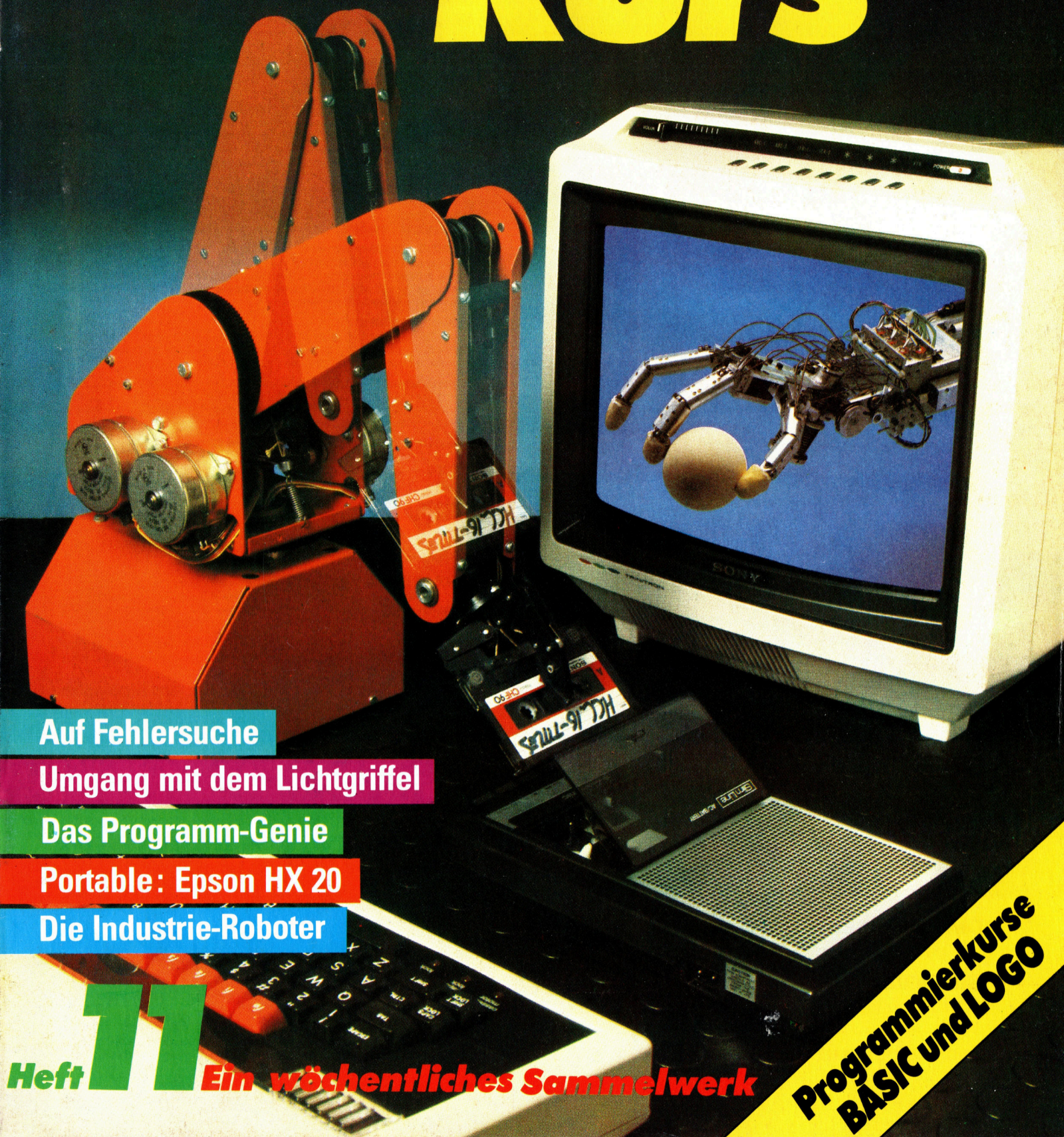


**Einsteigen - Verstehen - Beherrschen**

DM 3,80 öS 30 sfr 3,80

# computer kurs



Auf Fehlersuche

Umgang mit dem Lichtgriffel

Das Programm-Genie

Portable: Epson HX 20

Die Industrie-Roboter

Heft **11** Ein wöchentliches Sammelwerk

Programmierkurse  
BASIC und LOGO



# computer kurs

## Heft 11

### Inhalt

#### Computer Welt



**„Kollege“ Roboter** 281  
Computer-Einsatz in der Massenproduktion

**Pinball Wizard** 304  
Der Flipper-Baukasten von Electronic Arts

#### Hardware



**Epson HX-20** 287

**Magischer Kristall** 290  
LCDs in Taschenrechnern und Computern

#### Software



**Das Programm-Genie** 284  
Personalcomputer ‚Lisa‘ von Apple

**Micro-Labyrinth** 300  
Irrgärten als Programm-Grundlage

#### BASIC 11



**Unter Kontrolle** 296

#### Peripherie



**Der Zauberstab** 294  
Lichtgriffel zur Programmwahl

#### Tips für die Praxis



**Der tönende VC** 292

**Der zeichnende Dragon** 293

#### LOGO 11



**Stück für Stück . . .** 302

#### Bits und Bytes



**Detektivarbeit** 306  
Aufspüren und Beseitigen von gekippten Bits

**Auf Fehlersuche** 308  
„Gerade Parität“ beim Datenaustausch

#### Fachwörter auf einen Blick

### WIE SIE JEDE WOCHE IHR HEFT BEKOMMEN

Computer Kurs ist ein wöchentlich erscheinendes Sammelwerk. Die Gesamtzahl der Hefte ergibt ein vollständiges Computer-Nachschlagewerk. Damit Sie jede Woche Ihr Heft erhalten, bitten Sie Ihren Zeitschriftenhändler, Computer Kurs für Sie zu reservieren.

#### Zurückliegende Hefte

Ihr Zeitschriftenhändler besorgt Ihnen gerne zurückliegende Hefte. Sie können sie aber auch direkt beim Verlag bestellen.

**Deutschland:** Das einzelne Heft kostet DM 3,80. Bitte füllen Sie eine Postzahlkarte aus an: Marshall Cavendish Int. Ltd. (MCI), Sammelwerk-Service, Postgiroamt Hamburg 48064-202, Postfach 105703, 2000 Hamburg 1, Kennwort: Computer Kurs

**Österreich:** Das einzelne Heft kostet öS 30. Bitte füllen Sie eine Zahlkarte aus an: Computer Kurs, Wollzeile 11, 1011 Wien, Postscheckkonto Wien 7857201 oder legen Sie Ihrer Bestellung einen Verrechnungsscheck bei. Kennwort: Computer Kurs.

**Schweiz:** Das einzelne Heft kostet sfr 3,80. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Computer Kurs, und notieren Sie ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

#### Abonnement

Sie können Computer Kurs auch alle 2 Wochen (je 2 Ausgaben) per Post zum gleichen Preis im Abonnement beziehen. Der Abopreis für 12 Ausgaben beträgt DM 45,60 inkl. Mwst., den wir Ihnen nach Eingang der Bestellung berechnen. Bitte senden Sie Ihre Bestellung an: Marshall Cavendish Int. Ltd. (MCI), Sammelwerk Service, Postgiroamt Hamburg 86853-201, Postfach 105703, 2000 Hamburg 1, Kennwort: Abo Computer Kurs. Bitte geben Sie an, ab welcher Nummer das Abo beginnen soll und ob Sie regelmäßig für jeweils 12 Folgen einen Sammelordner wünschen. Bei Bestellungen aus Österreich oder Schweiz senden Sie Ihren Auftrag bitte auch an die Hamburger Adresse. Berechnung und Zahlung erfolgen in Landeswährung zum Ladenpreis.

**WICHTIG: Bei Ihren Bestellungen muß der linke Abschnitt der Zahlkarte Ihre vollständige Adresse enthalten, damit Sie die Hefte schnell und sicher erhalten. Überweisen Sie durch Ihre Bank, so muß die Überweiskopie Ihre vollständige Anschrift gut lesbar enthalten.**

#### SAMMELORDNER

Sie können die Sammelordner entweder direkt bei Ihrem Zeitschriftenhändler kaufen (falls nicht vorrätig, bestellt er sie gerne für Sie) oder aber Sie bestellen die Sammelordner für den gleichen Preis beim Verlag wie folgt:

**Deutschland:** Der Sammelordner kostet DM 12. Bitte füllen Sie eine Zahlkarte aus an: Marshall Cavendish International Ltd. (MCI), Sammelwerk-Service, Postgiroamt Hamburg 48064-202, Postfach 105703, 2000 Hamburg 1, Kennwort: Sammelordner Computer Kurs.

**Österreich:** Der Sammelordner kostet öS 98. Bitte füllen Sie eine Zahlkarte aus an: Computer Kurs Wollzeile 11, 1011 Wien, Postscheckkonto Wien 7857201 oder legen Sie Ihrer Bestellung einen Verrechnungsscheck bei. Kennwort: Sammelordner Computer Kurs

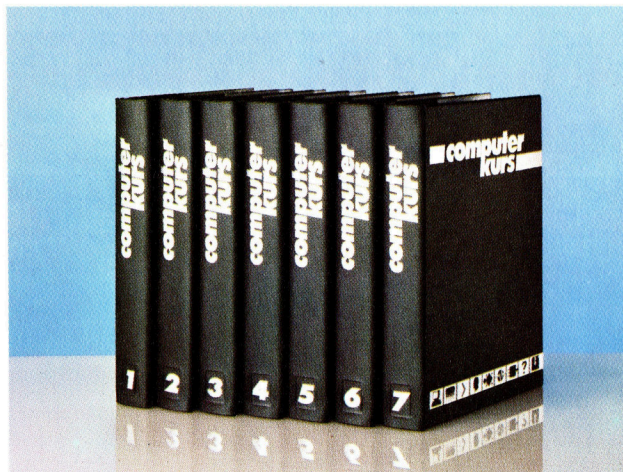
**Schweiz:** Der Sammelordner kostet sfr 15. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Sammelordner Computer Kurs, und notieren Sie Ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

#### INHALTSVERZEICHNIS

Alle 12 Hefte erscheint ein Teilindex. Die letzte Ausgabe von Computer Kurs enthält den Gesamtindex — darin einbezogen sind Kreuzverweise auf die Artikel, die mit dem gesuchten Stichwort in Verbindung stehen.

**Redaktion:** Winfried Schmidt (verantw. f. d. Inhalt), Joachim Seidel, Elke Leibinger, Susanne Brandt, Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

**Vertrieb:** Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1, Tel.: 040/23 40 85



© APSIF, Copenhagen, 1982, 1983; © Orbis Publishing Ltd., 1982, 1983; © Marshall Cavendish Ltd., 1984, 1985; **Druck:** E. Schwend GmbH, Schmollerstraße 31, 7170 Schwäbisch Hall





# „Kollege“ Roboter

**Computergesteuerte Fertigungsroboter findet man in nahezu jeder modernen Fabrikhalle, die auf Massenproduktion von industriellen Gütern ausgelegt ist.**

**D**er Begriff „Roboter“ ist von dem tschechischen Wort „robota“ (= arbeiten) abgeleitet und wurde von dem Bühnenschriftsteller Karel Čapek für sein 1920 geschaffenes Theaterstück R.U.R. (Rossum's Universalroboter) geprägt. Er entwickelte sich seither zu einem Standardbegriff der Science-Fiction-Literatur. Trotz vieler Fähigkeiten, die man Robotern andichtete, bleiben sie elektromechanische Geräte, die von Computern gesteuert werden und damit deren Grenzen und Mängeln unterworfen sind.

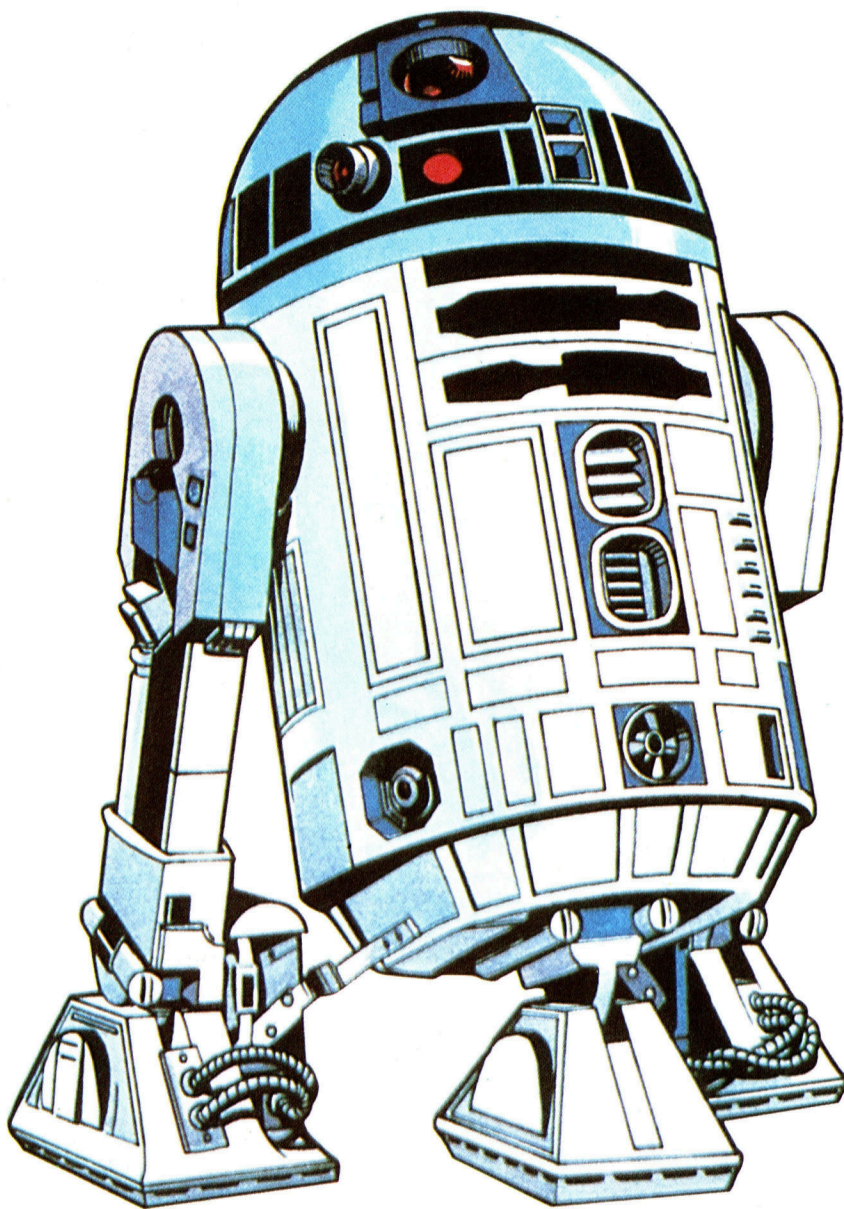
Die Ursprünge der Roboter lassen sich in die Fabrikhallen der fünfziger Jahre zurückverfolgen, in denen zum ersten Mal eine Maschinensteuerung über Zahlencodes eingesetzt wurde. Diese ersten Versuche waren noch recht primitiv: Die Steuerung arbeitete mit Lochstreifen (wie bei Telexgeräten) und konnte festmontierte Werkzeuge nur um ein Werkstück herum bewegen und an verschiedenen Punkten anhalten.

## Programmgesteuerte Werkzeuge

Im nächsten Entwicklungsstadium waren die Geräte dazu fähig, die Werkzeuge während eines Arbeitsvorgangs zu wechseln. Dabei entnahm die Maschine programmgesteuert aus einem „Karussell“ oder rotierenden Ständer der Werkzeuge, die mit identischen Haltevorrichtungen versehen waren, und setzte sie in einen normierten Halter ein.

Aber auch in diesem Entwicklungsstadium ließ sich mit einer programmgesteuerten Maschine immer nur eine Aufgabe ausführen: Eine Drehbank blieb trotzdem eine Drehbank, selbst wenn alle Arbeitsgänge an einem bestimmten Werkstück automatisch ausgeführt werden konnten. Gerade zu dieser Zeit eröffnete sich aber ein völlig neuer Ansatz: Für gefährliche Arbeiten, z. B. unter Wasser oder beim Umgang mit radioaktiven Materialien, wurden ferngesteuerte Greifarme entwickelt. Zunächst nur als Verlängerung des menschlichen Arms gedacht, setzte man schon bald Computer für die Steuerung dieser Arbeitshilfen ein. Seitdem werden Geräte dieser Art auch nicht mehr als „Roboter“ bezeichnet, sondern als „computergesteuerte Fertigungsarme“. Tatsächlich bestehen sie seitdem aus einem beweglichen Arm, an dessen Ende sich eine Haltevorrichtung für Werkzeuge befindet.

Um die Programmierung von Fertigungsro-



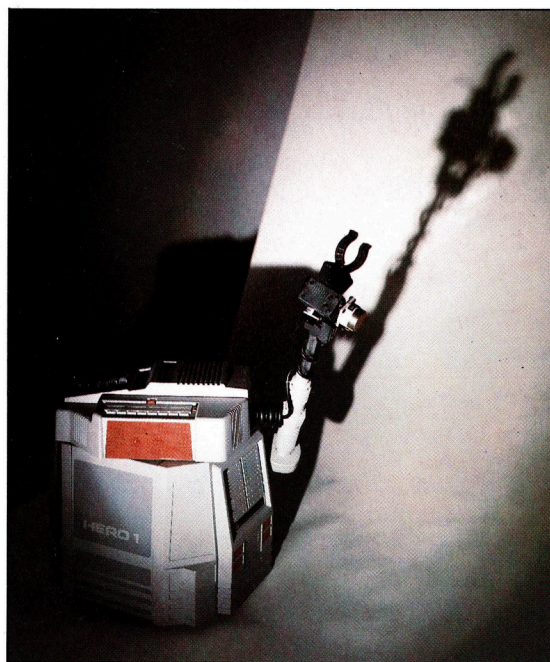
botern verstehen zu können, müssen wir uns zunächst mit dem Raum beschäftigen, in dem sie ihre Bewegungen ausführen. Da die meisten Industrieroboter fest an einem Platz montiert sind, hat dieser Bewegungsraum die Form einer Kugel, die an der Stelle des Bodens flach aufliegt. Die Steuerung eines Roboters läßt sich jetzt leicht als dreidimensionale Geometriaufgabe darstellen. Zentrum der Kugel ist das „Schultergelenk“ des Roboters, während die maximale Länge des Armes – gemessen von der „Schulter“ bis zu den „Fingern“, d. h.

**R2D2, der sympathische Roboter aus dem Film ‚Krieg der Sterne‘, wurde de facto von Menschen gesteuert. Seine Konstruktion entsprach jedoch dem Bild, das sich viele Menschen von einem Roboter machen.**





Der HERO-1 ist ein eigenständiger batteriebetriebener Roboter, der in sich die Funktionen einer Schildkröte und eines Fertigungsarmes vereint. Auf den ersten Blick scheint er nur ein teures Spielzeug zu sein. Tatsächlich aber befindet sich in ihm ein erstaunlich flexibles Computersystem mit hochentwickelten Fähigkeiten wie z. B. Sprachsynthese und Eingabe, Sensoren für unterschiedliche Grade von Helligkeit und ein Ultraschallgerät für das Erkennen von Hindernissen (der HERO-1 ist fahrbar), das auch als Bewegungsfühler eingesetzt werden kann.



dem Greifer oder Werkzeughalter, den Radius darstellt. Jeder Punkt innerhalb dieser Kugel kann durch die Angabe von drei Koordinaten beschrieben werden: z. B. als Entfernung in Richtung Nord/Süd, Ost/West und auf/ab – immer bezogen auf einen Ausgangs- oder Nullpunkt. Bei dieser Darstellungsmethode bezeichnet man die Koordinaten nach dem französischen Mathematiker René Descartes als kartesisch.

Da bei der Programmierung eines Roboters aber auch das Problem der Bewegung von einem relativen Ort zum anderen auftreten kann, gibt es noch eine dritte Möglichkeit, die Position der Werkzeughalterung zu bestimm-

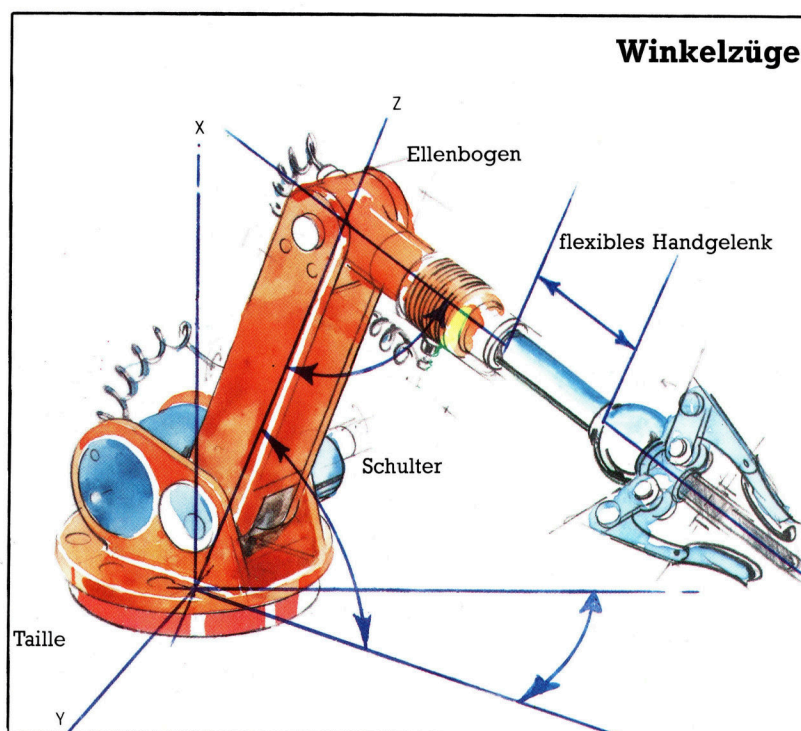
men. Bei dieser Methode bewegt sich der Ausgangspunkt mit der Halterung. Sie wird deshalb auch als „Point-to-point-Positionierung“ bezeichnet.

Ein moderner Industrieroboter läßt sich mit einer Genauigkeit von weniger als einem Millimeter Toleranz steuern. Selbst einfache Modelle, die schon für Preise zwischen 1000 und 2000 DM erhältlich sind und an jeden Heimcomputer über unterschiedliche Schnittstellen angeschlossen werden können, verfügen über eine Genauigkeit von nur wenigen Millimetern Toleranz.

## Einsatz von Schrittmotoren

Es gibt zwei allgemein akzeptierte Methoden zur Steuerung von Industrierobotern. Sind keine schweren Lasten zu transportieren, genügt der Einsatz von Schrittmotoren. Darunter versteht man Elektromotoren, die bei jedem Steuerimpuls eine vorgegebene Bewegung ausführen. Sie werden auch für die Positionierung des Schreib-/Lesekopfes einer Diskettenstation verwandt. Müssen in einer Fertigungsstraße allerdings schwere Gewichte transportiert werden, setzt man für die Bewegung der unterschiedlichen Armsegmente hauptsächlich hydraulisch gesteuerte Kolben ein. Da die Menge der hydraulischen Flüssigkeit, mit der die Kolben bewegt werden, leicht gemessen werden kann, erreicht man dadurch die für die industrielle Fertigung nötige Präzision.

Industrieroboter enthalten fast immer einen speziell für sie konstruierten Minicomputer (in den neueren Modellen findet man auch schon Microcomputer mit großer Kapazität), in dem ein eigenes dafür entwickeltes Maschinenpro-



## Winkelzüge

Schwierigste Aufgabe bei der Programmierung eines Roboters ist die Umsetzung geometrisch festgelegter Punkte in Bewegungsabläufe. Raumpunkte lassen sich über kartesische oder X,Y,Z-Koordinaten definieren. Ein Roboter benötigt jedoch Anweisungen für eine ganze Anzahl von Einzelementen: Bewegungswinkel für die „Ellbogen-“ und „Schulter“gelenke, die Drehung seiner „Taille“ und die Strecke, um die sich sein „Handgelenk“ ausdehnen soll. Auf einfachen Systemen muß der Programmierer alle vier Komponenten getrennt berücksichtigen. Bei hochentwickelten Robotern brauchen allerdings nur die kartesischen Koordinaten eingegeben werden.



