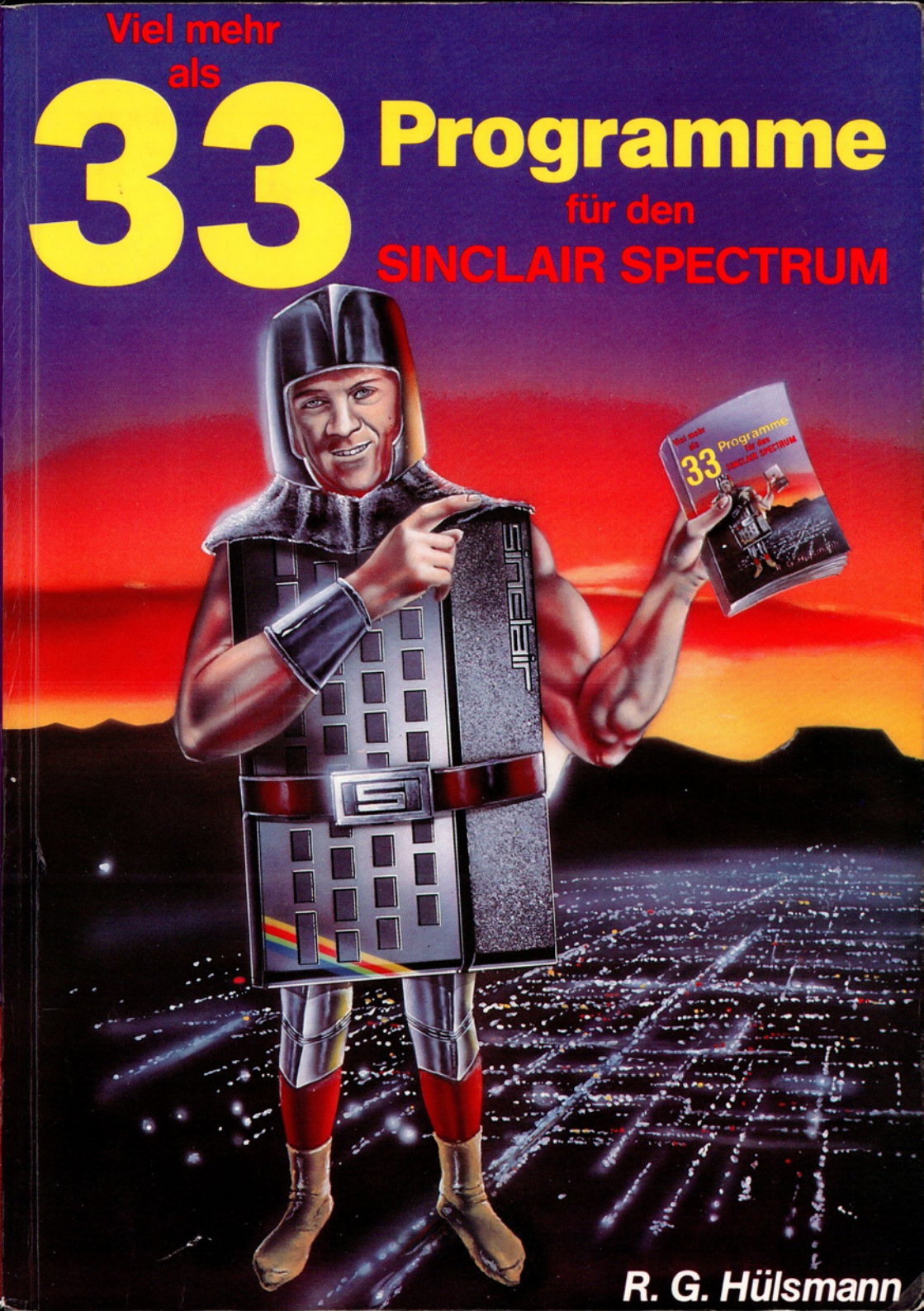
Viel mehr als

33 Programme

für den

SINCLAIR SPECTRUM



R. G. Hülsmann

ISBN 3–88963–144–4

Es kann keine Gewähr dafür übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen, sowie Programmlistings frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben auch keinen Aufschluß über eventuelle Verfügbarkeit oder Liefermöglichkeit. In jedem Falle sind die Unterlagen der Hersteller zur Information heranzuziehen.

Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen verboten. Programmlistings dürfen weiterhin nicht in irgendeiner Form vervielfältigt oder verbreitet werden. Alle Programmlistings sind Copyright der Fa. Ing. W. Hofacker GmbH. Verboten ist weiterhin die öffentliche Vorführung und Benutzung dieser Programme in Seminaren und Ausstellungen. Irrtum, sowie alle Rechte vorbehalten.

COPYRIGHT by Ing. W. HOFACKER ©1983,
Tegernseerstr. 18, 8150 Holzkirchen

1. Auflage 1983

Gedruckt in der Bundesrepublik Deutschland – Printed in West-Germany – Imprime'en RFA.

Inhaltsverzeichnis

1. Allgemeine Tips und Hinweise. 1
1.1 So wird Speicherplatz gespart 1
1.1.1 Wieviel Speicherplatz habe ich noch ? 1
1.1.2 Die Zeilenzahl reduzieren 1
1.1.3 Zahlen im Listing vermeiden 5
1.1.3.1 . . . durch den Einsatz von Variablen 5
1.1.3.2 . . . durch den Einsatz von Bedingungen 6
1.1.3.3 . . . durch den Einsatz von VAL 6
1.1.4 Einsatz von Zeilen ohne Zeilennummer. 7
1.2 Bedienelemente für schnelle Spiele 8
1.2.1 Das Problem 8
1.2.2 Die Lösung 9
1.2.3 Bauanleitung 9
1.2.3.1 Material 9
1.2.3.2 Einbau der Buchse in den SPECTRUM 9
1.2.3.3 Basteln der Bedienelemente. 11
1.3 Interaktive Graphik mit dem SPECTRUM. 13
1.3.1 Die Organisation des Video-Speichers 13
1.3.2 Die Funktionen SCREEN$, ATTR, POINT und PEEK 14
1.3.3 Drei "sehende Raupen". 14
1.3.4 24-Zeilen Display 18
1.3.5 Programm 0: "Testbild" 19

2. Programme für den SPECTRUM 16K 21
2.1 Programm 1: "Meteor" 21
2.2 Programm 2: "Space Invader". 23
2.3 Programm 3: "Mondlandung". 24
2.4 Programm 4: "Treibmine" 26
2.5 Programm 5: "MiniSector" 28
2.5.1 Minisector I. 28
2.5.2 Minisector II 31
2.6 Programm 6: "Ari" 32
2.7 Programm 7: "Ballistik" 34

2.8 Programm 8: "Spielautomat" 36
2.9 Programm 9: "Irrgarten" 39
2.10 Programm 10: "Todeshöhle" 41

3. Unterprogramme in Maschinensprache 45
3.1 Zur Plazierung von Unterprogrammen in Maschinensprache . 45
3.2 Zwei Programme, um Maschinencode zu laden 46
3.2.1 Programm 11: "Hex-Loader I" 46
3.2.2 Programm 12: "Hex-Loader II" 48
3.3 Programm 13: "Screen-Loader" 49
3.4 Programm 14: "Inversion" 50
3.5 Programme 15 + 16: "Vertical Scroll" 51
3.6 Programm 17: "Memory-Mapped-Display" 53

4. Superprogramme für den ZX SPECTRUM 57
4.1 Programm 18: "LP-Register" (48K) 57
4.2 Programm 19: "Kartei" (48K) 61
4.2.1 Einrichten der Kartei 65
4.2.2 Benutzen der Kartei 66
4.3 Programm 20: "Haushaltsbuchführung" (48K) 69
4.4 Programm 21: "Mondphasen" 73
4.5 Programm 22: "Biorhythmus" 76
4.6 Programm 23: "Uhr" 83
4.7 Programm 24: "Buchstaben-Puzzle" 85
4.8 Programm 25: "Das Monsterspiel" 89
4.9 Programm 26: "Bombendrohung" 95
4.10 Programm 27: "Minenfeld" 99
4.11 Programm 28: "ZX-Sector" 105
4.12 Programm 29: "Labyrint" (48K) 114
4.13 Programm 30: "Music-Computer" (48K) 121
4.14 Programm 31: "3-D-Graphik" (48K) 125
4.15 Programm 32: "Quadro" 129
4.16 Programm 33: "Crazy Kong" (48K) 133

Empfehlenswerte Literatur 138

1 Allgemeine Tips und Hinweise

1.1 So wird Speicherplatz gespart

1.1.1 Wieviel Speicherplatz habe ich noch ?

Um den genauen Speicherplatz zu erfahren, der noch für BASIC-Programm und Variablen zur Verfügung steht, ist ein Unterprogramm in Maschinensprache notwendig. Für die meisten Zwecke reicht aber folgende BASIC-Zeile aus:

```
9999 PRINT "NOCH ";PEEK 23730+256*PEEK 23731–PEEK
     23653–256*PEEK 23654;" BYTES"
```

Lediglich der Maschinenstapelzeiger wird nicht berücksichtigt (vgl. Handbuch, Seite 171). Bei größeren Programmen, bei denen die Gefahr besteht, den Speicher bis zum Schluß zu füllen, füge man diese Zeile immer ein. Dann hat man durch GOTO 9999 immer noch den vorhandenen Platz auf dem Bildschirm.

Was macht man nun, wenn der Speicherplatz eher zu Ende geht, als die Ideen für das Programm? Besonders häufig kommt dies vor, wenn man nur 16K-Byte RAM zur Verfügung hat.

Im folgenden wollen wir Ihnen kurz aufzeigen, wie man bis zu 20% und mehr Speicherplatz durch geschicktes Schreiben des Programmes sparen kann.

1.1.2 Die Zeilenzahl reduzieren

Wichtig:
Schreibe nie mehr Programmzeilen als unbedingt notwendig, wenn Du Platz sparen mußt. Im Notfall muß man eben mangelnde Übersichtlich-

keit in Kauf nehmen. Im ZX SPECTRUM benötigt jede Zeile, abgesehen vom Zeileninhalt, 5 Bytes: 2 Bytes Zeilennummer + 2 Bytes Zeilenlänge (nicht sichtbar) + 1 Byte Zeilenende (13; nicht sichtbar). Faßt man z. B. zwei Zeilen zu einer zusammen, so spart man 5 Bytes! Beispiel:

```
10 PRINT "ANZAHL DER GOLDSTUECKE: ";G
20 PRINT AT 5,0;"ANZAHL DER SCHWERTER: ";S
```

Dies kann man auch in einer Zeile schreiben:

```
10 PRINT "ANZAHL DER GOLDSTUECKE: ";G;AT 5,0; "ANZAHL
   DER SCHWERTER: ";S
```

Weiterhin können wir uns den Umstand zu Nutze machen, daß ein Text, der mittels AND mit einer Bedingung verknüpft wird, nur dann gedruckt wird, wenn die Bedingung erfüllt ist. So wird zum Beispiel

```
10 PRINT "DER TANK IST LEER" AND T=0
```

nur dann ausgeführt, wenn T wirklich exakt den Wert 0 angenommen hat. Folgende Zeilen können wir also zu einer zusammenfassen:

```
10 PRINT "DEIN WURF WAR: ";
20 IF W=1 THEN PRINT "EINS";
30 IF W=2 THEN PRINT "ZWEI";
40 IF W=3 THEN PRINT "DREI";
50 IF W=4 THEN PRINT "VIER";
60 IF W=5 THEN PRINT "FUENF";
70 IF W=6 THEN PRINT "SECHS";
80 PRINT "."
90 IF W>4 THEN PRINT "SEHR GUT."
```

Dieses Programm benötigt 207 Bytes. Fassen wir diese Zeilen nun zusammen:

```
10 PRINT "DEIN WURF WAR: ";"EINS" AND W=1; "ZWEI" AND
   W=2;"DREI" AND W=3;"VIER" AND W=4;"FUENF" AND W=5;
   "SECHS" AND W=6;"." ' "SEHR GUT." AND W>4
```

In Zeile 80 wird der SPECTRUM angewiesen eine neue Zeile zu benutzen, indem sie ohne "," oder ";" aufhört. Dasselbe wird in der komprimierten Fassung durch das " ' " erreicht.

Übrigens: Die komprimierte Fassung benötigt nur noch 153 Bytes. So konnten wir über 50 Bytes einsparen.

Auf diese Weise ist es also möglich, die verschiedensten Texte, die unter verschiedensten Bedingungen an den verschiedensten Ecken des Bildschirmes auftauchen sollen, in einem PRINT-Befehl zusammenfassen.

Ebenso können wir uns die Eigenart des SPECTRUM zu Nutze machen, daß eine erfüllte Bedingung den Wert "1" hat, während eine nicht erfüllte Bedingung den Wert "0" hat. Auf diese Weise kann man Programmschritte, die man normalerweise mit IF . . . THEN ausführt, auch ohne diesen Befehl ausführen:

```
IF A=5 THEN GOTO 1500
```

hat, wenn A=5 dieselbe Wirkung wie

```
GOTO 1500*(A=5)
```

Auch hier können wir wieder Zeilen zusammenfassen:

```
10 INPUT X
20 IF X=1 THEN GOTO 870
30 IF X=2 THEN GOTO 980
40 IF X=3 THEN GOTO 1220
50 GOTO 10
```

Die Zeilen 20 bis 50 fassen wir zusammen:

```
20 GOTO 870*(X=1)+980*(X=2)+1220*(X=3)
```

Übrigens: Wenn X weder 1, 2 oder 3 ist, dann hat der Ausdruck hinter dem GOTO den Wert 0. Soll er aber, was auch häufig vorkommt, nicht 0, sondern einen bestimmten Wert, in diesem Fall 10, haben, können

wir schreiben:

```
20 GOTO 10+850*(X=1)+970*(X=2)+1210*(X=3)
```

Auf diese Weise lassen sich übrigens alle ON . . . GOTO Befehle anderer BASIC-Dialekte in eine Zeile des SPECTRUM umwandeln. (Obige Zeile könnte z. B. im Originalprogramm geheißen haben: 20 ON X GOTO 870,980,1220)

Ein weiteres Beispiel:

```
100 IF Q$="JA" THEN GOTO 10
110 IF Q$="NEIN" THEN GOTO 560
130 GOTO 90
```

Fassen wir zusammen:

```
100 GOTO 90–80*(Q$="JA")+470*(Q$="NEIN")
```

Selbstverständlich funktioniert diese Technik nicht nur bei GOTO- oder GOSUB-Befehlen, sondern z. B. auch beim LET-Befehl:

```
100 IF A=0 THEN LET X=25*X+2↑X
110 IF A<0 THEN LET X=X–X/2
120 IF A>0 THEN LET X=A
```

Fassen wir zusammen:

```
100 LET X=(25*X+2↑X)*(A=0)+(X–X/2)*(A<0)+A*(A>0)
```

In dieser Summe, die aus drei Produkten besteht, ist nur das Produkt ungleich 0, welches die erfüllte Bedingung enthält.

Da in den meisten Programmen ja viele IF . . . THEN-Befehle vorkommen, läßt sich auf diese Weise recht viel Platz sparen. Aber dies ist nicht die einzige Methode.

1.1.3 Zahlen im Listing vermeiden . . .

1.1.3.1 . . . durch den Einsatz von Variablen

Jede Zahl im Listing des SPECTRUM benötigt außer den sichtbaren Ziffern noch 6 weitere Bytes, die nicht angezeigt werden: 14, der Code für "ZAHL", der dem SPECTRUM anzeigt, daß jetzt eine Zahl kommt und dann die 5 Bytes für die Zahl. Wie können wir diese 6 unsichtbaren Bytes vermeiden?

Eine Möglichkeit ist die Verwendung von Variablen. Einer Zahl, die sehr häufig im Programm vorkommt, kann man eine Variable zuweisen. Dann muß diese Zahl nur einmal benutzt werden. Die Variable hat im Listing nur die sichtbaren Zeichen.

Und was liegt hier näher, als römische Zahlzeichen zu benutzen, damit man die Variable im Listing sofort erkennt und weiß, welchen Wert sie hat. (Hier geht es natürlich um Variablen, die während des Programmablaufes nicht verändert werden. "Konstanten" wäre der treffendere Ausdruck.)

Bei größeren Spielprogrammen, die oft kleinere Zahlen benötigen, haben wir uns zur Angewohnheit gemacht, folgende Zeile an den Anfang zu stellen:

```
10 LET I=1:
   LET II=I+I:
   LET III=II+I:
   LET IV=III+I:
   LET V=IV+I:
   LET VI=VI+I:
   LET X=V+V:
   LET XX=X+X:
   LET C=X*X:
   LET M=X*C:
```

Zum Beispiel benötigt LET A=A+IV fünf Bytes weniger als LET A=A+4! GOTO V*M benötigt 7 Bytes weniger als GOTO 5000.

1.1.3.2 . . . durch den Einsatz von Bedingungen

Eine Bedingung ist 1, wenn sie erfüllt ist und 0, wenn sie nicht erfüllt ist. Die Zahlen 0 und 1 können wir also durch Bedingungen ersetzen. Nehmen wir an, Q sei im Programm als irgendeine Zahl bereits definiert, dann gilt: Q=Q hat den Wert 1 und Q)Q hat den Wert 0.

Man kann also schreiben:

```
20 LET P=Q=Q                statt  20 LET P=1
30 LET X=Q)Q                statt  30 LET X=0
40 PRINT TAB Q)Q;           statt  40 PRINT TAB 0;
50 FOR J=Q=Q TO 10          statt  50 FOR J=1 TO 10
```

Vielfache weitere Beispiele sind denkbar.

1.1.3.3 . . . durch den Einsatz von VAL

Mittels der Funktion VAL wird der rechnerische Wert einer Zeichenkette ausgerechnet. VAL "5–2" würde z. B. 3 ergeben.

Man kann Speicherplatz sparen, indem man Zahlen in Zeichenketten verwandelt und dann mittels VAL errechnen läßt. Beispiele:

```
20 LET P=VAL "235"          statt  20 LET P=235
30 GOTO VAL"1220"           statt  30 GOTO 1220
```

Pro in VAL " " gesetzter Zahl werden drei Bytes gespart: die sechs unsichtbaren Bytes verschwinden, dafür kommen drei neue Bytes (VAL " ") hinzu.

Mehr wird gespart, wenn man ganze Formeln und Rechenvorgänge in VAL " " setzen kann:

```
40 LET J=VAL "235+18/X+5*(23*X–22)"
```

Hier werden 30 (5*6) unsichtbare Bytes gespart, dafür aber nur drei neue hinzugefügt.

Da Bedingungen ja auch rechnerische Ausdrücke sind, die entweder den

Wert 1 oder 0 haben, kann man sie auch in VAL " " setzen. So wird aus

```
50 IF X=23 OR Y=12 THEN GOTO 340
```

die sparsame Zeile:

```
50 IF VAL "X=23 OR Y=12" THEN GOTO VAL "340"
```

Alles, was zwischen IF und THEN steht, kann immer in VAL " " gesetzt werden, wodurch Speicherplatz gespart wird, insofern in der Bedingung Zahlen vorkommen.

1.1.4 Einsatz von Zeilen ohne Zeilennummer

Diese letzte Methode ist am wirkungsvollsten! Viele Zeilen eines Programmes dienen ja nur dazu, Variablen und Arrays zu initieren. Wenn man diese Zeilen, nachdem das Programm einmal gelaufen ist, wieder gelöscht werden, sind die Variablen und Felder immer noch im Variablenspeicher enthalten. Das heißt: Man kann diese Zeilen auch ebensogut gleich direkt (ohne Zeilennummer) eingeben. Nur darf man das Programm dann nicht mehr mit RUN starten.

Diese Methode ist natürlich nur sinnvoll bei Variablen die ihren Wert während des Programmablaufes nicht verändern oder bei Variablen, deren Wert bei einem evtl. Neustart des Programmes bedeutungslos ist, weil er sowieso durch das Programm erst definiert wird. Auf diese Weise können auch Sätze und Wörter gespeichert werden, ohne daß sie im Listing auftauchen.

Bei Programmen, in denen auf diese Weise Platz gespart wurde, ist es sinnvoll sie so einzurichten, daß sie sich automatisch nach dem LOADen mit GOTO starten. Dann kann nichts mehr schief gehen.

Wenn man alle diese Methoden anwendet, sollte es möglich sein, wesentlich mehr als ohne sie in den Speicher zu bekommen. Man sollte in allen Fällen überlegen, ob wirklich Platz gespart wird. Es ist sinnlos, – um ein Beispiel zu nennen – einer Zahl, die nur einmal benutzt wird, eine Variable zuzuweisen. Dann steht die Zahl nämlich im Listing und im Variablenspeicher, was nur Platz ver-

braucht. Der durch die Verwendung der Variablen gewonnene Platz ist dann wieder zerronnen.

1.2 Bedienelemente für schnelle Spiele

Wer kennt das nicht:
Da hat man eine Kassette mit "SPACE INVADERS" in schnellem Maschinencode erstanden, ist froh, wenn man sie dem SPECTRUM verfüttert hat, drückt auf RUN . . . und dann ist man ständig dabei, die richtigen Tasten zu suchen, um den Angriffen auszuweichen und zu schießen. Dabei muß man ständig darauf achten, daß man nicht zu fest auf die Tasten einschlägt, sonst steigt der SPECTRUM wegen einer Erschütterung aus. Und spätestens wenn Freunde kommen, verzweifeln sie gänzlich an der Tastatur des SPECTRUM.

1.2.1 Das Problem

Das Problem ist zum einen, das der SPECTRUM nur eine Gummi-Tastatur hat und zum anderer, daß man für Spiele nie 40 Tasten braucht, sondern – je nach Spiel – zwei bis zehn! Selbst bei einer Normaltastatur, welche verschiedenste Anbieter verkaufen, wäre bei manchem schnellen Spiel die Suche nach der richtigen Taste zu langwierig. Das gilt besonders für Leute, die mit uns spielen wollen, aber keine Lust haben, sich erst einmal ein Viertel Jahr lang an die Tastatur des SPECTRUM zu gewöhnen.

Ein weiterer Aspekt: Nicht jedes Spiel benötigt die gleiche Tastatur. Für Space-Invaders ist jeder Knopf, der über "Rechts-" / "Links-" "Schuß" hinausgeht von übel, für andere Spiele ist es gut, sich in 8 Himmelsrichtungen bewegen zu können und zwei weitere Tasten für übrige Funktionen zu haben.

Es genügt also weder eine Normaltastatur anstelle einer Gummitastatur, noch eine fest am SPECTRUM angeschlossene Zusatztastatur. Ebenso sollten die Tasten nicht am SPECTRUM sein, denn es ist blöd, in größerer Runde den Computer zum Spiel herum zu reichen.
Irgendwann gibt's dann wegen unvorsichtiger Bewegung einen Crash!

1.2.2 Die Lösung

Als Lösung eignet sich daher eine Buchse am SPECTRUM, an der beliebige Steuereinheiten anzuschließen sind. Die Steuereinheiten sollten – mit Rücksicht auf unseren Geldbeutel – sehr billig selbst herzustellen sein. Dann lohnt es sich u. U. für ein besonders interessantes Spiel eine eigene Steuereinheit zu erstellen.

Da in der Regel zehn Tasten ausreichen, haben wir die oberste Tastenreihe (1 – 0) gewählt, die wir nach außen legen möchte. In der folgenden Bauanleitung beschreiben wir die Herstellung zweier Bedienelemente, die für die meisten Spiele ausreichen. Bei der Erstellung eigener Spiele kann man diese Steuereinheiten dann berücksichtigen und fremde Spiele kann man evtl. umändern.

1.2.3 Bauanleitung

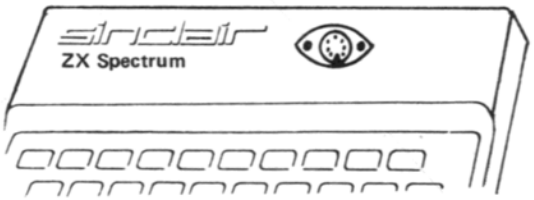
1.2.3.1 Material

Werkzeug (Lötkolben 15W, Zangen, Schraubenzieher, Lötzinn . . .)

1 7-polige DIN-Einbaubuchse
2 7-polige DIN-Stecker
13 Miniatur-Taster
4m siebenpoliges Kabel (Telefonkabel o. ä.)
2 kleine Gehäuse

Die Tasten sollten direkt am Gehäuse zu befestigen sein. Für die zweite Steuereinheit läßt sich evtl. eine alte Dia-Projektor-Fernbedingung verwenden. Manchmal sind einzelne Fernbedienungen günstig zu bekommen.

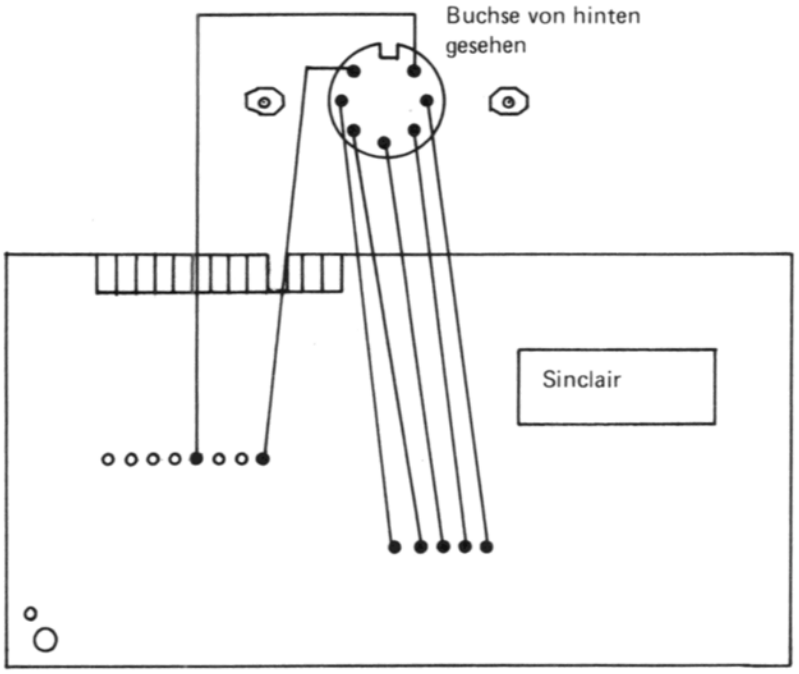
1.2.3.2 Einbau der Buchse in den SPECTRUM



Es empfiehlt sich, zuerst mit dem Lötkolben ein Loch entsprechender Größe in das Gehäuseoberteil zu machen, es dann auszufeilen und dann

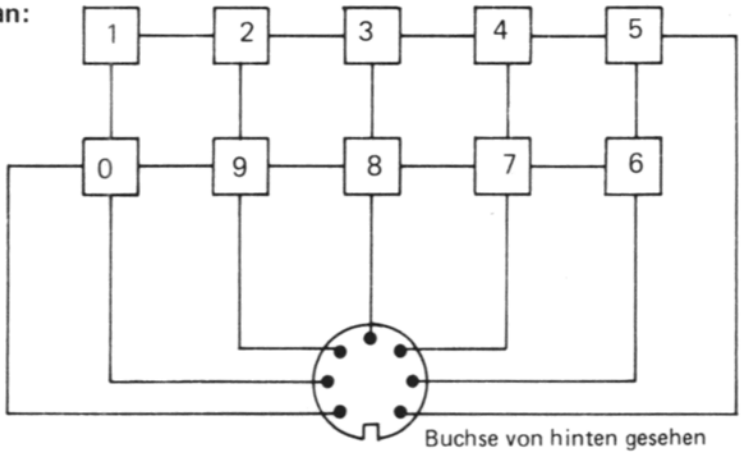
die Löcher für die Schrauben zu bohren. Nachdem die Buchse eingebaut ist (siehe Abbildung), geht die Kleinarbeit, die Verdrahtung los. Nach folgenden Zeichnungen aber ist auch dies kein Problem:

Verdrahtungsskizze:



Ansicht von der Unterseite der Platine

Schaltplan:



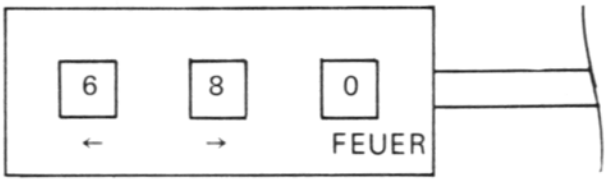
Nach der Verdrahtung bauen wir den SPECTRUM wieder zusammen und überprüfen die Funktion der Tasten 1 bis 0. Sollten die Tasten nicht mehr so recht funktionieren: Lötstellen überprüfen, ob irgendwo zwei Kabel Kontakt miteinander haben, die dies nicht dürfen! Ansich ist der Aufbau unproblematisch und es dürfte nichts schief gehen.

Der SPECTRUM ist nun dafür vorbereitet, die Tasten 1 – 0 über beliebige Steuergeräte von außen ansteuern zu lassen!

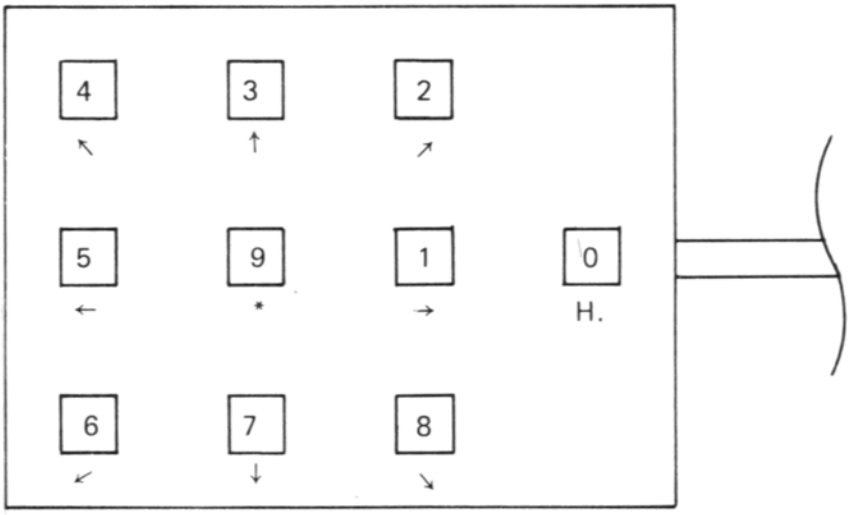
1.2.3.3 Basteln der Bedienelemente

Folgende beiden Bedienelemente sind nahezu universell einsetzbar. Unter den Tasten ist eine mögliche Bedeutung angedeutet:

Bedienelement 1



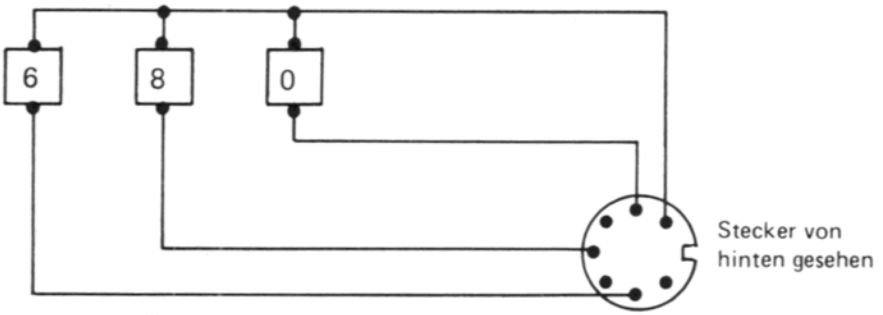
Bedienelement 2



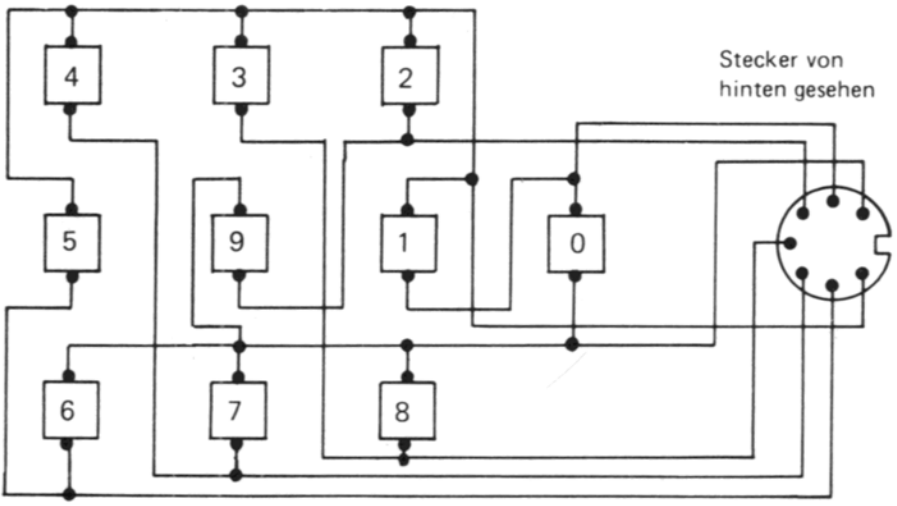
Zunächst fertigen wir die entsprechenden Bohrungen und bauen die Miniaturtaster ein. Je Bedienelement benötigen wir etwa 1,2 bis 1,5 m Kabel. Diese Länge hat sich als recht brauchbar für bis zu sechs Mitspieler erwiesen.

Auch hier ist der Aufbau derart unproblematisch, daß folgende Verdrahtungsskizzen ausreichen sollten:

Bedienelement 1



Bedienelement 2



Die Anordnung der Tasten im Bedienelement 2 erscheint vielleicht ungewöhnlich, ist aber für das Programmieren sehr praktisch. Alle Tastendrücke zwischen 2 und 4 beinhalten eine Bewegung nach oben, alle zwischen 4 und 6 nach links, alle größer als 5 nach unten und alle kleiner als zwei oder größer als 7 nach rechts.

Sehr einfach läßt sich ein Programm zur Abfrage der Tasten in einem Spiel entwickeln, indem man sich nach acht Himmelsrichtungen bewegen muß. Siehe "Minenfeld" und "ZX-SECTOR" in diesem Buch. Wenn alle Lötstellen in Ordnung sind, funktionieren die Bedienelemente und sind einsatzbereit. In diesem Buch wird auf die Bedienelemente hingewiesen, wenn sie eingesetzt werden sollten.

1.3 Interaktive Graphik mit dem SPECTRUM

1.3.1 Die Organisation des Video-Speichers

Der SPECTRUM hat ein Memmory-Mapped-Display, welches sich in zwei Teile gliedert:

Von 16384 bis 22527 ist das "Display File". In 6144 Bytes a' 8 Bit ist für die 49152 möglichen Bildpunkte angegeben, ob sie gesetzt (=INK-Farbe) oder gelöscht (=PAPER-Farbe) sind. Der Aufbau dieses Speicherbereiches ist derart kompliziert, daß PEEKen und POKEn aufwendiger und langsamer wäre, als die speziellen Funktionen POINT und SCREEN$, bzw. PRINT AT und PLOT.

Probieren Sie bitte zur Demonstration folgendes kurzes Programm:

```
10 FOR i=16384 TO 22527:POKE i,255: NEXT i
```

In den Adressen 22528 bis 23297 befinden sich die Attribute der 768 möglichen PRINT-Positionen. Jedes Zeichen ist 8 x 8 Bildpunkte groß. Ein Attribut kennzeichnet INK-Farbe, PAPER-Farbe, BRIGHT 1 oder 0, FLASH 1 oder 0. Der Wert eines Charakterfeldes errechnet sich folgendermaßen: Code für die INK-Farbe + 8 x den Code für die PAPER-Farbe + 64, wenn das Zeichen BRIGHT 1 ist + 128, wenn das Zeichen FLASH 1 ist.

Ein strahlendes blinkendes Zeichen, gelb auf blauem Grund hätte dem-

nach den Attributswert 206. Probieren Sie: POKE 22528,206.

1.3.2 Die Funktionen SCREEN$, ATTR, POINT und PEEK

Eines haben alle vier Funktionen gemeinsam: Sie haben als Ergebnis den Zustand eines bestimmten Speicherbereiches.

SCREEN$(x,y) gibt an, welches Zeichen sich in Zeile x an der Position y befindet. Aber Achtung: Diese Funktion funktioniert nur bei Zeichen, deren Code kleiner als 128 ist! Bei Graphikzeichen und selbstdefinierten Zeichen ist das Ergebnis entweder ein Leerfeld (Code 32) oder die leere Zeichenkette (Nullstring, Code 0). Positiv: Die Funktion ist erstaunlich schnell.

ATTR (x,y) gibt an, welches Attribut das Zeichen in Zeile x, Position y hat.

POINT (x,y) ist 1, wenn der Bildpunkt x,y gesetzt ist (etwa durch PLOT x,y) und 0, wenn er nicht gesetzt ist.

PEEK x ergibt den Wert, den die Speicherzelle (ein Byte) x hat. Dies ist ein Wert zwischen 0 und 255. Mittels dieser vier Funktionen gibt der SPECTRUM erstaunliche Möglichkeiten, interaktive Graphik zu programmieren. Mit ihm ist interaktive Graphik wesentlich einfacher, als mit vielen anderen Computern, die lediglich die Möglichkeit haben, durch PEEK den Bildspeicher zu lesen.

Als Beispiel möge die bekannte "sehende Raupe" dienen, die drei Möglichkeiten, interaktive Graphik zu programmieren, zeigt:

1.3.3 Drei "sehende" Raupen

Hier wird deutlich, was "interaktiv" bei Graphik heißt: Der Computer schreibt nicht nur in Bildspeicher, etwa mittels PRINT, PLOT, DRAW oder POKE, sondern liest auch in ihm. Der Computer kann also aktiv auf das reagieren, was bereits auf dem Bildschirm ist. Dies ist unerläßlich bei Spielen, wo es wichtig ist, ob eine Figur mit anderen zusammenstößt, etc.!

Tippen Sie ruhig diese drei Programme nacheinander ein und versuchen Sie, diese zu verstehen:

```
 10 REM Sehende Raupe
 20 LET zz=22
 30 LET zl=32
100 FOR x=1 TO zz-2: FOR y=1 TO
zl-2
110 PRINT AT x,y;"■": NEXT y: N
EXT x
120 FOR x=0 TO zz-1
130 PRINT AT x,0;"#"
140 PRINT AT x,zl-1;"#"
150 NEXT x
160 FOR y=0 TO zl-1
170 PRINT AT 0,y;"#"
180 PRINT AT zz-1,y;"#"
190 NEXT y
300 DIM x(7): DIM y(7)
310 LET x=zl/2
320 LET y=0
330 LET x1=1
340 LET y1=1
350 IF SCREEN$ (y+y1,x)="#" THE
N LET y1=-y1
360 IF SCREEN$ (y,x+x1)="#" THE
N LET x1=-x1
370 LET x=x+x1: LET y=y+y1
380 PRINT AT y,x;"■"
390 IF x=x(6) AND y=y(6) AND x=
x(2) AND y=y(2) THEN STOP
400 LET x(7)=x(6): LET y(7)=y(6
)
410 LET x(6)=x(5): LET y(6)=y(5
)
420 LET x(5)=x(4): LET y(5)=y(4
)
430 LET x(4)=x(3): LET y(4)=y(3
)
440 LET x(3)=x(2): LET y(3)=y(2
)
450 LET x(2)=x(1): LET y(2)=y(1
)
460 LET x(1)=x: LET y(1)=y
470 PRINT AT y(7),x(7);"■"
480 PRINT AT INT (RND*(zz-2)+1)
,INT (RND*(zl-2)+1);"#"
490 PRINT AT INT (RND*(zz-2)+1)
,INT (RND*(zl-2)+1);"■"
500 GO TO 350
```

Drücken Sie bitte RUN und los geht's!

Wie funktioniert es ?

20 – 30 : Zeilenzahl (zz) und Zeilenlänge (zl) werden bestimmt.

100–110 : Das Feld wird gezeichnet. Die Umrandung wird frei gelassen.
120–150 : Die senkrechten Randstreifen werden gezeichnet.
160–190 : Die waagrechten Randstreifen.

Jetzt beginnt das eigentliche Programm:

300 : In x(7) und y(7) sind die sieben Positionen des Raupenkörpers gespeichert.
310–320 : Der Anfangspunkt wird errechnet.
330–340 : Die Anfangsrichtung wird eingegeben: Auf der x- und der y-Achse jeweils +1, d. h. nach rechts unten.
350 : Es wird mit SCREEN$ überprüft, ob in der waagrechten Richtung ein "#" ist. Wenn ja, wird die waargrechte
Wenn Ja, wird die waagrechte Richtung umgekehrt.
360 : Hier geschieht das gleiche für die senkrechte Richtung. Das Feld, auf das die Raupe direkt zuläuft (y+y1, x+x1) wird nicht wahrgenommen. Daher wird es gefressen, indem die Raupe einfach drüberläuft.
370–380 : Der nächste Punkt des Raupenkopfes wird errechnet und gezeichnet.
390 : Wenn die Raupe hoffnungslos mit sich selbst verknotet ist, stoppt das Programm.
400–460 : Die Raupe zieht den Schwanz nach. Wieder ist der eben gezeichnete Kopf in x(1), y(1) und das Schwanzende in x(7), y(7).
470 : Das alte Schwanzende wird gelöscht.
480–490 : Jetzt kommt Bewegung ins Spiel: Innerhalb der Spielfeldgrenzen wird per Zufall ein Begrenzungszeichen gesetzt und eines gelöscht.
500 : Zurück nach 350.

Bei diesem Programm wurde hauptsächlich die Funktion SCREEN$ verwendet. Das folgende Programm erreicht ähnliches mit Farbe und der ATTR-Funktion:

```
 10 REM Sehende Raupe
 20 LET zz=22
 30 LET zl=32
 40 BORDER 2: PAPER 2: CLS
100 FOR x=1 TO zz-2: FOR y=1 TO
zl-2
110 PRINT AT x,y; INK 1;"■": NE
XT y: NEXT x
120 FOR x=0 TO zz-1
130 PRINT AT x,0; INK 6;"■"
140 PRINT AT x,zl-1; INK 6;"■"
150 NEXT x
```

```
  160 FOR y=0 TO zl-1
  170 PRINT AT 0,y; INK 6;"■"
  180 PRINT AT zz-1,y; INK 6;"■"
  190 NEXT y
  300 DIM x(7): DIM y(7)
  310 LET x=zl/2
  320 LET y=0
  330 LET x1=1
  340 LET y1=1
  350 IF ATTR (y+y1,x)=22 THEN LE
T y1=-y1
  360 IF ATTR (y,x+x1)=22 THEN LE
T x1=-x1
  370 LET x=x+x1: LET y=y+y1
  380 PRINT AT y,x; INK 5;"O"
  390 IF x=x(6) AND y=y(6) AND x=
x(2) AND y=y(2) THEN STOP
  400 LET x(7)=x(6): LET y(7)=y(6
)
  410 LET x(6)=x(5): LET y(6)=y(5
)
  420 LET x(5)=x(4): LET y(5)=y(4
)
  430 LET x(4)=x(3): LET y(4)=y(3
)
  440 LET x(3)=x(2): LET y(3)=y(2
)
  450 LET x(2)=x(1): LET y(2)=y(1
)
  460 LET x(1)=x: LET y(1)=y
  470 PRINT AT y(7),x(7); INK 1;"
■"
  480 PRINT AT INT (RND*(zz-2)+1)
,INT (RND*(zl-2)+1); INK 6;"■"
  490 PRINT AT INT (RND*(zz-2)+1)
,INT (RND*(zl-2)+1); INK 1;"■"
  500 GO TO 350
```

Sie können ja mal messen, welche Technik schneller ist !

