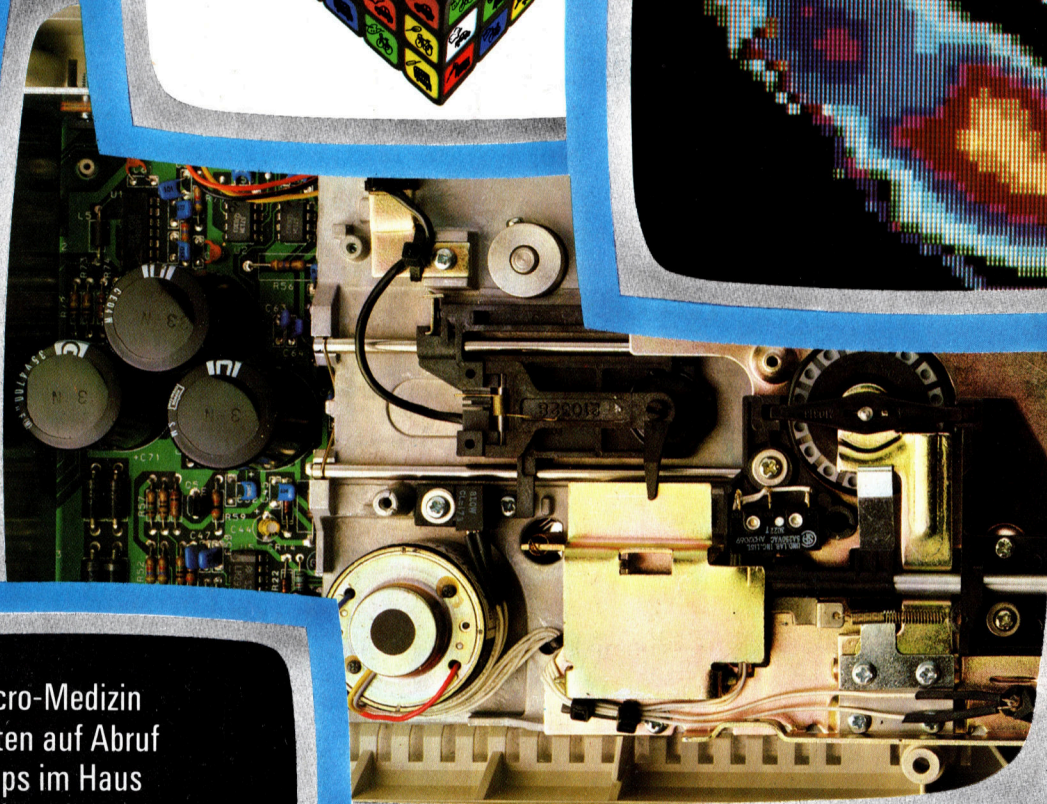
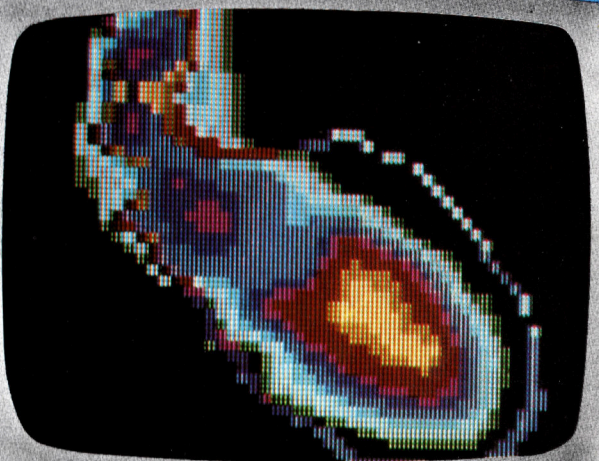


Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer KURS

Heft **5**



- Micro-Medizin
- Daten auf Abruf
- Chips im Haus
- Bodenroboter
- Anschluß per Modem
- BBC Modell B
- Miniaturtechnik

Ein wöchentliches Sammelwerk

**Programmierkurse
BASIC und LOGO**

computer kurs

Heft 5

Inhalt

Computer Welt

Miniaturltechnik auf Sand gebaut 113

Silizium – der Rohstoff für die Chips

Chips im Haus 116

Micro-Medizin 132

Computer-Einsatz in Klinik und Arztpraxis

Hardware

Auf die Perspektive kommt es an 118

Die Gittergrafik der Video-Spielautomaten

BBC Modell B 120

Software

Der Weg zur Lösung 122

Strukturieren mit Flußdiagrammen

Daten auf Abruf 124

Programme für übersichtliche Dateien

BASIC 5

Der BASIC-Kalender 126

Peripherie

Auf zwei Rädern 129

Programmierte Bewegungsabläufe mit Robotern

„Reise“ per Telefon 137

Weltweiter Datentransfer mit Modems

Tips für die Praxis

Kein Anschluß ohne FTZ! 138

Was beim Datenaustausch zu beachten ist

LOGO 5

Bogen- und Listenverarbeitung 134

Bits und Bytes

Wenn 1+1=10 ... 139

Fachwörter auf einen Blick

WIE SIE JEDE WOCHE IHR HEFT BEKOMMEN

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

Zurückliegende Hefte

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

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

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

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

Abonnement

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

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

SAMMELORDNER

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

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

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

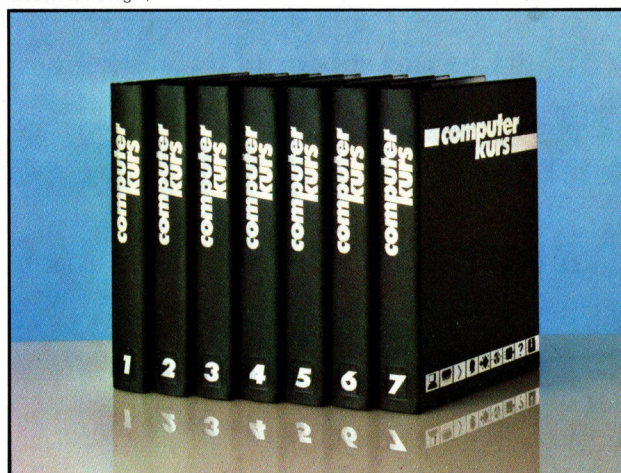
Schweiz: Der Sammelordner kostet sfr 15. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Sammelordner Computer Kurs, und notieren Sie Ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

INHALTSVERZEICHNIS

Heft 84 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, Volker Andrae, Susanne Brandt, Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

Vertrieb: Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1, Tel.: 040/23 40 80





Miniaturtechnik auf Sand gebaut

Silizium ist der Rohstoff, aus dem Chips hergestellt werden. Ein aufwendiger Fertigungsprozeß geht dem Endprodukt voraus.

Silizium ist, neben Sauerstoff, das auf der Erde am häufigsten vorkommende Element. Aus ihm bestehen alle Sandkörnchen der Welt. Die Bedeutung für die microelektronische Industrie liegt in der physikalischen Struktur des Siliziums. In seiner reinsten Form ist es ein schlechter Elektrizitätsleiter. Wenn man es aber mit einer genau dosierten Menge bestimmter Fremdkörper verunreinigt, wird es zu einem Halbleiter.

Die elektrische Leitfähigkeit eines Materials wird durch Anzahl und Art der Bindung der Elektronen bestimmt, die die Atomkerne umgeben. In Metallen wird Strom durch fast ungebundene Elektronen transportiert. Diese können sich innerhalb des Metalls frei bewegen und übertragen ihre Bindung und elektrische Ladung von einem Atom zum nächsten. Ein

Isolator kann keinen Strom leiten, weil er keine freien Elektronen besitzt.