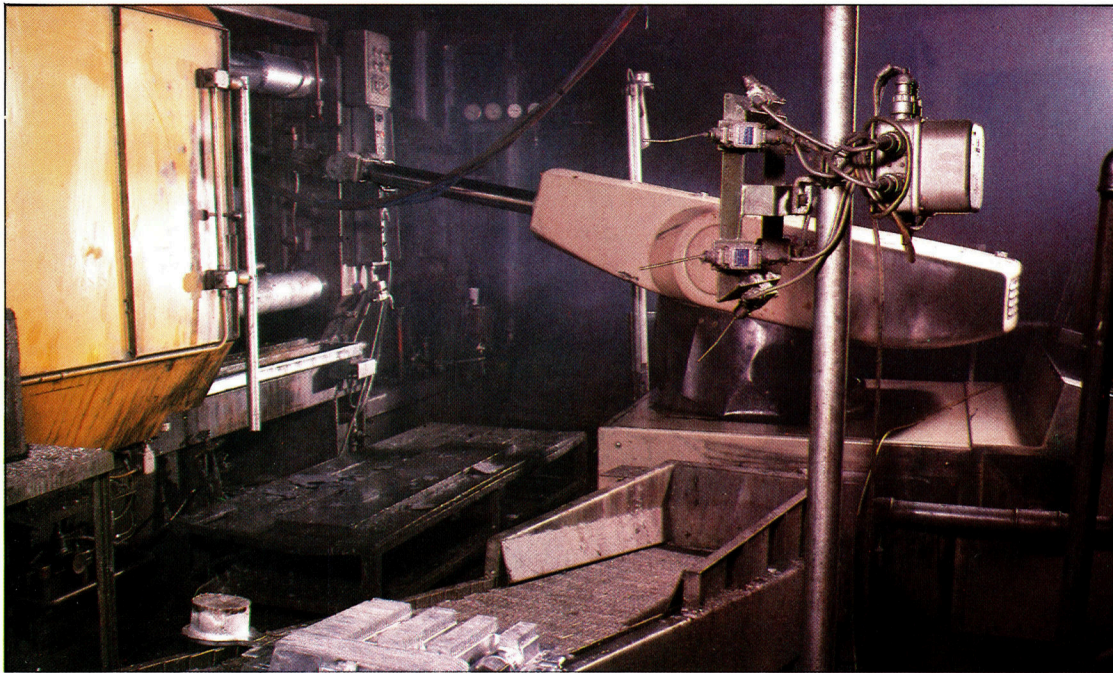


gramm die Bewegungen des Armes steuert. Die Programme werden über eine großformatige Zahlentastatur eingegeben, die über ein langes Kabel mit dem Computer verbunden ist. Durch die Länge des Kabels kann sich der Programmierer um den Roboter herum bewegen, ohne in dessen Wirkungsbereich zu gelangen. Neuere Versionen dieser beweglichen Konsolen verfügen schon über einen integrierten Präzisions-Joystick.

Eine weitere Programmiermethode, die als

holen. Maschinen dieser Art werden hauptsächlich für die Herstellung von Kraftfahrzeugen eingesetzt. Da hier schon seit langem Fertigungsstraßen existieren, in denen ein Werkstück sich zu einem bestimmten Zeitpunkt an immer dem gleichen Ort befindet, eignet sich dieser Industriezweig besonders gut für eine automatisierte Produktion.

Der Lichtsensor stellt eine Alternative zu dem Druckfühler dar. Wird ein Werkstück zwischen eine Lichtquelle und eine auf die Werk-



**Fertigungsarme, wie der im Bild gezeigte Roboter einer Gußmetallfabrik, werden in der Industrie mehr und mehr für gefährliche, schmutzige und Fließbandarbeiten eingesetzt. Das Polieren von gegossenen Formteilen ist dafür ein gutes Beispiel. Der kaum erstarrte Rohling ist für die Handbearbeitung viel zu heiß und müßte normalerweise erst eine Weile zur Abkühlung beiseite gelegt werden. Ein Roboter läßt sich jedoch von dem hohen Wärmegrad nicht beeinflussen und kann das Werkstück ohne Zeitverlust sofort weiterverarbeiten.**

„Follow Me“ bekannt ist, wird für Aufgaben eingesetzt, bei denen eine exakte Positionierung des Werkzeuges nicht notwendig ist, zum Beispiel beim Aufsprühen von Farbe mit einer Spritzpistole. Hierbei speichert der Programmierer die Abläufe in den Computer ein, indem er den Arm des Roboters direkt den gewünschten Weg entlang bewegt. Bei jedem Lauf des Programmes führt der Roboter dann diesen „gelernten“ Bewegungsablauf automatisch aus.

### Computer-Koordinierung

Bei all diesen Steuerungsmethoden wird nur die Position der Werkzeughalterung bestimmt. Der Programmierer braucht sich nicht um die Bewegungen der einzelnen Teile des Roboterarms zu kümmern, da die Programmsprache des in den Roboter eingebauten Computers diese Aufgabe übernimmt. Dabei werden auch alle Wege optimiert, um das Werkzeug in der schnellstmöglichen Zeit von einem Platz zum anderen zu bringen.

Bis jetzt haben wir uns mit Robotern beschäftigt, die „blind gehorchen“, das heißt, die einen Vorgang unabhängig von äußeren Einflüssen auf immer die gleiche Weise wieder-

zeughalterung montierte Fotozelle gebracht, verbleibt der Roboter solange in Wartestellung, bis das zu bearbeitende Material an der richtigen Stelle ist, dann erst wird das Werkzeug eingesetzt. Aber auch mit dieser Methode lassen sich nicht alle Fehler vermeiden. Für Aufgaben, bei denen völlige Zuverlässigkeit verlangt wird, setzt man deshalb Bilderkennungssysteme ein, die auf CCD (Charge-coupled Device = ladungsgekoppelte Schaltung)-Fernsehkameras aufgebaut sind. Diese Kameras projektieren das empfangene Bild auf einen speziellen Microchip, auf dessen Oberfläche sich hundert oder mehr Fotozellen befinden, die nicht nur hell und dunkel erkennen können, sondern auch verschiedene Zwischenstufen. Jeder einzelne Sensor benötigt in etwa ein Byte für die Speicherung eines bestimmten Grautones. Zu Beginn wird jedes Objekt mehrfach „fotografiert“ und die unterschiedlichen Ergebniswerte werden von einem Lernprogramm zu einem Durchschnittswert pro Bildpunkt verdichtet. Während des eigentlichen Arbeitsablaufes nimmt die CCD-Kamera das Bild des zu bearbeitenden Werkstückes auf und vergleicht es mit seinen gespeicherten Werten. Stimmen beide Bilder überein, wird mit der Bearbeitung begonnen.





# Das Programm-Genie

**Der ‚Lisa‘ von Apple hat auf dem Business-Computermarkt neue Maßstäbe gesetzt. Viele seiner Möglichkeiten werden jedoch schon bald auch auf Heimcomputern verfügbar sein.**



**Der Apple-Rechner Lisa ist speziell für EDV-Einsteiger im kommerziellen Bereich gedacht. Durch die Verwendung der Maus wird die Tastatur des Lisa weitaus weniger gebraucht als bei anderen Systemen.**

**D**er ‚Lisa‘ von Apple wurde ausschließlich für den kommerziellen Bereich konzipiert. Das Gerät kostet ohne den Drucker rund 18 000 Mark, und Sie werden sich wahrscheinlich fragen, warum wir überhaupt darüber berichten. Dies hat mehrere Gründe: Zum einen ist der Lisa seiner Zeit weit voraus, und zum anderen werden viele seiner Möglichkeiten nach und nach auch auf Heimcomputern verfügbar sein. Mit dem Macintosh hat Apple eine verkleinerte Version des Lisa auf den Markt gebracht, und schon jetzt bemüht sich die Konkurrenz, Computer herzustellen, deren Fähigkeiten mit denen des Lisas vergleichbar sind.

Interessanterweise ist es nicht die Hardware, die an dem Lisa das revolutionär Neue darstellt, sondern der Standard der mitgelieferten Software. Da heute weniger Zeit für die Konstruktion eines neuen Computers benötigt wird als für die Entwicklung eines neuen Arcadespiels oder kommerziellen Programms, ist der Trend im Augenblick, ausgefeilte Software auf den Markt zu bringen. Software wird zu-

nehmend zu dem wichtigsten – und auch teuersten – Element jedes Computersystems, und viele Heimcomputerbesitzer werden feststellen, daß sie im Verlauf eines Jahres möglicherweise mehr Geld für Programme (ob auf Cassetten, Cartridges oder Disketten) ausgegeben haben als für den Computer selbst.

Zunächst jedoch zur Hardware. Der Lisa wird standardmäßig mit einem RAM-Bereich von einem Megabyte Speicherkapazität geliefert (tausendmal die Kapazität des ZX81). Bei einem derart großen Speicher muß der Mikroprozessor relativ viel Zeit für dessen Verwaltung verwenden, d. h. Informationen im Speicher bewegen und deren Adressen festhalten.

Der 16-Bit-Prozessor 68000 von Motorola, mit dem der Lisa ausgerüstet ist, verarbeitet 16 Datenbits gleichzeitig, während die meisten Heimcomputer nur acht Bits parallel handhaben können.

Gemessen am Standard der Heimcomputer ist der 68000 ein außerordentlich schneller Prozessor mit einem hochentwickelten Befehlssatz. Als Speichermedium verfügt der Lisa über zwei Diskettenstationen und eine Festplatte in einem separaten Gehäuse, die nur vom Computer aus gesteuert werden kann. Die fünf Megabyte der Harddisk werden von dem Lisa dringend benötigt, um eine hohe Verarbeitungsgeschwindigkeit erreichen zu können. Außerdem wird die Festplatte gebraucht, um die vielen Programm-Module unterzubringen, die ständig in den RAM-Bereich nachgeladen werden müssen.

Interessant ist auch der integrierte einfarbige Monitor des Lisa mit seiner hohen Auflösung von  $720 \times 364$  Bildpunkten (Pixeln). Dadurch ist es möglich, eine Anzahl unterschiedlicher Schriftarten und die besondere Art von Grafik darzustellen, auf die später noch genauer eingegangen wird. Spezialprozessoren und -schaltungen wurden speziell dafür konstruiert, eine Vielzahl von Grafikelementen gleichzeitig abzubilden und schnell bewegen zu können.

Schließt man einen geeigneten Drucker an – z. B. einen Hochgeschwindigkeitsdrucker mit hochauflösender Punktmatrix –, kann der Bildschirminhalt unmittelbar auf das Papier gebracht werden. Verfügt der Drucker nicht über ein so hohes Auflösungsvermögen, versucht der Lisa automatisch, die höchste Druckqualität zu erreichen.

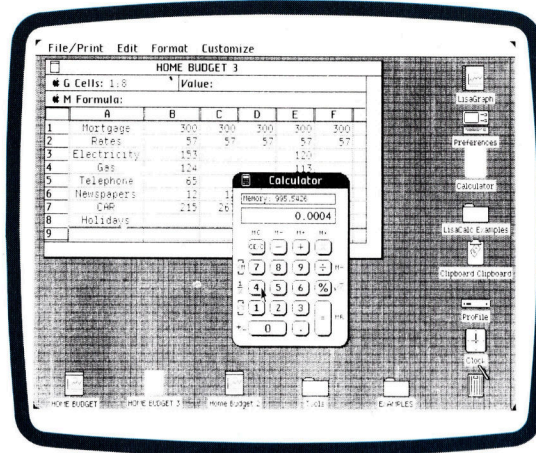




Die Tastatur des Lisa ist frei beweglich und gut ausgelegt. Da das Gerät jedoch über eine Maus verfügt, wird die Tastatur im Vergleich zu anderen Maschinen nur wenig benutzt. Eine Maus ist eine der vielen Möglichkeiten, einen Computer ohne die Tastatur nur über Anwahl von Bildschirmsymbolen zu steuern. Außer der Maus werden dafür auch Joysticks, Lichtgriffel, Touch-Screen und Spracherkennungssysteme verwandt. Unter einer Maus versteht man ein Gerät, nicht größer als eine Packung Zigaretten, das durch ein Kabel mit dem Computer verbunden ist und per Hand über die Oberfläche des Schreibtisches geführt wird. Die Bewegungen der Maus werden auf einen Cursor oder Zeiger auf den Bildschirm übertragen. Mit dem Zeiger kann man im Monitor auf Daten oder Befehle „fahren“ und über die SELECT-Taste der Maus die entsprechenden Daten abrufen oder Befehle auslösen. Die Tastatur wird nur gebraucht, wenn Daten oder Texte neu eingegeben werden müssen.

### Bildschirm-Schreibtisch

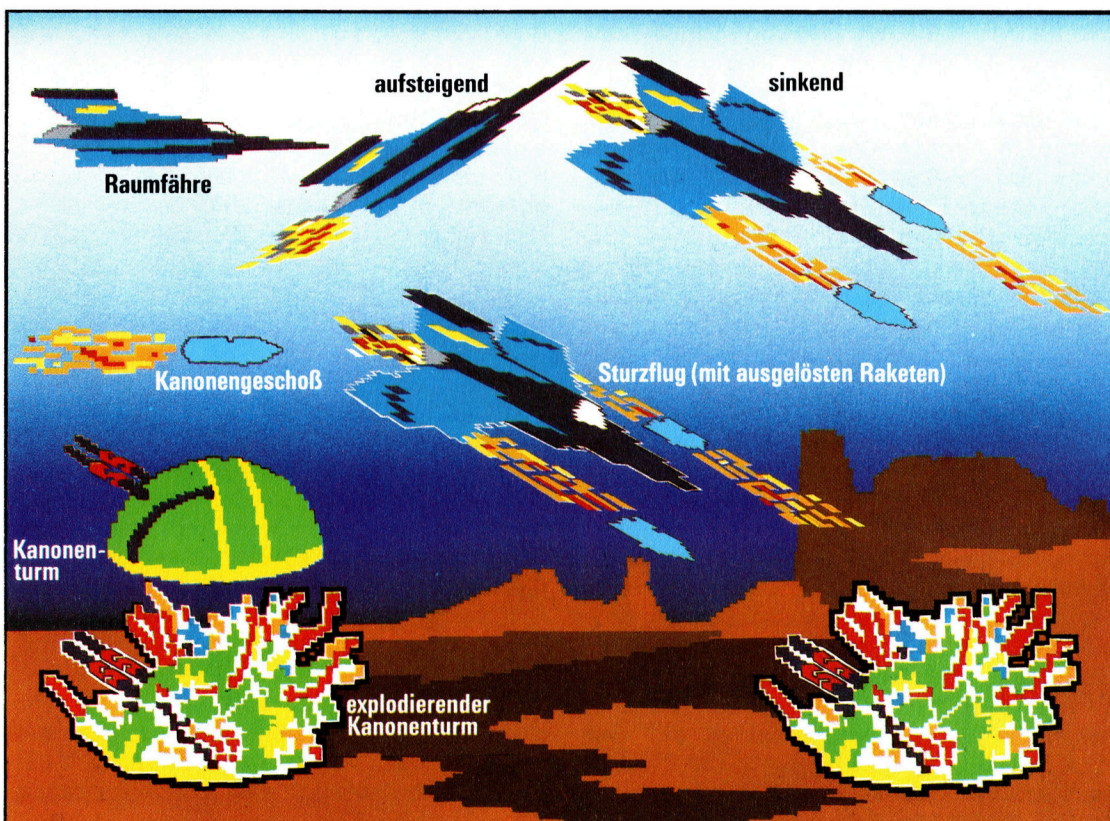
Doch nun zur Software: Schaltet man den Computer an, dann zeigt der Lisa auf dem Monitor einen „Schreibtisch“, auf dem sich verschiedene Gegenstände befinden. Jedes dieser Symbole stellt eine bestimmte Funktion dar. Nehmen wir als Beispiel die Darstellung der Uhr. Bewegt man den Cursor mit Hilfe der Maus auf das kleine Bild der Uhr und drückt dann die SELECT-Taste, erscheint eine weit- aus größere Uhr und das Tagesdatum auf dem



Jede Funktion des Lisa wird von einem Symbol dargestellt. Für den Abruf der Funktion wird der Cursor mit Hilfe der Maus auf das Symbol gesetzt und die SELECT-Taste der Maus gedrückt. Damit ist die Funktion „eröffnet“, und die entsprechenden Informationen erscheinen auf dem Bildschirm.

Monitor. Je nach Wunsch kann man sie wieder auf ihre ursprüngliche Größe „herunterfahren“. Nach der gleichen Methode läßt sich beispielsweise auch der Taschenrechner anwählen, auf dem einfache Rechenaufgaben gelöst werden können. Ist man mit der Anordnung der Gegenstände auf dem Schreibtisch nicht zufrieden, kann man alle Symbole durch Druck der SELECT-Taste bei gleichzeitiger Bewegung der Maus auf andere Plätze stellen. Eines der amüsantesten Beispiele für die Imitation gewohnter Arbeitsmethoden bietet der Lisa in der Handhabung des „Papierkorbs“. Benötigt man einen bestimmten „Gegenstand“ nicht mehr, „wirft“ man ihn mit der Maus einfach in das Papierkorbsymbol. Dank dieser Methode können wichtige Daten kaum noch zufällig gelöscht werden. Wenn der Lisa zwei-

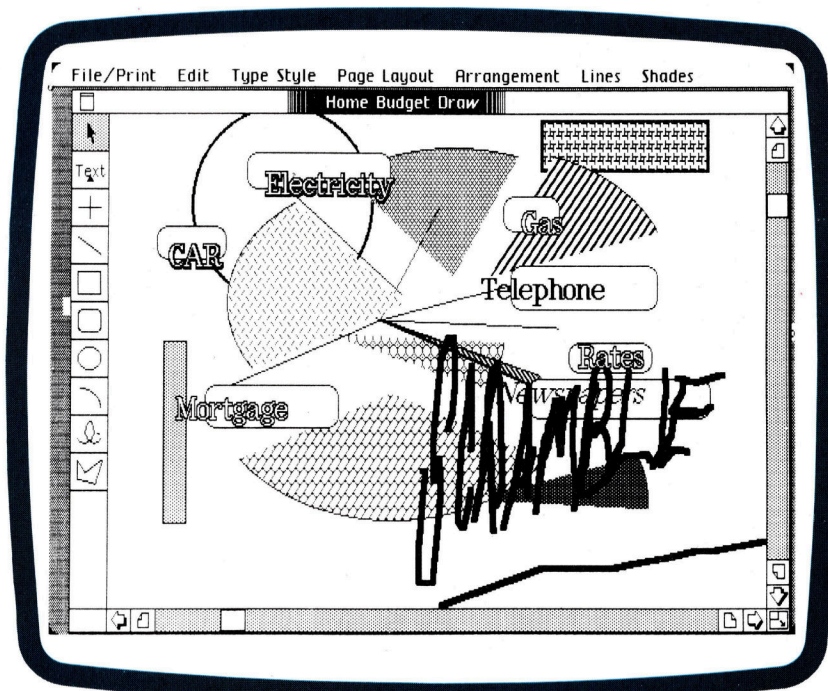
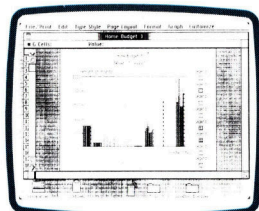
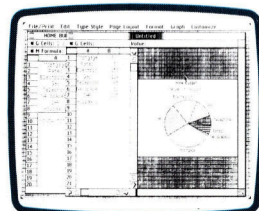
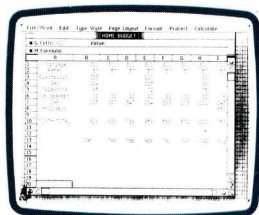
Für ein Verteidigungsspiel im Arcadestil mußten in der herkömmlichen Programmierweise eine Anzahl Bildschirminhalte einzeln definiert und ein Programm für die Steuerung des Spiels von Anfang an entwickelt werden. Bei objektorientierter Programmierung brauchen Sie sich nur auf die einzelnen Elemente des Spiels zu konzentrieren. Dabei wird zunächst eingegeben, wie eine Raumfähre sich von links nach rechts über den Bildschirm bewegt. Drückt man den Joystick nach oben oder unten, bewegt sich das Objekt entsprechend auf oder ab; wird der Feuerknopf betätigt, löst sich eine Rakete. Als nächstes muß die Rakete definiert werden. Auch hier wird eine bestimmte Bewegungsrichtung programmiert. Trifft die Rakete ein anderes Objekt, verschwindet sie von dem Bildschirm. Der feststehende Kanonenturm wird als einfacher Umriss definiert, der bei dem Auftreffen einer Rakete in das Bild einer Explosion übergeht.







Eine der erstaunlichsten Fähigkeiten des Lisa ist seine Möglichkeit, Daten mit der COPY-Funktion von einer Anwendung in eine andere zu übertragen. Dabei werden die Informationen mit dem Schnellheftersymbol zwischengespeichert und dann in ein anderes Fenster übernommen. Nachdem man z. B. einige Zahlen mit LisaCalc (dem Kalkulationssystem) berechnet hat, kopiert man die Ergebnisse auf LisaGraph, das automatisch ein Torten- oder Balkendiagramm davon anfertigt. Das fertige Diagramm kann auf LisaDraw übertragen, grafisch verfeinert und z. B. mit Zusatzbezeichnungen, Pfeilen, weiteren Diagrammen usw. versehen werden. Das Endergebnis ist ein druckreifes Bild.



schenzeitlich nicht abgeschaltet wurde, ist es sogar möglich, den Inhalt des Papierkorbs nochmals unter die Lupe zu nehmen und erneut zu verwenden.

Der größte Teil der Arbeit kann mit einem der sechs Standardprogramme des Lisa erledigt werden: Mit LisaWrite werden Texte bearbeitet, Kalkulationen können mit LisaCalc ausgeführt werden, LisaGraph wird für Diagrammdarstellungen verwandt, LisaList bietet eine Datenbank mit Listenverwaltung, LisaProject eine Planungshilfe und LisaDraw ein hochentwickeltes Hilfsmittel für die Darstellung aller Arten von Grafik. Die Symbole für diese Anwendungsprogramme sind einfache weiße Papierblöcke. Für den Ablauf des Kalkulationssystems wird der Cursor einfach auf die Darstellung eines in Zeilen und Spalten unterteilten Schreibblocks gesetzt. Drückt man die SELECT-Taste, wird dieses Stück Papier vom Block „abgerissen“, an eine andere Stelle des Schreibtisches gelegt und kann dann mit einem Titel wie z. B. „Verkaufsplanung“ versehen werden.

Der Anwender kann mit diesem System sogar mehrere unterschiedliche Programmelemente gleichzeitig auf dem Bildschirm darstellen. Für den gleichen Vorgang müßte bei einem normalen Computersystem zuerst das Kalkulationsprogramm geladen und dann die gewünschte Datei angegeben werden. Auf dem Lisa sind Programme und Daten jedoch miteinander verbunden. Eine weitere interessante Eigenschaft des Lisa ist seine Fähigkeit, „Fenster“ zu bilden. Dabei wird ein gewähltes Programm als großes Papierblatt auf dem Schreibtisch dargestellt. Die Größe des Papiers kann wiederum mit der Maus bestimmt werden. Wurde mehr als eine Anwendung ge-

wählt, überlagern sich die Papiere, wobei das augenblicklich bearbeitete oben auf dem Stapel liegt – genau wie auf einem echten Schreibtisch. Auf dem Lisa ist ein Blatt Papier jedoch ein Fenster, das einen Ausschnitt der Anwendung anzeigt und über die gesamte Oberfläche des Dokuments bewegt werden kann.