Die dritte Version der Raupe benutzt PEEK und POKE, um den Attributsspeicher zu füllen und zu lesen. In bp wird das erste Byte dieses Speichers festgehalten, daß die Printposition 0,0 kennzeichnet. Die Adressen der Körperteile sind in p(7) gespeichert, p ist der aktuelle Raupenkopf:

```
   10 REM Sehende Raupe
   20 LET zz=22
   30 LET zl=32
   40 LET bp=22528
  100 FOR x=1 TO zz-2: FOR y=1 TO
 zl-2
  110 POKE bp+x*zl+y,8: NEXT y: N
EXT x
```

```
120 FOR x=0 TO zz-1
130 POKE bp+x*zl,48
140 POKE bp+x*zl+zl-1,48
150 NEXT x
160 FOR y=0 TO zl-1
170 POKE bp+y,48
180 POKE bp+y+(zz-1)*zl,48
190 NEXT y
300 DIM p(7)
310 LET X=zl/2
320 LET p=bp+zl+x
330 LET x1=1
340 LET y1=zl
350 IF PEEK (p+y1)=48 THEN LET
y1=-y1
360 IF PEEK (p+x1)=48 THEN LET
x1=-x1
370 LET p=p+x1+y1
380 POKE p,40
390 IF p=p(6) AND p=p(2) THEN S
TOP
400 LET p(7)=p(6)
410 LET p(6)=p(5)
420 LET p(5)=p(4)
430 LET p(4)=p(3)
440 LET p(3)=p(2)
450 LET p(2)=p(1)
460 LET p(1)=p
470 POKE p(7),8
480 POKE bp+INT (RND*(zz-2)+1)*
zl+INT (RND*(zl-2)+1),48
490 POKE bp+INT (RND*(zz-2)+1)*
zl+INT (RND*(zl-2)+1),8
500 GO TO 350
```

Weitere Beispiele interaktiver Graphik befinden sich z. B. in den Programmen "Irrgarten","Labyrinth", "Crazy Kong" und anderen.

1.3.4 24-Zeilen Display

Beim ZX81 war es sehr einfach, 24 Zeilen zu benutzen. Man brauchte nur 16418 auf Null zu POKEn und aufzupassen, daß in diesem Zustand keine INPUTs verlangt wurden. Versucht man das gleiche beim SPECTRUM (die entsprechende Systemvariable hat die Adresse 23659), so wird der Bildschirm schwarz und nichts geht mehr. Der SPECTRUM, der im Vergleich zum ZX81 erfreulich selten aussteigt, ist ausgestiegen.

Folgende Technik hingegen funktioniert:

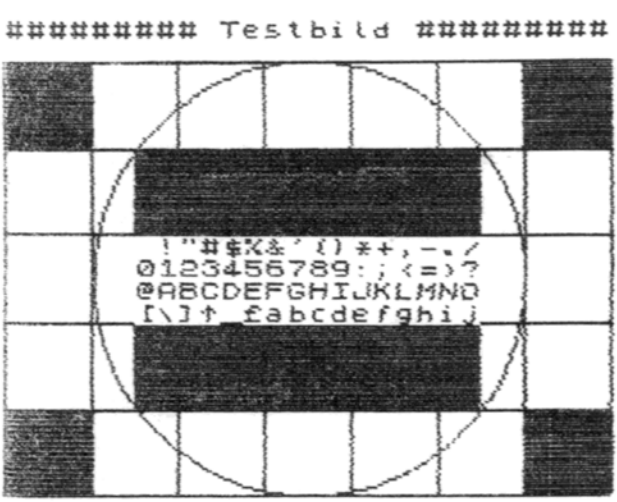
```
  10 PRINT AT 21,0;"Letzte Zeile
"
  20 PRINT #0;AT 0,0; INK 2;"xxx
xx"; INK 3; FLASH 1;"oooooooo"
  30 GO TO 30
```

Zeile 20 schreibt in den unteren Bildschirmbereich. Auf diese Weise lassen sich bei Graphik-Spielen da erklärende Texte, Scores und ähnliches unterbringen. Jede Fehlermeldung und jedes INPUT löschen diesen Bereich. Um den Bereich im Programm zu löschen geben Sie einfach ein Schein-INPUT ein: INPUT " ". Die untersten zwei Zeilen werden gelöscht, das Programm aber nicht angehalten. Die Funktion SCREEN$, ATTR und PEEK funktionieren auch im unteren Bildschirmteil, in den Zeilen 22 und 23. Ebenso sind in diesem Bereich Farben und FLASH möglich:

```
  10 PRINT AT 0,0; FLASH 1; INK
1;"Oberer Bereich";#0;AT 0,0; FL
ASH 1; INK 2;"Unterer Bereich"
  20 GO TO 20
```

1.3.5 Programm 0: "Testbild"

Zum Abschluß dieses Kapitels gleich ein erstes nützliches Programm. Ein Testbild, um Farbe und Kontrast auf dem Fernseher optimal einzustellen:



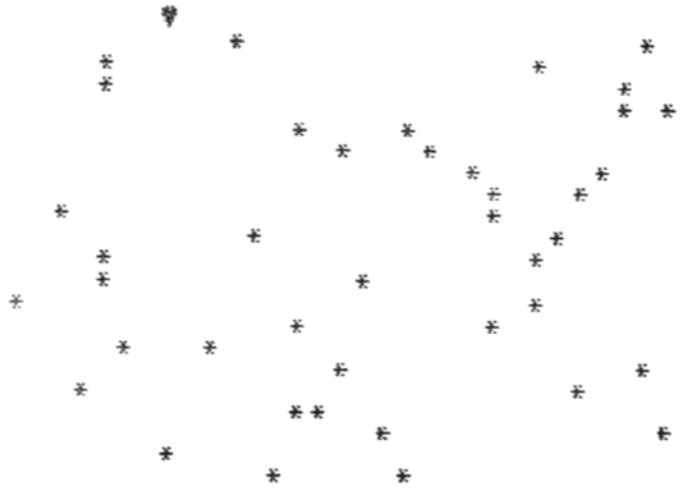
Der Drucker druckt leider keine Farben! In den großen schwarzen Kästen sind alle acht Farben in BRIGHT 0 und in BRIGHT 1 zu sehen, insgesamt also 16 Farben.

```
  10 REM TESTBILD
  20 BORDER 0: PAPER 0: INK 7: C
LS : PRINT "########## Testbild #
#########"
  30 FOR i=0 TO 255 STEP 32: PLO
T i,0: DRAW 0,159: NEXT i
  40 FOR i=0 TO 172 STEP 32: PLO
T 0,i: DRAW 224,0: NEXT i
  50 CIRCLE 112,80,80
  60 FOR i=2 TO 5: PRINT AT i,0;
 INK 7;"      ": NEXT i
  70 FOR i=18 TO 21: PRINT AT i,
0; INK 6;"      ": NEXT i
  80 FOR i=18 TO 21: PRINT AT i,
24; INK 5;"      ": NEXT i
  90 FOR i=2 TO 5: PRINT AT i,24
; INK 4;"      ": NEXT i
 100 FOR i=6 TO 9: PRINT AT i,6;
: FOR j=0 TO 7: PRINT INK j;"  "
;: NEXT j: NEXT i
 110 FOR i=14 TO 17: PRINT AT i,
6;: FOR j=7 TO 0 STEP -1: PRINT
BRIGHT 1; INK j;"  ";: NEXT j: N
EXT i
 120 PAPER 7: PRINT INK 0;AT 10,
6;" !""#$%&'()*+,-./"; INK 2;AT
11,6;"0123456789:;<=>?"; INK 1;A
T 12,6;"@ABCDEFGHIJKLMNO"; INK 3
;AT 13,6;"[\]↑_£abcdefghij"
 130 PAPER 0: GO TO 130
```

2 Programme für den SPECTRUM 16K

Anschließend folgen 10 kurze Programme für den SPECTRUM mit 16K-RAM.

2.1 Programm 1: "Meteor"



```
  10 BORDER 0: PAPER 1: INK 6: C
LS
  20 DIM s(2)
  30 PRINT "  METEOR  "
  40 LET z=7
  60 RESTORE
  70 FOR i=0 TO 7: READ a: POKE
USR "A"+i,a: NEXT i
  80 DATA BIN 00101000,BIN 01101
100,BIN 01111100,BIN 01101100,BI
N 00111000,BIN 00111000,BIN 0001
0000,BIN 00010000
```

```
100 LET x=0: INPUT "Druecke auf
""ENTER""";q$
110 LET x=x+1
130 IF SCREEN$ (0,z)<>"*" THEN
GO TO 300
140 BEEP .1,-5: BEEP .1,20
150 INPUT (x-22;" Punkte")'"Spi
eler? ";s
160 IF s<1 OR s>2 THEN GO TO 15
0
190 LET s(s)=s(s)+x-22
200 CLS : PRINT "Spieler 1: ";s
(1)'"Spieler 2: ";s(2)
210 GO TO 100
300 PRINT AT 0,z;"#": BEEP .02,
-10
310 LET z=z+(INKEY$="8" AND z<3
1)-(INKEY$="5" AND z>1)
320 LET a=INT (RND*31)
330 LET b=INT (RND*31)
340 PRINT BRIGHT 1;AT 21,a;"*";
AT 21,b;"*"
350 POKE 23692,255: PRINT
360 GO TO 110
```

Variablen:

S(2)	Punkte der Spieler 1 + 2
X	zählt die erschienenen Meteore. Da immer 22 Meteore auf dem Bildschirm sind, ist X-22 die Punktzahl, d. h. die Zahl der Meteore, denen man ausgewichen ist.
Z	Position des Raumschiffes in Zeile 0
A, B	Zufallszahlen, die die Positionen der neu erscheinenden Meteore bestimmen.

Nach dem das Programm geRUNt wird, erscheint auf dem Bildschirm der Titel "Meteor". Durch Druck auf ENTER wird das Spiel gestartet. In der obersten Zeile erscheint das eigene Raumschiff. Von unten kommt ein Meteorsturm auf das Raumschiff zu, welches durch die Tasten "5" (⟨) und "8" (⟩) innerhalb der Grenzen des benutzten Bildschirmes bewegt werden kann (Zeile 310). Weicht man einem ankommenden Meteor nicht rechtzeitig aus, so wird der Schirm blau und die Punkte werden angezeigt. Mit "Spieler?" fragt der SPECTRUM, ob dies der erste oder der zweite Spieler war. Nach der Eingabe (1 oder 2) zeigt er den Punktestand beider Spieler, wobei der Punktestand des ersten Spielers links steht. Durch neuerliches Drücken von ENTER wird das Spiel für den nächsten Spieler gestartet.

2.2 Programm 2: "Space Invader"

```
  10 BORDER 0: PAPER 1: INK 6: C
LS
  20 LET y=8: LET q=0
  30 DIM z(4)
  40 PRINT AT 21,0; PAPER 4;"
     SPACE INVADER              ";
PAPER 1
  50 GO SUB 1000
  60 LET b$="  "
  70 LET a$="Ⓐ"
  80 IF LEN a$>32 THEN GO TO 70
  90 PRINT AT 0,0; INK 2;a$; INK
 6
 100 BEEP .02,12: BEEP .02,14: L
ET a$=b$+a$
 110 PRINT AT 0,31;b$
 120 FOR i=1 TO 4: LET z(i)=INT
(RND*32): NEXT i
 130 IF INKEY$="0" THEN GO SUB 5
00
 140 FOR x=2 TO 20 STEP 6
 150 LET y=y+(INKEY$="8" AND y<2
9)-(INKEY$="5" AND y>0)
 160 PRINT AT 20,y; INK 5;" ▲ "
 170 INK 6: FOR i=1 TO 4: PRINT
AT x,z(i);".": IF x>6 THEN PRINT
 AT x-6,z(i);" ": NEXT i
 180 IF x<20 THEN NEXT x
 190 FOR i=1 TO 4: IF z(i)=y+1 T
HEN GO TO 230
 200 NEXT i: NEXT x
 210 LET q=q+1
 220 GO TO 80
 230 BEEP .05,-8: BEEP .1,0: INP
UT (q;" Punkte. Noch einmal(? ");
q$
 240 IF q$<>"Nein" AND q$<>"nein
" THEN RUN
 250 STOP
 500 INK 7: BRIGHT 1: PLOT 8*y+1
1,14: DRAW 0,154
 510 INK 1: BRIGHT 0: PLOT 8*y+1
1,14: DRAW 0,154
 520 IF LEN a$<>y+3 THEN RETURN
 530 PRINT INK 6;AT 0,0;a$
 540 LET q=q+100
 550 RETURN
1000 RESTORE : FOR i=0 TO 7: REA
D a: POKE USR "▲"+i,a: NEXT i
1010 DATA 0,0,16,16,56,56,254,23
8
1020 FOR i=0 TO 7: READ a: POKE
USR "Ⓐ"+i,a: NEXT i
1030 DATA 129,126,165,255,255,66
,129,66
1040 RETURN
```

Variablen:

Y	Standort der Abwehrstation in Zeile 20
Q	zählt die Punkte (1 Punkt pro Bombe)
B$	Die Anzahl der Spaces vor dem feindlichen Mutterschiff (A$)
Z(4)	Zufallszahlen, die bestimmen, wo die Bomben der Fremden fallen.
X	Vertikale Position der Bombe.

Nach RUN erscheint das Bild auf dem Bildschirm. Ziel ist es, durch Drücken der Tasten "5" und "8" dem Bombenhagel der Fremden auszuweichen. Jede Bombe, die nicht trifft, zählt einen Punkt. 100 Sonderpunkte gibt es, wenn man das Mutterschiff der Angreifer trifft. Um mit Laser zu feuern, muß auf "0" gedrückt werden. Wird der Schutzschirm des Mutterschiffes getroffen, leuchtet er auf.

Wird die Abwehrstation von einer Bombe getroffen, stoppt das Spiel und die Punkte werden angezeigt.

2.3 Programm 3: "Mondlandung"

```
   5 BORDER 1: PAPER 1: INK 7: C
LS
  10 LET x=0: LET v=0: LET m=750
  20 LET h=1000: LET t=0: LET s=
0
  30 LET l=1.6: LET g=2000
  40 LET g$="--GUT!--": LET s$="
* SOS *"
  50 GO SUB 800
  60 PRINT "■■■■ MONDLANDUNG ■■■
■"
  70 PRINT '''"Hoehe   Tempo   T
ank    Beschl. __________________
________________"
  80 GO SUB 700
 100 INPUT AT 0,0;"Zeit: ";z'"Be
schleunigung: ";s
 120 LET z=INT z: IF z<0 THEN GO
TO 100
 130 FOR x=t TO t+z
 140 LET r=g+m
 150 LET v=v+l-s
 160 LET h=h-v
 170 GO SUB 700
 180 LET m=m-r*ABS s/800
 190 IF h<=0 THEN GO TO 500
 200 IF m<0 THEN GO TO 600
```

```
 210 NEXT x
 220 LET t=x
 230 GO TO 100
 500 IF v>5 THEN LET m=0: GO SUB
 710: FOR i=50 TO -50 STEP -1: B
EEP .01,i: NEXT i
 510 PRINT '''"Gelandet mit ";v;
" m/s²"'"nach ";x;" Sekunden."
 520 PRINT 'g$ AND v<3;s$ AND v>
5
 530 INPUT "Noch einmal? ";q$
 540 IF q$<>"Nein" AND q$<>"nein
" THEN RUN
 550 STOP
 600 PRINT s$
 610 LET t=x: LET z=9e9
 620 LET m=0
 630 LET s=0
 640 GO TO 130
 700 PRINT AT 0,26;x;AT 7,0;INT
h;"    ";TAB 8;INT v;"    ";TAB 16
;INT m;"    ";TAB 24;s;"    "
 710 IF m=0 THEN RESTORE 720: FO
R i=1 TO 9: READ a: BEEP a/20,20
: PAUSE 2: NEXT i: RETURN
 720 DATA 1,1,1,3,3,3,1,1,1
 740 BEEP .01,-20: PAUSE 20: RET
URN
 800 RESTORE 810: FOR i=0 TO 7:
READ a: POKE USR "1"+i,a: NEXT i
 810 DATA BIN 01100000,BIN 00010
000,BIN 00100000,BIN 01110000,0.
0,0,0
 820 RETURN
 900 SAVE "M" LINE 1
 910 RUN
```

Variablen:

V	Geschwindigkeit der Kapsel in Richtung Mondboden (negativ = vom Mond weg) (m/sec)
M	Treibstoffmasse zu Beginn (kg)
H	Hoehe über dem Mondboden in Meter
T	Zeit in Sekunden
S	Beschleunigung der Kapsel vom Mond weg (negativ = auf den Mond zu) m/sec^2)
Z	Beschleunigungszeit
L	Mondanziehung (1.6 m/sec^2)
G	Kapselgewicht (2000 kg)

Der SPECTRUM erwartet bei der Eingabe zuerst die Zeit (in Sekunden) und dann die Beschleunigung (in m/sec^2) auf eine Stelle hinter dem

Dezimalpunkt genau. Bedenken Sie bei der Eingabe der Beschleunigung, daß der Mond eine Anziehungskraft von 1,6 m/sec^2 hat. Wollen Sie die Fähre zum Schweben bringen? Bringen Sie sie auf Geschwindigkeit 0 und beschleunigen Sie dann mit 1,6 m/sec^2; dann schweben Sie in gleichbleibender Höhe über der Mondoberfläche.

Empfehlung:
Die Fähre zunächst fallen lassen, dann abbremesen und langsam landen.

Nach der Landung zeigt der SPECTRUM Landegeschwindigkeit und Zeit an. Bei einer sehr guten Landung zeigt er zusätzlich "––GUT––" und beim einem Aufprall zusätzlich "* SOS * " an. Landegeschwindigkeiten zwischen 3 und 5 m/sec bleiben unkommentiert.

2.4 Programm 4: "Treibmine"

```
  10 BORDER 1: PAPER 1: INK 7: C
LS
  20 LET q=0
  30 LET y=12: LET x=12: LET a=1
2: LET b=12
  40 RESTORE : FOR i=0 TO 23: RE
AD c
  50 POKE USR "a"+i,c: NEXT i
  60 DATA 60,126,255,255,255,255
,126,60,16,84,56,254,56,84,16,0,
130,219,24,68,82,72,44,193
  70 FOR j=12 TO 1 STEP -1
  80 IF j>9 THEN PRINT INVERSE 1
;CHR$ (j+55)
  90 IF j<=9 THEN PRINT INVERSE
1;CHR$ (j+48)
 100 NEXT j
 110 PRINT INVERSE 1;"0123456789
ABC"; INVERSE 0;"Y"
 120 PRINT AT 0,0;"X"
 150 LET a1=a
 160 LET b1=b
 200 LET a=ABS (a-INT (RND*3))
 210 LET b=ABS (b-INT (RND*3))
 220 IF x<>a OR y<>b THEN GO TO
300
 230 FOR i=1 TO 20: BEEP .01,20:
 BEEP .01,-20: NEXT i
 240 PRINT AT a1-12,b1;" ";AT x-
12,y; BRIGHT 1; INK 4;"X";AT 21,
12; FLASH 1; INK 6;"TREFFER!"
 250 INPUT "Druecke auf ""ENTER"
"! ";q$
 260 IF q$<>"" THEN STOP
 270 CLS
```

```
 280 LET q=q+1
 290 GO TO 30
 300 BEEP .1,-10: PRINT AT a1-12
,b1;" "
 310 PRINT AT a-12,b; INK 2; FLA
SH 1;"●"
 315 IF NOT a OR NOT b THEN GO T
O 400
 320 LET x1=x
 330 LET y1=y
 340 INPUT AT 0,0;"X=";x;"Y=";y
 360 PRINT AT x1-12,y1;" "
 370 PRINT AT x-12,y; INK 6;"*"
 380 GO TO 150
 400 FOR i=6 TO -6 STEP -1: BEEP
 .2,i: NEXT i
 405 PRINT AT 20,0; BRIGHT 1; PA
PER 7; INK 2;"●●● ";q;" TREFFER
●●●"
 410 INPUT "Noch ein Spiel? (j/n
) ";q$
 420 IF q$="j" THEN RUN
```

Variablen:

Q	Anzahl der zerstörten Minen
K	Konstante mit dem Wert 12
X,Y	Schußkoordinaten
X1,Y1	Alte Schußkoordinaten (um alten Schuß auf dem Bildschirm zu löschen)
A,B	Koordinaten der Treibmine
A1,B1	Alte Koordinaten der Treibmine

Zunächst zeichnet der SPECTRUM das Koordinatiensystem von 12 x 12 Feldern. Zahlen über 9 sind durch Buchstaben (A,B und C) wiedergegeben.
Mit sehr viel Glück (wenn die Treibmine bei 12,12 aufgetaucht ist) wird gleich ein Treffer gemeldet. Ansonsten erscheint auf dem Bildschirm die Treibmine als schwarzes Quadrat. Diese Treibmine bewegt sich, durch Zufall gesteuert, auf die beiden Linien des Koordinatensystems zu und zwar zwischen 0 und 2 senkrecht und 0 und 2 waagrecht. Es gibt also 9 mögliche Felder, auf denen die Treibmine beim nächsten Mal auftauchen kann.

Es geht nun darum, dieses Feld zu erraten, auf dem die Mine das nächste Mal ist. Der SPECTRUM erwartet zunächst die X-, dann die Y-Koordinate! Geben Sie den Punkt ein, auf dem die Mine beim

nächsten Mal ist. Die Chance ist 1:9!

Hat man die Mine getroffen, meldet der SPECTRUM "TREFFER". Durch Eingabe von ENTER beginnt das Spiel erneut mit der 2. Teibmene . . .

Hat man nicht getroffen, zeigt ein gelber Stern den eigenen Schuß und ein roter Kreis den neuen Standort der Mine. Der nächste Versuch !

Kommt die Mine nun an eine der beiden Null-Linien, so ist die Abwehrstation zerstört und der SPECTRUM meldet die Anzahl der bereits zerstörten Minen.
Wer erreicht die höchste Anzahl, bevor er selbst getroffen wird?

2.5 Programm 5: "MiniSector"

SECTOR ist der Name eines elektronischen Spieles, bei dem es darauf ankommt, U-Boote in einem Ozean zu suchen, wobei sich die U-Boote während des Spieles fortbewegen.

2.5.1 Minisector I

```
  10 RESTORE : RANDOMIZE : BORDE
R 0: PAPER 0: INK 7: CLS
  20 LET x=INT (RND*14+5)
  30 LET y=INT (RND*14+5)
  40 LET a=INT (RND*4)
  50 LET q=0
  70 LET x2=0
  80 LET y2=0
  90 GO TO 500
 100 IF a=0 THEN LET x=x+1
 110 IF a=1 THEN LET x=x-1
 120 IF a=2 THEN LET y=y+1
 130 IF a=3 THEN LET y=y-1
 140 IF x=25 OR y=25 THEN LET a=
a+2
 150 IF x=0 OR y=0 THEN LET a=a-
2
 160 LET q=q+1
 200 INPUT "X=";x1,"y=";y1
 220 IF ABS (x2-x1)>4 THEN GO TO
 200
 250 IF ABS (y2-y1)>4 THEN GO TO
 200
 260 LET x2=x1: LET y2=y1
 270 IF x1<>x OR y1<>y THEN GO T
O 500
```

```
 275 FOR i=1 TO 10: BEEP .1,i: N
EXT i
 280 BORDER 6: BORDER 0: PRINT '
''' FLASH 1; INK 6;"TREFFER"''',
FLASH 0;"nach ";q;" Zuegen!"
 290 INPUT "Noch ein Spiel? (j/n
) ";q$
 300 IF q$="j" THEN RUN
 310 STOP
 500 BEEP .1,1: CLS
 510 PRINT "Standort: ";x2;",";y
2''"Entfernung: ";ABS (y2-y)*(ABS
 (y2-y)>=ABS (x2-x))+ABS (x2-x)*
(ABS (x2-x)>ABS (y2-y))
 520 GO TO 100
```

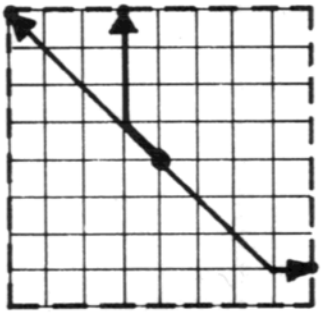
Variablen:

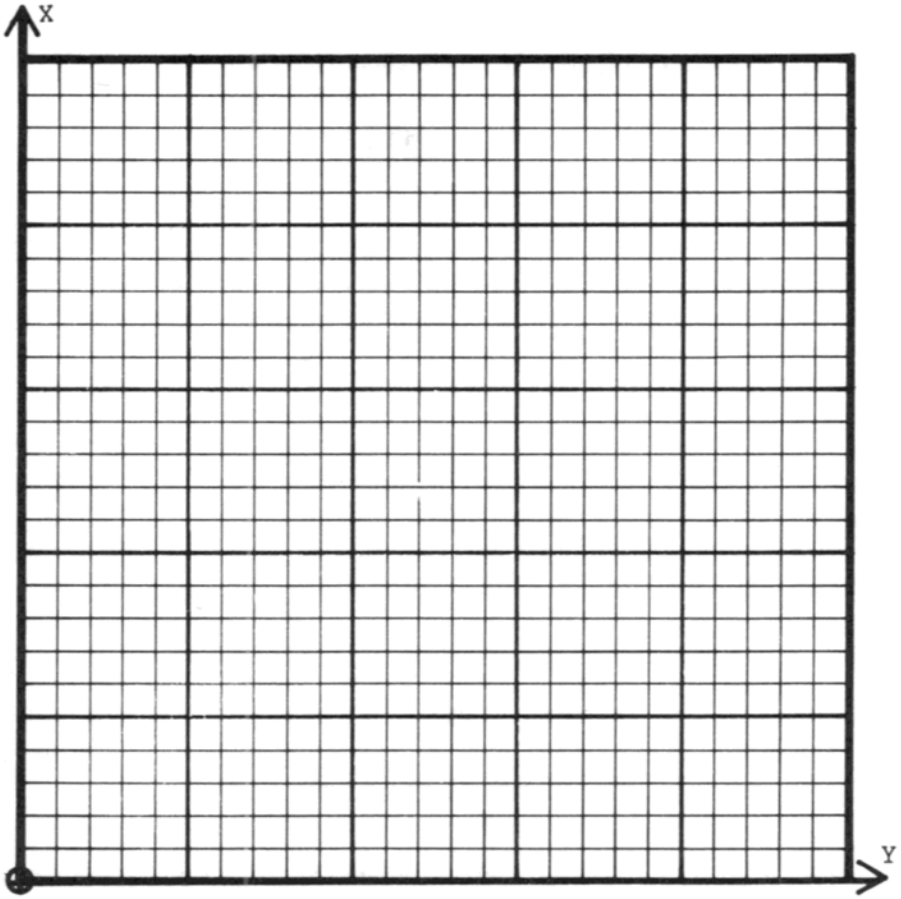
Q	Anzahl der Züge
X,Y	Standort des U-Bootes
A	Richtung des U-Bootes
X2,Y2	Standort des Zerstörers
X1,Y1	Eingaben zum neuen Standort des Zerstörers

Nach RUN zeigt der SPECTRUM den Standort und die Entfernung zum U-Boot an. Bei der Angabe des eigenen Standortes bedeutet die erste Zahl die X-Koordinate, die zweite Zahl die Y-Koordinate. Gespielt wird in einem 25 x 25 großem Feld (siehe nächste Seite).

Es empfiehlt sich, mehrere Kopien des Spielfeldes anzufertigen.

Die Entfernung 4 bedeutet, daß das U-Boot 4 Kästchen in irgendeiner Richtung entfernt ist. Folgendes Bild zeigt alle Punkte, die 4 Felder vom Zerstörer entfernt sind (das gestrichelte Quadrat):





25 x 25 großes Feld

Es geht nun darum, mit seinem Zerstörer exakt über das U-Boot zu fahren, um dann eine Bombe fallen zu lassen.

Aber aufgepaßt: Während man mit dem Zerstörer fährt, bewegt sich auch das U-Boot um 1 Feld in eine der vier Himmelsrichtungen (nicht diagonal).

Es ist möglich, mit dem Zerstörer in einer Runde maximal die Entfernung 4 zurückzulegen. Andere Eingaben werden nicht akzeptiert. Zuerst wird die neue X-Koordinate und dann die neue Y-Koordinate

des Zerstörers eingegeben. Danach zeigt der SPECTRUM den neuen eigenen Standort und die neue Entfernung zum U-Boot an.

Sehr bald wird man Position und Bewegungsrichtung des U-Bootes herausgefunden haben. Dann kommt es nur noch darauf an, sich genau über das U-Boot zu stellen. Dabei nicht vergessen: Während der eigenen Bewegung bewegt sich auch das U-Boot. Hat man das U-Boot getroffen, zeigt der SPECTRUM "Treffer" und die Anzahl der benötigten Züge an. Durch Eingabe von NEWLINE wird das Spiel neu gestartet. Jede andere Eingabe stoppt das Spiel.

Ein Hinweis:
Zu Beginn ist der eigene Standort immer 0;0!

Das U-Boot wird übrigens "reflektiert", wenn es an die Grenzen der Spielfläche kommt (Zeilen 140 und 150 im Listing).

Wer hier zu rasch die U-Boote findet, dem sei folgende Version empfohlen, die etwas schwieriger ist:

2.5.2 Minisector II

```
  10 RESTORE : RANDOMIZE : BORDE
R 0: PAPER 0: INK 7: CLS
  15 DEF FN a()=INT (RND*14+5)
  20 LET x=FN a()
  30 LET y=FN a()
  40 LET a=INT (RND*8)
  50 LET q=0
  70 LET x2=FN a()
  80 LET y2=FN a()
  90 GO TO 500
 100 LET x=x+(a=0 OR a=1 OR a=2)
-(a=3 OR a=4 OR a=5)
 110 LET y=y+(a=1 OR a=4 OR a=6)
-(a=2 OR a=5 OR a=7)
 120 LET a=a+3*(x=25)+(y=25)-3*(
x=0)-(y=0)
 160 LET q=q+1
 200 INPUT "X=";x1,"y=";y1
 220 IF ABS (x2-x1)>4 THEN GO TO
 200
 250 IF ABS (y2-y1)>4 THEN GO TO
 200
 260 LET x2=x1: LET y2=y1
 270 IF x1<>x OR y1<>y THEN GO T
O 500
```

```
 275 FOR i=1 TO 10: BEEP .1,i: N
EXT i
 280 BORDER 6: BORDER 0: PRINT '
''' FLASH 1; INK 6;"TREFFER"''',
FLASH 0;"nach ";q;" Zuegen!"
 290 INPUT "Noch ein Spiel? (j/n
) ";q$
 300 IF q$="j" THEN RUN
 310 STOP
 500 BEEP .1,1: CLS
 510 PRINT "Standort: ";x2;";";y
2''"Entfernung: ";ABS (y2-y)*(ABS
 (y2-y)>=ABS (x2-x))+ABS (x2-x)*
(ABS (x2-x)>ABS (y2-y))
 520 GO TO 100
```

Variablen:

(siehe Minisector I, zusätzlich:)

FNa() Da der Ausdruck INT(RND*14)+5 viermal vorkommt, ist es günstiger, ihn in einer Funktion zu speichern. FNa() rechnet diesen Ausdruck dann jedesmal neu aus.

Zu den Zeilen 100 bis 120 vergleiche das 1. Kapitel.

Zum Spiel:
Minisector II wird genauso gespielt wie Minisector I. Der Schwierigkeitsgrad ist nur erhöht worden:

1. Das U-Boot kann sich in 8 Richtungen (also auch diagonal) bewegen. An den Spielfeldgrenzen wird es reflektiert.
2. Der Ausgangspunkt des Zerstörers wird per Zufall festgelegt. Er liegt zwischen 5;5 und 20;20.

2.6 Programm 6: "Ari"



U(0)	Winkel	Entfernung
>12	>5	3076

```
  10 BORDER 2: PAPER 6: INK 0: C
LS
  20 PRINT TAB 8;"▚▚ A R I ▚▚"
;TAB 14;"▀"
  30 PRINT TAB 11;"▛▀▀▀▀▀▜";TAB
11;"▌     ▐";TAB 11;"▙▄▄▄▄▄▟"
  40 PRINT '"V(0)";TAB 7;"Winkel
","Entfernung": PLOT 0,116: DRAW
 255,0
 100 LET q=1
 110 LET x=INT (RND*15000+50)
 120 PRINT AT 3,14; INK 4;"■";AT
 3,25; INK 0;">";q;"<"
 200 PRINT AT 8,0; FLASH 1;"?";
 210 INPUT v
 220 IF v<10 THEN GO TO 200
 230 PRINT CHR$ 8;">";v;TAB 7; F
LASH 1;"?";
 240 INPUT w
 250 IF w<3 OR w>89 THEN GO TO 2
40
 260 PRINT CHR$ 8;">";w;TAB 16;
 270 LET z=INT (v↑2*SIN (w/90*PI
);/9.81)
 280 LET e=x-z
 290 PRINT e;"      "
 300 IF z=x THEN GO TO 500
 310 LET q=q+1
 320 GO TO 120
 500 PRINT AT 3,14; INK 4;"#": P
AUSE 20: BEEP .1,10: PRINT AT 3,
14;" "
 510 INPUT "Noch einmal (j/n) ";
a$
 520 IF a$="j" THEN RUN
```

Variablen:

Q	abgegebene Schüsse
X	Entfernung zum Ziel
V	Geschoßgeschwindigkeit
W	Abschußwinkel
Z	geschossene Entfernung
E	Differenz zwischen wirklicher und geschlossener Entfernung

Sie sind der Geschützführer. Das Ziel ist das grüne Quadrat im schwarzen Rahmen. Daneben wird die Anzahl der Schüsse angegegeben, die sie verbraucht haben. Es gilt nun Anfangsgeschwindigkeit und Abschußwinkel des Geschosses einzugeben. Das blinkende Fragezeichen zeigt an, welche Eingabe (bis auf zwei Stellen hinter dem Dezimalpunkt genau) der SPECTRUM erwartet.

Sofort nimmt die Elektronik die Auswertung vor und zeigt an, wie weit Sie daneben geschossen haben. Negative Zahlen bedeuten, daß sie zu weit, positive daß Sie zu nah geschossen haben. Ändern Sie Winkel und/oder Geschwindigkeit!

Die Geschwindigkeit wird in Metern pro Sekunde, der Winkel in Grad eingegeben und die Entfernung wird in Metern angezeigt. Der SPECTRUM rechnet mit der korrekten Erdbeschleunigung von 9.81 m/sec^2, die ballistische Formel lautet: $v^2 \sin 2\alpha$ / g, wobei α der Winkel in Grad und g die Erdbeschleunigung ist. (Der SPECTRUM rechnet im Bogenmaß, daher die entsprechend veränderte Formel.) Die Entfernung zum Ziel liegt zwischen 50 und 14999 Metern. Geschwindigkeiten unter 100 m/sec werden ebensowenig akzeptiert wie Winkel unter 8 oder über 80 Grad (Zeilen 70 und 110).

2.7 Programm 7: "Ballistik"

```
  10 BORDER 0: PAPER 7: INK 0: C
LS
  20 PRINT "▝BALLISTIK▘";AT 20,0
; INK 2;"▀"
  30 LET q=0
  40 LET a=INT (RND*26)+5
  50 PRINT AT 20,a; INK 3;"▀"
  60 INPUT AT 0,0;"Abwurfgeschwi
ndigkeit=";g''"Winkel (in Grad)="
;p
  70 IF g<100 THEN GO TO 60
  80 LET v=g/5
 110 IF p<8 OR p>80 THEN GO TO 6
0
 120 LET w=p/180*PI
 140 LET q=q+1
 150 LET z=TAN w
 160 LET c=5/(v*COS w)↑2
 170 FOR x=10 TO 250 STEP 4
 180 LET y=(x*z-c*x↑2)
 190 IF x>10 AND y<160 THEN PLOT
 OVER 1;PEEK 23677,PEEK 23678
 200 IF y<160 THEN IF POINT (x,y
+8)=1 THEN GO TO 800
 210 IF y<4 AND x>8 THEN GO TO 5
0
 220 IF y<160 THEN PLOT x,y+8
 230 NEXT x
 240 GO TO 50
 800 PRINT AT 20,a; FLASH 1;"#";
AT 21,0; INK 1;"TREFFER"; FLASH
0;" nach ";q;" Versuchen!"
 810 INPUT "Noch einmal (j/n)? "
;q$
 820 IF q$="j" THEN RUN
```

Variablen:

Q	Anzahl der Schüsse
A	Entfernung zum Ziel
G	Eingabe der Geschwindigkeit
V	Die Umrechnung der Geschwindigkeit auf Bildschirmformat
P	Der Winkel in Grad
W	Der Winkel im Bogenmaß
Z;C	Teilergebnisse der Formel, die die Höhe des Geschosses über dem Boden errechnet. Dadurch daß diese Berechnungen aus der FOR-NEXT-Schleife herausgenommen wurden, fliegt die Kugel schneller.
Y	Die Höhe der Kugel in Abhänigkeit von X
X	Die jeweilige Flugweite

Hier wird nicht die Schußweite, sondern die jeweilige Höhe des Geschosses über dem Boden errechnet. Die Schußweite ergibt sich automatisch: Es ist diejenige Weite, bei der das Geschoß wieder den Boden erreicht.

Übrigens:
In PEEK 23677 und PEEK 23678 sind immer die Werte der letzten PLOT-Position enthalten. Der SPECTRUM merkt sich also, wo er zuletzt hingePLOTet hat!

Der Computer erwartet auch die Eingaben in m/sec und Winkelgrad. Nach der Eingabe sieht man auf dem Bildschirm die Geschoßbahn. Wenn man nicht trifft, erwartet der SPECTRUM neue Eingaben. Trifft man, zeigt er "TREFFER" und die benötigten Schüsse. Durch ENTER beginnt dann das Spiel von vorn.

Änderungsvorschlag:
Wollen Sie die Geschoßbahn nicht als fliegende Kugel, sondern als komplette Kurve sehen (siehe Bild), dann nehmen Sie bitte folgende Änderungen vor:



170 FOR X=10 TO 250 (ohne STEP!)
190 löschen
200 IF y<8 THEN IF POINT (x,y+8) THEN IF POINT (x+1,y+8) THEN GO TO 800

Jetzt eignet sich das Programm einwandfrei, um die Abhängigkeit von Winkel und Entfernung zu demonstrieren !

2.8 Programm 8: "Spielautomat"




```
10 RANDOMIZE : GO SUB 1000
20 LET g=50
30 PRINT AT 3,3;"  ";g;" £"
40 DIM w(3)
50 FOR j=1 TO 3: LET w(j)=-16
```

```
  60 NEXT j
  70 GO SUB 500
  80 PRINT '"[redacted]": INPUT q$
  85 IF q$="n" OR g<10 THEN GO T
O 700
  90 LET g=g-10: PRINT AT 3,3;"[redacted]
";g;" £ "
 100 FOR j=1 TO 3
 110 LET w(j)=INT (RND*6)
 120 GO SUB 500
 130 PRINT '"[redacted]"
 140 INPUT x$
 150 IF x$="" THEN GO TO 200
 160 IF x$="a" THEN LET w(j)=INT
 (RND*6)
 170 GO SUB 500
 180 PRINT '"[redacted]"
 190 INPUT a$
 200 NEXT j
 210 FOR j=1 TO 3
 220 LET g=g+w(j)
 230 NEXT j
 240 IF w(1)=w(2) AND w(2)=w(3)
THEN LET g=g+10: FOR i=1 TO 45:
BEEP .02,i-25: NEXT i
 250 GO TO 30
 500 PRINT AT 6,0;"[redacted]"
 510 FOR k=1 TO 3: LET c=w(k)+1:
 IF c<0 THEN LET c=0
 520 PRINT "[redacted]"; INK c; PAPER 9;C
HR$ (w(k)+144); INK 0; PAPER 7;"
[redacted] ";
 530 NEXT k
 540 PRINT '"[redacted]"
 550 RETURN
 700 PRINT AT 0,0;"Auszahlung: "
;g;" £!"
 710 INPUT "Noch einmal (j/n)? "
;q$
 720 IF q$="j" THEN RUN
 730 STOP
1000 BORDER 7: PAPER 7: INK 0: C
LS : RESTORE
1010 FOR i=0 TO 47: READ a
1020 POKE USR "a"+i,a: NEXT i
1025 PRINT AT 12,0;
1030 FOR i=1 TO 6: PRINT INK i;
PAPER 9;CHR$ (143+i); INK 0; PAP
ER 7;" = ";i-1;" £": NEXT i
1035 PRINT '"Drei"'"Gleiche: +10
 £": RETURN
1040 DATA 255,129,153,165,165,15
3,129,255
1050 DATA 4,8,60,110,126,126,126
,60
1060 DATA 2,4,12,20,100,110,14,1
4
1070 DATA 16,214,254,254,124,56,
16,0
1080 DATA 0,60,126,126,126,126,6
0,0
1090 DATA 7,30,55,112,96,224,192
,128
```

Variablen:

G noch vorhandener Geldbetrag (in £)
W(3) Die Anzeige der drei Fenster des Automaten (je 0 bis 5)

Ihr Startkapital ist 50 £. Jedes Spiel kostet 10 £.

Zu Beginn zeigt der SPECTRUM Ihr Kapital und die leeren Fenster des Automaten. Darunter die Frage "10£?)".

Wollen Sie aufhören zu Spielen und sich den Betrag auszahlen lassen, geben Sie "n" (für Nein) ein. Jede andere Eingabe wertet der SPECTRUM als Ja. Haben Sie nicht mehr genug Geld für ein Spiel, zahlt Ihnen der SPECTRUM den verbleibenden Rest aus und das Spiel ist zu Ende. Können Sie aber noch spielen, zeigt der SPECTRUM nun im ersten Fenster eine Frucht.

Unter dem Fenster zeigt er "a/E.". Wenn Sie die erschienene Frucht ändern wollen, geben Sie "a" ein, ansonsten drücken Sie nur ENTER.

Haben Sie ENTER eingegeben, so öffnet sich das nächste Fenster. Hatten sie "a" eingegeben ändert sich das schon geöffnete Fenster. Das nächste wird durch ein weiteres ENTER geöffnet.

Ebenso geht es beim zweiten und dritten Fenster. Nachdem auch im dritten Fenster des Automaten eine Frucht steht, wird das gewonnene Geld hinzugezählt. Ist es mehr, als die Runde gekostet hat?

Gewinnplan:
Drei gleiche Zahlen ergeben den 13fachen Wert der Zahl in £. Höchstgewinn 5–5–5 (13*5= 65 £).
Ansonsten zählt nur die einfache Summe aller angezeigten Zahlen.

Nun zeigt der SPECTRUM Ihr neues Guthaben. Die Fenster werden wieder weiß und er fragt nach einer weiteren Runde.

2.9 Programm 9: "Irrgarten"



```
  10 BORDER 1: CLS
  20 RANDOMIZE
  30 RESTORE
  40 LET x=0
  50 LET y=0
  60 FOR i=0 TO 7: READ a
  70 POKE USR "o"+i,a: NEXT i
  80 DATA 24,60,24,60,90,24,36,5
6
 100 INK 2: FOR j=1 TO 200
 110 PRINT AT RND*21,RND*31;"#"
 120 NEXT j: INK 0
 130 PRINT AT 21,31; INK 1; PAPE
R 6; BRIGHT 1; FLASH 1;"$"
 140 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
 200 PRINT AT x,y;"Ø"
 210 LET b=y
 220 LET a=x
 230 LET x=x+(INKEY$="6" AND x<2
1)-(INKEY$="7" AND x>0)
 240 LET y=y+(INKEY$="8" AND y<3
1)-(INKEY$="5" AND y>0)
 250 PRINT AT a,b;" "
 260 IF INKEY$="q" THEN RUN
 270 LET p$=SCREEN$ (x,y)
 280 IF p$="$" THEN GO TO 400
 290 IF p$<>"#" THEN BEEP .02 AN
D INKEY$<>"",-10: GO TO 200
 300 FOR i=1 TO 10: BEEP .01,45:
 NEXT i
 310 LET x=a: LET y=b
 320 GO TO 200
 400 PRINT AT x,y;"Ø"
```

```
 410 LET s=65536*PEEK 23674+256*
PEEK 23673+PEEK 23672: LET s=s/5
0
 420 INPUT AT 0,0;("Geschafft in
 ";s;" Sekunden.")'"Noch einmal?
 (j/n) ";q$
 430 IF q$<>"n" THEN RUN
```

Variablen:

x,y Koordinaten des Mannes
a,b alte Koordinaten des Mannes, auf denen er bleibt, wenn er nicht weiter kann .
p$ Zeichen, auf das der Mann drauf laufen will.
s Zeit in Sekunden

Hinweis zum Programm:
Die Speicherzelllen 23672 bis 23674 enthalten einen Fernsehbildzähler. Je Fernschild, d. h. 50 mal pro Sekunde, wird dieser Zähler um 2 erhöht. In Zeile 140 wird dieser Zähler auf Null gesetzt. Da er drei bytes benutzt, vergehen bis zum Höchstwert etwa 335 544 Sekunden, das sind über 93 Stunden !

