

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

# computer kurs

Heft **16**

Rechner in der Schule

Das Zeichengenie

Spracherkennungs-Systeme

Klang und Farbdarstellung

Maschinencode

Memotech MTX 512

Ein wöchentliches Sammelwerk

# computer kurs

## Heft 16

### Inhalt

#### Computer Welt

**Schulcomputer** 421

Der Rechner im Klassenzimmer

**Norbert Wiener** 432

Ein amerikanischer Kybemetiker

#### Hardware

**Memotech MTX 512** 424

Ausgereift, leistungsfähig und formschön

#### BASIC 16

**Mehr über Dateien** 426

#### Software

**Spielend programmiert** 430

Anwendungsgeneratoren

**Aufregende Spiele** 441

Simulation zum Lernen und Unterhalten

**Die Stimme seines Herrn** 442

Spracherkennungssysteme für Microrechner

#### Peripherie

**Das Zeichengenie** 433

Ein Grafpad für den Acorn B

#### Tips für die Praxis

**Sinfonie, Lichtorgel** 436

Oric und Spectrum im Einsatz

**Einfach überspielen!** 448

#### LOGO 16

**Käfer bei der Arbeit** 438

Animationseffekte durch Sprites

#### Bits und Bytes

**Punkte, Punkte** 444

Auswahlkriterien für den Druckerkauf

**Maschinensprache** 446

Grundlagen dieses schnellen Programmiercodes

#### Fachwörter von A—Z

### 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 wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

#### 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 Überweisungskopie Ihre vollständige Anschrift gut leserlich enthalten.**

#### SAMMELORDNER

Sie können den 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

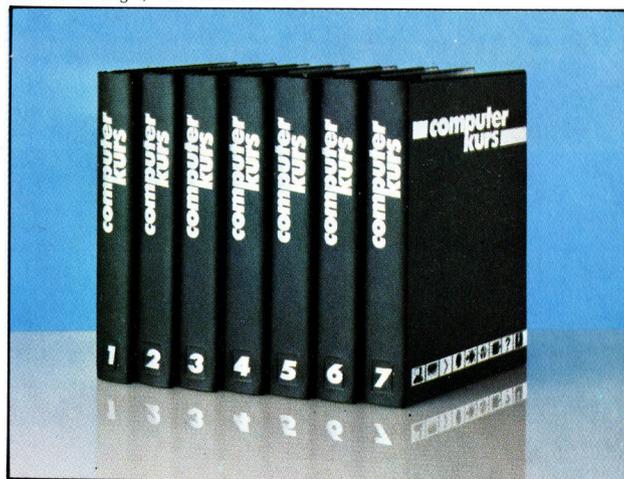
**Schweiz:** Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

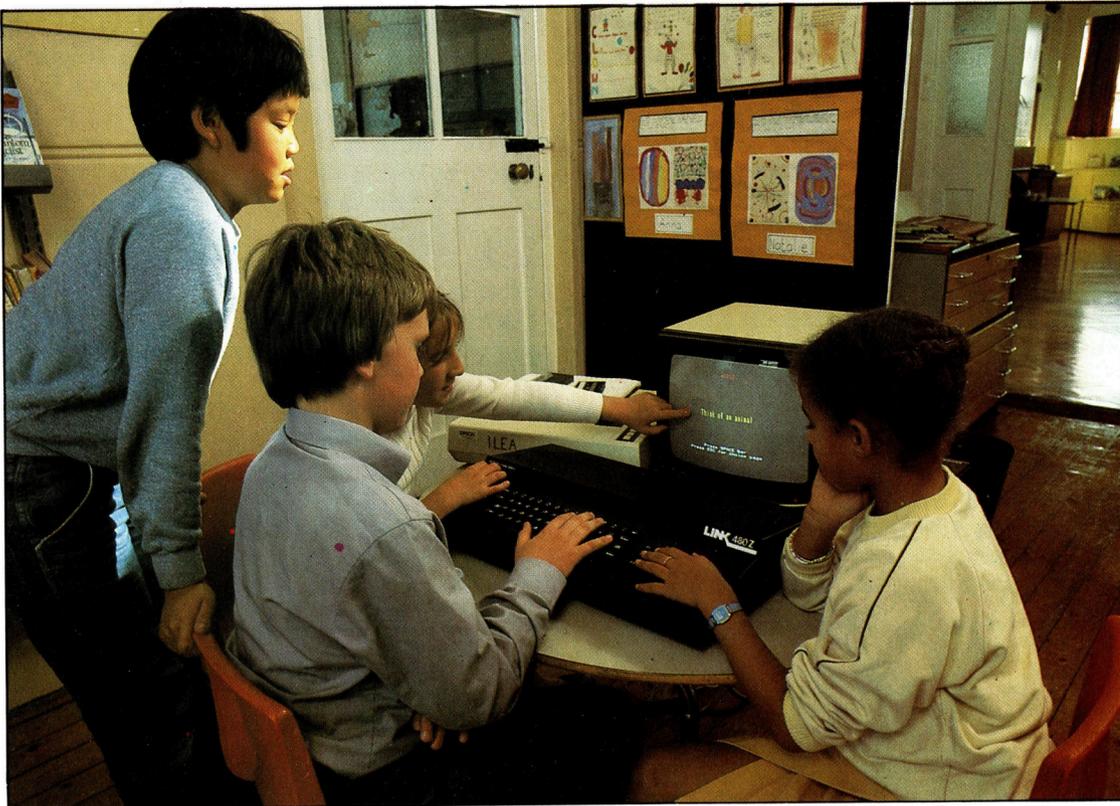
#### INHALTSVERZEICHNIS

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, Uta Brandl (Layout), Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

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





England ist führend beim „Lernen mit dem Rechner“. Schon in den ersten Schulklassen werden die Kinder mit dem Computer bekanntgemacht. Ziel ist aber nicht die Gewöhnung an neue Technologien, sondern der Einsatz der Geräte als Unterrichtshilfe für die verschiedensten Fächer, von der Biologie bis zum Vokabellernen. Computer sind geduldige Lehrmeister, die sich der Geschwindigkeit ihrer Schützlinge ohne Schwierigkeiten anpassen.

# Schulcomputer

**Die Auswahl an nützlichen Lehr- und Lernprogrammen wird laufend größer. Das Computerzeitalter hat mittlerweile auch im Klassenzimmer Einzug gehalten.**

Von drei deutschen Gymnasien sind heute schon zwei mit Computern ausgerüstet – aber der Siegeszug des Rechners macht auch vor anderen Schulzweigen nicht Halt: Lehrer, Eltern und Schüler verlangen immer dringender nach Computern als Lehr- und Lernhilfe – nicht allein fürs Rechnen, auch die Unterstützung schwächerer Schüler oder die Unterrichtung in Fremdsprachen traut man dem „Kollegen Computer“ heute zu.

An Lehrprogrammen herrscht zwar kein Mangel, oft sind sie jedoch noch nicht ausgereift: Nach einem Programm, in dem der Unterrichtsstoff didaktisch gut aufbereitet ist, muß man lange suchen.

## Kindgerechte Programme

Programmierer verfügen selten über die pädagogischen Erfahrungen eines Lehrers, diese machen dafür häufig grobe Fehler beim Programmieren. Ein weiterer Schwachpunkt bei der von begeisterten Lehrern „handgestrickten“ Software ist oft die Dokumentation: Das Programm ist ohne ausführliche Anleitung und Erklärung wenig nützlich. Manchmal fehlen

selbst die einfachsten Hilfen. Woher soll der Anfänger wissen, daß ohne den Befehl „LOAD“ sein neues Programm nicht von der Cassette in den Rechner gelangt?

Ein gutes Programm muß nicht nur genau erklärt sein, es sollte auch mit jedem möglichen Fehler des Anfängers zurechtkommen. Es reicht nicht aus, die Software so weit zu perfektionieren, daß beim Drücken einer bestimmten Taste das Richtige passiert – auch eine Fehlbedienung sollte kein Unglück oder gar einen „Absturz“ des Systems heraufbeschwören. Kinder sind schnell enttäuscht, und die schwere Aufgabe des Programmierers besteht darin, ihnen durch „narrensichere“ Programme nicht den Spaß am Rechner zu verderben.

Trotz dieser Erschwernisse nimmt die Anzahl brauchbarer Lernprogramme stetig zu. Der Computer kann ein hervorragendes Lehr- und Lernmittel sein, das sich für die unterschiedlichsten Wissensgebiete gleichermaßen gut eignet. In Verbindung mit hochwertiger Software sind Kinder vom Rechner oft fasziniert und übertragen ihre Begeisterung auf den Lernehalt – als Motivation ist dies nicht zu unterschätzen!



Die meisten Lernprogramme verfahren nach dem Motto „Einüben und Ausprobieren“. Oft wird ein Problem beispielhaft vorgeführt, danach geht es an das Lösen ähnlicher Aufgaben. Daneben überwacht das Programm die Leistung des Schülers, lobt bei richtigen Antworten und macht bei falschen mit einem freundlichen „Versuch's noch einmal“ neuen Mut.

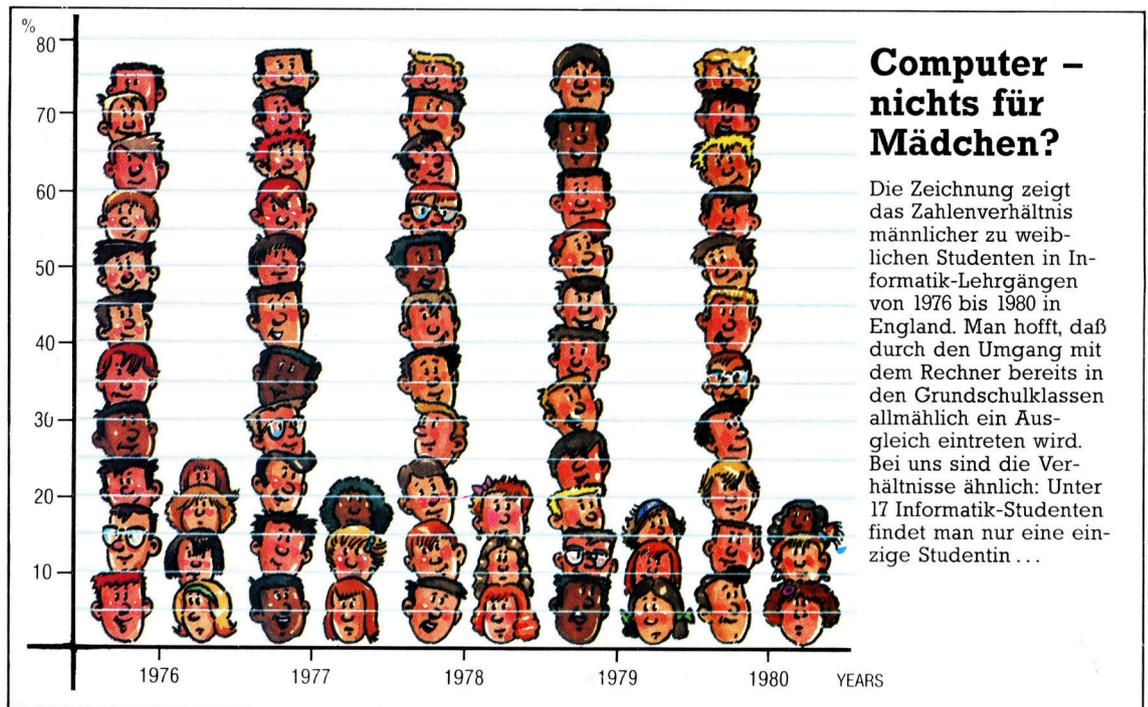
Wenn Sie Lernprogramme für ein Kind kaufen wollen, müssen verschiedene Faktoren beachtet werden: das Alter des Kindes, der verwendete Rechner und der Lehrstoff, der gerade in der Schule behandelt wird.

Im Gegensatz zu den Verhältnissen in England und den USA gibt es bisher nur an wenigen deutschen Grundschulen Computer. Es ist anzunehmen, daß dieser Zustand sich in kurzer Zeit ändern wird. Wenn Sie noch keinen Heimcomputer besitzen, kann es klug sein, bei

eigneten Programmen gleichzuziehen.

Neben den Computer-Produzenten stellt eine große Zahl unabhängiger Software-Häuser ebenfalls Lernprogramme her, über die Sie durch Computer-Zeitschriften und auch in Computer-Läden informiert werden. Für Kinder unter acht Jahren beschränken sich die meisten Programme auf das Training grundlegender Fähigkeiten wie Schreiben, Lesen und Rechnen.

Texas Instruments hatte sich dabei besonders verdient gemacht. Der von der Firma entwickelte Heimcomputer TI 99/4A wird, trotz recht guter Verkaufszahlen, zwar nicht mehr hergestellt, die TI-Lernprogramme zeichnen sich jedoch durch besondere Anwenderfreundlichkeit aus. Das 16-Bit-Gerät wurde aufgrund der besonders schnellen und komfortablen Programme bekannt, und viele Besitzer schätzen noch heute die einfache Bedienung,



### Computer - nichts für Mädchen?

Die Zeichnung zeigt das Zahlenverhältnis männlicher zu weiblichen Studenten in Informatik-Lehrgängen von 1976 bis 1980 in England. Man hofft, daß durch den Umgang mit dem Rechner bereits in den Grundschulklassen allmählich ein Ausgleich eintreten wird. Bei uns sind die Verhältnisse ähnlich: Unter 17 Informatik-Studenten findet man nur eine einzige Studentin ...

der Schulverwaltung nach deren Computer-Plänen zu fragen. Damit hätten Sie später die Möglichkeit, die Lernprogramme Ihres Kindes auch zu Hause auszuprobieren. Es ist jedoch kein Unglück, wenn Ihr Kind daheim mit einem anderen Rechner als in der Schule umgeht: Dadurch kann es seine Erfahrungen auch mit unterschiedlichen Systemen sammeln.

### Die Qual der Wahl

Für weiter verbreitete Computer gibt es natürlich mehr Lernprogramme als für unbekanntere Marken. Einige Hersteller legen jedoch besonderen Wert gerade auf diesen Teil ihres Software-Angebots. Dazu zählen zum Beispiel Apple, Atari, Commodore und Sinclair. Allerdings versuchen auch andere Firmen mit ge-

mit der auch jüngere Kinder gut zurechtzukommen. Die Programme sind auf Cartridges gespeichert, die lediglich in den Computer eingesteckt werden müssen.

Gerade in der Bundesrepublik haben es Eltern nicht leicht, für Kinder ein geeignetes Gerät zu finden. Die Schwierigkeiten beginnen oft schon bei der Tastatur, die im englischen QWERTY-System angelegt ist und auf dem die Umlaute fehlen. Das erweist sich als störend, wenn Kinder den Rechner auch für das Rechtschreibtraining oder zum Vokabellernen verwenden sollen. Es scheint, daß die deutschen Schulen nicht frühzeitig genug eine Entscheidung zugunsten eines bestimmten Systems getroffen haben, so daß interessierte Eltern auch bei den Lehrern gelegentlich eher Verwirrung als hilfreiche Tipps vorfinden.



Ein Vorbild für den Rechner-Einsatz in der Schule könnte England sein: Dort steht inzwischen in jeder der 29 000 Grundschulen ein Computer, während bei uns immer noch an 25 000 Lehranstalten solche Geräte fehlen. Inzwischen haben aber auch die deutschen Behörden die Zeichen der Zeit erkannt und versuchen – von Lehrern und Eltern gedrängt –, den Rückstand aufzuholen. Unterstützt werden sie jetzt auch von der Computer-Industrie, die allmählich anfängt, ihre Produkte „schulthauglich“ zu machen.

### Nachhilfe per Computer

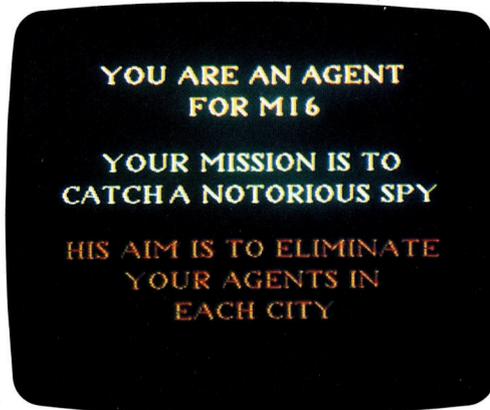
Auch die Schulbuch-Verlage wollen nicht zurückstehen und bereiten sich auf das Geschäft mit attraktiven Lehr- und Nachhilfe-Programmen vor. Es scheint, daß der Computer in Deutschland zwar spät, dafür aber mit beschleunigtem Tempo ins Klassenzimmer vorrückt.

Auf die oben erwähnte Spiel/Lern-Software für Kinder unter acht Jahren folgt die Programmgruppe für die 8–12jährigen. Hier werden schon bestimmte Interessen gezielt angesprochen, erworbenes Wissen soll vertieft und erweitert werden. So gibt es für die meisten Heimcomputer Fremdsprachen- oder Musikprogramme, mit denen das Kind selbständig lernen kann. Falls Sie in der Schule keine Auskunft über geeignete Software bekommen (Ansprechpartner dafür ist der Informatik-Lehrer), probieren Sie die Programme vor dem Kauf am besten selbst aus. Manchmal hilft schon die Vorführung im Geschäft, die Spreu vom Weizen zu trennen – Finger weg von Programmen, mit denen man nicht auf Anhieb klarkommt!

Zumindest eine Kategorie von Lernprogrammen ist mit Sicherheit kein Mißgriff: Mit der Computer-Sprache LOGO finden sich gelegentlich sogar schon Kinder im Vorschulalter zurecht. LOGO gibt es für nahezu jeden Computer. Für Kinder liegt der besondere Reiz in der „Schildkröte“, manchmal auch „Igel“ genannt. Sie wird als Dreieck auf dem Bildschirm dargestellt und erlaubt durch einfache Kommandos das Zeichnen auf dem Monitor. Spielerisch finden die Kinder dabei den Zugang zur Geometrie – und damit zur Mathematik. Die Schildkröte wird mit Hilfe von Prozeduren über den Bildschirm geführt und vermittelt dadurch dem „Steuermann“ einen problemlosen Einstieg in die Technik des Programmierens.

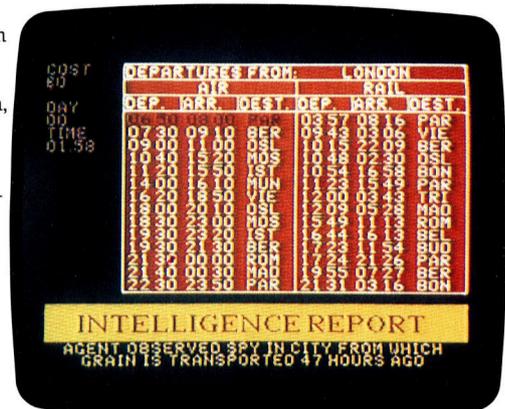
Die Verschiedenartigkeit der Spiel- und Lernprogramme macht die Auswahl nicht gerade einfach. Eine gute Informationsmöglichkeit bieten jedoch die Computer-Messen. Hier präsentieren nicht nur die Herstellerfirmen, sondern auch die Software-Produzenten ihre Erzeugnisse. Meist gibt es auch die Chance, verschiedene Programme zu vergleichen und dabei fündig zu werden.

## Geheimagent



In diesem spannenden Lehrprogramm muß der Schüler irgendwo in Europa einen Spion ausfindig machen. Als erstes müssen Indizien für den Aufenthalt des Spions entschlüsselt werden, dann kann die Jagd beginnen ...

Der Meisterspion kann nur in einer Stadt stecken – er bleibt aber nie länger als einige Stunden am selben Ort. Weiß man, wo er sich aufhält, kann der Spieler entweder mit dem Zug oder per Flugzeug hinterherreisen. Dabei muß entschieden werden, ob die Geschwindigkeit oder der Fahrpreis wichtiger ist.



Wenn von einem Unteragenten eine Nachricht eingeht, blinkt auf der Landkarte ein Licht, das den Standort zeigt. Ist der Agent „beseitigt“, erscheint die Nachricht nicht mehr. Man kann auch – selbstverständlich gegen Bezahlung – neue Hilfskräfte engagieren.

Informanten bieten ihre Berichte zum Kauf an. Sie sind aber verschlüsselt, und Hilfe aus dem Hauptquartier ist unverzichtbar ... Am Ende des Spiels kennt der Spieler Ort und Namen der europäischen Städte, weiß, wie Fahrpläne zu lesen sind und hat außerdem viel über sparsamen Umgang mit dem Reisegeld gelernt.