Die Herstellung reinen Siliziums ist ein relativ einfacher Prozeß. Zuerst wird das Oxyd chemisch bis zu 99,99prozentiger Reinheit veredelt. Dann wird es in einem Tiegel bis zum Schmelzpunkt (1410 °C) erhitzt. Dies geschieht in einer Edelgasatmosphäre, um störende Elemente abzuhalten. Für die Dotierung, die kontrollierte Verunreinigung des Siliziums, verwendet man Phosphor und Bor. Dabei bewirkt Phosphor eine n-Dotierung (negative Ladung). Bor ruft eine p-Dotierung hervor. Man erhält große Kristalle, indem man einen perfekten Einkristall in die Schmelze taucht und diesen langsam drehend wieder hochzieht. Auf diesem Weg erhält man Kristalle mit einem Durchmesser von sieben bis zehn und einer

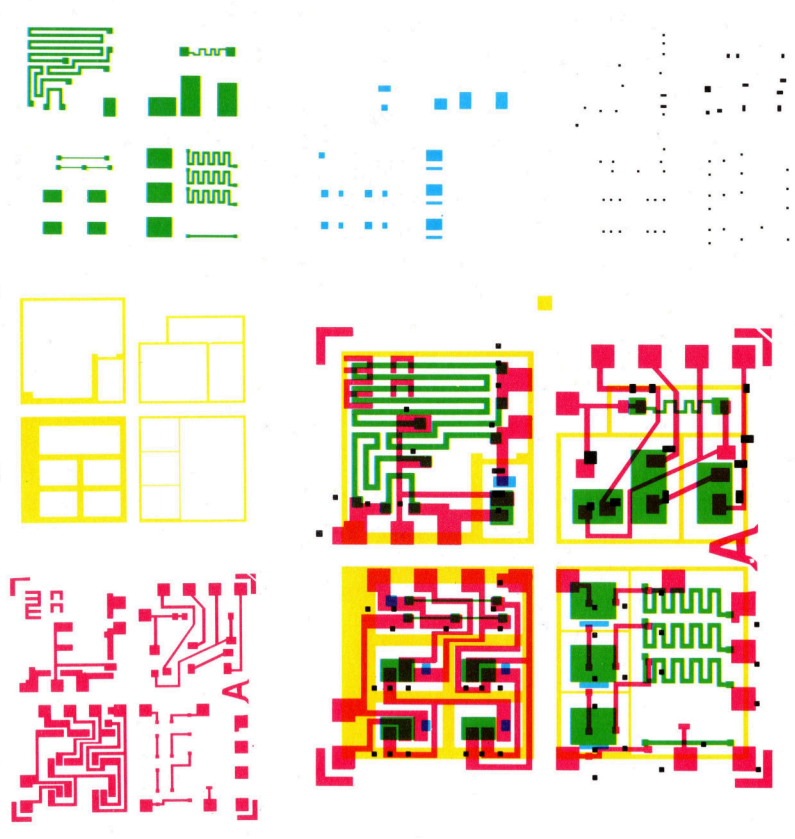


Das Foto zeigt einen Silizium-Brennofen. Nachdem das Silizium gereinigt, zersägt, belichtet und entwickelt wurde, erhält es eine Oberfläche aus Siliziumdioxid. Dazu werden die Wafer auf 1050 °C erhitzt, und reiner Sauerstoff wird darübergeleitet. Während sie in einem 'Boot' aus geschmolzenem Quarz im Ofen liegen, bildet sich eine Isolierschicht aus Siliziumdioxid SiO_2 . Diese Schicht wird dann beim nächsten Entwickeln teilweise entfernt, und dieser Zyklus wiederholt sich für jede weitere Schicht.



Die Summe der Teile

Jedes Foto wird in vier Farben und Schwarz aufgespalten. Diese werden dann einzeln übereinander gedruckt, um den hier sichtbaren Effekt zu erzielen. Die Chipherstellung benutzt Schichten aus dotiertem Silizium und anderen Materialien. Das Drucken ist wesentlich einfacher als das Beschichten/Belichten/Entwickeln der Chips. Wie man auf dem Bild sieht, werden die einzelnen Masken nacheinander kombiniert, um den Schaltkreis aufzubauen.



Länge von gut 60 Zentimetern. Diese werden dann auf einen Durchmesser von 76 oder 100 mm abgeschliffen. Der Kristall wird in Scheiben zerschnitten, beidseitig abgeschliffen, jedoch nur von einer Seite poliert. Der fertige Wafer, eine Siliziumscheibe, auf der integrierte Schaltkreise aufgetragen werden, ist dann nur noch einen halben Millimeter dick.

Wenn aber der Prozeß so einfach und das Rohmaterial so billig ist, warum ist das reine Silizium so teuer, bis zu 40 DM pro Scheibe?

Die Antwort liegt in der absoluten Reinheit des Materials. Man muß genau darauf achten, daß nicht ein einziges Fremdatom ins Silizium gelangt. Die Luftreinheitswerte in Siliziumfabriken sind beachtlich, sie betragen weniger als 3000 Partikel pro Kubikmeter. Damit ist die Luft dort mehr als 100mal sauberer als in modernsten Krankenhäusern.

Fotolithografie

Die Herstellung eines integrierten Schaltkreises, eines ICs, benötigt Microgravierungen auf der Chipoberfläche. In der Massenproduktion wird dies durch einen Prozeß erreicht, der als Fotolithografie bekannt ist.

Jede Schicht des Schaltkreises wird dabei als separate Einheit behandelt. Die Originalvorlagen werden mit Hilfe von Computern entwickelt, abfotografiert und zuletzt auf die geforderte Größe verkleinert. Die Maske

selbst wird produziert, indem man das Foto vervielfältigt und gitterartig zusammensetzt, um den gesamten Wafer zu bedecken.

Zuerst wird der Wafer in reinem Sauerstoff auf 1050 °C erhitzt, um eine Isolationsschicht aus Siliziumdioxid herzustellen. Diese Schicht wird später selektiv entfernt, und man erhält ‚Fenster‘ zum reinen Silizium darunter.

Die oxidierte Oberfläche des Wafers wird zuerst mit einem lichtempfindlichen Film beschichtet, dessen Löslichkeit sich unter UV-Licht extrem erhöht. Dann wird die Maske der ersten Schicht des Schaltkreises auf den Wafer gelegt, die freigebliebene Oberfläche wird UV-Licht ausgesetzt und in einer Flüssigkeit entwickelt, die den gesamten Film vollständig entfernt.

Eine ähnliche Methode wird bei Kupferstichen angewendet. Die Kupferoberfläche wird mit Wachs beschichtet, der Entwurf ausgekratzt und in eine schwache Säure getaucht. Wenn das Wachs entfernt ist, bleibt der Entwurf sichtbar. Unter der Wachsschicht fand keine chemische Reaktion statt.

Nachdem die erste Stufe beendet ist, wiederholt sich der Prozeß mit neuen Masken und Chemikalien, bis der Chip komplett ist. Zum Schluß wird der gesamte Wafer nochmals mit einer Isolation aus Siliziumdioxid beschichtet. Dieser Vorgang kann mehr als zehn Wiederholungen (Beschichten/Belichten/Entwickeln) erfordern. In jeder Stufe können viele Fehler



auftreten, und die Ausschußrate in der Prüfphase ist hoch. Je komplexer der Schaltkreis und je dichter der Wafer gepackt ist, desto höher ist der Ausschuß.

In der ersten Teststufe kommt der gesamte Wafer mit seinen Hunderten von Schaltkreisen in ein computerkontrolliertes Gerät, das alle Chips testet. Dazu sind mehrere Chips auf dem Wafer nur für Testzwecke reserviert. Die Prägeeinheit markiert nicht nur jeden Ausfall mit einem Tintenkleck, sondern liefert auch Informationen über Ausfallraten und Positionen der Fehler bei Chips und Wafer. Danach werden die Wafer in einzelne Chips zerlegt und die markierten aussortiert. Die verbleibenden werden auf Stempeln befestigt.

Der Chip wird mit sehr feinen Drähten an den Stempel angeschlossen, wiederum vom Computer kontrolliert. Das Ganze wird in ein Plastik- oder Keramikgehäuse eingegossen, aus dem die Anschlußbeine (Pins) herausragen, bevor die letzte Prüfung ansteht. Gegenwärtig liegt die Miniaturisierungsgrenze bei der Wellenlänge des UV-Lichts, also bei etwa einem 20millionsten Teil eines Meters. Neuerdings setzt man auf die Verwendung von Röntgenlicht, das voraussichtlich eine weitere Verdichtung um den Faktor 50 ergeben wird.