In Zeile 410 wird dieser Wert gelesen. Das Erbebnis ist zunächst die Anzahl der Fernsehbilder seit der Zeile 140. Diese Zahl durch 50 dividiert ergibt die Zeit in Sekunden. Dieser Timer ist im allgemeinen sehr genau. Eines ist zu beachten: Bei der Ansteuerung von Zusatzgeräten (z. B. Printer oder Mikrodrives), sowie bei BEEP, LOAD und SAVE geht Zeit verloren. Während exakter Zeitmessung sind derartige Befehle zu vermeiden. Bei unserem Spiel hingegen stört das Beepen nur wenig. Die exakte Zeit wäre jeweils etwas länger als die angegebene.

Ziel des Spieles ist es, das Männchen von oben links nach unten rechts zum blinkenden Dollar-Zeichen zu bewegen. Das Männchen hinterläßt eine Spur und macht Geh-Geräusche. Rennt es irgendwo dagegen, klirrt es. Sobald das Ziel erreicht ist, zeigt der SPECTRUM die Zeit an. (Ein Hinweis: recht selten kommt es vor, daß kein möglicher Weg existiert. Drücken Sie dann einfach auf "q", und das Spiel beginnt von vorne.)

Wenn Sie bereits die Bedienelemente nach Kapitel 1 gebaut haben,

ändern Sie für Bedienelement 2 bitte folgende Zeilen um (nur die unterrstrichenen Zeichen sind zu ändern):

230 LET x=x+(INKEY$="7" AND x<21)−(INKEY$="3" AND x>0)
240 LET y=y+(INKEY$="1" AND y<31)−(INKEY$="5" AND y>0)

2.10 Programm 10: "Todeshöhle"

```
  10 RANDOMIZE : BORDER 1: PAPER
 1: INK 7: CLS
  20 DIM q(2): DIM r(2): LET l=1
  LET e=0: LET p=29
  30 LET d=INT (RND*9): LET k=IN
T (RND*9)
  40 LET f=INT (RND*22+1)
  50 LET m=INT (RND*23+1)
  60 FOR n=1 TO 24
  70 LET x=m*n-1
  80 IF INT (x/p)*p=x THEN GO TO
 200
  90 NEXT n
 100 GO TO 50
 200 PRINT '"Spieler ";CHR$ (l+6
4);" ist in Hoehle ";q(l);"."'"I
n welche Richtung geht er?"
 210 LET r(l)=q(l)
 220 INPUT AT 0,0;"Vorwaerts Rei
twaerts Rueckwaerts";a$: IF a$<>
"v" AND a$<>"s" AND a$<>"r" THEN
 GO TO 220
 230 LET a=(q(l)*n AND a$="v")+(
q(l)*m AND a$="r")+(2+p-q(l) AND
 a$="s")
 240 LET a=a-INT (a/p)*p
 250 CLS
 260 IF a<24 THEN LET q(l)=a
 270 PRINT "Spieler ";CHR$ (l+64
);" kommt ";"rueckwaerts" AND a$
="r";"seitwaerts" AND a$="s";"vo
rwaerts" AND a$="v";"in ";: IF r
(l)=q(l) THEN PRINT "eine Sackga
sse"; INK 2;"■■■■■■"
 275 IF q(l)<>r(l) THEN PRINT "H
oehle ";q(l);"."
 280 IF q(l)<>23 THEN GO TO 320
 290 PRINT ' INK 6; BRIGHT 1;"Sc
hluessel Nr.1 ist "; FLASH 1;d:
PRINT '"Schluessel Nr.2 ist in h
oehle    Nr.";; FLASH 1;f; FLASH 0
;"."
 310 LET e=k
 320 IF q(l)<>f OR NOT e THEN GO
 TO 340
```

```
 330 PRINT ' INK 6; BRIGHT 1;"Sc
hluessel Nr.2 ist "; FLASH 1;e:
PRINT '"Gehen Sie nun zurueck in
 Hoehle Nr. 0!"
 340 IF q(l)<>0 OR NOT e THEN GO
 TO 400
 350 PRINT '"Nennen Sie die Summ
e beider      Schluessel: ";
 360 INPUT h: PRINT h
 370 IF h<>d+e THEN GO TO 400
 380 PRINT ' INK 6; BRIGHT 1; FL
ASH 1;"--- SIEG ---": FOR i=1 TO
 3: BEEP .2,-5: BEEP .8,0: NEXT
i
 385 INPUT "Noch ein Spiel (j/n)
? ";x$: IF x$<>"n" THEN RUN
 390 STOP
 400 LET l=3-l
 410 GO TO 200
```

Variablen:

Q(2)	Die neue Höhle beider Spieler
R(2)	Die alten Höhlen
E	Variable, in die der Wert des "2. Schlüssels" geladen wird, sobald der "1. Schlüssel" auftaucht. Dient auch als Flagge, welche anzeigt, ob der "1. Schlüssel" bereits gefunden wurde.
D	Wert des "1. Schlüssels"
K	Wert des "2. Schlüssels"
F	Nr. der Höhle, in der der "2. Schlüssel" versteckt ist.
M,N	Zwei Konstanten, die je Spiel neu gebildet werden (Zeilen 50 – 100) und das Höhlensystem errechnen.
A	Zwischenspeicher zur Errechnung einer neuen Höhle mittels M, N und P
P	Die Primzahl, die auf die maximale Höhlenzahl folgt. (In obigen Programm mit 23 Höhlen ist es die 29.)
L	Der jeweilige Spieler: 1 oder 2 (in der Anzeige A oder B)

Dies ist beinahe ein Mini-Adventure für zwei Spieler. Beide Spieler sind beim ersten Spiel zu Beginn in der Höhle 0. Ziel des Spieles ist es, in einem System von 24 Höhlen (0 – 23) zwei Schlüssel zu finden. Zunächst muß einer der beiden Spieler Schlüssel 1 finden (eine Zahl zwischen 0 und 9), der in Höhle 23 liegt. Dort erfährt man auch, in welcher Höhle Schlüssel 2 versteckt ist (auch ein Wert zwischen 0 und 9). Wohlgemerkt: Schlüssel 2 kann nur gefunden werden, wenn mindestens 1 Spieler in Höhle 23 beim ersten Schlüssel war! Hat man beide Schlüssel gefunden, kehrt man in Höhle 0 zurück, wo nach der

Summe beider Schlüssel gefragt wird. Gibt man Sie richtig an, ist man Sieger.

In jeder Runde zeigt der SPECTRUM pro Spieler zunächst den Standort an. Anschließend fragt er "Wohin?". Folgende Antworten sind erlaubt: "v" für Vorwärts, "r" für Rückwärts und "s" für Seitwärts.

Nach der Eingabe zeigt der SPECTRUM die Eingabe und wohin man damit gekommen ist.

Der nächste Spieler ist an der Reihe, sein Standort wird angegeben.

Kommt man nach Höhle 23 zeigt der SPECTRUM zusätzlich zum Standort: "Schlüssel Nr. 1 ist x. Schlüssel Nr. 2 ist in Höhle Nr. f". Das bedeutet, daß der Schlüssel 1 den Wert x hat und Schlüssel 2 in der Höhle f zu finden ist. In Höhle f wird dann, sobald einer hinfindet, der den Wert des Schlüssels 2 bekanntgegeben.

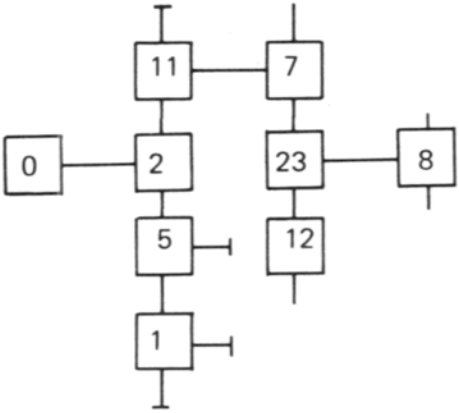
Antwortet man dann in der Höhle 0, wo man wieder hingefunden hat, auf die Frage mit der richtigen Summe beider Schlüssel ist das Spiel zu Ende. Ist die Antwort falsch, geht das Spiel weiter.

Ist das Spiel zu Ende, kann man durch ENTER die nächste Schlüsselsuche starten. Der siegreiche Spieler beginnt in Höhle 0, der andere an seinem letzten Standort.

Achtung:
Der SPECTRUM hat ein neues Höhlensystem erzeugt!

Wie findet man sich zurecht? Jede Höhle hat (meist) drei Ein-/Ausgänge: Vorwärts, Seitwärts und Rückwärts. Jede Höhle hat nur einen seitwärtigen Ausgang. Aus Höhle 0 kommt man nur seitwärts heraus in Höhle 2. (Umgekehrt kommt man von Höhle 2 seitwärts immer in Höhle 0). Die restlichen Höhlen werden je neu generiert.

Wenn man aus Höhle 8 seitwärts in 23 kommt, so kommt man umgekehrt auch seitwärts aus 23 in 8. oder ein anderes Beispiel: Wenn man aus 2 vorwärts nach 11 kommt, so kommt man aus 11 rückwärts nach 2. Am besten wird es sein, wenn Sie sich eine Karte zeichnen; etwa so:



Ohne Karte wird es Ihnen schwer fallen, jemals wieder nach Höhle 0 zu kommen.

Änderungsvorschläge:
Wollen Sie das Höhlensystem vergrößern, das Spiel länger gestalten? Ändern Sie folgende Zeilen um:

```
10 . . .: LET p=k
. . .
40 LET f=INT (RND*i)+1
50 LET m=INT (RND*h)+1
60 FOR n=1 TO j
. . .
260 IF a(j THEN LET q(l)=a
. . .
280 IF q(1)<>h THEN GO TO 320
```

Dabei muß h eine Primzahl sein und k die nächstgrößere Primzahl. Wenn h z. B. 43 sein soll (=43 Höhlen außer Höhle 0!), dann muß k den Wert 47 haben!
i = h−1 und j = h+1! Das Höhlensystem hat außer 0 dann h Höhlen und der Schlüssel 1 ist in Höhle h versteckt.

3 Unterprogramme in Maschinensprache

Dieses Kapitel soll keine Einführung in das Programmieren in Maschinensprache sein. Interessenten finden im Literaturverzeichnis Hinweise auf entsprechende Bücher, die für die Grundkenntnisse des Programmierens in Maschinensprache zu empfehlen sind.

Dieses Kapitel soll dazu dienen, fünf Maschinensprache-Unterprogramme vorzustellen. Der Leser wird mit Leichtigkeit diese Routinen auch in seinen Programmen verwenden können. Für den Gebrauch ist es nicht notwendig, Maschinensprache zu verstehen. Man muß nur wissen, wie mat mit den Unterprogrammen umgeht. Und genau das will das Kapitel zeigen.

3.1 Zur Plazierung von Unterprogrammen in Maschinensprache

Im Prinzip gibt es nur zwei brauchbare Plätze für Unterprogramme in Maschinensprache:

1. Jenseits von RAMTOP
2. Die erste REM-Zeile

Ein Maschinenprogramm hinter vorgezogenem RAMTOP hat den Vorteil, daß es gegen RUN oder gar NEW gesichert ist. Das heißt, es ist möglich, bei verschiedensten BASIC-Programmen die gleichen Routinen zu verwenden, ohne sie jeweils neu laden oder eingeben zu müssen.

Diese Sicherung der Programme bezahlt man damit, daß sie eigens geSAVEt und geLOADet werden müssen. Maschinenprogramme in einer REM-Zeile haben den Vorteil beim LOADen und SAVEn eines BASIC-Programmes einfach mit berücksichtigt zu werden.

Im folgenden zwei Maschinencode-Ladeprogramme, die es ermöglichen kürzere und längere Maschinencode Programme aus dem Hex-Code direkt in die erste REM-Zeile zu POKEn.

3.2 Zwei Programme, um Maschinencode zu laden

3.2.1 Programm 11: "Hex-Loader I"

```
  1 REM Soviele beliebige
        Zeichen, wie das
        Maschinenprogram
        Bytes lang ist.
  2 LET name=23760
 10 LET m$=" hier wird das ma-
    schinenprogram in hex-code
    hineingeschrieben."
 20 LET kontr=0: LET m$=m$+"*"
 30 LET a=name
 40 LET p=16*(CODE m$-48-(39*(m
$>"£"))+CODE m$(2)-48-(39*(m$(2
)>"£"))
 50 LET kontr=kontr+p: POKE a,p
 60 LET m$=m$(3 TO )
 70 IF m$="*" THEN GO TO 100
 80 LET a=a+1
 90 GO TO 40
100 PRINT "Kontrollzahl: ";kont
r
```

Es ist sehr wichtig, mindestens so viele Zeichen in der REM-Zeile zu haben, wie das Maschinenprogramm lang ist. Ein Byte wird durch zwei Zeichen zwischen 0 und f dargestellt; z. B. 2a oder 0f. Bei dieser Routine ist es wichtig, daß die Buchstaben klein geschrieben sind !

In Zeile 2 bekommt das Programm einen Namen, z. B. rsc für Right-Scroll. Mit diesem Namen läßt es sich dann später aufrufen; in unserem Beispiel durch RANDOMIZE USR rsc, bzw. LET x=USR rsc. Wir bevorzugen den Aufruf mit RANDOMIZE, da er keine Variablen benötigt. Wird jedoch mit der RND-Funktion gearbeitet, sollte IF USR rsc THEN verwendet werden. Ja, der SPECTRUM nimmt diese Zeile trotz des Syntax-Chechs an.

In Zeile 10 wird das Maschinenprogramm in M$ geschrieben. Es muß in Hex-Code mit kleinen Buchstaben sein. Die meisten Programmzeitschriften drucken Maschinenprogramme in Hex-Code. Die Methode ist einfacher, als jedes Byte einzeln in Data-Anweisungen mit Anführungszeichen und Kommas dazwischen zu schreiben.

Zeile 10 sieht dann z. B. so aus:

LET m$="2a0c4001b4. . ."

In Zeile 20 wird kontr, die Kontrollsumme auf 0 gesetzt und ein Stern an das Programm gehängt, der in Zeile 70 das Ende signalisiert.

Zeile 40 wandelt je zwei Zeichen, die ein Byte in Hex-Code darstellen, in einen Dezimalwert um, der in Zeile 50 zur Kontrollsumme addiert wird und in den Speicherplatz a gePOKEt wird. Die erste zu POKEnde Adresse wurde übrigens in den Zeilen 2 und 30 auf 23760, dem ersten Byte hinter dem REM der ersten Zeile, festgelegt.

Zeile 60 verkürzt m$ um die ersten zwei Zeichen, Zeile 70 überprüft, ob das Programm zu Ende ist und Zeile 80 erhöht a um 1; das nächste Byte kann bearbeitet werden. Zu guter Letzt wird die Kontrollsumme ausgedruckt. Zur Sicherheit wird das Ganze dann geSAVEt und danach (!) ausprobiert.

Übrigens:
Sollten Sie versuchen, die erste REM-Zeile zu LISTen, so wird das nicht gehen. LISTen Sie dann ab der nächsten Zeile; das ist ohne Probleme möglich. Absturzgefahr besteht nicht.

Ist das Programm in Ordnung, können die Zeilen 10 bis 100 gelöscht werden. Sie werden nicht mehr benötigt. Durch POKE 23756,0 setzt man nun die erste Zeilennummer auf 0. Sie kann nun nicht mehr EDITiert werden, was u. U. sie zerstören würde oder andere unangenehme Folgen haben kann.

Nun kann das fertige Maschinenprogramm auf Kassette gespeichert werden. Wenn man es nun benötigt, kann man es LOADen, bevor man ein Programm entwickelt oder mittels MERGE einem bestehenden BASIC-Programm anfügen.

3.2.2 Programm 12: "Hex-Loader II"

```
  1 REM Soviele beliebige
        Zeichen, wie das
        Maschinenprogramm
        Bytes lang ist.
 10 LET a=23760
 20 LET kontr=0
100 LET name1=a
110 LET m$="> hier wird ein
    teil des maschinenprogram-
    mes in hex-code hineinge-
    schrieben."
120 GO SUB 1000
130 LET name2=a
140 LET m$="> hier wird ein
    teil des maschinenprogram-
    mes in hex-code hineinge-
    schrieben."
150 GO SUB 1000
160 LET name3=a
170 LET m$="> hier wird ein
    teil des maschinenprogram-
    mes in hex-code hineinge-
    schrieben."
180 GO SUB 1000
500 LET ende=a-1
510 PRINT "name1=";name1
520 PRINT "name2=";name2
530 PRINT "name3=";name3
600 PRINT "ende =";ende
610 PRINT "kontr=";kontr
620 STOP
1000 LET m$=m$+"*"
1010 LET p=16*(CODE m$-48-(39*(m
$>"£"))+CODE m$(2)-48-(39*(m$(2
)>"£"))
1020 LET kontr=kontr+p: POKE a,p
1030 LET m$=m$(3 TO )
1040 LET a=a+1: IF m$="*" THEN R
ETURN
1050 GO TO 1010
```

Dieses Programm eignet sich besonders, wenn man längere Maschinenprogramme oder auch mehrere Einzelprogramme verwenden möchte.

Zum Programm selbst:
Zeile 10 setzt a auf das erste Byte hinter dem REM in der ersten Zeile; Zeile 20 setzt die Kontrollsumme auf 0.

Dann folgen für jeden Teil des Maschinenprogrammes drei Zeilen: die erste Zeile (z. B. 100) hält in einer Variablen den Anfang des Programmteiles fest. In der nächsten Zeile (z. B. 110) wird in M$ der

Code hineingeschrieben und in der letzten Zeile einer solchen Dreiergruppe wird ein Unterprogramm aufgerufen, welches in bekannter Weise M$ in Maschinencode verwandelt.

Nachdem alle Programmteile bearbeitet wurden, zeigt der SPECTRUM am Bildschirm die Anfänge der einzelnen Teile, das letzte Byte und die Kontrollsumme an. Mit dem fertigen Programm verfährt man, wie oben beschrieben.

Wenden wir uns nun den Maschinenprogrammen zu:

3.3 Programm 13: "Screen-Loader"

Dieses Programm dient dazu, den Bildschirm schnell mit einem bestimmten Zeichen zu füllen. Es kann ein beliebiges Rechteck des Bildschirmes gewählt werden!

Das Programm:

```
    1 REM 28 beliebige Zeichen ..
.....
    2 LET scr=23760
   10 LET m$="2100000616c506203e1
6d77cd77dd73e23d72c10f3242e00c11
0eac9"
   20 LET kontr=0: LET m$=m$+"*"
   30 LET a=scr
   40 LET p=16*(CODE m$-48-(39*(m
$>"£")))+CODE m$(2)-48-(39*(m$(2
)>"£"))
   50 LET kontr=kontr+p: POKE a,p
   60 LET m$=m$(3 TO )
   70 IF m$="*" THEN GO TO 100
   80 LET a=a+1
   90 GO TO 40
  100 PRINT "Kontrollzahl: ";kont
r


Kontrollzahl: 2615
```

In diesem Fall wird Hex-Loader I verwandt. Vor jedem Aufruf des Programmes muß im Programm bereits ein PRINT-Befehl vorgekommen sein. Nachdem obiges Programm geRUNt ist und die Kontrollzahl

stimmt, können Sie mittels "PRINT: RANDOMIZE USR scr" ausprobieren. Der Bildschirm müßte sich mit dem Zeichen "#" füllen.

Das folgende kleine Programm, zusätzlich zur REM-Zeile eingegeben, ermöglicht ein beliebiges Testen des Programmes. Zeile 15 ermöglicht, das beliebige Rechteck aus beliebigen Zeichen übereinander geprintet werden. Es lassen sich recht interessante Graphiken entwickeln.

Nun können Sie die REM-Zeile und Zeile 2 abspeichern, um sie bei Bedarf wieder zu verwenden. Die Bedeutung der Variablen ergeben sich aus dem Test-Programm:

```
   2>LET scr=23760: LET chr=2377
6: LET zz=23764: LET zi=23767: L
ET zx=23762: LET zy1=23761: LET
zy2=23763
  10 REM testprogra##
  15 OVER 1
  20 INPUT "Zeichen:";a$: POKE c
hr,CODE a$
  30 INPUT "Startpunkt: x=";x;"
y=";y: POKE zx,x: POKE zy1,y: PO
KE zy2,y
  40 INPUT "Format: Zeilen=";z;"
 Spalten=";s: POKE zz,z: POKE zi
,s
  50 PRINT AT 0,0: RANDOMIZE USR
 scr
  60 GO TO 20
```

3.4 Programm 14: "Inversion"

Dieses Programm invertiert blitzartig den ganzen Bildschirm oder einen Teil desselben. Es ist sehr gut geeignet, Action in Spiele zu bringen. Als Beispiele mögen MINENFELD und ZX-SECTOR in diesem Buch dienen.

Zunächst das Programm in Hex-Loader I:

```
   1 REM 19 beliebigeZeichen
   2 LET inv=23760
  10 LET m$="2100400618c506007ea
eff772310f9c110f3c9"
  20 LET kontr=0: LET m$=m$+"*"
  30 LET a=inv
  40 LET p=16*(CODE m$-48-(39*(m
$>"£"))+CODE m$(2)-48-(39*(m$(2
)>"£"))
  50 LET kontr=kontr+p: POKE a,p
```

```
  50 LET m$=m$(3 TO )
  70 IF m$="*" THEN GO TO 100
  80 LET a=a+1
  90 GO TO 40
 100 PRINT "Kontrollzahl: ";kont
```

```
Kontrollzahl: 2021
```

Starten Sie das Programm mit RUN und testen Sie es mit RANDOMIZE USR inv!

Geben Sie zum auführlichen Testen folgendes Testprogramm ein. In zz (=23764) können Sie verschiedene Zeilenzahlen hineinPOKEn. Allerdings sind nur 8, 16 und 24 sinnvoll, wie Sie bald merken werden.

```
   2>LET inv=23760: LET zz=inv+4
  10 REM invTest
  20 INPUT "Zeilenzahl: (8/16/24
) ";z: POKE zz,z: RANDOMIZE USR
inv: PAUSE 0
  30 FOR i=0 TO 7: RANDOMIZE USR
 inv: NEXT i: PAUSE 0: GO TO 20
```

Auch hier emfiehlt es sich die REM-Zeile und Zeile 2 abzuspeichern. Haben Sie die Zeilennummer der REM-Zeile auf 0 gesetzt?

3.5 Programme 15 + 16: "Vertical-Scroll

Eine ganze Reihe von Spielen benötigt die Möglichkeit auch die Vertikale SCROLLen zu können, d. h. von links nach rechts oder umgekehrt zu SCROLLen. Diese Aufgabe erledigen folgende zwei Programme:
"Right-Scroll" SCROLLt von links nach rechts und "Left-Scroll" in die Gegenrichtung.

Häufig werden beide Programme gemeinsam benötigt, so daß hier Hex-Loader II zur Anwendung kommt:

```
   1 REM 50 beliebige Zeichen ..
.........................
  10 LET a=23760
```

```
  20 LET kontr=0
 100 LET lsc=a
 110 LET m$="06c0110040d5e123c50
11f001aedb02b7700232313c110f0c9"
 120 GO SUB 1000
 130 LET rsc=a
 140 LET m$="06c011ff57d5e12bc50
11f001aedb82377002b2b1bc110f0c9"
 150 GO SUB 1000
 500 LET ende=a-1
 510 PRINT "lsc= ";lsc
 520 PRINT "rsc= ";rsc
 600 PRINT "ende =";ende
 610 PRINT "kontr=";kontr
 620 STOP
1000 LET m$=m$+"*"
1010 LET p=16*(CODE m$-48-(39*(m
$>"f"))+CODE m$(2)-48-(39*(m$(2
)>"f"))
1020 LET kontr=kontr+p: POKE a,p
1030 LET m$=m$(3 TO )
1040 LET a=a+1: IF m$="*" THEN R
ETURN
1050 GO TO 1010


lsc= 23760
rsc= 23785
ende =23809
kontr=4952
```

Das folgende kleine Programm zeigt die Möglichkeiten:

```
   2>LET lsc=23760: LET lm=lsc+1
6: LET rsc=23785: LET rm=rsc+16
  10 REM vscTest
  20 BORDER 2: PAPER 0: INK 6: C
LS
  30 FOR i=0 TO 31
  40 FOR j=21 TO RND*20 STEP -1
  50 PRINT AT j,i;"#": NEXT j: N
EXT i
  60 IF INKEY$="5" THEN RANDOMIZ
E USR lsc
  70 IF INKEY$="8" THEN RANDOMIZ
E USR rsc
  80 IF INKEY$="6" THEN GO SUB 1
00
  85 IF INKEY$="7" THEN GO SUB 1
10
  90 GO TO 60
 100 POKE rm,54: POKE lm,54: RET
URN
 110 POKE rm,119: POKE lm,119: R
ETURN
```

In die Adressen rm und lm darf nur (!) 119 oder 54 gePOKEt werden. Ist der Wert 54, so wird in der entsprechenden Richtung beim Scrollen eine Leerzeile eingefügt, ist er 119 wird die am anderen Ende verschwindende Zeile wieder angefügt. Beim Testprogramm erreichen Sie den Wert 54 durch Drücken auf "6", 119 durch drücken auf "7". Mit den Tasten "5" und "8" Scrollen Sie.

Auch hier empfiehlt es sich, die REM-Zeile und Zeile 2 zu SAVEn.

3.6 Programm 17: "Memory-Mapped-Display" (48K)

Dieses Maschinenprogramm hat mehrere Besonderheiten: Zunächst zeigt es, wie man Hex-Loader II benutzen kann, um in beliebige Speicherbereiche zu POKEn.

Zum anderen ist es ein Programm, daß nur einmal vom Programmierer aufgerufen wird, um dann vom SPECTRUM andauernd aufgerufen zu werden. Normalerweise produziert der SPECTRUM 50mal in der Sekunde eine Unterbrechung, in der die Rechnerei gestoppt wird.

Wenn dieses Programm gestartet ist, führt er jedesmal zusätzlich ein Maschinenprogramm aus, wodurch der SPECTRUM allerdings etwas langsamer wird. Alles braucht seine Zeit.

In diesem Programm wird jeder Interrupt ausgenutzt den Inhalt der Speicherzellen 64000 bis 64703 auf den Bildschirm zu bringen. Während es kaum möglich ist, in den Original-Bildschirmspicher zu POKEn, ist dies nun möglich. Beliebige Zeichen können direkt in den Bildschirm gePOKEt werden.

Außerdem ist es möglich, zwischen zwei verschiedenen Modi umzuschalten:

Wird in Adresse 63570 der Wert 182 hineingePOKEt, werden der normale Bildspeicher und der neue Bildspeicher gleichzeitig angezeigt, übereinander geschrieben. In diesem Modus kann man BASIC-Programme listen oder mit normalen BASIC-Befehlen Text und Zeichen auf den Bildschirm bringen.

Wird aber der Wert 183 in Adresse 63570 hineingePOKEt, wird ausschließlich der neue Bildspeicher zur Anzeige gebracht. Lediglich die beiden untersten Zeilen (Zeile 22 und 23) bleiben sichtbar.

```
MMD = 63479
MMD-Ende = 63580
Interrupt = 65040
Ende = 65062
Kontrollzahl = 16417
```

```
  10 CLEAR 63400
  20 PRINT "Einen Moment bitte .
."
  30 LET kontr=0
 100 LET a=63479: LET mmd=a
 110 LET m$="dde5f5c5d5e52100fa1
1004006600e5c5cd42f8c1e12310f6dd2
10007dd19dde5d10600e5c5cd42f8c1e
12310f6dd210007dd19dde5d106c0e5c
5cd42f8c1e12310f6e1d1c1f1dde1c33
800d5ed4b365c6e26002929290906087
eebb6eb12231410f7d113c9"
 120 GO SUB 1000
 125 LET mmdend=a
 130 LET a=65040: LET interrupt=
a
 140 LET m$="f321f8fc01100136f72
30b78b120f83efded47ed5efbc9"
 150 GO SUB 1000
 200 FOR i=64000 TO 64703: POKE
i,32: NEXT i
 500 CLS : LET ende=a-1
 510 PRINT "MMD = ";mmd
 515 PRINT "MMD-Ende = ";mmdend
 520 PRINT "Interrupt = ";interr
upt
 600 PRINT "Ende = ";ende
 610 PRINT "Kontrollzahl = ";kon
tr
 620 STOP
1000 LET m$=m$+"*"
1010 LET p=16*(CODE m$-48-(39*(m
$>"£"))+CODE m$(2)-48-(39*(m$(2
)>"£"))
1020 LET kontr=kontr+p: POKE a,p
1030 LET m$=m$(3 TO )
1040 LET a=a+1: IF m$="*" THEN R
ETURN
1050 GO TO 1010
9999 LET mode=mmd+91: CLS : POKE
 23561,5: POKE 23562,1: RANDOMIZ
E USR interrupt: POKE 64000,CODE
 "O": POKE 64001,CODE "K"
```

Übrigens:
Durch GOTO 9999 schaltet man das neue Display ein. Ausschalten ist nicht möglich, außer durch NEW. RANDOMIZE USR 65040 schaltet es dann wieder ein. In 23561 und 23562 sind die Zeiten für die Repeat-Funktion gespeichert. Diese werden verändert, um auch beim verlangsamten SPECTRUM, die Tasten normal bedienen zu können.

Versuchen Sie doch einmal folgenden "Einzeiler":

```
FOR i=64000 TO 64703: POKE i,35: NEXT i: STOP
```

Versuchen Sie doch einmal, die Raupe aus Kapitel 1 für den neuen Bildschirmspeicher umzuschreiben! Interessant ist diese Programm sicher auf für jene, die in Maschinencode Bildschirmspiele erzeugen wollen. Maschinencode ist ja schnell genug, um das verlangsamte Denktempo des SPECTRUM zu verkraften.

NOTIZEN

4 Superprogramme für den ZX SPECTRUM

4.1 Programm 18: "LP-Register" (48K)

```
    1 BORDER 7: PAPER 7: INK 2: C
LS : LET v=0
    2 PRINT "Wieviele LP's wollen
 Sie          speichern? (max. 100
0 LP's) ";
    3 INPUT z: IF z<1 OR z>1000 T
HEN GO TO 3
    4 PRINT z
    5 DIM a$(z,10)
    6 DIM b$(z,16)
    7 DIM c$(z,1)
    8 PRINT "Name des Registers?"
    9 INPUT n$
   10 LET n=1
   15 BORDER 6: PAPER 6: INK 0: C
LS
   20 PRINT "LP-REGISTER: ";n$
   40 PRINT '''"1 Eingabe"''"2 Int
erpret?"''"3 Titel?"''"4 Code?"'
'"5 Liste (TV)"''"6 Liste (Print
er)"''"7 Loeschen einer LP"''"8
Speichern auf Kasette"
   50 PAUSE 0: LET q$=INKEY$
   53 IF q$<"1" OR q$>"8" THEN GO
 TO 50
   60 LET a=VAL q$
   70 GO TO 100*a
  100 FOR s=n TO z
  110 IF a$(s,1)=" " THEN GO TO 1
30
  120 NEXT s
  125 GO TO 15
  130 POKE 23692,255: PRINT '''"LP
-Nr. ";s;
  140 PRINT "Interpret?",
  145 INPUT AT 1,0;"_____";A
T 1,0; LINE a$(s)
  147 IF a$(s,1)=" " THEN GO TO 1
5
  150 PRINT a$(s)''"Titel?",
  155 INPUT AT 1,0;"_____
____";AT 1,0; LINE b$(s)
```

```
 160 PRINT b$(s)'"Code?",
 165 INPUT AT 1,0;"_";AT 1,0; LI
NE c$(s)
 170 IF s>v THEN LET v=s
 175 LET n=s: GO TO 100
 200 DIM x$(10)
 210 INPUT AT 0,0;"Interpret?";A
T 1,0;"__________";AT 1,0; LINE
x$: CLS
 220 FOR x=1 TO v
 230 IF x$=a$(x) THEN GO SUB 100
0
 240 NEXT x
 250 GO TO 2000
 300 DIM y$(16)
 310 INPUT AT 0,0;"Titel";AT 1,0
;"________________";AT 1,0; LINE
y$
 320 CLS : FOR x=1 TO v
 330 IF y$=b$(x) THEN GO SUB 100
0
 340 NEXT x
 350 GO TO 2000
 400 CLS : PRINT "Code?"
 410 PRINT ''"B Bildplatte"''"D
Digitalaufnahme"''"F Farbiges Ma
terial"''"J Japan-Pressung"''"K
Kasette"''"L Live"''"R Raubpress
ung"
 420 DIM d$(1): INPUT AT 1,0;"_"
;AT 1,0; LINE d$
 430 CLS : FOR x=1 TO v
 440 IF d$=c$(x) THEN GO SUB 100
0
 450 NEXT x: GO TO 2000
 500 INPUT "Ab welcher LP-Nr.? "
;Nr
 505 IF Nr<1 OR Nr>v THEN GO TO
500
 510 CLS : FOR x=Nr TO v
 515 IF a$(x,1)<>" " THEN GO SUB
 1000
 520 IF INKEY$="a" THEN GO TO 15
 530 NEXT x: GO TO 2000
 600 INPUT "Ab welcher LP-Nr.? "
;Nr
 605 IF Nr<1 OR Nr>v THEN GO TO
600
 610 FOR x=Nr TO v
 615 IF a$(x,1)=" " THEN GO TO 6
30
 620 LPRINT "0" AND x<10;"0" AND
 x<100;x;" ";a$(x);":";b$(x); IN
VERSE 1;c$(x) AND c$(x)<>" "
 630 NEXT x: GO TO 50
 700 INPUT AT 0,0;"LP-Nr.? ";Nr
 710 IF a$(Nr,1)=" " OR Nr<1 OR
Nr>v THEN GO TO 700
 715 IF Nr<n THEN LET n=Nr
 720 INPUT AT 1,0;(a$(Nr);" ";b$
(Nr));"[illegible]"; LINE q$
 730 IF q$="n" THEN GO TO 700
```

```
 740 LET a$(Nr,1)="  "
 750 GO TO 50
 800 SAVE n$ LINE 15
1000 IF PEEK 23689=4 THEN GO SUB
 1050: CLS
1010 PRINT "0" AND x<10;"0" AND
x<100;x;" ";a$(x);":";b$(x); INV
ERSE 1;c$(x) AND c$(x)<>" "
1020 RETURN
1050 PAUSE 0
1060 LET q$=INKEY$
1070 IF q$="z" THEN COPY : GO TO
 1060
1080 RETURN
2000 INPUT "Ende "; LINE q$
2010 GO TO 15
```

Wichtige Variablen:

V	Anzahl der bereits gespeicherten LPs
N	Nr. der zuletzt eingegebenen, bzw. gelöschten LP.
Z	Anzahl der maximal zu speichernden LPs
A$(Z,10)	Interpret für Z LPs
B$(Z,16)	Titel
C$(Z,1)	Code

Mit obigem Programm können Sie maximal 1000 LPs speichern. Nachdem Sie das Programm eingegeben haben, speichern Sie es mehrmals auf Kassette, damit Sie jederzeit ein leeres LP-Register zur Hand haben.

Wird nun das unbeschriebene Register geLOADet, fragt der SPECTRUM nach der zu erwartenden max. LP-Zahl. (Je größer diese ist, desto länger dauert das LOADen und SAVEn.) Anschließend fragt er nach dem Namen des Registers, z. B. "Neue Deutsche Welle". Der Name sollte nicht mehr als 19 Zeichen haben. Der Benutzer hat nun folgende Möglichkeiten:

1. EINGABE

Der SPECTRUM sucht den ersten leeeren Speicherplatz und zeigt dessen NR an. Diese sollte zur Numerierung der LPs verwandt werden. Nacheinander fragt der SPECTRUM nun nach Interpret, Titel und Code der LP. Dann sucht er die nächste freie Speicherstelle . . .

Wenn man bei Interpret nur ENTER drückt, merkt der Spectrum, daß man nichts mehr eingeben will und kehrt zum Menue zurück.

2. INTERPRET?

Man gibt einen Interpreten ein und der SPECTRUM listet alle LPs mit Nr, Interpret, Titel und Code auf, die von diesem Interpreten sind. Ist der Bildschirm voll, werden durch Druck auf eine Taste die nächsten LPs angezeigt. Hat er alle LPs des Interpreten gefunden, zigt er "ENDE" und kehrt durch Drücken irgendeiner Taste zum Titelbild zurück.

3. TITEL?

Das gleiche wie unter "2", nur wird hier nach dem Titel gefragt.

4. CODE?

Noch einmal dasselbe. Jetzt wird nach dem Code gefragt. Die Nutzung des Codes in Zeile 410 ist nur ein Vorschlag.

5. LISTE, TV

Zunächst fragt der SPECTRUM, ab welcher LP-Nr. die Liste erstellt werden soll. Die LPs werden nach Nummern geordnet aufgelistet; immer 20 LPs auf einmal. Durch Drücken irgendeiner Taste kommen die nächsten 20 LPs zur Anzeige, durch Drücken auf "Z" wird eine Kopie auf den Printer gegeben.

6. LISTE, GEDRUCKT

Wenn der ZX-Printer angeschlossen ist, druckt er alle LPs ab der einzugebenden Anfangs-Nr. aus.

7. LOESCHEN

Die LP-Nr. der LP, die gelöscht werden soll, ist einzugeben. Der SPECTRUM nennt die LP und fragt "OK?". Bei jeder Eingabe außer "n" wird die LP gelöscht. Bei der nächsten Eingabe wird dieser Platz wieder beschrieben .

8. SPEICHERN AUF KASSETTE

Der SPECTRUM speichert das Register im gegenwärtigen Zustand ab. Beim nächsten LOADen des Registers wird direkt das Menue angezeigt.

Nachdem Sie das Programm eingegeben haben, speichern Sie das Programm mehrmals auf Kassette, um immer eine unbeschriebene Version zur Verfügung zu haben.

4.2 Programm 19: "Kartei" (48K)

```
    0>REM KARTEI
          (C) ROLAND G. HUELSMANN
    1 LET LETZTES=0
    5 CLS
   10 PRINT "WIEVIELE WOERTER PRO
 EINTRAG? ";
   15 INPUT W
   16 IF W<1 OR W>21 THEN GO TO 1
5
   20 PRINT W
   30 PRINT
   50 DIM A(W+1)
   52 DIM W$(W,15)
   54 LET X=0
   55 FOR I=1 TO W
   60 PRINT "AUS WIEVIEL ZEICHEN
SOLL DAS",I;". WORT BESTEHEN? ";
   62 INPUT A
   65 IF A<0 OR A>32 THEN GO TO 6
2
   66 PRINT A
   67 LET A(I+1)=X+A
   68 PRINT
   70 LET X=X+A
   71 PRINT "NAME DES ";I;". WORT
ES?"
   72 INPUT W$(I)
   73 PRINT W$(I)
   74 PRINT
   80 NEXT I
   82 LET EI=INT (36000/X)
   84 PRINT "SIE KOENNEN ";EI;" E
INTRAEGE","SPEICHERN, OK?"
   90 INPUT 0$
   95 IF 0$="Nein" OR 0$="nein" T
HEN GO TO 84
  100 DIM A$(EI,X)
  110 LET ZCH=X
  111 PRINT
  115 PRINT "WELCHEN NAMEN SOLL D
IE KARTEI   HABEN?"
  117 INPUT K$
  180 BORDER 1: PAPER 1: INK 7: C
LS
  190 PRINT "[redacted]"
  200 PRINT '"[redacted] NEUER EINTRAG"''
  210 PRINT "[redacted] KARTEI ALPHABETISC
H ORDNEN"''
  220 PRINT "[redacted] EINEN BESTIMMTEN E
INTRAG          SUCHEN"''
  225 PRINT "[redacted] EINEN EINTRAG LOES
CHEN"''
  230 PRINT "[redacted] INHALT IN JETZIGER
              ORDNUNG AUSDRUCKEN
"''
  240 PRINT "[redacted] GESAMTINHALT AUF D
EN BILD-        SCHIRM"''
  290 PRINT INK 6; BRIGHT 1;"[redacted] DI
E KARTEI SPEICHERN"
```

```
 295 PRINT AT 20,1; FLASH 1;"DRU
ECKEN SIE DIE ENTSPRECHENDE";AT
21,5;"ZAHL."
 300 PAUSE 0
 306 LET I$=INKEY$
 307 IF I$<"1" OR I$>"6" AND I$<
>"9" THEN GO TO 300
 310 GO TO 1000+VAL I$
1000 CLS
1010 FOR I=1 TO EI
1015 IF A$(I,1)=" " THEN GO TO 1
100
1020 NEXT I
1030 PRINT "DIE KARTEI IST VOLL.
"
1040 INPUT Q$
1050 GO TO 180
1100 PRINT I;". EINTRAG:"
1105 PRINT
1110 FOR J=1 TO W
1115 IF A(J)=A(J+1) THEN GO TO 1
145
1120 PRINT W$(J)
1130 INPUT A$(I,A(J)+1 TO A(J+1)
)
1132 PRINT A$(I,A(J)+1 TO A(J+1)
)
1133 PRINT
1135 IF A$(I,1)<>" " THEN GO TO
1140
1136 IF I-1>LETZTES THEN LET LET
ZTES=I-1
1137 IF I>1 AND A$(I)=A$(I-1) TH
EN LET A$(I,1)=" "
1138 GO TO 180
1145 NEXT J
1150 GO TO 1010
2000 GO SUB 8000
2010 PRINT AT 20,0;"WELCHES WORT
 DES EINTRAGES SOLL ALPHABETISCH
 GEORDNET SEIN?"
2020 INPUT WO
2025 IF WO>W OR WO<1 THEN GO TO
2020
2026 LET A=A(WO)+1
2030 DIM S(99): LET P=1: LET S(1
)=1: LET S(2)=LETZTES
2035 LET L=S(P): LET R=S(P+1): L
ET P=P-2
2040 LET I=L: LET J=R: LET D=(L+
R)/2: LET G$=A$(D,A TO )
2050 IF A$(I,A TO )>=G$ THEN GO
TO 2070
2060 LET I=I+1: GO TO 2100
2070 IF A$(J,A TO )<=G$ THEN GO
TO 2085
2080 LET J=J-1: GO TO 2100
2085 IF I>J THEN GO TO 2100
2090 LET H$=A$(I): LET A$(I)=A$(
J): LET A$(J)=H$: LET I=I+1: LET
 J=J-1
2100 IF I<=J THEN GO TO 2050
```

```
2110 IF I>=R THEN GO TO 2130
2120 LET P=P+2: LET S(P)=I: LET
S(P+1)=R
2130 LET R=J: IF L<R THEN GO TO
2040
2140 IF P<>-1 THEN GO TO 2035
2150 GO TO 180
3000 GO SUB 8000
3010 PRINT AT 20,0;"DAS WIEVIELT
E WORT EINES EIN-    TRAGES SUCHE
N SIE?"
3020 INPUT WO
3025 IF WO>W THEN GO TO 3020
3026 CLS
3028 PRINT W$(WO),,,
3029 LET A=A(WO)+1
3030 PRINT "GENAUIGKEIT? (ZEICHE
N/              0   = ALLE ZEICHEN)
";
3031 INPUT GEN
3032 IF GEN<0 OR GEN>32 THEN GO
TO 3031
3033 LET E=A(WO+1)
3034 IF GEN<>0 AND GEN<E-A+1 THE
N LET E=A+GEN-1
3035 PRINT E-A+1
3036 PRINT
3040 PRINT "WELCHES WORT SUCHEN
SIE?"
3045 DIM X$(1,ZCH)
3050 INPUT X$(1,A TO E)
3060 CLS
3100 FOR Z=1 TO LETZTES
3110 IF X$(1,A TO E)=A$(Z,A TO E
) THEN GO SUB 7100
3116 INPUT Q$
3117 IF Q$="Z" OR Q$="z" THEN CO
PY
3120 NEXT Z
3123 IF PEEK 23689=24 AND E-A>0
THEN GO TO 3200
3125 PRINT
3130 PRINT "ENDE"
3140 INPUT Q$
3145 IF Q$="Z" THEN COPY
3150 GO TO 180
3200 PRINT "SOLL ICH ETWAS UNGEN
AUER SUCHEN?"
3210 INPUT Q$
3220 IF Q$="Nein" OR Q$="nein" T
HEN GO TO 180
3225 CLS
3230 LET E=E-1
3240 GO TO 3100
3250 GO TO 3123
4000 CLS
4010 PRINT "WIE HEISST DAS ERSTE
 WORT DES ZULOESCHENDEN EINTRAGE
S?"
4020 DIM X$(1,A(2))
4030 INPUT X$(1,1 TO A(2))
4040 FOR Z=1 TO LETZTES
```

```
4050 IF X$(1,1 TO A(2))=A$(Z,1 T
O A(2)) THEN GO SUB 7100: INPUT
"ok? ";o$: IF o$="ja" OR o$="ja"
 THEN LET A$(Z,1)=" "
4060 NEXT Z
4070 GO TO 180
5000 CLS
5001 PRINT "AB WELCHEM EINTRAG?"
5002 INPUT ANF
5003 IF ANF<1 OR ANF>LETZTES THE
N GO TO 5002
5005 FOR Z=ANF TO LETZTES
5010 GO SUB 7000
5020 NEXT Z
5050 GO TO 2083
6000 CLS
6010 FOR Z=1 TO LETZTES
6015 IF A$(Z,1)=" " THEN GO TO 6
040
6020 GO SUB 7100
6030 INPUT Q$: IF Q$="z" OR Q$="
Z" THEN COPY
6040 NEXT Z
6050 PRINT ,,"ENDE"
6060 INPUT Q$
6065 IF Q$="Z" OR Q$="z" THEN CO
PY
6070 GO TO 180
7000 FOR J=1 TO W
7010 LPRINT A$(Z,A(J)+1 TO A(J+1
))
7020 NEXT J
7030 LPRINT
7040 RETURN
7100 FOR J=1 TO W
7110 PRINT A$(Z,A(J)+1 TO A(J+1)
)
7120 NEXT J
7125 PRINT
7130 RETURN
8000 CLS
8010 FOR L=1 TO W
8020 IF A(L)<>A(L+1) THEN PRINT
L;" ";W$(L)
8030 NEXT L
8040 RETURN
9000 SAVE "Kartei" LINE 9010
9010 IF LETZTES=0 THEN GO TO 90
9020 GO TO 180
```

Nachdem Sie das Programm eingegeben haben, speichern Sie es mehrmals auf Kassette, um immer eine unbeschriebene Version zur Verfügung zu haben.