### Ausgezeichnete Flexibilität

Eine weitere Stärke des Lisa ist seine Fähigkeit, Informationen von einer Anwendung in eine andere zu übernehmen. Angenommen, Sie analysieren gerade Ihre monatlichen Verkaufszahlen mit LisaCalc. Mit der COPY-Funktion, die Sie aus einem speziellen Menü am oberen Rand des Bildschirms ausgewählt haben, ziehen Sie sich eine Zwischenkopie Ihrer Berechnungsergebnisse und speichern diese unter dem Schnellhefter-Symbol ab. Dann wählen Sie ein Blatt des LisaGraph-Papiers aus und tragen mit der PASTE-Funktion die Berechnungsergebnisse in dessen INPUT DATA-Bereich ein. Falls gewünscht, produziert LisaGraph jetzt ein Tortendiagramm (oder Balken- oder Liniendiagramm), komplett beschriftet und mit schraffierten Flächen versehen. Über die Funktionen PASTE und COPY kann dieses Bild wiederum auf ein Blatt des LisaDraw-Papiers kopiert werden. Das Endergebnis ließe sich drucken, auf eine Folie für die Overhead-Projektion kopieren oder als Reinzeichnung für einen Zeitschriftenartikel verwenden.

Die Möglichkeiten objektorientierter Programmierung werden bald auch auf preiswerten Maschinen Verwendung finden, da Verarbeitungsgeschwindigkeit und Speicherkapazitäten ständig zunehmen.





# Epson HX-20

**Dieser Computer findet in einem Aktenkoffer Platz, und man kann ihn überallhin mitnehmen – das ideale Büro auf Reisen.**

**D**er Epson HX-20 ist einer der ersten Computer, der die Bezeichnung 'Portable' zu Recht trägt. Diese Maschine, die bequem in jeden Aktenkoffer paßt, kann erheblich mehr als die leistungsfähigsten Taschenrechner, denn sie ist voll in BASIC programmierbar. Und obwohl alles, was zu einem Personalcomputer gehört, sich unter dem HX-20-Gehäuse befindet, wiegt er keine zwei Kilogramm. Bis zum Erscheinen des HX-20 wurden wesentlich gewichtigere Geräte mit dem Prädikat 'Portable' bedacht, die von Anwendern allerdings bald das Prädikat „untragbar“ erhielten.

Die Käufer des HX-20 rekrutieren sich aus verschiedenen Microcomputer-Anwendern. Vertreten sind sowohl Hobbyisten, als auch Geschäftsleute und Ingenieure.

Der HX-20 eignet sich für das Erlernen von BASIC – das Softwareangebot ist jedoch relativ gering. Der Käufer hat häufig den Wunsch, ein Programm für eine ganz spezielle Anwendung zu schreiben. Sei es ein Beratungsprogramm für Versicherungsvertreter, ein Navigationsprogramm für Segler oder ein elektronischer Notizblock für Journalisten: Allen kann der HX-20 gute Dienste leisten.

Die Flüssigkristallanzeige (LCD) kann bis zu

vier Zeilen zu je 20 Zeichen darstellen. Einfache grafische Darstellungen kann sie mit einer Auflösung von bis zu  $120 \times 32$  Punkten abbilden. Mit Hilfe der Steuerungstasten läßt sich der Cursor in alle vier Richtungen bewegen. Der eingebaute Drucker benutzt normales, fünf Zentimeter breites Rollenpapier, auf dem er bis zu 24 Zeichen umfassende Textzeilen drucken kann. Ein Extra, das für die meisten Anwender unentbehrlich ist, ist das Microcassettengerät. An seiner Stelle könnte auch ein Software-Modul eingesteckt werden. Einem normalen Cassettengerät ist das viel kleinere Microcassettengerät weit überlegen, denn der Computer findet über den schnellen Vorlauf gesuchte Programmteile oder Daten automatisch.

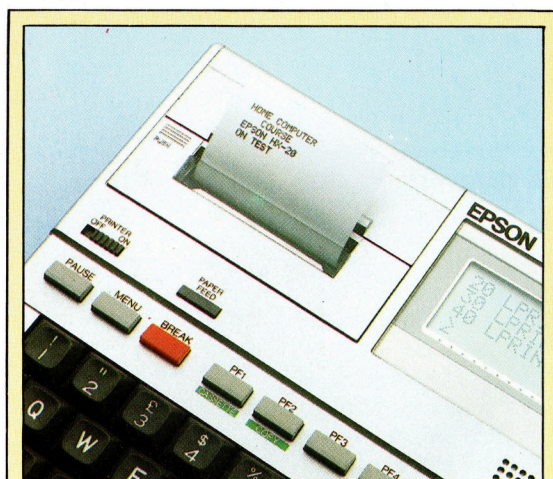
Den weiten Anwendungsbereich des HX-20 unterstützen die vielen integrierten Schnittstellen. Sogar einen Strich-Code-Leser kann man anschließen. Der 16 KByte umfassende Arbeitsspeicher läßt sich per Steckmodul auf 32 KByte erweitern.

Per Telefon kann der HX-20 Verbindung mit einem anderen HX-20 oder einer Großrechenanlage aufnehmen, um auf deren Speicher zugreifen zu können.

Eine vollwertige Schreibmaschinen-Tastatur bestimmt die Abmessungen des HX-20. Obwohl sich der Anschlag von dem einer Schreibmaschine unterscheidet, ist die Tastatur auch bei professionellen Schreibern beliebt. Zusätzlich zu den Zifferntasten in der obersten Reihe können die Tasten U, I, O, J, K, L, M in Verbindung mit den Zifferntasten 7, 8, 9 durch Drücken der NUM-Taste in einen Ziffernblock umfunktioniert werden. Große Mengen numerischer Daten lassen sich so viel bequemer eingeben. Oben rechts befinden sich die Editor-Tasten. Die Cursorsteuerungstasten liegen zwei Reihen tiefer. Mit der SCRN-Taste und den entsprechenden Pfeiltasten läßt sich das Bildschirmfenster rauf und runter bewegen. Die mit PF1 und PF5 gekennzeichneten Funktionstasten sind frei programmierbar. Zur besseren Unterscheidung haben sie eine andere Form. Zwei dieser Tasten sind zusätzlich für die Bedienung des Microcassettengerätes vorgesehen. Mit ihnen läßt sich auch der Bildschirminhalt auf Microcassette kopieren oder auf den Drucker übertragen.







### Der eingebaute Drucker

Als renommierter Druckerhersteller konnte Epson es sich nicht leisten, den HX-20 ohne eigenen Ausdruck zu konzipieren. Wen wundert es da, daß Epson einen kleinen Matrixdrucker wählte, der nicht nur Texte, sondern auch gut erkennbare Grafiken zu Papier bringt.



### Versorgungsanschluß

Hier können Sie einen Netztrafo anschließen, um die Batterien des HX-20 aufzuladen.

### Strich-Code-Leser-Anschluß

Hier läßt sich ein Lesestift für Strich-Codes anschließen.

### Reset-Taste

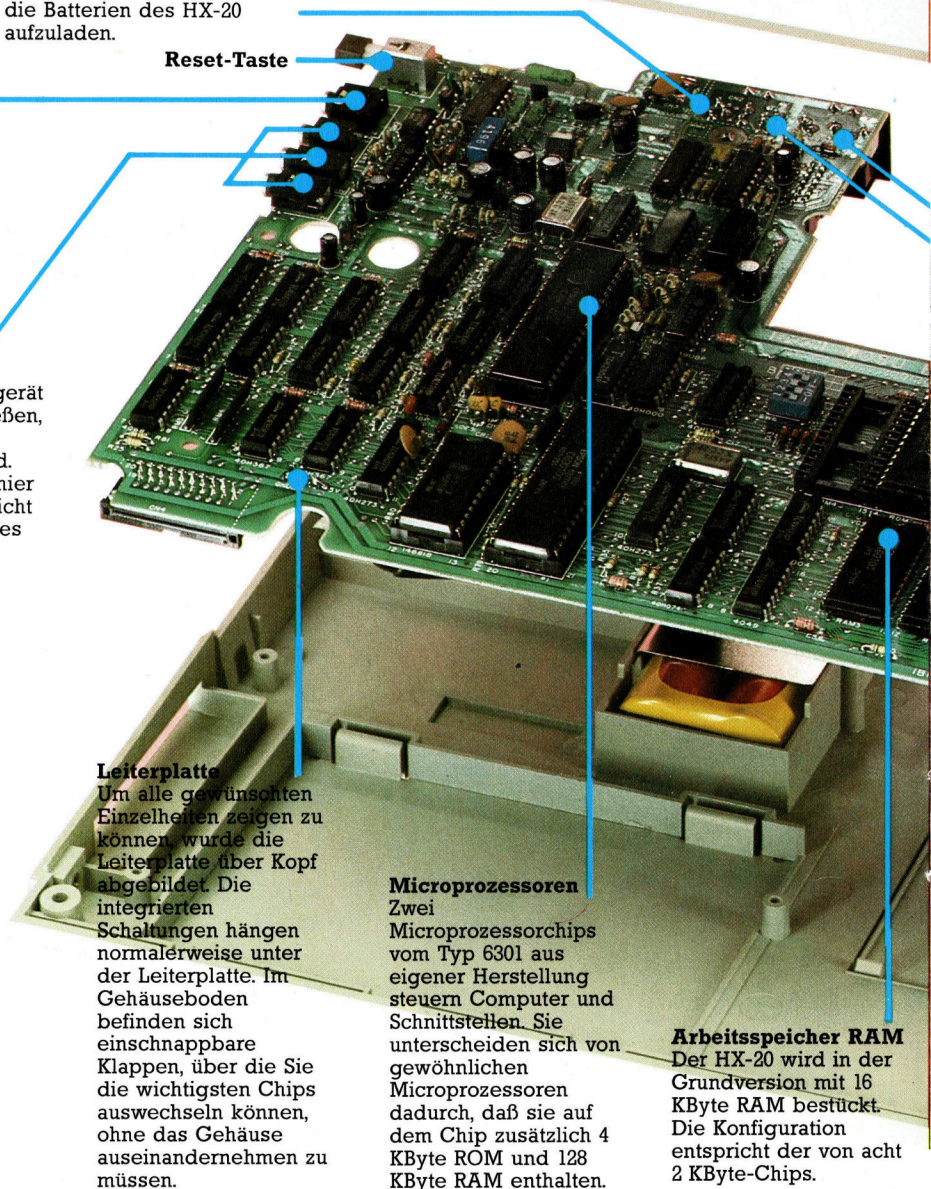
### Anschluß für Cassettengerät

Ein normales Cassettengerät können Sie hier anschließen, wobei der Motor vom Computer gesteuert wird. Allerdings erreicht das hier angeschlossene Gerät nicht die Leistungsfähigkeit des integrierten Microcassettengerätes.



### Der Massenspeicher

Das Microcassettengerät wird in die dafür vorgesehene Aussparung gesteckt. Microcassetten sind als Datenträger in Diktiergeräten weit verbreitet. Das Gerät selbst hat aber gegenüber gewöhnlichen Cassettengeräten den Vorteil, daß der schnelle Vorlauf vom Computer selbst gesteuert wird. Dies reduziert die Zeit für das Auffinden gewünschter Daten.



**Leiterplatte**  
Um alle gewünschten Einzelheiten zeigen zu können, wurde die Leiterplatte über Kopf abgebildet. Die integrierten Schaltungen hängen normalerweise unter der Leiterplatte. Im Gehäuseboden befinden sich einschnappbare Klappen, über die Sie die wichtigsten Chips auswechseln können, ohne das Gehäuse auseinandernehmen zu müssen.

### Microprozessoren

Zwei Microprozessorschips vom Typ 6301 aus eigener Herstellung steuern Computer und Schnittstellen. Sie unterscheiden sich von gewöhnlichen Microprozessoren dadurch, daß sie auf dem Chip zusätzlich 4 KByte ROM und 128 KByte RAM enthalten.

### Arbeitsspeicher RAM

Der HX-20 wird in der Grundversion mit 16 KByte RAM bestückt. Die Konfiguration entspricht der von acht 2 KByte-Chips.





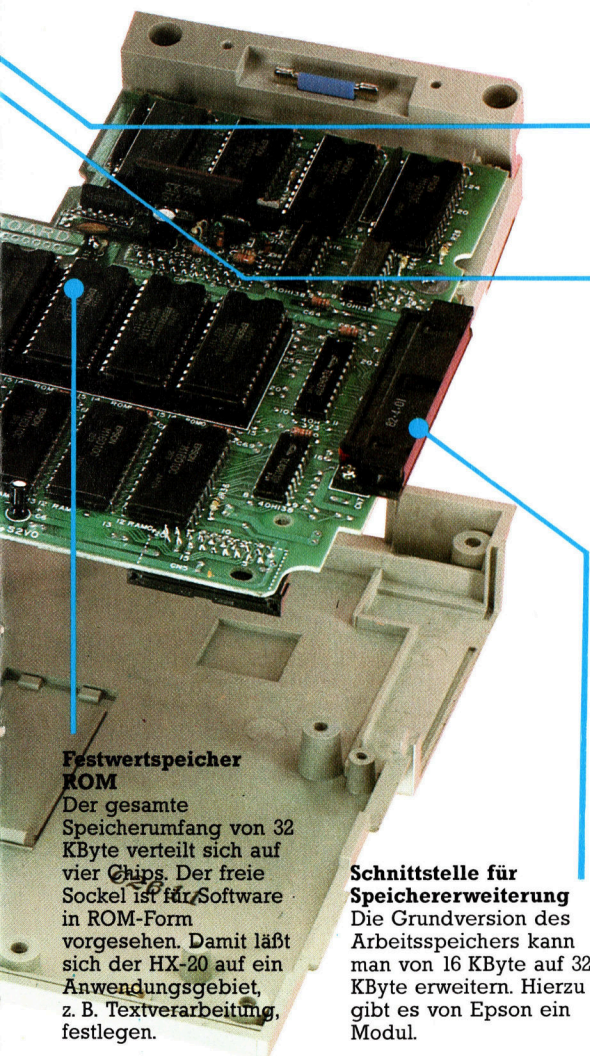
### Modul-Halterung

Mit diesem Knopf können Sie die Halterung des Microcassettengerätes lösen und danach das komplette Modul entfernen. Epson plant ein Halbleiter-Modul, das anstelle des Microcassettengerätes eingesteckt werden kann und den HX-20 auf eine bestimmte Anwendung festlegt.

### Blickwinkel-Anpassung

Gewöhnliche Flüssigkristallanzeigen (LCD) müssen, um gut lesbar zu sein, aus dem richtigen Winkel betrachtet werden. Nicht so beim HX-20, denn mit diesem Drehknopf können Sie die Lesbarkeit der HX-20-Anzeige Ihrem momentanen Blickwinkel anpassen.

Ein-Ausschalter



### Serielle Schnittstelle

Diese Schnittstelle ist für zusätzliche Peripheriegeräte vorgesehen.

### RS 232-Schnittstelle

Über diese Standard-Schnittstelle können Sie einen Drucker oder ein Modem anschließen. Über dieses Telefonmodem kann Ihr HX-20 mit anderen Computern Verbindung aufnehmen.

### Festwertspeicher ROM

Der gesamte Speicherumfang von 32 KByte verteilt sich auf vier Chips. Der freie Sockel ist für Software in ROM-Form vorgesehen. Damit läßt sich der HX-20 auf ein Anwendungsgebiet, z. B. Textverarbeitung, festlegen.

### Schnittstelle für Speichererweiterung

Die Grundversion des Arbeitsspeichers kann man von 16 KByte auf 32 KByte erweitern. Hierzu gibt es von Epson ein Modul.



### Der neue Dreh

Die Kathodenstrahlröhre, das Herzstück jedes Monitors, benötigt in Microcomputern den größten Raum. Um dem Anspruch zu genügen, ein portables Gerät zu sein, mußte Epson bei HX-20 neue Wege gehen und anstelle eines konventionellen Monitors eine hochstellbare LCD-Anzeige einsetzen.

## EPSON HX-20

### PREIS

1798 Mark mit Microcassette

### ABMESSUNGEN

290 × 210 × 45 mm

### GEWICHT

1,7 kg

### CPU

6301 (entwickelt von Hitachi)

### TAKTFREQUENZ

2,45 MHz

### MASCHINENSPEICHER

Erweiterbar auf 32 KByte mit Steckmodul, 32 KByte Festwertspeicher im ROM erweiterbar auf 64 KByte

### BILDSCHIRM-DARSTELLUNG

LCD-Anzeige, 4 Zeilen zu je 20 Zeichen, mit einstellbarer Blickwinkel-Anpassung. Kann als Fenster über einen Bereich von 255 × 255 Zeichen bewegt werden. Maximale grafische Auflösung beträgt 120 × 32 Pixel, jedes Pixel ist einzeln adressierbar.

### SNITTSTELLEN

Für Speichererweiterung, externes Cassettengerät, externen Drucker über RS 232, Strichcode-Lesestift, serielle Schnittstelle für Peripheriegeräte, z. B. Diskettengerät.

### PROGRAMMIER-SPRACHE

BASIC

### WEITERE SPRACHEN

FORTH-Interpreter

### ZUBEHÖR

Tragekoffer aus Kunststoff, Netzgerät, Bedienungsanleitung

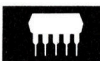
### TASTATUR

Schreibmaschinen-Tastatur mit 60 Tasten.

### DOKUMENTATION

Sie erhalten ein Betriebsanleitung und ein BASIC-Handbuch. Epson hat sich bei der Auswahl des BASIC-Dialektes für den Industriestandard von Microsoft entschieden. Das Betriebsanleitung enthält auch spezielle Angaben, wie z. B. eine komplette Liste der E/A-Belegung für die RS 232- und die serielle Schnittstelle.





# Magischer Kristall

**Flüssigkristallanzeigen, bekannt als „Zifferblätter“ der Digital-Uhren, werden heute auch in Computer eingebaut.**

**F**lüssigkristallanzeigen (englisch: Liquid Crystal Display – LCD) tauchten erstmals Ende 1973 als Anzeigefelder von Taschenrechnern auf. Kurze Zeit danach wurden LCDs auch bei der Produktion von Digitaluhren verwendet. Ein weiterer Schritt war, diese Anzeigetechnik bei Computern einzusetzen. Zunächst waren es nur kleine tragbare Geräte, wie der HX-20 von Epson, der Tandy TRS80 Modell 100 Portable Computer und der Sharp PC 5000, die mit einer Flüssigkristallanzeige ausgerüstet wurden. Inzwischen aber ist die Fläche der LCDs auf die Größe des Standardbildschirms angewachsen, und es gibt eine Reihe von Computermodellen, die anstelle eines Monitors Flüssigkristallanzeigen verwenden.

Wie funktionieren LCDs? Jede Materie geht bei bestimmtem Druck und Temperaturveränderungen in einen anderen Aggregatzustand über – von gasförmig zu flüssig zu fest (kristallin) oder umgekehrt. Eine regelmäßige Ausrichtung der Moleküle eines Stoffes läßt sich aber nur im festen Zustand finden. Von dieser allgemeingültigen Regel gibt es jedoch Ausnahmen: Flüssigkristalle – ihre Zusammensetzung ist ein streng gehütetes Geheimnis.

## Neue Darstellungsform

Bei der Suche nach alternativen Darstellungsmöglichkeiten von Informationen wurde entdeckt, daß Flüssigkristallmoleküle ihre Ausrichtung verändern, wenn eine Spannung angelegt wird, und daß sich diese Veränderung auf einen Teilbereich des Flüssigkristalls begrenzen läßt. Auf dieser Grundlage wurden die ersten Flüssigkristallanzeigen entwickelt. Im ersten Schritt wurden hierbei Elektroden in Form des darzustellenden Zeichens auf die Innenseiten von zwei Glasplatten gebracht. Zwischen den Glasplatten wurde ein dünner Film aus „Flüssigkristall“ eingeschlossen und Spannung wurde angelegt. Unter normalem Licht schien sich nichts zu verändern, wurden aber polarisierende Filter (siehe Bild rechts oben) auf die Vorder- und Rückseite der beiden Glasplatten gelegt und die Anzeige vor eine Spiegelfolie gesetzt, zeigte sich ein scharf umrissenes Zeichen vor neutralem Hintergrund.

### Flüssigkristall

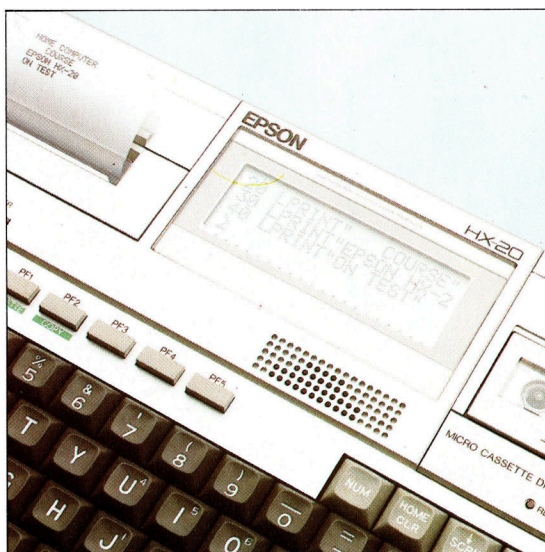
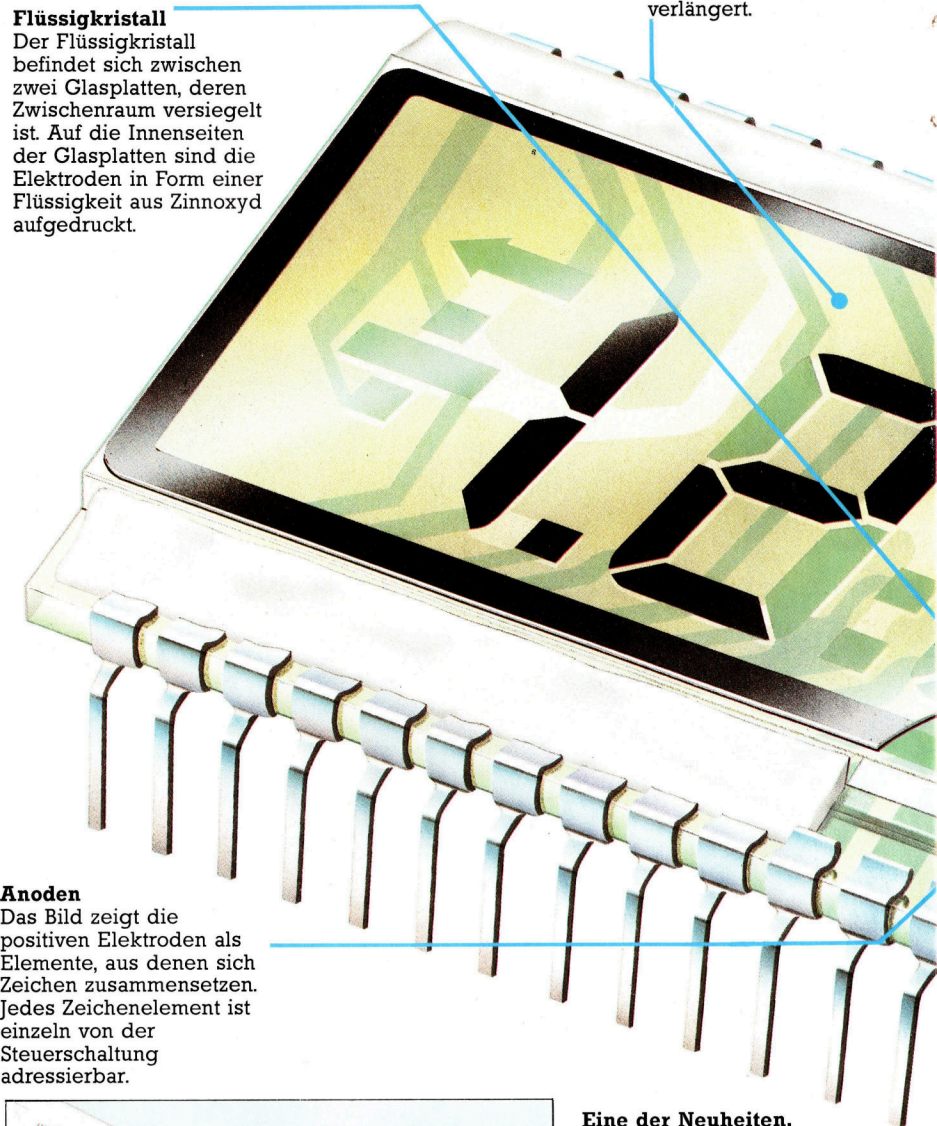
Der Flüssigkristall befindet sich zwischen zwei Glasplatten, deren Zwischenraum versiegelt ist. Auf die Innenseiten der Glasplatten sind die Elektroden in Form einer Flüssigkeit aus Zinnoxid aufgedruckt.