# Memotech MTX 512

**Eine ausgereifte Konstruktion und interessante Standardprogramme für Maschinencode sind die besonderen Merkmale dieses Heimcomputers.**

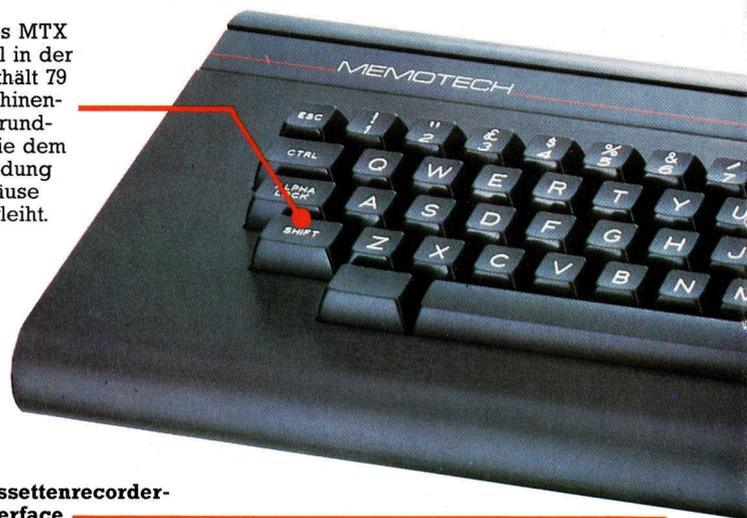
Der MTX 512 von Memotech erfüllt fast vollständig die Anforderungen des MSX-Standards (Microsoft Extended), nicht konform ist lediglich der verwendete Ton-Baustein Texas Instruments 76489 (MSX-Spezifikation: General Instruments AY-3-8910). Normgerecht sind die Z80-Zentraleinheit, der TMS9918/9928-Video-Prozessor von Texas Instruments und der verwendete BASIC-Dialekt, der der Microsoft-Version schon sehr nahekommt.

Der MTX 512 zeichnet sich neben den guten technischen Fähigkeiten durch ein formschönes Design aus. In der Aufmachung übertrifft er viele andere Geräte, bei denen raffinierte Elektronik häufig in zerbrechliches Plastik „gestopft“ ist – der MTX 512 präsentiert sich in einem wohlgestalteten mattschwarzen Gehäuse aus Aluminium in flacher Pultform.

Selbst der Zugang zum Innenleben der Maschine ist denkbar einfach: Nach Lösen zweier Inbusschrauben und Wegklappen des Bodenteils ist die Leiterplatte frei zugänglich. Im Vergleich zu anderen Rechnern ist die Anzahl der Bausteine darauf verhältnismäßig groß. Die Entwickler haben offensichtlich bewußt auf die Verwendung nur weniger großer Universalbausteine verzichtet und den traditionellen Aufbau mit vielen Einzelbausteinen bevorzugt, was die Fehlersuche erleichtert. In einem ULA sind Fehler schwer zu finden und überhaupt nicht zu beheben.

Die Gestaltung des Handbuchs ist dagegen nicht vorbildlich; es fehlen farbliche Hervorhebungen zur Erleichterung beim Nachschlagen und ein Stichwortverzeichnis. Ansonsten ist das Handbuch recht ausführlich – Memotech hat sich für einen „offenen“ Rechner entschieden, d. h., daß dem Käufer nichts vorenthalten werden soll. Daher finden Sie viele detaillierte Angaben über Speicherplatzverwaltung, wichtige Adressen, Ein- und Ausgabe-Adressierung, Schaltpläne sowie eine gute BASIC-Einführung. Ferner gibt es spezielle Kapitel über NODDY (s. Info-Kasten), den Assembler/Dis-

**Tastatur**  
Das Tastenfeld des MTX 512 ist komfortabel in der Bedienung. Es enthält 79 Profi-Schreibmaschinentasten auf einer Grundplatte aus Stahl, die dem Rechner in Verbindung mit dem Alu-Gehäuse hohe Stabilität verleiht.



**Cassettenrecorder-Interface**

**Joystick-Anschlußbuchsen**  
Der MTX 512 hat zwei Eingänge, an die Steuerknüppel nach Atari-Norm anschließbar sind.

**Erweiterungs-Steckerleiste**  
Bei diesem Rechner bestehen gute Ausbaumöglichkeiten. Die erste ernsthafte Ergänzung sollte eine Speichererweiterungskarte und eine serielle Interfacekarte sein, die zwei RS232-Schnittstellen enthält. Damit ist bei geeigneter Software u. a. Datenaustausch und der Aufbau von Rechnernetzen durchaus möglich.

**Uhr/Zeitgeber-Baustein**  
Der Baustein Z80 CTC dient als Taktgeber und zur Steuerung für den Microprozessor.

**Zentraleinheit**  
Als CPU ist der Microprozessor Z80 eingebaut; Taktfrequenz: 4 MHz.

**Benutzer-RAM**  
Standardmäßig mit 64 KByte; die Version MTX 500 hat nur 32 KByte.

**Parallelschnittstelle**  
Dieser Anschluß ist Centronics-kompatibel; bei zusätzlicher Ausstattung mit der RS232-Karte kann der MTX 512 eine Vielzahl von Druckern ansteuern.

assembler und die Grafik. Der MTX 512 fällt insbesondere durch seinen ROM-Assembler/Disassembler aus dem üblichen Rahmen. In Verbindung mit dem „Front Panel“ (Schaltbrett)-Softwarepaket wird dadurch eine sehr einfache Programmierung im Maschinencode ermöglicht. Der Assembler des MTX 512 kennt weder symbolische Adressen noch Sprungadressen (Labels), aber bei sorgfältiger Protokollführung während des Codierens reicht es für Programme mäßiger Größe. Der Umgang



## Memotech MTX 512:

### PREIS

ca. 1000 Mark

### GRÖSSE

488x202x56 mm

### ZENTRALEINHEIT

Z80

### TAKTFREQUENZ

4 MHz

### SPEICHERKAPAZITÄT

ROM: 24 KByte  
RAM: 64 KByte, dazu 16 KByte  
(Farb)Video-RAM. Auf 512  
KByte erweiterbar

### BILDSCHIRM- DARSTELLUNG

24 Zeilen mit je 40 Zeichen, 16  
Farben mit unabhängiger  
Wahl von Vorder- und Hinter-  
grund; 127 feste und 127 frei  
definierbare Zeichen

### SCHNITTSTELLEN

Cassettenrecorder, Fernseher,  
Farbmonitor

### PROGRAMMIER- SPRACHEN

BASIC, NODDY, Assembler

### WEITERE SPRACHEN

In Vorbereitung

### ZUBEHÖR

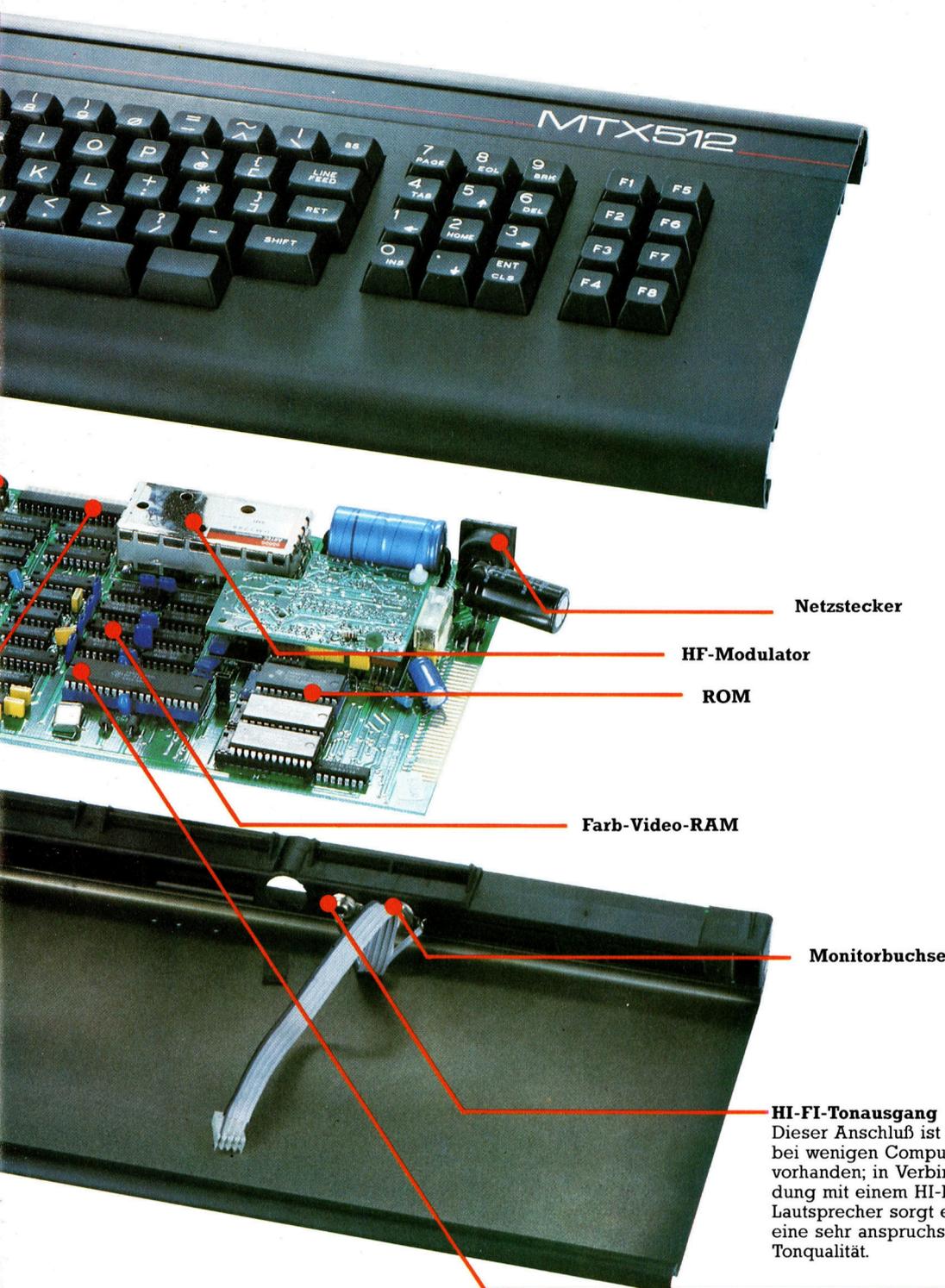
Handbuch für die Inbetrieb-  
nahme, BASIC-Einführung,  
Fernseherkabel

### TASTATUR

79 hochwertige Tasten

### DOKUMENTATION

Gründlich und vollständig,  
aber nicht sehr attraktiv; ent-  
hält alle erforderlichen Infor-  
mationen, um anspruchsvollen  
Programmierern die volle  
Nutzung des Rechners zu er-  
lauben.



mit dem Assembler und der Maschinensprache wird später noch ausführlich behandelt.

Das Front Panel ist für Rechner dieser Klasse eine echte Neuerung; es ermöglicht eine sehr wirksame Fehlersuche und -korrektur im Maschinencode. Leider ist diese Besonderheit nur dürftig dokumentiert. Zwar gibt es eine vollständige Liste der Befehle, aber nur wenige Informationen über ihre Funktion und fast keine Beispiele.

## NODDY

Das Betriebssystem enthält einen Auszug der Sprache NODDY, was dem Rechner eine Sonderstellung verleiht. NODDY ist als Anfängersprache konzipiert und erscheint zunächst äußerst simpel, aber bei näherem Hinsehen wird deutlich, wie trickreich etliche Anweisungen sind. Es gibt nur elf Befehle, jedoch keine arithmetischen – die Sprache ist vorwiegend für die Handhabung von Texten gedacht.

### Grafik-Baustein

Der Video-Prozessor TMS 9928 von Texas Instruments steuert die gesamte Bilderzeugung und verleiht dem MTX Grafik-Eigenschaften, die denen des TI 99/4A ähnlich sind. Eine Reihe von zusätzlichen Annehmlichkeiten bietet das Betriebssystem des MTX, z. B. die Möglichkeit, den Bildschirm in mehrere Fenster aufzuteilen.

# Mehr über Dateien

**Nachdem eine allgemeine Struktur des Programmprojektes ausgearbeitet wurde, werden wir uns der Datenverwaltung widmen.**

**D**as Adreßbuchprogramm, das in den vorangegangenen Teilen des BASIC-Kurses entwickelt wurde, entspricht in seiner Struktur einem einfachen Datenverwaltungsprogramm. Somit ist es ein gutes Beispiel, um das Konzept von „Files“ bzw. Dateien zu erklären. Dieses Wort kann mehrere verschiedene Bedeutungen haben.

In der Programmierung läßt sich eine Datei mit einem Karteikasten vergleichen. Sie ist eine Sammlung bestimmter Informationen, die gemeinsam gespeichert werden. Computer speichern Dateien auf magnetischen Bändern oder Disketten. Jede Datei mit Informationen erhält einen Namen, so daß der Computer jederzeit Zugriff zu diesen Daten hat. Die Informationen auf einem Band oder einer Diskette können ein Programm sein oder auch nur einfach Daten, die von einem Programm verwendet werden. Wenn wir als Beispiel unser Adreßbuch-Programm nehmen, so bestehen die benötigten Informationen aus zwei separaten Teilen: den Programmdateien und den Daten, die das Programm verarbeitet. Das Programm ist die Ansammlung an Instruktionen, die dem Computer (und dem Benutzer) gestattet, die Daten zu manipulieren.

Die vom Programm verwendeten Daten sind die Informationen, die Sie sonst in einem Adreßbuch vorfinden – Namen, Adressen usw. Sie umfassen aber auch Daten, die der Anwender normalerweise nicht zu sehen bekommt. Dies sind die sogenannten „Hilfsdaten“, die das Programm zur Unterstützung seiner Arbeit verwendet. Beispiele für solche Daten sind „Flags“, darunter versteht man Informationen, die auf die momentane Größe der Stammdatei hinweisen (z. B. die Anzahl der enthaltenen Verzeichnisse). Der Grund, weshalb solche Daten getrennt von den Programmdateien abgelegt werden, wird offensichtlich, wenn wir versuchen, sie in das Programm zu integrieren.

Die Daten des Adreßbuch-Programms müssen oft geändert werden. In der Theorie könnten alle Verzeichnisse innerhalb des Programms gespeichert und mit READ- und DATA-Anweisungen gelesen werden. Jedesmal, wenn Sie Änderungen vornehmen – z. B. Namen und Adressen hinzufügen oder löschen –, müßten aufwendige Änderungen am Programm vorgenommen werden. Im äußersten Fall könnte das dazu führen, daß das Programm ausgedruckt werden müßte, um festzustellen, wo die Änderungen einzubauen sind. Danach würde die Neuprogrammierung

der entsprechenden Passagen folgen. Das größte Problem hierbei liegt darin, daß die neuen Programmpassagen keine lauffähigen Module darstellen und somit nicht getestet werden können – die Änderungen am Programm sind daher möglicherweise schwerwiegende Fehlerquellen. Die einzige Möglichkeit zur Überprüfung des Programmes bestünde dann darin, das Programm zu starten.

Glücklicherweise kann all dies umgangen werden, da Sie die Daten unabhängig vom Programm speichern können. Dies geschieht mit Hilfe von Dateien, die auf Cassette oder Diskette angelegt werden. Diese Dateien bestehen aus Verzeichnissen, die genauso gehandhabt werden wie die Daten in DATA-Anweisungen. Das Programm ist in der Lage, eine oder mehrere solche Dateien zu „öffnen“, die Daten zu lesen und dann die Datei zu „schließen“.

Mit diskettenorientierten Computersystemen funktioniert das Lokalisieren, Lesen und Schreiben einer bestimmten Datei sehr schnell – das Lokalisieren einer Datei dauert nicht länger als den Bruchteil einer Sekunde, das Lesen oder Schreiben höchstens ein paar Sekunden. Ein cassettenorientiertes Computersystem andererseits ist da erheblich langsamer. Der Anwender muß das Band so lange zurückspulen und dann abspielen, bis der Rechner die richtige Datei gefunden hat. Ein weiterer Vorteil bei der Verwendung von Disketten ist, daß Sie die Möglichkeit haben, mit mehreren gleichzeitig geöffneten Dateien arbeiten zu können.

---

## OPEN und CLOSE

---

Dateien lassen sich auf verschiedene Art und Weise handhaben, entsprechend der verwendeten BASIC-Version. Der beste Weg, die richtige Methode herauszufinden, ist, im Bedienungshandbuch Ihres Computers unter den Anweisungen OPEN und CLOSE nachzulesen. Experimentieren Sie dann mit diesen Anweisungen. Die Erklärungen, die wir Ihnen hier geben, sind sehr allgemein gehalten und sollen einen Überblick über die Handhabung von Dateien vermitteln.

Dateien können sequentiell oder „zufällig“ (random access file) aufgebaut sein. In einer sequentiellen Datei sind die Informationen der Reihe nach gespeichert, angefangen mit der ersten, dann der zweiten, dritten Information und so weiter. Eine „zufällig“ aufgebaute Datei ist so organisiert, daß der Computer benötigte



Daten sofort ansteuern kann, ohne alle Informationen durchsuchen zu müssen. Das Betrachten eines Films im Kino kann man mit einer sequentiellen Datei vergleichen: Man verfolgt den Film vom Anfang bis zum Ende. Der Videorecorder dagegen kann mit einer „zufälligen“ Datei verglichen werden: Man kann das Band vor- oder zurückschleppen und jede gewünschte Stelle anwählen. Wir werden uns überwiegend mit sequentiellen Dateien befassen, da sie mit dem Cassettenrecorder als Speichermedium einfacher zu realisieren sind.

Stellen Sie sich vor, Sie wollten ein Verzeichnis der durchschnittlichen Tagestemperaturen einer Woche anlegen. Dazu einige Datenbeispiele:

MONTAG	13.6
DIENSTAG	9.6
MITTWOCH	11.4
DONNERSTAG	10.6
FREITAG	11.5
SAMSTAG	11.1
SONNTAG	10.9

Um die Sache einfach zu halten, behandeln wir die Daten als numerische Daten, wobei Montag Tag 1 ist und Sonntag Tag 7. Die Daten können dann so dargestellt werden:

1,13.6,2,9.6,3,11.4,4,10.6,5,11.5,6,11.1,7,10.9

Um diese Daten in einer sequentiellen Datei zu speichern, werden die folgenden Programmschritte benötigt:

OPEN (ÖFFNE) die Datei  
Schreibe die Daten in die Datei  
CLOSE (SCHLIESSE) die Datei

Wann immer die OPEN-Anweisung verwendet wird, ist es notwendig festzulegen, ob Daten vom Computer in die Datei geschrieben werden (ein sogenannter Output) oder von der Datei in den Computer gelesen werden (ein sogenannter Input). Beim Microsoft-BASIC lauten die Anweisungen OPEN"O" und OPEN"I". Ein kurzes Programm, um die Daten in eine Datei zu schreiben, sieht wie folgt aus:

```
100 OPEN"O",#1,"TEMP.DAT"
110 PRINT #1,1,13.6,2,9.6,3,11.4,4,
    10.6,5,11.5,6,11.1,7,10.9
120 CLOSE #1
```

Das Wort OPEN in Zeile 100 macht die Datei für das Programm verfügbar. Hinter OPEN steht "O", um anzuzeigen, daß die Daten vom Programm in der Datei gespeichert werden sollen. Danach folgt #1, wodurch dem Computer mit-

geteilt wird, daß es sich um die erste Datei im Programm handelt. Jede Datei erhält ihre eigene Nummer, die dann jeweils mit den INPUT#- oder PRINT#-Anweisungen verwendet wird. Abschließend folgt der Dateiname in Anführungsstrichen.

Ein komplettes Beispielprogramm zum Schreiben, Lesen und Drucken der Daten sieht wie folgt aus:

```
100 OPEN "O",#1,"TEMP.DAT"
110 PRINT #1,1,13.6,2,9.6,3,11.4,4,10.6,
    5,11.5,6,11.1,7,10.9
120 CLOSE #1
130 REM ZEILEN 130, 140 REPRAESENTIE-
    REN
140 REM WEITERE PROGRAMMZEILEN
150 OPEN "I",#1,"TEMP.DAT"
160 FOR X=1 TO 7
170 INPUT #1, TAG,TEMP
180 PRINT "TAG ";TAG,TEMP
190 NEXT X
200 CLOSE #1
210 END
```

In diesem Programm wird eine Datei mit der Nummer #1 geöffnet und als TEMP.DAT bezeichnet. Dann werden mit Hilfe der PRINT #-Anweisung Daten in die Datei geschrieben, und anschließend wird sie mit CLOSE # geschlossen. Will man diese Datei im weiteren Verlauf des Programms wieder öffnen, müssen die Nummer und der Dateiname verwendet werden. (Die Nummer muß nicht dieselbe sein wie bei der Erstellung der Datei. Wichtig ist nur, daß die Nummer immer mit der übereinstimmt, die beim Öffnen der Datei verwendet wird.) INPUT #1 in Zeile 170 zeigt an, daß die Eingaben von einer mit #1 bezeichneten Datei stammen (dies ist die Datei TEMP.DAT) und nicht über die Tastatur eingegeben werden.

Nun werden wir uns wieder der \*INITIALISIERUNG\*-Unterroutine zuwenden. Die Verwendung von fest vorgegebenen Daten-Feldlängen ist eine ziemliche Verschwendung von Speicherplatz, doch wird dadurch die Programmierung erheblich einfacher. Wenn wir für jedes Feld eine ganze Zeile mit 40 Zeichen festlegen, werden immer 40 Zeichen für dieses Feld gespeichert, auch wenn der größte Teil der Zeile mit Leerzeichen gefüllt ist. In einigen BASIC-Versionen ist es möglich, Felder auf eine Länge von bis zu 256 Zeichen zu dimensionieren. Die Dimensionierung bestimmt lediglich die Anzahl der Elemente in einem Feld und nicht die Größe jedes Elementes.