4.2.1 Einrichten der Kartei

Im leeren Zustand ist das Programm noch zu (fast) allem fähig. In der Eingangsroutine (bis Zeile 117) wird die Kartei von Ihrer ersten Inbetriebnahme für den späteren Gebrauch eingerichtet. Die Einträge können in 1 bis 20 Zeilen, genannt Wörter, mit 0 bis 32 Zeichen je Zeile gespeichert werden. Zeilen ohne Zeichen werden bei der Ausgabe als Freizeilen gedruckt.

Vor der ersten Benutzung der Kartei sind, sowohl Anzahl der Wörter (Zeilen) als auch der Zeichen pro Wort zu definieren. Außerdem bekommt jedes Wort und die ganze Kartei einen Namen. Dann ist die Kartei auf Ihre individuellen Zwecke hin ausgerichtet und betriebsbereit.

Ein Beispiel:
Sie wollen eine Kundenkartei anlegen. Die Kunden sollen wahlweise nach Kundennummer, Nachname oder Wohnbereich (Postleitzahl) sortiert werden können. Zwischen Straße und Wohnort soll eine Freizeile sein. Die Anzeige soll folgendermaßen aussehen:

```
0001
HUELSMANN, ROLAND
NIDDASTRASSE 104

6000 FRANKFURT 1
```

Die Anzeige besteht aus 5 Wörtern:

1. Kundennummer
2. Name, Nachname
3. Straße Nr.
4. Freizeile
5. PLZ Wohnhort

Auf die Frage "WIEVIELE WOERTER PRO EINTRAG?" geben Sie also eine fünf ein. Für jedes Wort fragt der SPECTRUM jetzt nach Anzahl der Zeichen und dem Namen.

Dem 4. Wort, der Freizeile, braucht kein Name gegeben zu werden. Es reciht, einfach ENTER zu drücken.

Nach dem letzten Wort errechnet der SPECTRUM nun den Speicherbedarf pro Eintrag und wieviele Einträge dieser Größe Sie in der Kartei speichern können. Antworten Sie auf die Frage OK? mit "Nein", beginnt der SPECTRUM von vorn, anderenfalls ("Ja") fragt der SPECTRUM zum Schluß nach dem Namen der Kartei (bitte nicht mehr als 28 Zeichen!), um dann das "Menue" anzuzeigen.

Die Kundenkartei ist eingerichtet und betriebsbereit. Sobald auch nur ein Eintrag gemacht und die Kartei abgespeichert wurde, startet sie sich automatisch mit dem Menue. Die Vorfragen brauchen nicht mehr gestellt zu werden.

Auf oben beschriebene Weise lassen sich beliebige Karteien, Register und Kataloge erstellen, da ja jedes Wort alphabetisch geordnet werden kann. In obigem Beispiel kann die Kartei sogar nach Straßennamen geordnet werden.

So können Sie also Ihre Bücher ordnen und nach Autor oder Titel oder Schlagwort ausdrucken lassen. Den Möglichkeiten dieses Programmes sind außer dem RAM keine Grenzen gesetzt !

4.2.2. Benutzen der Kartei

Nach dem Laden der beschriebenen Kartei von Kassette erscheint das Menue.

Sie haben nun folgende Möglichkeiten:

1. NEUER EINTRAG

Der SPECTRUM sucht den ersten freien Eintrag und fragt nacheinander ab, mit was die einzelnen Zeilen zu füllen sind (ausgenommen etwaige Freizeilen). Wenn die eingegebene Zeichenzahl über die definierte Zeichenzahl hinausgeht, werden die überzähligen Zeichen ignoriert.

Übrigens:
Es können alle Zeichen außer dem " eingegeben werden. Sobald ein Eintrag komplett ist, sucht der SPECTRUM nach dem nächsten freien Platz. Will man mit den Eingaben aufhören und zum Menue zurückkehren, gibt man für das erste Wort einfach nur ENTER ein. Der SPECTRUM nimmt an, daß man nun nichts mehr speichern will.

2. EINE "ALPHABETISCHE" LISTE DRUCKEN

Der SPECTRUM führt alle verwendeten Wörter auf und fragt, welches denn alphabetisch (bzw. nach Nummern) geordnet werden soll. Sollen die Kundennummern geordnet sein, gibt man nun eine 1 ein, soll es der Nachname sein, eine 2, etc. Nach kurzem Überlegen erscheint wieder das Menue.

3. EINEN BESTIMMTEN EINTRAG SUCHEN

Dies ist der Clou dieses Programmes:
Zunächst fragt der SPECTRUM, wie oben, nach was für einer Zeile gesucht wird. Nehmen wir an, wir wollen alle Postleitzahlen, die mit 6 beginnen heraussuchen. "PLZ WOHNORT" ist die 5. Zeile, also geben wir 5 ein. Dann fragt der SPECTRUM nach der Genauigkeit.
Geben wir 0 ein, müssen alle Zeichen übereinstimmen, bei 4 (z. B.) nur die ersten 4, bei 2 die ersten 2, etc. In unserem Fall wird nur nach der ersten Ziffer gefragt, also geben wir eine 1 ein. Dann fragt er nach dem zu suchenden Wort, bzw. Begriff oder Zeichen. Hier geben wir die "6" ein. Nun listet der SPECTRUM alle Einträge, deren Postleitzahl mit "6" beginnt.
Nach jedem Eintrag zeigt der SPECTRUM das String-Input-Zeichen ("L"). Gibt man nun ein "Z" ein wird der Bildschirmausdruck auf den Drucker kopiert, bei jeder anderen Eingabe wird der nächste Eintrag angezeigt. Hat er alle gesuchten Einträge angezeigt, zeigt er "ENDE". Durch ENTER kann man zum Menue zurückkehren.
Eine Besonderheit des Programmes:
Wenn der SPECTRUM bei einer gewünschten Genauigkeit, die mehr als ein Zeichen beträgt, keinen Eintrag findet, dann fragt er "SOLL ICH ETWAS UNGENAUER SUCHEN?". Bei "Nein" kehrt er zum Menue zurück ansonsten sucht er mit um 1 reduzierter Genauigkeit. Dieses Spiel wiederholt sich so oft, bis er entweder etwas gefunden hat oder auch bei der Genauigkeit 1 nicht fündig geworden ist.

4. EINEN EINTRAG LOESCHEN

Es ist lediglich das erste Wort des zu löschenden Eintrages einzugeben. Der SPECTRUM zeigt den Eintrag und fragt "OK?". Nur wenn "Ja" geantwortet wird (oder "ja"), wird der Eintrag gelöscht. Der SPECTRUM setzt das erste Zeichen des ersten Wortes auf " " (Leerzeichen), was für ihn bedeutet, der Eintrag existiert nicht. Der Platz kann wieder neu beschrieben werden.

5. INHALT IN JETZIGER ORDNUNG AUSDRUCKEN

Der SPECTRUM druckt die gesamte Kartei in jetziger Ordnung aus. Wenn die Kartei einmal, sagen wir nach Namen, alphabetisch geordnet ist, bleibt sie dies solange, bis neue Einträge gemacht werden, de hinten angehängt werden, oder die Kartei nach einer anderen Zeile geordnet wird. Blieb die Kartei unverändert, und man benötigt sie in der gleichen Ordnung ein weiteres mal, braucht sie nicht neu geordnet zu werden. So läßt sich Zeit sparen.

6. GESAMTINHALT AUF DEM BILDSCHIRM

Die gesamte Kartei wird, Bildschirmseite für Bildschirmseite, auf den Bildschirm gebracht; in der momentanen Ordnung. Nach jedem Eintrag kann durch ENTER weitergemacht werden; Z und ENTER sorgen zunächst für eine Kopie des Bildschirmes auf den Drucker.

7. DIE KARTEI SPEICHERN

Schalten Sie den Kassettenrecorder auf Aufnahme, drücken Sie auf "9" und die Kartei wird in momentaner Ordnung mit dem momentanen Inhalt auf Band gespeichert. Anschließend – und nach dem LOADen – erscheint das Menue auf dem Bildschirm.

4.3 Programm 20: "Haushaltsbuchführung"

```
  100 DIM e$(200,8)
  110 DIM c$(200,2)
  120 DIM t$(200,13)
  130 DIM t(200)
  140 DIM m(200)
  150 LET haben=0
  160 LET e=1
  170 LET a$="Datum   Ein/Aus   Tex
t           Code"
  180 LET b$="_____________________
_____________"
  190 LET c=99
  191 LET s=100
  192 DIM y$(2)
  200 REM : Menue
  210 BORDER 7: PAPER 7: INK 0: C
LS
  220 PRINT INK 1;"Haushaltsbuch
führung"
  230 PRINT AT 3,0;"Was wuenschen
Sie:"'"_____________________"
  240 PRINT ,,,,"E Neuer Eintrag"
  AND e<200
  250 PRINT ,,"K Kontoauszug"
  260 PRINT ,,"S Kontostand +
                Aufschluesselung
  Nach Codes"
  270 PRINT ,,"S Konto speichern"
  300 PRINT ,,"E Einblick In Zuru
eckliegende     Kontoauszuege"
  400 PRINT ,,,,"Druecken Sie Bit
te Die          Entsprechende Ta
ste."
  500 PAUSE 0
  510 LET i$=INKEY$
  520 IF i$<"a" OR i$>"e" THEN GO
  TO 500
  530 GO TO 1000*(CODE i$-96)
1000 CLS
1010 PRINT a$;b$
1012 LET k=12
1014 IF k>=e THEN LET k=e-1
1016 LET a=e-k
1018 FOR i=a TO e-1
1020 GO SUB 6000
1022 NEXT i
1024 LET zeile=k+2
1040 IF zeile<22 THEN GO TO 1045
1042 PRINT AT 0,0;a$;b$
1043 LET zeile=zeile-1
1045 PRINT AT zeile,0;"[ ]"
1050 INPUT t(e)
1052 LET t(e)=INT t(e)
1055 IF t(e)=s THEN GO TO 200
1056 IF t(e)=c THEN GO TO 1300
1060 IF t(e)<0 OR t(e)>51 THEN G
O TO 1050
1070 PRINT AT zeile,0;t(e);".[ ]"
1080 INPUT m(e)
```

```
1065 LET m(e)=INT m(e)
1100 IF m(e)<0 OR m(e)>12 THEN G
O TO 1080
1115 PRINT AT zeile,2+(t(e)>9);m
(e);"."
1120 PRINT AT zeile,8;"-/+?"
1130 INPUT ein
1140 IF ein<=-10000 OR ein>=1000
0 THEN GO TO 1130
1160 LET haben=haben+ein
1170 LET z=ein
1180 GO SUB 2500
1190 LET e$(e)=z$
1200 PRINT AT zeile,7;e$(e);" "
1210 INPUT t$(e)
1220 PRINT AT zeile,16;t$(e);"
";
1230 INPUT c$(e)
1240 PRINT AT zeile,30;c$(e)
1250 LET zeile=zeile+1
1260 LET e=e+1
1270 IF e>200 THEN GO TO 1500
1280 GO TO 1040
1300 IF zeile<3 THEN GO TO 1040
1310 LET zeile=zeile-1
1320 LET e=e-1
1330 LET haben=haben-VAL e$(e)
1335 PRINT AT zeile,0;"
                        "
1340 GO TO 1040
1500 CLS
1510 PRINT "Der reservierte Spei
cherplatz   ist voll."
1520 PRINT ,,"Soll ich die erste
n 50 Eintrae- ge loeschen?"
1530 INPUT q$
1540 IF q$="Nein" OR q$="nein" T
HEN GO TO 200
1550 CLS
1570 FOR j=1 TO 150
1580 LET t(j)=t(j+50)
1590 LET m(j)=m(j+50)
1600 LET e$(j)=e$(j+50)
1642 LET t$(j)=t$(j+50)
1652 LET c$(j)=c$(j+50)
1630 NEXT j
1640 LET e=151
1650 CLS
1660 PRINT "Neuer Platz ist gesc
haffen."
1670 GO TO 230
2000 IF e=0 THEN GO TO 200
2010 LET von=e-10
2020 IF von<1 THEN LET von=1
2030 LET zu=e-1
2040 CLS
2050 PRINT "Datum"
2060 INPUT i$
2070 CLS
2080 PRINT "Kontoauszug"
2090 PRINT ,,i$
2100 PRINT ,,a$;b$
```

```
2110 FOR i=von TO zu
2120 GO SUB 6000
2130 NEXT i
2135 PRINT b$
2140 LET z=ABS haben
2142 GO SUB 2500
2145 PRINT ,, PAPER 1; INK 7;"Ha
ben:" AND haben>=0; PAPER 2; FLA
SH 1;"Soll:" AND haben<0; PAPER
7; FLASH 0; INK 0;TAB 16;z$;" DM
"
2150 PRINT ,"============"
2160 PAUSE 0
2170 IF INKEY$="z" THEN COPY
2180 GO TO 200
2500 LET i$=STR$ z
2510 IF INT z=z THEN LET i$=i$+"
.0"
2520 IF i$(LEN i$-1)="." THEN LE
T i$=i$+"0"
2530 DIM z$(8)
2540 LET z$(9-LEN i$ TO )=i$
2550 RETURN
3000 CLS
3010 LET z=haben
3020 GO SUB 2500
3030 PRINT "Kontostand:",z$;" DM
"
3040 PRINT ,"------------"
3050 PRINT "Monat? ";
3060 INPUT m
3070 PRINT m
3080 PRINT "Code?",
3090 INPUT y$
3100 PRINT y$;" >";TAB 20;
3105 LET sum=0
3110 FOR j=1 TO e-1
3120 IF m=m(j) AND y$=c$(j) THEN
 LET sum=sum+VAL e$(j)
3130 NEXT j
3140 LET z=sum
3150 GO SUB 2500
3160 PRINT z$
3170 IF PEEK 23689<6 THEN GO TO
3210
3180 PRINT "Code?",
3190 INPUT y$
3200 IF y$<>"   " THEN GO TO 3100
3210 PRINT AT 21,0;"Copy=""z"""
3215 PAUSE 0
3220 IF INKEY$="z" THEN COPY
3230 GO TO 200
4000 GO TO 9000
5000 CLS
5010 PRINT "Ab welchem Eintrag?"
5020 INPUT x
5025 CLS
5030 PRINT a$;b$
5040 LET von=x
5050 IF von<1 THEN LET von=1
5060 LET zu=von+19
5070 IF zu>e-1 THEN LET zu=e-1
```

```
5080 FOR i=von TO zu
5090 GO SUB 6000
5100 NEXT i
5106 PRINT #0;von;" > ";zu
5110 PAUSE 0
5120 IF INKEY$="Z" THEN COPY
5130 GO TO 200
6000 PRINT t(i);".";m(i);".";TAB
 7;e$(i);" ";t$(i);TAB 30;c$(i)
6010 RETURN
9000 SAVE "Konto" LINE 200
9010 GO TO 200
```

Wichtige Variablen:

e$ (200,8)	Ein-/Ausgaben im formatierten Strings
C$(200)	Codes der einzelenen Posten
T$(200,12)	Text zu den einzelnen Posten
T(200),M(200)	Eingabedaten der Posten
HABEN	aktueller Kontostand
E	der nächste Eintrag

Geben Sie das Programm ein. Starten Sie es mit RUN und speichern Sie es durch Drücken von "B" auf Kassette.
Nach dem LOADen erscheint das Menue.

A NEUER EINTRAG

In der obersten Zeile erscheint "T?". Der Tag ist einzugeben. Als nächstes erscheint "M?". Jetzt ist der Monat einzugeben. Dann zeigt der SPECTRUM "–/+?". Jetzt ist der Eintrag zu machen. Wenn Geld ausgegeben wird, einen negativen Betrag eingeben! Durch GOSUB 2500 bringt der SPECTRUM dieses einheitliche Format. Nun haben Sie noch Gelegenheit einen kurzen Text und den Code einzugeben. Der Code dient dazu, die einzelnen Ausgaben später in verschiedene Rubriken aufzuschlüsseln. Nach der Eingabe des Codes, erwartet der SPECTRUM wieder die Eingabe eines Datums. Geben Sie als Tag ein "s" ein, kehrt der SPECTRUM zum Menue zurück, geben sie ein "c" ein, wird der vorangehende Eintrag gelöscht.

B KONTOAUSZUG

Zunächst fragt der SPECTRUM "DATUM?". Hier können Sie das Datum des Auszuges eingeben. Danach druckt der SPECTRUM die letzten 10 Einträge und den aktuellen Kontostand aus. Durch Drücken irgendeiner Taste kehrt der SPECTRUM zum Menue zurück. Drücken Sie "Z" gibt er vorher eine Kopie auf den Drucker.

C KONTOSTAND + AUFSCHLÜSSELUNG NACH CODES

Zunächst fragt der SPECTRUM nach dem Monat, für den er diese Aufschlüsselung vornehmen soll. Als nächstes fragt er nach dem gesuchten Code. Nun forstet er alle Einträge des gewünschten Monats nach dem gesuchten Code durch und gibt die Summe dieser Einträge an. Nun fragt er nach einem weiteren Code. Gibt man hier nur ENTER ein, kehrt er zum Menue zurück, "z" besorgt eine Kopie auf den Printer.

D KONTO SPEICHERN

Das Programm wird im gegenwärtigen Zustand auf Kassette gespeichert.

E EINBLICK IN ZURÜCKLIEGENDE KONTOAUSZÜGE

Der SPECTRUM fragt "AB WELCHEN EINTRAG?". Dann druckt er die folgenden Einträge aus. Durch Drücken irgendeiner Taste ("Z"= COPY) kehrt er zum Menue zurück.

Übrigens:
Sind alle 200 Einträge gemacht, so fragt der SPECTRUM, ob die ersten 50 zu löschen sind. Wird diese Frage bejaht, transportiert der Computer die Einträge 51 bis 200 in die Plätze 1 bis 150, wodurch wieder Platz ab Eintrag 151 geschaffen wurde.

4.4 Programm 21: "Mondphasen"

```
  10 BORDER 0: PAPER 0: INK 6: C
LS
  20 PRINT "▚▚▚ MONDPHASEN ▚▚▚
": GO SUB 1000
  30 PRINT '"ANFANGSDATUM? ";
  40 INPUT TAG
  50 IF TAG<1 OR TAG>31 THEN GO
TO 40
  60 PRINT TAG;".";
  70 INPUT MONAT
  80 IF MONAT<1 OR MONAT>12 THEN
 GO TO 70
  90 PRINT MONAT;".";
 100 INPUT JAHR
 110 PRINT JAHR
 120 IF TAG>29 AND MONAT=2 OR JA
HR<1583 THEN GO TO 420
 130 IF TAG>30 AND (MONAT=4 OR M
ONAT=6 OR MONAT=9 OR MONAT=11) T
HEN GO TO 420
```

```
 140 CLS
 150 PRINT INK 5;"DATUM
  PH.    TAGE N.J.N."
 160 LET SYN UML=29.53059
 170 LET SN=SYN UML/2
 180 LET TAGE1=TAG+31*(MONAT-1)+
365*JAHR
 190 IF MONAT>2 THEN GO TO 250
 200 LET TAGE2=INT ((JAHR-1)/4)-
INT ((3/4)*(INT ((JAHR-1)/100))+
1)
 210 LET TAGE=TAGE1+TAGE2
 220 LET WOCHENTAG=TAGE-(INT (TA
GE/7)*7)
 230 GO SUB 500+20*WOCHENTAG
 240 GO TO 270
 250 LET TAGE=TAGE1-INT (.4*MONA
T+2.3)+INT (JAHR/4)-INT ((3/4)*(
INT (JAHR/100)+1))
 260 GO TO 220
 270 LET A=(TAGE-578201)/SYN UML
 280 REM : TNN=TAGE NACH NEUMOND
 290 LET TNN=INT (SYN UML*(A-INT
 (A))+.5)
 300 IF TNN>SN THEN GO TO 330
 310 LET PHASE=4*(TNN/SN)
 320 GO TO 340
 330 LET PHASE=4*((TNN-SYN UML)/
SN)
 340 LET PHASE=INT (PHASE+.5)
 350 GO SUB 640
 360 LET TAG=TAG+1
 370 IF TAG<=29 OR TAG<=30 AND M
ONAT<>2 OR TAG<=31 AND (MONAT<>2
 AND MONAT<>4 AND MONAT<>6 AND M
ONAT<>9 AND MONAT<>12) THEN GO T
O 180
 380 PRINT '"NEUES DATUM?"
 390 INPUT 0$
 400 IF 0$="Ja" OR 0$="ja" THEN
RUN
 410 STOP
 420 REM
 430 PRINT
 440 PRINT FLASH 1;"FALSCHE EING
ABE."
 450 IF JAHR<1583 THEN PRINT "DA
TEN VOR 1583 WERDEN NICHT      AK
ZEPTIERT."
 460 PRINT
 470 PRINT
 480 PRINT "DATUM? ";
 490 GO TO 40
 500 LET T$="SAM "
 510 RETURN
 520 LET T$="SON "
 530 RETURN
 540 LET T$="MON "
 550 RETURN
 560 LET T$="DIE "
 570 RETURN
 580 LET T$="MIT "
```

```
590 RETURN
600 LET T$="DON "
610 RETURN
620 LET T$="FRE "
630 RETURN
640 PRINT T$;"0" AND TAG<10;TAG
;".";"0" AND MONAT<10;MONAT;".";
JAHR;TAB 16;
650 IF ABS PHASE=4 THEN PRINT "
●";
660 IF PHASE=3 THEN PRINT "●";
670 IF PHASE=2 THEN PRINT "●";
680 IF PHASE=1 THEN PRINT "●";
690 IF PHASE=0 THEN PRINT " ";
700 IF PHASE=-1 THEN PRINT "●";
710 IF PHASE=-2 THEN PRINT "●";
720 IF PHASE=-3 THEN PRINT "●";
730 PRINT TAB 24;"0" AND TNN<10
;TNN
740 RETURN
1000 RESTORE : FOR I=0 TO 55: RE
AD A
1010 POKE USR "A"+I,A: NEXT I
1020 RETURN
1030 DATA 12,6,3,3,3,3,6,12,12,1
4,15,15,15,15,14,12,12,30,63,63,
63,63,30,12,60,126,255,255,255,2
55,126,60,48,120,252,252,252,252
,120,48,48,112,240,240,240,240,1
12,48,48,96,192,192,192,192,96,4
8
```

Wichtige Variablen:

SYN UML	Synodische Umlaufszeit des Mondes
TNN	Tage nach Neumond
T$	Tagesname

Die übrigen Variablen ergeben sich aus ihren Bezeichnungen.

Das Programm errechnet Wochentag, Mondphase und Zahl der Tage nach dem letzten Neumond.

Nach dem LOADen fragt der SPECTRUM nach dem Datum. Geben Sie das Datum ein; und zwar nacheinander (durch ENTER getrennt) Tag, Monat und Jahr. Der SPECTRUM druckt nun die Tabelle bis zum Monatsende, bzw. bis der Bildschirm fast voll ist aus. Danach fragt er, ob ein neues Datum gewünscht wird.

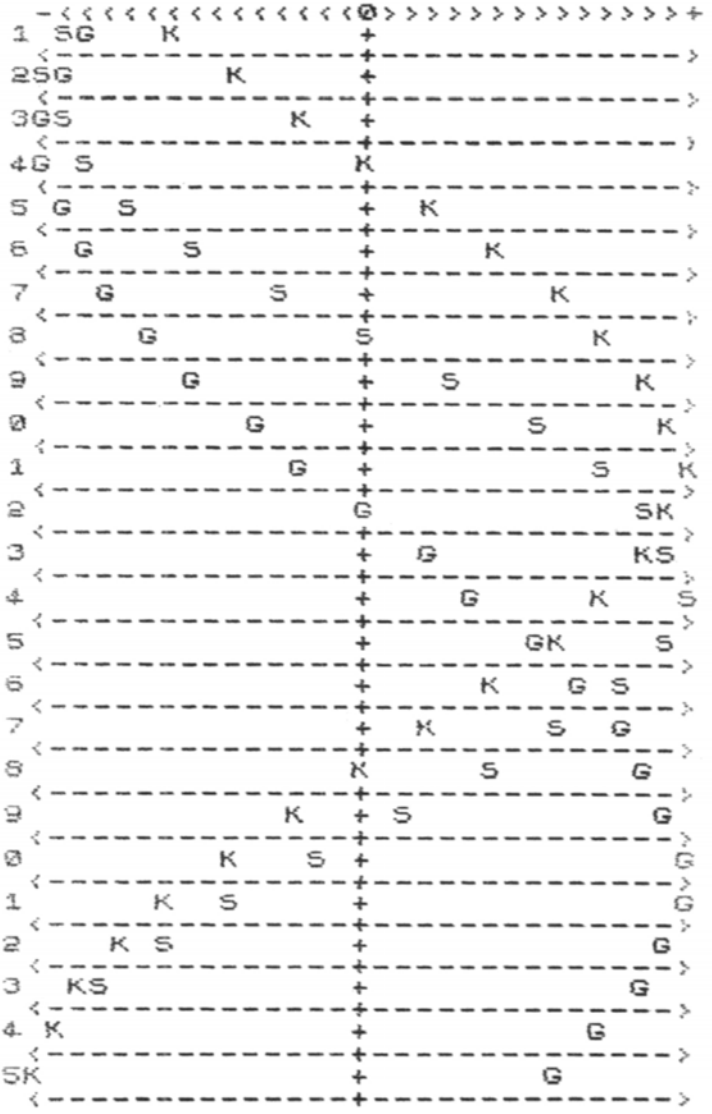
Der Computer zeigt acht verschiedene Phasen an.

4.5 Programm 22: "Biorhythmus"

```
RGH-SOFTWARE PRAESENTIERT:

** ZX    - BIORHYTHMUS **

GEBURTSDATUM: 23 09 1955
ANFANGSDATUM: 01 03 1983
```



```
  10 GO SUB 6000: REM : MENUE
  20 BORDER 4: PAPER 4: INK 0: C
LS
  30 PRINT "> BIORHYTHMUS :
        RGH"
  40 PRINT AT 6,0;"A WOCHENTAG-B
ESTIMMUNG"''"B BIORHYTHMUS (TAGE
SWERT)"
  50 PRINT ''"C BIORHYTHMUS (TV-G
RAPHIK)"''"D BIORHYTHMUS (PRINTE
R-GRAPHIK)"
  60 PAUSE 0
  70 LET A$=INKEY$
  80 GO TO 100*(A$="A" OR A$="a"
)+200*(A$="B" OR A$="b")+500*(A$
="C" OR A$="c")+1000*(A$="D" OR
A$="d")

 100 REM : WOCHENTAG-
           BESTIMMUNG
 105 BORDER 1: PAPER 1: INK 7: C
LS
 108 PRINT PAPER 2; INK 7;"> WOC
HENTAG-BESTIMMUNG <"
 110 PRINT
 120 PRINT "DATUM: ";
 130 GO SUB 5000
 140 PRINT D$;" = ";
 150 GO SUB 5200
 160 GO SUB 5300
 170 PRINT T$
 180 GO TO 120

 200 REM : BIORHYTHMUS
           -TAGESWERT
 210 BORDER 2: PAPER 2: INK 7: C
LS
 220 PRINT PAPER 1;"> BIORHYTHMU
S - TAGESWERT <"
 230 PRINT
 240 PRINT "GEBURTSDATUM: ";
 250 GO SUB 5000
 260 PRINT D$
 270 GO SUB 5200
 280 LET T1=TAGE
 290 GO SUB 5300
 300 LET A$=T$
 310 PRINT "ANALYSEDATUM: ";
 320 GO SUB 5000
 330 PRINT D$
 340 GO SUB 5200
 350 LET T2=TAGE
 360 GO SUB 5300
 370 LET B$=T$
 380 PRINT
 390 LET DT=T2-T1
 400 GO SUB 5400
 410 PRINT "SIE SIND AN EINEM ";
A$'"GEBOREN."'"AM ";B$;", DEN ";
D$'"SIND SIE ";DT;" TAGE ALT."
 420 PRINT
```

```
 430 PRINT "IHRE WERTE AM ";D$;"
:"
 440 PRINT
 450 PRINT "GEIST:",INT (G+.5),"
SEELE:",INT (S+.5),"KOERPER:",IN
T (K+.5)
 460 PRINT AT 21,0;"NAECHSTE TAG
E?"
 470 PAUSE 0
 480 IF INKEY$="j" THEN GO TO 80
0
 490 RUN

 500 REM : BIORHYTHMUS
           TV-GRAPHIK
 505 BORDER 0: PAPER 0: INK 7: C
LS
 507 LET PG=0
 510 PRINT PAPER 6; INK 0;"> TV-
GRAPHIK <"
 520 PRINT
 530 PRINT "GEBURTSDATUM: ";
 535 GO SUB 5000
 537 PRINT D$
 538 LET A$=D$
 540 GO SUB 5200
 545 LET T1=TAGE
 550 PRINT "ANFANGSDATUM: ";
 555 GO SUB 5000
 560 PRINT D$
 565 GO SUB 5200
 570 LET T2=TAGE
 575 IF PG THEN RETURN
 580 CLS
 585 PRINT PAPER 7; INK 1;"<-
    1 = ";D$;TAB 30;"+>"
 590 LET DT=T2-T1
 600 DIM G(20)
 610 DIM S(20)
 620 DIM K(20)
 625 LET DT=DT-1
 627
 630 FOR J=1 TO 20
 640 LET DT=DT+1
 650 GO SUB 5400
 660 LET G(J)=G/6.8
 670 LET S(J)=S/6.8
 680 LET K(J)=K/6.8
 685 PRINT AT J,0;(STR$ J)(LEN S
TR$ J),"+"
 690 NEXT J
 710 FOR J=1 TO 20
 720 PRINT AT J,16+G(J); INK 6;"
*";AT J,16+S(J); INK 4;"*";AT J,
16+K(J); INK 2;"*"
 730 NEXT J
 740 PAUSE 0
 750 RUN

 800 REM : TABELLE
 810 BORDER 0: PAPER 1: INK 7: C
LS
```

```
 820 PRINT AT 0,0;"TAG          DA
TUM   G     S     K
"
 830 LET T=T+1
 835 LET TAGE=TAGE+1
 837 LET DT=DT+1
 840 IF T>28 THEN GO SUB 5600
 855 GO SUB 5300
 860 GO SUB 5400
 870 PRINT T$;TAB 11;T;".";M;"."
;TAB 18;INT (G+.5);TAB 23;INT (S
+.5);TAB 28;INT (K+.5)
 875 IF PEEK 23689<5 THEN GO TO
460
 880 IF INKEY$="" THEN GO TO 830
 890 IF INKEY$<>"" THEN GO TO 89
0
 900 GO TO 460

1000 REM : BIORHYTHMUS
           PRINTER-GRAPHIK
1005 BORDER 7: PAPER 7: INK 1: C
LS
1010 PRINT "> PRINTER-GRAPHIK <"
1020 LPRINT "RGH-SOFTWARE PRAESE
NTIERT:",,,"** ZX    - BIORHYTHMU
S **",,,
1030 LET PG=1
1040 GO SUB 520
1050 LET DT=T2-T1-1
1055 LPRINT "GEBURTSDATUM:  ";A$
1060 LPRINT "ANFANGSDATUM:  ";D$
1070 LPRINT
1090 LPRINT "  -<<<<<<<<<<<<<<0>>
>>>>>>>>>>>>+"
1100
1105 FOR J=1 TO 30
1110 LET DT=DT+1
1112 LET J$=STR$ J
1115 LET J$=J$(LEN J$)
1120 LPRINT J$,"+";
1130 GO SUB 5400
1140 LET G=16+G/6.8
1150 LET S=16+S/6.8
1160 LET K=16+K/6.8
1170 LPRINT AT 0,G;"G";AT 0,S;"S
";AT 0,K;"K";
1200 LPRINT
1210 LPRINT "  <--------------+--
-------------->"
1220 NEXT J
1230
1240 PRINT AT 10,0;"WEITER?"
1250 PAUSE 0
1260 IF INKEY$="j" THEN GO TO 11
00
1265 LPRINT
1270 LPRINT
1275 LPRINT
1280 LPRINT
1285 LPRINT
1290 LPRINT
```

```
1300 RUN

5000 REM : DATUMSEINGABE
5010 INPUT AT 0,0;"TT.MM.JJJJ"'
LINE D$
5015 LET K$="G"
5017 IF D$="" THEN GO TO 5010
5018 IF D$(1)="S" OR D$(1)="s" T
HEN RUN
5020 IF LEN D$<7 THEN GO TO 5010
5021 IF D$(1)<>"J" AND D$(1)<>"j
" THEN GO TO 5024
5022 LET D$=D$(3 TO )
5023 LET K$="J"
5024 IF D$(3)<>"." AND D$(3)<>"
" OR D$(6)<>"." AND D$(6)<>" " T
HEN GO TO 5010
5030 LET T=VAL D$( TO 2)
5040 LET M=VAL D$(4 TO 5)
5050 LET J=VAL D$(7 TO )
5090 RETURN

5200 REM : ANZAHL DER TAGE SEIT
           DEM 01.01.-1
5210 LET TAGE=T+30*M+(M>1)-2*(M>
2)+(M>3)+(M>5)+(M>7)+(M>8)+(M>10
)+365*J+INT (J/4)
5220 IF K$="J" THEN GO TO 5260
5230 LET TAGE=TAGE-INT (J/100)+I
NT (J/400)
5240 IF M>2 THEN IF INT (J/4)=J/
4 THEN IF INT (J/100)<>J/100 OR
INT (J/400)=J/400 THEN LET TAGE=
TAGE+1
5250 RETURN
5260 IF M>2 THEN IF INT (J/4)=J/
4 THEN LET TAGE=TAGE+1
5270 LET TAGE=TAGE-2
5280 RETURN

5300 REM : WOCHENTAG
5305 LET L$="77666:7"
5310 LET W$="SAMSTAG   SONNTAG
 MONTAG    DIENSTAG  MITTWOCH  D
ONNERSTAGFREITAG    "
5315 LET WT=TAGE+5
5320 LET WT=INT (WT-INT (WT/7)*7
+.5)
5330 LET A=WT*10+1
5340 LET E=WT*10+CODE L$(WT+1)-4
8
5350 LET T$=W$(A TO E)
5360 RETURN

5400 REM : BIOWERTE AUS DEM
           ALTER IN TAGEN (DT)
           ERRECHNEN
5420 LET Y=33
5430 GO SUB 5500
5440 LET G=100*F
5450 LET Y=23
```

```
5460 GO SUB 5500
5470 LET S=100*F
5475 LET Y=28
5480 GO SUB 5500
5485 LET K=100*F
5490 RETURN
5500 LET F=INT (DT-INT (DT/Y)*Y+
.5)
5510 LET F=(F/Y)*PI*2
5520 LET F=SIN F
5530 RETURN

5600 REM : DATUMSUEBERLAUF
5610 IF T>31 AND (M=1 OR M=3 OR
M=5 OR M=7 OR M=8 OR M=10 OR M=1
2) THEN GO TO 5700
5620 IF T>30 AND (M=4 OR M=6 OR
M=9) THEN GO TO 5700
5630 IF T>28 AND M=2 AND INT (J/
4)=J/4 AND (INT (J/100)<>J/100 O
R J/400=INT (J/400)) THEN GO TO
5700
5640 IF T>29 AND M=2 THEN GO TO
5700
5650 RETURN
5700 LET T=1
5710 LET M=M+1
5720 IF M<13 THEN RETURN
5730 LET J=J+1
5740 LET M=1
5750 RETURN

6000 RESTORE : FOR I=0 TO 7: REA
D A
6010 POKE USR "g"+I,A: NEXT I: R
ETURN
6020 DATA 0,0,24,60,60,24,0,0
```

Wichtige Variablen:

D$	Datum nach der Datumseingabe (GOSUB 5000)
T$	Wochentag des Tatums nach GOSUB 5300
L$	Code-Tabelle für die Länge der einzelnen Namen der Wochentage
W$	Tabelle der Wochentagsnamen
T, M, J	Datum aufgeschlüsselt in Tage, Monat und Jahr nach GOSUB 5000
G	Geist-Wert
K	Körper-Wert
S	Seelischer Wert

Geben Sie das Programm ein und speichern Sie es mit GOTO 9000.

Dann startet es sich nach dem LOADen von selbst.

ZUR DATUMSEINGABE

Das Programm verarbeitet auch Daten nach dem Julianischen Kalender (z. B. für Daten vor dem 15. Oktober 1582). In diesem Fall ist vor dem Datum ein "J." einzugeben (z. B. "J.23.11.1553") Daten ohne "J." interpretiert der SPECTRUM als gemäß dem gültigen Gregorianischen Kalender.

Das Programm verarbeitet alle Daten seit Christi Geburt. Monat und Tag müssen immer mit je zwei Stellen eingegeben werden. Die Punkte dürfen durch SPACEs ersetzt werden. Jedes Datum muß die Form "TT.MM.J" haben, wobei die Stellenzahl des Jahres egal ist. Falsche Eingaben nimmt der SPECTRUM in der Regel nicht an.

A WOCHENTAG–BESTIMMUNG

Geben Sie ein Datum ein, und der SPECTRUM sagt Ihnen, was für ein Tag es war oder sein wird. Wollen Sie zum Menue zurückkehren, geben Sie einfach ein "S" für STOP ein.

B BIORHYTMUS TAGESWERT

Hier erwartet der SPECTRUM zunächst Ihr Geburtsdatum und dann den ersten Tag, für den die biorhythmischen Werte ausgerechnet werden sollen. Der niedrigste Wert ist jeweils –100, der höchste +100. Gemäß den Lehren der Biorhythmik hat die Geist-Kurve eine Frequenz von 33 Tagen, die Körperkurve von 28 Tagen und die Seelenkurve von 23 Tagen. Besonders kritische Tage sind solche, an denen eine Linie den Wert 0 über- oder unterschreitet.

Wollen Sie außer dem einen Werte-Triplet noch die nächsten Tage erfahren, antworten Sie die Frage "NAECHSTE TAGE?" mit J. Bei N erscheint wieder das Menue. Durch Drücken von S zwischen zwei Tagen, können Sie die Tabelle anhalten. Durch Drücken auf N erscheint dann wieder das Menue.

C BIORHYTHMUS TV–GRAPHIK

Der SPECTRUM erwartet das Geburtsdatum und den ersten Tag, der dargestellt werden soll. Die Null-Linie ist eine Reihe von "+". Die Zahlen am linken Rand erleichtern das Auffinden eines bestimmten Tages. Durch Druck auf irgendeine Taste kehrt der SPECTRUM zum

Menue zurück.

D BIORHYTHMUS PRINTER–GRAPHIK

Das Non-Plus-Ultra dieses Programmes! Nachdem Sie D gedrückt haben – erschrecken Sie nicht! – rast der Printer los, um den Titel zu drucken. Dnach erfragt der SPECTRUM wieder Geburts- und Anfangsdatum und los gehts. Er printet die Werte der nächsten 30 Tage in eine übersichtliche Graphik. Dann erscheint auf dem Bildschirm die Frage "WEITER?". Drückt man auf J, folgen die nächsten Tage, bei allen anderen Eingaben transportiert der SPECTRUM das Papier noch soweit, daß man es abreißen kann und kehrt zum Menue zurück.

4.6 Programm 23: "Uhr"




```
  10 BORDER 1: PAPER 1: INK 7: C
LS : GO SUB 2000
  20 INPUT AT 0,0;"Stunde=";h'"M
inute=";m
  30 LET m=m-1
  40 PLOT 83,92: DRAW 66*SIN (m/
30*PI),66*COS (m/30*PI)
  50 PLOT 83,92: DRAW 55*SIN ((h
+m/60)/6*PI),55*COS ((h+m/60)/6*
PI)
 100 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
 110 LET m=m+1: IF m=60 THEN LET
 m=0: LET h=h+1
```

```
 120 IF h=24 THEN LET h=0
 130 GO SUB 1000
 140 LET s=(PEEK 23672+256*PEEK
23673)/50: IF s>60 THEN GO TO 10
0
 150 PRINT AT 10,10; OVER 1;"0";
 PRINT AT 20,25;"0" AND s<10;INT
 s: GO TO 140
1000 PRINT AT 20,19;"0" AND h<10
;h;":";"0" AND m<10;m
1010 PLOT 83,92: DRAW OVER 1;66*
SIN ((m-1)/30*PI),66*COS ((m-1)/
30*PI)
1020 PLOT 83,92: DRAW 66*SIN (m/
30*PI),66*COS (m/30*PI)
1030 PLOT 83,92: DRAW OVER 1;55*
SIN ((h+(m-1)/60)/6*PI),55*COS (
(h+(m-1)/60)/6*PI)
1040 PLOT 83,92: DRAW 55*SIN ((h
+m/60)/6*PI),55*COS ((h+m/60)/6*
PI)
1050 RETURN
2000 RESTORE : FOR n=USR "a" TO
USR "a"+95: READ a: POKE n,a: NE
XT n
2010 FOR n=1 TO 12: PRINT AT 10-
9*COS (n/6*PI),10+9*SIN (n/6*PI)
;CHR$ (143+n): NEXT n
2020 FOR n=0 TO 5: CIRCLE 83,92,
n: NEXT n
2030 CIRCLE 83,92,82: CIRCLE 83,
92,68
2040 RETURN
3000 DATA 0,0,16,16,16,16,0,0,0,
10,10,10,10,0,0,0,0,42,42,42,4
2,0,0,0,0,0,0,85,85,85,85,0,0,40,4
0,16,16,0,0,0,0,0,0,42,42,18,18,
0,0,170,170,74,74,0,0,0,0,0,0,18
1,181,85,85,0,0,0,84,72,84,84,0,
0,0,80,32,80,80,0,0,0,0,0,42,18,
42,42,0,0,0,85,37,85,85,0,0,0
```

Wichtige Variablen:

H	Stundenzahl
M	Minutenzahl
S	Sekundenzahl

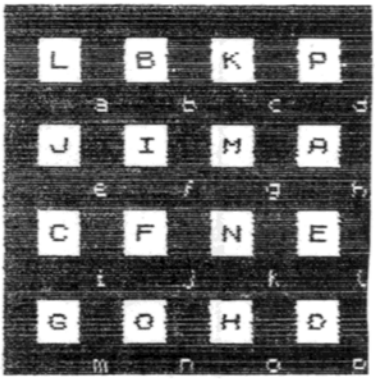
Nach dem LOADen zeichnet der SPECTRUM das Ziffernblatt. Dann sind nacheinander Stunden und Minuten einzugeben. Die Zeit läuft sofort nach Eingabe der Minuten !

Einige Hinweise zum Programm:

GOSUB 2000:	zeichnet das Ziffernblatt
GOSUB 1000:	zeichnet jede Minute neu Digitalanzeige und Zeiger
100:	23672, 23673 und 23674 zählen die Fernsehbilder, d. h. 50 pro Sekunde. Diese Zeilen setzt den Bildzähler auf Null.
140:	errechnet die Sekunden, die seit der Zeile 100 vergangen sind. Nach 60 Sekunden springt das Programm wieder zur Zeile 100, der Wert M wird um 1 erhöht.