Alle Aspekte der Chipherstellung, vom Design über den Prototyp bis hin zur Massenproduktion und Endprüfung, sind so komplex, daß sie ohne Unterstützung derjenigen Einheiten, die gerade hergestellt werden, gar nicht realisierbar wären – ein echtes Paradoxon.

Produktionshilfe vom Computer

Die Verwendung von Computern, zum Beispiel in der Designphase, ermöglicht es, einzelne Sektionen einmalig zu definieren und sie wieder aus dem Speicher abzurufen, wenn sie benötigt werden. Nehmen wir beispielsweise einen Schreib/Lesespeicher (RAM). Jedes Bit benötigt eine Zelle mit je einem Transistor. Um also 2KByte zu speichern, muß der Chip 16 384 identische Zellen enthalten. Der Designer braucht also nur eine Zelle zu definieren und den Designcomputer anzuweisen, diese 16 384 mal zu multiplizieren. Die Nutzung von Computern beim Design ist nicht nur auf diese praktische Anwendung beschränkt. Die Konstrukteure arbeiten nicht an Zeichenplatten, sondern an Computerterminals, und zeichnen mit Lightpens direkt auf dem Bildschirm. Der fertige Entwurf wird dann vom Rechner auf einem Mehrfarbplotter ausgegeben. Nicht nur das Design der Masken wird mit Hilfe des Computers verbessert. Spezielle Programme, die auf ein bestimmtes Problem abgestimmt sind, werden auch für die Schaltkreisentwicklung eingesetzt. Dies ermöglicht dem Konstrukteur, verschiedene Lösungsmöglichkeiten auszuprobieren, bevor er sich für die geeignetste Variante entschließt. Dadurch lassen

sich kostenintensive Tests von unausgereiften Prototypen vermeiden.

Wenn wir von der microelektronischen Revolution sprechen, sind auch die enorme Kostensenkung und die Miniaturisierung der Computer gemeint. Deshalb sollten wir uns die Nachteile eines alten Röhrenrechners ins Gedächtnis rufen:

- Enormer Platzbedarf
- Geringe Arbeitsgeschwindigkeit
- Großer Energiebedarf
- Beschränkter Speicherplatz und Befehlssatz
- Hohe Kosten

All diese Bereiche waren unmittelbar von der Entdeckung des Transistors betroffen, wobei aber die Computerherstellung immer noch arbeitsintensiv und teuer blieb, weil die Bauteile auf gedruckten Leiterplatten zusammengefügt werden mußten.

Erst die Entwicklung des integrierten Schaltkreises, und vor allem des Mikroprozessors, ermöglichten der Industrie die volle Nutzung aller Vorteile computerkontrollierter Prozesse und Herstellungsverfahren.



Texas Instruments

Jack Kilby wird die Entdeckung des Integrierten Schaltkreises zuerkannt, den er 1958 bei Texas Instruments entwickelte. Er konstruierte ein Modul von 1,3 cm × 0,6 cm Größe, das mehrere Transistoren enthielt. Moderne Chips vereinigen Hunderttausende Komponenten auf dem gleichen Platz.

Die Statistiken sind teilweise atemberaubend. 1959 gab es nur Bauteile wie Transistoren und Dioden. Schon 1978 konnten LSI-Schaltungen (Large Scale Integrated circuit) bis zu einer viertel Million Teile auf einem einzigen Chip enthalten. Von 1973 bis 1983 sanken die Kosten je Bit-Speicherplatz um den Faktor 20, und die Nutzung elektronischer Geräte jeder Art stieg weltweit um den Faktor 1000. Diese Trends werden sich in der näheren Zukunft wohl fortsetzen. Es wird geschätzt, daß die Anzahl der genutzten Groß- und Kleincomputer während der nächsten drei Jahre um das Hundertfache ansteigen wird – besonders deshalb, weil die Geräte günstiger werden.



Chips im Haus

In vielen modernen Haushaltsgeräten befinden sich heute Microprozessoren. Von der Waschmaschine bis zur Heizungsanlage: Chips steuern, kontrollieren und helfen beim Sparen.

Wissen Sie, wie viele Microprozessoren sich in Ihrer Wohnung befinden? Bestimmt mehr als Sie vermuten. Denn außer dem Heimcomputer sind fast alle modernen Haushaltsgeräte, auch die Stereoanlage, die Waschmaschine, der Videorecorder oder auch die Autozündung und der Elektroherd mit Chips bestückt. Nicht zu vergessen der Taschenrechner und die Digitaluhr, den ersten Massenproduktionen von Microprozessoren.

Spielzeug

Kinder wissen Computer im Haus oft besser zu nutzen als ihre Eltern, weil sie damit genauso selbstverständlich umgehen wie mit einem Fernsehgerät. Die Computersprache LOGO fördert beispielsweise selbständiges Lernen und die Kreativität der Kinder. Außerdem sind gute Bildungsprogramme erhältlich. Spiele sind zwar unterhaltend, aber die Fähigkeiten der Kinder entwickeln sich am besten beim Programmieren. LOGO ist inzwischen überall für die meisten Heimcomputer erhältlich und bietet enorme Möglichkeiten für Problemlösungen.

Steve Cross

Der Reinigungschip

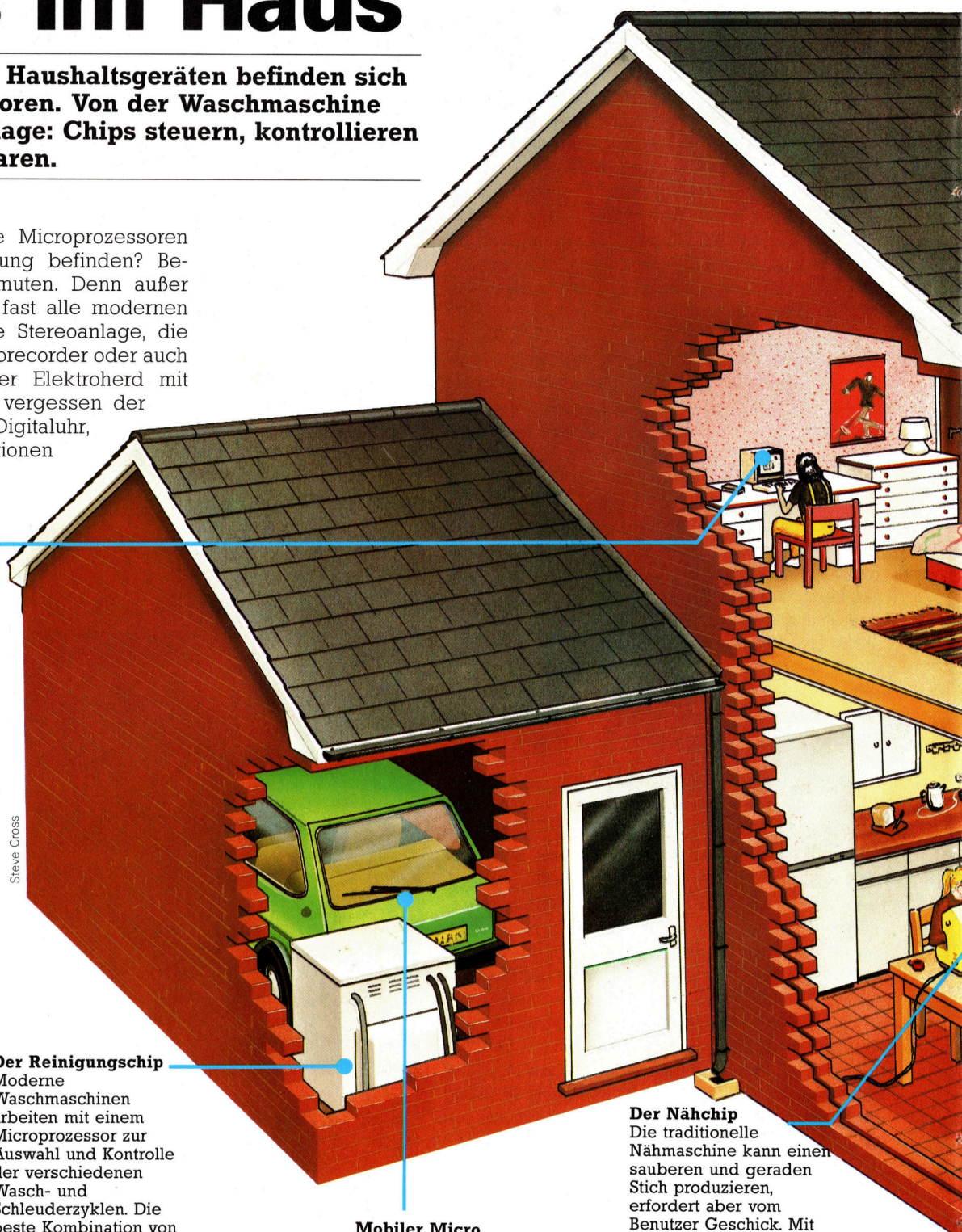
Moderne Waschmaschinen arbeiten mit einem Microprozessor zur Auswahl und Kontrolle der verschiedenen Wasch- und Schleuderzyklen. Die beste Kombination von Drehzahl, Temperatur und Wassermenge kann durch Tasten-Berührung angezeigt und gewählt werden. Weil keine elektromechanischen Teile enthalten sind, sinkt auch die Reparaturanfälligkeit.