Wenn Ihre BASIC-Version mehrdimensionale Felder handhaben kann, wäre es möglich, eine separate Dimensionierung für jedes Feld durchzuführen. Da jedoch nur wenige BASIC-Versionen hierzu in der Lage sind, befassen wir uns mit Alternativen. Die einfachste Methode ist, ein separates String-Feld für jedes der Felder zu verwenden. Hier gibt es eine

Möglichkeit zu „tricksen“, wenn Sie mehrdimensionale Felder verwenden wollen, Ihr BASIC jedoch dazu nicht in der Lage ist.

Der Trick liegt darin, alle Elemente, die in einem mehrdimensionalen Feld sind, so zu behandeln, als befänden sie sich in einem eindimensionalen Feld. Ein zweidimensionales Feld mit drei Reihen und 5 Spalten würde wie folgt dimensioniert: DIM A(3,5). Somit enthielte es insgesamt 15 Elemente: A(1,1) bis A(3,5). Dieselbe Information könnte in einem einfachen Feld wie diesem abgelegt werden: DIM A(15). Immer, wenn dann auf ein zweidimensionales Feld in der Form A(R,S) Bezug genommen wird, muß wie folgt umgerechnet werden: A((R-1)×5+S).

Soll ein separates String-Feld für jedes Feld verwendet werden, müssen Sie entscheiden, wie die Felder zu dimensionieren sind. Der einfachste Weg ist, eine festgelegte Feldgröße zu verwenden. Doch dies beschränkt die maximale Anzahl an speicherbaren Verzeichnissen im Datensatz. Eine bessere Lösung ist, die Feldgröße in Abhängigkeit von der Anzahl der benutzten Verzeichnisse festzulegen. Trotzdem muß man beachten, daß nicht alle BASIC-Versionen String-Felder in beliebiger Größe zulassen. Selbst wenn es möglich ist, kann eine große Anzahl an Verzeichnissen in einem Datensatz schnell den gesamten zur Verfügung stehenden Speicherplatz aufbrauchen. Im folgenden finden Sie ein Programm, das die maximal mögliche Anzahl an Elementen herausfindet. Jedesmal, wenn das Programm Sie fragt „WELCHE FELDGROESSE?“, geben Sie einen größeren Wert ein, bis eventuell eine Fehlermeldung erscheint. Die Anweisung CLEAR in Zeile 100 bewirkt, daß das Feld am Ende jedes Programmlaufs eliminiert wird. Ohne diese Anweisung erhielten Sie in Zeile 30 eine Fehlermeldung, da versucht würde, ein Feld mehrmals zu dimensionieren.

```

10 READ D$
20 INPUT "WELCHE FELDGROESSE?";A
30 DIM N$(A)
40 FOR L=1 TO A
50 LET N$(L)=D$
60 NEXT L
70 FOR L=1 TO A
80 PRINT L,N$(L)
90 NEXT L
100 CLEAR
110 GOTO 10
120 DATA "COMPUTER KURS"
130 END

```

Selbst wenn in jedem Element nur 40 Zeichen vorgesehen sind, mit 5 Feldern je Verzeichnis, und wenn für jeden Bereich 256 Elemente reserviert werden, wird der für alle Daten benötigte Speicherplatz gewaltig. Wenn für jedes zu speichernde Zeichen 1 Byte gebraucht wird, müßten insgesamt 51 200 Bytes (5×40×256

Bytes) nur für die Daten zur Verfügung stehen. Es ist offensichtlich, daß es nicht effizient ist, derart viel Speicherplatz für die Daten zu verbrauchen. Deshalb verwendet man separate Daten-Dateien.

Bedauerlicherweise sind Routinen zur Handhabung von Dateien relativ kompliziert. Wenn Sie die Verwendung externer Dateien vermeiden wollen, ist die einzige Alternative, die Daten in DATA-Anweisungen abzulegen, so daß sie im Programm ständig zur Verfügung stehen. Wenn Sie bereits mit einigen kurzen Programmen etwas mit dem Lesen und Schreiben von externen Daten experimentiert haben, wird Ihnen der ganze Vorgang, sicher leichter verständlich. Zur Demonstration werden hier zwei unterschiedliche Computer und BASIC-Versionen verwendet, um das Beispiel der Tagestemperaturen zu verdeutlichen. Zum einen wird der Sinclair Spectrum und zum anderen der Acorn B verwendet. Beide BASIC-Versionen unterscheiden sich von dem normalerweise verwendeten Microsoft-BASIC. Lesen Sie also gegebenenfalls in den Abschnitten über BASIC-Dialekte der vorangegangenen Hefte nach.

Beim Spectrum-BASIC sind die Anweisungen OPEN# und CLOSE# für die Arbeit mit dem Microdrive reserviert. Verwendet man als Speichermedium einen Programmrecorder, braucht man spezielle Formen der SAVE- und LOAD-Befehle. Der normale SAVE-Befehl wird benutzt, um Programme und Programmvariablen auf Cassette abzuspeichern (und natürlich auch normale Daten in DATA-Anweisungen). Felder können auf Cassette gespeichert werden, indem man die SAVE-DATA-Anweisung verwendet. Die Eingabeform sieht wie folgt aus:

SAVE Dateiname DATA Feld-Name( )

Um die Tagestemperaturen abzuspeichern, müssen Sie zuerst ein String-Feld DIMENSIONIEREN und die Daten hineinschreiben (beispielsweise mit READ-DATA-Anweisungen). Um den Unterschied zwischen dem Dateinamen und dem Feld-Namen deutlicher zu machen, nennen wir das Feld C\$ und verwenden als Dateinamen „TEMPDAT“.

```

10 DIM C$(14)
20 FOR x=1 TO 14
30 READ C$(x)
40 NEXT x
50 DATA 1,13.6,2,9.6,3,11.4,4,10.6,5,11.5,6,
  11.1,7,10.9
60 SAVE "TEMPDAT" DATA C$( )
70 STOP
80 LOAD "TEMPDAT" DATA C$( )
90 FOR L=1 TO 14 STEP 2
100 PRINT "TAG";C$(L)C$(L+1)
110 NEXT L
120 STOP

```

Zeile 60 speichert alle Daten im String-Feld C\$ in einer Datei mit dem Dateinamen TEMPDAT. Das Programm stoppt dann in Zeile 70. Sie sollten dann die Cassette zurückspulen. Das Befehlswort CONT setzt den Programmablauf fort. Zeile 80 vollzieht den Vorgang in umgekehrter Richtung.

Der Acorn B hat eine der komfortabelsten BASIC-Versionen, die man bei Heimcomputern finden kann. Sie erlaubt eine strukturierte Programmierung mit fortschrittlichen Details wie REPEAT-UNTIL und Prozeduren. Wie die Prozeduren definiert werden, können Sie in der nachfolgenden Programmversion für den Acorn B sehen. Beachten Sie, daß im BBC-BASIC die Richtung, in der die Daten verschoben werden, durch Verwendung von OPENOUT und OPENIN bestimmt wird:

```

10 DIM C$(2,7)
20 FOR R=1 TO 7
30   FOR S=1 TO 2
40     READ C$(S,R)
50   NEXT S
60 NEXT R
70 DATA 1,13.6,2,9.6,3,11.4,4,10.6,5,11.5,
      6,11.1,7,10.9
80 INPUT "TIPPEN SIE S ZUM SPEICHERN
      DER DATEN",K$
90 IF K$ <> "S" THEN GOTO 80
100 PROCSAVE: CLEAR: DIM C$(2,7)
110 INPUT "SPULEN SIE DAS BAND ZU-
      RUECK UND TIPPEN SIE DANN L",K$
120 IF K$ <> "L" THEN GOTO 110
130 PROCLOAD
140 PRINT "TAG TEMPERATUR"
150 FOR R=1 TO 7
160   FOR S=1 TO 2
170     PRINT C$(S,R); " "
180   NEXT S:PRINT
190 NEXT R
200 END
300 DEF PROCSAVE
310 X=OPENOUT ("TEMPDAT")
320 FOR R=1 TO 7
330   FOR S=1 TO 2
340     PRINT # X,C$(S,R)
350   NEXT S
360 NEXT R
370 CLOSE #X
380 ENDPROC
400 DEF PROCLOAD
410 X=OPENIN ("TEMPDAT")
420 FOR R=1 TO 7
430   FOR S=1 TO 2
440     INPUT # X,C$(S,R)
450   NEXT S
460 NEXT R
470 CLOSE #X
480 ENDPROC

```

Einer der großen Vorteile des BBC-BASIC ist, daß die Handhabung von Dateien sowohl bei Cassetten als auch bei Disketten gleich ist. So-

mit können für den Programmrecorder geschriebene Programme ohne Änderungen direkt auch mit einer Diskettenstation verwendet werden.

Bis jetzt haben Sie gesehen, wie Daten aus DATA-Anweisungen in Felder und dann in Dateien auf Cassette (oder Diskette) übertragen werden und umgekehrt. Als nächstes werden wir uns eingehender mit dem INITIALISIERUNGs-Vorgang beschäftigen. Sie werden sehen, wieviele Felder gebraucht werden, wieviele Elemente jedes haben muß und an welchen Punkten innerhalb des Programms Daten übertragen werden müssen (sowohl in den Computer als auch aus dem Computer in die Datei). Dabei ist besonders auf die dialektbezogenen Eingabeformate zu achten.

## BASIC-Dialekte



Einige Computer wie z. B. der Acorn B und der Atari können lange Variablenamen verarbeiten (KOMPLETTNAMES). Der Spectrum erlaubt lange numerische Variablenamen. Bei alphanumerischen Variablenamen ist nur jeweils ein Buchstabe möglich. Dragon 32, VC 20 und Commodore 64 lassen lange Variablenamen zwar zu, doch werden nur die ersten beiden Buchstaben berücksichtigt. Der Variablenname KOMPLETTNAMES ist zwar möglich, doch wird er identisch mit einem Namen wie KOLLER\$ gehandhabt: Beide Namen haben dieselben Anfangsbuchstaben.

Beim Oric können Variablenamen generell nicht länger als zwei Buchstaben sein (zuerst ein Buchstabe und dann eine Zahl oder ein Buchstabe).

Beim Dragon 32 müssen Sie dieses Format verwenden:

```

OPEN "O",#-1,"TEMPDAT"
PRINT #-1,1,13.6,2,9.6,3 usw.
CLOSE #-1

```

und

```

OPEN "I",#-1,"TEMPDAT"
INPUT #-1,D,T
CLOSE #-1

```



Beim Commodore 64 und VC 20 müssen Sie dieses Format verwenden:

```

OPEN 1,1,2,"TEMPDAT"
PRINT 1,1:PRINT 1,13.6:PRINT usw.
CLOSE 1

```

und

```

OPEN 1,1,0,"TEMPDAT"
INPUT #1,D,T
CLOSE 1

```



Der Lynx und der Oric-1 in der Grundaufführung können keine Cassetten-Dateien handhaben.



Beim ZX 81 ersetzen Sie bitte die ersten Zeilen durch:

```

10 DIM C$(14,4)
20 FOR X=1 TO 14
30 PRINT "EINTRAG-NR.:";X
40 INPUT C$(X)
50 NEXT X
60 STOP

```

Geben Sie nun die Zeilen 90 bis 120 des Spectrum-Programms ein. Wenn die Programmausführung in Zeile 60 abbricht, speichern Sie das Programm. Beim anschließenden Laden enthält C\$ die Daten. Um das Programm zu starten, geben Sie GOTO 90 ein; RUN würde alle Variablen auf Null setzen.



# Spielend programmiert

**Anwendungsgeneratoren haben viel gemeinsam mit Programmgeneratoren, und sie lassen sich sowohl für Spiele als auch für kommerzielle Anwendungen einsetzen.**



**Dieses Programmpaket ist eine Art Anwendungsgenerator für Spiele. Über ein Objektmenü und eine Anzahl grafisch dargestellter Bildelemente bestimmt der Anwender den Spiel Aufbau.**

In einer der vorigen Ausgaben haben wir Spezialprogramme untersucht, die, vom Anwender mit Anweisungen versehen, Programmstrukturen selber komplettieren. Programmgeneratoren dieser Art sind auf den meisten kommerziellen Computersystemen verfügbar, werden aber auch für einige Heimcomputer angeboten. Diese müssen allerdings mit mindestens einem Diskettenlaufwerk ausgerüstet sein.

Weitaus verbreiteter sind jedoch Programmgeneratoren, die auf eine bestimmte Art von Abläufen ausgerichtet sind, sogenannte Anwendungsgeneratoren. Sie unterscheiden sich von den universell einsetzbaren Programmgeneratoren dadurch, daß die Programme nur in Verbindung mit dem Generator selbst lauffähig sind. Um die Unterschiede zwischen diesen beiden Generatoren deutlich zu machen, werden wir auf beiden ein Rechnungsprogramm erstellen.

Bei einem Programmgenerator laden Sie zunächst den Generator von der Diskette in den Arbeitsspeicher. Nachdem Sie alle Fragen über Dateistrukturen, Berechnungsmethoden,

Bildschirm Aufbau und Druckformat beantwortet (d. h. die gewünschte Anwendung beschrieben) haben, legen Sie eine leere Diskette in das Laufwerk, auf die der Generator das neue Programm speichert. Dieser Vorgang läßt sich wiederholen, so daß jede Filiale einer Firma mit einer Kopie des Programms ausgerüstet werden könnte.

Ein Anwendungsgenerator scheint auf den ersten Blick längst nicht so flexibel zu sein. Das erstellte Programm wird nach Eingabe der Parameter auf der gleichen Diskette gespeichert, da zur Ausführung des Programms der Generator selbst benötigt wird. Das Programm kann zwar auch auf einer anderen Diskette abgelegt werden, braucht zur Ausführung aber auch hier die Originaldiskette mit dem Generator. Es lassen sich also eine unbegrenzte Anzahl einzelner Anwendungen erstellen, die jedoch nur in Verbindung mit dem Generator selbst funktionieren. Möchten Sie Ihre Anwendung einem Bekannten zur Verfügung stellen, müssen Sie eine weitere Kopie des Generators kaufen. Die Generatoren selbst sind vom Hersteller natürlich auf mehrfache Weise gegen unerlaubtes Kopieren geschützt.

Ein Anwendungsgenerator ist in Wirklichkeit ein hochentwickeltes Allzweckprogramm. Mit Eingabe der von Ihnen gewünschten Programmabläufe belegen Sie die internen Variablen des Programms („Parameter“ genannt) mit Werten. Damit wird der Programmfluß gesteuert, die Datenstruktur bestimmt und die Bildschirm- und Druckausgabe festgelegt. Auf der Diskette wird für jede Anwendung nur die Liste dieser Variablen oder Parameter gespeichert. Diese Liste, die manchmal auch als „Anwendungsmodul“ bezeichnet wird, besteht nur aus einer Reihe von Instruktionen, die dem Generator mitteilen, wie er eine bestimmte Anwendung ausführen soll.

Manche Generatorpakete geben Ihnen sogar die Möglichkeit, die Anwendung in Form einer sehr hohen Programmsprache zu schreiben (ähnlich der Pseudosprache, die wir für die Entwicklung einer neuen Programmroutine im BASIC-Kurs einsetzen). Die Parameterliste wird von dem Generator interpretiert, der wiederum von einem BASIC-Interpreter übersetzt

wird (vorausgesetzt natürlich, der Generator ist in BASIC geschrieben).

Nicht selten werden Anwendungsmodule und Generatoren von unterschiedlichen Herstellern angeboten. Bestes Beispiel dafür ist dBase II (eines der populärsten Datenbankpakete für Microcomputer). Es kann als Anwendungsgenerator bezeichnet werden, da seine Anwendungsmodule aus Folgen von Datenbankbefehlen bestehen, die einer Hochsprache ähnlich sind. Anwendungsmodule für Spezialbereiche (z. B. Buchungssysteme für Börsenmakler) lassen sich erstellen, ohne daß die entsprechenden Dateizugriffe von Grund auf neu programmiert werden müssen. Es ist durchaus möglich, daß eine unter dBase II geschriebene Anwendung besser ist als ein BASIC-Programm, da der Programmierer sich bei dBase nur auf die Abläufe konzentrieren muß und nicht hauptsächlich auf den Programmcode wie bei BASIC. Bei einem Anwendungsgenerator wurden die fehleranfälligen Teile des Programms (z. B. Dateiaufbau und Steuerung) von den Autoren des Generators erstellt und in der Praxis von Tausenden von Anwendern getestet.

### Benutzerfreundlich

Hauptunterschied zwischen einem Programm und einem Anwendungsgenerator ist die unterschiedliche Benutzerfreundlichkeit. Das Ergebnis des erstgenannten besteht hauptsächlich aus maschinell erzeugtem Code, der sowohl in puncto Effektivität als auch im Stil dem von einem Programmierer erzeugten Code unterlegen ist. Bei einem Anwendungsgenerator bestehen jedoch bis zu 99 Prozent des Programms aus dem Code des Softwarehauses, der mit großer Wahrscheinlichkeit im Maschinencode geschrieben wurde. Bestes Beispiel dafür ist der „Silicon Office“, ein hochentwickelter Anwendungsgenerator für Microcompu-

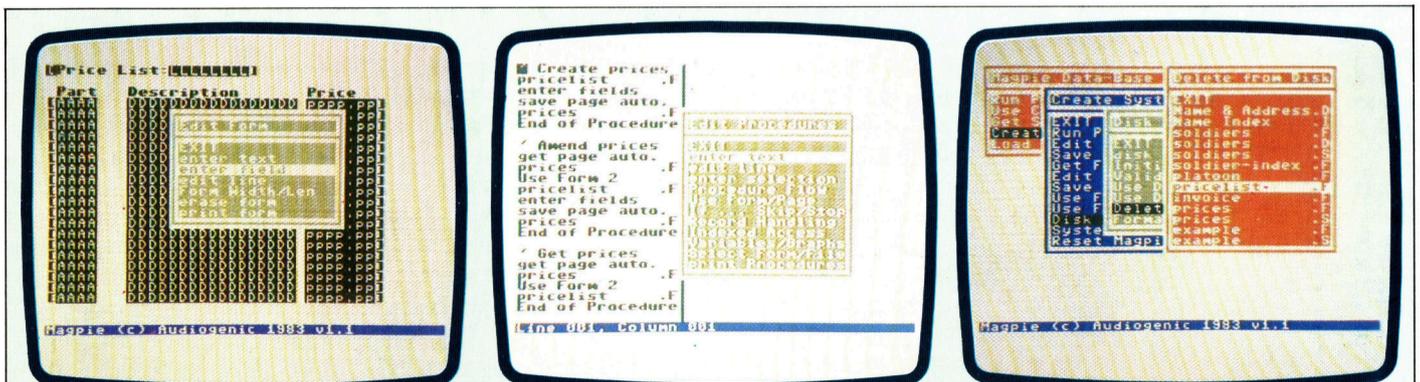
ter, der außerdem auch noch leicht zu bedienen ist. Die erzeugten Programme sind schnell und effektiv, verfügen über Prüfroutinen für Eingabefehler und zeichnen sich durch einen menügesteuerten Bildschirmaufbau aus.

Anwendungsgeneratoren sind nicht nur auf kommerziellen Einsatz begrenzt. Das beste Beispiel dafür ist das „Pinball Construction Set“, bei dem das Anwendungsmodul nur die Anordnung der einzelnen Flipper-elemente enthält.

Tatsächlich gibt es eine Reihe von Ähnlichkeiten zwischen Anwendungsgeneratoren und der „objektorientierten Programmierung“. Bei beiden gibt der Programmierer nur an, welches Ergebnis das Programm erzielen soll. Selbst einfachste Kalkulationssysteme für Heimcomputer können als Anwendungsgeneratoren bezeichnet werden: Sind die Beziehungen zwischen den unterschiedlichen Feldern einmal definiert, erledigt das Programmpaket alle weitere Arbeit.

Magpie, ein Anwendungsgenerator für den Commodore 64, ist auf kommerzielle Anwendung ausgelegt. Dieser Generator ist ein gutes Beispiel dafür, wie objektorientierte Programmierung mit Hilfe visueller Darstellung anwenderfreundlich gestaltet werden kann: Über das Bildschirmlayout werden bei der Festlegung von Dateistrukturen gleich die Beziehungen zwischen den Feldern unterschiedlicher Datensätze definiert.

Es gibt inzwischen eine Anzahl Programme, die zwar nicht als Anwendungsgeneratoren bezeichnet werden können, aber einige der Grundelemente enthalten. Bei ihrem ersten Lauf stellen diese „parametergesteuerten“ Programme dem Anwender eine Reihe Fragen, deren Antworten auf der Programmdiskette gespeichert werden und so die weitere Arbeit des Programms in bestimmten Bereichen selbstständig steuern. Dies befreit Sie von langweiligen Routinearbeiten.