Hinweis:
Durch geringfügiges Verändern des Wertes 60 in Zeile 140 kann die Ganggenauigkeit der Uhr beliebig eingestellt werden.

4.7 Programm 24: "Buchstaben-Puzzle"




```
Drehpunkt: ?

 10 REM BUCHSTABEN PUZZLE
 20 DIM B$(16)
 30 GO SUB 700
 40 PRINT "Einen Moment Geduld
    "
 50 FOR X=1 TO 200
 60 NEXT X
 70 BORDER 6: PAPER 6: INK 1: C
LS
 80 LET G=0
 90 LET M1=0
```

```
  100 GO SUB 600
  105 FOR I=1 TO 16
  110 LET T$=CHR$ (INT (16*RND+65
))
  120 FOR J=1 TO I
  130 IF B$(J)=T$ THEN GO TO 110
  140 NEXT J
  150 LET B$(I)=T$
  160 NEXT I
  180 LET S=0
  190 LET M=0
  195 GO SUB 400
  200 PRINT AT 18,0;"Drehpunkt: ?
"
  205 IF INKEY$="" THEN GO TO 205
  206 LET I$=INKEY$
  210 IF S AND I$="s" OR I$<"a" O
R I$>"k" AND I$<>"s" AND I$<>"q"
 OR I$="d" OR I$="b" THEN GO TO
205
  214 PRINT AT 18,11;I$
  220 IF I$="q" THEN GO TO 100
  230 IF I$="s" THEN GO TO 500
  235 LET I=CODE I$-96
  240 LET M=M+1
  250 LET T$=B$(I)
  255 LET B$(I)=B$(I+4)
  260 LET B$(I+4)=B$(I+5)
  265 LET B$(I+5)=B$(I+1)
  270 LET B$(I+1)=T$
  290 GO SUB 400
  300 FOR I=1 TO 16
  310 IF CHR$ (I+64)<>B$(I) THEN
GO TO 200
  320 NEXT I
  325 RESTORE : FOR I=1 TO 30: RE
AD A,B: BEEP A/7,B: NEXT I
  330 PRINT AT 18,0;"Du hast das
Puzzle in ";M;" Zuegen""geordne
t."
  340 LET M1=M1+M
  345 LET G=G+1
  350 PRINT "Ein neues Spiel? ";
  354 IF INKEY$<>"" THEN GO TO 35
4
  355 IF INKEY$="" THEN GO TO 355
  356 LET A$=INKEY$
  357 PRINT A$
  360 IF A$="j" THEN GO TO 100
  363 GO SUB 422
  365 PRINT AT 18,0;"Du spieltest
 ";G;" Spiel";"e" AND G>1,"und b
rauchtest durchschnittlich ";M1/
G;" Zuege."
  370 STOP
  400 FOR I=1 TO 13 STEP 4
  410 PRINT AT I+1,9;B$(I);AT I+1
,13;B$(I+1);AT I+1,17;B$(I+2);AT
 I+1,21;B$(I+3)
  420 NEXT I
  422 FOR X=17 TO 21
  424 PRINT AT X,0;"
```

```
 426 NEXT X
 430 RETURN
 500 PRINT AT 18,0;"Welche Posit
ionen sollen ver-    tauscht werd
en? ";
 505 IF INKEY$<>"" THEN GO TO 50
5
 506 IF INKEY$="" THEN GO TO 506
 510 LET I$=INKEY$
 512 IF I$<"a" OR I$>"p" THEN GO
 TO 505
 514 PRINT I$;"<>";
 515 IF INKEY$<>"" THEN GO TO 51
5
 516 IF INKEY$="" THEN GO TO 516
 517 LET J$=INKEY$
 520 IF J$<"a" OR J$>"p" THEN GO
 TO 516
 521 LET X=CODE I$-96
 522 LET Y=CODE J$-96
 524 IF X<>Y+1 AND X<>Y-1 OR Y<1
 OR Y>16 THEN GO TO 520
 525 PRINT J$
 530 LET S=S+1
 540 LET T$=B$(X)
 545 LET B$(X)=B$(Y)
 550 LET B$(Y)=T$
 560 GO TO 290
 600 FOR X=1 TO 13 STEP 4
 605 PRINT AT 0,7;"
 610 PRINT AT X,7;"
 620 PRINT INVERSE 1;AT X+3,11;C
HR$ (X+96);AT X+3,15;CHR$ (X+97)
;AT X+3,19;CHR$ (X+98);AT X+3,23
;CHR$ (X+99)
 630 NEXT X
 640 RETURN
 700 BORDER 6: PAPER 6: INK 2: L
ET R$="RGH-SOFTWARE PRAESENTIERT
:"
 710 FOR X=0 TO 21
 720 PRINT AT X,3; INVERSE 1;R$(
TO 13)
 730 PRINT AT 21-X,16; INVERSE 1
;R$(14 TO )
 740 NEXT X
 750 PRINT AT 8,0;"ABCDEFGHIJKL
MNOPABCDEFGHIJKLMNOP
                    BUCHSTA
BEN - PUZZLE
EFGHIJKLMNOPA
BCDEFGHIJKLMNOPABCD"
 752 FOR X=1 TO 200
 754 NEXT X
 760 FOR X=0 TO 21
 770 IF X<8 OR X>12 THEN PRINT A
T X,0;"
```

```
780 NEXT X
790 PRINT AT 17,0;
795 RETURN
800 DATA 3,7,1,5,1,4,1,7,1,12,1
,14,4,16,4,12,3,9,1,11,1,12,1,11
,1,12,1,9,4,7,4,4,3,7,1,5,1,4,1,
7,1,12,1,14,4,16,2,12,2,12,2,14,
2,14,2,12,2,11,6,12
```

Wichtige Variablen:

B$(16)	enthält die 16 Buchstaben in der je aktuellen Reihenfolge
G	Anzahl der Spiele
M1	Anzahl der Züge aller Spiele
M	Anzahl der Züge im letzten Spiel
S	Anzahl der "Sonderzüge" (0 oder 1)

Nach dem Vorspann und einer kurzen Zeit der Überlegung erscheint das 4 x 4 Buchstaben große Puzzle.

Ziel der Aktion ist es, die Buchstaben durch Verdrehen von Viererblöcken in alphabetischer Reihenfolge zu bringen. Es soll in 20 bis 25 Zügen möglich sein.

Das Drehen geschieht, indem die Taste des Zeichens gedrückt wird, um das sich die vier Buchstaben im Uhrzeigersinn drehen sollen (a, b, c, e, f, g, i, j oder k). Wenn Sie aufgeben und ein neues Puzzle beginnen wollen, drücken Sie bitte Taste "q".

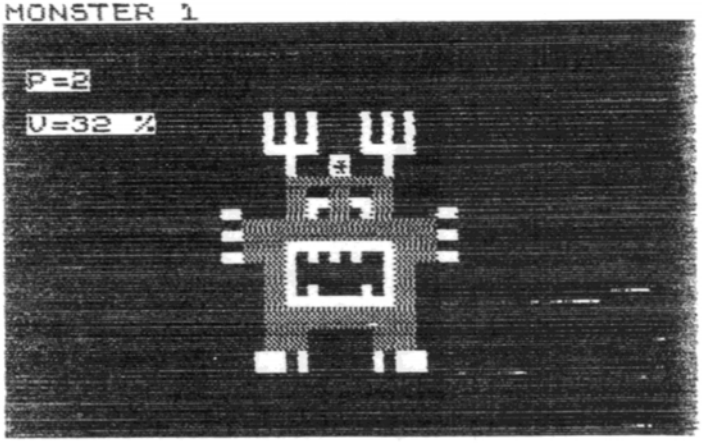
Durch Drücken von "s" können Sie einmal (!) während eines Puzzles einen Sonderzug durchführen, bei dem nicht gedreht wird, sondern zwei horizontal nebeneinanderliegende Buchstaben miteinander vertauscht werden. Für das Vertauschen ist das Zeichen rechts unter dem Buchstaben maßgebend. Die Zeichen beider zu vertauschender Zeichen sind einzugeben.

Wenn das Puzzle geordnet ist, teilt der SPECTRUM mit, wieviele Züge man gebraucht hat und fragt, ob man ein weiteres Puzzle machen möchte. Bei "N" gibt der SPECTRUM an, wieviele Puzzles man gelöst und wieviele Züge man durchschnittlich gebraucht hat.

4.8 Programm 25: "Das Monsterspiel"

```
Roland Monsterkiller
```

MONSTER 1




```
   5 BORDER 0: PAPER 5: INK 0: C
LS : GO SUB 6500
  10 PRINT "R&H-SOFTWARE PRAESEN
TIERT"
  15 PRINT '''  DAS MONSTE
RSPIEL "
  20 INPUT "Anzahl der Spieler?
(1-8) ";SPI
  30 IF SPI<1 OR SPI>8 THEN GO T
O 20
  40 PRINT ''SPI;" Spieler"
  43 INPUT "Kampfstaerke der Mon
ster? (0-20)  0 = Die Kampfstae
rke wird     per Zufallsgenerator
 zwischen    4 und 8 bei jedem Mo
nster neu    festgelegt. ";MS
  45 IF MS<0 OR MS>20 THEN GO TO
 43
  50 PRINT ''"Kampfstaerke: ";MS
: LET MS=MS*10: GO SUB 2000
  60 LET L=1: DIM V(SPI)
  70 FOR X=1 TO SPI: LET V(X)=10
: NEXT X
  80 DIM P(SPI): LET RUNDE=1
  90 PRINT ''"AUF GEHT'S.": GO TO
 104
 100 PRINT AT 1,0;"
    "''"DER NAECHSTE BITTE."
 101 LET L=L+1: GO TO 9920
 103 IF L>SPI THEN LET RUNDE=RUN
DE+1: LET L=1
 105 IF M(L)<RUNDE THEN GO TO 10
1
```

```
 107 IF V(L)>0 THEN GO TO 110
 109 GO TO 100
 110 PAUSE 0
 120 IF INKEY$<>"" THEN GO TO 12
0
 140 PRINT AT 0,0;N$(L)
 150 FOR X=3 TO 21
 160 PRINT AT X,0;"
                                "
 170 NEXT X
 200 PRINT AT 2,0;"MONSTER ";RUN
DE;"                "
 210 FOR J=1 TO 12
 220 PRINT PAPER 6;AT J+6,10;M$(
J): NEXT J
 230 LET ANGR=MS
 240 IF NOT MS THEN LET ANGR=RND
*50+30
 250 LET AN=0: LET X=10
 260 LET A=INT (RND*6)+1
 270 IF X<2 THEN LET V(L)=V(L)+.
1
 280 PRINT PAPER 6;AT 7,1;"V=";I
NT (10*V(L));" %"
 290 PRINT AT X(A),Y(A);"*"
 300 FOR X=1 TO 10: PAUSE 2
 310 IF A=1 THEN IF INKEY$="3" T
HEN GO TO 400
 315 IF A=2 THEN IF INKEY$="5" T
HEN GO TO 400
 320 IF A=3 THEN IF INKEY$="1" T
HEN GO TO 400
 325 IF A=4 THEN IF INKEY$="9" T
HEN GO TO 400
 330 IF A=5 THEN IF INKEY$="6" T
HEN GO TO 400
 335 IF A=6 THEN IF INKEY$="8" T
HEN GO TO 400
 340 BEEP .02,10: LET V(L)=V(L)-
.1
 350 IF V(L)<=0 THEN GO TO 9900
 360 NEXT X
 370 PRINT AT X(A),Y(A);"■"
 380 GO TO 260
 400 BEEP .2,-10: PRINT AT 11,14
; PAPER 6;"-■-"
 410 LET ANGR=ANGR-11+X
 420 LET P(L)=P(L)+10-X
 430 PRINT AT 5,1;"P=";P(L)
 440 LET AN=AN+1
 450 PRINT AT X(A),Y(A);"■"
 460 IF AN<ANGR THEN PRINT AT 11
,14; PAPER 7; BRIGHT 1;" "; PAPE
R 6; BRIGHT 0;"■"; PAPER 7; BRIG
HT 1;" "
 470 IF AN<ANGR THEN GO TO 260
 480 PRINT AT 13,13; PAPER 6;"■■
■■■";AT 14,13; PAPER 4;"     ";A
T 15,13; PAPER 6;"■■■■■"
 490 PRINT AT 7,11;"■■■■■■■■■■■";A
T 8,11;"■■■ ■■■ ■■■";AT 9,11;"■■■
■■■■"
```

```
 500 PRINT FLASH 1;AT 21,2;"DAS
MONSTER IST TOT."
 510 IF INKEY$<>"" THEN GO TO 51
0
 520 PAUSE 50
 530 GO TO 100
2000 DIM N$(SPI,32)
2005 DIM M(SPI)
2010 FOR X=1 TO SPI
2020 INPUT "Name des ";(X);". Sp
ielers?"'N$(X)
2050 INPUT "Wieviel Monster trau
t sich",(N$(X));"zu? (3-50) ";M(
X)
2070 IF M(X)<3 OR M(X)>50 THEN G
O TO 2050
2090 NEXT X
2100 RETURN
6500 DIM M$(12,11)
6510 LET M$(1)="
6511 LET M$(2)="
6512 LET M$(3)="
6513 LET M$(4)="
6514 LET M$(5)="
6515 LET M$(6)="
6516 LET M$(7)="
6517 LET M$(8)="
6518 LET M$(9)="
6519 LET M$(10)="
6520 LET M$(11)="
6521 LET M$(12)="
6530 DIM X(6)
6540 DIM Y(6)
6550 LET X(1)=9
6551 LET X(2)=12
6552 LET X(3)=12
6553 LET X(4)=14
6554 LET X(5)=18
6555 LET X(6)=18
6556 LET Y(1)=15
6557 LET Y(2)=9
6558 LET Y(3)=21
6559 LET Y(4)=15
6560 LET Y(5)=10
6561 LET Y(6)=20
6570 RESTORE : FOR i=0 TO 15: RE
AD a: POKE USR "a"+i,a: NEXT i
6580 RETURN
7000 DATA 170,65,170,85,170,85,1
70,85,170,85,170,85,255,255,255,
255
9500 CLS
9510 PRINT "Punkte:"
9520 PRINT
9530 FOR X=1 TO SPI
9535 IF P(X) THEN LET P(X)=P(X)+
INT V(X)*M(X)
9550 IF P(X) THEN PRINT N$(X,1 T
O 25);": ";P(X)
9560 NEXT X
9561 PRINT '''"Folgende Helden si
nd im Kampf   gegen Monster gefa
llen:"''
```

```
9562 FOR X=1 TO SPI
9563 IF NOT P(X) THEN PRINT N$(X
)
9564 NEXT X
9570 INPUT "Noch einmal unter gl
eichen       Bedingungen? ";Q$
9590 IF Q$<>"Ja" AND Q$<>"ja" AN
D Q$<>"j" THEN GO TO 9610
9595 CLS
9600 GO TO 70
9620 INPUT "Ein neues Spiel? ";Q
$
9630 IF Q$="Ja" OR Q$="ja" OR Q$
="j" THEN RUN
9640 STOP
9900 PRINT FLASH 1;AT 0,0;N$(L);
"IST TOT."
9901 LET P(L)=0
9902 FOR X=1 TO SPI
9903 IF V(X)>0 THEN GO TO 9910
9904 NEXT X
9905 GO TO 9500
9910 PAUSE 100: PAUSE 0: PAUSE 0
: GO TO 100
9920 FOR X=1 TO SPI
9922 IF M(X)>RUNDE THEN GO TO 99
30
9923 NEXT X
9924 GO TO 9500
9930 GO TO 103
```

Wichtige Variablen:

MS	Kampfstärke der Monster
SPI	Anzahl der Spieler
L	Spieler, der gerade dran ist
V(L)	Vitalität des Spielers (10 = 100%)
P(L)	Punkte des Spielers
M(L)	Monster, die sich ein Spieler zutraut
ANGR	Angriffsstärke des Monsters
AN	Anzahl der Angriffe des Monsters
M$(12,11)	enthält das Bild des Monsters
X(6),Y(6)	enthalten die Koordinaten der Angriffspunkte der Monster

Das gelistete Programm ist für den Anschluß der im 1. Kapitel beschriebenen Bedieneinheit mit 10 Tasten ausgelegt. Wenn Sie keine Bedieneinheit haben, ändern Sie bitte das Programm wie folgt um:

```
290 PRINT AT X(A),Y(A);A
300 FOR X=1 TO 10: PAUSE 2
```

```
310 IF A=1 THEN IF INKEY$="1" THEN GOTO 4
315       2                 2
320       3                 3
325       4                 4
330       5                 5
335       6                 6
```

(Nur die unterstrichenen Zeichen sind zu ändern !)

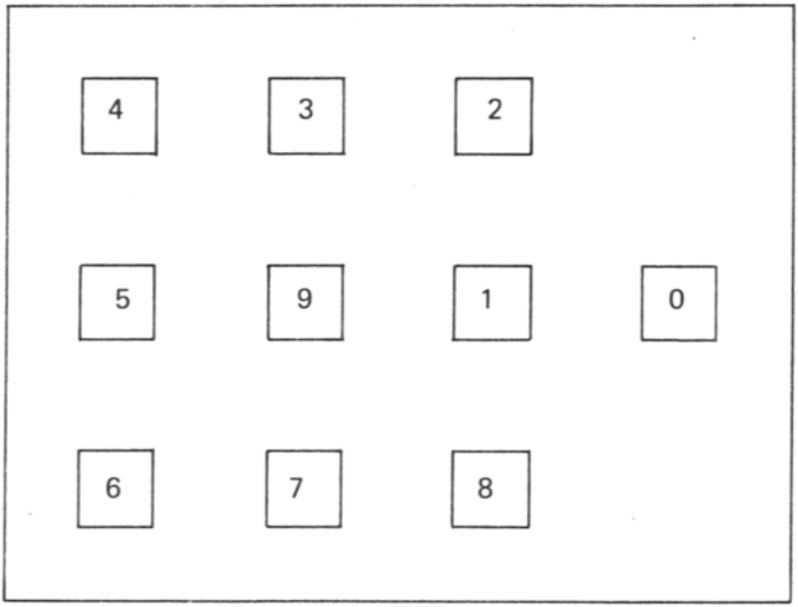
Nach dem Start erscheint die Frage "ANZAHL DER SPIELER?". Geben Sie eine Zahl zwischen 1 und 8 ein. Danach fragt der SPECTRUM nach dem Schwierigkeitsgrad. Der Schwierigkeitsgrad entsprciht der Anzahl der Schläge, die ein sehr guter Spieler braucht, um ein Monster zu erlegen. Nur excellente Spieler schaffen es manchmal schneller. Bei Kampfstärke 0 hat jedes Monster eine andere Kampfstärke, die vom Zufall bestimmt wird.

Nun fragt der SPECTRUM nach dem namen der Spieler und wieviele Monster sich jeder Spieler zutraut. Hier soll man sich nicht überschützen! Wenn man im Kampf unterliegt, weil man sich zu viel zugemutet hat, wird der Punktestand auf 0 gesetzt. Nur die Punkte der Überlebenden zählen! Wer sich übernommen hat, ist selber schuld. Ein guter Anfangswert sind 5 Monster im 5. Schwierigkeitsgrad. Mit Ausdauer und Übung kann man sich dann steigern.

Mit Taste "0" starten Sie nun das Spiel. Der SPECTRUM zeigt Ihren Namen an und wieviele Monster Sie angehen. Außerdem zeigt er Punkte (P) und Lebenskraft (V, in Prozent) an. Sinkt die Lebenskraft unter 0 scheidet man aus !

Und nun aufgepaßt:
Es erscheinen helle Sterne am Monster. Drücken Sie so schnell wie möglich die entsprechende Taste Ihrer Bedieneinheit (s. nächste Seite):

(Kopf = 3 / rechte Klaue = 1 / linke Klaue = 5 / rechter Fuß = 8 / linker Fuß = 6 / Maul = 9)



Je nach Raktionsschnelle mindern Sie Ihre Vitalität und schwächen die Stärke des Monsters. Je langsamer Sie sind, desto mehr Schläge benötigen Sie um das Monster zu erledigen. Entsprechend rapide verlieren Sie Ihre Lebensgeister. Daß ein Schlag getroffen hat, sieht man daran, daß es kurz die Augen schließt; ist es tötlich getroffen, schließt es Augen und Mund und läßt die Hörner hängen.

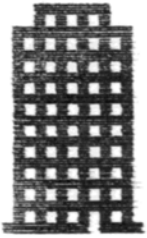
Wer keine Bedieneinheit hat und die Änderungen durchgeführt hat, drückt die Taste der Zahl, die am Monster erscheint.

Durch Drücken von "0" kommt der nächste Spieler dran. Wenn alle Spieler tot sind oder die vorgegebene Monsterzahl erreicht haben, listet der SPECTRUM den Punktestand der noch Lebenden und fragt, ob unter den gleichen Bedingungen noch einmal gespielt werden soll. Bei "Ja" geht das Spiel mit den gleichen Spielern und den gleichen Monstern von vorne los. Bei "Nein" fragt der SPECTRUM, ob man ein neues Spiel beginnen möchte.

4.9 Programm 26: "Bombendrohung"

RGH 03/82

BOMBENDROHUNG



IN DIESEM HAUS
LIEGT EINE BOMBE
VERBORGEN.
FINDEN SIE SIE,
BEVOR SIE EXPLO-
DIERT.

SIE HABEN NUR
WENIGE MINUTEN
ZEIT.

DAS HAUS IST 100*100*100
RAEUME (=KOORDINATEN JE 0 - 99)
GROSS.

```
   5 BORDER 6: PAPER 6: INK 0: C
LS
  10 PRINT "RGH 03/82",,,"
     BOMBENDROHUNG
                              "
  11 PRINT
  12 PRINT "     ","IN DIESEM
HAUS","     ","LIEGT EINE BOM
BE     ","VERBORGEN."
  13 PRINT "     ","FINDEN SI
E SIE,","     ","BEVOR SIE EX
PLO-     ","DIERT."
  14 PRINT "     ",,"     
","SIE HABEN NUR","     ","WE
NIGE MINUTEN","     ","ZEIT."
  15 PRINT "     ","DAS HAUS
 IST 100*100*100      RAEUME (
=KOORDINATEN JE 0 - 99) GROSS."
  18 PAUSE 500
  20 PAPER 7: CLS
  22 PRINT
  24 PRINT "GEBEN SIE DIE KOORDI
NATEN      L,T UND H EIN. DAS D
ETEKTOR-   SIGNAL SAGT IHNEN, O
B SIE SICH  DER BOMBE NAEHERN OD
ER NICHT."
  25 PRINT
  30 PRINT "IHRE KOORDINATEN STI
MMEN, WENN  DIE STAERKE DES SIGN
```

```
ALES 10000  IST. MERKEN SIE SICH
 DIE WERTE  UND GEBEN SIE UNTER
""L"" EIN ""X"" EIN. UNTEN ERSCH
EINT DER KURSOR ■. ENTSCHAERFEN
SIE DIE BOMBE,"
  35 PRINT "INDEM SIE ZIFFER FUE
R ZIFFER DIERICHTIGEN WERTE EING
EBEN. ABER  SCHNELL, DIE ZEIT LA
EUFT."
  40 PRINT
  45 PRINT "SCHWIERIGKEITSGRAD?"
,,,"■ = 200 SEK. MAXIMALE ZEIT.
        (WENN SIE DEN TRICK KENN
EN,     REICHEN 30 SEK.)"
  46 PRINT "■ = 400 SEK. MAXIMAL
E ZEIT."
  48 PAUSE 0
  50 LET I$=INKEY$
  55 IF I$<>"1" AND I$<>"2" THEN
 GO TO 48
  58 LET SW=VAL I$
  60 CLS
  70 LET A=INT (RND*99)+1
  71 LET B=INT (RND*99)+1
  72 LET C=INT (RND*99)+1
  80 LET D=0
  81 LET E=0
  82 LET F=0
  90 LET G=0
  92 LET X=1000
  94 LET K$=""
  95 INK 1: CLS
 100 PRINT "SEC  SIGNAL      L.
T.  H.
                "

 115 LET G=G+10
 117 PRINT AT (G-200*(G>200))/10
+1,16;
 119 INPUT "L,T,H: ";D: IF D<>X
THEN INPUT "L,T,H: ";(D);",";E;"
,";F
 120 IF D=X THEN GO TO 1000
 121 PRINT D;TAB 20;E;TAB 24;F
 122 IF SW=1 THEN LET SIGNAL=100
00-ABS ((A/100+B+C*100)-(D/100+E
+F*100))
 123 IF SW=2 THEN LET SIGNAL=100
00-57*SQR (ABS (A-D)↑2+ABS (B-E)
↑2+ABS (C-F)↑2)
 125 BEEP 1,SIGNAL/2000
 127 IF SW=1 THEN PRINT AT G/10+
1,0;G;TAB 4;SIGNAL
 128 IF SW=2 THEN PRINT AT (G-20
0*(G>200))/10+1,0;G;TAB 4;SIGNAL
 130 IF G=200 AND SW=1 OR G=400
AND SW=2 THEN GO TO 1000
 133 IF G>=200 THEN IF SW=2 THEN
 PRINT AT (G-200)/10+2,0;"
 135 GO TO 100
 140 PAPER 6: INK 0: CLS
```

```
 145 PRINT "DAS HAUS IST NACH ";
G+10*(G<200*SW);" SEKUNDEN","EXP
LODIERT.",,"DIE KOORDINATEN DER
BOMBE:       L=";A;" T=";B;" H=";
C
 150 PRINT
 152 PRINT "NOCH EIN SPIEL?"
 154 INPUT 0$
 156 IF 0$<>"Ja" AND 0$<>"ja" AN
D 0$<>"j" THEN STOP
 158 RUN 5
 170 PRINT AT 4,0;
 175 PRINT "HURRA.",,"DIE BOMBE
WURDE NACH",G;" SEKUNDEN AUFGESP
UERT."
 177 GO TO 150
 180 IF D>99 OR D<0 OR E>99 OR E
<0 THEN GO TO 200
 190 IF F>99 THEN GO TO 220
 195 IF F<0 THEN GO TO 230
 197 GO TO 110
 200 CLS
 210 PRINT "SIE LIEFEN SOEBEN IM
 ";F;".","STOCKWERK AUS DEM FENS
TER."
 215 PRINT "SIE SIND TOT UND IN
";200+200*(SW=2)-G,"SEKUNDEN EXP
LODIERT DAS HAUS."
 217 GO TO 150
 220 CLS
 222 PRINT "SIE BEFINDEN SICH ";
2.5*F;" METER","IN DER LUFT."
 225 GO TO 215
 230 CLS
 232 PRINT "SIE HABEN SICH ";-2.
5*F;" METER","IN DEN UNTERGRUND
EINGEGRABEN."
 235 GO TO 215
1000 PAPER 6: INK 0: CLS
1005 PRINT "DAS HAUS EXPLODIERT
IN 10 SEK."
1010 PRINT AT 3,10;"    "
1020 FOR X=4 TO 18
1030 PRINT TAB 10;"  "
1040 NEXT X
1050 PRINT TAB 10;"  "
1060 PRINT AT 3,24;"L T H"
1070 LET TI=300
1080 FOR J=1 TO 200
1085 PRINT #0;AT 0,0;"■"
1090 IF INKEY$<>"" THEN GO TO 11
30
1100 LET TI=TI-1
1110 PRINT INVERSE 1;AT 0,24;" "
;INT (TI/30)
1120 NEXT J
1125 GO TO 2000
1130 LET I$=INKEY$: IF INKEY$="0
" THEN LET K$="": PRINT AT 4,24;
"  "
1131 IF (I$<"0" OR I$>"9") AND I
$<>"." THEN GO TO 1090
```

```
1133 LET K$=K$+I$
1134 INPUT ""
1135 PRINT AT 4,24;K$
1140 IF LEN K$=6 THEN GO TO 1200
1150 GO TO 1100
1200 LET D=VAL K$(1 TO 2)
1205 LET E=VAL K$(3 TO 4)
1210 LET F=VAL K$(5 TO 6)
1220 IF D=A AND E=B AND C=F THEN
 GO TO 170
2000 PRINT AT 0,0;"
                                "
2005 FOR X=0 TO 50: BEEP 0.01,-R
ND*20: NEXT X
2006 FOR X=0 TO -5 STEP -1: BEEP
 0.01,X: NEXT X
2010 FOR I=1 TO 5: PLOT 120,120
2011 LET X=RND*256: LET Y=RND*15
8: DRAW OVER 1;X-120,Y-120: PLOT
 120,120: DRAW OVER 1;X-120,Y-12
0: NEXT I
2012 PAUSE 5
2014 PRINT AT 7,15;""
2020 PRINT AT 3,10;"        "
2025 FOR J=4 TO 17 STEP 2
2030 PRINT "                       "
;"                 "
2035 NEXT J
2040 PRINT AT 3,10;"        "
2045 FOR J=4 TO 9 STEP 2
2050 PRINT "                       "
;"                 "
2055 NEXT J
2060 FOR J=10 TO 17 STEP 2
2065 PRINT "
                          "
2070 NEXT J
2080 FOR J=3 TO 12
2085 PRINT AT J,12;"           "
2090 NEXT J
2094 PRINT AT 19,8;"
  "
2095 PRINT AT 18,9;"
 "
2096 PRINT AT 17,10;"
"
2097 PRINT AT 16,10;"            "
2098 PRINT AT 15,10;"           "
2099 PRINT AT 14,10;"           "
2100 PRINT AT 13,10;"           "
2110 FOR J=1 TO 10
2120 NEXT J: PAUSE 30
2130 PRINT AT 11,16;"   "
2131 PRINT AT 10,16;"     "
2132 PRINT AT 9,16;"     "
2133 PRINT AT 8,16;"     "
2135 PRINT AT 11,16;"     "
2136 PRINT AT 10,16;"      "
2137 PRINT AT 9,16;"     "
2138 PRINT AT 8,16;"     "
2140 PRINT AT 6,0;
2300 GO TO 145
```

Wichtige Variablen:

A, B, C	Die Koordinaten der Bombe
D, E, F	Die jeweiligen Suchkoordinaten
G	"Zeitmesser"
SW	Schwierigkeitsgrad

Der Sinn des Spieles ist es, in einem Haus von 100 x 100 x 100 Räumen (je 0 – 99) eine versteckte Bombe zu finden. Hilfe dabei bietet das Detektorsignal, das "10000" beträgt, wenn man den richtigen Raum gefunden hat. Wenn Sie die Bombe gefunden haben, indem Sie die richtigen Koordinaten eingegeben haben, müssen Sie die Bombe entschärfen. Merken Sie sich die Koordinaten und geben Sie unter "L" ein "X" ein.

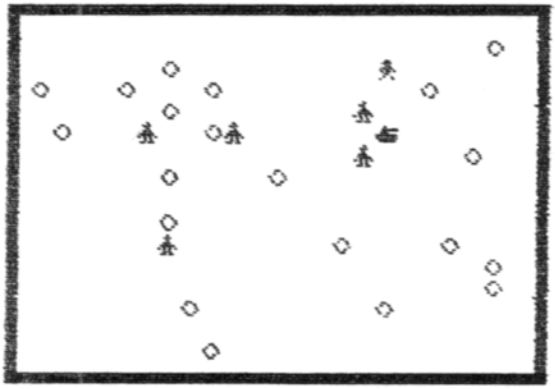
Sofort erscheint das Hochhaus und die Mitteilung, daß Sie noch 10 Sekunden Zeit haben. Ein invertiertes "?" erscheint als Kursor. Geben Sie nun Ziffer für Ziffer die Koordinaten der Bombe ein: Zuerst zweistellig (!0 den Wert "L", dann zweistellig den Wert "T" und anschließend zweistellig den Wert "H". Ruhe bewahren! Wenn der Kursor kurz verschwindet, hat der SPECTRUM die Zahl angenommen. Durch Drücken auf "C" können Sie von vorne beginnen. Sind Ihre Werte richtig, haben Sie das Haus gerettet, wenn nicht, explodiert die Bombe (Zeichentrick!) und das Haus stürzt zusammen.

Übrigens
Bei Schwierigkeitsgrad 1 ist ein Trick dabei. Wenn Sie ihn kennen, reicht ein Versuch, um die richtigen Koordinaten zu bestimmen. Bei Schwierigkeitsstufe 2 ist die Stärke des Detektorsignales nur von der tatsächlichen räumlichen Entfernung des Detektors von der Bombe abhängig.

4.10 Programm 27: "Minenfeld"

Bei diesem Programm können Sie die Maschinensprache-Routine "Inversion" aus diesem Buch benutzen. Wenn Sie die REM-Zeile mit dem Maschinenprogramm und Zeile 2 (LET INV=23760) mittels MERGE hinzugefügt haben, können Sie die REMs aus dem Programm entfernen, die vor dem "LET Q=USR INV" eingefügt sind.

```
432
5*1
678
```



```
Du hast eine Mine beruehrt.

DU HAST VERLOREN.
```

```
   7 BORDER 6: PAPER 6: INK 1: C
LS : GO SUB 1000
  10 PRINT "  MINENFELD
        RGH"
  11 PRINT
  12 PRINT "  = ANGREIFER"
  13 PRINT "O = MINE"
  15 PRINT "  = AMOKFAHRENDER PA
NZER"
  16 PRINT "  = DAS BIST DU."
  17 PRINT
  18 PRINT "Nach jedem meiner Zu
ege ziehst  Du, wobei "9" stehen
blei-"
  19 PRINT "ben bedeutet und "0"
 (max.","2 mal) Dich irgendwohin
","wirbelt."
  20 PRINT "Uebrige Zugrichtunge
n:"''
  21 PRINT "      4 3 2"
  22 PRINT
  23 PRINT "      5   1    [0]"
  24 PRINT
  25 PRINT "      6 7 8"
  26 PRINT
  27 PRINT "Schwierigkeitsgrad?
(1-9) ";
  30 PAUSE 0
  35 LET Q$=INKEY$
  40 IF Q$<"1" OR Q$>"9" THEN GO
 TO 30
  50 PRINT Q$: LET ZZ=161-20*VAL
 Q$
  60 IF ZZ=161 THEN LET ZZ=250
 100 CLS
 102 DIM A$(18,25)
```

```
105 DIM B(2,6)
107 GO SUB 800
110 LET A8=18
112 LET A9=25
120 FOR X=1 TO A9
130 LET B=INT (RND*A8)+1
140 LET C=INT (RND*A9)+1
150 LET A$(B,C)="0"
160 NEXT X
205 LET N3=0
210 LET N2=2
220 FOR D=1 TO A9
222 LET A$(1,D)="0"
224 LET A$(A8,D)="0"
226 NEXT D
230 FOR D=1 TO A8
232 LET A$(D,1)="0"
234 LET A$(D,A9)="0"
236 NEXT D
240 GO TO 270
250 LET H=INT (A8*RND)+1
252 LET I=INT (A9*RND)+1
260 IF A$(H,I)<>" " THEN GO TO
250
265 RETURN
270 FOR D=1 TO 7
272 GO SUB 250
275 LET A$(H,I)="♣"
277 IF D<>7 THEN GO TO 290
280 LET A$(H,I)="♣"
282 LET J=H
284 LET K=I
286 GO TO 310
290 IF D<>1 THEN GO TO 300
295 LET A$(H,I)="♣"
300 LET B(1,D)=H
305 LET B(2,D)=I
310 NEXT D
320 FOR D=2 TO A8-1
350 PRINT AT D-1,5;A$(D,2 TO A9
-1)
380 NEXT D
400 PRINT AT 18,0;"

                    "
402 LET Y=9
405 PRINT AT 20,0;"Dein Zug?
    "
406 FOR Z=1 TO ZZ
407 LET X$=INKEY$
408 IF X$<>"" THEN GO TO 412
409 NEXT Z
410 GO TO 414
412 BEEP .1,-5: IF X$>"." OR X$
<"A" THEN LET Y=VAL X$
414 PRINT AT 20,0;"          "
415 PRINT AT 18,0;
417 IF Y=9 THEN GO TO 520
420 LET U=J
421 LET W=K
425 IF Y>0 THEN GO TO 440
```

```
 427 IF N2<>0 THEN GO TO 490
 435 GO TO 400
 440 IF Y<>1 THEN GO TO 450
 442 LET J=0
 445 GO TO 460
 450 LET J=SGN (Y-5)
 460 IF Y>3 THEN GO TO 480
 462 IF Y=3 THEN GO TO 470
 465 LET K=1
 467 GO TO 510
 470 LET K=0
 475 GO TO 510
 480 LET K=SGN (Y-7)
 485 GO TO 510
 490 GO SUB 250
 495 LET A$(J,K)=" "
 496 PRINT AT J-1,K+3;" "
 497 LET A$(H,I)="☺"
 498 PRINT AT H-1,I+3; FLASH 1;"
☺"
 500 LET J=H
 501 LET K=I
 502 LET N2=N2-1
 505 GO TO 520
 510 LET K=K+V
 515 LET J=J+V
 520 PRINT AT J-1,K+3; FLASH 1;"
☺"
 521 IF Y=9 THEN GO TO 700
 522 PRINT AT V-1,W+3;" "
 524 IF A$(J,K)="O" THEN GO TO 7
50
 525 LET A$(V,W)=" "
 526 PRINT AT V-1,W+3;" "
 530 LET A$(J,K)="☺"
 531 PRINT AT J-1,K+3; FLASH 1;"
☺"
 535 GO TO 700
 540 IF X<>99 THEN GO TO 550
 545 RETURN
 550 LET V=X
 551 LET M=Y
 553 LET X=SGN (J-X)
 554 LET Y=SGN (K-Y)
 555 LET X=X+V
 556 LET Y=Y+M
 557 PRINT AT V-1,M+3;" "
 558 PRINT AT X-1,Y+3;"☻" AND D=
1;"☺" AND D>1
 560 IF A$(X,Y)="☺" THEN GO TO 6
90
 565 IF A$(X,Y)=" " THEN GO TO 6
80
 570 IF A$(V,M)="☻" THEN GO TO 6
40
 580 IF A$(X,Y)<>"☺" THEN GO TO
510
 582 GO SUB 250
 585 LET X=H
 587 LET Y=M
 590 FOR I=1 TO 5: BEEP .01,10:
NEXT I: REM LET Q=USR INV
```

```
  592 REM LET Q=USR INV
  600 LET A$(V,M)=" "
  601 PRINT AT V-1,M+3;" "
  602 LET A$(X,Y)="♣"
  603 PRINT AT X-1,Y+3;"♣"
  604 RETURN
  610 LET A$(V,M)=" "
  612 IF A$(X,Y)="♣" THEN GO TO 6
20
  615 LET A$(X,Y)=" "
  616 PRINT AT X-1,Y+3;":"
  620 GO SUB 630
  625 LET X=99
  627 RETURN
  630 FOR I=1 TO 5: BEEP .01,10:
NEXT I: REM LET Q=USR INV
  634 REM LET Q=USR INV
  638 LET N3=N3+1
  639 RETURN
  640 IF A$(X,Y)<>"♣" THEN GO TO
680
  645 LET M9=2
  650 IF B(1,M9)<>X THEN GO TO 66
0
  655 IF B(2,M9)=Y THEN GO TO 670
  660 LET M9=M9+1
  665 GO TO 650
  670 LET B(1,M9)=99
  675 GO SUB 630
  680 LET A$(X,Y)=A$(V,M)
  682 LET A$(V,M)=" "
  683 PRINT AT V-1,M+3;" "
  685 RETURN
  690 FOR I=0 TO 30: BEEP .01,I:
NEXT I
  692 REM LET Q=USR INV
  693 REM LET Q=USR INV
  694 REM LET Q=USR INV
  695 REM LET Q=USR INV
  696 PRINT FLASH 1;AT 19,1;"Ich
habe Dich erwischt."
  697 REM LET Q=USR INV
  698 GO TO 760
  700 FOR D=1 TO 6
  702 LET X=B(1,D)
  704 LET Y=B(2,D)
  706 GO SUB 540
  710 LET B(1,D)=X
  712 LET B(2,D)=Y
  715 NEXT D
  720 IF N3<>5 THEN GO TO 400
  730 FOR I=0 TO 5: BEEP .3,I: NE
XT I
  735 PRINT AT 19,1;"Alle Angrei
er sind geschlagen. Du hast gewo
nnen.                "
  736 REM LET Q=USR INV
  740 GO TO 770
  750 FOR I=1 TO 10: BEEP .01,40:
 NEXT I: REM LET Q=USR INV
  751 REM LET Q=USR INV
  752 REM LET Q=USR INV
```

```
  753 REM LET Q=USR INV
  758 PRINT FLASH 1;AT 19,1;"Du h
ast eine Mine beruehrt."
  760 PRINT FLASH 1; BRIGHT 1;AT
21,1;"DU HAST VERLOREN. "
  765 REM LET Q=USR INV
  770 IF INKEY$<>"" THEN GO TO 77
0
  772 PRINT #0;"Noch ein Spiel? (
Ja="9")"
  775 PAUSE 0: LET Q$=INKEY$
  780 IF Q$="9" THEN GO TO 90
  790 IF INKEY$<>"" THEN GO TO 79
0
  792 PRINT #0;AT 0,0;"Willst Du
einen anderen    "'"Schwierigkeit
sgrad? (Ja='9')"
  793 PAUSE 0: LET Q$=INKEY$
  795 IF Q$="9" THEN RUN
  798 STOP
  800 PRINT AT 0,4;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
  810 FOR X=1 TO 16
  820 PRINT AT X,4;"▌";AT X,28;"▐
"
  830 NEXT X
  840 PRINT AT 17,4;"▌▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▐"
  850 PRINT AT 0,0;"432"
  860 PRINT "5*1"
  870 PRINT "678"
  880 RETURN
1000 RESTORE : FOR I=0 TO 7: REA
D A: POKE USR "D"+I,A: NEXT I
1010 FOR I=0 TO 7: READ A: POKE
USR "M"+I,A: NEXT I
1020 FOR I=0 TO 7: READ A: POKE
USR "P"+I,A: NEXT I
1030 FOR I=0 TO 7: READ A: POKE
USR "A"+I,A: NEXT I
1040 RETURN
1050 DATA 24,60,24,60,90,24,36,6
6,0,24,36,66,66,36,24,0,0,16,63,
56,255,255,126,0,16,56,16,56,124
,170,40,108
```

Nach dem Start zeigt der SPECTRUM zunächst einmal die Spielanleitung und erklärt die Zeichen.

Die Zugrichtungen sind so gewählt, das das Programm über die Bedieneinheit (1. Kapitel) bequem zu steuern ist.

Zunächst geben Sie den Schwierigkeitsgrad ein. Bei 1 haben Sie sehr viel Zeit, Ihren Zug in eine der acht Richtungen zu überlegen, während es bei 9 sehr rasch geht. Der SPECTRUM wartet nicht, bis Sie einen Zug

gemacht haben, sondern zieht selbsttätig, wenn Sie zu lange brauchen. Nach der Eingabe erscheint das Spielfeld.

Panzer und Angreifer bewegen sich auf Sie zu. Weichen Sie so aus, daß die Angreifer unweigerlich über Minen laufen müssen. Dann werden Sie vernichtet und hinterlassen ein ":". Der Panzer wird niemals vernichtet, läuft er aber gegen einen Angreifer, hat dieser das Nachsehen. Der SPECTRUM hat gewonnen, wenn ein Angreifer oder der Panzer Sie erreicht hat. Sie haben gewonnen, wenn nur noch der Panzer übrig ist. Treten Sie nicht auf Minen! Auch der Spielfeldrand ist vermint !

Übrigens:
Zweimal können Sie durch Drücken von "0" verschwinden, um an einen Zufallsort wieder aufzutauchen. Ob es Ihnen dann besser geht, kann niemand vorher wissen.