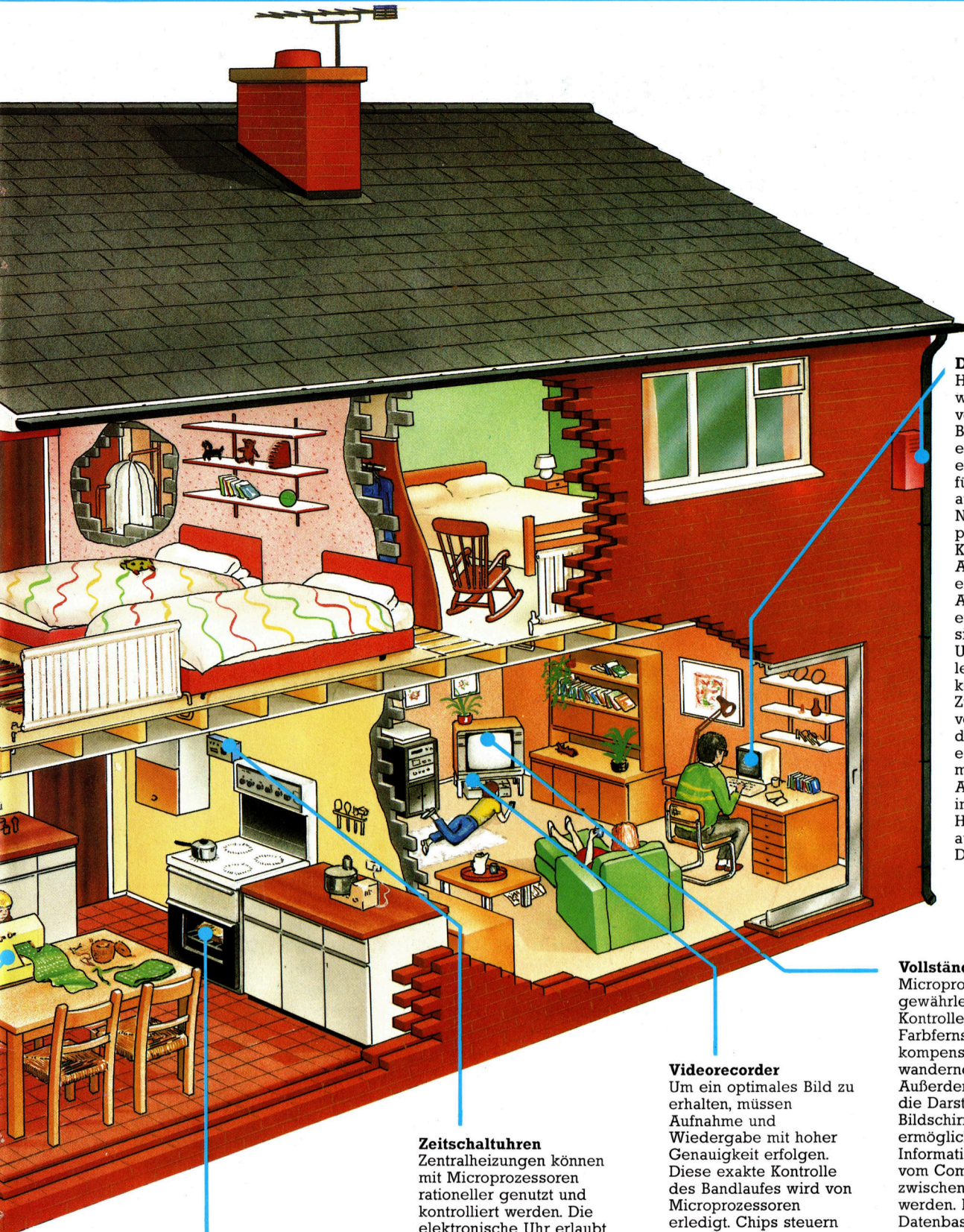
Mobiler Micro

Chips im Auto überwachen die Geschwindigkeit, die Drehzahl, den Benzinverbrauch; sie arbeiten als Alarmanlage oder weisen den Fahrer auf zu niedrigen Ölstand und abgenutzte Bremsbeläge hin.

Der Nähchip

Die traditionelle Nähmaschine kann einen sauberen und geraden Stich produzieren, erfordert aber vom Benutzer Geschick. Mit einer Nähmaschine, die von einem Micro gesteuert wird, lassen sich komplizierte Sticharten realisieren.





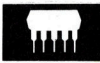
Computerherd
Ein Computerherd erleichtert das Kochen, indem er Zeit und Temperatur perfekt kontrolliert.

Zeitschaltuhren
Zentralheizungen können mit Microprozessoren rationeller genutzt und kontrolliert werden. Die elektronische Uhr erlaubt die separate Programmierung unterschiedlicher Tages- und Nachttemperaturen für jeden Raum im Haus. Diese Kontrollmöglichkeiten helfen beim Energiesparen.

Videorecorder
Um ein optimales Bild zu erhalten, müssen Aufnahme und Wiedergabe mit hoher Genauigkeit erfolgen. Diese exakte Kontrolle des Bandlaufes wird von Microprozessoren erledigt. Chips steuern auch die Aufzeichnung von TV-Sendungen, wenn man außer Haus ist. Dazu müssen nur Uhrzeit und Programm eingegeben werden.

Vollständige Kontrolle
Microprozessoren gewährleisten eine gute Kontrolle des Farbfernsehers. Sie kompensieren ein wanderndes Bild. Außerdem können sie die Darstellung von Bildschirmtext ermöglichen. Die Informationen können vom Computer zwischengespeichert werden. Mit dieser Datenbasis sind die aktuellsten Informationen dann jederzeit abrufbar.

Der Heimcomputer
Heimcomputer werden in den verschiedensten Bereichen eingesetzt. Sie eignen sich sowohl für die berufliche als auch für die private Nutzung. Die praktischen Kleingeräte können Abrechnungen erstellen, Aktenordner ersetzen und eignen sich vorzüglich zur Unterhaltung. Kinder lernen spielend komplexe Zusammenhänge zu verstehen und darüber hinaus gibt es noch mannigfaltige Anwendungsbereiche, in denen ein Heimcomputer ausgezeichnete Dienste leisten kann.