### Filter für vertikale Polarisation

Der vordere Polarisationsfilter ist nur für Licht mit vertikaler Schwingung durchlässig. Darin integriert ist ein Ultraviolettfilter, der die Lebensdauer des Flüssigkristalls verlängert.

### Anoden

Das Bild zeigt die positiven Elektroden als Elemente, aus denen sich Zeichen zusammensetzen. Jedes Zeichenelement ist einzeln von der Steuerschaltung adressierbar.



Eine der Neuheiten, die Epson in seinen tragbaren HX-20 Microcomputer eingebaut hat, ist die Regulierbarkeit des Darstellungswinkels der Flüssigkristallanzeige. Das LCD ist so groß, daß 4 Zeilen à 20 Zeichen gleichzeitig angezeigt werden können.





### Verbindungsstege

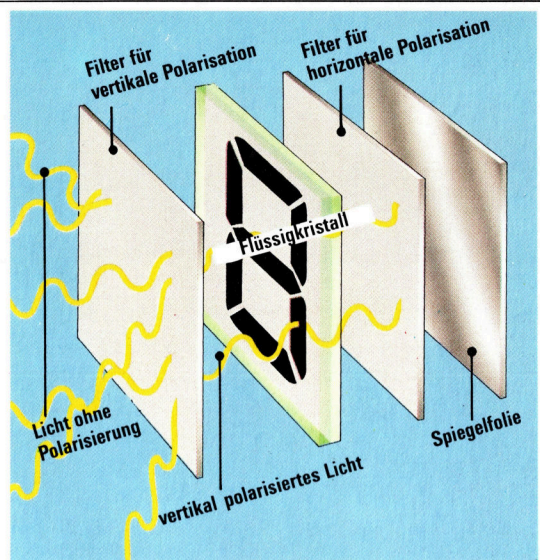
Jede Elektrode ist durch fast unsichtbare Stege aus leitender „Tinte“ mit der Steuerschaltung verbunden. Die Stege verlaufen auf der Innenfläche einer der beiden Glasplatten.

### Filter für horizontale Polarisation

Die Polarisationsachse des hinteren Filters ist zu dem vorderen Filter um 90 Grad versetzt. Er ist nur für Licht mit horizontaler Schwingung durchlässig.

### Polarisierung

Der vor dem Flüssigkristall angebrachte Filter ist nur für Licht durchlässig, dessen elektromagnetische Schwingung vertikal ausgerichtet ist. Der Flüssigkristall dreht das polarisierte Licht um 90 Grad und ermöglicht ihm damit, den hinteren Polarisationsfilter zu durchdringen. Die Spiegelfolie reflektiert das Licht. Wird an einen Teil der LCD-Anzeige Spannung gelegt, rotiert der Flüssigkristall das polarisierte Licht nicht mehr und scheint schwarz zu sein.



Die meisten Flüssigkristallanzeigen setzen sich aus stabförmigen Elementen zusammen. Inzwischen werden aber auch LCDs mit Punktmatrix hergestellt. Da sich der Kontrast der Punkte nicht verändern läßt, mußten die einzelnen Matrixelemente auf die Größe eines Kathodenstrahlpixels verkleinert werden, damit eine akzeptable Auflösung des Bildes überhaupt möglich war. Außerdem wurden mehrere LCDs hintereinander montiert und gleichzeitig betrieben, um bei begrenztem Anzeigepplatz komplexe Daten darzustellen.

### Minimaler Stromverbrauch

Circa 70 Millisekunden benötigt eine qualitativ hochwertige Flüssigkristallanzeige bei normaler Arbeitstemperatur (20 Grad Celsius), um vom neutralen Zustand aus eine schwarze Fläche aufzubauen. Weitere 80 Millisekunden sind nötig, um die Fläche wieder in den neutralen Zustand zu bringen. Diese insgesamt 150 Millisekunden liegen wesentlich höher als die Antwortzeiten einer Kathodenstrahlröhre (0,00025 Millisekunden). Dennoch haben LCD-Anzeigen eine Reihe wesentlicher Vorteile: Der Platzbedarf ist gering, und der Stromverbrauch beträgt nur 10 Milli watt pro Quadratzen timer Darstellungsfläche.

Ein interessanter Effekt wird erzielt, wenn man die angelegte Spannung verändert: Bei unterschiedlicher Voltzahl verändert sich auch der Ausrichtungswinkel der Moleküle. Diese Eigenschaft hat die Firma Epson in die Konstruktion ihres HX-20 eingebracht, bei dem über einen Regler der Blickwinkel der LCD-Anzeige stufenlos regelbar ist.

Ein weiterer Vorteil der Flüssigkristallanzeigen wird bei Sonneneinstrahlung deutlich: Der Kontrast von Kathodenstrahlröhren läßt bei starkem Sonnenlicht schnell nach, LCD-Anzeigen jedoch vergrößern bei einer Zunahme externen Lichtes den Kontrast.

### Spiegelfolie

Die meisten LCD-Anzeigen arbeiten mit Licht, das von einer Spiegelfolie reflektiert wird.

### Kathoden

Die negativen Elektroden sind in einem Feld zusammengefaßt, das alle Zeichenelemente miteinander verbindet.

Dieser Effekt entsteht folgendermaßen: Licht, das durch den vorderen Filter fällt, wird vertikal polarisiert. Das Flüssigkristallfeld rotiert es um 90 Grad und ermöglicht dem Licht damit, den hinteren (horizontalen) Filter zu passieren. Wird jedoch Elektrizität durch das Flüssigkristallfeld geleitet, findet keine Rotation statt, und das vertikal polarisierte Licht kann den hinteren Polarisationsfilter nicht mehr ungehindert durchdringen. Als Ergebnis erscheint im Spannungsbereich eine schwarze Fläche. Fast die gleiche Methode wird noch heute für Flüssigkristallanzeigen verwandt, nur wurden die Elektroden durch farblose, durchsichtige Tinte ersetzt, die auf die Glasoberfläche aufgedruckt wird und nach dem Trocknen fast unsichtbar ist.





# Der tönende VC

## Die vielfältigen Möglichkeiten der Sounderzeugung auf dem Commodore VC 20.

**D**ie gesamte Klangerzeugung des Commodore VC 20 wird über POKE-Befehle gesteuert, die bestimmte Speicherstellen im RAM-Bereich direkt ansprechen.

Der VC 20 ist mit drei Oszillatoren für Rechteckschwingungen und einem Generator für die Erzeugung von Rauschen ausgestattet. Jeder Oszillator deckt einen Tonbereich von etwa drei Oktaven ab, der jeweils die im Kasten verzeichneten Frequenzen umfaßt:

Osz. 1	Osz. 2	Osz. 3	Freq. Bereich (Hz)	Oktave
●			(65,41-123,47)	1
●	●		(130,81-246,94)	2
●	●	●	(261,63-493,88)	3
	●	●	(523,25-987,77)	4
		●	(1046,5-1975,53)	5

Dieser Aufbau ermöglicht die Verwendung von insgesamt fünf Oktaven, wobei für jede Oktave mindestens ein Oszillator zur Verfügung steht. Oktave 3 kann dabei von allen drei Oszillatoren angesprochen werden. Sie beginnt bei dem mittleren C und erhält damit auch den Kammerton A mit 440 Hz.

Die Oszillatoren werden über fünf Speicherstellen gesteuert:

Speicherstelle	Oszillator
POKE 36874,X	1
POKE 36875,X	2
POKE 36876,X	3
POKE 36877,X	noise

In jedem Fall ist X eine Ganzzahl zwischen 135 und 241 (0 schaltet den Oszillator ab). Im Handbuch des VC 20 befindet sich eine Tabelle, in der neben jeder Zahl die entsprechenden Tonwerte aufgeführt sind. Bevor nun die gewählte Frequenz gehört werden kann, muß erst die Lautstärke über folgenden Befehl festgelegt werden:

POKE 36878,V

V kann dabei die Zahlen 0 (aus) bis 15 (laut) annehmen und steuert die Lautstärke aller drei Oszillatoren und des Rauschgenerators. Ein Beispiel:

POKE 36874,219:POKE 36875,219:POKE  
36876,219:POKE 36878,7

Mit diesen Befehlen wird ein A mit 440 Hz auf dem Oszillator 1 angesprochen, weiterhin das A eine Oktave höher auf dem Oszillator 2 und noch eine Oktave höher auf dem Oszillator 3 — alle mit der mittleren Lautstärke 7. Für das Abschalten des Klanges muß jede der angesprochenen Speicherstellen über POKE wieder auf 0 gesetzt werden.

## Noten und Pausen

Wird die Dauer eines Tones nicht definiert, d. h. die entsprechenden Pausen zwischen den Tönen nicht gesetzt, verwischen sich bei einer Tonfolge die Unterschiede zwischen den einzelnen Tönen. Es gibt zwei Methoden, mit denen „Warteperioden“ zwischen die POKE-Befehle gesetzt werden können. Die erste Methode verwendet FOR...NEXT-Schleifen, bei denen die Dauer der Pause durch die Anzahl langer Warteschleifen definiert wird:

```
10 POKE 36878,7
20 POKE 36876,203
30 FOR P=1 TO 200
40 NEXT P
50 POKE 36878,0
60 POKE 36876,0
```

Diese Befehlsfolge spielt den Ton Dis für die Länge von 200 FOR...NEXT-Schleifen. Bei dieser Methode muß allerdings die Schleifenlänge für jeden Ton erst experimentell bestimmt werden. Eine elegantere Methode, Tondauer und Pausen festzusetzen, bietet die interne Uhr des VC 20. Die kleinste Zeiteinheit dieser Uhr ist der sechzigste Teil einer Sekunde (ein Jiffy). Sie kann über die Variable TI angesprochen werden. Mit dieser Methode kann die Zeit, die ein Programm „warten“ soll, exakt bestimmt werden. Beispiel:

```
10 POKE 36878,7
20 POKE 36876,203:D=TI
30 IF TI-D<15 THEN 30
40 POKE 36878,0
50 POKE 36876,0
```

Dieses Programm spielt den gleichen Ton wie unser letztes Beispiel, diesmal aber für die Dauer von 15 Jiffys (eine Viertelsekunde). Bei Einschalten des Tones wird der Variablen D dabei der Wert von TI zugewiesen. Zeile 30 zählt 15 Jiffys ab, bevor das Programm die Zeile 40 ausführt. Nach dem gleichen Prinzip können Tonfolgen konstruiert werden, bei denen vor jedem neuen Ton eine Pause eingelegt wird.





# Der zeichnende Dragon

## Dragon 32 – gute Grafikfähigkeiten durch ‚Microsoft Extended Colour BASIC‘

**D**er Dragon 32 verfügt über einen besonderen BASIC-Dialekt, der als ‚Microsoft Extended Colour BASIC‘ bekannt ist. Auch auf einigen anderen Computermodellen – speziell der Farbcomputerserie von Tandy – läuft diese BASIC-Variante. Microsoft BASIC läßt sich einfach programmieren und verfügt über einen umfangreichen Befehlssatz, mit dem unter anderem das Zeichnen von Kreisen, Linien und anderen geometrischen Formen unterstützt wird. Diese Formen lassen sich auch farblich gestalten, und mit wenig Programmieraufwand kann damit eine interessante Grafik entstehen. Acht Farben sind verfügbar, bei hoher Auflösung können aber nur zwei oder vier Farben eingesetzt werden.

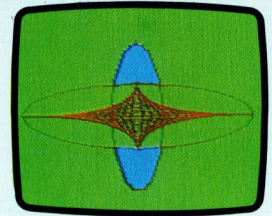
### Verschiedene Auflösungsarten

Die Standarddarstellung von 16 Zeilen mit je 32 Zeichen ist die niedrigste verfügbare Auflösung. Mit dem PRINT-Befehl kann ein Zeichen an jede der 512 Positionen des Bildschirms gebracht werden. Zusätzlich zu dem normalen Zeichensatz sind in diesem Modus noch weitere 16 Grafikzeichen in acht Farben verfügbar.

Die nächsthöhere Auflösungsebene unterteilt den Bildschirm in 32 Reihen zu je 64 Spalten. Jedes Zeichen mißt damit nur ein Viertel der Größe eines normalen Zeichens. Mit dem SET-Befehl können Punkte dieser Auflösungsart auf den Bildschirm „geplottet“ werden; der RESET-Befehl gibt die Möglichkeit, die Anweisung wieder rückgängig zu machen. Diese beiden Darstellungsmodi können miteinander kombiniert werden. Sie werden als „Textdarstellung mit niedriger Auflösung“ bezeichnet. Die fünf Ebenen mit hoher Auflösung lassen sich allerdings nicht in Verbindung mit anderen Grafikarten verwenden. Sie werden über den Befehl PMODE angesprochen und bieten eine ganze Reihe von Kombinationsmöglichkeiten in der Feinheit ihrer Auflösung und in der verfügbaren Farbenzahl.

PMODE	Auflösung	Anzahl Farben
0	128 x 96	2
1	128 x 96	4
2	128 x 192	2
3	128 x 192	4
4	256 x 192	2

Es besteht natürlich eine Beziehung zwischen dem Grad der Auflösung, der Anzahl der Farben und der für die Darstellung benötigten Speicherkapazität. Bei der Entwicklung von BASIC-Programmen mit hochauflösender Grafik müssen diese Komponenten berücksichtigt werden. Obwohl bei hoher Bildschirmauflösung nur eine begrenzte Anzahl Farben zur



Auf dem Dragon lassen sich interessante Grafiken mit nur wenigen Befehlen erzeugen.

#### Paint

Mit diesem Befehl können interessante Effekte erzielt werden. Paint färbt von einem bestimmten Punkt aus den Bildschirm ein, bis die Farbfläche auf eine Grenzlinie trifft. Auf diese Weise können alle geschlossenen Figuren wie Kreise, Dreiecke etc. auf einfache Weise mit Farbe ausgefüllt werden.

#### DRAW

DRAW imitiert die Bewegung eines Bleistiftes auf dem Bildschirm und gibt dem Anwender die Möglichkeit, Linien in alle vier Richtungen zu ziehen. Der Befehl DRAW erlaubt auch die Vergrößerung und die Rotation einer Figur.

#### GET und PUT

Mit GET wird ein ganzer Bildschirminhalt im Computer gespeichert, während PUT den gespeicherten Inhalt wieder auf den Bildschirm bringt.

#### PSET und PRESET

Diese beiden Befehle gelten für die hochauflösende Grafik und schalten einen Punkt an oder aus. Sie entsprechen den SET- und RESET-Befehlen. Mit ihnen kann auch die Farbe eines Bildpunktes bestimmt werden.

#### LINE

Bei hoher Auflösung verbindet dieser Befehl zwei Punkte durch eine gerade Linie.

#### CIRCLE

CIRCLE zeichnet in der hohen Auflösung von einem festgelegten Punkt aus einen Kreis mit einem bestimmten Radius. Über diesen Befehl können auch Teile von Kreisen oder Ellipsen dargestellt werden.

Dieses kurze Programm führt einige Möglichkeiten der hohen Auflösungsebene vor. Der verwendete PMODE 3 ist zwar nicht die höchste Auflösung, läßt aber den Gebrauch von vier Farben zu.

```
10 PCLS:PMODE3,1
20 SCREEN 1,0
30 COLOR 0,1
40 FOR X=0 TO 127 STEP 10
50 LINE(X,85)-(127,85+X/3),
   PSET
60 LINE (X,85)-(127,85+X/3),
   PSET
70 LINE (255-X,85)-
   (127,85+X/3),PSET
80 LINE(255-X,85)-
   (127,85+X/3),PSET
90 NEXT X
100 CIRCLE (127,85),128,4,0,3
110 CIRCLE (127,85),30,4,3
120 PAINT (130,30),3,4
130 PAINT (130,130),3,4
140 GOTO 140
150 END
```

Verfügung steht, bietet der Dragon mit dem SCREEN-Befehl die Möglichkeit, zwei Farbsätze anzusprechen. So wählt zum Beispiel SCREEN 1,0 eine hohe Auflösungsebene mit dem Farbsatz 0 aus, während SCREEN 1,1 die gleiche Auflösung anspricht, diesmal aber mit dem Farbsatz 1. Für seinen niedrigen Preis verfügt der Dragon 32 über viele Grafikbefehle.



# Der Zauberstab

**Der Lichtgriffel – oder Lightpen – ist ein Computerzubehör zum Zeichnen oder zum Auswählen von Details am Bildschirm. Wie funktioniert so ein Wunderding?**

**E**in Lichtgriffel ist ein stabförmiges Gerät (einem Schreibgriffel ähnlich, daher der Name), das mit einem Spiralkabel versehen ist. Wenn man die Spitze des Lichtgriffels dicht an den Bildschirm führt – bei manchen Systemen muß der Schirm zur Betätigung eines Schalters in der Spitze noch leicht berührt werden –, kann der Rechner genau feststellen, auf welchen Punkt des Schirms gezeigt wird.

Ermöglicht wird dies durch einen Lichtempfänger in der Spitze des Griffels, der bei der Darstellung eines Lichtsignals einen elektrischen Impuls registriert – das Bild wird ja zeilenweise durch einen über den Schirm wandernden Lichtsignal aufgebaut. Eine spezielle Schaltung auf dem Video-Steuerbaustein (Video-Controller) hält fest, an welcher Bildschirmstelle sich das Signal befand, als der Impuls vom Lichtgriffel eintraf.

## Menü-Auswahl

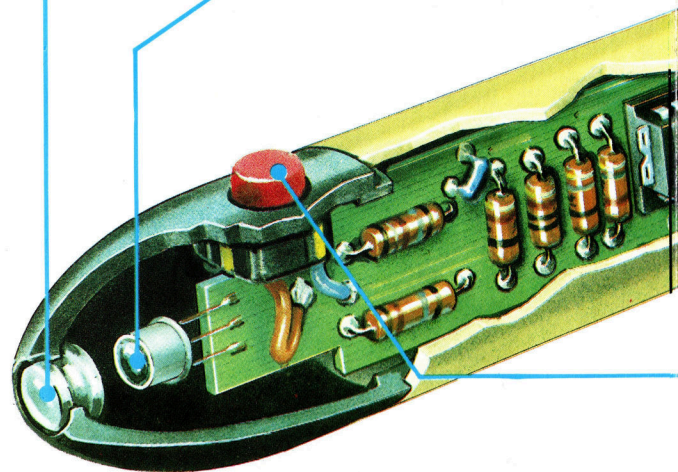
Der Lichtgriffel wird hauptsächlich eingesetzt, wenn auf dem Schirm dargestellte Details auszuwählen sind. Bei vielen Anwenderprogrammen wird mit 'Menü'-Angeboten gearbeitet. Ein Menü ist eine Auswahl-Liste, die auf dem Schirm dargestellt wird, wobei der Anwender bestimmte Programme oder Funktionen auswählen kann. Bei einem Kontenführungsprogramm könnte das Menü etwa so aussehen:

- 1) Auszahlung –
- 2) Kontostandprüfung –
- 3) Zahlungseingang –
- 4) ...

Normalerweise müßte der Anwender die gewünschte Programmfunktion durch Drücken einer Zifferntaste oder durch Eingabe des entsprechenden Befehls definieren. Mit dem Lichtgriffel braucht er statt dessen nur auf die gewählte Variante zu zeigen. Je nach Programmspezifikation wird die gewählte Funktion optisch herausgehoben.

### Linse

Die Lichtmenge, die bei der Abtastung von einem einzelnen Bildpunkt emittiert wird, ist so klein, daß man eine Sammellinse braucht, die die Strahlen auf den Eingang des Lichtempfängers bündelt.



Eine Schwierigkeit liegt darin, daß ein Programm speziell auf das Arbeiten mit dem Lichtgriffel abgestellt werden muß. Man braucht aber eigentlich nur ein kleines Unterprogramm zu schreiben, daß die Koordinaten der Lichtgriffelposition vom Video-Controller abfragt und dann auswertet, welche Wahlmöglichkeit an der entsprechenden Stelle des Schirms gestanden hat. Leider bieten nur wenige Hersteller ihre Software-Pakete in einer lichtgriffelfähigen Version an.

Man kann den Lichtgriffel aber nicht nur für Menüs, sondern auch zum Entwurf von Zeichnungen am Bildschirm verwenden. Bei den meisten Heimcomputern, für die es einen Lichtgriffel gibt, wird ein entsprechendes Software-Paket geliefert. Der Bildschirm kann nun als „Maltafel“ verwendet werden, also als Eingabegerät zur Erstellung eigener Grafiken. Die zur Verfügung stehenden Farben und Zusatzfunktionen sind am Schirmrand aufgeführt. Der Lichtgriffel wird kurz auf die gewünschte Farbe gesetzt und hinterläßt dann bei anschließender Bewegung über die Zeichenfläche eine Linie in dieser Farbe.

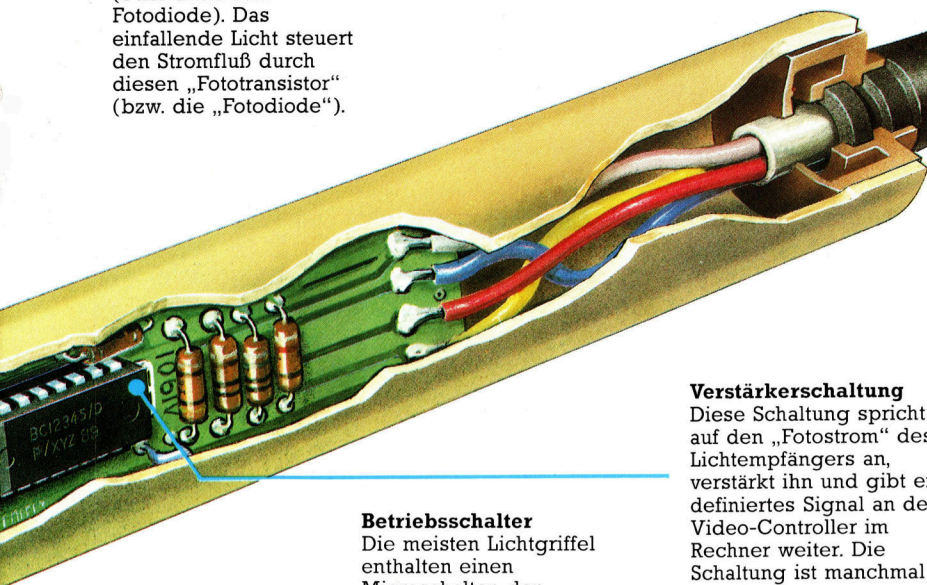
Der Benutzer kann am unteren Bildrand auch unterschiedliche Strichstärken und Linienstrukturen oder Bildelemente wie Kreise und Quadrate aufrufen. Kurz gesagt: Alles in unserem Artikel „Ein Mosaik aus tausend Punkten“ (s. S. 40) Besprochene ist mit dem





### Lichtempfänger

Zum Lichtnachweis dient ein Halbleiter-Bauelement – im Prinzip ein Transistor mit abgesägter Kappe (oder auch eine Fotodiode). Das einfallende Licht steuert den Stromfluß durch diesen „Fototransistor“ (bzw. die „Fotodiode“).