### Magpie auf dem Commodore 64

Bei der Erstellung einer Anwendung wird zunächst der Aufbau der Formulare, Abläufe und Anzeigeformate festgelegt. Der Anwender füllt jede Spalte mit Buchstaben (A, D, P), die definieren, welche Felder der

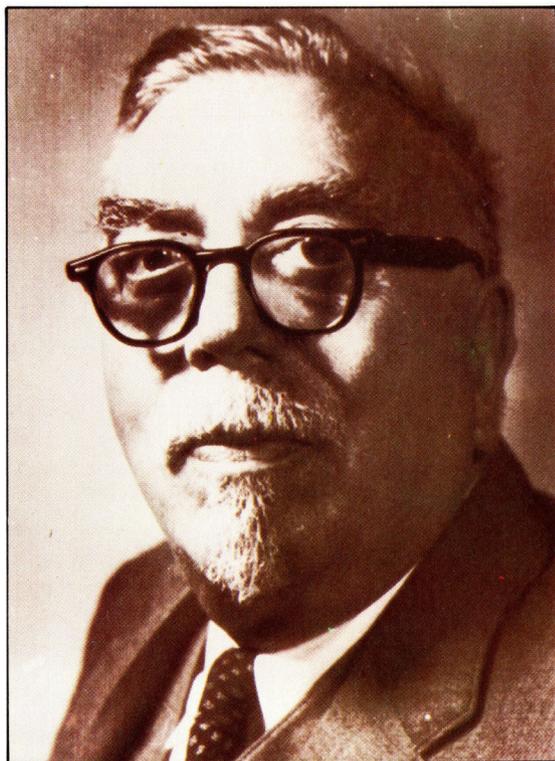
Dateien dort aufgebildet werden sollen. Als nächstes werden in einer hohen Programmiersprache alle Abläufe für eine Befehlsliste definiert, die dann von Magpie interpretiert wird. Hier im Bild die Befehls-

folgen, mit denen die Preisliste von der Diskette abgerufen werden kann. Magpie ist menügesteuert. Das Bild zeigt, wie nacheinander CREATE, DISK und DELETE angewählt wurden.



# Norbert Wiener

**Die mathematischen Studien dieses genialen Mannes bildeten den Grundstein für das neue Forschungsgebiet „Kybernetik“.**



Kybernetik ist die Wissenschaft von sich selbst regulierenden, stabilen Systemen, die man in der Mechanik ebenso findet wie in der Biologie oder Elektrotechnik. Wiener stellte die Information in eine Reihe zu den Faktoren Energie und Materie: Kupferdraht beispielsweise läßt sich sowohl auf die transportierte Leistung wie auch auf die übermittelte Information hin untersuchen.

Der Begriff Kybernetik stammt aus dem Lateinischen und bedeutet „Steuermann“. Wiener hatte die Funktionen von James Watts Dampfmaschinen studiert und dabei erkannt, daß Computer eine besondere Fähigkeit der Selbstregulation erfordern, um richtig zu funktionieren.

## Negative Rückkopplung

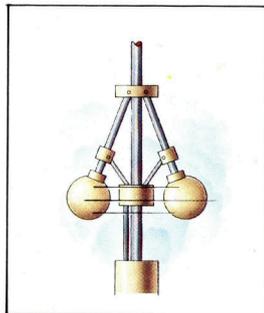
Beispiel für eine solche Regelung ist etwa ein Thermostat: Er steuert die Heizung in einem Hause je nach dem Über- oder Unterschreiten einer Temperatur. Der Mensch stellt nur noch den gewünschten Wert ein. Diese Fähigkeit zur Selbstüberwachung bezeichnete Wiener als „negative Rückkopplung“ – Rückkopplung, weil der vom System erzeugte Effekt (die Wärme) das zukünftige Verhalten des Systems beeinflusst, negativ, weil das Schalten des Thermostates die Heizung auf die eingestellte Temperatur drosselt.

Es gibt auch Systeme mit „positiver Rückkopplung“. Sie würden – in unserem Beispiel – auch die Temperatur selbst wählen. Ein Automat, der sich neben der Auswahl seiner „Ziele“ auch noch selbst reproduzieren könnte, wäre schon beinahe perfekt.

Wieners kybernetische Theorie hat die Forschung im Bereich Information und Regelung stark erweitert. Alles kann als Information betrachtet werden, denn wir erkennen die Umwelt mit Ohr, Auge und Tastsinn. Jeder dieser „Sensoren“ beschränkt unsere Wahrnehmung auf eine vergleichsweise winzige Bandbreite des eigentlichen Geschehens, das uns in seiner Gesamtheit überfordern würde. Die Analyse von Zusammenhängen dieser eingeschränkten Sicht gehört ebenfalls zum Bereich der Kybernetik. Aber auch das Entschlüsseln von Daten mittels statistischer Methoden rechnet die Wissenschaft dazu.

Norbert Wiener starb 1964, also lange vor der Microcomputer-Revolution. Dennoch hat er viele der heute auftretenden Probleme vorausgesehen und auch darüber geschrieben.

Wiener war vom Fliehkraftregler der ersten Dampfmaschinen als einfachstem Beispiel für negative Rückkopplung fasziniert. Dabei sind zwei Gewichte an Hebelarmen befestigt, die auf einer mit dem Schwungrad der Maschine gekoppelten Achse sitzen. Bei zunehmender Drehzahl wandern die Gewichte nach außen und schließen über eine mechanische Verbindung die Drosselklappe. Die Drehzahl stabilisiert sich beim eingestellten Wert. In modernen Rechnern ist diese Rückkopplung sehr viel komplexer, das Prinzip bleibt jedoch gleich.



Norbert Wiener wurde 1894 in Missouri, USA, geboren. Das Diplom der Mathematik erhielt er schon als 14-jähriger, mit 18 war er bereits Professor für Logik. Wiener war auch in Deutschland tätig: als Gastprofessor unter dem berühmten Mathematiker David Hilbert an der Universität Göttingen.

Zur Computer-Entwicklung kam Wiener erst recht spät in seinem Leben. Zuvor hatte er lange Jahre am Massachusetts Institute of Technology (MIT) die Bewegung von Festkörpern in Flüssigkeiten (Brownsche Bewegung) studiert. Der Bewegungsablauf von Teilchen war mit den klassischen Methoden der Physik nicht vorauszusagen, so daß eine neue Berechnungsweise entwickelt wurde. Damit konnte immerhin die Wahrscheinlichkeit einer bestimmten Konstellation von Teilchen zu einem vorgegebenen Zeitpunkt berechnet werden.

Beim Ausbruch des Zweiten Weltkriegs stellte sich Wiener in den Dienst der US-Regierung und widmete sich den mathematischen Problemen, die das Erfassen bewegter Objekte ermöglichten. Die Entwicklung automatischer Zielvorrichtungen, seine physikalischen Studien und breitgestreute Interessen von der Philosophie bis zur Neurologie verarbeitete er 1948 in dem Buch „Kybernetik“.

# Das Zeichengenie

**„Grafpad“ heißt ein Digitalisieretablett, mit dem Sie detaillierte Zeichnungen anfertigen können.**

**G**rafiktablets gehören zu den vielseitigsten und nützlichsten Peripheriegeräten. Sie werden vorwiegend als Entwurfs- und Zeichenhilfen verwendet, sei es beim Freihandzeichnen, beim Entwerfen von Schaltungen oder in der Kartografie. Sie sind aber auch als zusätzliches Eingabegerät geeignet. Auf einer Tablett-Auflage lassen sich alle Steuerungsmöglichkeiten eines Programms darstellen. Sie müssen dann nur das entsprechende Feld mit einem Stift berühren, und die Software wertet aus, welche Option gewählt wurde.

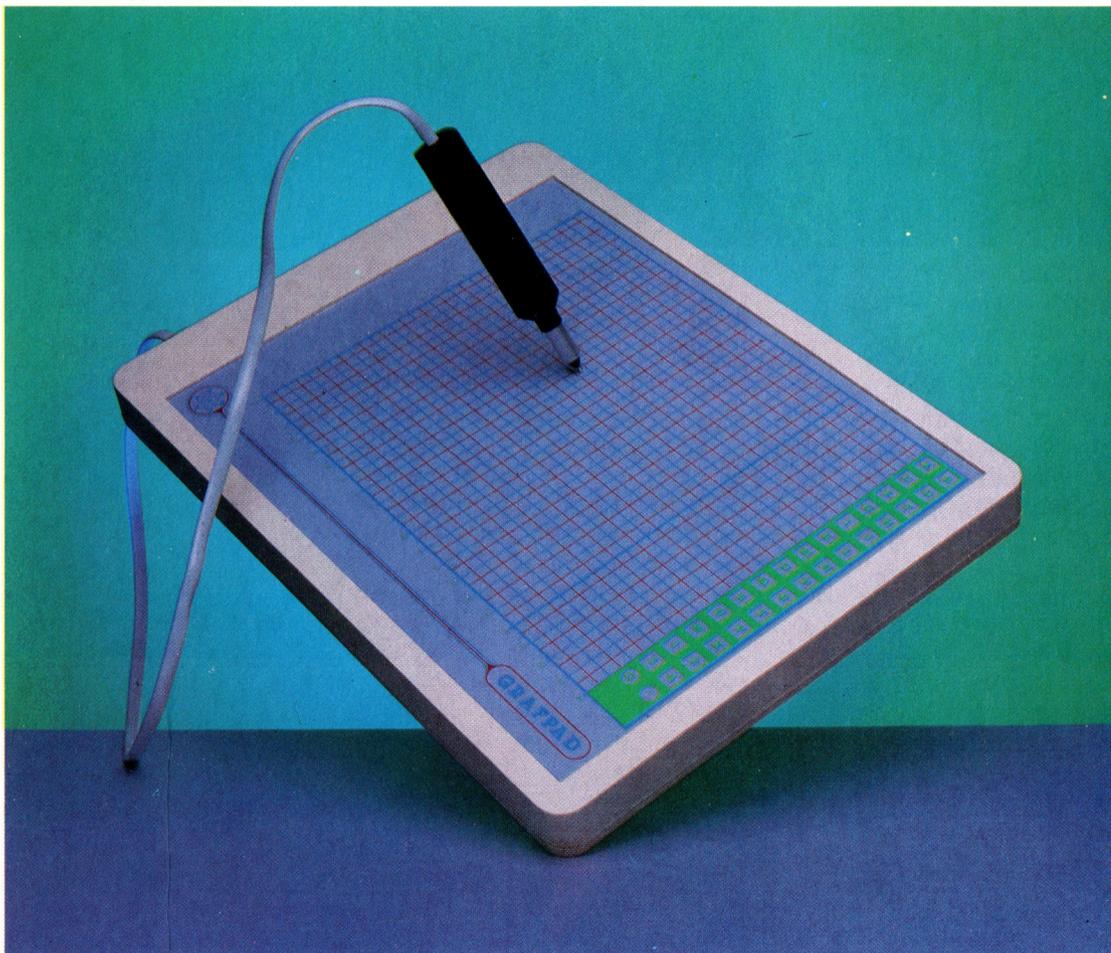
Dieses Zubehör wurde früher als Spezialgerät fast ausschließlich für Ingenieure und Designer produziert. Bei den jetzigen Preisen können sich aber auch Heimcomputer-Benutzer die Anschaffung leisten. Das vorgestellte „Grafpad“ ist in Deutschland für den Acorn B erhältlich.

Der Preis für dieses Grafiktablett liegt zwar deutlich über dem, der für andere Tablets zu

zahlen ist – die Vorteile des Grafpads gegenüber anderen Produkten sind jedoch augenscheinlich: professionelles Design, präzises Ablesen der angewählten Koordinaten, gute Software usw.

Das Grafpad besteht aus drei Komponenten: dem Tablett selbst, dem Griffel und der zugehörigen Software. Das Tablett wird mit dem User Port des Rechners verbunden, und das Stiftkabel wird seitlich eingesteckt. Das Tablett ist mit einem Koordinatenraster überzogen. Daneben ist eine Befehlsleiste mit Buchstaben- und Ziffern-Beschriftung angebracht, mit der Sie, ohne die Tastatur zu benutzen, Steuerbefehle eingeben können. Obenauf befindet sich ein Plexiglas-Schutz, unter dem Sie selbstgefertigte Folien (mit eigenen Mustern oder Kommandos) legen können.

Im Tablett eingelassen ist ein Gitter aus 320x256 Drähten mit einer Maschenweite von 1,2 mm. Die Griffelspitze ist ein kleiner Schal-



**Das Grafpad kann unter Verwendung der mitgelieferten Software zur Gestaltung von Entwürfen und Zeichnungen benutzt werden. Sie können es aber auch mit selbstgeschriebenen Programmen als vielseitiges Eingabegerät z. B. für die Menüsteuerung verwenden.**



### Addiermaschine

Dieses kleine Programm ist für den Acorn mit dem Grafpad als Eingabegerät gedacht. Die Befehle werden mit Hilfe des Zeichenstiftes eingegeben. Die Berührung mit dem Stift setzt das Programm in Gang.

```

10 REM GRAFPAD DEMO
20 REM
30 REM An adding machine using the Grafpad
40 REM
50 MODE 1
60 PRINT "ADDING MACHINE":PRINT
70 HIMEM=&29FF
80 XADJUST=7
90 REM
100 REM set up co-ordinate table
110 REM
120 DIM B(15):FOR I=1 TO 15:B(I)=I*25:NEXT I
130 R1=0:R=0
140 REM
150 REM load Grafpad driver program
160 REM
170 *LOAD "PADREAD" 2A00
180 PENZ=&2A00
190 REM
200 REM MAIN PROGRAM STARTS HERE
210 REM
220 REM wait for pen to be pressed
230 CALL PENZ
240 !X%=!X%-XADJUST:IF !X%<0 THEN !X%=0
250 IF ?UX>0 THEN 230
260 REM beep to register pen press
270 SOUND 1,-15,120,1
280 X=!X%
290 REM convert position to 0-12
300 l=0
310 IF X>B(I) THEN I=I+1:GOTO 310
320 REM interpret codes 0-12
330 IF l<10 THEN R=R*10+l
340 IF l=10 THEN PRINT R;" +":R1=R1+R:R=0
350 IF l=11 THEN PRINT R;" =":PRINT "
R1+R:PRINT R1:R1=0:R=0:PRINT:PRINT
360 IF l=12 THEN PRINT:PRINT"CLEAR":PRINT:R=0:R1=0
370 REM wait until pen is lifted again
380 CALL PENZ
390 IF ?UX=0 THEN 380
400 REM loop for next pen press
410 GOTO 230

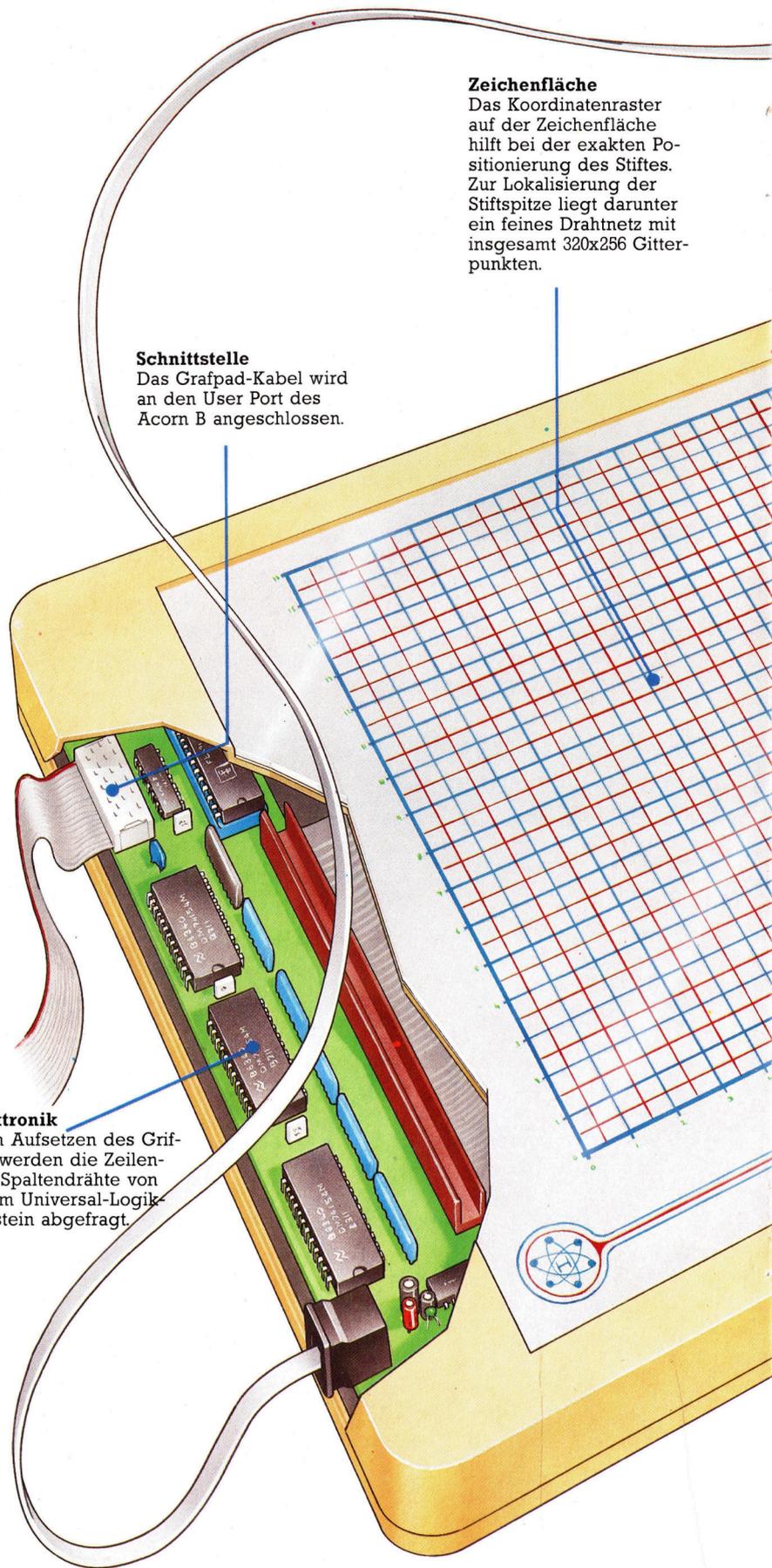
```

ADDING MACHINE											
34	+	4	+								
96	=	8	=								
130		12									
3	+	19876	+								
9	+	8876	+								
6	+	8864	+								
28	+	1243	=								
126	=										
172		38859									

ter: Beim Berühren des Tablett erhält ein Draht nach dem andern einen Spannungsimpuls, bewirkt durch eine Universal-Logikschaltung (ULA=Uncommitted Logic Array), bis die Stiftspitze kapazitiv geortet ist. Dieser Abtastvorgang spielt sich 2000mal pro Sekunde ab. Die Lokalisierung des Stifts erfolgt also sehr schnell und zuverlässig, wenn der Griffel dicht auf der Tablettoberfläche geführt wird.

Sobald der Stift das Tablett berührt, bekommt der Rechner das Signal „stylus down“ (Stift sitzt auf), und die Koordinaten des Aufsetzpunktes werden übertragen. Was danach passiert, ist eine Frage der Software. Je nach Programm erscheint an der entsprechenden Bildschirmposition ein Kreuz, oder es wird ein bestimmter Befehl ausgelöst. Allerdings ist beim Grafpad die Anzahl der Rasterpunkte (320x256) zum Zeichnen von ganz glatten Kurven oder sehr feiner Details nicht ausreichend.