Auf die Perspektive kommt es an

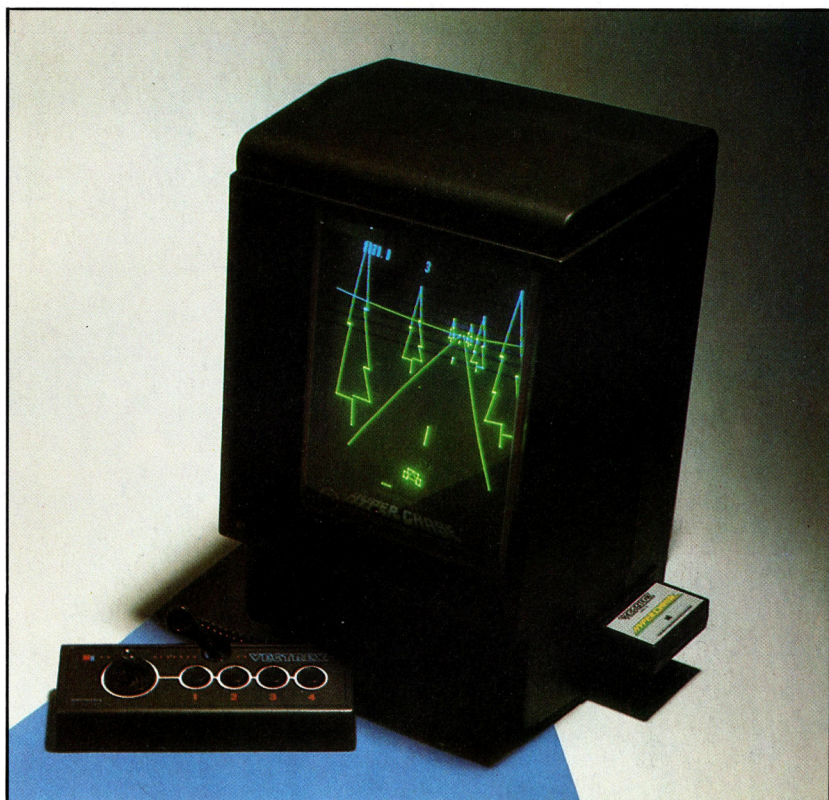
Warum Video-Spielautomaten bessere Bilder zeichnen.

Bestimmt haben Sie sich schon einmal gefragt, warum die Bildwiedergabe bei Video-Spielautomaten von erheblich höherer Qualität ist als bei den meisten Heimcomputern. Auch mit viel Erfahrung ist es nicht leicht, fließende Zoom-Effekte und dreidimensionale Darstellungen zu erreichen. Wäre es einfach, dann würden Sie mehr Programme finden, die diese Techniken nutzen. Es gibt jedoch nur wenige – und die sind nicht gerade beeindruckend.

Die vergleichsweise hohe Bildqualität bei Spielautomaten beruht auf Unterschieden in der Elektronik und in den Programmen sowie auf dem speziellen Aufbau der Bildröhre. Wer das Schirmbild eines Video-Spielautomaten genau betrachtet, stellt folgendes fest: Die abgebildeten Figuren sehen aus, als seien sie aus Draht gebogen. Sie weisen weder Flächen noch Schattierungen auf, und die Striche haben nur einen Helligkeitsgrad. Ihre Farbe ist meistens grün oder weiß – je nach Bildschirmbeschichtung. Hierin unterscheidet sich das Spielautomaten-Bild vom Fernsehbild: Ersteres wird nicht wie beim TV-Gerät Punkt für Punkt und in Zeilen von oben nach unten aufgebaut, der Elektronenstrahl zeichnet vielmehr die Figur direkt auf den Bildschirm des Spielautomaten. Soll zum Beispiel ein Quadrat gezeigt werden, schreibt der Strahl nur die vier Begrenzungslinien. Um das Quadrat zu vergrößern oder zu verkleinern, wird einfach die Ablenkspannung entsprechend erhöht oder verringert. Die auf dem Hals der Bildröhre sitzenden Ablenkspulen dirigieren (ähnlich wie Magnete) den Elektronenstrahl. Um mehrere gleiche Figuren auf den Bildschirm zu zeichnen, wird der Strahl nach Vollendung der ersten Figur ausgeschaltet, zur neuen Startposition abgelenkt und wieder eingeschaltet.

Drahtgittermodelle

Will man eine Zeichnung auf einem Stück Papier anfertigen, stehen dem Maler nur zwei Dimensionen zur Verfügung. Und genauso ist es beim Bildschirm. Um beeindruckende dreidimensionale Effekte zu erzielen, bedient man sich derselben Methoden, die ein Maler für sein Blatt Papier verwendet. Nur ist die Übertragung dieser Methoden auf ein Computerprogramm ein komplexes Problem, das



lan McKinnell

schnelle trigonometrische Rechengänge erfordert. Mit solchen Programmen kann das dargestellte Objekt durch Änderung des „Blickwinkels“ gedreht werden. Es kann sich dem Betrachter nähern, sich von ihm entfernen oder viele andere Bewegungen mit ausreichender Natürlichkeit ausführen. Ein noch nicht gelöstes Problem stellen bei 3-D-Figuren die verdeckten Linien dar. Bisher ist hierfür kein praktikabler Algorithmus gefunden worden. Alle Objekte erscheinen deshalb durchsichtig, so als seien sie aus feinem Draht gebogen. Daher der Name Drahtgittermodell oder Gittergrafik. Außerdem wird diese Art der Bildwiedergabe auch als Vektorgrafik bezeichnet.

Wie können Heimcomputer-Besitzer nun in den Genuß der Vektorgrafik kommen? Bis vor kurzem war dies gar nicht möglich. Heute gibt es jedoch den Vectrex, einen speziellen Computer für Video-Spiele. Dieses Gerät ist mit einer besonderen Eingabekontrolle für Joystick und einem Vektorgrafik-Bildschirm ausgerüstet, außerdem können handelsübliche Cassetten verwendet werden.

Der Vectrex wurde speziell für Video-Spiele entwickelt. Er ist der einzige Computer auf dem Weltmarkt, der einen Bildschirm für Vektorgrafik hat (im Gegensatz zu den Rastergrafik-Schirmen normaler Fernsehgeräte oder Monitore). Die Abbildung zeigt die Vektorgrafik für das Rennfahrerspiel „Hyperchase“. Ungewöhnlich ist auch das festinstallierte Spiel „Minestorm“, das beim Einschalten automatisch abläuft, bis eine Cassette eingeschoben wird.



BBC Modell B

Seine guten technischen Eigenschaften und das gezielte Marketing sowie das Know-how der BBC haben diesen Rechner weit über England hinaus populär gemacht.

Der BBC Modell B und sein kleinerer Bruder, der Acorn Electron, werden von der Firma Acorn Computers im britischen Cambridge hergestellt. Der Electron wurde im Sommer 1984 auf dem deutschen Markt eingeführt.

Das Modell B hat sich sehr gut an britische Schulen verkauft und steht auf der Liste der von der englischen Regierung genehmigten Computer. Für diesen Rechner wurden zahlreiche Schulprogramme entwickelt, die von der Einführung in die Programmiersprachen bis hin zu Lehrprogrammen reichen.

Die technischen Eigenschaften des Modell B werden trotz der großen Anzahl neuerer Rechner nach wie vor als hervorragend angesehen. Insbesondere die Programmiersprache BBC BASIC ist für die Steuerung von Spezialfunktionen ausgezeichnet geeignet. Sie erleichtert zudem die Entwicklung und Korrektur von Programmen.

Acht unterschiedliche Grafik-Arten stehen dem Anwender zur Verfügung. Er hat die Wahl zwischen niedriger, mittlerer und hoher Auflösung, jedoch ist bei der letzteren Möglichkeit

die Anzahl der verfügbaren Farben begrenzt. Die höchste Auflösung beträgt 1280×1224 Punkte. Viele Anwender wählen zwar als Bildschirm ihren Fernseher, jedoch empfiehlt es sich, einen guten Monitor anzuschließen. Nur dann erzielt das Grafiksystem des BBC Modell B die besten Ergebnisse.