### Betriebsschalter

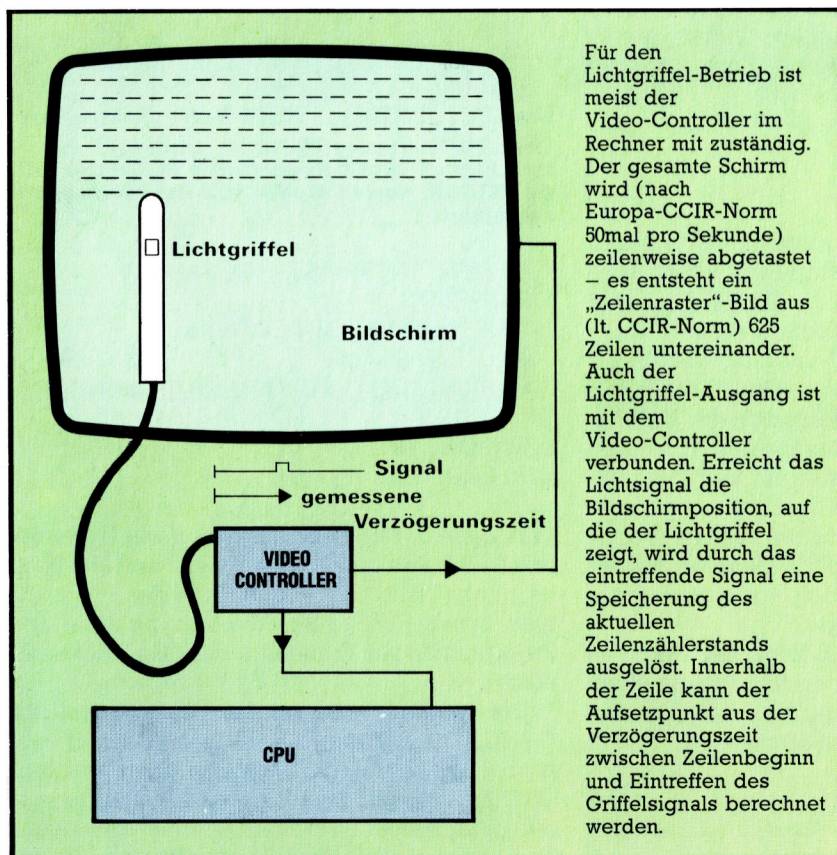
Die meisten Lichtgriffel enthalten einen Microschalter, der entweder durch Fingerdruck oder durch Anlegen der Griffelspitze an den Schirm betätigt wird. Dieser Schalter ist erforderlich, um ein Ansprechen des Griffels auf Störlicht (Raumbeleuchtung) bei Nichtgebrauch zu verhindern.

### Verstärkerschaltung

Diese Schaltung spricht auf den „Fotostrom“ des Lichtempfängers an, verstärkt ihn und gibt ein definiertes Signal an den Video-Controller im Rechner weiter. Die Schaltung ist manchmal auch in einem separaten Gehäuse außerhalb des eigentlichen Lichtgriffels untergebracht.

### Anschlußkabel

Dieses Spiralkabel (wie am Telefonhörer) wird mit dem Computer verbunden und führt zum Video-Controller.



Für den Lichtgriffel-Betrieb ist meist der Video-Controller im Rechner mit zuständig. Der gesamte Schirm wird (nach Europa-CCIR-Norm 50mal pro Sekunde) zeilenweise abgetastet – es entsteht ein „Zeilenraster“-Bild aus (lt. CCIR-Norm) 625 Zeilen untereinander. Auch der Lichtgriffel-Ausgang ist mit dem Video-Controller verbunden. Erreicht das Lichtsignal die Bildschirmposition, auf die der Lichtgriffel zeigt, wird durch das eintreffende Signal eine Speicherung des aktuellen Zeilenzählerstands ausgelöst. Innerhalb der Zeile kann der Aufsetzpunkt aus der Verzögerungszeit zwischen Zeilenbeginn und Eintreffen des Griffelsignals berechnet werden.

Lichtgriffel machbar, und zwar viel komfortabler als mit der Tastatur. Es kommen jetzt auch schon Computerspiele auf den Markt, die vom Lichtgriffel Gebrauch machen. Die bösen Monster kann man mit dem Lichtgriffel viel leichter eliminieren als über die Tastatur. Der Lichtgriffel eignet sich auch für gemächliche Spiele wie Computer-Schach – man zeigt, wohin der Springer soll, und der Rechner erledigt das eigentliche Setzen.

## Computer Aided Design

Die größte Anwendergruppe stellen z. Z. aber wohl Planungs- und Konstruktionsbüros. Systeme für rechnerunterstützten Entwurf (Computer Aided Design = CAD) werden in der Werbung für Pkw-Neuentwicklungen groß herausgestellt – tatsächlich sind es ganz normale Rechnersysteme, jedoch mit spezieller Software und hervorragender Grafik-Ausstattung. Ein CAD-System zum Entwurf elektronischer Schaltungen etwa bietet dem Benutzer auf dem Schirm eine Palette aller möglichen Bauelemente an; sie können vom Entwickler mit dem Lichtgriffel „aufgespickt“ und je nach Bedarf auf dem Schirm ins Schaltbild eingefügt werden.

Der Lichtgriffel ist eins der besten Beispiele für ein praktisches und nützliches Computer-Zubehör, dessen Gebrauch außerdem noch Spaß macht.



# Unter Kontrolle

**Alle BASIC-Versionen beinhalten ‚Kontroll-Strukturen‘, die den Ablauf eines Programmes bestimmen. Einige Maschinen bieten eine große Auswahl verschiedener Alternativen, wobei zahlreiche Unterschiede zu beachten sind.**

In dieser Folge des Kurses geben wir Ihnen eine Zusammenfassung des bisher Behandelten. Zunächst einmal die Zusammenfassung: Eine höhere Programmiersprache als BASIC bietet dem Anwender einen Befehlssatz, der intern in eine dem Computer verständliche Form umgewandelt wird. Jedes Computerprogramm kann durch Verwendung von zwei einfachen Ausdrücken, sogenannten ‚Konstruktionen‘, geschrieben werden. Dies sind ‚Sequenz‘-Konstruktionen und ‚Kontroll-Strukturen‘, von denen in BASIC nur zwei von Bedeutung sind: IF... THEN... ELSE und WHILE... DO.

Die Sequenz-Konstruktion gestattet, die Aufgabe in mehrere Unteraufgaben aufzuteilen, die dann, nacheinander ausgeführt, die Hauptaufgabe ergeben. Die Größe einer Unteraufgabe ist abhängig von der verwendeten Programmiersprache. In BASIC wird eine Unteraufgabe durch die Anweisungen in einer Zeile, und die Sequenz durch die Zeilennummer repräsentiert. Nehmen wir an, Sie hätten die Aufgabe, einen einer Variablen zugeordneten Wert mit 10 zu multiplizieren. Eine mögliche Sequenz zur Lösung wäre:

```
110 INPUT N
120 LET N=N*10
130 PRINT N
```

Zusätzlich zu den Sequenz-Konstruktionen brauchen wir auch Kontroll-Strukturen. Dies sind Konstruktionen, die die Reihenfolge der Programmanweisungen beeinflussen.

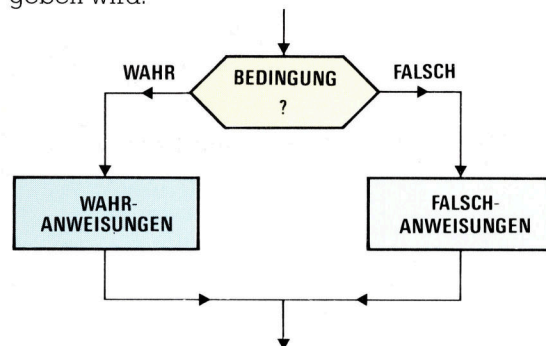
Die einfachste Kontroll-Struktur, die BASIC bietet, ist GOTO. Dies ist ein unbedingter Sprung (oder auch Verzweigung), der die Programmausführung zur angegebenen Zeilennummer verzweigt. Es wird kein Vergleich benötigt. GOSUB ist ebenfalls eine unbedingte Verzweigung. Das Programm kehrt jedoch nach dem zugehörigen RETURN sofort wieder zu dem Programmteil zurück, von dem aus die GOSUB-Anweisung erfolgte.

Die IF... THEN... ELSE-Kontroll-Struktur ist in BASIC verfügbar. Sie verwendet die Form der IF... THEN-Anweisung und hat folgende Syntax (‚Syntax‘ bedeutet ‚Form‘):

IF (angegebene Bedingung) ist wahr  
THEN führe angegebene Anweisung aus  
(ELSE) führe die nächste Anweisung aus

Beachten Sie, daß im Standard-BASIC der ELSE-Teil von IF... THEN... ELSE bereits implementiert ist. In einigen BASIC-Varianten und bestimmten anderen Sprachen ist ELSE Bestandteil der Anweisung.

IF... THEN... ELSE (IF... THEN in BASIC) führt eine von zwei Unteraufgaben aus, je nachdem, ob eine bestimmte Bedingung wahr ist oder nicht. Betrachten Sie das folgende Programm. Es dient zur Berechnung der Quadratwurzeln von Zahlen, die über die Tastatur eingegeben werden. Der Programmlauf wird beendet, sobald der ‚Flag‘-Wert -9999 eingegeben wird:



## Die IF... THEN... ELSE-Kontroll-Struktur

Wenn die Bedingung WAHR ist, werden die entsprechenden WAHR-Anweisungen ausgeführt. Ist sie FALSCH, werden die FALSCH-Anweisungen ausgeführt.

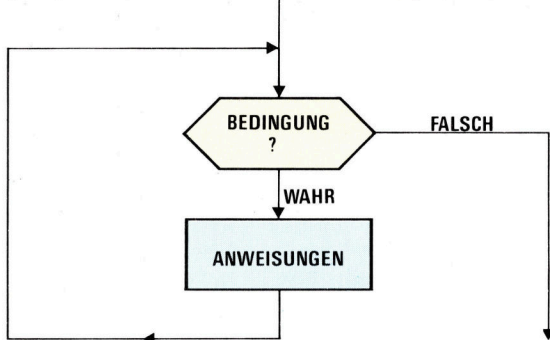
```
10 PRINT "GEBEN SIE EINE ZAHL EIN"
20 INPUT N
30 IF N=-9999 THEN GOTO 70
40 LET S=SQR(N)
50 PRINT "DIE QUADRATWURZEL VON ";N;" IST ";S
60 GOTO 10
70 END
```

Was Zeile 30 bedeutet, kann man wie folgt ausdrücken: „Wenn (IF) es wahr ist, daß  $N = -9999$  ist, dann (THEN) gehe zum Ende des Programmes, sonst (ELSE) führe die nächste Zeile des Programmes zur Ermittlung der Quadratwurzel aus.“

Die andere wesentliche Kontroll-Struktur (WHILE... DO) ist in BASIC nicht direkt verfügbar, kann jedoch leicht simuliert werden. WHILE... DO ist eine Art Schleife und bedeutet: „Wiederhole eine oder mehrere Anweisungen solange, bis (WHILE) eine bestimmte Be-



dingung wahr ist" oder „solange, bis (WHILE) eine bestimmte Bedingung wahr ist, mache (DO) etwas". WHILE... DO überprüft zuerst



### Die DO...WHILE-Kontroll-Struktur

Die Schleife wird wiederholt, so lange die Bedingung wahr ist. Die Anweisungen werden nicht mehr ausgeführt, wenn die Start-Bedingung falsch ist.

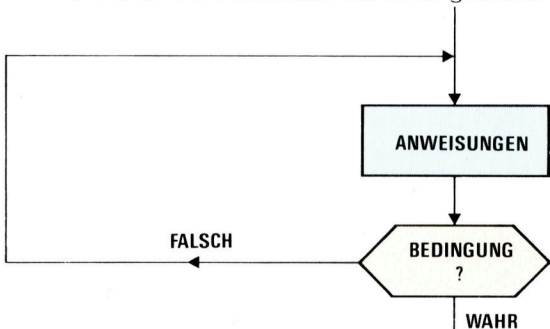
immer die Bedingung, bevor die Anweisungen ausgeführt werden. Wenn also dieser Test bereits beim ersten Mal negativ ausfällt, werden die Anweisungen nicht ausgeführt. Nehmen wir als Beispiel einmal ein Spielprogramm, das den Spieler dazu auffordert 'DRUECKE LEERTASTE, WENN BEREIT'. Dieser Programmteil könnte auf folgende Weise geschrieben werden:

```
WHILE Leertaste nicht gedrückt wird,
DO Abfrage der Tastatur
starte Spiel
```

```
250 PRINT "DRUECKE LEERTASTE, WENN
BEREIT"
260 FOR X=0 TO 1 STEP 0
270 IF INKEY$="" THEN LET X=2
280 NEXT X
290 GOSUB *START*
```

In diesem Programmteil wird die Schleife (zur Abfrage der Tastatur) nur ausgeführt, wenn die Leertaste nicht gedrückt wurde. Wenn die Leertaste gedrückt wurde (z. B. INKEY\$=" "), verläßt das Programm die FOR...NEXT-Schleife und geht zu Zeile 290, die die START-Unterroutine aufruft.

Bisher haben Sie noch nie mit STEP gearbeitet.



### Die REPEAT...UNTIL-Kontroll-Struktur

Die Schleife wird so lange wiederholt, bis die Bedingung wahr ist. Die Anweisungen werden mindestens einmal ausgeführt.

tet. Und gerade hier haben wir eine etwas ungewöhnliche Anwendung für diese Anweisung. Wenn Sie eine FOR...NEXT-Schleife verwenden, gestattet STEP den Wert der Zählvariablen in einer anderen Schrittweite als 1 zu erhöhen. FOR I=1 TO 10 STEP 2 bewirkt, daß I beim ersten Schleifendurchlauf den Wert 1 hat, danach den Wert 3, 5, 7 und 9. Die nächste Erhöhung (auf 11) überschreitet die angegebene Grenze von 10 und somit wird die Schleife beendet. Man kann die Zählvariable sogar rückwärts zählen lassen. FOR I=10 TO 1 STEP -1 bewirkt, daß I von 10 rückwärts bis 1 gezählt wird. Verwendet man STEP 0, so ist das ein Trick, mit dem man sichergehen kann, daß die Schleife nicht beendet wird, bevor X künstlich erhöht wird, so wie in unserer IF...THEN-Anweisung.

Eine weitere, nicht unbedingt notwendige, aber nützliche Kontroll-Struktur, ist normalerweise unter der Bezeichnung CASE bekannt. In BASIC ist diese Art Anweisung bei der Verwendung von ON...GOTO oder ON...GOSUB implementiert. Und so funktioniert sie: ON...GOTO ist eine Anweisung, die mehrere verschiedene Verzweigungen bewirken kann. In ihr sind eine Vielzahl von IF...THEN-Vergleichen in einer einzigen Anweisung vereinigt. Betrachten Sie den folgenden Programmteil, der die Zahlen 1 bis 7 in die Bezeichnungen der sieben Tage einer Woche umwandelt:

```
1050 IF D=1 THEN GOTO 2020
1060 IF D=2 THEN GOTO 2040
1070 IF D=3 THEN GOTO 2060
1080 IF D=4 THEN GOTO 2080
1090 IF D=5 THEN GOTO 3000
2000 IF D=6 THEN GOTO 3020
2010 IF D=7 THEN GOTO 3040
2020 PRINT "MONTAG"
2030 GOTO *ENDE*
2040 PRINT "DIENSTAG"
2050 GOTO *ENDE*
2060 PRINT "MITTWOCH"
2070 GOTO *ENDE*
2080 PRINT "DONNERSTAG"
2090 GOTO *ENDE*
3000 PRINT "FREITAG"
3010 GOTO *ENDE*
3020 PRINT "SAMSTAG"
3030 GOTO *ENDE*
3040 PRINT "SONNTAG"
3050 GOTO *ENDE*
```

Eine erheblich effektivere Methode, um eine solche Zuordnung in BASIC durchzuführen, ist die Verwendung von ON...GOTO. Betrachten Sie folgendes Beispiel:

```
1050 ON D GOTO 2020, 2040, 2060, 2080,
3000, 3020, 3040
```

ON...GOSUB arbeitet auf dieselbe Art und Weise, mit dem Unterschied, daß der Wert der

DEZIMAL	BINÄR	ZEICHEN
32	0 0 1 0 0 0 0 0	= (space)
33	0 0 1 0 0 0 0 1	= !
34	0 0 1 0 0 0 1 0	= "
35	0 0 1 0 0 0 1 1	= #
36	0 0 1 0 0 1 0 0	= \$
37	0 0 1 0 0 1 0 1	= %
38	0 0 1 0 0 1 1 0	= &
39	0 0 1 0 0 1 1 1	= '
40	0 0 1 0 1 0 0 0	= (
41	0 0 1 0 1 0 0 1	= )
42	0 0 1 0 1 0 1 0	= *
43	0 0 1 0 1 0 1 1	= +
44	0 0 1 0 1 1 0 0	= ,
45	0 0 1 0 1 1 0 1	= -
46	0 0 1 0 1 1 1 0	= .
47	0 0 1 0 1 1 1 1	= /
48	0 0 1 1 0 0 0 0	= 0
49	0 0 1 1 0 0 0 1	= 1
50	0 0 1 1 0 0 1 0	= 2
51	0 0 1 1 0 0 1 1	= 3
52	0 0 1 1 0 1 0 0	= 4
53	0 0 1 1 0 1 0 1	= 5
54	0 0 1 1 0 1 1 0	= 6
55	0 0 1 1 0 1 1 1	= 7
56	0 0 1 1 1 0 0 0	= 8
57	0 0 1 1 1 0 0 1	= 9
58	0 0 1 1 1 0 1 0	= :
59	0 0 1 1 1 0 1 1	= ;
60	0 0 1 1 1 1 0 0	= <
61	0 0 1 1 1 1 0 1	= =
62	0 0 1 1 1 1 1 0	= >
63	0 0 1 1 1 1 1 1	= ?
64	0 1 0 0 0 0 0 0	= @
65	0 1 0 0 0 0 0 1	= A
66	0 1 0 0 0 0 1 0	= B
67	0 1 0 0 0 0 1 1	= C
68	0 1 0 0 0 1 0 0	= D
69	0 1 0 0 0 1 0 1	= E
70	0 1 0 0 0 1 1 0	= F
71	0 1 0 0 0 1 1 1	= G
72	0 1 0 0 1 0 0 0	= H
73	0 1 0 0 1 0 0 1	= I
74	0 1 0 0 1 0 1 0	= J
75	0 1 0 0 1 0 1 1	= K
76	0 1 0 0 1 1 0 0	= L
77	0 1 0 0 1 1 0 1	= M
78	0 1 0 0 1 1 1 0	= N
79	0 1 0 0 1 1 1 1	= O
80	0 1 0 1 0 0 0 0	= P
81	0 1 0 1 0 0 0 1	= Q
82	0 1 0 1 0 0 1 0	= R
83	0 1 0 1 0 0 1 1	= S
84	0 1 0 1 0 1 0 0	= T
85	0 1 0 1 0 1 0 1	= U
86	0 1 0 1 0 1 1 0	= V
87	0 1 0 1 0 1 1 1	= W
88	0 1 0 1 1 0 0 0	= X
89	0 1 0 1 1 0 0 1	= Y
90	0 1 0 1 1 0 1 0	= Z
91	0 1 0 1 1 0 1 1	= [
92	0 1 0 1 1 1 0 0	= \
93	0 1 0 1 1 1 0 1	= ]
94	0 1 0 1 1 1 1 0	= ^
95	0 1 0 1 1 1 1 1	= _
96	0 1 1 0 0 0 0 0	= `
97	0 1 1 0 0 0 0 1	= a
98	0 1 1 0 0 0 1 0	= b
99	0 1 1 0 0 0 1 1	= c
100	0 1 1 0 0 1 0 0	= d
101	0 1 1 0 0 1 0 1	= e
102	0 1 1 0 0 1 1 0	= f
103	0 1 1 0 0 1 1 1	= g
104	0 1 1 0 1 0 0 0	= h
105	0 1 1 0 1 0 0 1	= i
106	0 1 1 0 1 0 1 0	= j
107	0 1 1 0 1 0 1 1	= k
108	0 1 1 0 1 1 0 0	= l
109	0 1 1 0 1 1 0 1	= m
110	0 1 1 0 1 1 1 0	= n
111	0 1 1 0 1 1 1 1	= o
112	0 1 1 1 0 0 0 0	= p
113	0 1 1 1 0 0 0 1	= q
114	0 1 1 1 0 0 1 0	= r
115	0 1 1 1 0 0 1 1	= s
116	0 1 1 1 0 1 0 0	= t
117	0 1 1 1 0 1 0 1	= u
118	0 1 1 1 0 1 1 0	= v
119	0 1 1 1 0 1 1 1	= w
120	0 1 1 1 1 0 0 0	= x
121	0 1 1 1 1 0 0 1	= y
122	0 1 1 1 1 0 1 0	= z
123	0 1 1 1 1 0 1 1	= {
124	0 1 1 1 1 1 0 0	=
125	0 1 1 1 1 1 0 1	= }
126	0 1 1 1 1 1 1 0	= ~

### ASCII

Hier ist eine Liste aller ASCII-Werte zwischen 32 und 126 und ihrer binären Äquivalente. Die Bedeutung der Zeichen variiert von Gerät zu Gerät.



Variablen bestimmt, zu welcher Unterroutine verzweigt werden soll.

Im Laufe der Jahre sind viele verschiedene Möglichkeiten zur Darstellung der Zeichen des Alphabets und der Satzzeichen entwickelt worden. Das zur Zeit universellste System ist der ASCII-Code (ASCII = American Standard Code for Information Interchange). Der ASCII-Code verwendet ein Byte, um 94 druckbare Zeichen, das Leerzeichen und eine Anzahl von Kontroll-Zeichen darzustellen. Acht Bit könnten 256 verschiedene Kombinationen ermöglichen, doch sind dies zu viele, als daß man sie zur Darstellung der Zeichen einer normalen Schreibmaschine oder einer Computer-Tastatur benötigen würde. Aus diesem Grund werden nur sieben Bit benutzt, aus denen 128 Kombinationen gebildet werden können. (Das achte Bit wird gewöhnlich überhaupt nicht genutzt, außer zur Darstellung eines zusätzlichen Zeichensatzes oder Grafikzeichens). Die binären und dezimalen ASCII-Codes für den Standard-Bereich sind in der Tabelle aufgeführt.

Wie Sie aus der Tabelle ersehen können, ist der ASCII-Code für den Buchstaben A = 65 und für den Buchstaben B = 66. Die Codes für die Kleinbuchstaben a und b sind 97 und 98. Jeder Kleinbuchstabe hat einen ASCII-Wert, der um den Wert 32 höher ist als der des entsprechenden Großbuchstabens. Dieser Umstand macht die Umwandlung von Kleinbuchstaben in Großbuchstaben und umgekehrt sehr leicht. Um eine solche Umwandlung durchzuführen, benötigen Sie zwei weitere Funktionen: ASC und CHR\$.