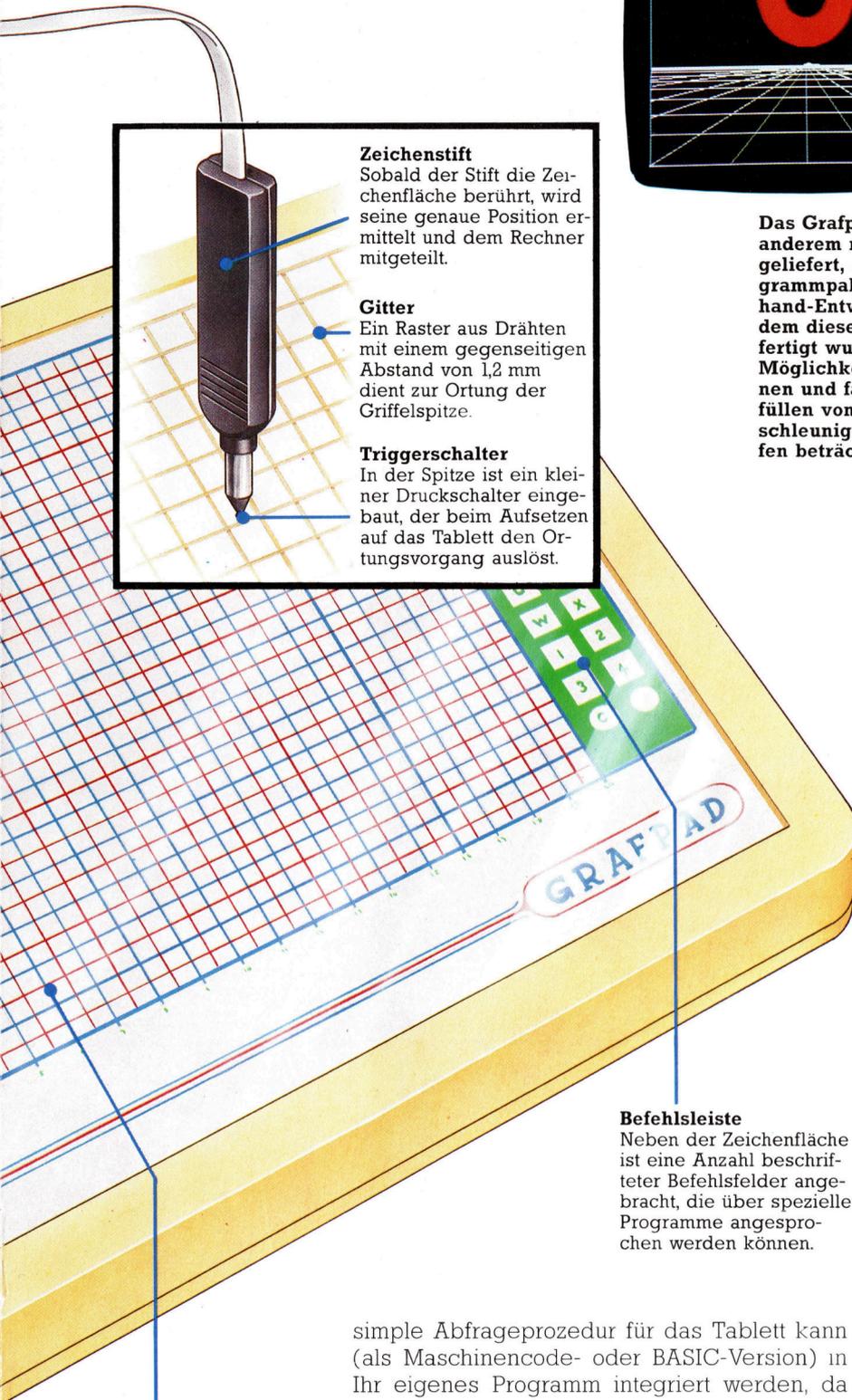
Zum Grafpad werden ein Demonstrationsprogramm, ein einfaches Zeichenprogramm und ein CAD (Computer Aided Design=Rechnergestützter Entwurf)-Paket geliefert. Die



**Zeichenfläche**  
Das Koordinatenraster auf der Zeichenfläche hilft bei der exakten Positionierung des Stiftes. Zur Lokalisierung der Stiftspitze liegt darunter ein feines Drahtnetz mit insgesamt 320x256 Gitterpunkten.

**Schnittstelle**  
Das Grafpad-Kabel wird an den User Port des Acorn B angeschlossen.

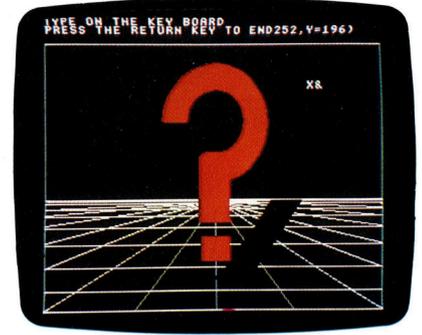
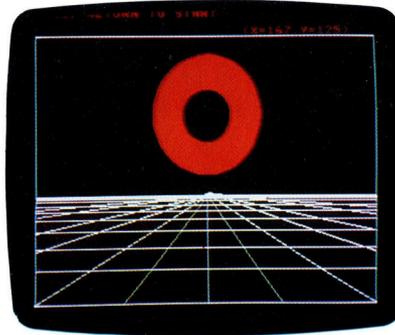
**Elektronik**  
Beim Aufsetzen des Griffels werden die Zeilen- und Spaltendrähte von einem Universal-Logikbaustein abgefragt.



**Zeichenstift**  
Sobald der Stift die Zeichenfläche berührt, wird seine genaue Position ermittelt und dem Rechner mitgeteilt.

**Gitter**  
Ein Raster aus Drähten mit einem gegenseitigen Abstand von 1,2 mm dient zur Ortung der Griffelspitze.

**Triggerschalter**  
In der Spitze ist ein kleiner Druckschalter eingebaut, der beim Aufsetzen auf das Tablett den Ortungsvorgang auslöst.



Das Grafpad wird unter anderem mit **PROG2** geliefert, einem Programmpaket für Freihand-Entwürfe, mit dem diese Bilder angefertigt wurden. Die Möglichkeit zum Zeichnen und farbigen Ausfüllen von Kreisen beschleunigt das Entwerfen beträchtlich.



werden. Tricks wie das Kopieren oder Verschieben von Details sind nicht vorgesehen. Natürlich könnte man dies alles auch mit der Tastatur allein machen, aber mit einem Grafiktablett lassen sich die Zeichnungen viel bequemer anfertigen. Ein Nachteil des Grafpads ist, daß sich jeweils nur vier verschiedene Farben wählen bzw. einsetzen lassen.

Das CAD-Programm ist eigentlich nur eine Demonstration der Grundlagen. Zuerst müssen Sie die für einen Entwurf benötigten Bildsymbole erzeugen, für Schaltpläne z. B. die Zeichen für Transistoren, Widerstände oder Logik-Gatter. Bei Bedarf können Sie aber auch Möbelstücke oder Kachelmuster entwerfen. Wenn die Grafiksymbole erst einmal definiert wurden, beginnen Sie mit dem eigentlichen Entwurf, wobei Sie die Elemente beliebig anordnen, vergrößern und durch gerade Linien verbinden können.

Ein „echtes“ CAD-System funktioniert fast genauso, aber die Grafpad-Software ist nicht für Profis gedacht. Es fehlen Möglichkeiten zur Beschriftung, zur Maßstabsänderung, zur genauen Ausrichtung und zum Drehen von Details, zur Vergrößerung von Bildausschnitten usw. Wichtig wäre auch mehr Flexibilität bei der Fehlerkorrektur. Trotz der Menüsteuerung durch das Tablett wird oft die Rechnertastatur benötigt, was die Bedienung erschwert.

Das Grafpad selbst ist ein vielseitiges Peripheriegerät, das sein Geld wert ist. Hinsichtlich Fläche, Auflösung und Zuverlässigkeit setzt der Preis natürlich Grenzen. Die mitgelieferte Software ist eher enttäuschend, daher ist das Gerät vorwiegend für Benutzer zu empfehlen, die ihre Programme selbst entwickeln und lediglich die vorhandene Technik sinnvoll nutzen wollen.

**Befehlsleiste**

Neben der Zeichenfläche ist eine Anzahl beschrifteter Befehlsfelder angebracht, die über spezielle Programme angesprochen werden können.

simple Abfrageprozedur für das Tablett kann (als Maschinencode- oder BASIC-Version) in Ihr eigenes Programm integriert werden, da das zugehörige Handbuch die kompletten Listings der Programme enthält.

Nach dem Aufruf des Zeichenprogramms muß zuerst die Ausgangsposition des Stiftes auf dem Koordinatenraster festgelegt werden. Sie können nun zwischen Linien, Kästchen, Kreisen, Dreiecken und auch „Freihand“ wählen. Geschlossene Flächen können eingefärbt

**Transparente Abdeckung**

Die Oberseite des Tabletts ist durch eine Plexiglasplatte geschützt. Darauf lassen sich selbstgefertigte Auflagen anbringen.



# Sinfonie

**Für wenig Geld ausgezeichnete Möglichkeiten der Klangerzeugung auf dem Oric.**

## MUSIC und PLAY

Dieses Programm spielt einen Akkord in C-DUR (C, G & E) nacheinander in allen Hüllkurven:

```
10 REM*****
20 REM*AKKORD*
30 REM*****
40 MUSIC 1,4,1,0:REM
   *C*
50 MUSIC 2,3,8,0:REM
   *G*
60 MUSIC 3,3,5,0:REM
   *E*
70 FOR E=1TO7:REM
   *WAHL HUELL-
   KURVE*
80 PLAY 7,0,E,750:REM
   *AKKORD SPIELLEN*
90 PLAY 0,0,0,0:REM
   *ENDE AKKORD*
100 WAIT 50:REM
   *PAUSE*
110 NEXT E:REM
   *NAECHSTE
   HUELLKURVE*
```

Der Oric verfügt über eine Reihe interessanter Fähigkeiten, von denen besonders die Klangerzeugung gut ausgebaut ist. Sein Tonspektrum umfaßt sieben Oktaven. Die Klänge werden über drei Oszillatoren, einen Rauschgenerator und sieben feste Hüllkurven gesteuert.

Das BASIC des Oric beinhaltet einen Satz von fertigen Soundroutinen wie ZAP, PING, SHOOT und EXPLODE. Das folgende Programm zeigt, wie diese Befehle eingesetzt werden können. Darin enthalten ist ebenfalls der Befehl WAIT, mit dem der Computer veranlaßt wird, eine Weile zu pausieren. Die Zeit wird in Hundertstelsekunden angegeben und beträgt in unserem Beispiel (WAIT 200) zwei Sekunden:

```
10 ZAP:WAIT 200
20 EXPLODE
30 GOTO 10
```

Der Befehl SOUND eignet sich am besten für Spezialeffekte. Er ist so aufgebaut:

SOUND C, P, V

wobei C den Kanal oder die Nummer des Oszillators (1–6) bezeichnet, P die Tonhöhe (10–5000) und V die Lautstärke (0–15). Mit den Kanälen 1 bis 3 wird einer der Oszillatoren angesprochen, während 4 bis 6 die gleichen Funktionen haben, jedoch den Rauschgenerator dazuschalten. Die Tonhöhe ist etwas ungenau, wobei 10 den höchsten Ton (ungefähr bei 10 KHz) und 5000 den niedrigsten (etwa 100 Hz) angibt. Die höchste Lautstärke liegt bei 15, eine angenehme mittlere Lage ist der Wert 6. Wird die Lautstärke auf 0 gesetzt, übernimmt die in dem Befehl PLAY angegebene Hüllkurve die Steuerung.

Der MUSIC-Befehl eignet sich ausgezeichnet für das Spielen von exakten Tonhöhen. Mit seinem einfachen Aufbau lassen sich auch komplizierte Musikprogramme leicht verstehen. Das Format sieht folgendermaßen aus:

MUSIC C, O, N, V

wobei C der Kanal (1, 2 oder 3) ist, O die Oktave (0 bis 6) bezeichnet, N den Ton (1 bis 12) und V die Lautstärke (0 bis 15). Der Befehl arbeitet ähnlich wie SOUND. Mit der Angabe des Kanals werden die Oszillatoren 1, 2 oder 3

angewählt (dieser Befehl kann allerdings kein Rauschen hinzuschalten) und die Lautstärke reicht von 0 (hier übernimmt PLAY die Steuerung) bis 15. Die Variable O ermöglicht die Auswahl einer bestimmten Oktave, in der der Ton (den die Variable N des gleichen Befehls kennzeichnet) gespielt werden soll. Der Ton (N) kann die Werte 1–12 annehmen und entspricht damit den herkömmlichen Notenbezeichnungen:

1	2	3	4	5	6
C	C#	D	D#	E	F
7	8	9	10	11	12
F#	G	G#	A	A#	B

Der folgende Befehl spielt den Ton A bis 440 Hz auf Kanal 1 mit Lautstärke 6:

MUSIC 1, 3, 10, 6

Mit der Kombination der Befehle MUSIC und PLAY lassen sich die Sound-Fähigkeiten des Oric jedoch weitaus besser ausnutzen. Der PLAY-Befehl hat folgendes Format:

PLAY C, N, E, P

wobei C der Kanal (0–7) ist, N das Rauschen (1–7), E die Hüllkurve (1–7) und P die Ablaufdauer der Hüllkurve (0–32767).

Nummer	Kanal	Rauschen
0	Alle Osz..	aus aus
1	Osz. 1	+Osz. 1
2	Osz. 2	+Osz. 2
3	Osz. 1 & 2	+Osz. 1 oder 2
4	Osz. 3	+Osz. 3
5	Osz. 1 & 3	+Osz. 1 oder 3
6	Osz. 2 & 3	+Osz. 2 oder 3
7	Osz. 1, 2 & 3	+Osz. 1, 2 oder 3

Zuvor definierte MUSIC- (oder SOUND-) Befehle, deren Lautstärken auf 0 gesetzt wurden, können über PLAY gleichzeitig gespielt werden und mit entsprechend gewählten Kanalnummern Akkorde mit bis zu drei Tönen produzieren. Über die Wahl der Kanalnummer kann auch Rauschen in die Klänge gemischt werden. Mit der Variablen E wird eine der sieben festgelegten Lautstärkenhüllkurven für einen bestimmten Ton (oder Töne) definiert. Diese Möglichkeiten sind im Handbuch des Oric genau beschrieben.

## SOUND

Mit SOUND erzeugt dieses Programm die Geräusche eines Raumschiffes.

```
10 REM*****
20 REM*LANDUNG*
30 REM*****
40 FOR P=10 TO 3000
   STEP 10
50 SOUND 2,P,6
60 PLAY 2,0,1,1
70 NEXT P
80 WAIT 75
90 PLAY 0,0,0,0
100 END
```



# Lichtorgel

**Die Grafikmöglichkeiten des Spectrum lassen sich durch strukturierte Befehle einfach handhaben.**

**D**er Spectrum ist ein ausgezeichnetes Gerät für alle Einsteiger, die an guter Grafik und Farbdarstellungen interessiert sind. Der Standardzeichensatz mit Groß- und Kleinbuchstaben ist ebenso vorhanden wie einige Sonderzeichen von Sinclair. Über PRINT lassen sich diese Zeichen in einer von acht Farben auf den Bildschirm bringen. Die Befehle INK, PAPER und BORDER legen die Farbe für Zeichen, Bildschirm und Umrahmung fest.

Die Bildschirmdarstellung besteht aus 24 Zeilen mit je 32 Zeichenfeldern, wobei die beiden untersten Zeilen als Informationsfeld reserviert sind. In höchster Auflösung stehen 176x256 Pixel zur Verfügung. Die Möglichkeit, hochauflösende Grafik und Text mischen zu können, ist eine der vorteilhaften Fähigkeiten des Spectrum, weil damit beschriftete Diagramme und Bilder erstellt werden können. Der Bildschirminhalt kann mit SAVE auf Cassette gespeichert und bei Bedarf wieder zurückgeladen werden. Dieser Vorgang wird über den Befehl SCREEN\$ gesteuert, mit dem der Bildschirminhalt auch ausgedruckt werden kann.

## Spezialeffekte

In der niedrigen Auflösung lassen sich mit dem Befehl PRINT AT Zeichen auf dem Bildschirm horizontal und vertikal positionieren. Zusätzlich steht eine Reihe von Spezialeffekten zur Verfügung. Außer der inversen Darstellung gibt es noch blinkende Zeichen (FLASH) und doppelte Helligkeit (BRIGHT). Eine weitere interessante Möglichkeit der niedrigen Auflösung bietet der Befehl OVER, über den zwei Zeichen an jeder beliebigen Stelle des Bildschirms übereinandergelegt werden können. Besonders interessant ist diese Fähigkeit bei der gleichzeitigen Verwendung von Text und Grafik, wenn z. B. Diagramme überschrieben werden können, ohne daß die darunterliegende Grafik ausgelöscht wird. Dieser Befehl sollte jedoch mit Vorsicht eingesetzt werden, da bei einer Neufestlegung der Farbe (INK) auch das alte Zeichen seine Farbe ändert.

Zwei Bereiche des Arbeitsspeichers steuern den Bildschirminhalt: Ein Bereich enthält die Zeichen, während der andere Informationen über den Status jedes einzelnen Zeichenfeldes speichert. In diesem zweiten Bereich sind Farbe (INK), Hintergrundfarbe (PAPER) und Status des Zeichens (blinkend – FLASH etc.)

enthalten. Der Status eines Zeichenfeldes ist in einem einzigen Byte gespeichert, das von BASIC aus mit dem Befehl ATTR abgefragt werden kann.

Der Befehl

PLOT $x,y$

blendet ein Pixel an den Bildschirmkoordinaten ( $x,y$ ) in der Farbe (INK) auf.

DRAW $x,y,p$

zeichnet eine Linie zwischen der augenblicklichen Position des Cursors und den im Befehl angegebenen Koordinaten. Wird eine dritte Zahl hinzugefügt, entsteht ein Bogen. Normalerweise ist die dritte Zahl ein Teil von Pi (3,14159...). Bei der Angabe von Pi erscheint ein Halbkreis, während  $\text{Pi}/2$  einen flacheren Bogen zeichnet. Die Bögen lassen sich entweder nach links oder rechts beugen, je nachdem, ob die dritte Zahl negativ oder positiv ist.

CIRCLE $x,y,r$

zeichnet einen Kreis um das Zentrum ( $x,y$ ) mit dem Radius  $r$ . Bei den meisten CIRCLE-Befehlen in BASIC kann der Kreis zu Ellipsen abgeflacht werden. Auf dem Spectrum ist das jedoch nicht möglich.

Der Einsatz von Farbe in der hochauflösenden Grafik ist nicht ohne Probleme. Da sich Grafik und Text mischen lassen, kann für jedes Zeichenelement (je acht mal acht Pixel) nur eine Farbe (INK) angegeben werden. Überkreuzen sich zwei Linien mit unterschiedlichen Farben, nimmt das Zeichenfeld, in dem sich diese beiden Linien treffen, die Farbe (INK) der zuletzt gezeichneten Linie an.

```
10 REM*SMILEY*
20 CLS
30 BORDER 6
40 PAPER 6
50 INK 2
60 CIRCLE 122,88,50
70 CIRCLE 97,108,5
80 CIRCLE 147,108,5
90 PLOT 92,68
100 DRAW 60,0,PI/3
110 STOP
```

### SMILEY

Dieses Programm demonstriert die Befehle PLOT, CIRCLE und DRAW und zeichnet das bekannte SMILEY-Gesicht auf den Bildschirm.

# Käfer bei der Arbeit

**Nachdem die Turtle-Geometrie unter verschiedenen Gesichtspunkten dargestellt wurde, behandelt der Kurs die Verwendung von Sprites. Anhand von gezeichneten Beispielen mit dem Commodore-LOGO wird demonstriert, wie man Animationseffekte erzeugt.**

**L**OGO-Sprites werden ähnlich wie Turtles gesteuert. Anders als bei Turtles kann man die Kontur eines Sprite selbst definieren. Allerdings drehen sich diese Konturen nicht auf dem Bildschirm, wenn die Richtung des Sprite geändert wird.

Beim Commodore-LOGO zählt die Turtle als Sprite Nummer 0. Es gibt sieben weitere, die von 1 bis 7 beziffert werden. Zu Beginn ist Sprite 0 das „Arbeitssprite“, das den eingegebenen Befehlen folgt. Nach Eingabe von TELL 1 rückt Sprite 1 an seine Stelle. Alle folgenden Anweisungen werden von Sprite 1 ausgeführt, solange kein anderes Sprite aufgerufen wurde.

Auf dem Bildschirm sieht man jedoch noch nichts. Grund dafür ist, daß die Sprites, anders als die Turtles, im „Verborgenen“ arbeiten. Um Sprite 1 sichtbar zu machen und seine Richtung feststellen zu können, rufen Sie ST auf. Daraufhin erscheint ein Spriteraster, mit dem Sie unter Verwendung der Turtle-Prozeduren FD, BK, RT, LT, PD, PU, ST und HT experimentieren können.

Wird Sprite 1 auf dieselbe Position wie die Turtle gebracht, scheint es dahinter zu verschwinden. Generell gilt, daß Sprites mit niedrigen Zahlen „vor“ Sprites mit hohen Zahlen gezeigt werden.

Die Commodore LOGO Utility Diskette enthält einen Sprite-Editor, der mit dem Befehl READ "SPRED" eingelesen wird. Um die Kontur von Sprite 1 editieren zu können, muß es mit TELL 1 zum Arbeitssprite ernannt werden. Geben Sie nun EDSH ein. Auf dem Schirm erscheint anschließend ein vergrößertes Spriteraster, auf dem der Cursor beliebig bewegt werden kann. Durch Druck auf die \*-Taste wird ein Pixel gefüllt. Durch Druck auf die Leertaste wird es gelöscht.

Haben Sie Ihr Sprite gestaltet, erfolgt die Definition durch CTRL-C. Bleibt das Sprite unsichtbar, versuchen Sie, ST einzugeben. Andere Sprites können die gleiche Kontur haben. Mit SETSHAPE 1 erhält das Arbeitssprite die Definition von Sprite 1. Nach Definition von mehreren Formen können die Sprites mit SAVESHAPES "FILENAMES" gespeichert und mit READSHAPES "FILENAMES" geladen werden. Es gibt ein bekanntes mathematisches Problem, bei dem vier Käfer in den Winkeln eines Quadrates plaziert sind. Sie bewegen sich alle

mit der gleichen Geschwindigkeit, und jeder folgt dem Käfer zu seiner Rechten. Aufgabe ist es, ihre Wege zu verfolgen. Unser LOGO-Programm löst diese Aufgabe unter Verwendung von Sprites.