Da die Stromversorgung innerhalb des Gehäuses untergebracht ist, wirkt der Computer ausgesprochen handlich. An der Rückwand und auf dem Boden befinden sich mehr Schnittstellen als in den meisten anderen Rechnern. Das Standardgerät kann deshalb mit einer Vielzahl von Peripheriegeräten ausgerüstet werden.

Zusätzlich zu den Schnittstellen gibt es einen Netzwerkanschluß. Er ist ideal für die Verwendung in einem Klassenzimmer, wenn mehrere Anwender sich einen Drucker oder eine Diskettenstation teilen müssen.

Und schließlich ist da noch „The Tube“, eine hochentwickelte Schnittstelle, an die ein anderer Microprozessor zur Zusammenarbeit angeschlossen werden kann.



Die Tastatur des BBC Modell B ist gut ausgelegt und stabil gebaut. Seine Tasten sind konturiert. Der Cursor wird beim Bearbeiten von Texten oder Programmen per Pfeiltasten auf dem Bildschirm bewegt. In der obersten Tastenreihe liegen 10 rote, frei programmierbare Funktionstasten. Sie sind besonders nützlich für Lehrprogramme. Bequem sind auch die drei eingebauten LEDs (Light Emitting Diodes), die anzeigen, ob der Motor der Diskettenstation eingeschaltet ist und ob die SHIFT LOCK- oder die CAPS LOCK-Tasten gedrückt sind.



Diskettenstation

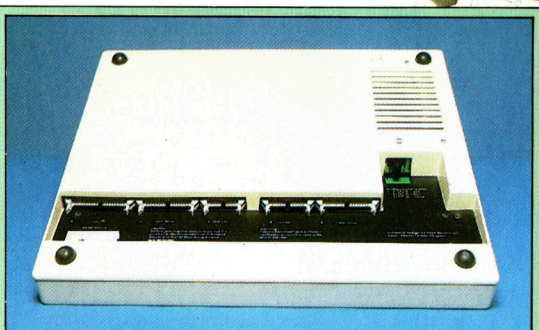
Diese Station für den BBC Micro ist zwar sehr attraktiv, kostet aber mehr als 1000 Mark und kann nur 100 KByte pro Diskette speichern. Es gibt jedoch günstigere Stationen von anderen Herstellern.

Internationale Chips

Wer sich die Chips genau ansieht, wird feststellen, daß es eine breite Palette internationaler Hersteller gibt. Der BBC Modell B enthält Chips aus Malaysia, Japan, Portugal, Schottland und den USA.

Chips für Schnittstellen

Flexible Adapter für Schnittstellen, wie dieser Chip MOS Technology 6522, steuern die Schnittstellen für externe Geräte.



Peripherieanschlüsse

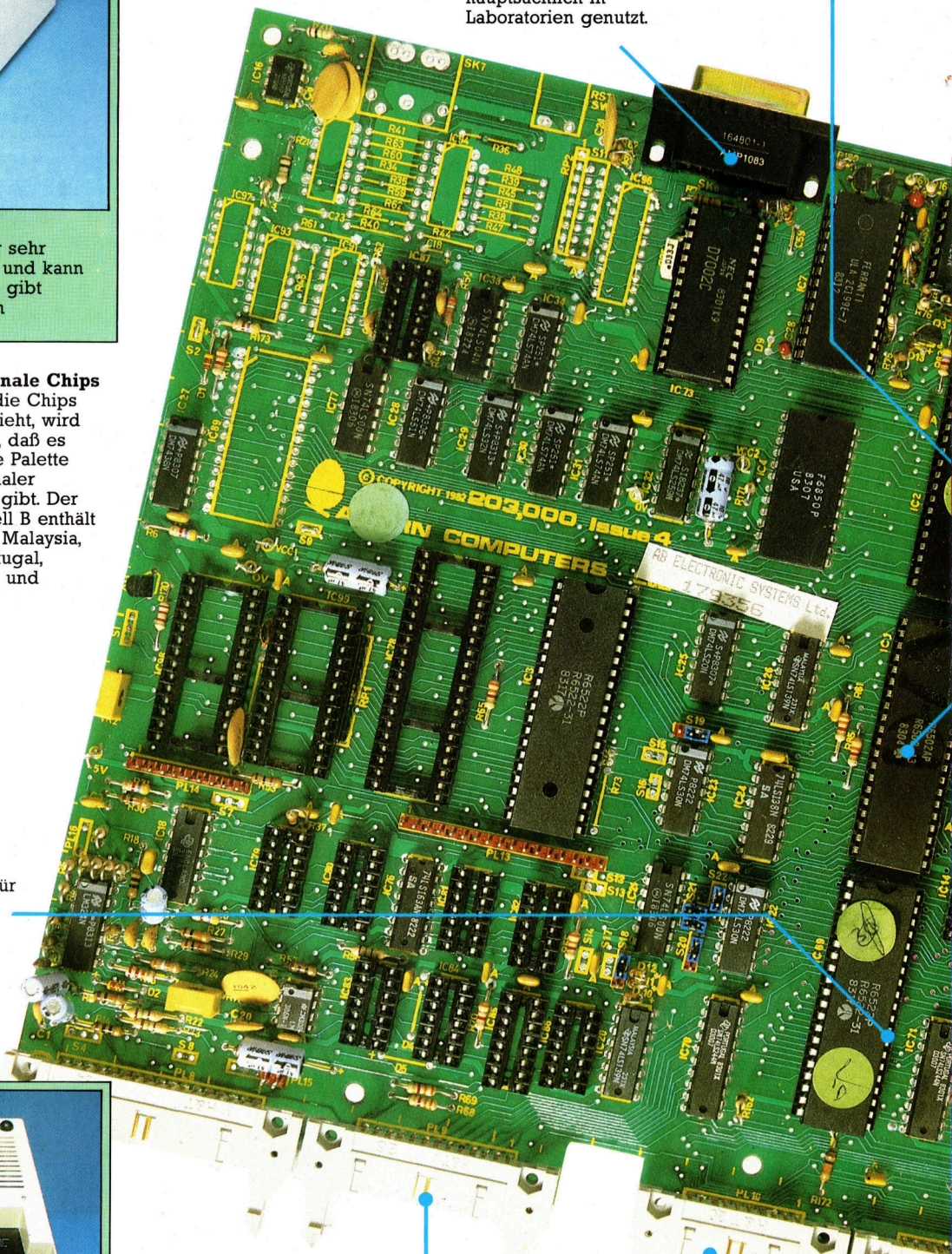
Auf der Unterseite des BBC Modell B befinden sich zahlreiche Schnittstellen.

Eingang für analoge Signale

Hierüber kann der Computer die Spannung eines nicht digitalen Gerätes, wie z. B. eines Temperaturfühlers, empfangen. Er wird hauptsächlich in Laboratorien genutzt.

Video Controller

Dieser Chip wandelt die Informationen aus dem RAM Bereich in ein Videosignal für die Anzeige auf dem Bildschirm um.



Schnittstelle für Drucker

Hier kann ein Drucker angeschlossen werden, der über ein paralleles Signal gesteuert wird.

Benutzer-Ein- und Ausgang

Für Experimente mit Digitalgeräten und selbstgebauten Logikelementen.



Cassettenanschluß
Programme können mit erhöhter oder normaler Bandgeschwindigkeit auf eine Cassette gespeichert werden.

Ausgang für Farbmonitore
Gibt die Signale für die Rot-, Grün- und Blaukomponenten an einen Farbmonitor weiter.

Anschlußbuchse für Fernseher
Sie wird an das Antennenkabel eines Fernsehers angeschlossen.

RS 232 Schnittstelle
Ein Hochgeschwindigkeitsinterface für die serielle Verbindung mit Peripheriegeräten.

Videoausgang
Ausgang für einen Einfarbenmonitor.

Modulator
Der Modulator übernimmt das Farbsignal von dem Video Controller und wandelt es in Impulse für ein TV-Gerät um.

Quarzkristall
Ein pulsierender Quarzkristall ist das Herz der internen Uhr, die alle Vorgänge synchronisiert.