4.11 Programm 28: "ZX-Sector"

```
> SECTOR <*

ZX-SECTOR wird nach den ueb-
lichen Regeln gespielt,
jedoch kennt der SPECTRUM nur
eine Tiefe. Dafuer wird das
U-Boot nur selten beim 1.Treffer
100%ig zerstoert.

4   3   2          0:-Schiffsbewegung
     N                -Schuss
                      -naechster
5W   +  O1             Spieler
                     9:Erhoehung von
                       -Tempo
     S                 -Schussweite
6   7   8
```

```
   2>LET inv=23760
   5 BORDER 1: PAPER 1: INK 7: C
LS
  10 PRINT "> SECTOR <*"
  20 PRINT
  30 PRINT "ZX-SECTOR wird nach
den ueb-     lichen Regeln gespie
lt,"
  40 PRINT "jedoch kennt der SPE
CTRUM nur    eine Tiefe. Dafuer w
ird das"
```

```
  50 PRINT "U-Boot nur selten be
im 1.Treffer100%ig zerstoert."
  55 PRINT
  60 PRINT "4    3    2      0:-Sch
iffsbewegung"
  61 PRINT "     ▓              -Sch
uss"
  62 PRINT "     ■              -nae
chster"
  63 PRINT "     ▟▙              Spi
eler"
  64 PRINT "5▓   ▜▛   ▓1"
  65 PRINT "     ▀              9:Erho
ehung von"
  66 PRINT "     ▓              -Tem
po"
  67 PRINT "     ▓              -Sch
ussweite"
  68 PRINT "6    7    8"
  70 PRINT "                    * Eing
etr. Warenz."
  80 PRINT
 100 INPUT "Wieviele Spieler? ";
SP
 120 IF SP<1 OR SP>4 THEN GO TO
100
 140 INPUT "Schwierigkeitsgrad (
1-3)? ";SW
 150 IF SW<1 OR SW>3 THEN GO TO
140
 160 DIM N$(SP,10)
 161 CLS
 162 PRINT "Geben Sie die Namen
der Spieler ein."
 163 FOR X=1 TO SP
 164 PRINT "X;". Spieler(in)?",
 165 INPUT LINE N$(X)
 166 PRINT N$(X)
 167 NEXT X
 170 CLS
 180 DIM X(4)
 190 DIM Y(4)
 200 DIM P(SP)
 205 DIM R(SP)
 210 DIM T(SP)
 220 DIM R$(SP,9)
 230 LET S=0
 240 LET X(1)=0
 250 LET X(2)=0
 260 LET X(3)=5
 270 LET X(4)=10
 280 LET Y(1)=10
 290 LET Y(2)=5
 300 LET Y(3)=0
 310 LET Y(4)=0
 320 LET AD=0
 330 LET SOS=0
 350 RANDOMIZE
 355 LET X=INT (RND*28)+7
 360 LET Y=INT (RND*28)+7
 370 LET RU=INT (RND*8)
 380 LET P=0
```

```
 400 LET S=S+1
 402 CLS
 405 IF SU=3 THEN GO SUB 2800
 410 IF S>SP THEN LET S=1
 412 IF X>34 THEN LET RU=RU+1
 413 IF Y>34 THEN LET RU=RU+3
 414 IF X<6 THEN LET RU=RU-1
 415 IF Y<6 THEN LET RU=RU-3
 420 IF RU=1 OR RU=6 OR RU=4 THE
N LET X=X+1
 430 IF RU=2 OR RU=7 OR RU=5 THE
N LET X=X-1
 440 IF RU=0 OR RU=1 OR RU=2 THE
N LET Y=Y+1
 450 IF RU=3 OR RU=4 OR RU=5 THE
N LET Y=Y-1
 495 PRINT AT 0,0;
 500 PRINT "** ";
 502 GO SUB 2200
 505 PRINT N$(S)
 510 PRINT
 520 GO SUB 2000
 530 PRINT "Standort:",X(S);";";
Y(S)
 540 PRINT "Fahrtrichtung:",R$(S
)
 550 PRINT "Tempo:",T(S)
 560 PRINT "Entfernung:",E
 570 PRINT
 580 PRINT "Neue Fahrtrichtung?
";
 590 GO SUB 2100
 591 LET RR=R(S)
 592 LET R(S)=R
 594 LET DR=ABS (RR-R(S))
 595 IF DR>4 THEN LET DR=8-DR
 597 IF DR>4-INT (P(S)/200) THEN
 GO TO 2700
 600 LET R$(S)=X$
 605 PRINT AT 20,0;"
                    "
 607 PRINT AT 7,20;
 610 PRINT R$(S)
 620 PRINT "Neue Geschwindigkeit
? ";T(S)
 625 LET TT=T(S)
 630 IF INKEY$<"9" AND INKEY$<>
"0" THEN GO TO 630
 640 IF INKEY$="0" THEN GO TO 69
0
 650 LET T(S)=T(S)+1
 655 IF T(S)>9 THEN LET T(S)=0
 660 PRINT AT 8,22;T(S)
 670 IF INKEY$<>"" THEN GO TO 67
0
 680 GO TO 630
 690 LET DT=T(S)-TT
 692 IF DT>0 AND DT>16-P(S)/50 T
HEN GO TO 2500
 694 IF DT<0 AND -DT>9-P(S)/100
THEN GO TO 2600
```

```
 695 PRINT AT 20,0;"
                    "
 697 PRINT AT 8,0
 700 FOR T=1 TO T(S)
 705 IF R<3 OR R>7 THEN LET X(S)
=X(S)+1
 710 IF R>3 AND R<7 THEN LET X(S
)=X(S)-1
 720 IF R>1 AND R<5 THEN LET Y(S
)=Y(S)+1
 725 IF R>5 THEN LET Y(S)=Y(S)-1
 727 GO SUB 2300
 728 IF SOS=1 THEN GO TO 735
 730 NEXT T
 735 PRINT "Neuer Standort: ";X(
S);";";Y(S)
 740 GO SUB 2000
 750 PRINT "Neue Entfernung: ";E
 760 PRINT
 765 IF SOS=1 THEN GO TO 1100
 770 PRINT "Schussrichtung? ";
 780 GO SUB 2100
 790 PRINT X$
 800 LET W=0
 810 PRINT "Schussweite? ";W
 820 IF INKEY$<>"9" AND INKEY$<>
"0" THEN GO TO 820
 825 IF INKEY$="0" THEN GO TO 90
0
 830 LET W=W+1
 835 IF W>2 THEN LET W=0
 840 PRINT AT 13,13;W
 850 IF INKEY$<>"" THEN GO TO 85
0
 860 GO TO 820
 900 LET XX=X(S)
 902 LET YY=Y(S)
 905 IF R<3 OR R>7 THEN LET XX=X
(S)+W
 910 IF R>3 AND R<7 THEN LET XX=
X(S)-W
 920 IF R>1 AND R<5 THEN LET YY=
Y(S)+W
 930 IF R>5 THEN LET YY=Y(S)-W
 935 FOR I=69 TO 45 STEP -.3: BE
EP .05,I: NEXT I
 940 IF XX=X AND YY=Y THEN GO TO
 1000
 950 PRINT
 970 LET P(S)=P(S)-100
 980 GO TO 1100
1000 LET PX=INT (RND*90)+11
1005 IF 100-P<PX THEN LET PX=100
-P
1010 LET P(S)=P(S)+PX
1020 LET P=P+PX
1025 FOR I=0 TO 20: BEEP .01,-10
: BEEP .01,-5: BEEP .01,-60: NEX
T I
1030 LET Q=USR INW
```

```
1031 LET 0=USR INV
1032 LET 0=USR INV
1033 LET 0=USR INV
1035 PRINT
1040 IF P=100 THEN CLS
1043 PRINT PX;"%iger "; FLASH 1;
"Treffer";: PRINT "."
1046 PRINT "Das U-Boot ist zu"
1047 PRINT P;" % zerstoert."
1050 IF P<100 THEN GO TO 1100
1055 PRINT "Es existiert nicht m
ehr."
1056 PRINT
1058 LET SS=S
1060 FOR S=1 TO SP
1061 PRINT
1062 GO SUB 2200
1063 PRINT N$(S);TAB 20;P(S);" P
unkte"
1064 IF P(S)>=750 THEN LET AD=1
1065 NEXT S
1066 LET S=SS
1068 PRINT
1070 IF AD>0 THEN GO TO 1085
1071 PRINT "Wollen Sie noch ein
U-Boot auf- spueren?"
1075 INPUT 0$
1077 IF 0$="Ja" OR 0$="ja" OR 0$
="j" THEN GO TO 350
1080 CLS
1085 PRINT "Ein neues Spiel?"
1090 INPUT 0$
1092 IF 0$="Ja" OR 0$="ja" OR 0$
="j" THEN RUN
1096 STOP
1100 IF INKEY$="" THEN GO TO 110
0
1105 IF SOS=0 AND SW>1 THEN IF X
>5 AND X<35 AND Y>5 AND Y<35 THE
N LET RU=INT (RND*8)
1107 LET SOS=0
1110 GO TO 400
2000 LET EX=ABS (X(S)-X)
2010 LET EY=ABS (Y(S)-Y)
2020 LET E=EX
2030 IF EY>E THEN LET E=EY
2040 RETURN
2100 LET R=CODE INKEY$-48
2110 IF R<0 OR R>8 THEN GO TO 21
00
2115 IF R=9 THEN GO TO 2100
2120 IF R=0 THEN GO TO 400
2130 IF R=1 THEN LET X$="Ost"
2132 IF R=2 THEN LET X$="Nord-Os
t"
2134 IF R=3 THEN LET X$="Nord"
2136 IF R=4 THEN LET X$="Nord-We
st"
2138 IF R=5 THEN LET X$="West"
2140 IF R=6 THEN LET X$="Sued-We
st"
2142 IF R=7 THEN LET X$="Sued"
```

```
2144 IF R=8 THEN LET X$="Sued-Os
t"
2150 IF INKEY$<>"" THEN GO TO 21
50
2160 RETURN
2200 IF P(S)<0 THEN PRINT "Matro
se ";
2210 IF P(S)>=0 AND P(S)<150 THE
N PRINT "Seekadett ";
2220 IF P(S)>=150 AND P(S)<300 T
HEN PRINT "Faehnrich z.S. ";
2230 IF P(S)>=300 AND P(S)<450 T
HEN PRINT "Leutnant z.S. ";
2240 IF P(S)>=450 AND P(S)<600 T
HEN PRINT "Kapitaenleutnant ";
2250 IF P(S)>=600 AND P(S)<750 T
HEN PRINT "Kapitaen z.S. ";
2260 IF P(S)>=750 THEN PRINT "Ad
miral ";
2270 RETURN
2300 FOR J=1 TO SP
2310 IF J<>S AND X(S)=X(J) AND Y
(S)=Y(J) THEN GO TO 2350
2320 NEXT J
2330 RETURN
2350 LET 0=USR INV+USR INV
2351 FOR I=1 TO 10: BEEP RND/10,
10*RND: NEXT I
2355 PRINT FLASH 1; BRIGHT 1;"##
##OLLISION ##"
2357 LET P(S)=P(S)-50
2360 LET X(S)=X(S)+INT (RND*T(S)
)
2370 LET Y(S)=Y(S)+INT (RND*T(S)
)
2380 LET SOS=1
2390 GO TO 2300
2500 PRINT AT 20,0;"In Ihrer Pos
ition sollte man    etwas besche
idener sein.(max. ";INT (16-P(S)
/50);")"
2510 GO TO 630
2600 PRINT AT 20,0;"Soviel Verzo
egerung vertraegt   Ihr Schiff n
icht mehr.  (max. ";INT (9-P(S)/
100);")"
2610 GO TO 630
2700 PRINT AT 20,0;"Wollen Sie k
entern? Ihre Wende- moeglichkeit
 ist max. ";(4-INT (P(S)/200))*4
5;" Grad."
2710 GO TO 590
2800 IF RND*100>2 THEN RETURN
2805 IF RND*100<1 THEN LET RU=RU
+1
2810 IF RND*100>=1 THEN LET RU=R
U-1
2820 IF RU<0 THEN LET RU=7
2830 IF RU>7 THEN LET RU=0
2840 PRINT AT 20,0; FLASH 1;"WAR
NUNG: U-BOOT AENDERT GERING- FUE
GIG SEINE FAHRTRICHTUNG."
2850 RETURN
```

Auch dieses Programm benutzt eine Maschinensprache-Routine, um mehr Action zu erzeugen. Es handelt sich um das Maschinenprogramm "Inversion" im 3. Kapitel dieses Buches. Fügen Sie es mittels MERGE hinzu !

Ja, es gibt das bekannte SECTOR nun auch für den SPECTRUM in einer sehr reizvollen Version des bekannten Computerspieles:
Die Spielregeln ähneln sehr denen von "MiniSector II" (2. Kapitel).

Das Spielfeld ist 40 mal 40 Felder groß, wobei es für die Schiffe auch möglich ist, außerhalb dieses Spielfeldes zu operieren. Den Schiffen sind keine Grenzen gesetzt !

Der Bereich, in dem sich das U-Boot bewegt, ist eingegrenzt: Es bewegt sich zwischen 5;5 und 35;35. Nach jedem Zug eines Spielers bewegt sich das U-Boot um 1 Feld in eine der acht Richtungen. Kommt es an eine Grenze, d. h. wird eine Koordinate kleiner als 5 oder größer als 35, dann wird es reflektiert.

Spielverlauf:
Bis zu vier Spieler können mitspielen. Jeder Spieler beginnt als "Seekadett" und hat die Möglichkeit "Admiral" zu werden. Ist ein Spieler Admiral, endet das Spiel. Der Anfangsstandort eines jeden Spielers zu Beginn des Spieles ist immer gleich:

Spieler 1 :	0 ; 10
Spieler 2 :	0 ; 5
Spieler 3 :	5 ; 0
Spieler 4 :	10 ; 0

Bei diesen Angaben bezeichnet die erste Ziffer die östliche Länge und die zweite die nördliche Breite.

Um sich zurechtzufinden, fertigen Sie am besten Seekarten an (ähnlich, wie bei MiniSector), die 40 mal 40 Felder groß sind. Der Punkt 0;0 ist dann links unten, 40;40 links oben.

Der SPECTRUM zeigt für jeden Spieler Fahrtrichtung, Tempo. Standort

und Entfernung zum U-Boot an. Zur Entfernungsangabe vgl. die Beschreibung von "MiniSector". Danach fragt der SPECTRUM nach der neuen Fahrtrichtung, die direkt über die Bedieneinheit aus Kapitel 1 eingegeben werden kann. Ansonsten ist die entsprechende Zahl zu drücken:

OST:	1
NORDOST:	2
NORD:	3
NORDWEST:	4
WEST:	5
SÜDWEST:	6
SÜD:	7
SÜDOST:	8

Danach wird nach der Geschwindigkeit gefragt. Ist man mit der, die man hat, zufrieden, drückt man nur auf 0. Ansonsten erhöht man sie, indem man auf 9 drückt jeweils um 1, wobei nach Geschwindigkeit 9 wieder Geschwindigkeit 0 kommt. Hat man die gewünschte Geschwindigkeit, drückt man auf 0.

Nun fährt das Schiff. Der SPECTRUM überprüft, daß man nicht mit einem anderen Schiff kollidiert. Das gibt dann Strafpunkte und evtl. sogar eine Degradierung. Außerdem wird das eigene Schiff ein wenig herumgewirbelt. Also aufgepaßt, wo die Mitspieler sind !

Anschließend zeigt der SPECTRUM den neuen Standort an und die neue Entfernung. Dann fragt er nach der Schußrichtung. Ist man nicht in Schußweite, so drückt man auf 0 und der nächste Spieler ist dran. Wenn man aber in Schußweite ist, d. h. die Entfernung zum U-Boot 2 oder weniger ist, kann man auf das U-Boot schießen.

Achtung:
Bei Entfernung zwei kann man nur schießen, wenn das U-Boot in einer der acht möglichen Richtungen zum eigenen Schiff steht.

Hat man die Schußrichtung eingegeben, fragt der SPECTRUM nach der Schußweite. Diese wird, wie die Geschwindigkeit, mittels der Taste "9"

eingestellt. Stimmt die Weite kann man schießen, indem man auf 0 drückt. Wenn man nicht getroffen hat (weil man sich bezüglich des Standortes des U-Bootes irrte), verliert man 100 Punkte, was häufig zur Degradierung führt.

Trifft man, so wird das U-Boot zerstört. Es wird pro Schuß zwischen 11 und 100% zerstört. Wenn es noch nicht zu 100% zerstört ist, ändert es die Bewegungsrichtung und fährt in irgendeine Richtung weiter. Es muß weiter verfolgt werden. Ist ein U-Boot zu 100% zerstört, fragt der SPECTRUM, ob ein weiteres U-Boot gesucht werden soll. Dieses taucht dann irgendwo auf.

Das Spiel geht so lange, bis einer der Mitspieler Admiral ist.

Aber Aufgepaßt:
Je größer die Punktezahl, d. h. je Höher der Dienstgrad, desto größer werden die Schiffe, die Sie befehligen. Das hat zur Folge, daß Ihre Manöverfähigkeit nachläßt:

Punkte	Beschleunigung	Abbremsen	Wenden
bis 0	9	9	180°
1 – 100	9	8	180°
101 – 200	9	7	180°
201 – 300	9	6	180°
301 – 350	9	5	135°
351 – 400	8	5	135°
401 – 450	7	4	90°
451 – 500	6	4	90°
501 – 550	5	3	90°
551 – 600	4	3	90°
601 – 650	3	2	45°
651 – 700	2	2	45°
701 . . .	1	1	45°

Dienstgrade in Abhängigkeit von der Punktezahl:

Punkte	Dienstgrad
bis −1	Matrose
0 – 149	Seekadett
150 – 299	Fähnrich zur See
300 – 449	Leutnant zur See
450 – 599	Kapitänleutnant
600 – 749	Kapitän zur See
750 . . .	Admiral

Für jeden Treffer bekommt man soviel Punkte, wie das U-Boot prozentual zerstört wurde. Ist das U-Boot zu 34% getroffen worden, bekommt man 34 Punkte. Bei einer Kollision mit einem anderen Schiff verliert man 50 Punkte, bei einem Fehlschuß 100 Punkte.

Das Spiel kennt drei Schwierigkeitsgrade:
Im ersten Schwierigkeitsgrad stört es das U-Boot überhaupt nicht, wenn es getroffen wird. Es ändert die Richtung nicht. Im zweiten Schwierigkeitsgrad ändert es nach jedem Treffer total die Richtung. Den Wert der Änderung bestimmt der Zufallsgenerator. Im dritten Schwierigkeitsgrad ändert das U-Boot auch von alleine ab und zu (etwa jeden 50. Zug) seine Richtung um max. 45 Grad. Die Richtungsänderung wird durch "WARNUNG: U-BOOT AENDERT GERINGFUEGIG SEINE FAHRTRICHTUNG" anzeigt.

4.12 Programm 29: "Labyrint" (48K)

```
  10 REM : LABYRINT
  20 REM
  30 REM : ROLAND G. HUELSMANN
  40 REM
  50 CLEAR 63999
  60 BORDER 1: PAPER 1: INK 7: C
LS
  70 PRINT AT 8,6; INK 4;"
";AT 10,6; INK 5; FLA
SH 1;"L A B Y R I N T H";AT 12,6
; FLASH 0; INK 4;"
"
  80 GO SUB 2000
  90 RANDOMIZE
 100 LET c$="+"
```

```
 120 PRINT ,,"LABYRINT-GROESSE?
(1-9)"
 121 PAUSE 0
 122 PAUSE 0
 123 LET I$=INKEY$
 124 IF I$<"1" OR I$>"9" THEN GO
TO 121
 125 LET H=VAL I$+5
 126 LET V=VAL I$+2: IF V=11 THE
N LET V=10
 130 CLS : PRINT #0;"Bitte einen
 Moment Geduld ..."
 131 LET K$="1"
 132 LET PT=0
 133 LET E=2
 134 LET SCH=1
 135 PRINT AT 0,0;"+";TAB 2*H;"+
"
 140 DIM W(15,11)
 150 DIM B(15,11)
 155 PRINT AT 1,0;
 160 LET Q=0
 161 LET Z=0
 162 LET X=INT (RND*H+1)
 163 FOR I=1 TO H
 170 IF I=X THEN GO TO 173
 171 PRINT "++";
 172 GO TO 180
 173 PRINT "+-";
 180 NEXT I
 190 PRINT "+"
 191 LET C=1
 192 LET W(X,1)=C
 193 LET C=C+1
 200 LET R=X
 201 LET S=1
 202 GO TO 260
 210 IF R<>H THEN GO TO 240
 211 IF S<>V THEN GO TO 230
 220 LET R=1
 221 LET S=1
 222 GO TO 250
 230 LET R=1
 231 LET S=S+1
 232 GO TO 250
 240 LET R=R+1
 250 IF W(R,S)=0 THEN GO TO 210
 260 IF R=1 THEN GO TO 530
 261 IF W(R-1,S)<>0 THEN GO TO 5
30
 270 IF S=1 THEN GO TO 390
 280 IF W(R,S-1)<>0 THEN GO TO 3
90
 290 IF R=H THEN GO TO 330
 300 IF W(R+1,S)<>0 THEN GO TO 3
30
 310 LET X=INT (RND*3+1)
 320 IF X=1 THEN GO TO 790
 321 IF X=2 THEN GO TO 820
 322 IF X=3 THEN GO TO 860
 330 IF S<>V THEN GO TO 340
 331 IF Z=1 THEN GO TO 370
 332 LET Q=1
```

```
 333 GO TO 350
 340 IF W(R,S+1)<>0 THEN GO TO 3
70
 350 LET X=INT (RND*3+1)
 360 IF X=1 THEN GO TO 790
 361 IF X=2 THEN GO TO 820
 362 IF X=3 THEN GO TO 910
 370 LET X=INT (RND*2+1)
 380 IF X=1 THEN GO TO 790
 381 IF X=2 THEN GO TO 820
 390 IF R=H THEN GO TO 470
 400 IF W(R+1,S)<>0 THEN GO TO 4
70
 401 IF S<>V THEN GO TO 420
 410 IF Z=1 THEN GO TO 450
 411 LET Q=1
 412 GO TO 430
 420 IF W(R,S+1)<>0 THEN GO TO 4
50
 430 LET X=INT (RND*3+1)
 440 IF X=1 THEN GO TO 790
 441 IF X=2 THEN GO TO 860
 442 IF X=3 THEN GO TO 910
 450 LET X=INT (RND*3+1)
 460 IF X=1 THEN GO TO 790
 461 IF X=2 THEN GO TO 860
 470 IF S<>V THEN GO TO 490
 480 IF Z=1 THEN GO TO 520
 481 LET Q=1
 482 GO TO 500
 490 IF W(R,S+1)<>0 THEN GO TO 5
20
 500 LET X=INT (RND*2+1)
 510 IF X=1 THEN GO TO 790
 511 IF X=2 THEN GO TO 910
 520 GO TO 790
 530 IF S=1 THEN GO TO 670
 540 IF W(R,S-1)<>0 THEN GO TO 6
70
 541 IF R=H THEN GO TO 610
 542 IF W(R+1,S)<>0 THEN GO TO 6
10
 550 IF S<>V THEN GO TO 550
 551 IF Z=1 THEN GO TO 590
 552 LET Q=1
 553 GO TO 570
 560 IF W(R,S+1)<>0 THEN GO TO 5
90
 570 LET X=INT (RND*3+1)
 580 IF X=1 THEN GO TO 820
 581 IF X=2 THEN GO TO 860
 582 IF X=3 THEN GO TO 910
 590 LET X=INT (RND*2+1)
 600 IF X=1 THEN GO TO 820
 601 IF X=2 THEN GO TO 860
 610 IF S<>V THEN GO TO 630
 620 IF Z=1 THEN GO TO 660
 621 LET Q=1
 622 GO TO 640
 630 IF W(R,S+1)<>0 THEN GO TO 6
60
```

```
640 LET X=INT (RND*2+1)
650 IF X=1 THEN GO TO 820
651 IF X=2 THEN GO TO 910
660 GO TO 820
670 IF R=H THEN GO TO 740
680 IF W(R+1,S)<>0 THEN GO TO 7
40
681 IF S<>V THEN GO TO 700
690 IF Z=1 THEN GO TO 730
691 LET Q=1
692 GO TO 830
700 IF W(R,S+1)<>0 THEN GO TO 7
30
710 LET X=INT (RND*2+1)
720 IF X=1 THEN GO TO 860
721 IF X=2 THEN GO TO 910
730 GO TO 860
740 IF S<>V THEN GO TO 760
750 IF Z=1 THEN GO TO 780
751 LET Q=1
752 GO TO 770
760 IF W(R,S+1)<>0 THEN GO TO 7
80
770 GO TO 910
780 GO TO 1000
790 LET W(R-1,S)=C
800 LET C=C+1
801 LET B(R-1,S)=2
802 LET R=R-1
810 IF C=H*V+1 THEN GO TO 1010
811 LET Q=0
812 GO TO 260
820 LET W(R,S-1)=C
830 LET C=C+1
840 LET B(R,S-1)=1
841 LET S=S-1
842 IF C=H*V+1 THEN GO TO 1010
850 LET Q=0
851 GO TO 260
860 LET W(R+1,S)=C
870 LET C=C+1
871 IF B(R,S)=0 THEN GO TO 880
872 LET B(R,S)=3
873 GO TO 890
880 LET B(R,S)=2
890 LET R=R+1
900 IF C=H*V+1 THEN GO TO 1010
902 GO TO 530
910 IF Q=1 THEN GO TO 960
920 LET W(R,S+1)=C
921 LET C=C+1
922 IF B(R,S)=0 THEN GO TO 940
930 LET B(R,S)=3
931 GO TO 950
940 LET B(R,S)=1
950 LET S=S+1
951 IF C=H*V+1 THEN GO TO 1010
952 GO TO 260
960 LET Z=1
970 IF B(R,S)=0 THEN GO TO 980
971 LET B(R,S)=3
972 LET Q=0
```

```
 973 GO TO 1000
 980 LET B(R,S)=1
 981 LET Q=0
 982 LET R=1
 990 LET S=1
 991 GO TO 250
1000 GO TO 210
1010 FOR J=1 TO V
1011 PRINT "+";
1012 FOR I=1 TO H
1013 IF B(I,J)<2 THEN GO TO 1030
1020 PRINT "  ";
1021 GO TO 1040
1030 PRINT "  +";
1040 NEXT I
1041 PRINT
1043 FOR I=1 TO H
1045 IF B(I,J)=0 THEN GO TO 1060
1050 IF B(I,J)=2 THEN GO TO 1060
1051 PRINT "+  ";
1052 GO TO 1070
1060 PRINT "++";
1070 NEXT I
1071 PRINT "+"
1072 NEXT J
1073 PRINT AT 1+2*V,0;
1075 FOR I=1 TO 2*H+1
1076 PRINT "+";
1077 NEXT I
1080 INPUT ""
1100 FOR I=1 TO (H*V)/4
1110 LET X=2+2*INT (RND*V)
1111 LET Y=1+2*INT (RND*H)
1120 PRINT AT X,Y; INK 3;"#"
1140 NEXT I
1145 PRINT AT 2*V,1+2*INT (RND*H
); INK 6;"£"
1200 LET A=0: LET B=1
1210 PRINT AT A,B;"*"
1250 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
1252 IF INKEY$="" THEN GO TO 125
2
1253 IF INKEY$="0" THEN GO TO 12
7
1255 LET I$=K$
1260 PRINT AT A,B;"*"
1262 IF C$<>"+" THEN BEEP .01,0
1265 LET A1=A: LET B1=B
1266 LET K$=I$
1269 IF INKEY$<>"" THEN LET I$=I
NKEY$
1270 IF I$="7" THEN LET A1=A-(A>
0)
1271 IF I$="6" THEN LET A1=A+1
1272 IF I$="5" THEN LET B1=B-1
1273 IF I$="8" THEN LET B1=B+1
1290 LET C$=SCREEN$ (A1,B1)
1295 IF C$=" " THEN GO TO 1350
1300 IF C$="+" THEN GO TO 1255
1310 IF C$="#" THEN LET PT=PT+10
0
```

```
1320 IF C$="£" THEN LET SCH=SCH-
1
1340 IF C$="-" THEN GO TO 1380
1350 PRINT AT A,B;" "
1360 LET A=A1: LET B=B1
1370 GO TO 1260
1390 LET E=E-1
1391 IF E<1 AND NOT SCH THEN GO
TO 1400
1392 PRINT AT A,B;" "
1393 LET A=A1+1
1395 GO TO 1260
1400 LET T=PEEK 23672+256*PEEK 2
3673+65536*PEEK 23674
1405 PRINT AT A1,B1;" "
1406 PRINT AT A,B;" "
1410 LET PT=PT-INT (T/10)
1415 FOR I=1 TO 60
1417 NEXT I
1420 LET X$="> "+STR$ PT+" <"
1460 PRINT AT 2*V+1,0; INVERSE 1
;X$
1470 PRINT AT 0,0;"NOCH  EINMAL?
"
1475 IF INKEY$<>"" THEN GO TO 14
75
1480 IF INKEY$="" THEN GO TO 148
0
1490 IF INKEY$="j" THEN GO TO 12
7
1500 PRINT AT 1,0;"
";"NEUE GROESSE?"
1505 IF INKEY$<>"" THEN GO TO 15
05
1510 IF INKEY$="" THEN GO TO 151
0
1520 IF INKEY$="j" THEN GO TO 15
40
1530 GO TO 9998
1540 CLS
1550 RUN 100
2000 PRINT #0;"Einen Moment Gedu
ld ..."
2010 LET B=48384: FOR I=15616 TO
 16383
2020 POKE (I+B),PEEK I
2030 NEXT I
2040 POKE 23607,249
2050 RESTORE
2060 FOR I=1 TO 4
2070 READ A$
2080 LET ADDR=63744+8*CODE A$
2090 FOR J=0 TO 7
2100 READ A
2110 POKE ADDR+J,A
2120 NEXT J
2130 NEXT I
2140 INPUT "": CLS
2150 PRINT "SPIELANLEITUNG:
                          "
2160 PRINT '"JE NACH GROESSE DAU
ERT ES EINEN MOMENT, BIS DAS LAD
```

```
YRINT ER-     SCHEINT. DANN WERDE
N DIE         SCHAETZE (#) EINGET
RAGEN UND ZUMSCHLUSS DER SCHLUES
SEL (£)."
2170 PRINT "SIE (*) MUESSEN SO V
IELE          SCHAETZE WIE MOEGLIC
H IN MOEG-    LICHST KURZER ZEIT M
ITNEHMEN.     OHNE SCHLUESSEL KOMM
EN SIE NICHTWIEDER ZURUECK."
2180 PRINT ,,"WER ERREICHT DIE H
OECHSTE PUNKT-ZAHL?"
2190 RETURN
3000 DATA "#",000,028,008,030,05
2,126,124,000
3010 DATA "£",000,000,054,150,19
1,167,069,000
3020 DATA "*",024,060,024,060,09
0,153,036,102
3030 DATA "+",255,255,195,195,19
5,195,255,255
9998 PRINT AT 3,0;"
" FLASH 1;" SPIELENDE <"
9999 GO TO 9999
```

Dieses Programm nutzt eine Technik, die im Buch bisher nicht verwandt wurde:

Um auch selbst definierte Zeichen mit SREEN$ zu überprüfen wird in den Zeilen 2010 bis 2020 der SPECTRUM-Zeichensatz aus dem ROM in den oberen RAM-Bereich (ab 64000) kopiert. Zeile 2040 schaltet den SPECTRUM auf den neuen Zeichensatz. Da dieser nun im RAM liegt, können auch die Zeichen mit Codes unter 128 frei verändert werden. Dieses besorgen für einige im Text nicht benutzte Zeichen die Zeilen ab 2050. Im Listing sind diese Zeichen der Übersichtlichkeit halber noch unverändert.

Zum Spiel selbst:
Wählen Sie eine Labyrint-Größe. Bei größeren Labyrinten dauert es eine Weile, bis der Computer sie errechnet hat, dann aber wird es angezeigt. Im Labyrint erscheinen Schätze und ein Schlüssel.

Aufgabe:
Sammeln sie in möglichst kurzer Zeit so viele Schätze wie möglich. Vergessen Sie auf keinen Fall den Schlüssel. Ohne ihn kommen Sie nicht wieder hinaus !

Wenn Sie den Schlüssel haben und zum Ausgang zurück kommen,

werden die Punkte angezeigt. Diese errechnen sich aus der benötigten Zeit und den gefundenen Schätzen.

Aber aufgepaßt:
Auch wenn Sie die entsprechende Taste loslassen, läuft das Männchen weiter! Es ist gar nicht so einfach, die Abzweigungen zu erwischen. In der abgedruckten Version wird das Männchen mit den Kursor-Steuertasten (5 – 8) gesteuert. Wenn Sie die Bedieneinheit aus Kapitel 1 benutzen wollen, ändern Sie bitte die Zeilen 1270 bis 1273 entsprchend um:

```
1270 IF I$="3" THEN LET A1=A-(A>0)
1271 IF I$="7" THEN LET A1=A+1
1272 IF I$="5" THEN LET B1=B-1
1273 IF I$="1" THEN LET B1=B+1
```

4.13 Programm 30: "Music-Computer" (48K)

```
/////      MUSIC-COMPUTER      \\\\\

1 Orgel-Spiel
2 Orgel-Spiel mit Speicherung
3 Speicher-Wiedergabe
4 Programmieren
5 Programmwiedergabe
6 SAVEn des Musik-Speichers
7 LOADen von Musik von Kasette
```




```
   5 DIM d$(3,2025): DIM a$(2025
): POKE 23658,8
  10 BORDER 6: PAPER 7: INK 1: C
LS : GO SUB 7000
  20 PRINT AT 0,0;"/////     MUSI
C-COMPUTER     \\\\\": GO SUB 600
0
  30 PRINT AT 3,0;"1 Orgel-Spiel
"
```

```
  40 PRINT '"■ Orgel-Spiel mit S
peicherung"
  50 PRINT '"■ Speicher-Wiederga
be"
  60 PRINT '"■ Programmieren"
  70 PRINT '"■ Programmwiedergab
e"
  80 PRINT '"■ SAVEn des Musik-S
peichers"''"■ LOADen von Musik v
on Kasette"
  90 PAUSE 0: LET i$=INKEY$
 100 IF i$<"1" OR i$>"7" THEN GO
 TO 90
 110 IF i$="6" THEN PRINT AT 13,
0; FLASH 1;"6": SAVE "m" DATA d$
(): GO TO 20
 120 IF i$="7" THEN PRINT AT 15,
0; FLASH 1;"7": LOAD "m" DATA d$
(): GO TO 10
 130 GO TO 1000*VAL i$
1000 PRINT AT 3,0; FLASH 1;"1"
1010 LET i$=INKEY$: IF i$=CHR$ 1
3 THEN GO TO 30
1020 IF i$>"/" AND i$<"[" THEN L
ET a=a(CODE i$-47): IF a<>99 THE
N BEEP .05,a
1030 GO TO 1010
2000 LET d=1: PRINT AT 5,0; FLAS
H 1;"2": PAUSE 0
2010 LET i$=INKEY$: IF i$=CHR$ 1
3 THEN GO TO 2050
2020 IF i$>"/" AND i$<"[" THEN L
ET a=a(CODE i$-47): IF a<>99 THE
N BEEP .05,a:
2030 LET d$(1,d)=i$: LET d=d+1:
IF d=2025 THEN GO TO 2050
2040 GO TO 2010
2050 LET d$(1,d)=i$: GO TO 30
3000 LET d=1: PRINT AT 7,0; FLAS
H 1;"3"
3010 LET i$=d$(1,d): LET d=d+1:
IF i$=CHR$ 13 THEN GO TO 30
3020 IF i$>"/" AND i$<"I" THEN L
ET a=a(CODE i$-47): IF a<>99 THE
N BEEP .05,a
3030 GO TO 3010
4000 CLS : GO SUB 7000: PRINT AT
 0,0;">>>> PROGRAMMIERUNG <<<<"
4010 PRINT '"Geben Sie je Note W
ert und Dauerein. Den Wert entne
hmen Sie der abgebildeten Tastat
ur.              "ENTER" = Pause.
                 Folgendes sind die
Zeiten:"
4020 PRINT '"0=1/8    1=1/4    2=3
/8    3=1/2    5=3/4    7=1/1
      X=Ende"
4030 LET d=0
4040 LET d=d+1: LET a=d-25: IF a
<1 THEN LET a=1
4050 PRINT AT 13,4;d$(2,a TO a+2
5);AT 14,4;d$(3,a TO a+25)
```

```
4052 LET c=28: IF d<26 THEN LET
c=2+d
4053 PRINT AT 12,c;"    "
4055 IF INKEY$<>"" AND INKEY$<>C
HR$ 8 AND INKEY$<>CHR$ 9 THEN GO
 TO 4055
4060 PRINT AT 13,0;"Wert"
4061 IF INKEY$="" THEN GO TO 406
1
4062 LET i$=INKEY$: IF i$=CHR$ 1
3 THEN LET i$="="
4063 IF i$=CHR$ 9 THEN GO TO 404
0
4064 IF i$=CHR$ 8 THEN LET d=d-1
-(d>1): GO TO 4040
4065 IF i$=CHR$ 12 THEN LET d$(2
)=a$: LET d$(3)=a$: GO TO 4000
4068 LET d$(2,d)=i$: PRINT AT 13
,0;"Wert"
4070 IF INKEY$<>"" THEN GO TO 40
70
4080 PRINT AT 14,0;"Zeit": PAUSE
 0: LET i$=INKEY$
4090 IF (i$<"0" OR i$>"9") AND i
$<>"X" THEN LET i$=INKEY$: GO TO
 4090
4100 LET d$(3,d)=i$: IF d=2000 T
HEN LET d$(3,d+1)="X": GO TO 10
4110 IF d$(3,d)="X" THEN GO TO 1
0
4120 PRINT AT 14,0;"Zeit": GO TO
 4040
5000 PRINT AT 11,0; FLASH 1;"5";
 INPUT "Tempo?";t;"  Wiederholun
gen?";w
5010 FOR i=1 TO w: LET d=1
5020 LET x=a(CODE d$(2,d)-47)
5030 LET z=(VAL d$(3,d)+1)/t: IF
 X<>99 THEN BEEP z,x
5035 IF X=99 THEN PAUSE 50*Z
5040 LET d=d+1: IF d$(3,d)<>"X"
THEN GO TO 5020
5050 NEXT i
5060 GO TO 20
6000 RESTORE : DIM a(43)
6010 FOR i=1 TO 43: READ a: LET
a(i)=a: NEXT i: RETURN
6020 DATA 3,99,-11,-9,99,-6,-4,-
2,99,1,99,99,99,99,99,99,99,99,1
2,9,8,-8,10,99,13,0,15,17,99,16,
14,2,4,-12,-7,6,-5,-1,11,-10,7,-
3,5
7000 FOR i=11 TO 255 STEP 16: PL
OT i,7: DRAW 0,24: NEXT i
7010 PLOT 0,32: DRAW 255,0: PLOT
 0,7: DRAW 255,0
7020 LET o$=CHR$ 21+CHR$ 1+" "+C
HR$ 21+CHR$ 0
7030 PRINT AT 18,0;" ■ ■ ";o$;"
■ ■ ■ ";o$;" ■ ■ ";o$;" ■ ■ ■ ";
o$;" ■ ■ ■ ■ ";o$;" ■ ■ ■ ■ ";o$;"
 ■ ■ ";o$;" ■ ■ ■ ";o$;" ■ ■ ■ ■"
7040 PRINT OVER 1;"Q W E R T Y U
 I O P Z X C V B N "
7050 RETURN
```

Wichtige Variablen:

d$(3,2025) Hier wird die programmierte Musik gespeichert. Bis zu 2025 Töne können mit Tonhöhe und Tondauer gespeichert werden, außerdem direkt gespielte Melodien.

a$(2025) Leerstring, um Teile aus d$ löschen zu können.

Dieses Programm macht Ihren 48K-SPECTRUM zu einer Musikmaschine! Sie haben folgende Möglichkeiten, die Sie vom Menue aus erreichen können:

1 Orgel-Spiel

Die Tastatur wird zu einer Orgeltastatur (siehe Bildschirm). Sie umfaßt drei Oktaven. Übrigens "POKE 23658,8" in Zeile 5 sorgt dafür, daß der SPECTRUM im Großbuchstaben-Modus ist, was unerläßlich für diese Spielweise ist. ENTER beendet das Spiel.

2 Orgel-Spiel mit Speicherung

Ihr Orgelspiel wird so gespeichert, wie Sie es gerade spielen, bis zu 4 Minuten! ENTER beendet Ihr Orgelspiel, die 2 im Menue blinkt nicht mehr.

3 Speicher-Wiedergabe

Das, was Sie gespielt haben, wird exakt so wiedergegeben. Mit allen Pausen und Fehlern. Eine gute Möglichkeit, das eigene Spiel zu kontrollieren. Die 3 blinkt, solange der Computer in diesem Modus ist.

4 Programmieren

Das Non-Plus-Ultra dieses Programmes. Programmieren Sie Musikstücke mit bis zu 2025 Noten bzw. Pausen! Hier wurde auf Bedienungskomfort geachtet.

Bei jeder Note drücken Sie zuerst die Taste für die Tonhöhe ("Wert" wird weiß und "Zeit" wird schwarz) und dann die Zahl für Zeit, die Tondauer. Der SPECTRUM zeigt durch die Invertierung des jeweiligen Wortes an, was für eine Eingabe er erwartet. Der dünne schwarze Balken ist der Kursor, der anzeigt, welche Note Sie gerade bearbeiten. Sie können ihn mit den Kursortasten (Shift nicht vergessen) beliebig hin und her bewegen, um neue Noten zu schreiben

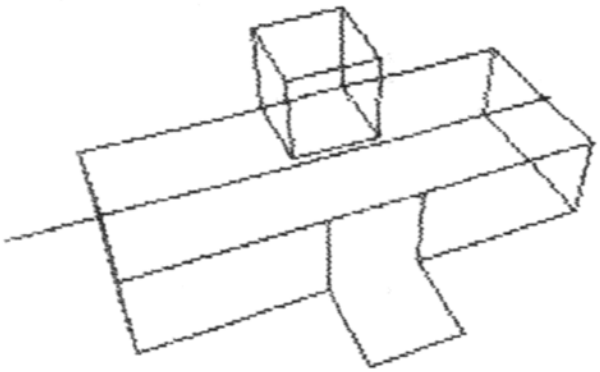
oder Fehler zu korrigieren. Durch "DELETE" (Shift-0) löschen Sie den Notenspeicher. Wollen Sie eine Pause programmieren, geben Sie als Wert bitte einfach ENTER ein. Um das Programmieren zu beenden, geben Sie als Zeit ein "X" ein. Der Computer kehrt zum Menue zurück. Sollten Sie nach dem Abspielen einen Fehler entdecken, so können Sie wieder auf 4 drücken und den Fehler korrigieren.

5 Programmwiedergabe
Bevor der SPECTRUM die programmierte Musik spielt, fragt er nach Tempo und Anzahl der Wiederholungen, bzw. Strophen. Tempo 5 ist für den Anfang ganz gut. Je größer der Wert, desto schneller wird der Computer.

6 SAVEn des Musik-Speichers
Das DIMensionierte Feld d$ wird auf Band gespeichert. Es enthält das unter "2" gespeicherte Musikstück und das unter "4" programmierte Stück. Wenn Ihnen ein besonders gutes Stück gelungen ist, können Sie es festhalten. Wichtig bei Kompositionen.

7 LOADen von Musik von Kassette
Hier können Sie unter "6" gespeicherte Musik wieder in den Computer holen. Sie können die Musik wiedergeben und die programmierte Musik sogar erneut korrigieren, verändern oder erweitern.