Mit den eingegebenen Prozeduren setzt man einfach eine Kopie des Sprites in jeden Winkel und läßt sie dann einander folgen. Die Käferform wird als Sprite 3 definiert; die anderen erhalten dieselbe Form durch SETSHAPE 3 beim Positionieren.

Die eigentliche Lösung erfolgt durch den FOLLOW-Befehl. X und Y werden zunächst auf die Koordinaten des Sprite gesetzt, dem gefolgt werden soll (:B), dann des nachrückenden (:A). Um den Vorgang einzuleiten, gibt man TOWARDS ein. Mit zwei Eingaben werden die Koordinaten des zu erreichenden Punktes dargestellt. Als Ergebnis erscheint die Richtung des Arbeitssprites zu diesem Punkt.

## Animation

Zu Beginn wird eine Reihe gleichartiger Spriteformen, die sich nur geringfügig voneinander unterscheiden, definiert. Nach Aufruf der Prozeduren entsteht der Eindruck von Bewegung.

```
TO RUNN
  TELL 0
  DRAW
  PU
  BIGX BIGY
  SETH 90
  RUNNING 2
END
```



```
TO RUNNING :SHAPE
  FD 5
  SETSHAPE :SHAPE
  IF :SHAPE=4 THEN MAKE "SHAPE 1
  RUNNING :SHAPE+1
END
```

Vor dem Aufruf der Prozeduren muß das File SPRITES von der Utility-Diskette eingelesen werden. Auf ihr sind nützliche Programme enthalten, u. a. BIGX und BIGY, mit denen die Spritgröße verdoppelt wird. SMALLX und SMALLY verkleinern das Sprite wieder auf Originalgröße. Die drei Sprites werden durch READSHAPES "RUNNER" eingegeben.





## Tobender Drache

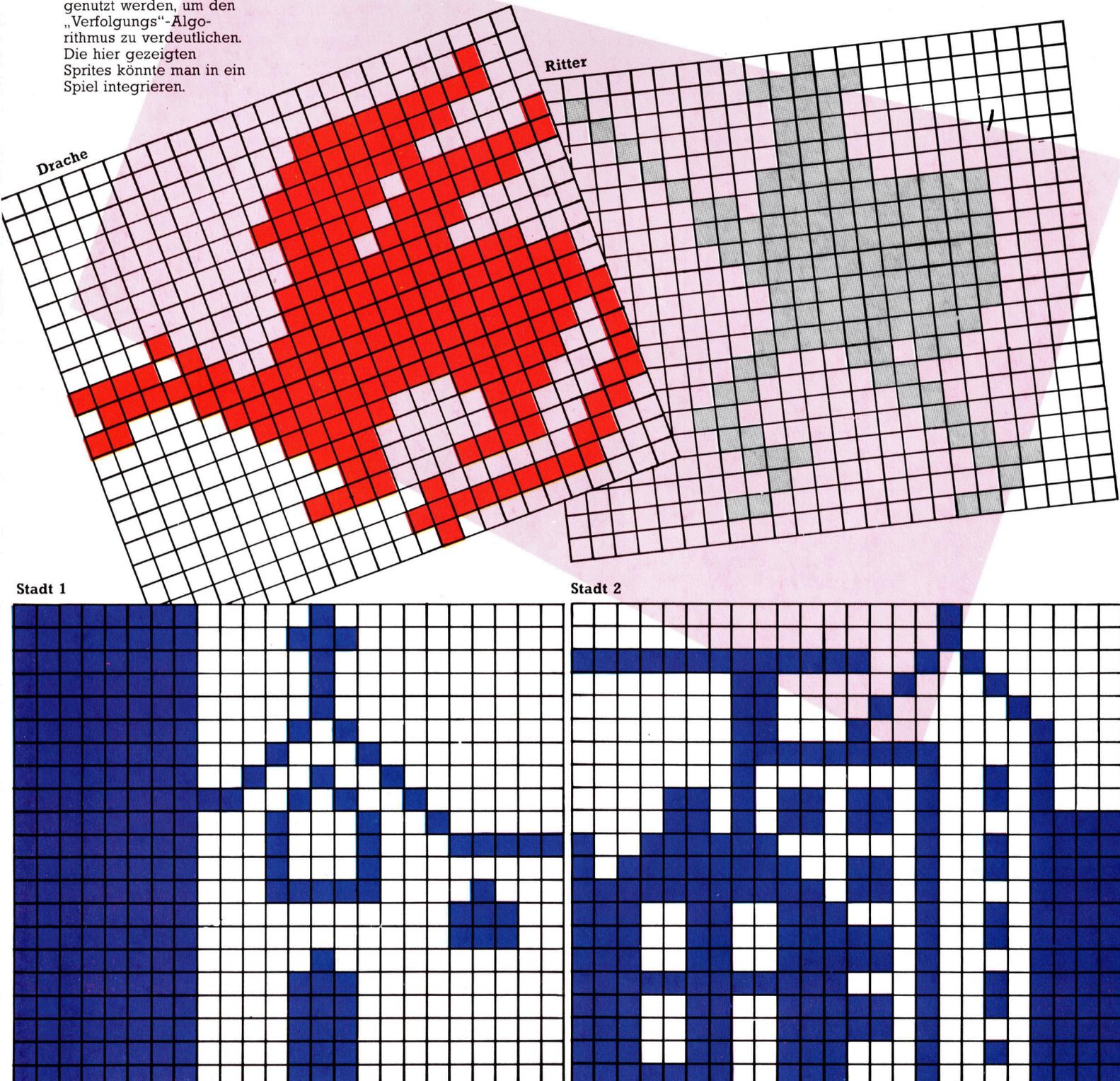
Zwei Beispiele (Ritter und Drache), bei denen die Sprite-Möglichkeiten des Commodore-LOGO genutzt werden, um den „Verfolgungs“-Algorithmus zu verdeutlichen. Die hier gezeigten Sprites könnte man in ein Spiel integrieren.

## LOGO-Dialekte

Weder das Spectrum- noch das Apple-LOGO verfügen über Sprites.

Für Atari-Besitzer gilt:

- 1) Nur vier Sprites stehen zur Verfügung.
- 2) Statt SETSHAPE wird SETSH verwendet.
- 3) Der Sprite-Editor ist Bestandteil des Basisbefehlssatzes. Durch das Drücken der Leertaste füllt oder löscht man ein Pixel.





### Vier Käfer

So erfolgt die grafische Lösung der zuvor genannten geometrischen Aufgabe. Der Algorithmus erzeugt eine einwärts gerichtete Spirale, deren vier Arme jeweils eine „Verfolgungs“-Richtung darstellen.

```

TO BUGS
  SETUP
  MOVE.BUGS
END

TO SETUP
  DRAW
  FULLSCREEN
  TELL 0
  HT
  PU
  SETXY (- 100) (- 100)
  SQUARE 200
  POSITION 1 (- 100) (- 100)
  POSITION 2 (- 100) 100
  POSITION 3 100 100
  POSITION 4 100 (- 100)
END

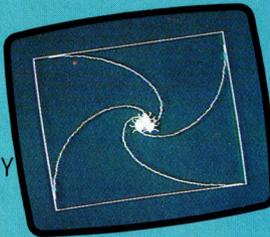
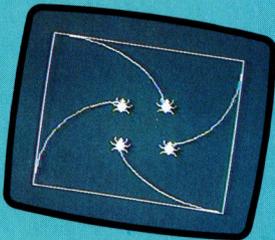
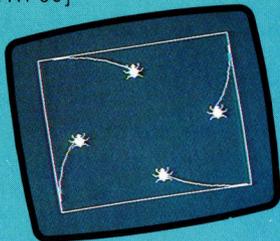
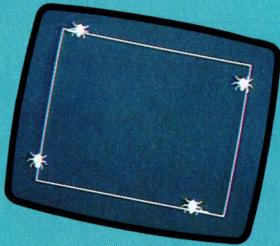
TO SQUARE :SIDE
  PD
  REPEAT 4 [FD :SIDE RT 90]
  PU
END

TO POSITION :NO :X :Y
  TELL :NO
  SETSHAPE 3
  PU
  SETXY :X :Y
  PD
  ST
END

TO MOVE.BUGS
  FOLLOW 1 2
  FOLLOW 2 3
  FOLLOW 3 4
  FOLLOW 4 1
  MOVE.BUGS
END

TO FOLLOW :A :B
  TELL :B
  MAKE "X XCOR
  MAKE "Y YCOR
  TELL :A
  SETH TOWARDS :X :Y
  FD 10
END

```



### Projekt „Lunar Lander“

Dieses Programm zeigt eine mögliche Lösung der im letzten Teil des Kurses gestellten Aufgabe.

```

TO LAND
  SETUP PLAY
END

TO SETUP
  DRAW DRAW.PLATFORM SET.ROCKET
END

TO DRAW.PLATFORM
  PU SETXY (- 20) (- 60) PD SETXY 20 (- 60)
  PU
END

TO SET.ROCKET
  SETXY 0 120 MAKE "VEL 0 MAKE "FUEL 50
END

TO PLAY
  COMMAND MOVE
  IF YCOR < - 53 THEN BOOM STOP
  GRAVITY FUEL.REPORT PLAY
END

TO COMMAND
  IF READKEY = "F THEN BURN
END

TO READKEY
  IF RC? THEN OUTPUT RC
  OUTPUT "
END

TO BURN
  IF :FUEL > 0 THEN MAKE "VEL :VEL + 0.5 MAKE
  "FUEL :FUEL - 1
END

TO MOVE
  SETY YCOR + :VEL
END

TO BOOM
  IF :VEL > (- 1) THEN PRINT [YOU LANDED
  SAFELY, CONGRATULATIONS] STOP
  PRINT [THE IMPACT KILLED ALL THE CREW!]
END

TO GRAVITY
  MAKE "VEL :VEL - 0.2
END

TO FUEL.REPORT
  (PRINT "FUEL :FUEL)
END

```

### Übungen

Entwickeln Sie eine Prozedur mit drei Käfern, die sich in den Ecken eines Dreiecks befinden.



# Aufregende Spiele

**Simulationsprogramme machen Spaß und helfen beim Lernen schwieriger Zusammenhänge.**

Ursprünglich als Arcadespiele entwickelt, gibt es mittlerweile ähnliche Spielprogramme für Heimcomputer. Sie dienen nicht nur zur Unterhaltung, sondern werden auch in modernen Schulen im Unterricht eingesetzt. Im folgenden werden drei Arten von Simulations-Software vorgestellt.

## Winds

Die Firma Longmans bietet für den Acorn B ein Programm an, das Sie zum Kapitän eines Segelschiffes werden läßt. Sie sehen auf dem Bildschirm die Weltkarte mit einem kleinen Punkt – Ihr Schiff. Sie bestimmen den Start- und den Zielhafen sowie den Tag, an dem Sie auslaufen. Die Fahrt des Schiffes hängt von den vorherrschenden Winden ab. Deren Richtungen und Stärken werden unten auf dem Bildschirm angezeigt. Hier ist auch die Schiffsposition in Längen- und Breitengraden abzulesen, außerdem die Windzone (z. B. Westdrift), das Datum, die bereits zurückgelegten Seemeilen sowie die Gesamtentfernung.

Nehmen Sie zum Beispiel die direkte Route von London nach Rio de Janeiro, und laufen Sie am 1. Januar aus. Mit Südkurs nutzen Sie gut die atlantischen Westwinde, bis Sie zum Äquator kommen. Dort liegen Sie in den „Doldrums“ drei Tage in der Flaute. Unter Umständen brist ein Südwestwind auf, doch wie segelt man gegen den Wind? Sie müssen kreuzen, erst nach Süden und dann nach Westen, bevor Sie 207 Tage später nach 15 480 Kilometern in Rio einlaufen.

Andere Reisen sind weit gefährlicher: Hurrikane, Eisberge oder Schiffbruch sind nur einige der Gefahren, denen Sie begegnen können. Dieses Simulationsprogramm kann vielfältig eingesetzt werden. Auf der einen Seite kann es Kindern erklären, wie ein Kompaß arbeitet. Auf der anderen Seite lernen auch ältere Schüler die Windzonen kennen und bekommen ein Gefühl für eine optimale Routenplanung.

## Flight Simulation

In diesem Spiel sind Sie der Pilot eines kleinen Flugzeuges. Der Bildschirm bietet Ihnen einen Blick aus dem Pilotenfenster und zeigt das Instrumentenbrett. Hier sind zahlreiche Schalter, Anzeigen und Lampen zu beobachten, um das Flugzeug erfolgreich fliegen zu können. Der Steuerknüppel wird durch die vier Cursortast-

ten bedient. Die anderen Funktionen wie Fahrwerk, Landeklappen und Motorleistung werden über weitere Tasten eingegeben.

Sie müssen nicht unbedingt vom Boden aus starten, sondern können den Flug gleich in der Luft beginnen. So gewöhnen Sie sich leichter an die Steuerung. Wenn Sie wissen wollen, wo sich das Flugzeug befindet, können Sie eine Karte mit den Funkfeuern und den Landebahnen einblenden. Am einfachsten ist die Navigation, wenn Sie ein bestimmtes Funkfeuer einstellen und so lange den Kurs korrigieren, bis Sie das Signal des Funkfeuers, das auf der RDF-Anzeige durch einen blinkenden Punkt dargestellt wird, am oberen Rand der Kreisskala sehen. Sie brauchen dann nur geradeaus zu fliegen, bis Sie das Funkfeuer erreicht haben. So gelangen Sie zum Flughafen und können nun den Landeanflug einleiten.

Dabei müssen Sie genau die Landebahn anfliegen und die richtige Geschwindigkeit, die Höhe und den Anstellwinkel einhalten.

## Physiological Simulation

In diesem Programm haben Sie die Aufgabe, den Körper 50 Minuten lang am Leben zu halten. Simuliert werden die verschiedenen physiologischen Veränderungen (Wasserverlust, Anstieg der Körpertemperatur), die eintreten, wenn der Körper einer bestimmten Belastung ausgesetzt ist. Zuerst müssen Sie Alter und Geschlecht der Person eingeben und danach die Aktivität, die Sie simulieren wollen.

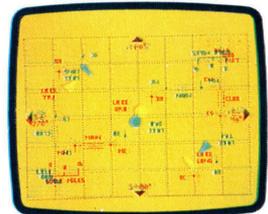
Sie müssen hierbei das Atemvolumen, die Atemfrequenz und das Schwitzen steuern. Nach Wahl der Anfangswerte, zum Beispiel 2,5 Liter Atemvolumen, 15 Atemzüge pro Minute sowie einer Schweißmenge von drei Gramm pro Minute, beginnt die Simulation.

Auf dem Bildschirm sehen Sie nun fünf Kurven verschiedener Körperfunktionen und eine Uhr. Die Kurven zeigen, wie gut der Körper im Laufe der Zeit die gewählte Aktivität bewältigt. Sie haben die Aufgabe, jedem gefährlichen Anwachsen der Kurven zu begegnen. Steigt zum Beispiel die Körpertemperatur zu stark an, können Sie die Aktivität für eine kurze Zeit unterbrechen und die Schweißmenge erhöhen, um die Temperatur zu senken. Gelingt Ihnen das nicht, und die Kurve überschreitet die letzte Warnmarkierung, erhalten Sie unter Umständen die harte Diagnose: „Die Person ist tot.“ Dieses Programm wird von Heinemann Software für den Acorn B angeboten.

Winds



Flight Simulation



Physiological Simulation





# Die Stimme seines Herrn

**Spracherkennungssysteme werden zunehmend kommerziell eingesetzt. Die Speicherkapazität der Computer bildet jedoch eine Barriere, die die Einsatzmöglichkeiten beschränkt.**

**D**er Einsatz von Computern ist nur dann sinnvoll, wenn von „außen“ Daten und Befehle eingegeben werden können. Normalerweise treten wir über die Tastatur (oder über eine Maus oder einen Joystick) mit einem Heimcomputer in Verbindung, müssen für die Kommunikation allerdings eine künstliche Sprache einsetzen. Befehle wie CLS, DIRECTORY, RUN, LOAD und SAVE haben zwar für das Betriebssystem eine genaue Bedeutung, sind mit Sicherheit aber nicht „natürlich“.

Die übliche Kommunikation des Menschen läuft über die Sprache. Könnte der Computer gesprochene Befehle erkennen – selbst wenn sie die gleichen Sprachelemente wie die Tastatur enthielten –, ließe er sich einfacher bedienen. Bevor ein Computer jedoch gesprochene Worte „verstehen“ kann, muß er die eingehenden Klänge analysieren und von der analogen in die digitale Form wandeln. Erst dann kann er sie bearbeiten. Doch obwohl sich die elektronischen Voraussetzungen dafür leicht herstellen lassen, ist die Sprache selbst ein großes Hindernis, da sie aus äußerst komplexen Klängen besteht.

Der Traum von unmittelbarer und vollständiger Spracherkennung (wie sie von dem Computer HAL im Film „2001 – Odyssee im Welt-

raum“ simuliert wurde) läßt sich in nächster Zukunft mit Sicherheit nicht verwirklichen. Und es ist fraglich, ob sie überhaupt in diesem Umfang realisiert werden kann. Eine Schreibmaschine mit Spracherkennung ist ebenfalls noch in weiter Ferne, obwohl die technischen Grundlagen dafür wie auch für den „verständigen“ Computer bereits existieren. Beide Systeme sind jedoch nicht billig, da es bei der Spracherkennung eine Hauptschwierigkeit gibt: Worte, die gleich klingen, nehmen je nach Zusammenhang unterschiedliche Bedeutungen an. Die Verarbeitungstechnik, mit der dieses Problem gelöst werden kann, ist noch nicht zu akzeptablen Preisen verfügbar.

Solche Systeme wurden bereits entwickelt. Aber die Menge der Wörter, die ein Computer gleichzeitig versteht, verkleinert sich, wenn sich die Anzahl der Sprecher, die er erkennen kann, erhöht. Ein typisches, auf mehrere Sprecher ausgerichtetes System unterscheidet mit einer Trefferquote von 85 bis 90 Prozent zwischen 20 und 30 Wörtern.

Die Einsatzmöglichkeiten für Spracherkennungssysteme sind enorm. So verwendet z. B. die Bundespost ein System zur Unterstützung der Briefsortierung, und auch in der Steuerung von Militär- und Zivilflugzeugen finden sich

## Sprachelemente

Bei einer Methode der Spracherkennung wird das eingehende Signal digitalisiert. Wirksamer ist eine andere Methode, bei der das Signal von mehreren unabhängig arbeitenden Schaltungen auf stimmhafte Klänge (z. B. Vokale), Zischlaute (z. B. s,f,t) und kurze Pausen (zwischen zwei Silben) untersucht wird. Diese Filter liefern eine Folge von Nullen und Einsen, die der Computer mit gespeicherten Wortmodellen vergleicht und das Modell mit der größten Übereinstimmung als Eingabe betrachtet.

Stimmhaft

Ruhe

Zischlaut





viele Anwendungen. Bei all diesen Anwendungen können gleichzeitig nur etwa 20 Wörter erkannt werden. Diese geringe Anzahl bedeutet jedoch nicht, daß das Gesamtsystem begrenzt ist. Mit den ersten 20 Wörtern wählt der Anwender aus einem „Menü“ ein Untermenü aus, das wiederum 20 Möglichkeiten zur Verfügung stellt usw. Nur wenn die gesamte Befehlsfolge eindeutig erkannt wurde, setzt der Computer die entsprechenden Abläufe in Bewegung. Im Fall der Postsortierung kann die erste Stufe aus der Angabe des Bundeslandes bestehen, die nächste Stufe aus dem Bezirk, dann der Stadt etc. Da der Brief nur nach Angabe der letzten Information auf den Weg geschickt wird, steht hier ein System mit hoher Zuverlässigkeit zur Verfügung.

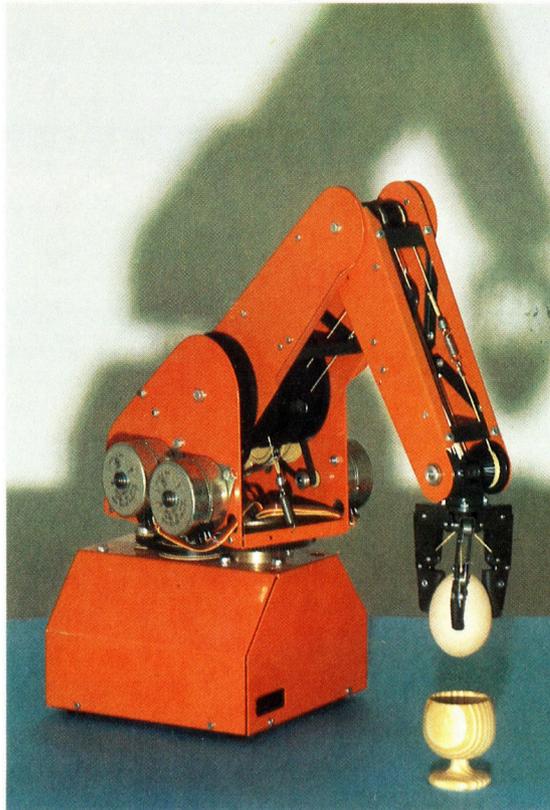
Spracherkennung funktioniert normalerweise nach zwei Methoden. Bei der „einfachen“ Methode wird die Sprache durch einen Analog/Digital-Wandler gesandt und danach die Computeranalyse durchgeführt. Diese Methode hat eine Reihe von Nachteilen, wobei die dafür benötigte Zeit das größte Problem darstellt. Systeme, die nach dieser Methode arbeiten, benötigen bis zu drei Sekunden für die Erkennung eines Wortes.