Microprozessor
Hauptprozessor ist der 6502 MOS Technology.

Freier Speicherplatz
Der BBC Modell B enthält 32 K RAM für die Speicherung von Programmen, Daten und Grafiken.

ULA
Dieses spezielle „Uncommitted Logic Array“ (Flexible Logikschaltung) erledigt Arbeiten, für die andere Computer viele Chips benötigen. Das Metallschild schützt den Chip vor Überhitzung.

ROM
Diese beiden ROM-Chips liefern die Programmiersprache BASIC und das Betriebssystem, das die internen Funktionen des Computers kontrolliert.

„The Tube“
Über dieses Spezial-Interface kann der BBC Modell B mit anderen Microprozessoren arbeiten.

BBC Modell B

PREIS

ca. 2000 DM

ABMESSUNGEN

409 × 358 × 78 mm

GEWICHT

3700 g

TAKTFREQUENZ

2 MHz

MASCHINENSPEICHER

32 KByte RAM
32 KByte ROM einschließlich BASIC und einem ausgereiften Betriebssystem.

BILDSCHIRM-DARSTELLUNG

8 Arten von Grafik liefern eine Auswahl an Bildschirmdarstellungen. 32 Zeilen mit je 80 Zeichen. Maximale hochauflösende Grafik: 1280 × 1224 Pixel. Bis zu 16 Farben in einer niedrigeren Auflösungsstufe.

SCHNITTSTELLEN

Fernseheranschluß, Farb- und Monochrommonitor, Diskettenstation, Drucker, Joysticks, Analogeingang, Benutzer Ein- und Ausgang, „The Tube“ (für den Anschluß anderer Microprozessoren)

PROGRAMMIERSPRACHE

BASIC

WEITERE SPRACHEN

LISP, FORTH, LOGO

ZUBEHÖR

Anschlußkabel für Cassettedeck und Fernseher. Handbuch. „Welcome“ Demonstrationscassette mit Broschüre.

TASTATUR

Schreibmaschinentastatur mit 74 Tasten und 10 frei programmierbaren Tasten

DOKUMENTATION

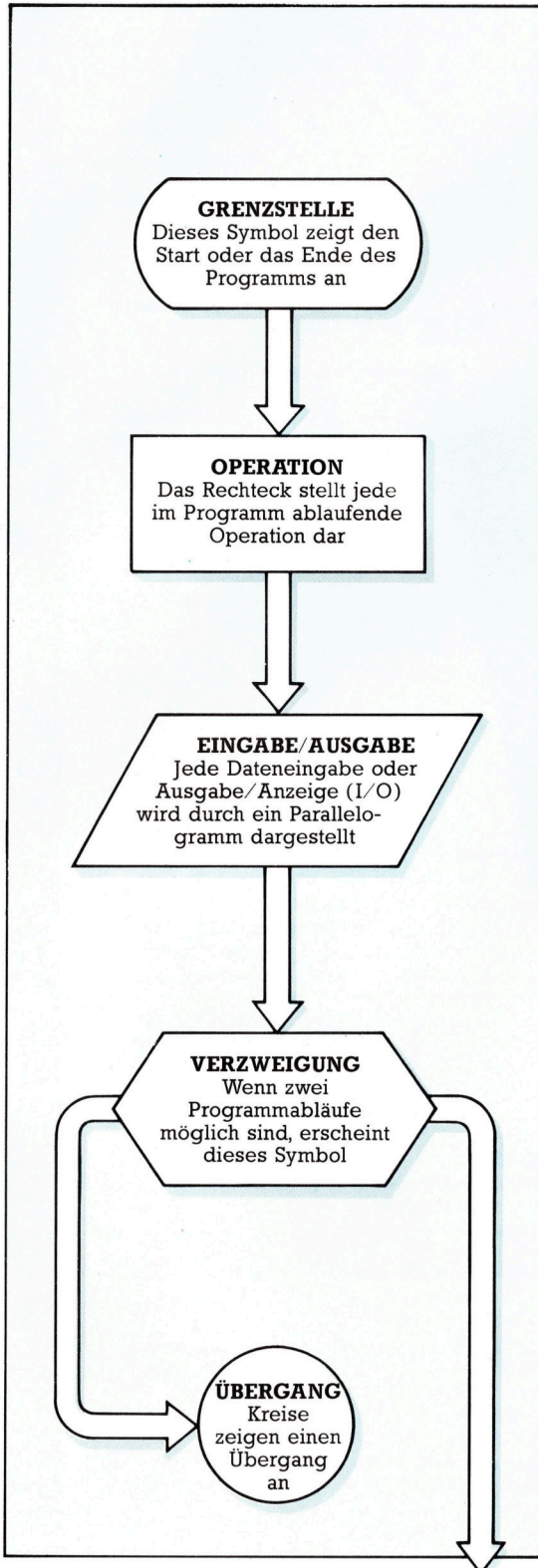
Das Benutzerhandbuch des BBC Modell B geht davon aus, daß die Anwender bereits mit Computern umgehen können. Die speziellen Systemprogramme für Grafik, Tonsystem und In-/Output werden genau erklärt. Auch eine detaillierte Erklärung der Funktionsweise des 6502 Microprozessors ist vorhanden – nicht einfach nur die Kopie der Originaldokumente von Rockwell.



Der Weg zur Lösung

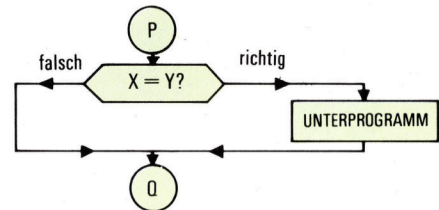
Mit Flußdiagrammen lassen sich Programme auf übersichtliche Weise strukturieren. Probleme werden grafisch dargestellt und in separate Teilbereiche zerlegt.

Der Sinn des Flußdiagramms ist, die Steuerung des Programmablaufs durch den Computer übersichtlich und umfassend darzustellen. Sehr wichtig sind dabei die Verzweigungen, die den Programmablauf auf eine bestimmte Anweisung hin umleiten. Eine grafische Darstellung dieses Vorganges ist viel leichter zu begreifen als eine Programmier-Anweisung. Daß ein Bild große Ausdruckskraft besitzt, trifft hier in besonderem Maße zu. Das Symbol für Verzweigungen ist ein Karo oder ein Sechseck. Wichtig sind die pfeilartigen Seiten.



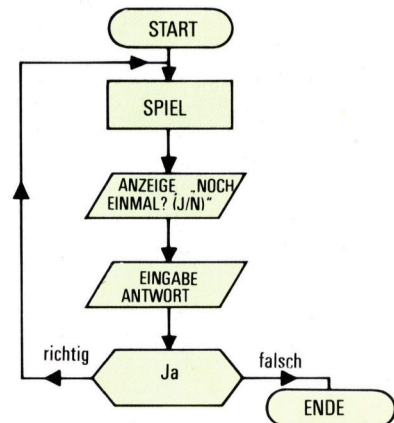
Kevin Jones

Je schwieriger ein Problem, desto nützlicher sind Flußdiagramme bei der Zerlegung des Gesamtproblems in einzelne Teilbereiche. Ein Rechteck stellt eine Operation oder einen Prozeß dar und der Pfeil der Flußlinie gibt eine mögliche Ablaufrichtung durch den Prozeß an. Flußlinien sind immer „Einbahnstraßen“, die gewöhnlich von oben nach unten und von links nach rechts verlaufen. An Punkten, wo eine Wahlmöglichkeit besteht und sich der Lösungsweg verzweigt, steht ein Karo oder ein Sechseck. An solch einer Verzweigung kann der Programmablauf, wenn bestimmte Bedingungen erfüllt sind, nach links oder rechts abfließen. Soll das Programm über die Durchführung nur einer Operation entscheiden, darf das entsprechende Rechteck nur im linken oder rechten Ausgang der Verzweigung liegen. Im folgenden Beispiel soll das Programm entscheiden, ob der Ablauf auf ein Unterprogramm abzweigen soll:



```
120 IF X = Y THEN GOSUB 300
```