Die ASC-Funktion nimmt ein druckbares Zeichen und verwandelt es in den entsprechenden ASCII-Code. PRINT ASC("A") ergäbe so den Wert 65 als Ergebnis auf dem Bildschirm. PRINT ASC("b") hätte als Ergebnis den Wert 98. Die CHR\$-Funktion macht genau das Gegenteil; sie nimmt eine Zahl, nimmt an, daß es ein ASCII-Code ist, und stellt dann als Ergebnis das zugehörige Zeichen dar. Als Ergebnis von PRINT CHR\$(65) würde A und bei PRINT CHR\$(98) würde b auf dem Bildschirm ausgegeben. Die CHR\$- und ASC-Funktionen haben ein weit gestreutes Anwendungsgebiet. Meist werden sie zusammen mit den Funktionen LEFT\$, RIGHT\$ und MID\$ in Programmen verwendet, in denen viel mit Zeichenketten gearbeitet wird. Hier ist ein kurzes Programm, das ein Zeichen von der Tastatur annimmt, dann überprüft, ob es ein Kleinbuchstabe ist und ihn gegebenenfalls in einen Großbuchstaben umwandelt:

```
10 REM UMWANDLUNG KLEIN- IN
    GROSSBUCHSTABEN
20 PRINT "GEBEN SIE EIN ZEICHEN EIN"
30 INPUT C$
40 LET C=ASC(C$)
50 IF C>90 THEN LET C=C-32
60 PRINT CHR$(C)
```

Abschließend ein Blick auf Funktionen, die eventuell in Ihrer BASIC-Version nicht verfügbar sind. Fast alle Versionen der Sprache BASIC gestatten den Programmierern, neue Funktionen zu entwickeln, die dann meist genauso einfach zu verwenden sind wie die direkt verfügbaren Funktionen. Die DEF-Anweisung signalisiert BASIC, daß eine neue Funktion definiert werden soll. Im folgenden Programm zeigen wir Ihnen, wie man eine Funktion zur Berechnung des Volumens einer Kugel (die Formel lautet  $V = \frac{4}{3}\pi r^3$ , wobei r der Radius der Kugel und  $\pi$  [pi] die Konstante mit einem Wert von annähernd 3.14159 ist):

```
10 REM FUNKTION ZUR BERECHNUNG DES
    VOLUMENS EINER KUGEL
20 DEF FNV(X)=4*3.14159*X*X*X/3
30 PRINT "GEBE RADIUS DER KUGEL EIN"
40 INPUT R
50 PRINT "DAS VOLUMEN EINER KUGEL MIT
    DEM RADIUS ";R;" BETRAEGT"
60 PRINT FNV(R)
70 END
```

Diese Art der Definierung einer Funktion ist die einfachste, doch lassen Sie uns einen genaueren Blick auf die Zeile 20 werfen:

DEFiniert	Funktions-Identifizierer
↓	↓
20 DEF FNV(X)=4*3.14159*X*X*X/3	
↑	↑
FuNktion	Schein-Variable

## „Schein-Variablen“

Wenn die Funktion definiert ist, steht hinter den Buchstaben FN ein Buchstabe zur Identifizierung – in unserem Beispiel ist es der Buchstabe V –, gefolgt von einer ‚Schein-Variable‘. Diese Schein-Variable muß auch in der Funktionsdefinition auf der rechten Seite des Gleichheitszeichens verwendet werden. Wenn die Funktion innerhalb eines Programmes verwendet wird, kann anstelle der Schein-Variable jede beliebige numerische Variable verwendet werden. Will man die eben definierte Formel zu einem späteren Zeitpunkt an einer anderen Stelle des Programmes noch einmal verwenden, so könnte das so aussehen:

```
999 LET A=66
1000 LET B=FNV(A)
1010 PRINT B
1020 LET C=5
1030 LET D=B+FNV(C)
1040 PRINT D
1050 LET G=FNV(16)
1060 PRINT G
```

Einige BASIC-Versionen erlauben die Verwendung verschiedener Variablen in definierten Funktionen. Ist dies der Fall, kann z. B. eine Funktion zur Berechnung des Durchschnitts



zweier Zahlen so geschrieben werden:

```
100 DEF FNA(B,C)=(B+C)/2
110 INPUT "GEBEN SIE ZWEI ZAHLEN EIN";B,C
120 LET A=FNA(B,C): REM DIE
    DURCHSCHNITTS FUNKTION
130 PRINT "DER DURCHSCHNITT VON ";B;
    " UND ";C; "IST "A;
```

Beachten Sie, daß in Zeile 110 zwei Anweisungen in einer vereint sind. Dies ist bei den meisten BASIC-Versionen möglich, da Zeichen, die in Anführungszeichen hinter der INPUT-Anweisung stehen, auf dem Bildschirm ausgegeben werden. Die Anweisung in Zeile 110 entspricht also folgenden zwei Zeilen:

```
110 PRINT "GEBEN SIE ZWEI ZAHLEN EIN"
115 INPUT A,B
```

In Zeile 120 werden ebenfalls zwei Anweisungen in einer Zeile geschrieben, indem ein Doppelpunkt als Trennung verwendet wurde. Man kann also Anweisungen, die normalerweise in einer Zeile geschrieben würden, auch in eine Zeile schreiben, wobei jede Anweisung

#### Antworten zu den Übungen Fehler

Sie erhalten einen 'OUT OF DATA ERROR', da insgesamt 12 Werte innerhalb der DATA-Anweisung in Zeile 130 stehen müßten. Ein weiterer Fehler tritt in Zeile 100 auf, sobald versucht wird, eine Speicheradresse A(4,1) zu adressieren. Zeile 100 muß wie folgt aussehen:

```
100 PRINT A(X,Y)
```

#### Zuordnen von Werten

Nachfolgend finden Sie ein Programm, das die notwendigen Funktionen ausführt. Ihr eigenes Programm kann natürlich etwas anders aussehen.

```
10 DIM A(8,13)
20 FOR R=1 TO 7
30 FOR C=1 TO 12
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 REM ADDIERE GESAMTAUSGABEN
80 GOSUB 300
90 REM DRUCKE ANGEFORDERTE DATEN
100 GOSUB 200
110 PRINT "WEITERE DATEN?"
120 PRINT "J ODER N"
130 INPUT A$
140 IF A$="N" THEN GOTO 160
150 GOTO 100
160 END
200 PRINT "WELCHER MONAT?"
210 PRINT "1- FUER JANUAR,"
220 PRINT "13- FUER GESAMTAUSGABEN, ETC"
230 INPUT M
240 PRINT "WELCHE AUSGABEN?"
250 PRINT "1- FUER BENZIN"
260 PRINT "8- FUER GESAMTAUSGABEN, ETC"
270 INPUT X
280 PRINT "WERT IST ";A(X,M)
290 RETURN
300 FOR R=1 TO 7
310 LET T=0
320 FOR C=1 TO 12
330 LET T=T+A(R,C)
340 NEXT C
350 LET A(R,13)=T
360 NEXT R
370 FOR C=1 TO 13
380 LET T=0
390 FOR R=1 TO 7
400 LET T=T+A(R,C)
410 NEXT R
420 LET A(8,C)=T
430 NEXT C
440 RETURN
500 REM HIER FOLGEN IHRE DATA
510 REM ANWEISUNGEN — INSGESAMT 84
520 REM 'DATA 11.35, 9.87' ETC.
```

von der vorherigen durch einen Doppelpunkt getrennt werden muß. Dies kann bei langen Programmen erheblich Platz einsparen. Der Nachteil ist, daß Programme nicht mehr so leicht verständlich sind und leichter Fehler auftreten können.

In den folgenden Teilen des BASIC-Kurses liegt der Schwerpunkt auf dem Aufbau und der Entwicklung von Programmen.

## BASIC-Dialekte

### ASCII

Der Dragon hat keine ASCII-Standard-Version. Somit verfügt er über keine Kleinbuchstaben. Lesen Sie im Handbuch Ihres Dragon nach, welche Möglichkeiten Sie mit dem Zeichensatz haben.

### DEF FN(A,B)

Auf dem Oric, dem VC-20, dem Dragon und dem Commodore 64 können nicht mehr als eine Variable innerhalb der eckigen Klammern verwendet werden.

### ASC()

Beim Spectrum und dem ZX81 ersetzen Sie:  
ASC(A\$) durch CODE A\$  
und  
CHR\$(65) durch CHR\$65

### CHR\$()

Wenn das Argument ein Ausdruck ist, muß es in eckige Klammern gesetzt werden. Einfache Argumente – wie A\$ – müssen nicht in Klammern gesetzt werden.

### ON... GOSUB

Diese Anweisungen sind auf dem ZX81, dem Spectrum und dem Lynx nicht verfügbar.

### ON... GOTO

### INPUT " "

Beim Acorn B ersetzen Sie:  
INPUT "MITTEILUNG";MS\$  
durch  
INPUT "MITTEILUNG";MS\$

### DEF FN

Beim Acorn B müssen Funktionen am Ende des Programmes nach dem Befehlswort END oder STOP definiert werden und nicht, wie in unserem Beispiel, am Beginn des Programmes.

### STEP 0

Ersetzen Sie beim Oric in den beiden Programmteilen zur Demonstration von Schleifen:  
IF INKEY\$=" " THEN LET X=2  
durch  
IF KEY\$=" " THEN LET X=1

Beim Dragon ersetzen Sie sie durch:

```
IF INKEY$=" " THEN LET X=1
```

Beim VC-20 und beim Commodore 64 ersetzen Sie:

```
IF INKEY$=" " THEN LET X=2
```

durch

```
GET A$:IF A$=" " THEN LET X=1
```





# Micro-Labyrinth

**Irrgärten üben seit jeher eine große Faszination auf jung und alt aus. Labyrinthspiele für Heimcomputer machen da keine Ausnahme.**

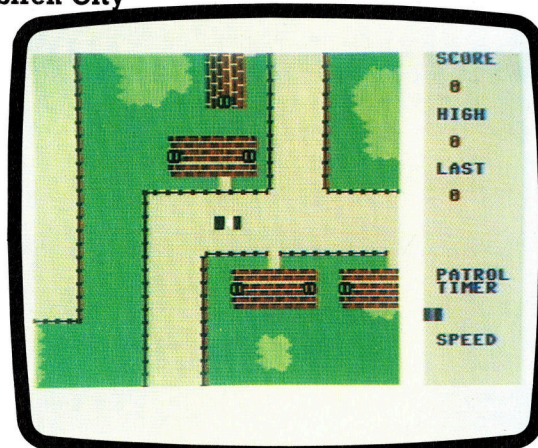
**B**ei Computerspielen sind Labyrinth die Grundlage vieler Spielvarianten, angefangen von einfachen Spielen mit zweidimensionaler Darstellung bis hin zu komplexen dreidimensionalen Irrgärten.

Nicht nur die Labyrinth sind heute durch raffinierte Video- und Toneffekte immer anspruchsvoller geworden, auch die Programmierer haben ihre Fertigkeiten ständig verbessert. Ein Spieler, der nur mal so zum Spaß durch ein solches Labyrinth spazieren möchte, tut gut daran, bestimmte Punkte zu umgehen, um allesfressenden Monstern auszuweichen. Ein Beispiel hierfür ist '3D-Glooper', ein Spiel für den Commodore 64. Der Spieler muß hier bestimmte Bodenmuster des Labyrinths su-

lich dreidimensional. Der Spieler sieht, wie sich die Perspektive seines Blickwinkels entsprechend ändert, wenn er den Steuerknüppel in die gewünschte Richtung bewegt.

Für die Speicherung der Labyrinth-Daten benutzt man zweidimensionale Datentabellen (arrays), die man z. B. mit M\$ (ROW, COLUMN) aufbaut. Jedes Feld dieser Tabelle, das durch Zeile (ROW) und Spalte (COLUMN) gekennzeichnet

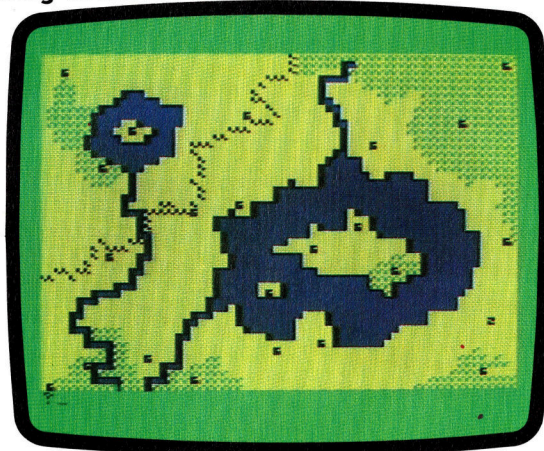
## Siren City



## Siren City

Dies ist ein Spiel für den Commodore 64 mit der weit verbreiteten Darstellung von Flächenausschnitten. Ein Streifenwagen der Polizei patrouilliert in den Straßen einer Stadt. Die Darstellung zeigt Straßenzüge in Form eines Stadtplanes.

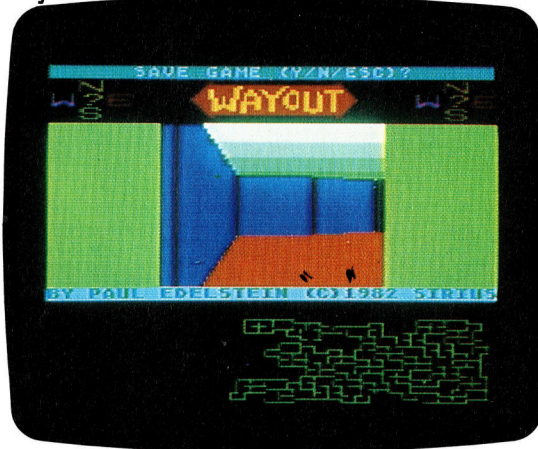
## Ring of Darkness



## Ring of Darkness

Dieses Abenteuerspiel wurde für den Dragon geschrieben. Es enthält als eine der Hauptattraktionen ein dreidimensionales Labyrinth. Fallgruben und Leitern gestalten den Weg des Spielers noch interessanter.

## Way Out



## Way Out

Mit diesem Spiel lassen sich auf einem Spectrum realistische dreidimensionale Darstellungen erzielen. Mit der Bewegung des Steuerknüppels ändert sich der Blickwinkel des Spielers.

chen, belauert von kleinen Monstern, die ihn urplötzlich überfallen und als bildschirmfüllende Ungeheuer in Erscheinung treten. Die Monster tauchen allerdings nicht ganz unerwartet auf, sie kündigen ihr Erscheinen durch einschüchternde Kaugeräusche an.

'Atic Atac' ist ein mit Bewegungsgrafik ausgerüstetes Spiel, das auch auf dem Spectrum läuft. Dargestellt wird eine Verfolgungsjagd, in der der Spieler in drei verschiedenen Rollen auftreten kann. Das Labyrinth besteht aus verschiedenen Ebenen – Fallgruben, Treppen und große Kerker –, die der Spieler im Kampf gegen die Uhr überwinden muß. Die Kerker sind von erschreckenden Kreaturen und anderen obskuren Objekten „bewohnt“.

Ein Programm, das sehr wirklichkeitsnah demonstriert, wie sich ein Spieler fühlt, der seinen Weg durch ein Labyrinth bahnt, heißt 'Way Out'. Die Grafik dieses Programms ist tatsäch-

lich dreidimensional. Der Spieler sieht, wie sich die Perspektive seines Blickwinkels entsprechend ändert, wenn er den Steuerknüppel in die gewünschte Richtung bewegt. Für die Speicherung der Labyrinth-Daten benutzt man zweidimensionale Datentabellen (arrays), die man z. B. mit M\$ (ROW, COLUMN) aufbaut. Jedes Feld dieser Tabelle, das durch Zeile (ROW) und Spalte (COLUMN) gekennzeichnet ist, bestimmt die Eigenschaften des entsprechenden Labyrinth-Abschnittes. So könnten Sie beispielsweise mit einem 4-Bit-String die Richtungen Süd, West, Nord und Ost festlegen, wobei eine Eins bedeutet, daß man in dieser Richtung auf eine Wand trifft, während man bei Null freie Bahn hat. Nehmen wir an, daß der String M\$ (5,6) 1011 enthält. Dies bedeutet, daß das Feld in Zeile fünf und Reihe sechs im Süden, Norden und Osten durch Wände begrenzt ist.



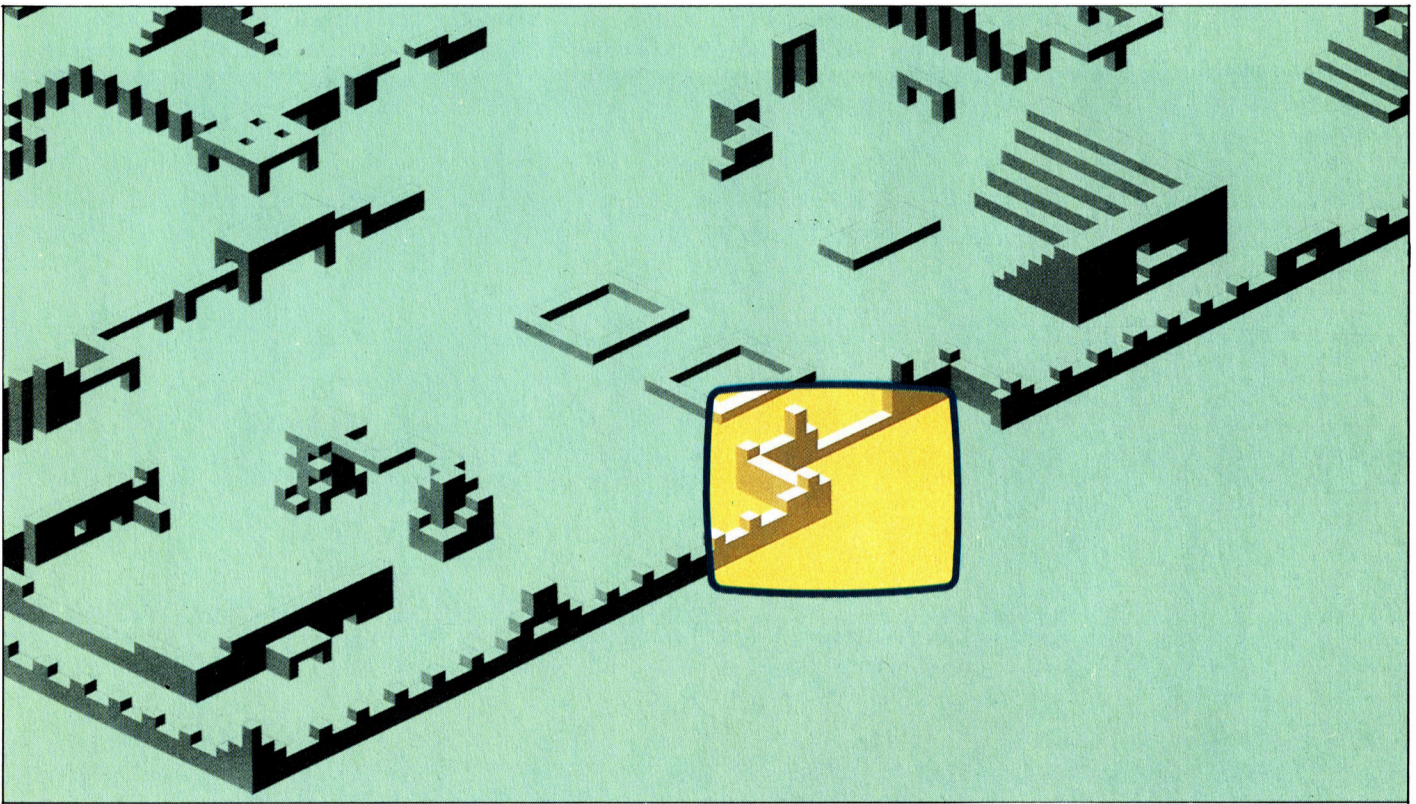


Um den Speicherplatzbedarf zu begrenzen, können die Strings der Datentabellen als 4-Bit-binäre Zahlen betrachtet werden. In unserem Beispiel entspricht demnach der Inhalt 1011 der Variablen (5,6) der Dezimalzahl 11. Zu Beginn haben alle Labyrinth-Abschnitte ihre vier Wände. Man kann sich über jedes beliebige Feld Zutritt zum Labyrinth verschaffen, indem man eine Wand wegläßt. Gleich danach hat man die drei umliegenden Abschnitte zur Auswahl, um sich seinen Weg durch den Irrgarten zu bahnen. Sobald man in das nächste Feld eingedrungen ist, beginnt das Spiel von vorn.

Für die Konstruktion dreidimensionaler Irrgärten gibt es eine besondere Methode, die die Beschriftung der Felder mit binären Zahlen effektiver nutzt. Insgesamt gibt es 16 Möglichkeiten, einen Labyrinth-Raum zu bauen: ohne Wände; Wände an allen vier Seiten; eine Wand (vier Möglichkeiten); Wände auf den gegenüberliegenden Seiten (zwei Möglichkeiten); zwei aneinandergrenzende Wände (vier

befindende Spieler blickt in Richtung Norden. Dreht sich der Spieler links herum, also nach Westen, sieht er auf eine „Nordwand“, weil in dreidimensionalen Darstellungen vorwärts gleichbedeutend mit Norden ist. Um eine Darstellung zu erreichen, die der Blickrichtung des Spielers entspricht, muß der binäre Ausdruck 0100 um eine Stelle nach rechts bewegt werden, damit 0010 (2) entsteht, also eine Nordwand ins Blickfeld des Spielers rückt. Dreht sich der Spieler rechts herum, muß um eine Stelle nach links verschoben werden. Für eine 180°-Drehung müssen zwei Stellen geändert werden. Wichtig ist, daß beim Verschieben kein Bit verlorengeht, daß also das Bit, das rechts „herausfällt“, links wieder hereingeschoben wird und umgekehrt. Sonst ändern sich die Gegebenheiten des Raumes bei jeder Drehung des Spielers und komplizieren das Spiel. Der binäre Ausdruck eines Feldes mit Nord- und Ostwand muß sich von 0011 auf 0110 ändern, wenn sich der Spieler nach rechts

**Bei dem Computerspiel 'Ant Attack' zeigt der Bildschirm immer nur einen Ausschnitt des gesamten Labyrinths. Mit der Bewegung des Spielers bewegt sich auch der Bildschirm und zeigt die erforderlichen Ansichten des Labyrinths.**



Möglichkeiten) und drei aneinandergrenzende Wände (vier Möglichkeiten).

All diese Variationen lassen sich mit einer Binärzahl (dezimal 0 bis 15) ausdrücken. So stellt z. B. 2 (0010) eine Nordwand, 8 (1000) eine Südwand, 1 (0001) eine Ostwand und 4 (0100) eine Westwand dar. Ein einmal festgelegter Abschnitt kann durch Verschieben der Binärzahl dem Weg des Spielers angepaßt werden. Wie dieses Verfahren funktioniert, wollen wir an einem Raum zeigen, der nur über die Westwand (4) verfügt. Der sich darin

dreht, und auf 1100 bei einer 180°-Drehung.

Für das Verschieben binärer Ausdrücke gibt es in der Maschinensprache besondere Befehle für jede Richtung. In BASIC werden 4-Bit-Binärzahlen, die Dezimalzahlen von 0 bis 15 repräsentieren, durch Multiplikation mit 2 und Subtraktion von 15 linksherum gedreht, wenn die Multiplikation einen Wert über 15 liefert. Einfaches Teilen durch 2 führt bei geraden Zahlen zu einer Bewegung nach rechts. Bei ungeraden Zahlen muß vor dem Teilen 15 addiert werden.





# Stück für Stück . . .

**Die Elemente eines Tangram-Puzzles werden mit Hilfe von LOGO dargestellt und zu Figuren geordnet.**

**W**ie bereits anhand praktischer Beispiele erklärt wurde, hat der LOGO-Anwender die Möglichkeit, Prozeduren zu entwickeln, die mehrere verschiedene Befehle beinhalten. Sobald eine Prozedur definiert wurde, kann sie in der gleichen Weise wie die 'Primitives' – das sind die in LOGO bereits festgelegten Befehle – in Programmen eingesetzt werden. Wie diese in verschachtelten Prozeduren aufgerufen und ausgeführt werden, erfahren Sie hier.

Als Beispiel denken Sie an ein einfaches Tangram-Puzzle, das man beliebig zu immer neuen Figuren zusammenfügen kann. Ausgangspunkt dabei ist ein Viereck, das in sieben geometrische Stücke aufgeteilt wurde.

Will man eine Figur wie etwa die eines Hundes bauen, müssen zuerst die Einzelteile in kurzen Prozeduren definiert werden, die anschließend in der HUND-Prozedur zusammengefaßt erscheinen. Um die Turtle vor dem Zeichnen der jeweiligen Teile richtig zu positionieren, werden die Prozeduren MOVE 1 bis MOVE 7 eingegeben.