Bei der zweiten Methode wird das zu verarbeitende Signal elektronisch aufbereitet, bevor der Computer es analysiert. Dadurch erhält der Rechner bereits komprimierte Informationen über Frequenzgang, Tonhöhe, Tonstärke etc. Ähnlich wie sich bei einer HiFi-Anlage mit dem Klangregler z. B. nur die Baßtöne verstärken lassen, werden in diesem Fall Frequenzen mit Hilfe von Filtern in einzelne Stufen zerlegt, deren Höhe dann einzeln festgestellt werden kann. Da diese elektronische Aufbereitung in der gleichen Zeit abläuft, in der das Wort gesprochen wird, funktioniert diese Methode der Spracherkennung beinahe zeitgleich.

### „Lernprozeß“

Sollen die Signale über einen Analog/Digital-Wandler laufen und die Daten in digitalisierter Form analysiert werden, müßten mehrere Computer die Daten gleichzeitig bearbeiten, um ähnliche Antwortzeiten erreichen zu können. Die Aufbereitungsmethode befindet sich zwar noch im Entwicklungsstadium, und es gibt auch noch keinen kommerziell verwertbaren Prototyp, die Richtung scheint jedoch vielversprechend zu sein.

Ist die Information über Frequenzen, Tonhöhe, Tonstärke etc. des Originalklanges erst einmal vorhanden, geschieht die eigentliche Erkennung über den Vergleich der augenblicklichen Werte mit einer Anzahl Modellwerte, die im Computer gespeichert sind. Diese Modelle werden durch einen „Lernprozeß“ von dem Spracherkennungssystem generiert. Dabei werden die Worte, die das System später erkennen soll, nacheinander gespro-



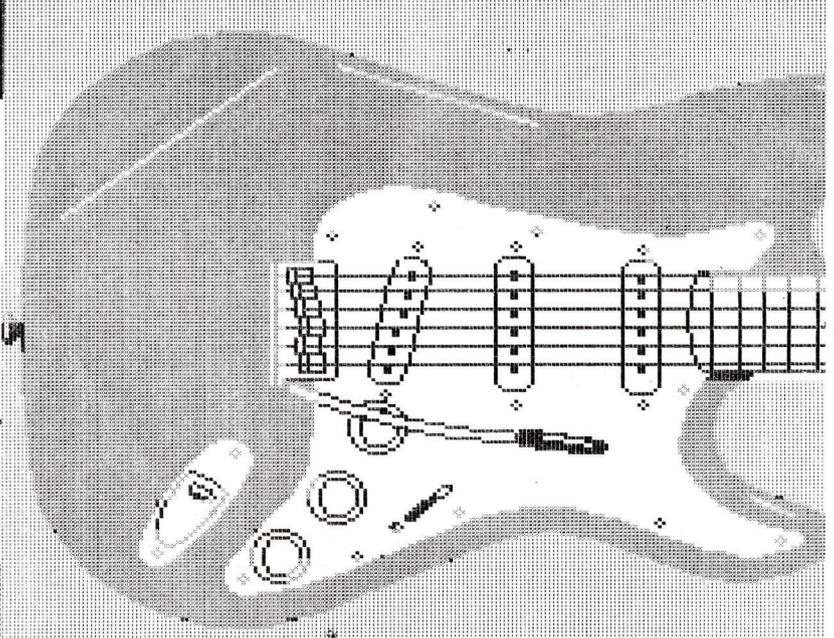
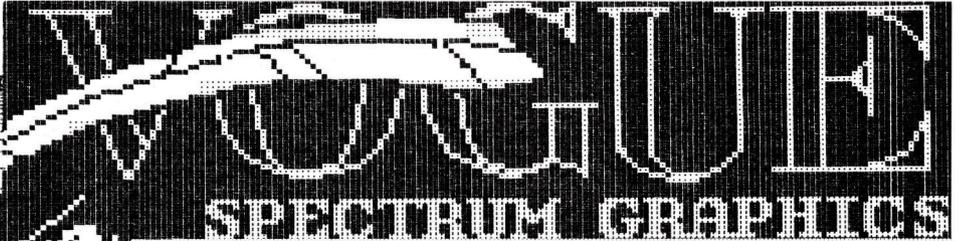
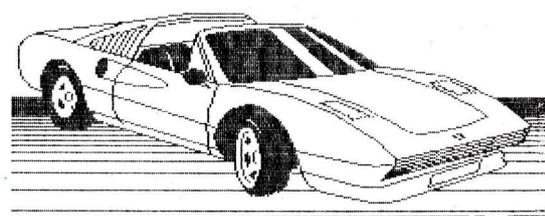
**Eine der neuesten Anwendungen der Spracherkennung trägt den Namen „Limited Environment“ und steuert einen Computer und einen Roboterarm. Über ein Mikrofon gibt der Anwender den Befehl „SETZ DAS EI IN DEN EIERBECHER“. Der Computer interpretiert die Befehle und sucht sich aus seinem Speicher die entsprechenden Gegenstände (z. B. das Ei) heraus und setzt sie an die angegebene Stelle.**

chen und das Ergebnis im Computer in digitaler Form gespeichert. Die Wortfolge wird mehrfach wiederholt, wobei der Computer bei jedem Durchlauf die neuen Werte mit den alten vergleicht und einen „Durchschnittswert“ errechnet. Dieser Vorgang wird so lange wiederholt, bis der Computer eine Vielzahl von Stimmen erkennen kann.

In der Zukunft wird man Spracherkennungssysteme in vielen Bereichen einsetzen können; sie eignen sich ausgezeichnet als „Einstiegssystem“ für komplexe Softwarepakete (z. B. Datenbanken), um etwa die anfängliche Menüwahl zu steuern.

Auf dem Markt werden inzwischen Spracherkennungseinheiten angeboten, die sich an Heimcomputer anschließen lassen. Diese Geräte sind allerdings recht primitiv. Systeme wie „Big Ears“ und das „Speech Lab“ von Heuristic Inc. benötigen eine große Verarbeitungskapazität, können nur wenige Wörter erkennen und lassen sich nur von einer Person „etwas sagen“. Bevor Spracherkennung wirklich einsetzbar ist, muß sie die Worte jedes Menschen unabhängig vom Dialekt oder Akzent verstehen können. Die Grenze wird heute noch von der Größe des Speichers bestimmt, in dem die Wortmodelle untergebracht sind. Durch die Verwendung der Videoplatte, die einen Satz von Standardmodellen enthält, könnte sich allerdings eine interessante Möglichkeit zur Lösung dieses Problems ergeben. Mit dieser Methode würde im Arbeitsspeicher kein Platz belegt werden, während die Verzögerung kaum bemerkt würde.

Ferrari



Dies sind Beispiele, welche Arten von Ausdrucken bestimmte Matrixdrucker herstellen können. Jede Nadel des Druckkopfes wird einzeln gesteuert, so daß zufriedenstellende Bildmuster gedruckt werden können. Die hier gezeigten Beispiele entstanden unter Verwendung der „Paintbox“ von Print'n'Plotter Products.

## Punkte, Punkte

Die Fehlersuche wird mit einer gedruckten Ausgabe der Listings um vieles leichter, und die Textverarbeitung ist ohne Drucker undenkbar.

Der Kauf eines Druckers ist, aufgrund des vielfältigen Angebotes, ebenso schwierig wie der eines Computers. Die richtige Wahl ist für den unerfahrenen Käufer dementsprechend schwierig. Die erste Entscheidung, die zu treffen ist, betrifft die Art des Druckers. Soll es ein Matrixdrucker, ein Typenraddrucker, ein Thermo- oder Tintenstrahldrucker sein? Typenraddrucker liefern die beste Schriftqualität, liegen dafür aber auch in den gehobeneren Preisklassen. Sie eignen sich gut zur Textverarbeitung. Der Matrixdrucker dagegen arbeitet schneller, ist ideal für Listingausdruck, grafikfähig und preisgünstiger.

Einfache Matrixdrucker sind schon ab 500 Mark zu bekommen, komfortablere Geräte dagegen können 3000 Mark und mehr kosten. Wesentliche Beurteilungs-Merkmale sind die Druckgeschwindigkeit und die Qualität des erzeugten Schriftbildes. Dazu kommen noch eventuelle Extras wie Proportionalsschrift, ver-

schiedene Schriftgrößen usw. Bei der Proportionalsschrift wird die Breite des einzelnen Buchstabens berücksichtigt und das nächste Zeichen entsprechend plazierte, so daß ein gleichmäßiges Schriftbild entsteht: Ein „i“ braucht zum Beispiel weniger Platz als ein „m“. Ob Ihnen diese Besonderheiten einen höheren Kaufpreis wert sind, müssen Sie selbst entscheiden.

Die Druckgeschwindigkeit beeinflusst die Zeitspanne, in der der Computer für den Ausdruck in Anspruch genommen wird. Währenddessen kann der Rechner nicht für andere Aufgaben genutzt werden. Die Druckgeschwindigkeit wird in Z/s (Zeichen je Sekunde, in Englisch cps = characters per second) angegeben. Bei einem Programm, für das ein mit 200 cps arbeitender Drucker beispielsweise eine Minute benötigt, würde ein einfacherer Drucker mit 30 cps mehr als sechs Minuten arbeiten müssen – und den Computer auch so



lange blockieren. Diese unfreiwillige Arbeitspause läßt sich mit Hilfe eines Zwischenspeichers vermeiden. Ein solcher Buffer ist nichts anderes als eine Platine mit RAM-Chips, die zwischen Computer und Drucker geschaltet ist. Der Computer gibt die auszudruckenden Daten in der ihm eigenen hohen Geschwindigkeit in den Zwischenspeicher und ist dann für andere Aufgaben frei. Der Zwischenspeicher („Buffer“) übermittelt anschließend die auszudruckenden Zeichen in einer dem Drucker angepaßten Geschwindigkeit weiter. In komfortablen Druckern ist ein solcher Speicher bereits eingebaut.

## Die Geschwindigkeit

Mit den vom Hersteller angegebenen Werten für die Druckgeschwindigkeit ist es wie mit den Angaben über den Benzinverbrauch bei den Autos: Sie sollten nicht wörtlich genommen werden, denn sie beruhen auf idealen Bedingungen und haben mit der Realität nur selten etwas gemein. Beispielsweise wird die Druckgeschwindigkeit beim Drucken einer Zeile mit immer dem gleichen Zeichen gemessen. Unterschiedliche Schriftzeichen, Wagenrücklauf und Zeilenvorschub setzen natürlich die Druckgeschwindigkeit herab. In der Praxis heißt dies, daß ein Drucker, dessen „Spitzengeschwindigkeit“ mit 160 cps angegeben ist, tatsächlich kaum mehr als 100 Zeichen je Sekunde erreicht.

Die Druckqualität der Matrixdrucker ist unterschiedlich. Sie hängt davon ab, wie viele Nadeln im Druckkopf zur Verfügung stehen. Es gibt Modelle mit sieben Nadeln, aber auch solche mit 16 oder mehr. Der Commodore-Drucker zum Beispiel bildet die Zeichen in Form einer 7x6-Matrix. Der PW1080 von Canon dagegen hat 16 Stifte und erzeugt für jedes Zeichen eine 16x23-Matrix. Da gedruckte Schriftzeichen aus einzelnen Punkten bestehen, wird die Schrift um so klarer sein, je mehr Punkte zum Bilden eines Zeichens verwendet werden. Geht es um reine Programmlistings, kann auf eine hochwertige Druckqualität verzichtet werden. Bei der Textverarbeitung dagegen ist ein

gutes Schriftbild meist sehr wichtig.

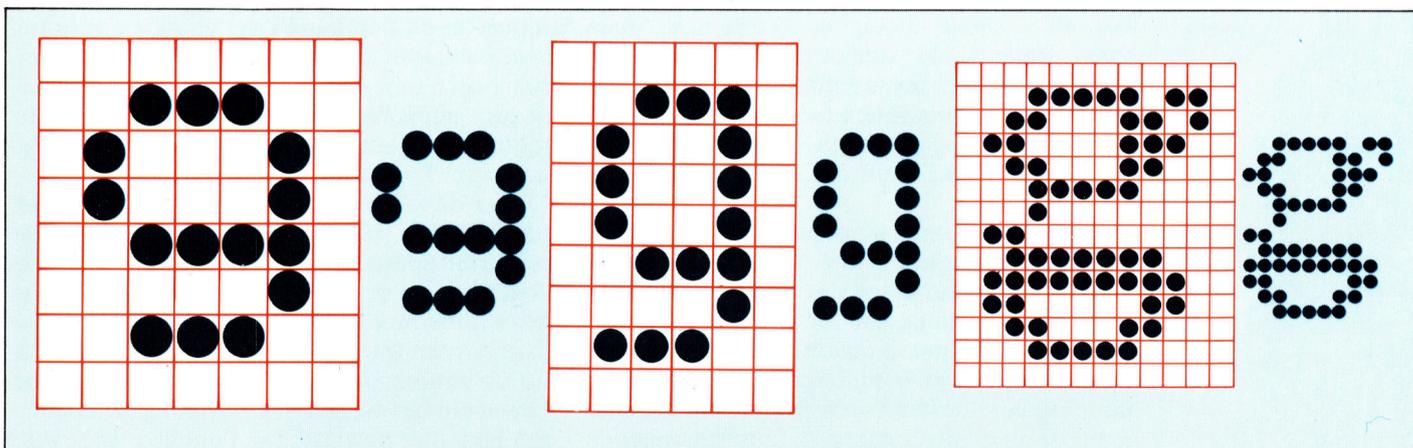
Der Matrixdrucker ist im Grunde genommen nichts weiter als ein Einzweck-Microcomputer; denn er besitzt einen Microprozessor und arbeitet mit ROM und RAM. Aufgrund dieser technischen Voraussetzungen ist es möglich, den Drucker zu programmieren. Dies geschieht durch bestimmte Steuerzeichen oder durch Verstellen des DIL-Schalters (Dual-in-Line) im oder am Gehäuse. Auf diese Weise kann beispielsweise der standardmäßige ASCII-Zeichensatz, der im Drucker gespeichert ist, umdefiniert werden.

Gespernte, halbfette oder fette Schrift und unterschiedliche Zeilenabstände sind weitere Extras. Der FX80 von Epson beispielsweise besitzt zahlreiche solcher Besonderheiten, wie kursive Schriftzeichen, automatisches Unterstreichen oder Proportionaldruck, die ihn zu einem der vielseitigsten Drucker machen. Die Epson-Drucker sind mittlerweile zu einer Art „industriellem Standard“ geworden. Das hat dazu geführt, daß viel Software so entwickelt wird, daß sie nur mit einem Epson-Drucker arbeiten kann. Dies ist besonders interessant, da die Drucker der verschiedenen Hersteller keineswegs kompatibel sind.

Zu den weiteren Aspekten, die beim Kauf eines Druckers bedacht werden sollten, gehört die Zuverlässigkeit. Ein „Billig-Drucker“ mag sich zwar für das gelegentliche Ausdrucken von Programmen als brauchbar erweisen, ist aber denkbar ungeeignet für den Dauereinsatz im Büro. Ein anderer Faktor ist der Lärm. Sollten Sie ein ausgesprochener „Nachtarbeiter“ sein, kann ein laut hämmernder Drucker zur Katastrophe werden. Auch die Papierzuführung ist der Betrachtung wert. Zwar können alle Matrixdrucker Endlospapier verarbeiten, für Einzelblätter wird jedoch ein Rolleneinzug benötigt.

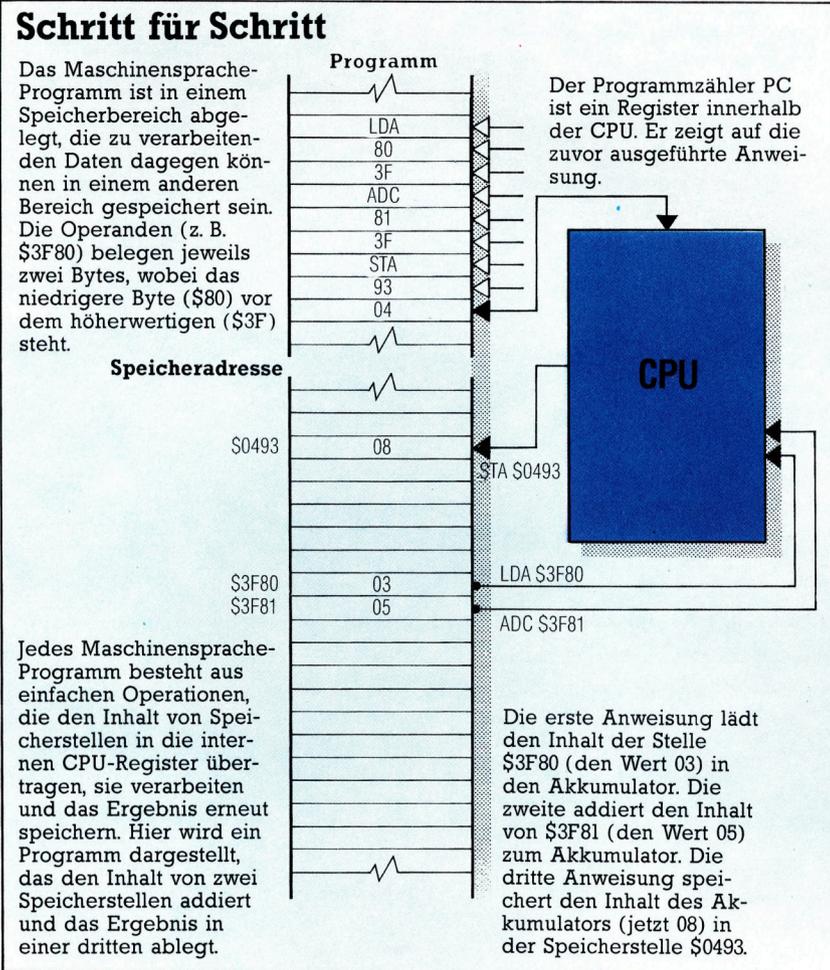
Ganz zum Schluß sollte die wichtigste Frage nicht vergessen werden: „Paßt der Drucker auch an meinen Computer?“ Die meisten Matrixdrucker haben entweder eine Centronics- oder eine RS232-Schnittstelle. Bei Anschlußschwierigkeiten hilft unter Umständen eine Adapter-Schnittstelle.

**Diese Druckmuster zeigen die Qualitätsunterschiede zwischen verschiedenen Matrixdruckern. Die Druckqualität hängt im wesentlichen von der Anzahl der „Nadeln“ ab, die sich im Druckkopf befinden – je mehr Druckstifte, desto besser die Schriftzeichen.**



# Maschinensprache

**Der Sprung von BASIC zur Maschinensprache erfordert ein völliges Umdenken. Als Belohnung winkt jedoch eine enorme Steigerung der Ausführungsgeschwindigkeit und Leistungsfähigkeit der Programme.**



hat kaum etwas mit einer anderen Programmiersprache gemein. Es lohnt sich jedoch, diese Logik zu lernen.

Die Zentraleinheit, Herz eines jeden Computers, versteht nur Maschinensprache. Außerdem kann sie nur ganz simple Funktionen ausführen (beispielsweise Addieren, nicht aber Multiplizieren).

Ein in Maschinensprache geschriebenes Programm besteht aus diesem Grund aus einer Vielzahl von Anweisungen, wobei jede Funktion „von Hand“ aus einfachen Befehlen aufgebaut werden muß. Programmieren in Maschinensprache heißt, individuelle Bits und Bytes im Speicher unter Verwendung einfacher logischer Operationen wie AND, OR, NOT und elementarer Binärarithmetik zu manipulieren.

Dies ist der eine Grund, warum das Programmieren in Maschinensprache so langsam vonstatten geht. Der andere ist die Notwendigkeit, daß der Programmierer genau wissen muß, wo welche Daten im Speicher zu finden sind. Tritt in BASIC der Befehl LET A=5 auf, ist es Sache des BASIC-Interpreters, einen Platz im Speicher zu finden, wo die Variable abgelegt werden kann. Er muß sich ebenfalls immer an diese Stelle „ernnern“, wenn die Variable A verlangt wird. Beim Programmieren in Maschinensprache dagegen muß jede zu speichernde Information eine Adresse erhalten. Und der Programmierer selbst muß dafür sorgen, daß es nicht zum unbeabsichtigten Überschreiben von Daten kommt.