4.14 Programm 31: "3-D-Graphik" (48K)



```
  10 BORDER 1: PAPER 1: INK 7: L
ET i=1000: DIM s$(i): DIM z$(10,
i): FOR x=1 TO 10: LET z$(x)="aa
": NEXT x: LET s$="aa"
  20 GO SUB 5000: GO SUB 1000: C
LS : GO SUB 7000
  30 IF p=1 THEN GO TO 200
  40 PAUSE 0: LET r$=INKEY$: LET
 s$(a)=r$: LET a=a+1: IF a=i THE
N PRINT AT 0,0;"Speicher voll!":
 PAUSE 100: GO TO 5000
  50 IF r$="a" OR r$="A" THEN GO
 TO 20
  60 GO SUB 3000: GO SUB 2000: I
F c<0 OR d<0 OR c>255 OR d>172 T
HEN PRINT #0;"BILD ZU GROSS!": P
AUSE 20: INPUT "": GO TO 40
  70 DRAW c-PEEK 23677,d-PEEK 23
676
  80 GO TO 40
 200 GO SUB 6000: PAUSE 0: IF IN
KEY$="z" THEN COPY
 210 GO TO 20
1000 LET s=l*l+m*m: LET t=s+n*n:
 LET q=SQR t: LET h=SQR s: RETUR
N
2000 LET o=t-u*l-v*m-w*n: LET c=
3*t*(v*l-u*m)/(h*o)+128: LET d=3
*q*(w*s-n*(u*l+v*m))/(h*o)+96: R
ETURN
3000 IF r$="7" THEN LET w=w+g
3010 IF r$="6" THEN LET w=w-g
3020 IF r$="8" THEN LET u=u-g
3030 IF r$="5" THEN LET u=u+g
3040 IF r$="9" THEN LET v=v+g
3050 IF r$="0" THEN LET v=v-g
3060 RETURN
4000 INPUT AT 0,0;"Groesse? ";g'
"x-Blickpunkt? ";l'"y-Blickpunkt
? ";m'"z-Blickpunkt? ";n: LET l=
l+(l=0): RETURN
5000 IF INKEY$<>"" THEN GO TO 50
00
5010 CLS : LET a=1
5020 PRINT "Was wuenschen Sie?"'
"
5030 PRINT '''■ Neues Bild malen
'''■ Aktuelles Bild"'"  auf dem
 Bildschirm'''■ Zufalls-Blickpu
nkte'''■ SAVEn des aktuellen"'"
  Bildes auf Kasette'''■ LOADen
 eines Bildes von"'"  Kasette"'
"■ bis ■ Austausch des aktuellen
           Bildes mit dem ent-
           sprechenden Speichers"
5040 PAUSE 0: LET r$=INKEY$
5050 IF r$="n" OR r$="N" THEN LE
T p=0: GO SUB 4000: RETURN
5060 IF r$="g" OR r$="G" THEN LE
T p=1: GO SUB 4000: RETURN
5070 IF r$="z" OR r$="Z" THEN GO
 TO 8000
```

```
5080 IF r$="s" OR r$="S" THEN IN
PUT "Titel? ";t$: GO SUB 5500: S
AVE t$ DATA s$()
5090 IF r$="l" OR r$="L" THEN IN
PUT "Titel? ";t$: GO SUB 5510: L
OAD t$ DATA s$()
5100 IF r$>="0" AND r$<="9" THEN
 LET r=VAL r$+1: LET d$=s$: LET
s$=z$(r): LET z$(r)=d$
5110 PRINT #0;"OK": PAUSE 50: PA
USE 0: INPUT "": GO TO 5040
5500 IF t$="" THEN LET t$="3-D"
5510 IF LEN t$>10 THEN LET t$=t$
( TO 10)
5520 RETURN
6000 GO SUB 7000
6010 LET r$=s$(a): GO SUB 3000:
GO SUB 2000: IF c<0 OR d<0 OR c>
255 OR d>172 THEN PRINT #0;"BILD
 ZU GROSS!": PAUSE 20: INPUT "":
 RETURN
6020 DRAW c-PEEK 23677,d-PEEK 23
678
6030 LET a=a+1: IF s$(a)<>"a" AN
D s$(a)<>"A" AND a<i THEN GO TO
6010
6040 RETURN
7000 LET w=0: LET u=0: LET v=0:
GO SUB 2000: PLOT c,d: RETURN
8000 INPUT "Groesse? ";g
8010 LET a=1: LET l=RND*200-100:
 LET m=RND*200-100: LET n=RND*20
0-100: LET l=l+(l=0): GO SUB 100
0: CLS : GO SUB 6000
8020 IF INKEY$="a" OR INKEY$="A"
 THEN GO TO 5000
8030 PAUSE 50: IF INKEY$="z" THE
N COPY
8040 GO TO 8010
```

Dieses Programm, welches die hochauflösende Graphik des SPECTRUM zur Geltung bringt, geht auf ein Programm von Tim Hartnell in "The ZX Spectrum Explored" (London 1982) zurück, welches hier korrigiert, verändert und wesentlich erweitert wurde.

Mit dem Programm können dreidimensionale Körper gezeichnet, von verschiedenen Seiten betrachtet und auf Band gespeichert werden. Der SPECTRUM ist in der Lage bis zu 10 verschiedene Bilder zu speichern. Die Funtkionen im einzelnen:

N: Neues Bild malen

Zunächst fragt der SPECTRUM nach der Größe. Diese ist ein Maß dafür, wie lang ein Strich mindestens ist. Danach wird die Blick-

richtung, aus der Sie das Objekt betrachten festgelegt. Geben Sie die Koordinaten Ihres Blickpunktes ein. Ein guter Wert für den Anfang ist 70,100,150. Nach diesen Eingaben ist der Bildschirm leer, nur ein Punkt in der Mitte. Mit den Tasten 5 bis 0 können Sie nun zeichnen. Die Pfeile über den Tasten zeigen an, in welche Richtung ein Strich gezeichnet wird, "9" nach vorne, quasi in den Bildschirm hinein und "0" ist zurück, quasi auf Sie zu. Versuchen Sie bei Größe 10 und angegebenen Blickpunkt folgende Eingaben:
7, 8, 6, 5, 9, 7, 8, 6, 5, 7, 0, 8, 9, 6, 0

Auf dem Bildschirm sollte ein Würfel zu sehen sein.
Drücken Sie auf STOP ("A"), um die Eingabe zu beenden.

G: Aktuelles Bild auf den Bildschirm

Sie können das gerade gezeichnete Bild nun aus einem beliebigen Blickwinkel in beliebiger Größe ansehen. Bei zu großer Größe unterbricht der SPECTRUM die Ausführung, um einer Fehlermeldung zuvor zu kommen.

Z: Zufallsblickpunkte

Das aktuelle Bild wird nacheinander unter zufällig bestimmten Blickwinkeln auf den Schirm gezeichnet. Sehr Eindrucksvoll! Auch hier muß zunächst die Größe eingegeben werden. Sollte ein Bild zu groß werden, um auf den Bildschirm zu passen, bricht der SPECTRUM dieses Bild ab und wendet sich dem nächsten Blickwinkel zu. Bei komplizierteren Körpern kann man stundenlang zusehen.

S: SAVEn des aktuellen Bildes auf Kassette

Besonders gelungene Bilder können mit einem Namen versehen und auf Kassette gespeichert werden. Wird kein Name eingegeben, bekommt das Array den Namen "3-D".

L: LOADen eines Bildes von Kassette

Der SPECTRUM fragt zunächst nach dem Namen. Wird keiner eingegeben lädt der SPECTRUM das nächste Bild, ansonsten das mit dem eingegebenen Namen.

0 – 9: Austausch des aktuellen Bildes mit dem des entsprechenden Speichers

Es existieren zehn weitere Bildspeicher, in denen die Bilder aus dem Hauptspeicher eingelesen und herausgelesen werden können, indem der jeweilige Speicherinhalt vertauscht wird. So können Sie also 10 Bilder erzeugen und diese nacheinander auf Kassette SAVEn, oder aber nacheinander 10 Bilder von Kassette LOADen und in die verschiedenen Speicher einlesen . . . auf alle Fälle haben Sie die Möglichkeit, gleichzeitig 10 Bilder im Computer gespeichert zu haben.

4.15 Programm 32: "Quadro"

```
RBH-SOFTWARE PRAESENTIERT

QUADRO

Ziel des Spieles ist es, durch
das Zeichnen von Rechtecken
moeglichst viele Punkte zu er-
reichen.Wenn ein Rechteck fertig
ist, wird es eingefaerbt. Seine
Linien zaehlen dann nicht mehr.
Fuer jedes eingeschlossene "#"
gibt es 25 Sonderpunkte, Fuer
jedes Feld "■" 1 Punkt.
Das Spiel ist aus, wenn Sie ge-
gen ein "#" stossen. Nur die
blinkenden "#" sind gefaehrlich
und zaehlen.
Nach jeweils 2500 Punkten wird
das Feld geleert und die "#"
erscheinen haufiger.

  10 REM QUADRO
         ©Roland G. Huelsmann
          Niddastrasse 104
          6000 Frankfurt
          03/83
  20 BORDER 0: PAPER 0: INK 1: C
LS
  30 DIM r$(4): DIM l(3): LET l=
.0
  40 INK 9: BRIGHT 0: CLS
  50 GO SUB 8000
  60 LET hsc=0
  70 PAPER 0: CLS : LET s=.1: LE
T sc=0
  80 DEF FN a$(x)=("0" AND x<100
00)+("0" AND x<1000)+("0" AND x<
100)+("0" AND x<10)+STR$ x
```

```
  85 LET z$=CHR$ 16+CHR$ 9+CHR$
17+CHR$ 8+CHR$ 19+CHR$ 1+CHR$ 18
+CHR$ 1+"#"
  90 PRINT #0;AT 1,0;"SCORE: 000
00","HI-SCORE: ";FN a$(hsc): LET
 a=128: LET b=0: PLOT a,b
 100 INK 9: LET x=0: LET y=0: LE
T i$=INKEY$: IF i$<>"" AND i$<>r
$(1) THEN LET r$(4)=r$(3): LET r
$(3)=r$(2): LET r$(2)=r$(1): LET
 r$(1)=i$: LET l(3)=l(2): LET l(
2)=l(1): LET l(1)=l: LET l=0
 110 IF i$="5" THEN IF a>7 THEN
LET x=-8
 120 IF i$="8" THEN IF a<248 THE
N LET x=8
 130 IF i$="6" THEN IF b>7 THEN
LET y=-8
 140 IF i$="7" THEN IF b<168 THE
N LET y=8
 145 IF i$=r$(1) AND x OR y THEN
 LET l=l+1
 150 LET a=a+x: LET b=b+y: IF PO
INT (a,b) THEN IF l THEN GO SUB
1000
 155 DRAW x,y
 160 OVER 1: PLOT a,b: PLOT a+1,
b: PLOT a-1,b: PLOT a,b+1: PLOT
a,b-1: PLOT a+1,b+1: PLOT a+1,b-
1: PLOT a-1,b+1: PLOT a-1,b-1
 170 PLOT a+1,b: PLOT a-1,b: PLO
T a,b+1: PLOT a,b-1: PLOT a+1,b+
1: PLOT a+1,b-1: PLOT a-1,b+1: P
LOT a-1,b-1: OVER 0: PLOT a,b:
 180 IF RND<s THEN PRINT AT RND*
21,RND*31;z$
 185 IF ATTR (21-(b+y)/8,(a+x)/8
)>128 THEN GO TO 500
 190 GO TO 100
 500 FOR i=1 TO 10: BEEP .01,30:
 NEXT i
 510 IF hsc<sc THEN LET hsc=sc
 515 IF INKEY$<>"" THEN GO TO 51
5
 520 PRINT #0;AT 0,0; INVERSE 1;
"### SPIELENDE ###  Noch einmal?
": PAUSE 0: IF INKEY$<>"n" THEN
INPUT "": GO TO 70
 530 STOP
1000 IF l(2)<>l OR l(1)>l(3) THE
N RETURN
1010 RESTORE : FOR i=1 TO 8: REA
D x$
1020 IF r$=x$ THEN GO TO 2000
1030 NEXT i: LET l=0: RETURN
2000 OVER 1: BRIGHT INT (RND*2):
 PAPER INT (RND*7)+1
2010 IF i=1 THEN LET xp=a/8: LET
 yp=22-b/8-l(1): LET x1=xp+l-1:
LET y1=yp+l(1)-1
```

```
2020 IF i=2 THEN LET xp=a/8-l: L
ET yp=22-b/8-l(1): LET xl=xp+l-1
: LET yl=yp+l(1)-1
2030 IF i=3 THEN LET xp=a/8-l(1)
: LET yp=22-b/8-l: LET xl=xp+l(1
)-1: LET yl=yp+l-1
2040 IF i=4 THEN LET xp=a/8-l(1)
: LET yp=22-b/8: LET xl=xp+l(1)-
1: LET yl=yp+l-1
2050 IF i=5 THEN LET xp=a/8-l: L
ET yp=22-b/8: LET xl=xp+l-1: LET
 yl=yp+l(1)-1
2060 IF i=6 THEN LET xp=a/8: LET
 yp=22-b/8: LET xl=xp+l-1: LET y
l=yp+l(1)-1
2070 IF i=7 THEN LET xp=a/8: LET
 yp=22-b/8: LET xl=xp+l(1)-1: LE
T yl=yp+l-1
2080 IF i=8 THEN LET xp=a/8: LET
 yp=22-b/8-l: LET xl=xp+l(1)-1:
LET yl=yp+l-1
2090 LET sc=sc+l*l(1): PRINT #0;
AT 1,7;FN a$(sc)
2100 FOR i=xp TO xl: FOR j=yp TO
 yl: LET sc=sc+25*(ATTR (j,i)>12
8): PRINT AT j,i;" ": NEXT j: NE
XT i: OVER 0: LET l=0: DIM l(3)
2105 PRINT #0;AT 1,7;FN a$(sc)
2110 IF sc>25000*s THEN GO TO 25
00
2120 RETURN
2500 FOR i=1 TO 10: BEEP .2,i: N
EXT i
2510 PAPER 0: CLS : PLOT a-x,b-y
: LET s=s+.1
2520 IF sc>hsc THEN LET hsc=sc
2530 PRINT #0;AT 1,0;"SCORE :";F
N a$(sc),"HI-SCORE: ";FN a$(hsc)
: RETURN
8000 PRINT INK 6;"RGH-SOFTWARE P
RASENTIERT:": INK 3;"▀▄ ▀▄ ▀▄ ▀▄
▀▄ ▀▄ ▀▄ ▀▄"
8010 PRINT INK 2;"▌▄▐▌ ▐ ▀▄▐ ▐
▐ ▀▄▀ ▐ ▐"
8020 PRINT INK 6;"▀▄▀ ▀▄▐▌ ▐ ▐▄
▐ ▐ ▀▄ ▀▄"
8030 PRINT INK 7;"Ziel des Spiel
es ist es, durch das Zeichnen v
on Rechtecken moeglichst vie
le Punkte zu er- reichen.Wenn e
in Rechteck fertigist, wird es e
ingefaerbt. Seine Linien zaehlen
 dann nicht mehr. Fuer jedes ein
geschlossene "#" gibt es 25 Son
derpunkte, Fuer jedes Feld "▀"
 1 Punkt."
8040 PRINT "Das Spiel ist aus, w
enn Sie ge- gen ein "#" stossen.
 Nur die blinkenden "#" sind
gefaehrlich und zaehlen."
```

```
8050 PRINT "Nach jeweils 2500 Pu
nkten wird  das Feld geleert und
 die "#"     erscheinen haufiger.
"
8060 PRINT #0;AT 0,13; FLASH 1;"
Viel Spass!";AT 1,0; FLASH 0;"Dr
uecken Sie irgendeine Taste!": P
AUSE 0: INPUT "": RETURN
9000 DATA "5667","8657","6875","
7865","8756","5786","7568","6578
"
9120 PLOT a,b: RETURN
```

Da sich das Spiel selbst erklärt, hier nur ein paar kurze Zusatzinformationen:

Bei der Punktewertung zählt die Fläche eines gezeichneten Rechteckes, wobei die Fläche eines Zeichens den Wert 1 hat. Jedes blinkende "#" zählt 25 Sonderpunkte. Außerdem ist zu beachten, daß zum Rechteck nur die Linien zählen, die nach (!) dem Beenden des letzten Rechteckes gezeichnet wurden. Dabei kann eine neue Linie auch über eine alte gezeichnet werden, aber niemals zählt eine nur einmal gezogene Linie für zwei Rechtecke.

Ein Hinweis noch:
An den blinkenden Zeichen kann man rechts und oben extrem dicht vorbeifahren, während man links und unten etwas Abstand halten muß. Probieren Sie es aus. Sie werden bald herausbekommen, was Sie dürfen und was Sie nicht dürfen.

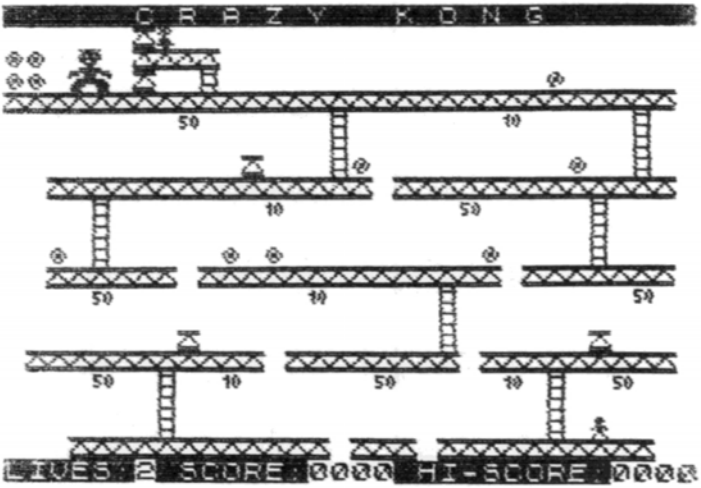
Da die Farbe, mit der ein geschaffenes Rechteck gefüllt wird vom Zufall bestimmt wird, kann es sein, daß die gleiche Farbe gewählt wird, die schon vorhanden ist. Dann erkennt man das Füllen des Rechteckes nur daran, daß plötzlich nichts mehr blinkt und man Punkte bekommt. Um hohe Punktezahlen (z. B. über 5000!) zu erreichen, ist es wichtig, zu Beginn recht große Quadrate zu zeichnen. Die Zeiten, da man sich mit weniger zufrieden geben muß, kommen früh genug !

4.16 Programm 33: "Crazy Kong" (48K)

```
          C R A Z Y   K O N G

Ziel des Spieles ist es, mit
den drei Maennchen, die zur
Verfuegung stehen, soviele
Punkte wie moeglich zu errei-
chen.
Punkte werden erzielt beim
Springen ueber Hindernisse und
Abgruende und beim Springen nach
den Zahlen. Einen Superbonus
gibt es beim Erreichen des
Maedchens auf der obersten
Plattform.

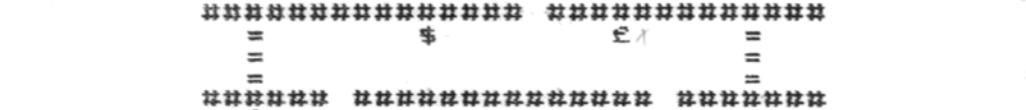
Tastenbelegung:
[I] Links
[O] Runter
[P] Hoch
[Q] Rechts
[A] Sprung nach links
[S] Sprung nach rechts
```



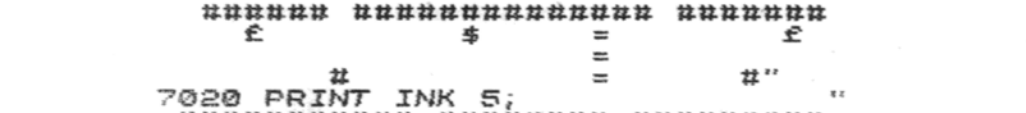

```
  10 REM CRAZY KONG
         ©Roland G. Huelsmann
           03/83
  20 CLEAR 63999: RESTORE : RAND
OMIZE : BORDER 1: PAPER 1: INK 7
: CLS
  30 GO SUB 8000: REM Anleitung
  40 GO SUB 9000: REM udg
  45 INPUT "": PRINT #0;"Druecke
n Sie irgendeine Taste.": PAUSE
0: INPUT ""
```

```
  50 PRINT AT 21,0; INK 6;"LIVES
    SCORE:      HI-SCORE:     ": PO
KE 23607,249
  60 LET hsc=0
  70 LET sc=0: LET s=1: LET l=3
  80 DEF FN a$(a)=("0" AND a<100
0)+("0" AND a<100)+("0" AND a<10
)+STR$ a
  90 LET s=s+1
 100 OVER 0: GO SUB 7000: PAPER
6: INK 1: PRINT AT 21,6;l;AT 21,
14;FN a$(sc);AT 21,28;FN a$(hsc)
: PAPER 1: INK 7
 105 PRINT INK 7; BRIGHT 1;AT 1,
6;":!": FOR i=1 TO 20: BEEP .1,5
: NEXT i: PRINT AT 1,8;"  "
 110 LET x=19: LET y=27: LET x1=
x: LET y1=y: OVER 1: PRINT AT x,
y;"x"
 115 INK 8: PRINT AT 3,0;a$( TO
31);AT 7,2;a$(32 TO 59);AT 11,2;
a$(60 TO 87);AT 15,1;a$(88 TO 11
6);AT 19,3;a$(117 TO )
 120 BEEP .01,-5: INK 4: PRINT A
T x1,y1;"x";AT x,y;"x": LET x1=x
: LET y1=y
 130 LET q$=SCREEN$ (x,y)
 132 INK 8: PRINT AT 3,0;a$( TO
31);AT 7,2;a$(32 TO 59);AT 11,2;
a$(60 TO 87);AT 15,1;a$(88 TO 11
6);AT 19,3;a$(117 TO ): LET a$=a
$(132)+a$: IF SCREEN$ (x,y)<>q$
THEN GO TO 5100
 134 PRINT AT 3,0;a$( TO 31);AT
7,2;a$(32 TO 59);AT 11,2;a$(60 T
O 87);AT 15,1;a$(88 TO 116);AT 1
9,3;a$(117 TO )
 135 IF INKEY$="5" THEN LET y=y-
1
 140 IF INKEY$="8" THEN LET y=y+
1
 150 IF INKEY$="6" THEN IF SCREE
N$ (x+2,y)="=" OR SCREEN$ (x+1,y
)="=" THEN LET x=x+2
 155 IF INKEY$<>"7" THEN GO TO 1
70
 160 IF SCREEN$ (x-1,y)<>" " THE
N LET x=x-2
 165 IF x=1 THEN GO TO 4000: REM
 Ziel
 170 IF INKEY$="0" OR INKEY$="9"
 THEN GO SUB 500
 180 IF x=x1 AND y=y1 THEN GO TO
 130
 185 IF SCREEN$ (x,y)="#" THEN G
O TO 5100
 190 IF SCREEN$ (x,y)<>"o" AND S
CREEN$ (x,y)<>"#" AND SCREEN$ (x
+1,y)<>" " THEN GO TO 120
 200 IF SCREEN$ (x+1,y)<>"#" AND
 SCREEN$ (x+1,y)<>"=" THEN GO TO
```

```
 5000: REM Fall+Tod
 210 GO TO 5100: REM Tod
 500 LET i$=INKEY$: PRINT AT x1,
y1;"x"
 510 LET y=y-(i$="9")+(i$="0")
 520 PRINT AT x-1,y; INK 4;"x":
BEEP .05,30
 530 LET q$=SCREEN$ (x-2,y)
 540 IF q$="£" THEN LET sc=sc+50
: PRINT OVER 0;AT x-2,y;":"
 550 IF q$="$" THEN LET sc=sc+10
: PRINT OVER 0;AT x-2,y;" "
 560 IF SCREEN$ (x,y)="@" THEN L
ET sc=sc+5
 570 IF SCREEN$ (x+1,y)=" " THEN
 LET sc=sc+15
 580 IF SCREEN$ (x,y)="#" THEN L
ET sc=sc+25
 590 LET x1=x-1: LET y1=y
 600 LET y=y-(i$="9")+(i$="0")
 610 RETURN
4000 BORDER 6: POKE 23607,60: PR
INT AT 1,12; INK 6; BRIGHT 1; FL
ASH 1;"Grallierei": POKE 23607,
249
4010 LET sc=sc+250: FOR i=0 TO 1
0: BEEP i/10,i: NEXT i
4020 BORDER 1: IF hsc<sc THEN LE
T hsc=sc
4030 GO TO 90
5000 OVER 0: PRINT AT x1,y1;" ":
 FOR i=x TO 20: PRINT AT i,y;"x"
;AT i-1,y;" ": BEEP .1,10: NEXT
i
5100 FOR i=0 TO -10 STEP -1: BEE
P .1,i: NEXT i: OVER 0: IF hsc<s
c THEN LET hsc=sc
5110 LET l=l-1: IF l=0 THEN GO T
O 6000: REM Ende
5120 GO TO 100
6000 LET s=0: PAPER 6: INK 1: PR
INT AT 21,6;l;AT 21,14;FN a$(sc)
;AT 21,28;FN a$(hsc): PAPER 1: I
NK 7
6010 FOR i=0 TO -20 STEP -1: BEE
P ABS i/10,i: NEXT i: IF hsc<sc
THEN LET hsc=sc
6020 POKE 23607,60: PRINT #0;"No
ch ein Spiel? (j/n) ": POKE 2360
7,249: PAUSE 0: IF INKEY$="j" TH
EN INPUT "": GO TO 70
6030 OVER 0: STOP
7000 PRINT AT 0,0; INK 6;"
@ R A Z Y   K O N G "
7010 PRINT INK 5;
        #y
oo ab ####
oo cd #   =
##############################
         £       =       $     =
                 =             =
             #   =             =
```



```
7020 PRINT INK 5;
```



```
7040 DIM a$(132): FOR i=1 TO 132
 STEP 2: IF RND<s*.05 THEN LET a
$(i)="G"
7050 NEXT i: RETURN
8000 PRINT AT 0,0; INK 5;"
C A Z Y   K O N G
8010 POKE 23607,60: PRINT '"Ziel
 des Spieles ist es, mit
drei Maennchen, die zur
uegung stehen, soviele
te wie moeglich zu errei-
."
8020 PRINT "Punkte werden erziel
t beim
nisse und
pringen nachden Zahlen. Einen Su
perbonus
en des
ersten
8030 PRINT '"Tastenbelegung:"'"
 Links"'"
echts"'"
Sprung nach rechts"
8040 PRINT #0;"Bitte einen Momen
t Geduld ..."
8050 RETURN
9000 LET b=46384: FOR i=15616 TO
 16383: POKE (i+b),PEEK i: NEXT
i:
9010 FOR i=1 TO 13: READ a$: LET
 adr=63744+8*CODE a$
9020 FOR j=0 TO 7: READ a: POKE
adr+j,a: NEXT j: NEXT i
9030 RETURN
9040 DATA "a",BIN 00001011,BIN 0
0000111,BIN 00001101,BIN 0000111
1,BIN 00000110,BIN 00000011,BIN
00000001,BIN 00001111
9050 DATA "b",BIN 11010000,BIN 1
1100000,BIN 10110000,BIN 1111000
0,BIN 01100000,BIN 11000000,BIN
10000000,BIN 11110000
9060 DATA "c",BIN 01111111,BIN 1
1110011,BIN 10000011,BIN 0001111
1,BIN 00111111,BIN 01111100,BIN
01111000,BIN 01111000
```

```
9070 DATA "d",BIN 11111110,BIN 1
1001111,BIN 11000001,BIN 1111100
0,BIN 11111100,BIN 00111110,BIN
00011110,BIN 00011110
9080 DATA "x",24,60,24,60,90,24,
36,66
9090 DATA "o",0,24,36,90,90,36,2
4,0
9100 DATA "$",0,36,106,42,42,36,
0,0
9110 DATA "£",0,114,89,101,21,98
,0,0
9120 DATA "#",255,255,24,36,66,1
29,255,255
9130 DATA "y",88,52,88,16,56,60,
126,24
9140 DATA "=",66,126,66,66,66,12
6,66,66
9150 DATA ":",0,173,169,237,169,
41,13,0
9160 DATA "!",0,58,42,58,34,34,1
50,34
```

Zu diesem Spiel braucht wohl nichts gesagt werden. Nur ein Hinweis vielleicht:

Wenn das Männchen vor einer Leiter oder auf einer Leiter steht, ist es vor den rollenden Fässern geschützt. Beim Herabsteigen der Leitern ist darauf zu achten, daß man rechtzeitig mit dem Herabklettern aufhört, sonst fällt man sehr tief . . .

Übrigens:
Wenn man die obere Plattform erreicht hat, geht es mit mehr Fässern von vorne los.

Zur Technik:
Auch bei diesem Spiel ist, wie bei Labyrint, der SPECTRUM-Zeichensatz in den RAM verlegt worden ab Adresse 64000 (ab Zeile 9000), um Zeichen mit einem Code unter 128 umzudefinieren.
Mit POKE 23607,60 wird zwischenzeitlich zum Originalzeichensatz zurückgeschaltet. Beim SPECTRUM ist es möglich, daß Zeichen aus verschiedenen Zeichensätzen gleichzeitig auf dem Bildschirm sind.

Empfehlenswerte Literatur

Rund um den SPECTRUM, E. Flögel
Hofacker-Verlag, Nr. 108 29,80 DM

Programmieren in Maschinensprache (Z80) – Band I, C. Lorenz
Hofacker-Verlag, Nr. 119 39,00 DM

Programmieren in Maschinensprache mit Z80 – Band II, Dr. Schmitter
Hofacker-Verlag, Nr. 24 29,80 DM

Z80-Programmier-Handbuch
Hofacker-Verlag, Nr. 8029 29,80 DM
Jeder Z80-Befehl wird ausführlich beschrieben und erläutert! Unentbehrlich für den Maschinenspracheprogrammierer.

57 Programme in BASIC, Lorzenz
Hofacker-Verlag, Nr. 31 39,00 DM
Praktische Programme, aber auch Spielprogramme in BASIC. Meist einfach für den SPECTRUM umzuschreiben.

NOTIZEN

NOTIZEN

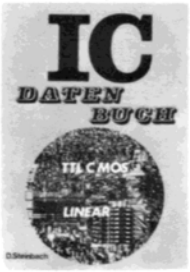
Hofacker-Bücher Deutsch



TBB-Transistor-, Berechnungs- und Bauleitungs-HB, Band 1, Hofacker
Völlig neu überarbeitete Auflage. Das Buch soll bei der täglichen Arbeit im Labor, in der Werkstatt oder am Elektronik-Hobbytisch Ihnen ein guter Begleiter sein. Berechnungsgrundlagen, Berechnungsbeispiele, Tabellen, Vergleichslisten, Digitaltechnik, Netzgeräte, BASIC-Programme zur Berechnung spezifischer Schaltungen usw. sind in übersichtlicher Form dargestellt. Ca. 300 Seiten.
Best.-Nr. 1 **29,80 DM**



Electronic im Auto, H. Gebauer
Mit Handbuch für Polizeiradar. Ein Buch für jeden technisch interessierten Autofahrer. Es zeigt Ihnen die vielen Möglichkeiten zur Verbesserung von Sicherheit, Leistung und Fahrkomfort in Ihrem Auto. Thyristorzündung, Beschleunigungsmesser, Drehzahlmesser, Batterieladegerät, Alarmanlagen u. v. a.
Best.-Nr. 3 **9,80 DM**



IC-Datenbuch, D. Steinbach
Daten- und Auswahllisten der gebräuchlichsten integrierten Schaltkreise. Digital und analog. Gerade bei ICs ist es wichtig die Anschlußfolgen genau zu kennen. Die wichtigsten TTL-Schaltkreise, NF-Verstärker, C-Mos Serie, lineare Schaltungen wie Operationsverstärker, Komparatoren, Spannungsregler, Trigger-Schaltungen, u. v. a. Das IC-Datenbuch wird auch Ihnen ein unentbehrlicher Begleiter bei allen Arbeiten mit integrierten Schaltungen sein.
Best.-Nr. 5 **9,80 DM**



TBB-Handbuch, Band 2, Hofacker
Dieses Buch ist die Fortsetzung des erfolgreichen Handbuches, Band 1. Ein Buch, das sich in der Hand des Praktikers bestens bewährt hat. Weitere neueste Schaltbeispiele und Berechnungsgrundl., Experimentier- und Versuchsbeschreibungen. Integrierte Spannungsregler, Wärmeableitung, Operationsverstärker Einführung, RC-Zeitglieder, Transistortester u. v. a.
Best.-Nr. 2 **19,80 DM**



IC-Handbuch, C. Lorenz
Ein Handbuch für digitale und lineare integrierte Schaltungen. Daten- und Auswahl-, Vergleichslisten, Gehäuseformen, Grundlagen, viele Schaltbeispiele, Printvorlagen, u. v. a. Alles über TTL-Technik, C MOS, MOS-Schaltungen, integrierte NF-Verstärker u. v. a.
Best.-Nr. 4 **19,80 DM**



IC-Schaltungen, D. Steinbach
Hier finden Sie eine gelungene Zusammenstellung der wichtigsten Anwendungsbeispiele aus dem Bereich der integrierten Schaltungen. TTL, C MOS, Linear. Alle Schaltungen sind übersichtlich und klar dargestellt und mit einer kurzen, jedoch sehr genauen Beschreibung versehen. Tastenenprellung, Zähler, Impulsgeber, Codierer, Dekodierer, Datenübertragung, Serien-Parallel-Wandler, Digitalvoltmeter u. v. a.
Best.-Nr. 6 **19,80 DM**



Elektronik Schaltungen, Hofacker
Die ideale Schaltungssammlung zum Basteln u. Experimentieren, Schaltungen mit Operationsverstärkern, Spannungsreglern, TTL, C-MOS-Schaltkreisen. MOS Uhr mit Weckerelektronischer Würfel u. v. a.
Best.-Nr. 7 **9,80 DM**



IC Bauanleitungs Handbuch -IC -KIT, C. Lorenz
Ein Bauanleitungsbuch mit vielen hochinteressanten Bauanleitungen aus dem Bereich der LSI Schaltungstechnik. Schaltbeispiele mit Printvorlagen zum Selbstherstellen der Leiterplatten mit genauesten Beschreibungen. Hochaktuell und brandneu: Funktionsgenerator XR 2206, MOS-Uhr mit Wecker, programmierbarem Wecktongenerator, Schlummerautomatik, IC-Netzteil, Experimentieranleitung und Grundkurs über Flip Flops, u. v. a. Zu allen Schaltungen finden Sie Platinenvorlagen oder Sie können die Experimentierschaltungen auf der Experimentierplatine WH-1 g durchführen. Über 100 Seiten.
Best.-Nr. 8 **19,80 DM**



Feldeffekttransistoren, C. Lorenz
Der Feldeffekttransistor (FET) gehört heute zu den interessantesten Bauteilen überhaupt. Wie man damit experimentiert, wie man seine Funktion versteht und wie man damit brauchbare u. hochinteressante Schaltungen aufbauen kann, zeigt Ihnen dieses Buch. Grundlagen, Kennlinienfelder, Tabellen, Rechenbeispiele, Anschlußbilder und eine Vergleichsliste für Feldeffektransistoren bilden den Kern dieser umfangreichen Darstellung. Alles in allem finden Sie hier eine praxisnahe und komplette Arbietsunterlage, mit der Sie im Beruf und auch im Hobby erfolgreich arbeiten können.
Best.-Nr. 9 **9,80 DM**



Elektronik und Radio, C. Lorenz
4. Auflage. Völlig neu bearbeitet und stark erweitert. Eine Schritt für Schritt Einführung in die Radiotechnik mit vielen Bildern. Vom einfachen Diodenempfänger (Detektor) bis zu interessanten Sender- und Empfängerschaltungen (Minispione). IC-Radio, IC-Sender, Antennen, Berechnungsgrundlagen, Tabellen u. v. a. Über 150 Seiten.
Best.-Nr. 10 **19,80 DM**



NF-Verstärker, C. Lorenz
Grundlagen der integrierten NF-Verstärker, Berechnung von kompletten IC-NF-Verstärkerstufen. Anwendungsbeispiele mit den interessantesten und gebräuchlichsten Standard IC-NF-Verstärkern wie TBA 800, TBA 830, usw. Printvorlagen, Auswahltabellen, Experimentieranleitungen und Anschlußbilder machen dieses Buch zu einem unentbehrlichen Begleiter für alle, die sich m. NF-Verstärkern beschäftigen wollen.
Best.-Nr. 11 **9,80 DM**



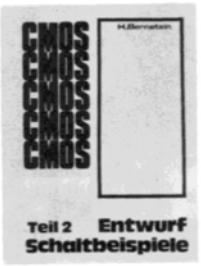
BIS, Beispiele integrierter Schaltungen, H. Bernstein
Auf über 130 Seiten Anwendungsbeispiele mit integrierten Schaltkreisen, Zeitgeber 555, Funktionsgenerator ICL 8038, Opto Elektronik, Operationsverstärker, Festwertspeicher (ROM), u. v. a.
Best.-Nr. 12 **19,80 DM**



HEH, Hobby Elektronik Handbuch **C. Lorenz**
Das Schaltungsbuch f. jeden Hobbyelektroniker, Schaltbeispiele und Bauanleitungen aus dem gesamten Hobbybereich. Lichtorgeln, Alarmanlagen, Eiswarngerät fürs Auto, PLL-Schaltungen u. v. a.
Best.-Nr. 13 **9,80 DM**



Opto-Handbuch, **C. Lorenz**
Das Handbuch für die gesamte Optoelektronik. Eine Einführung und ein ideales Nachschlagwerk. Grundlagen, Definitionen aller Kenngrößen, Opto-Lexikon, Berechnungsgrundlagen, Lichtsender, Lichtempfänger, Anzeigen, Infrarot Detektoren, Optokoppler, Opto-Vergleichsliste, u. v. a.106 Seiten.
Best.-Nr. 15 **19,80 DM**



C MOS Entwurf u. Schaltbeispiele, Teil 2, **H. Bernstein**
Fortsetzung von Best-Nr. 16. Anwendungsbeispiele mit genauen Schaltungsbeschreibungen und Bauelementunterlagen. Daten, Anschlußbelegungen weiterer wichtiger hochintegrierter C MOS Elemente. Ein komplettes Arbeits- u. Experimentierbuch. C-MOS Uhrenschaltungen, Schieberegisterschaltungen, Parallel-Serien Umsetzung, statische u. dynamische Speicherschaltungen, Zählschaltungen, Digital Analog-Wandler, Analog Digital Wandler. Digital Voltmeter, I/O Registerschaltungen. RAM und ROM Anwendungen. Über 140 Seiten.
Best.-Nr. 17 **19,80 DM**



Viel mehr als 33 Programme für den Sinclair SPECTRUM, R. G. Hülsmann
Ein echt aufregendes Buch für jeden SPECTRUM-Besitzer. Viele Tricks und Tips. 33 Superprogramme wie 3D-Graphik, Crazy Kong, Musik-Computer. Unterprogramme in Maschinensprache, zehn kurze Progr. für den Spectrum mit 16K RAM wie Mondlandung, Spielautomat, Irrgarten, Todeshöhle usw. Sie werden begeistert sein!

Best.-Nr. 144 **29,80 DM**



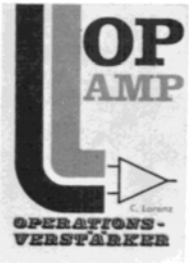
C MOS Einführung, Entwurf, Schaltbeispiele, Teil 1 **H. Bernstein**
Vom C MOS Gatterbaustein über Schieberegister und Zähler bis hin zum C MOS Schreib- Lesespeicher. Insgesamt werden neunzehn interessante und bekannte C MOS Schaltkreise beschrieben. Zu jedem Bauelement sind genaue Daten, Schaltbild und Anwendungsbeispiele angegeben. Im großen Applikationsteil finden Sie: C MOS-Kippstufen, Addierwerke u. Rechenschaltungen, Digital Analog Wandler, Schieberegister für analoge Spannungen, Multiplexsysteme f. analoge Signale u. v. a. Eine komplette Einführung u. gut geeignet für das Selbststudium der C MOS Technik. 140 Seiten.
Best.-Nr. 16 **19,80 DM**



C MOS Entwurf u. Schaltbeispiele, Teil 3, **H. Bernstein**
Fortstetzung von Best.-Nr. 17. Eine sehr umfangreiche Applikationssammlung mit hochintegrierten C MOS Elementen. Speicher- und Steuerschaltungen, Multiplex- und Datenbussysteme, Liquid Cristal Anzeigen, Uhrenschaltungen, PLL-Schaltungen, Optoelektronik in Verbindung mit C MOS. Aufbau und Wirkungsweise der Prozeßrechentechnik, Arithmetische Logische Einheiten (ALU) u. andere wichtige Funktionen aus der Prozeßrechentechnik. RAMs, ROMs, und FIFO-Speicherschaltungen.
Best.-Nr. 18 **19,80 DM**



IC Experimentier Handbuch -IC, -EX, C. Lorenz
Eine sehr umfangreiche Schaltungs- und Bauanleitungssammlung mit neuen, jedoch meist beim Fachhandel erhältliche Standard ICs. Rechnerschaltungen, Mikroprozessoren, I/O-Schaltungen, Stoppuhren, druckende und anzeigende Rechner, Digitalvoltmeter, Hilfschaltungen für den Elektronik Experimentierer, A/D-Wandler, Frequenzzähler u. v. a. Viele Schaltungen können auf der IC KIT Experimentierplatine WH-1g aufgebaut werden.
Best.-Nr. 19 19,80 DM



Operationsverstärker, C. Lorenz
Dieses Buch umfaßt das gesamte Gebiet der linearen Schaltungstechnik und stellt ein in dieser Preislage bisher noch nie dagewesenes Nachschlagwerk und Einführungshandbuch dar. Bestens geeignet für das Selbststudium. Nach einer pädagogisch geschickt gemachten Einführung folgen theoretische Arbeitsunterlagen und die zugehörigen Schaltbeispiele mit Daten und Gehäuseanschlüssen. Dieses wertvolle Buch dürfte seinen Platz auch bei Ihren Arbeitsunterlagen finden, und wird dann immer von Nutzen sein, wenn es um die Lösung von nicht routinemäßigen Aufgaben geht. Über 150 Seiten.
Best.-Nr. 20 19,80 DM



Digitaltechnik Grundkurs, C. Lorenz
Ein Einführungskurs in die Digitaltechnik für Anfänger und Fortgeschrittene. Ein Fachbuch für den programmierten Selbstunterricht. Der ideale Kurzlehrgang für das Selbststudium. Der Kurs vermittelt Ihnen alle wichtigen Grundkenntnisse vom TTL-Gatter bis zum Mikroprozessor und Lösung von Schaltungsaufgaben durch Software. Viele Versuchsaufbauten u. Experimente aus diesem Kurs können auf der IC-KIT Platine WH–1g durchgeführt werden. Grundlagen, Gatter, Zähler, programmierbare Zähler, IC-Tester, Schieberegister, Speicher, Mikroprozessoren u. v. a.
Best.-Nr. 21 19,80 DM



Mikroprozessoren, Eigenschaften u. Aufbau, Teil 1, H. Bernstein
Grundlagen, Eigenschaften u. Aufbau von Mikroprozessoren. Organisation von Recheneinheiten und Mikroprogr. Programmierung und Klassifizierung v. Mikroprozessoren. Ablaufdiagramm, Flußdiagramm. Ein Cip-Technik und Multi Chip-Technik, Transfer- und Sprungfunktionen. Speichertechnik: RAMs ROMs, FIFO, FILO. Programmierbare logische Arrays (PLA). Anwendungsbeispiele u. Anwendungsbereiche. Über 120 Seiten.
Best.-Nr. 22 19,80 DM



Elektronik Grundkurs, C. Lorenz
Eine leichtverständliche und pädagogisch geschickt gemachte Einführung in die Technik der elektronischen Schaltungen. Ein Kurzlehrgang und Schnellkurs zugleich. Aber auch ein recht brauchbares Nachschlagwerk für den fortgeschrittenen Elektroniker. Mit wenig Mühe können Sie sich hier die Grundkenntnisse der elektronischen Schaltungspraxis aneignen. Das Buch schafft die Voraussetzungen für ein erfolgreiches und sicheres Arbeiten mit interessanten Schaltkreisen modernster Technologien. Unentbehrlich f. das Experimentieren mit den heutigen modernen hochintegrierten Schaltkreisen.
Best.-Nr. 23 9,80 DM

ohne Abbildung

Programmieren in Maschinensprache mit Z-80 – Band II, Dr. Schmitter
Dieses sehr interessante Werk geht insbesonders auf TRS-80 und Video Genie ein. Es kann jedoch grundsätzlich auf für alle anderen Personalcomputer, die auf der Z80 CPU basieren, verwendet werden (Color Genie, ZX-81, Spectrum, Osborne, Sharp, usw.). Aus dem Inhalt: µC, µP, RAM, ROM - eine Einführung, Unsere ersten Programmierschritte, Schleifen, Flaps und Sprünge, PUSH, POP, Stapel und Unterprogramme, Input-/Output-Programmierung und Cassettenport. Arithmetik, relative Sprünge und logische Verknüpfungen. BASIC und Maschinencode, wichtige Hilfsmittel wie Disassembler und Assembler.
Best.-Nr. 24 29,80 DM

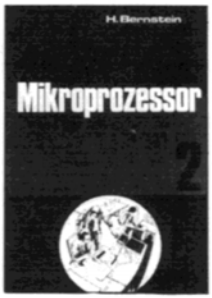
Achtung:
Unter Best.-Nr. 24 erschien bis vor kurzem das Buch Microcomputer-Technik von Blomeyer. Dieses ist vergriffen und wird nicht mehr neu aufgelegt.

ohne Abbildung

68000 Microcomputer Einführung
Eine leicht verständliche Anleitung zur Programmierung des leistungsfähigen 16 Bit Microcomputers von Motorola.
Erscheint ca. Mitte 1983.
Best.-Nr. 25 **39,00 DM**

Achtung:

Lieber Leser,
die Bestell-Nr. 25 und 27 waren früher Hobby Computer Handbuch (25) und Software Handbuch (27). Diese Titel sind vergriffen und werden vorerst nicht mehr neu aufgelegt.