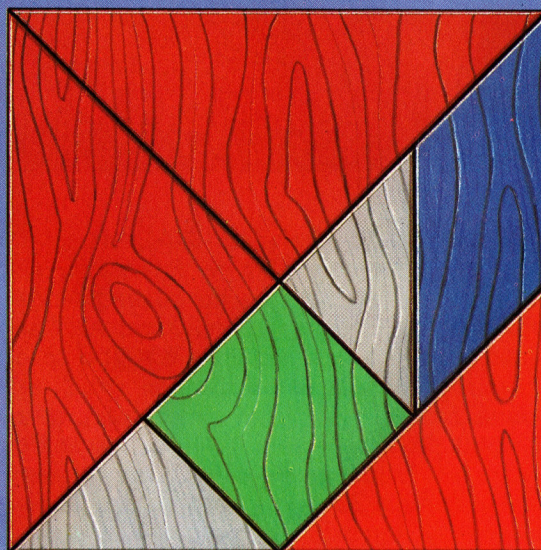
Eine Möglichkeit, die Zeichnung zu erstellen, wäre, alle Anweisungen nacheinander in einer langen Prozedur zu schreiben. Der zweite und programmtechnisch weitaus bessere Weg ist, das „Problem“ in kleine Teile zu zerlegen und diese durch Unterprogramme einzeln zu lösen. Das Hauptprogramm läuft na-

## Geometrische Formen

Das Tangram-Puzzle besteht aus sieben geometrischen Figuren, die sich auf unterschiedliche Arten zusammenfügen lassen. Die nebenstehenden Unterprogramme enthalten die Befehle zum Erstellen der einzelnen Elemente, die im Hauptprogramm, der HUND-Prozedur, aufgerufen werden.

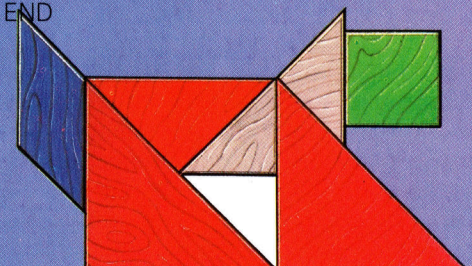
## Tangram-Prozeduren

```
TO RECHTECK
  REPEAT 4 [FD 25 RT 90]
END
TO PAR
  REPEAT 2 [FD 25 RT 45 FD 35 RT 135]
END
TO DR1
  FD 25 RT 135 FD 35 RT 135 FD 25 RT 90
END
TO DR2
  FD 35 RT 135 FD 50 RT 135 FD 35 RT 90
END
TO DR3
  FD 50 RT 135 FD 71 RT 135 FD 50 RT 90
END
```



## HUND-Programm

```
TO HUND
  DR3 MOVE1 PAR MOVE2 TR2 MOVE3
  DR1 MOVE4 DR3 MOVE5 DR1 MOVE6
  RECHTECK MOVE7
END
TO MOVE1
  PU FD 15 LT 45 PD
END
TO MOVE2
  PU RT 45 FD 35 LT 45 BK 35 PD
END
TO MOVE3
  PU LT 45 BK 25 PD
END
TO MOVE4
  PU RT 90 BK 25 PD
END
TO MOVE6
  PU FD 50 RT 45 PD
END
TO MOVE6
  PU FD 25 RT 135 FD 5 LT 90 PD
END
TO MOVE7
  PU LT 90 FD 5 RT 45 BK 25 RT 45
  BK 50 LT 90 BK 50 PD
END
```







türlich nur, wenn die Unterroutinen, die aufgerufen werden sollen, bereits existieren oder wenn zumindest eine vorläufige Version der Prozedur definiert wurde. Zur Verdeutlichung dieses Programmierprozesses dient wieder das HUND-Programm.

Bei dem Beispiel wurde das Hauptprogramm – also die HUND-Prozedur – zuerst geschrieben, wobei die einzelnen Unterroutinen noch nicht definiert waren. Anschließend wurden die Befehle zum Zeichnen der separaten geometrischen Figuren eingegeben und danach erst die Routinen zum Positionieren der Turtle. Sobald eine neue Prozedur fertiggestellt war, wurde überprüft, ob die Teile aneinanderpassen, indem HUND aufgerufen wurde. Das führt natürlich zur Fehlermeldung, so lange nicht alle MOVE-Prozeduren definiert sind. Man kann dadurch jedoch feststellen, ob bis zum Aufruf der MOVE-Prozedur alle Angaben korrekt waren.

Ein weiterer wichtiger Punkt ist, daß bei allen Prozeduren, gleich ob Unter- oder Hauptprozedur, Position und Richtung der Turtle gleich sind. Anfangs- und Endposition der jeweiligen Figuren sind folglich identisch. Dieses Verfahren erleichtert das Kombinieren von einzelnen Teilbildern zu einer komplexen Grafik. In dem aufgezeigten Programm erspart diese Technik zusätzliche Befehle, da die Position der Turtle nicht neu berechnet werden muß. Durch Anwendung dieser Methode ist es unkompliziert, das Programm so zu ändern, daß der ganze Bildschirm mit „HUNDen“ gefüllt wird.

## Die Prozeduren-Verwaltung

Nach abgeschlossener Programmeingabe befindet sich eine umfangreiche Anzahl von Prozeduren im Speicher Ihres Computers. Wie aber werden diese verwaltet? Beim Laden von LOGO wird der Arbeitsspeicher in einzelne Abschnitte (Nodes) zu je fünf Bytes unterteilt. Die Prozeduren werden in diesen Nodes abgelegt und zusätzlich beim Programmablauf mit Variablen oder anderen Funktionen belegt. Die Prozeduren, die sich im Speicher befinden, können mit dem Befehl POTS (Printout Titles) abgerufen werden. Möchte man aber nur eine bestimmte Prozedur abrufen, gibt man PO (Printout) ein, gefolgt von dem Namen des Programms, zum Beispiel PO WUERFEL. ERASE dagegen löscht alle im Speicher abgelegten Programme, ERASE WUERFEL löscht nur diese eine Prozedur. Im Gegensatz zu einigen anderen Programmiersprachen, bei denen der durch Löschen eines Programms freige-wordene Speicherplatz sofort wieder genutzt wird, gibt LOGO die freien Nodes nach dem Löschen nicht unmittelbar frei. Erst wenn der restliche Speicherplatz vollständig belegt ist, überprüft LOGO die Liste der freien Speicher-adressen und stellt anschließend die freie-

wordenen Nodes zur Verfügung. Dieser Prüfvorgang kann in manchen Fällen zu bis zu zwei Sekunden Wartezeit führen.

Damit die Prozeduren nach Abschalten des Computers nicht verlorengehen, können Sie diese auf Cassette oder Diskette abspeichern. Wählt man beispielsweise den Dateinamen FIGUREN, dann sieht der Befehl so aus: SAVE "FIGUREN (Die Anführungsstriche müssen vor dem Dateinamen stehen!). Um Ihr Programm später wieder in den Arbeitsspeicher zu laden, geben Sie READ "FIGUREN ein. Sollten sich weitere Prozeduren im Speicher befinden, werden die neuen einfach hinzugefügt. Allerdings werden alte Prozeduren mit demselben Namen wie die neuen gelöscht und durch die neuen Unterprogramme überschrieben. Weitere Diskettenbefehle sind CATALOG und ERASEFILE. CATALOG gibt eine Liste der gesamten abgelegten Dateien aus, und ERASEFILE "FIGUREN löscht die Datei FIGUREN von der Diskette. Die Befehle für den Recorder-Betrieb stehen im LOGO-Handbuch.

## Abkürzungen

ERASE ER  
PRINTOUT PO  
PRINTOUT TITLES  
POTS

## Übungen

1. Entwickeln Sie Prozeduren für die folgenden Figuren. Zuerst müssen Sie natürlich herausfinden, welche Einzelteile jeweils zu definieren sind.

2. Schreiben Sie ein Programm, das ein Haus auf den Bildschirm zeichnet (Quadrat plus gleichschenkliges Dreieck).  
3. Definieren Sie eine Prozedur, die ein Feld aus fünf × fünf

Quadraten zeichnet.  
4. Überlegen Sie, wie eine Prozedur aussehen müßte, die einen aus Dreiecken zusammengesetzten sechseckigen Stern zeichnet.

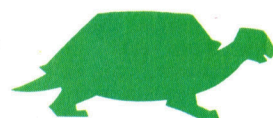


## LOGO-Dialekte

In fast allen LOGO-Versionen muß bei Befehlen wie PO oder ERASE der nachfolgende Dateiname mit Anführungsstrichen eingeleitet werden. Zum Beispiel: PO "WUERFEL oder ERASE "WUERFEL.

Eine gespeicherte Datei (in diesem Fall FIGUREN) wird mit LOAD "FIGUREN geladen.

Die Befehle zum Laden und Speichern der Programme auf Cassette sind im Handbuch Ihres Computers aufgeführt.







# Pinball Wizard

**Mit dem Flipper-Baukasten von Electronic Arts kann die Spielfläche nach eigenen Vorstellungen gestaltet werden. Das Programm stellt einen Meilenstein in der Software-Entwicklung dar.**

**D**ie Microcomputer-Industrie entwickelt sich in rasanter Geschwindigkeit: Neue Produkte überschwemmen den Markt und verwirren die Konsumenten. Tatsächliche Neuentwicklungen, die auf einzigartigen Ideen beruhen und sich durch hervorragende Qualität auszeichnen, sind jedoch rar. Das 'Pinball Construction Set' (PCS) von Bill Budge, im Vertrieb bei Electronic Arts, ist ein gutes Beispiel für perfektes Programmieren.

## 38 Hindernisse

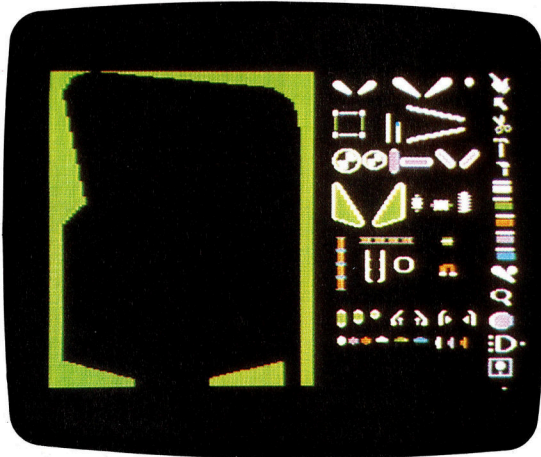
Das PCS-Programm läuft unter anderem auch auf dem 48 KByte Apple II, der mit einem Diskettenlaufwerk und einem Joystick ausgerüstet sein muß. Nach erstem Hinsehen erscheint es, als ob PCS nur recht einfache Funktionen ausführen kann. Es bietet dem Spieler einen unbestückten Spieltisch, ein Ausrüstungsmenü mit 38 unterschiedlichen Hindernissen und ein Werkzeugmenü für die Spielgestaltung.

PCS ist von Konzept und Ausführung her ein besonders anwenderfreundliches Programm. Der Spieler braucht nur eine Diskette ins Laufwerk zu schieben und RETURN zu drücken. Das erste Werkzeug für die Spielgestaltung ist eine „Hand“. Mit ihr wählt der Spieler die Hindernisse aus und bewegt sie zum gewünschten Platz auf dem Spielfeld. All dies geschieht durch Knopfdruck am Joystick.

Bemerkenswert an der Bewegung der Hindernisse sind nicht nur die realistischen Formen der Objekte, sondern besonders die Befehlsauswahl, durch die die Spielregeln eines bestimmten Hindernisses festgelegt werden. Ein einzelner Flipper schwenkt z. B. immer um 45°, zuerst nach vorn und dann zurück.

PCS bietet viele Gestaltungsmöglichkeiten, die man von guten Grafikprogrammen erwartet. So können beispielsweise mit der sogenannten Gummiband-Technik (rubber-banding) Linien, die zwei vorgegebene Linien verbinden, gedehnt und verformt werden. Für die

Das PCS-Baukasten-Programm zeigt auf dem Bildschirm einen leeren Spieltisch. Daneben verschiedene Arten von Hindernissen wie Puffer, Zielscheiben, Überlaufrollen, Ablenkklappen oder Flipper und vieles mehr. Ganz rechts sind die Werkzeuge für die Anordnung der Hindernisse auf dem Spieltisch. Zu den Werkzeugen gehören auch grafische Befehle für die Änderung der Größe, Form und Farbe der Hindernisse, ihres Zusammenspiels und für die Speicherung eines gerade aufgebauten Spiels. Das PCS-Programm erzeugt auch Geräusche und Lichtsignale, die für Flipperautomaten typisch sind. Bei Baukästen ist das Konstruieren und Aufbauen weit interessanter als das Spielen selbst, und so ist es auch mit PCS.



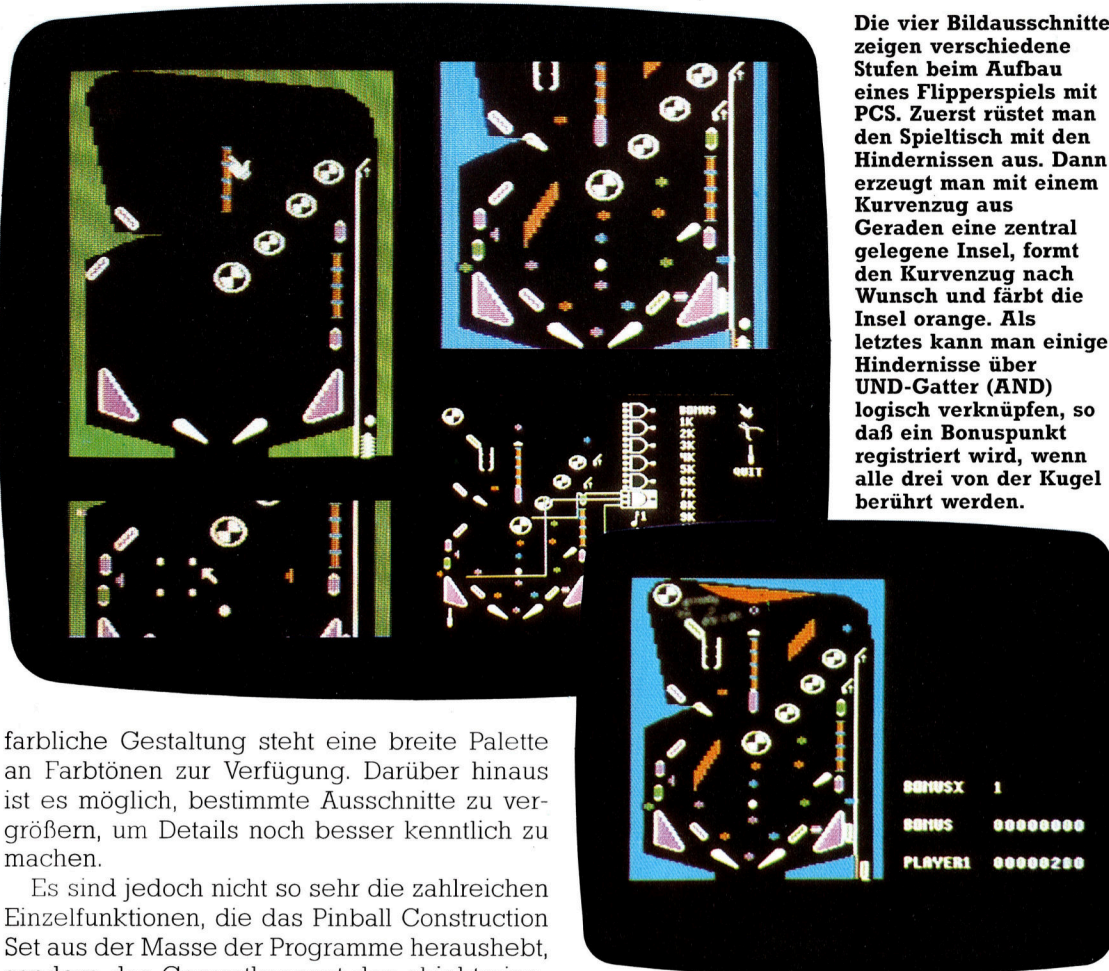
Zuerst wird die Spielfläche nach eigenen Vorstellungen gestaltet. Hierzu stehen 128 Hindernisse zur Verfügung, und für jedes gibt es unendliche Platzierungsmöglichkeiten. Wenn der „Flipper-Bildschirm“ fertig ist, kann das Spielvergnügen beginnen, an dem bis zu vier Spieler teilnehmen können. Im Gegensatz zu mechanischen Flipperspielen steht jedem Spieler aber statt drei Kugeln nur eine zur Verfügung und „Freibälle“ gibt es auch nicht. Ist das Spiel beendet, drücken Sie lediglich die ESCAPE-Taste und befinden sich danach wieder im Menü.







Die vier Bildausschnitte zeigen verschiedene Stufen beim Aufbau eines Flipperspiels mit PCS. Zuerst rüstet man den Spieltisch mit den Hindernissen aus. Dann erzeugt man mit einem Kurvenzug aus Geraden eine zentral gelegene Insel, formt den Kurvenzug nach Wunsch und färbt die Insel orange. Als letztes kann man einige Hindernisse über UND-Gatter (AND) logisch verknüpfen, so daß ein Bonuspunkt registriert wird, wenn alle drei von der Kugel berührt werden.



farbliche Gestaltung steht eine breite Palette an Farbtönen zur Verfügung. Darüber hinaus ist es möglich, bestimmte Ausschnitte zu vergrößern, um Details noch besser kenntlich zu machen.

Es sind jedoch nicht so sehr die zahlreichen Einzelfunktionen, die das Pinball Construction Set aus der Masse der Programme heraushebt, sondern das Gesamtkonzept des objektorientierten Programmierens. Dies bedeutet: Jedes Objekt oder Betriebselement des Programms enthält alle Vorschriften und Gesetze, wie es selbst und im Zusammenspiel mit anderen Objekten funktionieren muß.

### Neue Programmiermethode

Es liegt auf der Hand, daß objektorientiertes Programmieren besonders anwenderfreundlich ist, denn Erfahrungen im Programmieren der Objekte oder Betriebselemente selbst sind nicht erforderlich. Die zur Zeit entwickelten Computer der fünften Generation werden diese Programmiermethode fast ausschließlich anwenden. Sie genießt den Ruf, nach Einführung der hochentwickelten Programmiersprachen in den fünfziger Jahren, das bedeutendste Ereignis der gegenwärtigen Softwareforschung zu sein.

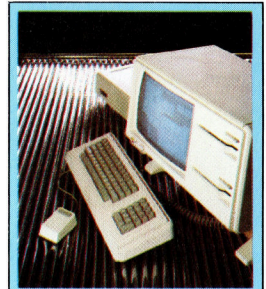
Die meisten Heimcomputer verfügen über ausreichende Speicherkapazitäten und Rechenleistungen, so daß eine Erhöhung von Speicherumfang und Leistung hauptsächlich der Anwenderfreundlichkeit zugute kommen wird. Eine der hervorstechendsten Eigenschaften des PCS-Programms ist, daß es für die Verwirklichung der vielen Benutzervorteile nur 48 KByte benötigt.

Für Spiele und generell für Grafikprogramme läßt sich objektorientiertes Program-

mieren relativ einfach anwenden. Etwas schwieriger wird es bei Programmen für den Geschäftsbereich. Diese Art Software, wie z. B. Visicalc und Supercalc für die Tabellenkalkulation, sind zumindest teilweise objektorientiert geschrieben, obwohl hier zur Kommunikation mit dem Anwender wenig Grafik eingesetzt wird. „Baukasten-Programme“ gibt es übrigens auch für andere Spiele.

Ein weiteres Beispiel für besondere Anwenderfreundlichkeit ist der ‚Lisa‘ von Apple. Für die Bewegung des Bildschirmzeigers zur Programmauswahl wird eine Maus verwendet. Anwählbare Programme oder Funktionen werden durch grafische Symbole dargestellt. Textverarbeitung wird z. B. durch ein leeres Blatt Papier, das Grafikprogramm durch ein kariertes Blatt dargestellt.

Am meisten beeindruckt Lisa durch die Art des Datenaustausches zwischen den Programmen. Das hierzu erforderliche Programm wird durch ein Symbol aufgerufen, das einem Clipboard oder Klammerbrett gleicht. Soll ein Ausschnitt einer Tabellenkalkulation grafisch dargestellt werden, braucht man nur das entsprechende Bildschirmfenster festzulegen, das Fenster oder den Ausschnitt auf dem Clipboard abzulegen und dann zum Plotprogramm weiterzureichen, um auf diese Weise den Ausdruck zu erhalten.



### Lichtblick für Anwender

Das Flipper-Baukastenprogramm PCS ist nicht nur faszinierend und lehrreich, sondern ein hervorragendes Beispiel für anwender- oder objektorientiertes Programmieren. Beim normalen Programmieren sind Daten und Manipulationen streng voneinander getrennt, denn zuerst wird hier die Datenstruktur festgelegt. Beim sogenannten objektorientierten Programmieren dagegen sind Daten und zugehörige Manipulationen fest miteinander verknüpft. Bewegt man z. B. das Flippersymbol auf dem Spieltisch, so bewegen sich mit ihm nicht nur die Daten für seine Form und Farbe, sondern auch Programmroutinen, die den Flipper als aktives Hindernis erscheinen lassen. Objektorientiertes Programmieren ist besonders gut in Verbindung mit grafischen Darstellungen anwendbar.





# Detektivarbeit

**Bei der Datenübertragung von einem Rechner zum anderen besteht die Gefahr, daß sich Fehler einschleichen. Sie können mit Hilfe von ‚Hamming-Codes‘ entdeckt und beseitigt werden.**

**G**eschichten von spektakulären Computerfehlern, z. B. von dem Computer, der ein und denselben Brief 500mal an den Empfänger geschickt hat, sind hinlänglich bekannt. Der Grund liegt hier natürlich nicht im Gerät, sondern in menschlichem Fehlverhalten. Der Computer hat lediglich die Wirkung eines Verstärkers – aus einem simplen Tippfehler wird dann der berühmte „Elefant“.

Es gibt allerdings auch Computerfehler, die nicht dem Menschen zuzuschreiben sind – allgemein als „Bit-Fehler“ bekannt. Ein solcher Fehler entsteht, wenn ein einzelnes Bit innerhalb einer Datenfolge „umkippt“, also von 1 zu 0 wird oder umgekehrt. Die Ursache kann in fehlerhafter Hardware liegen, z. B. in einem fehlerhaften RAM-Chip. Dies ist auch der Grund, warum viele Heimcomputer erst einmal eine „Diagnose“-Routine durchlaufen, sobald sie eingeschaltet werden.

Die meisten Bitfehler jedoch sind nicht hardwarebedingt. Eine sommerliche Hitzewelle kann dazu führen, daß irgendwo im Computer ein Hitzestau entsteht. Er verursacht zwar kaum einen dauerhaften Schaden, kann aber der Grund dafür sein, daß aus einem A auf dem Bildschirm plötzlich ein B wird. Ist das fehlerhafte Bit Teil eines wichtigen Zeigers, kann dies allerdings dazu führen, daß das Programm „abstürzt“ und der Computer neu aktiviert werden muß.