Nun zu den Bestandteilen der Maschinensprache. Alle folgenden Beispiele beziehen sich auf eine 8-Bit-CPU wie die Z80 oder die 6502 (16-Bit-CPU's arbeiten ähnlich, jedoch mit doppelter Bit-Zahl). Die Zentraleinheit ist durch zwei Bus-Leitungen mit dem Speicher verbunden: durch den Adreß-Bus und den Daten-Bus. Es gibt außerdem den Kontroll-Bus zur Übertragung der Zeitsteuersignale. Er wird jedoch nicht vom Programmierer benutzt.

Über den Adreß-Bus ist die CPU in der Lage, mit Hilfe der verschiedenen Bit-Muster jede der 65536 Speicherstellen im Speicherplan auszuwählen. In einem Heimcomputer bestehen diese Speicherstellen aus RAM-, ROM- oder auch besonderen Ein-/Ausgabe-Chips. Will die Zentraleinheit eine Speicherstelle lesen (durch ein Lese-/Schreib-Signal zeigt der Kontroll-Bus die gewünschte Funktion an), wird



der Inhalt des betreffenden Bytes in Form eines 8-Bit-Musters auf den Daten-Bus „gepackt“. In gleicher Weise kann die CPU ein solches Muster in eine ausgewählte Speicherstelle schreiben. Da die Zentraleinheit jedoch nicht „weiß“, wie die RAMs und ROMs verteilt sind, muß der Programmierer die richtige Adresse herausfinden.

Innerhalb der CPU gibt es eine Handvoll „Register“. Das sind individuelle Speicher, die zum zeitweiligen Abspeichern von Ergebnissen, logischen Operationen und binären Funktionen benötigt werden. Die meisten Register entsprechen einer 1-Byte-Speicherstelle. Es gibt aber auch 16-Bit-Register wie den Programmzähler PC (Program Counter), der immer die Adresse des gerade auszuführenden Maschinensprache-Befehls enthält. Man könnte dies mit der Zeilennummer in BASIC vergleichen.

Ein anderes wichtiges Register (diesmal mit 8 Bit) ist der „Akkumulator“. Er kann, wie der Name sagt, Ergebnisse „ansammeln“. Bytes können hinzuaddiert oder abgezogen werden. Es ist tatsächlich das einzige Register, das eine Arithmetik ausführen kann. Ein einfaches Maschinensprache-Programm könnte so beschrieben werden:

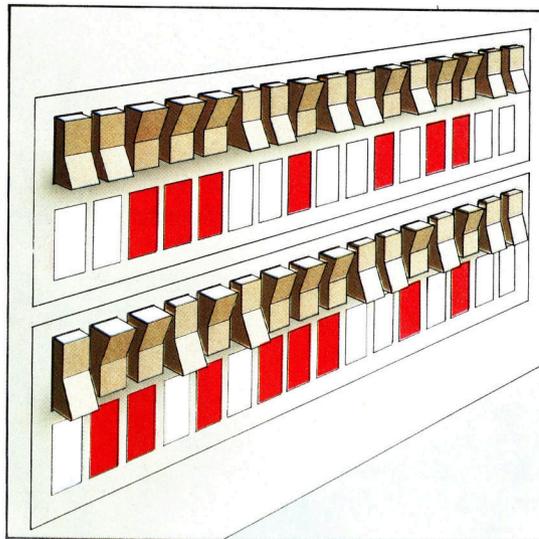
- 1) Lade den Akkumulator mit dem Inhalt der Speicherstelle \$3F80. (Adressen in Maschinensprache werden gewöhnlich hexadezimal geschrieben. Hexadezimalwerte werden üblicherweise durch das Zeichen \$ gekennzeichnet.)
- 2) Addiere zum Akkumulator den Inhalt der Speicherstelle \$3F81. Ist das Ergebnis größer als ein Byte, tritt das "carry bit" (Übertrag-Bit) in Aktion.
- 3) Speichere den neuen Inhalt des Akkumulators (das Ergebnis) in die Speicherstelle \$0493.

Jede dieser Anweisungen entspricht einem Maschinensprache-Befehl, und das Programm würde normalerweise so geschrieben:

```
LDA $3F80 (LoaD Accumulator; Lade
            Akkumulator)
ADC $3F81 (ADd with Carry; Addiere mit
            Übertrag)
STA $0493 (STore Accumulator; Speichere
            Akkumulator-Inhalt)
```

Die in Klammern stehenden Anmerkungen sind, wie die REMark-Kommentare in BASIC, ohne Auswirkung auf den Programmablauf. Jede Zeile beginnt mit dem „Opcode“ (Operationscode), der die Art der Operation angibt. Es folgt der „Operand“ mit Einzelheiten über die benötigten Daten. Normalerweise kann eine CPU mehrere Dutzend Opcodes verarbeiten, wobei eine Operation jeweils nur ein Byte belegt.

Ein Opcode kann daher nur Werte zwischen 0 und 255 (\$00 bis \$FF hexadezimal) anneh-



**Lichterketten**

Die Idee für die riesigen Tafeln mit blinkenden Lichtern, wie sie in Filmen an Computern zu sehen sind, resultiert aus den Fronttafeln, die früher bei vielen Microcomputern zu finden waren. Sie bestanden aus Reihen von Lampen und Schaltern für die Adreß- und Datenleitungen der CPU. Als Tastaturen noch nicht angeschlossen werden konnten, mußten alle Maschinensprache-Programme auf diese Weise eingegeben werden.

men. Aus Gründen der Lesbarkeit wird jedoch bei der Programmerstellung üblicherweise eine sinngemäße Abkürzung aus drei Buchstaben zur Hilfe genommen, beispielsweise LDA für LoaD Accumulator.

Jeder der drei oben genannten Operanden enthält eine Hex-Zahl zwischen \$0000 und \$FFFF und belegt damit bis zu zwei Speicherstellen. Manche Operanden sind jedoch nur ein Byte lang, und einige Opcodes haben überhaupt keine Operanden. Das hier gezeigte kurze Programm würde demnach nur neun Bytes beanspruchen, ausgenommen die drei Speicherstellen \$3F80, \$3F81 und \$0493, die adressiert werden. Für dieses einfache Beispiel könnte auch das folgende BASIC-Programm verwendet werden. Es würde jedoch 50 Bytes belegen und hundertmal langsamer laufen, weil der Interpreter erst einmal alles übersetzen muß:

```
10 A=PEEK (16256)
20 A=A+PEEK (16257)
30 POKE 1171,A
```

LDA	<b>LoaD Accumulator</b> überträgt den Inhalt einer einzelnen Speicherstelle (Byte) in das interne Akkumulator-Register.
STA	<b>STore Accumulator</b> führt den gegenteiligen Vorgang zu LDA aus.
ADC	<b>ADd with Carry</b> addiert den Inhalt einer Speicherstelle zum Inhalt des Akkumulators und erzeugt, wenn notwendig, ein Übertrag-Bit.
SBC	<b>SuBtract with Carry</b> ist die Gegenfunktion von ADC.
JMP	<b>JuMP</b> überträgt eine Operation zu einer neuen Stelle (ähnlich wie GOTO in BASIC).

Links finden Sie einige Opcode-Beispiele, die von den meisten Microprozessoren verarbeitet werden können.

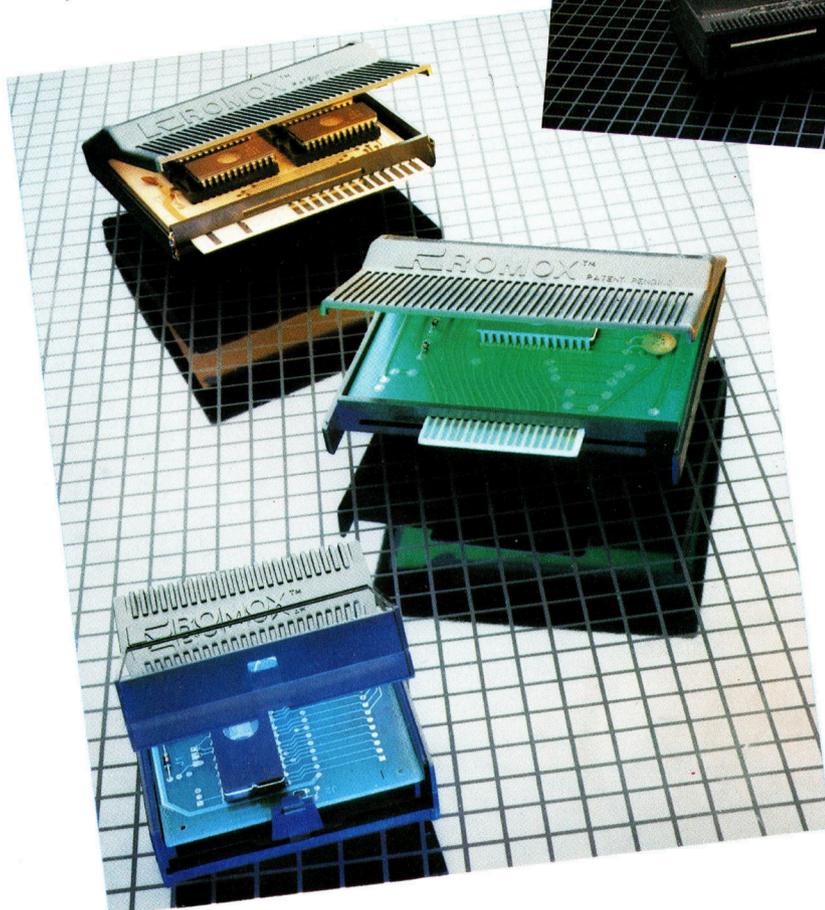


# Einfach überspielen!

**Eine preisgünstige Alternative für den Erwerb der relativ teuren Software wird von der Firma Romox in Amerika angeboten.**

**A**ufgrund des elektronischen Verkaufssystems Romox brauchen Computershops nicht mehr Hunderte Module bereitzuhalten, um alle Kundenwünsche für die unterschiedlichen Rechnertypen zu erfüllen. Romox ist ein Microcomputer, der im Laden über das Telefonnetz mit einem Großrechner verbunden ist. Dieser kann dem Microcomputer unzählige verschiedene Spiele übermitteln, die zunächst auf einer Winchester-Festplatte zwischengespeichert werden.

Der Kunde kauft zuerst ein leeres, für seinen Computer passendes Romox-Modul und steckt dieses in einen der neun Slots des Romox-Micros. Die Slots sind auf die unterschiedlichen Computer-Systeme abgestimmt. Auf Knopfdruck erscheint auf dem Bildschirm des Romox-Computers eine Liste der zur Zeit angebotenen Spiele. Der Kunde gibt nun die Nummer des gewählten Programms ein, und in weniger als einer Minute ist das Modul programmiert. Sollte das Spiel mit der Zeit langweilig werden, gehen Sie einfach mit dem Mo-



dul in den Laden zurück, wählen ein anderes Programm aus und lassen es auf derselben Cartridge mit dem Romox-System neu programmieren. Der Preis für die auf diese Weise zur Verfügung gestellte Software bezieht sich weitgehend auf deren Aktualität.

Das Anwachsen elektronisch verkaufter Software könnte bedeuten, daß das Laden eines Programms so einfach wird wie das Aufzeichnen einer Rundfunksendung per Tonband. Es ist dann möglich, den Heimcomputer über das Telefon direkt an einen Zentralrechner anzuschließen, von dem die Programme übermittelt werden.

In Virginia (USA) gibt es schon einen derartigen Service mit dem Namen „The Source“ (Die Quelle). Die Teilnehmer können mit ihren Computern untereinander in Verbindung treten und Programme über ein Telefonmodem laden. Jeder besitzt ein Paßwort, das den Zugang zu allen Informationen eröffnet.

In England entwickelte die BBC ihren „Telesoft-Service“. Hier werden die Programme an die Teilnehmer mit einem speziellen Adapter mit Hilfe der Fernsehanlage übermittelt.

# Fachwörter von A bis Z

## ALU = ALU

Die **A**rithmetisch **L**ogische Einheit (**U**nit) ist das Rechenwerk sämtlicher Computer. Die ALU ist Bestandteil der Rechner-Zentraleinheit (**C**entral **P**rocessing **U**nit = CPU), das sind bei Heimcomputern meist die Bausteine 6502 (Rockwell), Z80 (Zilog) oder 68000 (Motorola).

Wie der Name sagt, führt die ALU arithmetische und logische Operationen aus, z. B. Addition und Subtraktion sowie die Bedingungsabfragen mit UND, ODER, NICHT, aber jeweils nur mit Einzel-Bytes. Während die übrige CPU eigentlich nur damit zu tun hat, Daten zur rechten Zeit an den richtigen Ort zu transportieren, werden sie in der ALU tatsächlich mathematisch-logisch verändert.

## AND = UND

Das AND ist eine der grundlegenden logischen Funktionen, die Ihnen sicher schon vertraut sind. AND verknüpft jeweils zwei logische Variablen. Das Ergebnis wird als „wahr“ oder „falsch“ ausgegeben (im Dualsystem durch 1 bzw. 0 darstellbar). Nur wenn beide Variablen zugleich „wahr“ sind, ist auch das Ergebnis der Operation wahr.

Ein AND-Gatter ist eine Schaltung, mit der die AND-Funktion elektronisch realisiert wird. Ihr Heimcomputer enthält davon zahllose, und zwar integriert in größere Bausteine, beispielsweise die CPU. Aber auch beim Programmieren in der Maschinensprache gibt es meist einen AND-Befehl, der den Akkumulator-Inhalt mit einem beliebigen andern Speicherbyte verknüpft. Im Ergebnisbyte erhält eine Binärstelle genau dann den Wert 1, wenn die Operanden in der gleichen Stelle beide den Wert 1 aufweisen.

In den meisten BASIC-Versionen ist das AND ähnlich verwendbar; durch die Anweisung

```
LET C= A AND B
```

wird das Ergebnis der Variablen von A und B der Größe C zugewiesen. Ist etwa A= 13 (1101) und B= 11

**Hier werden einzelne Fachausdrücke eingehend behandelt. Da bei der Kommunikation mit dem Computer meist die englische Sprache verwendet wird, werden hier zunächst die englischen Begriffe genannt, dann die deutsche Übersetzung. In den Gesamtindex werden sowohl deutsche als auch englische Stichwörter aufgenommen, damit Sie es leichter haben, das von Ihnen Gesuchte zu finden.**

(1011), dann erhält C den Wert 9 (1001). Solche Operationen sind oft sehr nützlich, z. B. bewirkt die Anweisung

```
IF (ZEIGER AND 1) = 1 THEN  
GOTO 1000
```

dann einen Sprung auf Zeile 1000, wenn die Variable ZEIGER eine ungerade Binärzahl enthält.

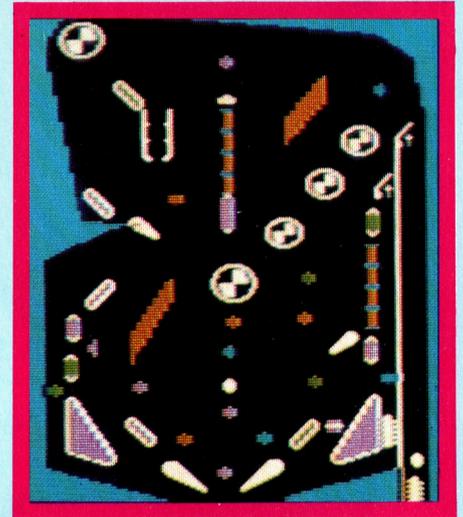
## APL = APL

Im Computer-Jargon gibt es eine Menge eigenwilliger Akronyme – dieses ist kaum zu übertreffen: APL steht für **A** Programming **L**anguage = Eine Programmiersprache! Im Widerspruch zu der schlichten Bezeichnung ist APL eine hochentwickelte Programmiersprache, die es ermöglicht, in einer Zeile APL soviel unterzubringen wie in Dutzenden von BASIC-Zeilen. Trotzdem hat APL eigentlich nur bei Mathematikern und in Hochschulkreisen Anhänger gefunden und ist für Kleinrechner zumeist nicht verfügbar. Ein anderer Grund für die geringe Verbreitung liegt darin, daß bei dieser Sprache einige Sonderzeichen verwendet werden, die es bei den wenigsten Rechnern gibt.

### Bildnachweise

421: Ian Dobbe  
422: Annl Axworthy  
423, 430, 431, 440, 441, 443, U3: Ian McKinnell  
432, 442, 445, 446, 447: Kevin Jones  
439: Liz Dixon

Geeignet für APL ist beispielsweise der Super-PET (eine Variante des Commodore 8000), der einen anderen Prozessor, ein größeres RAM und einen Zeichengenerator besitzt.



## Applications generator = Anwendungsgenerator

Ein Entwicklungssystem, das Sie nur mit den Leistungsvorgaben eines Benutzers füttern müßten, damit es Ihnen selbsttätig das gewünschte Programm erstellt, ist heute schon, zumindest teilweise, realisierbar. Was zur Zeit jedoch als „Programm-Generator“ verkauft wird, genügt nur bescheidenen Ansprüchen, und die erzeugten Programme benötigen wesentlich mehr Rechnerzeit und Speicherplätze als vergleichbare „handgeschriebene“.

Ein „Anwendungsgenerator“ hat da bessere Chancen. Die damit erzeugten Programme sind nicht selbstständig, sondern laufen nur in Verbindung mit dem Entwicklungssystem. Dafür gibt es maßgeschneiderte Programme, die aus effizienten, ursprünglich handgeschriebenen Modulen aufgebaut sind.

Das Verfahren beruht darauf, daß innerhalb eines bestimmten Anwendungsbereichs (seien es nun Abenteuerspiele oder Buchhaltungsprogramme) allen Programmen gewisse Grundfunktionen gemeinsam sind, die im Entwicklungssystem „vorgefertigt“ bereitstehen.

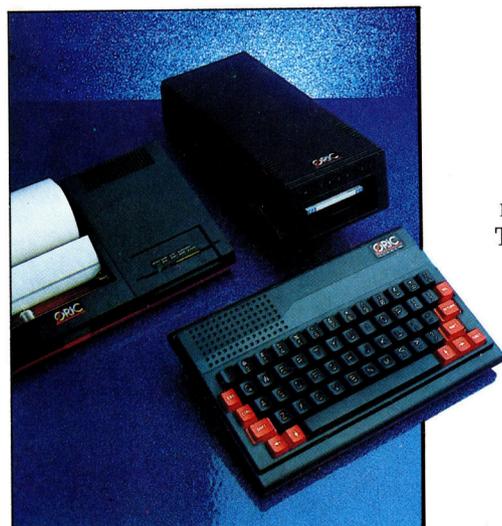
++ Vorschau +++ Vorschau +++ Vorschau +++

# computer kurs Heft 17



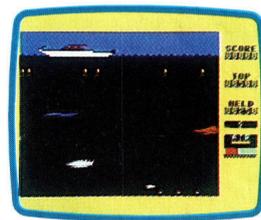
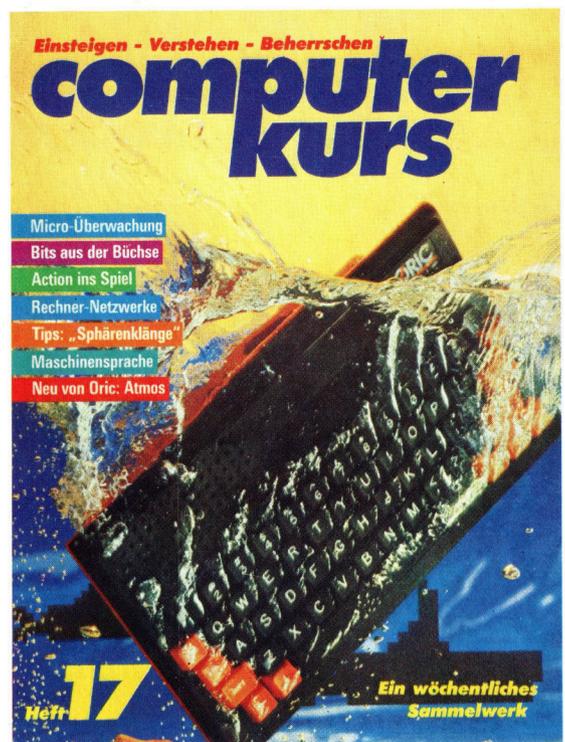
## Netzwerke

Computer können auch über große Entfernungen „zusammenarbeiten“. Entsprechende Geräte und Kommunikationsleitungen sind dafür allerdings die Voraussetzung.



## Nachfolger

Dem Heimcomputer Oric-1 ist nun ein verbessertes Modell gefolgt: Der Oric Atmos. Das Gerät verfügt über eine Qwertz-Tastatur, und das BASIC wurde weiter verbessert.



## Wassersport

Ein spannendes Spiel: Tauchen nach Perlen und Schätzen in der Tiefsee. Aber Gefahren lauern – der Taucher muß Kraken und Riesenmuscheln entgehen.

+++ „Verhexte“ Befehle +++ Bits aus  
der Büchse +++ Spiel mit Sprites +++  
Tips für die Praxis +++ Programmier-  
hilfen +++ BASIC und LOGO +++ Moto-  
rola 68000 +++ Fachwörter von A–Z +++