Mikroprozessor, Teil 2, H. Bernstein
Die Fortsetzung von Best.-Nr. 22. Technologie von Mikroprozessor- und Speicherbausteinen. Festwertspeicher, PROM, EPROM, FIFO, Schieberegister, MPR-, ARL- und SAR-Register. Aufbau eines Mikroprozessorsystems mit 8080, RAM- und ROM-Schnittstellen. Befehlssatz 8080. Über 120 Seiten.
Best.-Nr. 26 **19,80 DM**



BASIC-M für Motorola EXORset®
Anwender Handbuch
Eine ausgezeichnete und professionelle Einführung in die BASIC-Programmiersprache. Ideal für die Industrie, aber auch ein ausgezeichnetes Werk für den Anfänger. Diese BASIC-Einführung mit vielen Beispielen gehört zu den besten deutschsprachigen Werken auf diesem Gebiet. Der Stoff geht auf den BASIC-M - Dialekt ein, ist aber auch auf jeder andere BASIC-Version anwendbar.
Best.-Nr. 27 **29,80 DM**



Microcomputer Lexikon u. Wörterbuch von A – Z, C. Lorenz
Englisch/Deutsch – Der Fachausdruck wird übersetzt, ausführlich erklärt und erläutert. Deutsch/Englisch – Übersetzung des Fachausdrucks. Ein Hilfs- und Arbeitsbuch für jeden, der sich heute mit der modernsten Elektronik beschäftigt. Viele engl. Ausdrücke werden heute in der Elektronik, Computer- und Mikroprozessortechnik verwendet und oft fehlt uns eine genaueste und präzise Erläuterung. Ein Lexikon und Wörterbuch in einem einzigen Buch vereint.
Best.-Nr. 28 **29,80 DM**



Microcomputer Datenbuch - (englisch) – ELCOMP
Endlich es es da ! Viele unserer Leser haben seit langem auf dieses Buch gewartet. – Und es hat sich gelohnt. Auf über 800 Seiten haben wir hier für Sie die wichtigsten (fast alle) Bauelemente zusammengestellt (Daten, Anschlußbilder etc.), die sich in den heutigen Personalcomputern befinden. Wir haben alle wichtigen Systeme durchforstet und die Bauelemente herausgesucht. TTL, CMOS, Linear, Spannungsregler, CPU-Schaltkreise, 6502, Z80, 8080, 8085, 8086, 1802, 2650, Z8 u. v. a.
Best.-Nr. 29 **49,– DM**

ohne Abbildung

Floppy Disk Selbstbau-Handbuch
von E. Flögel
Dieses Buch soll Ihnen auf einfachste Weise erklären, wie Sie Ihren Personalcomputer mit möglichst geringen Kosten mit einer Floppy Disk Station ersetzten können. Genaue Bauanleitung mit Software, Hinweise und Tips. Multiuser und Mutlitasking macht die Sache ganz besonders interessant.
Erscheint ca. Ende 1983
Best.-Nr. 30 **49,00 DM**

Achtung:
Unter Best.-Nr. 30 wurde früher ein Buch "Aktivtraining" angeboten. Dieses ist vergriffen und wird nicht mehr neu aufgelegt.



57 Programme in BASIC, Lorenz
Ein Buch mit techn.-wissenschaftlichen Programmen u. einer großen Anzahl von Spielprogrammen in BASIC (Games). Ein Buch für jeden, der sich mit dem faszinierenden Hobby der Mikrocomputertechnik befassen will. Alle Listings sind in BASIC und können auf den meisten Personal Computer Systemen gefahren werden.
Best.-Nr. 31 **39,00 DM**



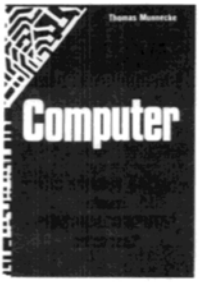
The Custom Apple & other Mysteries, von W. Hofacker und E. Flögel
Dieses Buch bringt auf 190 Seiten Großformat eine Vielzahl von Erweiterungsprojekten für Ihren APPLE II. 6522 Ein-/Ausgabekarte, Tonerzeugung mit AY-3-8912, Analog/Digital-Wandler, EPROM-Burner, Anschluß des 8253 von Intel an den APPLE II, Schrittmotorsteuerungen mit APPLE II in BASIC, PASCAL und FORTH. Zu den einzelnen Projekten sind die notwendigen Platinenvorlagen sowie die Software im Buch enthalten. Alle Platinen zum Buch können vom Hofacker Verlag sofort ab Lager bezogen werden.
Best.-Nr. 680 **(englisch)** **79,00 DM**



Mikrocomputer Programmierbeispiele für 2650, Dr. J. Hatzenbichler
Eine Einführung in die Programmierung von Mikrocomputern anhand des Prozessors 2650 von Signetics. Viele Programmierbeispiele in Maschinensprache, die Sie auf einem preiswerten Mikroprozessorsystem MIKIT 2650-P2 ausführen können. Zeitschleifenprogr., Blinkschaltung, Lauflicht, Stufenzähler, Stopuhr, Reaktionszeittester, u. a. Zu diesem Buch ist auch ein komplett aufgebautes und getestetes Mikrocomputersystem erhältlich, auf dem Sie alle beschriebenen Programme selbst ausführen können. Über 120 Seiten. (Nur solange Vorrat reicht, wird nicht mehr nachgedruckt.)
Best.-Nr. 33 **19,80 DM**



TINY BASIC Handbuch, Hermann
Das erste deutschsprachige Handbuch über Tom Pittman's TINY BASIC. Eine Einführung in die TINY BASIC-Programmiersprache. Wie kann ich meinen Computer (KIM-1) erweitern und BASIC programmieren. Systemvorschläge. Viele Programmierbeispiele, Tricks und Kniffe.
Best.-Nr. 34 **19,80 DM**



Der freundliche Computer – Was können Sie mit einem Personal Computer anfangen? T. Munnecke
Das Buch soll Ihnen auf die im Titel gestellte Frage eine ausführliche Antwort geben. Es eignet sich für alle, die bisher viel über Mikros gehört haben und gerne ausführlicher Bescheid wissen möchten. Viele interessante Fakten – Welche Computersprache ? Welche Anwendungen ? Welches Gerät soll ich mir kaufen ? 153 Seiten.
Best.-Nr. 35 **29,80 DM**

ohne Abbildung

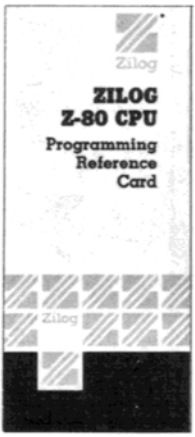
Microcomputer und Roboter
Ein Buch für denjenigen, der sich externe Schaltungen für Microcomputer bauen möchte, die roboterartige Funktionen ausführen können. Spracherkennung, Analog-/Digital-Wandler, Digital-/Analog-Wandler, Ultraschallsensoren, Lichtschranken, Tonerzeugung u. v. a. Erscheint Anfang 1982.
Best.-Nr. 36 **29,80 DM**

ohne Abbildung

Oszillographen Handbuch
Ein Buch für jeden, der seinen Oszillographen optimal nutzen will. Der Anfänger, der noch nie einen Oszillographen benutzt hat, wird auf einfache Weise mit der Technik und Handhabung vertraut gemacht. Auch die Anwendung in Zusammenhang mit den modernen Mikrocomputersystemen wird beschrieben. Oszillograph als alphanumeres Darstellgerät u. v. mehr. Erscheint ca. Anfang 1982.
Best.-Nr. 103 **19,80 DM**



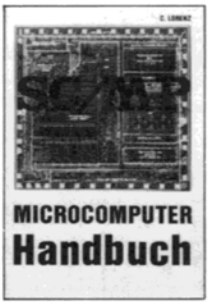
Z80 Assembler Handbuch
Dieses Handbuch braucht jeder, der ein Z80 Computersystem mit Z80 CPU hat (TRS-80, NEC, ALTOS, Sinclair, Video Genie usw.). Auf über 420 Seiten wird jeder Maschinenbefehl erklärt.
Best.-Nr. 8029 **29,80 DM**



Z-80 CPU Programming Reference Card
Eine praktische und kompakte Zusammenfassung der Z-80-Befehle, Z-80 Pinout, Registeraufbau, ASCII Zeichensatz kompletter Befehlssatz.
Best.-Nr. 252 **5,00 DM**



CMOS-Experimentierbuch
Eine kleine Einführung in die CMOS Schaltungstechnik. CMOS Grundlagen, Behandlungshinweise für CMOS Bausteine, Zusammenst. der wichtigsten CMOS Bausteine und deren Anschlußbilder. Viele praktische CMOS Schaltbeispiele. Ideal für jeden Elektroniker.
Best.-Nr. 106 1 **5,00 DM**



Handbuch für SC/MP
Ein echtes Handbuch für SC/MP (INS8060) Microcomputerbesitzer und solche die es werden wollen. Komplette Einführung mit vielen Schaltbeispielen, Schaltbildern und Programmlistings in Maschinensprache und TINY BASIC. Komplettes ELBUG-Listing, CPU-Karte, RAM-Karte, ROM-Karte, ROM-Programmierer, Cassetten-Interface, Fernsehinterface, Netzteil, Hex Ein-/Ausgabe, SC/MP Einkartenmicrocomputer u. v. a. 330 Seiten.
Best.-Nr. 108 **29,80 DM**



6502 Microcomputer Programmierung, P. Heuer
Eine deutschsprachige Einführung in die Maschinensprachenprogrammierung anhand des 6502 Microcomputers. Ein echtes Anleitungsbuch zum Einstieg in die Microcomputertechnik mit Hilfe des KIM-1. Viele Programmierbeispiele, die von einem Pädagogen speziell für Anfänger entwickelt wurden. Auch PET, AIM, SYM und ATARI-Besitzer brauchen dieses Buch.
Best.-Nr. 109 **29,80 DM**



Programmierhandbuch für PET
C. Lorenz
Ein für den PET produziertes Buch. Es beginnt da, wo Ihr mitgeliefertes Handbuch aufhört. Einführung Maschinenprogrammierung, Assembler, Ein-/Ausgabeprogrammierung, Programmiertricks, Analog / Digital-Wandler, Graphik, Spracherkennung u. v. a. mehr. Viele Listings, die Sie selbst eintippen können. Viele Informationen eignen sich auch für VC–20-Besitzer. 324 Seiten.
Best.-Nr. 110 **29,80 DM**



Programmieren mit TRS-80, Stübs
Das erste in einem deutschen Verlag produzierte Buch über den erfolgreichen Personal Computer von TANDY. Ein Buch für jeden, der einen TRS-80 bereits besitzt oder vor der Entscheidung steht, welchen Computer er sich anschaffen soll. Einführung, Programmiertricks, Erweiterungen, Maschinenprogrammierung und viele Programme (Listing mit Beschreibung). 202 Seiten.
Best.-Nr. 111 **29,80 DM**



BASIC Programmierhandbuch
Einführung und Nachschlagewerk. Speziell für die BASIC-Versionen der modernen Microcomputersysteme. Jeder Befehl wird ausführlich beschrieben und ein Beispielprogramm gezeigt. Sehr übersichtlich und praktisch. Am Schluß finden Sie ein komplettes BASIC-Programm, das Ihnen über einen Computer BASIC lehrt.
Best.-Nr. 113 **19,80 DM**



Der Microcomputer im Kleinbetrieb
Das Buch für jeden Geschäftsmann. Auf über 170 Seiten erfahren Sie, was Sie als Gewerbetreibender oder freiberuflich Tätige über Microcomputer und die Anwendung wissen sollten. Geschichtlicher Hintergrund Geräteauswahl, Beispiele aus der Praxis, Programmbeispiele wie z. B. Textverarbeitung, Reisebüro, Ladenkasse, Adressverwaltung, u. v. a. Betriebswirtschaftliche Auswertung, Finanzbuchhaltung, Erfolgsanalyse mit dem Microcomputer, Liquiditätsrechnung, kurzfristige Erfolgsrechnung, Microcomputer für Freiberufler, Grundlagen der Finanzbuchhaltung für Microcomputeranwender. Dieses Buch kann Ihnen als Geschäftsmann für die Zukunft tausende einsparen.
Best.-Nr. 114 **39,80 DM**



PASCAL Handbuch, **E. Flögel**
Von BASIC zu PASCAL. Ein Einführungs- Lehr und Arbeitsbuch für jeden der sich mit PASCAL beschäftigen will oder muß. Viele Programmbeispiele, viele Tricks wie PEEK und POKE, Einbinden von Maschinenprogrammen u. v. a.
Best.-Nr. 112 **29,80 DM**



16 Bit Microcomputer, **J. Koller**
Einführung, Daten, Eigenschaften, Anwendungen. Dieses Werk ist eine echte Sensation !
Alle 16 Bit Prozessoren werden beschrieben und erläutert. Applikationsbeispiele, Programmierhinweise. TMS 9900, 8086, Z8000, MC 68000, NS 16000, IAPX 486, IAPX 432. Über 370 Seiten.
Best.-Nr. 116 **29,80 DM**

ohne Abbildung

FORTRAN für Heimcomputer
Einführung in die FORTRAN-Programmiersprache mit vielen Beispielen. Grundsätzliches über die verschiedenen Microcomputersysteme, die bereits mit FORTRAN-Compiler lieferbar sind. Allgemeine Übersicht, Tips und Hinweise.
Erscheint ca. Ende 1982 .
Best.-Nr. 117 **19,80 DM**



Programmieren in Maschinensprache mit 6502, E. Flögel, W. Hofacker
Das deutschsprachige Werk über 6502 Maschinenprogrammierung. Einführung, Grundlagen, Eigenschaften, Adressierungsarten, Befehlsarten. Wie entwickelt man ein 6502 Maschinenprogramm? Handassemblierung, viele Programmbeispiele mit genauen Angaben direkt zum Eingeben in den Apple II mit Adressangabe (keine blutleeren Beispiele ohne Adresse), Verwendung von Assemblern. AIM-Assembler, Disassembler, Relocator, 6522 VIA, 6520, Interrupt, Fehlersuche in Maschinenprogrammen, Maschinensprache, Programmiertricks. Spezielle Abschnitte für Maschinensprachenprogrammierung über PET, CBM 3000, CBM 4000 u. VC-20, ATARI 400/800, Apple II, AIM sowie Ohio Scientific Challenger. Dieses Buch sollte jeder 6502 Systemanwender besitzen. Ca. 240 Seiten.
Best.-Nr. 118 **49,00 DM**



BASIC für Fortgeschrittene
Endlich ein BASIC-Buch für den fortgeschrittenen Programmierer. Alle wichtigen Befehle aus der Stringmanipulation, Disk-Befehle, WAIT, INSTR, WHILE WHEND usw. werden an Beispielen besprochen. Alle Befehle sind übersichtlich wie in einem Nachschlagwerk mit großen Überschriften angeordnet. Dann folgt ein umfangreicher BASIC Kurs für Fortgeschrittene mit vielen Beispielen (Inventur, Rechnungen schreiben, Adressenverwaltung usw.). Am Schluß finden Sie dann noch einen Vergleich der wichtigsten Sortiermethoden sowie ein Programm zur Vorhersage von Ereignissen.
Best.-Nr. 122 **39,00 DM**



Programmieren in Maschinensprache (Z80), C. Lorenz
Eine sehr ausführliche Einführung in die Z80 Maschinensprache mit vielen Beispielen. Die Beispiele können mit Hilfe des TRS-80 Level II sowie dem T-BUG von TANDY und den T-BUG-Erweiterungen (IN LOCO, T-STEP, T-LEGS) ausgeführt werden. Ein unentbehrliches Buch für jeden, dem die BASIC-Programmiersprache von der Geschwindigkeit her zur Lösung seiner Aufgaben nicht mehr ausreicht.
Best.-Nr. 119 **39,00 DM**

Anwenderprogramme für TRS-80 von Martin Stübs
Ein Buch, voll mit interessanten Anwenderprogrammen für TRS-80 Level II 16K und Video Genie (teilweise Diskette u./od. Cassette). Hauptsächlich Programme für den Manager, Geschäftsmann, Klein- und Mittelbetrieb. Auch einige interessante Spiele sind enthalten. Terminkalender, Reservierungsprogramm für Omnibusunternehmen und Hotels, Textverarbeitung, usw.
Best.-Nr. 120 **29,80 DM**





Microsoft BASIC-Handbuch
Die deutsche Übersetzung des erfolgreichen Microsoft BASIC-Handbooks. Leicht verständliche Einführung mit vielen interessanten Programmbeispielen. Das kompetente Werk von Microsoft selbst. Ideal als Zusatzliteratur zu jedem BASIC-Buch.
Best.-Nr. 121 **29,80 DM**



IEC Bus-Handbuch, M. P. Gottlob
Ein Handbuch und Nachschlagewerk für alle Besitzer von Computern mit IEC (IEEE 488 Bus). Dazu gehört auch der PET sowie alle CBM-Computer. Grundlagen, das BUS-System, Meßdatenübertragung, Adressierung eines Instruments, kl. IEC-BUS-Lexikon u.v.a.
Best.-Nr. 123 **19,80 DM**



Programmieren in Maschinensprache mit CBM
An Hand eines praktischen Beispiels (Sortierroutine) wird der Unterschied zwischen BASIC und Maschinenprogrammen gezeigt. Das Maschinenprogramm kann mit dem leistungsfähigen MONJANA/1 Monitor in ROM erstellt werden. Am Schluß finden Sie weitere wichtige Informationen wie Dez/Hex-Umrechnungstabelle, Befehlslisten, ASCII-Tabelle sowie eine ROM-Vergleichsliste zwischen 8k PET und den neuen CBM-Maschinen.
Best.-Nr. 124 **19,80 DM**

MONJANA Monitor im ROM
Best.-Nr. 1241 **79,00 DM**



Einführung in die Microcomputer Programmierung mit 6800
Eine sehr gute Einführung in die Microcomputertechnik mit Hilfe des Mikroprozessors 6800. Ausführliche Erklärungen mit vielen Beispielen und Anleitungen. Theoretische Grundlagen. CPU-Architektur, Befehlssatz, Systemaufbau, Hilfsmittel der Programmierung, Trainingsprogramme, Systemkomponenten, FIRMWARE. Ein komplettes Monitorprogramm (Betriebssystem) ist als Listing enthalten. Über 250 Seiten.
Best.-Nr. 127 **49,00 DM**



Programmierbeispiele für CBM
Ein Buch mit vielen BASIC-Programmen für CBM und PET. Spiele, Geschäftsbereich, Erziehung und Wissenschaft, Utilities, Hilfen für Maschinensprachenprogrammierung, trickreiche Programme. Viele Programme für wenig Geld.
Best.-Nr. 130 **19,80 DM**

ohne Abbildung

ELCOMP
Fachzeitschrift
für Microcomputertechnik

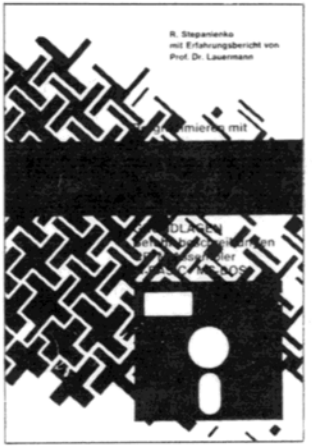
Die kompetente Fachzeitschrift für das moderne Gebiet der Microcomputertechnik. Erscheint 2 x pro Jahr. Preis pro Heft 29,80 DM incl. MwSt., Porto und Verpackung. Wer die neuesten Informationen aus diesem Gebiet für sich nützen möchte, muß ELCOMP lesen. Software, Technische Tips, Programmiertricks, Bauanleitungen, Systembeschreibungen, u. v. a.
Best.-Nr. 125 **29,80 DM**

ohne Abbildung

Programmieren mit dem CBM
Ein Hand- und Programmierbuch für alle CBM-Besitzer der 3000, 4000 sowie der 8000er Serie. Viele Tricks und Programmierbeispiele, Anleitungen.
Best.-Nr. 128 **29,80 DM**

ohne Abbildung

ELCOMP Leser Programmierhandb.
Hier fassen wir die besten Programme unserer ELCOMP-Leser zusammen. Programme für PET, CBM, TRS-80, AIM, Superboard, C4P, Exidy, Sharp, MZ80K, Apple II, Nascom I und II und TI 99/4 werden als Listing mit kurzer Beschreibung allen Lesern zugänglich gemacht. Erscheint Anfang 1982.
Best.-Nr. 129 **69,00 DM**



CP/M Handbuch
Grundlagen, Einführung, Hilfs- und Handbuch für jeden der mit dem "Software-Bus" arbeiten möchte. Ideal auch für Anfänger. Praktisches Handbuch für den Profi. Erscheint ca. Ende 1981.
Best.-Nr. 132 **19,80 DM**



FORTH Handbuch und Einführung von E. Flögel, W. Hofacker
FORTH ist nicht nur eine sehr leistungsfähige Programmiersprache – es ist schon fast eine "Religion". FORTH eignet sich bestens für industrielle Steuerungen, Grafik etc. Grundlagen und viele Programmbeispiele (Apple II, Ohio, ATARI usw.). Erscheint Mitte 1982.
Best.-Nr. 137 **39,00 DM**



BASIC für blutige Laien, Matthaei
Endlich ein Buch für den Anfänger und Laien. Ziel dies Buches ist es, dem "blutigen Laien" die Grundlage der Programmiersprache BASIC zu vermitteln. Lernen wird nun zum echten Vergnügen und Freizeitspaß. Auch der Preis macht Spaß.
Best.-Nr. 139 **nur 19,80 DM**



Programmieren in BASIC und Maschinencode mit dem ZX81, E.Flögel
Ein Buch für Sinclair ZX81 Besitzer und solche die es werden wollen. Was heißt Programmieren?, Programmerstellung in BASIC, Spiele, Spielelemente, Programme für die Schule, Datenverwaltungsprogramme, Lagerbestand, Schallplattenverzeichnis, Programmieren in Z80 Maschinensprache, Anschluß einer PIO und externer Schaltungen, Lösen von digitalen Steuerungsaufgaben mit dem ZX81 u. v. a. Dieses Buch gehört auf den Tisch eines jeden ZX81 Besitzers (mehr als 20 komplette Programme, Maschinensprachen-Monitor, etc.).
Best.-Nr. 140 **29,80 DM**



Programme für den VC-20 von W. Hofacker
Ein Buch des Hofacker Verlages speziell für den VC-20 Volkscomputer von Commodore. Viele komplette Programme wie Wortprozessor, Maschinensprachenmonitor, lustige Spiele, Programmieren in Maschinensprache, Ein-/Ausgabeprogrammierung. Wichtige Adressen des Betriebssystems, Tabellen, Speichererweiterungen, Dual-Joystick Bauanleitung u. v. a. Sie werden begeistert sein.
Best.-Nr. 141 **29,80 DM**



Astrologie mit dem Personal-Computer ATARI 800, W. Hofacker
Ein Blick in die Zukunft. Erstellen Sie selbst Ihr eigenes Horroskop. Dieses Büchlein zeigt dem interessanten Leser wie man sein eigenes Horroskop mit professioneller Genauigkeit berechnen kann. Was braucht man dazu? Wie geht man vor? Was hat es mit der Astrologie auf sich? Was braucht man für die Deutung? Enthält ein komplettes Listing in BASIC und Ma.-Code (ATARI 800, 48K RAM + Disk).
Best.-Nr. 175 **49,00 DM**

ELCOMP-Bücher Englisch

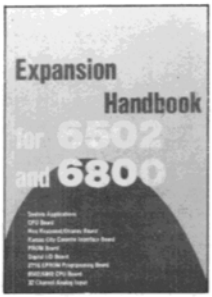


Care and Feeding of the Commodore PET
Das ideale Buch für den Hardware-Bastler. Viele Tricks, Schaltbilder, Hinweise und Erläuterungen für den, der gerne selbst Erweiterungen bauen möchte. Memory Map für 8k PET und CBM, Bauanleitung für eine serielle Schnittstelle u. v. a.
Best.-Nr. 150 **19,80 DM**



Complex Sound Generation with SN 76477
Ein Applikationsheft für einen der interessantesten integrierten Bausteine unserer Zeit. Ein LSI-Baustein zur Tonerzeugung. Je nach äußerer Beschaltung können Sie mit diesem Baustein die verrücktesten Töne erzeugen. Dampfeisenbahngeräusch mit Dampfpfeife, Vogelgezwitscher, Hundegebell, elektronische Orgel, Schuß mit Explosion u. v. a. mehr.
Best.-Nr. 154 9,80 DM

Expansion Handbook for 6502 and 6800
Das ideale Handbuch für alle KIM, SYM, AIM, PET und Challenger Computer-Freunde. Das Buch beschäftigt sich ausschließlich mit dem S-44-Bus. Dies ist exakt der Bus von SYM, AIM und KIM. Sehr viele Schaltbilder: CPU-Platine, Hex-Tastatur Eingabe, Knsas City Interface, RAM u. ROM-Karte, Analog-Eingabe Board u. v. a. Das Buch ist für jeden 6502 Systembesitzer unentbehrlich. Ca. 150 Seiten.
Best.-Nr. 152 **19,80 DM**

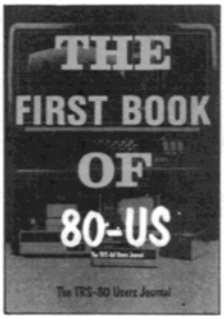




Microsoft 8k BASIC Reference Manual
Eine sehr gute BASIC-Einführung. Auch als Handbuch zum Nachschlagen bestens geeignet. Ideal für jeden PET, CBM, TRS-80, KIM-BASIC, SYM-BASIC, AIM- und APPLE-Besitzer. 73 Seiten DIN A4 mit vielen Beispielen.
Best.-Nr. 151 9,80 DM



Intel Application Notes (8080, 8085, 8255, 8251)
Dieses Buch braucht jeder, der mit 8080, 8085 oder Z-80 Mikroprozessoren arbeitet.
Wir haben die interessantesten Applikationsberichte in diesem Buch zusammgefasst. Aus dem Inhalt: Designing with Intel's Static RAM's 2102, Memory Design with the Intel 2107B. 8255 Programmable Peripheral Interface Applications, Using the 8202 Dynamic RAM Controller u. v. a.
Best.-Nr. 153 **29,80 DM**



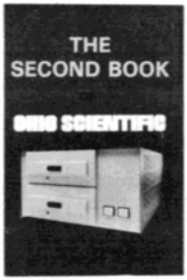
The First Book of 80-US
Für den TRS-80 Freund eine echte Preissensation. Die ersten fünf Hefte aus 80-US Journals in einem Sammelband zusammengefaßt. Voll mit vielen sehr interessanten Hard- und Softwareideen, Tricks. Viele komplette Programmbeispiele (Listings) in BASIC u. Z-80 Maschinensprache. Über 250 Seiten DIN A4. Farbiger Umschlag. Dieses Buch sollte jeder TRS-80 Besitzer oder der es werden will im Schrank haben.
Best.-Nr. 155 **29,80 DM**
(solange Vorrat reicht)



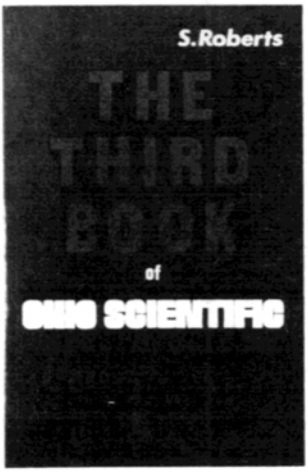
Small Business Programs, S. Roberts
Ein Buch für denjenigen, der die modernen Microcomputer (speziell TRS-80, Apple, PET, North Star, Challenger) zur Rationalisierung in seinem Klein- oder Mittelbetrieb einsetzen möchte. Viele nützliche Tips, Hinweise und Programmierbeispiele. Dieses Buch sollte jeder Geschäftsmann u. Microcomputerfreund besitzen.
Best.-Nr. 156 **29,80 DM**



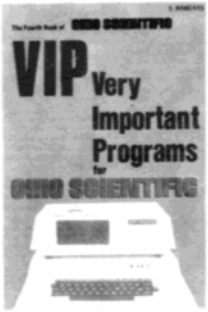
The first Book of Ohio Scientific
Das erste weltweit produzierte Buch für die erfolgreiche Ohio Scientific Challenger Computerserie. Grundlagen, viele Programmiertricks Hardwaretips, Umbauanleitungen, Programmierbeispiele u. v. a. Glanzumschlag, 186 Seiten.
Best.-Nr. 157 **19,80 DM**
(solange Vorrat reicht)



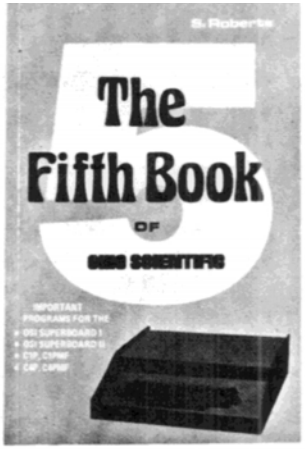
The second Book of Ohio Scientific
Eingehende Beschreibungen über praktische und geschäftsorientierte Software. Speicher Test Programm, Tricks und Tips für Disketten-Anwender. Mini-Floppy-Expansion u. v. a. 159 Seiten.
Best.-Nr. 158 **19,80 DM**



The third book of Ohio
Wie erweitere ich mein Challenger System ? Universelle I/O-Karte, EPROM-Burner für 2716, EPROM, RAM-Karte, 6522 VIA-Karte. Wo notwendig mit kompletter Software. Dieses Buch braucht jeder Ohio-Benutzer. Komplette Schaltbilder und Aufbauhinweise.
Best.-Nr. 159 **19,80 DM**



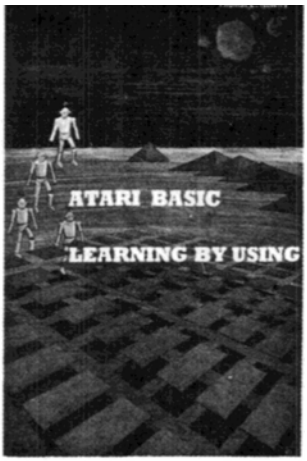
The fourth Book of Ohio Scientific
Ein Buch voll mit Programmen für das Superboard, C4P, C4PMF und C28P. Die Softwarequelle für jeden Challenger-Fan. Alle Programme sind getestet und auch auf Cassette verfügbar. 170 Seiten Listings und Beschreibungen.
Best.-Nr. 160 **29,80 DM**
Best.-Nr. 8324 Cassette 29,80 DM



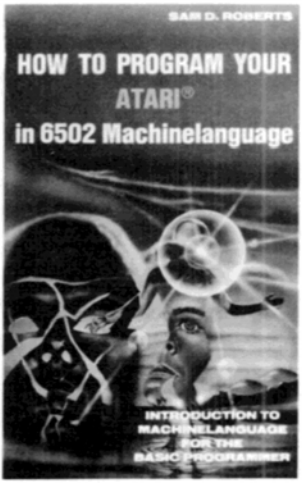
The Fifth Book of Ohio Scientific
Kein anderer Verlag der Welt hat fünf Bände für den so erfolgreichen und leistungsfähigen Personalcomputer bisher produziert. Der 5 Band bringt wieder eine Vielzahl von sehr interessanten BASIC- und 6502 Maschinenprogrammen. Neben einer Adressenverwaltung, einem Textprogramm, Fakturierprogrammen und Utilities finden Sie auch wieder viele interessante Spiele sowie eine komplette Abhandlung über Start- und Landesimulationen, die sich besonders als Hilfsmittel zur Entwicklung eigener Mondlandespiele eignen. Dem Buch kann vom erfahrenen BASIC-Programmierer auch viel Stoff zur Implementation f. andere Rechner entnommen werden (englisch).
Best.-Nr. 161 **19,80 DM**



ZX-81 / TIMEX
Programming in BASIC and Machine Language by E. Flögel
Ein Buch mit vielen Tips und Programmen für den erfolgreichen Sinclair ZX-81 Personalcomputer. Dieses Buch ist eine Übersetzung unseres Titels Nr. 140.
Sehr interessant ist die Anleitung zum Aufbau einer Z-80 PIO an den ZX-81.
Best.-Nr. 174 **29,80 DM**



ATARI BASIC-Learning by USING v. Thomas E. Rowley (engl.)
Ein echtes Action-Buch für Ihren ATARI 400/800. Hier findet mehr statt als nur lesen. Sie verwenden es und machen neue Entdeckungen. Viele nützliche Routinen und Hilfsprogramme. Grafik, Tonerzeugung, Joystickprogrammierung, PEEK und POKE und Special-"Struff".
Best.-Nr. 164 **19,80 DM**



How to program your ATARI in 6502 Machinelanguage, Roberts
Eine sehr gute Einführung in die 6502 Maschinensprache mit dem ATARI 400/800.
Im einzelnen leicht verständlich und überschaubaren Lektionen werden Sie an Hand von praktischen Beispielen in die wunderbare Welt der Maschinensprache geführt. Sehr viele interessante Programmbeispiele für den ATARI. Tricks, Kniffe und Tips.
Best.-Nr. 169 **29,80 DM**



Programming in 6502 Machinelanguage for PET + CBM
Dieses Buch enthält eine große Auswahl sehr wertvoller Informationen für den PET und CBM Besitzer der 3000er und 2000er Serie. Neben dem Listing und Beschreibung für einen sehr leistungsfähigen EDITOR/Assembler in Maschinensprache mit Beispiel für den Sie einen Disassembler, Linker, Editor, Assembler in BASIC und Maschinencode sowie einen kompletten Maschinensprachenmonitor für PET 2001. (engl.)
Best.-Nr. 166 **49,00 DM**

Hiermit gebe ich folgende Bestellung unter Anerkennung Ihrer Lieferungs- und Zahlungsbedingungen auf.

ELCOMP

Menge	Artikelbezeichnung	Einzelpreis

Menge	Artikelbezeichnung	Einzelpreis

Unterschrift (für Jugendliche unter 18 J. der Erziehungsberechtigte) Datum

---

Hiermit bestelle ich die ELCOMP-Sonderhefte bis auf Widerruf.

○ Ich bitte um Abbuchung von DM 29,80 (incl. Versand und Verpackung) von meinem Konto (Giro oder Postscheckkonto)

Kto.-Nr. ..... BLZ .....

Geldinstitut: ..... Ort: .....

○ Ich bitte um Lieferung per Nachnahme. Mit der Lieferung eines Heftes wird der Betrag von DM 29,80 + DM 6,50 NN-Gebühr erhoben.

Name/Vorname

Straße/Haus-Nr.

PLZ Ort

Datum..... Unterschrift .....

ELCOMP

POSTKARTE

Absender
Bitte deutlich ausfüllen

Vorname/Name

Beruf

Straße/Nr.

Plz Ort

ELCOMP

MIKROCOMPUTER BOOK STORE

Tegernseerstr. 18

D-8150 Holzkirchen /Obb.

ABSENDER:

Name, Vorname

Straße

( ) .....

PLZ Ort

Telefon

Die Bestellung der Sonderhefte kann JEDERZEIT widerrufen werden. Die Belieferung wird zum nächst erscheinenden Heft eingestellt.
Eine Kündigung NACH Auslieferung eines Sonderheftes (rückwirkend) ist NICHT möglich.

ELCOMP

Ing. W. Hofacker GmbH
Tegernseer Straße 18
D-8150 Holzkirchen

Leercassetten für Microcomputer



C–10

Die ideale Cassettenlänge für Ihren Personalcomputer.
Praktisch – handlich und betriebssicher

Kassetten mit nur 10 Minuten Spieldauer (2 x 5 Minuten) haben sich zur Aufzeichnung von Daten im Mikrocomputerbereich bestens bewährt.

Vorteile der C-10 Computer Cassette vom HOFACKER Verlag:

- weniger Bandsalat
- kurze Rückspulzeiten
- schnelles Auffinden von Programmen
- bessere Gleichlaufeigenschaften
- einfache Programmverwaltung

Die C-10 HOFACKER Datencassette bietet weiterhin:

- extrem hoch aussteuerbares Bandmaterial (Agfa)
- hochwertiges Cassettengehäuse, 5fach verschraubt
- Tefloneinlage für gute Laufruhe
- Staubdichtes Glasfenster

Die C-10 HOFACKER Datencassette wird seit 1978 speziell für Microcomputeranwender produziert. Die Cassetten bieten ein Höchstmaß an Betriebssicherheit bezüglich fehlerfreier Aufnahme und Wiedergabe.

Hier eine kurze Übersicht über die Anzahl der Bytes, die Sie auf eine C-10 Cassette abspeichern können:

Computer	Speichermöglichkeit	Computer	Speichermöglichkeit
ATARI 400/800	16K	APPLE	36K
Sharp MZ-80	32K	APPLE II	16K
AIM 65	16K	Heathkit	36K
Ohio Scientific	10K	Kansas City Std.	16K
TRS-80	16K	KIM-1	12K
TRS-80 Color Computer	24K	NASCOM	12K
Video Genie	16K	Exidy Sorcerer	12K
Sinclair ZX80/81	16K	SYM-1	12K

BESTELLSCHEIN

Menge	Beschreibung	Preis/DM	Gesamt
	1 Cassette	3,50	
	10 Cassetten	29,80	
	100 Cassetten	249,00	

Lieferanschrift

Name. . . . . .
Straße . . . . . .
Ort . . . . . .

Datum. . . . . Unterschrift . . . . . .

HOFACKER

Ing. W. Hofacker GmbH
Tegernseerstr. 18
D–8150 Holzkirchen
Tel.: (0 80 24) 73 31

Weitere interessante Bücher von Hofacker:

Best.-Nr.	Titel	Preis/DM
	Bücher in deutscher Sprache aus dem Hofacker-Verlag	
1	Transistor Berechnungs- und Bauanleitungsbuch – 1	29,80
2	Transistor Berechnungs- und Bauanleitungsbuch – 2	19,80
3	Elektronik im Auto	9,80
4	IC-Handbuch, TTL, CMOS, Linear	19,80
5	IC-Datenbuch, TTL, CMOS, Linear	9,80
6	IC-Schaltungen, TTL, CMOS, Linear (i. V.)	19,80
7	Elektronik Schaltungen (i. V.)	19,80
8	IC-Bauanleitungs-Handbuch	19,80
9	Feldeffekttransistoren (i. V.)	9,80
10	Elektronik und Radio	19,80
11	IC-NF Verstärker (i. V.)	9,80
12	Beispiele integrierter Schaltungen (BIS)	19,80
13	HEH, Hobby Elektronik Handbuch	9,80
15	Optoelektronik Handbuch	19,80
16	CMOS Teil 1, Einführung, Entwurf, Schaltbeispiele	19,80
17	CMOS Teil 2, Entwurf und Schaltbeispiele	19,80
18	CMOS Teil 3, Entwurf und Schaltbeispiele	19,80
19	IC-Experimentier Handbuch	19,80
20	Operationsverstärker	19,80
21	Digitaltechnik Grundkurs	19,80
22	Mikroprozessoren, Eigenschaften und Aufbau	19,80
23	Elektronik Grundkurs, Kurzlehrgang Elektronik	9,80
24	Progr. in Maschinensprache mit Z80, Band II	29,80
25	68000 Microcomputer Einführung (i. V.)	39,00
26	Mikroprozessor, Teil 2	19,80
27	BASIC-M Anwender-HB f. 6800/09/68000 (Motorola)	29,80
28	Lexikon + Wörterbuch f. Elektr. u. Mikroprozessor	29,80
29	Mikrocomputer Datenbuch (englisch)	49,00
30	Floppy Disk Selbstbau-Handbuch (i. V.)	49,00
31	57 Programme in BASIC	39,00
33	Microcomputer Programmierbeispiele	19,80
34	TINY-BASIC Handbuch	19,80
35	Der freundliche Computer	29,80
103	Oszillographen-Handbuch	19,80
108	Rund um den Spectrum (Progr., Tips u. Tricks)	29,80
109	6502 Microcomputer Programmierung	29,80
110	Programmierhandbuch für PET	29,80
111	Programmieren mit TRS-80 (Video Genie)	29,80
112	PASCAL-Programmier-Handbuch	29,80
113	BASIC-Programmier-Handbuch	19,80
114	Der Microcomputer im Kleinbetrieb	39,80
115	6809 Programmier Handbuch (i. V.)	49,00
116	Einführung 16-Bit Micorocomputer	29,80
117	FORTRAN für Heimcomputer	19,80
118	Programmieren in Maschinensprache mit dem 6502	49,00
119	Programmieren in Maschinensprache (Z80) Band I	39,00
120	Anwenderprogramme für TRS-80 u. Video Genie	29,80
121	Microsoft BASIC-Handbuch	29,80
122	BASIC für Fortgeschrittene	39,00
123	IEC-Bus Handbuch	19,80
124	Progr. i. Ma.-Spr. mit CBM, VC-20, C-64 (i. V.)	19,80
127	Einführung i. d. Microcmputer-Progr. mit 6800	49,00
128	Programmieren mit dem CBM	29,80
130	Programmierbeispiele für CBM	19,80
132	CP/M-Handbuch	19,80
133	Handbuch für MS/DOS (i. V.)	29,80
137	FORTH Handbuch	49,00
139	BASIC für blutige Laien	19,80
140	Progr. i. BASIC u. Maschinencode mit dem ZX81	29,80
141	Progr. für VC-20 (Spiele, Utilities, Erweiterungen)	29,80
143	35 Programme für den ZX81	29,80
144	33 Programme für den ZX-Spectrum	29,80
145	64 Programme für den Commodore 64 (i. V.)	39,00
146	Hardware-Erweiterungen für den C-64 (i. V.)	39,00
147	Beherrschen Sie Ihren Commodore 64	19,80
148	Programmierhandbuch für Sharp	49,00
149	Programme für TI 99/A (i. V.)	49,00
175	Astrologie auf dem ATARI 800	49,00
8029	Z-80 Assembler-Handbuch	29,80
	Bücher in englischer Sprache	
	1. Von ELCOMP Publishing, Inc., Los Angeles, CA.	
150	Care and Feeding of the Commodore PET	19,80
151	8K Microsoft BASIC Reference Manual	9,80
152	Expansion Handbook for 6502 and 6800	19,80
154	Complex Sound Generation using the SN76477	9,80
156	Small Business Programs	29,80
158	The Second Book of Ohio Scientific	19,80
159	The Third Book of Ohio Scientific	19,80
160	The Fourth Book of Ohio Scientific	29,80
161	The Fifth Book of Ohio Scientific	19,80
162	ATARI Games in BASIC	19,80
163	The Peripheral Handbook (i. V.)	29,80
164	ATARI-BASIC Learning by Using	19,80
166	Programming in 6502 Machinelanguage PET/CBM	49,00
169	How to Progr. your ATARI in 6502 Machinelanguage	29,80
170	FORTH on the ATARI – Learning by Using	29,80
171	See the Future with your ATARI (Astrology)	49,00
172	Hackerbook I (Tricks + Tips for your ATARI	29,80
173	PD-Program Descriptions (ATARI)	9,80
174	ZX-81/TIMEX Progr. i. BASIC a. Machine Lang.	29,80
176	Programs + Tricks for VIC's	29,80
177	CP/M – MBASIC and the OSBORNE (i. V.)	29,80
	2. Von IJG Inc., Upland Californien	
680	The Custom Apple & other Mysteries	79,00
681	Machine Language Disk I/O (TRS-80)	129,00
	Riesenprogrammsammlung in BASIC	
8048	BASIC Software Vol. VI	139,00
8049	BASIC Software Vol. VII	109,00
8050	BASIC Software Vol. I	69,00
8051	BASIC Software Vol. II	69,00
8052	BASIC Software Vol. III	104,00
8053	BASIC Software Vol. IV	29,00
8054	BASIC Software Vol. V	29,00

Der Hofacker Verlag produziert und vertreibt neben einer sehr großen Auswahl an Fachbüchern für Elektronik und Microcomputertechnik noch:
- Leerplatinen und Bauanleitungen für Zusatzeinrichtungen für Ihren Personalcomputer, sowie
- Programme (Software) für die bedeutenden Personalcomputer.

(i. V. bedeutet: Buch ist in Vorbereitung)

HOFACKER

HOLZKIRCHEN · SINGAPORE · LOS ANGELES

ISBN 3–88963–144–4