Auch Sonnenflecken können Bitfehler verursachen, wenn subatomare Partikel in die Atmosphäre eindringen und den Elektronenfluß in den Schaltkreisen des Rechners überlagern. In militärischen Systemen, industriellen Steuerungen, wissenschaftlichen Experimenten oder im internationalen Bankwesen könnten Bitfehler katastrophale Folgen haben. Deshalb wurden viele Methoden zum Aufspüren von Bitfehlern entwickelt. Die einfachste ist die Pa-

ritätsprüfung (beschrieben im folgenden Artikel). Eine Alternative dazu ist die ‚Kontrollsumme‘. Von dieser Methode wird weitgehend Gebrauch gemacht, wenn Daten auf Magnetband oder Diskette geschrieben werden. Die Kontrollsumme steht am Ende eines Datenblocks. Ein Datenblock besteht aus 128 Bytes; dabei hat jedes Byte einen Wert zwischen 0 und 255. Die gespeicherten Daten werden addiert und durch 256 geteilt. Der bei dieser Rechnung verbleibende „Rest“ stellt die ‚Kontrollsumme‘ dar. Ein Beispiel:

Daten 114, 67, 83 ... (121 weitere Werte)  
... 36, 154, 198  
Gesamtsumme der 137 Bytes = 16 673  
Dividiert durch 256 = 65, Rest 33  
Kontrollsumme daher = 33

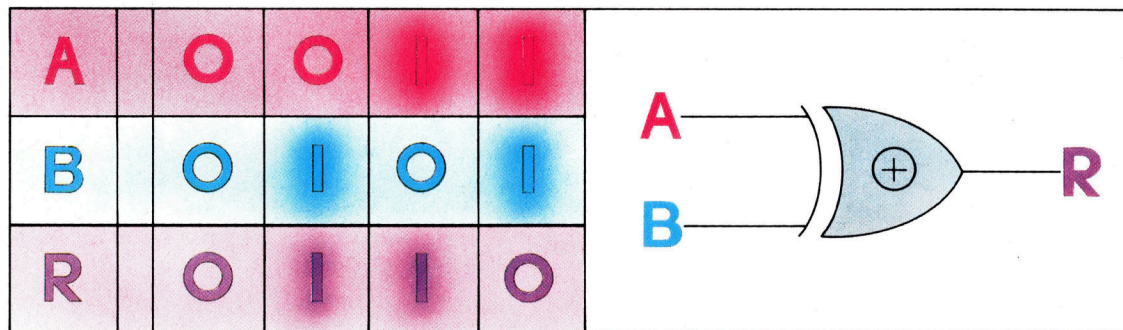
Die Kontrollsumme wird als 128. Byte geschrieben. Wenn der Rechner einen Datenblock liest, führt er eine eigene Kontrollsummenrechnung durch und vergleicht das Ergebnis mit der vom letzten Byte repräsentierten Kontrollsumme. Differieren die beiden Zahlen, weiß der Rechner, daß sich in die Übertragung ein Bitfehler eingeschlichen hat.

Beide Methoden sagen dem Computer nicht, welches Bit fehlerhaft ist. Taucht der Fehler nur beim Übertragen auf, kann der empfangende Computer veranlassen, daß ein bestimmtes Byte oder ein ganzer Byte-Block noch einmal übertragen wird.

## Der Hamming-Code

Für die Anwendungen, bei denen unter keinen Umständen Fehler auftreten dürfen, mußte eine Methode gefunden werden, die Fehler sowohl entdeckt als auch korrigiert. Eine solche Methode sind die sogenannten ‚Hamming-

Ein einfaches Exklusiv-Oder-Gatter (kurz Ex-Oder) hat zwei Eingänge und einen Ausgang. Liegt auf beiden Eingängen ein 0-Signal, erscheint am Ausgang ebenfalls ein 0-Signal. Liegt an einem der beiden Eingänge ein 1-Signal, ist auch der Ausgang 1, aber nicht, wenn an beiden Eingängen zugleich 1 anliegt. Dieser letzte Zustand unterscheidet das Ex-Oder vom Oder-Gatter. Der Vorgang läßt sich auch durch eine Wertetabelle darstellen. Ein Ex-Oder mit mehr als zwei Eingängen ergibt ein 1-Signal als Ausgang, wenn eingangsseitig eine ungerade Anzahl von 1-Signalen anliegt. Gatter dieser Art werden dazu verwendet, Paritäts- oder Prüfbits zu erzeugen.







Codes', benannt nach ihrem Erfinder R. W. Hamming von der amerikanischen Firma Bell Telephone Laboratories.

Jede Fehlerkonstruktion arbeitet nach dem Prinzip der Redundanz (was soviel wie „Überfluß“ bedeutet). Die menschliche Sprache enthält einen hohen Grad an überflüssigen, für die Information nicht wichtigen Elementen. In Schriftstücken lassen sich unleserliche Passagen oft aus dem inhaltlichen Zusammenhang heraus rekonstruieren. Im Funkverkehr werden statt der Buchstaben A, B, C usw. die Worte Alpha, Bravo und Charlie benutzt – eine spezielle Redundanz also –, um Verständigungsfehlern vorzubeugen.

Nehmen wir als Beispiel ein Wort von  $x$  Bit Länge. Es soll aus  $y$  Datenbits und  $z$  Redundanzbits bestehen ( $x = y + z$ ). Hamming-Codes erfordern einen hohen Wert für  $z$ . Man geht davon aus, daß ein Bitfehler in jedem Bit auftreten kann, auch in einem Redundanzbit. Ein Fehler tritt in einem Wort mit der Wahrscheinlichkeit von eins zu einer Million auf. Beim Empfang der Daten am anderen Übertragungsende bestehen folglich  $x + 1$  Möglichkeiten. Entweder gibt es überhaupt keine Fehler oder das erste Datenbit ist fehlerhaft und so weiter bis zum  $x$ -ten Bit. Mit  $z$  Redundanzbits können  $2^z$  Möglichkeiten repräsentiert werden, so daß für ein Wort, das die Prüfung auf einen Bitfehler bestehen soll, gilt:

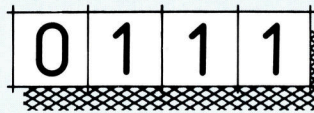
$$2^z = y + z + 1$$

Ist  $y = 7$  (bei ASCII-Codes), muß  $z = 4$  sein. Ist  $y = 4$ , muß  $z = 3$  sein. Bei  $y = 16$  jedoch muß der Wert für  $z$  nur auf 5 erhöht werden. Hamming-Codes sind also für längere Wörter sehr viel besser geeignet als für kürzere.

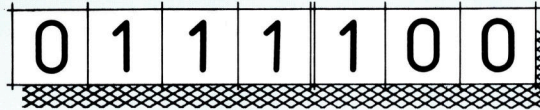
Im Hamming-Code wirkt jedes Redundanzbit als Prüfbit für ‚gerade Parität‘, und zwar abhängig von der Bitkombination im Wort. Tritt bei der Datenübertragung ein Bitfehler auf, wird eines (oder werden mehrere) der Prüfbits unrichtig sein und durch seine (ihre) Stellung innerhalb der Bitkombination auf das fehlerhafte Bit hinweisen. Der empfangende Computer ist damit in der Lage, das „entdeckte“ Fehlerbit zurückzuweisen.

Der Schlüssel zur Funktionsweise der Hamming-Codes liegt in der unterschiedlichen Kombination der zu prüfenden Bits. Die Gesamtzahl der Bits wird in unterschiedliche, sich überlappende Sätze aufgeteilt, und zwar so, daß sich zwei Bits nicht in der gleichen Kombination wiederholen. Der empfangende Computer prüft die gleichen Sätze auf Parität, die der sendende Computer prüfte, um den Hamming-Code zu erzeugen. Kippt beim Übertragen eines der Bits (einschließlich der Hamming-Bits), wird einer oder werden mehrere dieser Sätze die Paritätsprüfung nicht bestehen. Aus der Kombination der nicht bestandenen Paritätsprüfungen ergibt sich dann der Hinweis auf das fehlerhafte Bit.

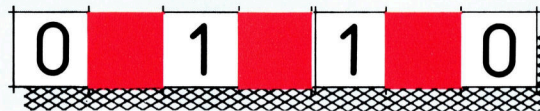
## Wie der Hamming-Code arbeitet



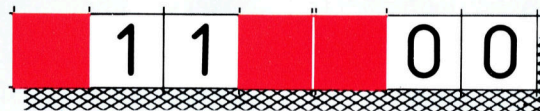
Diese vier Bits sollen übertragen werden.



Der Computer fügt drei weitere Bits als Hamming-Code hinzu. Sie sind nach einem bestimmten Muster angeordnet:



In diesem Vierermuster muß die 1 in gerader Anzahl vorhanden sein.



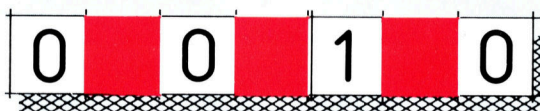
Auch in dieser Anordnung muß die 1 in gerader Anzahl erscheinen.



Eine gerade Anzahl von 1-Signalen muß auch dieses Muster zeigen. Der Computer muß drei Gleichungen hintereinander lösen.



In diesem Muster wird unterstellt, daß das dritte Bit von links von 1 auf 0 gekippt ist.



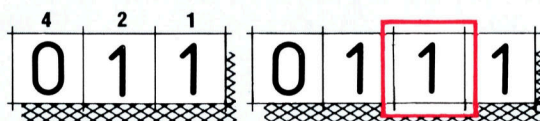
Führt der Computer nun die erste der drei Prüfungen durch, wird diese nicht bestanden, weil die 1 in ungerader Anzahl erscheint. Das zeigt zwar einen Fehler, nicht aber das fehlerhafte Bit.



Auch die zweite Prüfung wird nicht bestanden.



Die dritte Prüfung jedoch wird bestanden, weil die 1 in gerader Anzahl erscheint.



**Die Methode ist auch wirksam, wenn der Fehler bei einem Hamming-Bit auftritt. Schlagen alle drei Prüfungen fehl, erhält man binär 111, was anzeigt, daß das äußerste rechte Bit fehlerhaft ist. Werden alle drei Prüfungen bestanden, ist die Übertragung fehlerfrei.**

So entsteht der Hinweis auf das fehlerhafte Bit. Bezeichnet man eine nicht bestandene Prüfung mit 1 und eine bestandene mit 0 und schreibt die Ergebnisse in umgekehrter Reihenfolge auf, erhält man eine binäre 3 als Hinweis darauf, daß das dritte Bit das Fehlerbit ist und von 0 auf 1 zurückgestellt werden muß.





# Auf Fehlersuche

**„Gerade Parität“ stellt sicher, daß die Anzahl der 1-Bits in einem Byte immer gerade ist. Dies erleichtert das Erkennen von Übertragungsfehlern, hilft aber nicht bei der Suche nach der Position des „Bitfehlers“.**

**D**igitalcomputer zeichnen sich gegenüber Analogrechnern dadurch aus, daß sich Fehler oder Ungenauigkeiten, wie sie in elektrischen Stromkreisen auftreten können, beim Durchlaufen der Stromkreise nicht „aufschaukeln“. Bei der Fernübertragung von Daten, sei es über eine serielle Schnittstelle und zwei Drähte oder über die Fernsprechleitung, kann das elektrische „Hintergrundrauschen“ in der Leitung jedoch dazu führen, daß ein Bit von 1 auf 0 oder umgekehrt kippt. Normalerweise hat der empfangende Computer keine Möglichkeit, Übertragungsfehler zu erkennen. Er wird fehlerhafte Daten als korrekt annehmen.

Wie sich ein Bitfehler auswirken kann, zeigt nachfolgendes Beispiel:

Übertragen wird [] 1010001 (ASCII-Code für Q).

Empfangen wird [] 1000001 (ASCII-Code für A).

Ein Übertragungsfehler dieser Art ist im besten Fall „nur“ ein Ärgernis, kann aber auch katastrophale Folgen haben. ASCII-Codes arbeiten mit Werten bis zu 127, für die nur sieben Bits benötigt werden. Das achte Bit steht als „Paritätsbit“ zur Verfügung. Dieses MSB (Most Significant Bit) prüft, ob ein Übertragungsfehler vorliegt. Es steht in der hier verwendeten Schreibweise in eckigen Klammern vor den eigentlichen Datenbits.

Das Paritätsbit kann auf „gerade“ oder „ungerade“ Parität prüfen. Hier geht es um die gerade Parität. Sie ist gegeben, wenn das Paritätsbit so eingestellt ist, daß die Gesamtzahl der 1-Bits innerhalb eines Bytes immer gerade ist. Die ASCII-Codes für A und Q sehen bei gerader Parität so aus:

[0]1000001 für A  
[1]1010001 für Q.

Der A-Code enthält zwei 1-Bits; wird das Paritätsbit, auch Prüfbit genannt, als 0 gesetzt, ist die Gesamtzahl aller 1-Bits gerade. Im Q-Code dagegen finden sich drei 1-Bits; um eine gerade Anzahl von 1-Bits zu erhalten, wird das Prüfbit als 1 gesetzt, so daß dieses Byte jetzt vier 1-Bits enthält.

Was geschieht nun bei einem Übertragungsfehler wie im obigen Beispiel? Empfangen wurde:

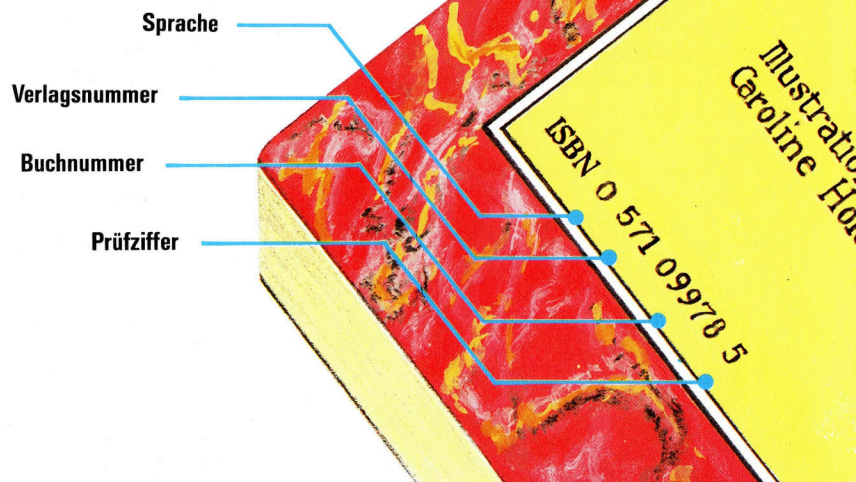
[1]1000001 (fehlerhafter ASCII-Code für Q).

Wird dieses Byte auf Parität geprüft (entweder durch die Software oder eine spezielle Hardware), stellt sich heraus, daß die Anzahl der 1-Bits ungerade und das Byte demnach fehlerhaft ist. Es kann zurückgewiesen werden. Diese Paritätsprüfung ist selbst dann wirksam, wenn das Prüfbit beim Übertragen kippt.

Betrachten Sie die in Ihrem Computer verwendeten ASCII-Codes. Sie stellen wahrscheinlich fest, daß das achte zwar tatsächlich verwendet wird, aber nicht als Prüfbit. Da Fehler in der internen Datenübertragung sehr selten sind, wird dieses Bit dem Computer für ein zusätzliches Symbol (meist ein Grafiksymbol) zur Verfügung gestellt. Die Paritätsprüfung wird normalerweise nur bei der Datenübertragung oder dann verwendet, wenn Daten auf eine magnetische Oberfläche (Cassette oder Diskette) geschrieben werden, weil diese Vorgänge anfällig für Übertragungsfehler sind.

Die Paritätsprüfung ist eine einfache, aber wirkungsvolle Methode zum Aufspüren von Übertragungsfehlern. Sie findet aber nicht das eigentliche Fehlerbit heraus, und der empfangende Computer ist nicht in der Lage, den Fehler zu korrigieren. Schwerer wiegt jedoch, daß ein Byte, das zwei Fehlerbits enthält, fälschlicherweise als korrekt angenommen werden kann. Ein Vorteil bleibt jedoch erhalten: Hat der empfangende Computer erst einmal festgestellt, daß ein Übertragungsfehler stattgefunden hat, kann er eine Fehlermeldung senden und das Programm veranlassen, das fehlerhafte Byte noch einmal zu übertragen. Höherentwickelte Methoden, die ein fehlerhaftes Bit genau identifizieren, sind der „Hamming-Code“ und die „Kontrollsumme“.

Die letzte Stelle in einer Internationalen Standard-Buch-Nummer (ISBN) ist eine Prüfziffer, vergleichbar mit dem Paritätsbit in einem Computer. Multiplizieren Sie die erste Ziffer der ISBN (0) mit 10, die zweite (5) mit 9 usw. und zählen Sie die Ergebnisse zusammen. Sie werden feststellen, daß die Prüfziffer so gesetzt wurde, daß das Ergebnis genau durch 11 teilbar ist.





# Fachwörter auf einen Blick

## **Array**

Bereich von Datenfeldern, die gleiche Eigenschaften haben

## **Bitfehler**

Übertragungsfehler, die dadurch entstehen, daß ein oder mehrere Bits von 0 auf 1 oder von 1 auf 0 „umkippen“

## **Exklusiv-Oder-Gatter**

Diese Gatter werden eingesetzt, um Prüfbits zu erzeugen; typisch ist: bei Eingabe von 0 und 0 ist die Ausgabe 0, bei Eingabe von 0 und 1 ergibt sich 1 und bei Eingabe von 1 und 1 ist die Ausgabe wiederum 0

## **Nodes**

Der Arbeitsspeicher ist bei LOGO in Bereiche zu je 5 Bytes unterteilt; diese Bereiche heißen auch Nodes

## **Optokoppler**

(Optoisolator) Baustein, der Daten auf Lichtstrahlen moduliert; durch den Einsatz eines Optokopplers kann erreicht werden, daß ein Datenverarbeitungssystem optisch zwar mit der Datenquelle gekoppelt, elektronisch jedoch von der Datenquelle isoliert ist

## **Operations-Code**

Ein Teil der Anweisung einer Maschinensprache oder Assembler, in welchem die auszuführende Operation gekennzeichnet wird

## **PEARL**

(Process and Experiment Automat in Realtime Language) Höhere, von BASIC abgeleitete Programmiersprache für die Automatisierung von Prozeßsteuerungen

## **Polarisationsfilter**

Ein solcher Filter dient dazu, die regellosen, natürlichen Schwingungen des Lichts (Transversalschwingungen) zu „ordnen“, so daß eine einheitliche Schwingungsrichtung (polarisiertes Licht) erzielt wird

## **Positionierzeit**

heißt der Zeitraum, den der Lese- bzw. Schreibkopf eines Diskettenlaufwerks benötigt, um auf einer bestimmten Spur anzukommen, bevor der eigentliche Lese- bzw. Schreibvorgang beginnt

## **Querysprachen**

ermöglichen die Kommunikation mit Datenbanken; mit ihren leicht erlernbaren Kommandos, die keine Programmierkenntnisse voraussetzen, lassen sich Daten abrufen, einfügen, ändern oder löschen



## **Redundanz**

Bezeichnung für das Vorhandensein von weglassbaren Elementen in einer Nachricht, die keine zusätzliche Information liefern, sondern lediglich die beabsichtigte Grundinformation stützen

## **Syntaxfehler**

(syntax error) Fehler bei der Eingabe eines Programms, wenn die formellen Regeln nicht eingehalten wurden; beispielsweise vergessene Leerstelle oder Semikolon

## **Timesharing**

Verschiedene Anwender arbeiten gleichzeitig über ein Netzwerk am selben Zentralcomputer; dabei wird abwechselnd allen Teilnehmern Verarbeitungszeit zur Verfügung gestellt; aufgrund der hohen Arbeitsge-

schwindigkeit des Rechners entsteht für den Benutzer der Eindruck, daß der Computer ohne Pause nur für ihn allein tätig ist

## **Turtle**

(englisch für Schildkröte), auch Igel genannt; kleine Bildschirmfigur bei LOGO-Grafiken; sie bewegt sich über den Screen und zeichnet dabei ihren Weg

## **Virtueller Speicher**

nur fiktiv vorhanden; durch automatischen Austausch von Daten zwischen Arbeitsspeicher und externem Massenspeicher beim Programmablauf befinden sich im Arbeitsspeicher immer nur die Daten, die gerade verarbeitet werden; so erscheint die Speicherkapazität größer zu sein, als sie tatsächlich ist

## **Warteschlange**

Daten, die in einer geordneten Reihenfolge auf ihre Verarbeitung in der Zentraleinheit warten

## **Zeichenfolge**

(string) Kette von codierten alphanumerischen Zeichen

## **Zweidimensionales Datenfeld**

Die Daten werden nach zwei unterschiedlichen Kriterien geordnet und zueinander in Beziehung gesetzt; das Feld besteht aus mindestens einer Spalte und einer Zeile, die sich kreuzen; das Feld dehnt sich in zwei Richtungen („Dimensionen“) aus

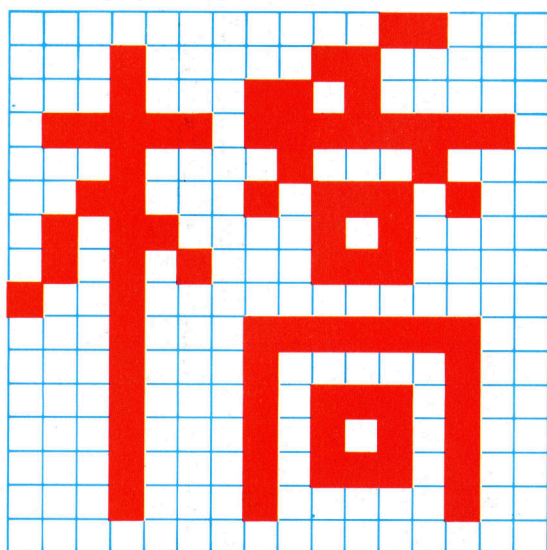
## **Bildnachweise**

282, 284, 293, 300, 305: Ian McKinnell  
282: Bob Freeman  
285: Tony Lodge  
286: Zeff / Whelan  
287, 305: Chris Stevens  
295: Kai Choi  
295, 303: Liz Dixon  
302, 306: Kevin Jones  
304: Image Maker  
Fachwörter: David Higham



+++ Vorschau +++ Vorschau +++ Vorschau +++

# computer kurs Heft 12



橋

## Schrift- zeichen

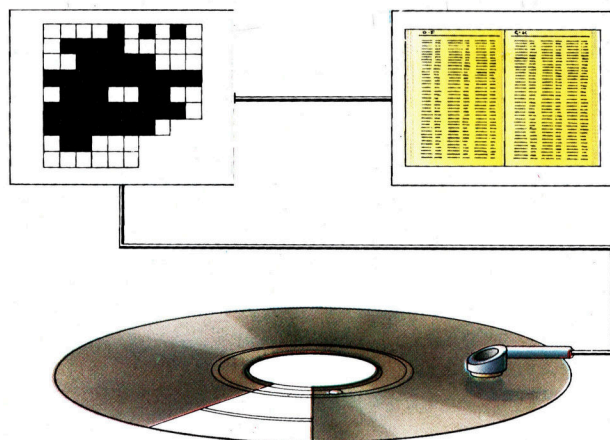
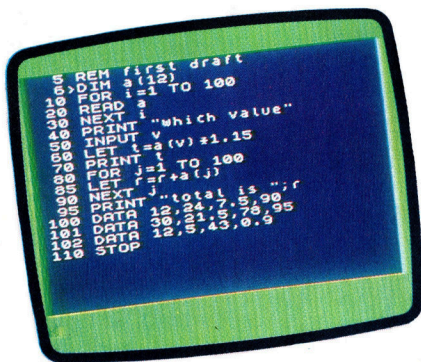
lassen sich auch  
mit dem Com-  
puter „schrei-  
ben“. Hier das  
Wort  
„Brücke“.

## Daten- aufbereitung

Fehler beim Eintip-  
pen kommen häufig  
vor. Der „Editor“  
hilft beim Korri-  
gieren.

## Der Sinclair QL

gilt als „Superstar“.  
Was leistet er  
wirklich?



## Betriebssystem

Den Platz auf einer Diskette zu verwalten und sich  
zu merken, wo welche Informationen gespeichert  
sind, ist die Aufgabe des DOS.



+++ Weiter geht's mit BASIC und LOGO

+++ Herman Hollerith +++ Tips für die

Praxis +++ Verkehrscomputer +++ Er-

weiterungen für den ZX +++ Gray-Code

contra Binärsystem +++ Fernspiele +++