Im nächsten Beispiel wird das Programm nur dann erneut durchlaufen, wenn der Spieler die Frage des Computers „Noch einmal?“ mit „J“ beantwortet:

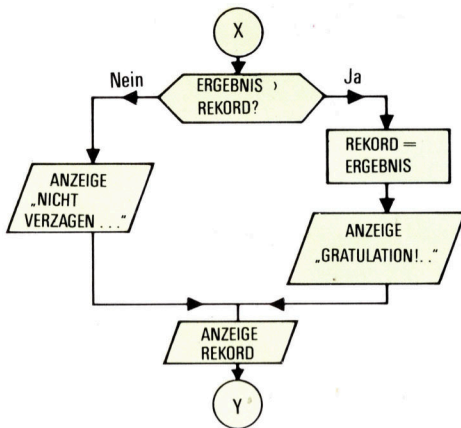


```
90 REM**SPIELBEGINN**
100
800
```




```
810 PRINT "NOCH EINMAL ? (J/N)";
820 INPUT R$
830 IF R$ = "J" THEN GOTO 100
840 END
```

Das nächste Beispiel zeigt eine Verzweigung, deren Ausgänge zu zwei völlig unterschiedlichen Operationen führen. Das Ergebnis eines Spielers wird mit der höchsten bisher erreichten Punktzahl, dem Rekord, verglichen:



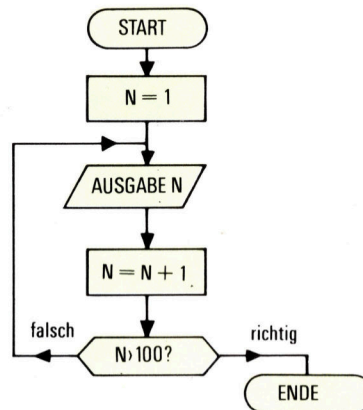
```
1200 IF ERGEBNIS > REKORD THEN GOTO 1230
1210 PRINT "NICHT VERZAGEN";
1220 GOTO 1250
1230 LET REKORD = ERGEBNIS
1240 PRINT "GRATULATION! EIN NEUER";
1250 PRINT "REKORD."; REKORD
```

Jeder der beiden möglichen Wege hinter der Verzweigung endet mit der Anzeige der Rekord-Punktzahl.

Zum Bestand jeder Programmiersprache gehört eine Anweisung, die Verzweigungen auf andere Programmteile durch einen bedingten Sprung ermöglicht. Der Sprung kann also nur erfolgen, wenn die Bedingung für die Verzweigung erfüllt ist. Andernfalls wird der Programmablauf mit der nächsten Anweisung fortgesetzt. Gehört zum Anweisungsvorrat Ihrer BASIC-Variante nur die IF-THEN-Kombination, so müssen Sie, wie in Zeile 1200 gezeigt, eine weitere GOTO-Anweisung benutzen, um die bedingte Verzweigung komplett zu machen. Die Anweisung in Zeile 1210 führt das Programm also nur aus, wenn das Ergebnis des Tests in Zeile 1200 falsch ist. Warum nun aber noch ein GOTO in Zeile 1220? Da mit dem ersten GOTO die Verzweigung eingeleitet wurde, müssen mit dem zweiten in Zeile 1220 die beiden möglichen Wege wieder zusammengeführt werden. Treffpunkt ist in diesem Fall Zeile 1250.

Die Anwendung des Flußdiagramms zur grafischen Darstellung des Programmablaufs ermutigt zwar zum häufigen Einsatz von GOTO. Die allzu lockere Handhabung dieses an keine Bedingungen geknüpften Sprungs ist jedoch nicht ohne Gefahren. Oft läßt einem aber die

verwendete BASIC-Variante keine andere Wahl. So bleibt das Flußdiagramm ein vortreffliches Mittel, die Übersicht über Verzweigungen und ihre korrekte Rückführung zum normalen Programmablauf zu behalten. An einem letzten Beispiel wollen wir zeigen, wie ein Flußdiagramm die einzelnen Schritte eines kleinen Programms, das alle Zahlen von 1 bis 100 ausdrückt, übersichtlich und korrekt wiedergibt.



```
10 LET N = 1
20 PRINT N
30 LET N = N + 1
40 IF N > 100 THEN END
50 GOTO 20
```

So überzeugend diese Beispiele auch sind, der Gebrauch von Flußdiagrammen kann durchaus auch zu unbefriedigenden Lösungen führen. Dies gilt insbesondere für umfangreiche Aufgaben und Projekte; denn die sich zwangsläufig aufdrängende „Schritt für Schritt“-Methode ist nicht immer vorteilhaft. In unserem letzten Beispiel ist unschwer zu erkennen, daß die mit FOR und NEXT aufgebaute Schleife eine wesentlich elegantere Lösung darstellt:

```
10 FOR N = 1 TO 100
20 PRINT N
30 NEXT N
40 END
```

Zweifellos sind Flußdiagramme während der Planungs- und Konzeptphase, besonders bei den komplizierten Teilen eines Programms, von sehr großem Nutzen für den Anwender. Erfahrene Programmierer, die von Berufs wegen mit Problemlösungen konfrontiert werden, greifen erfahrungsgemäß weniger oft auf dieses Hilfsmittel zurück. Meist verwenden sie die Flußdiagramme zur Illustration und Dokumentation eines Programms. Dabei erscheint es aber unerheblich, ob nun der Programmierer das Flußdiagramm auf ein Blatt Papier zeichnet oder ob er aufgrund seiner Routine in der Lage ist, die Vorgänge in Gedanken durchzuspielen.



Daten auf Abruf

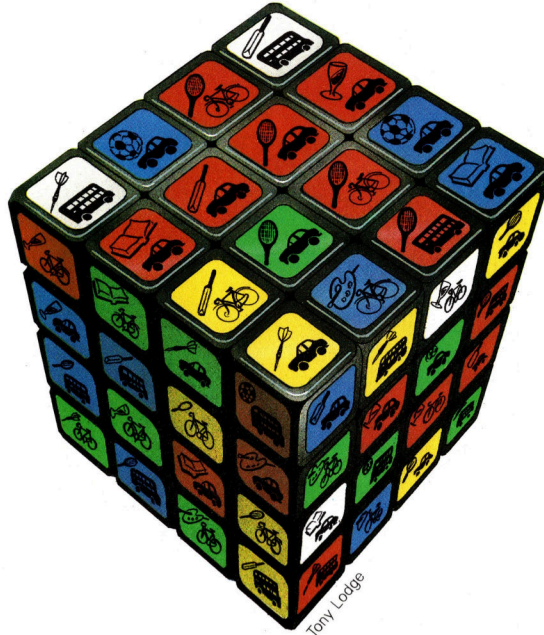
Mit Datenbank-Programmen lassen sich aus einer Masse ungeordneter Informationen übersichtliche Dateien zusammenstellen.

Eine Datenbank ist eine Ansammlung von Informationen, mit der ein Computer arbeiten kann. Auch im täglichen Leben greifen wir immer wieder auf Datenbanken zurück, wenn auch ohne Zuhilfenahme eines Computers. Bestes Beispiel: das Telefonbuch.

Datenbank-Programme beinhalten Regeln und Routinen, die eine vielseitige Ordnung der Daten erlauben. Es gibt Programme, die lediglich bessere Karteikästen sind und andere, die den Charakter einer eigenständigen Programmiersprache haben. Typische Computer-Datenbanken haben umfangreiche und sehr unterschiedliche Datenbestände angesammelt. Dies bedeutet aber nicht, daß man für die Handhabung unbedingt einen großen Computer besitzen muß. Fast jeder Computer kann mit einer Datenbank arbeiten. Grenzen setzen Speicherumfang und Zugriffszeit der verwendbaren Datenspeicher.

Wir könnten beispielsweise eine Personaliste anlegen, die Daten über verschiedene Personen enthält. Auf gewöhnliche Karteikarten übertragen entstünde auf diese Weise eine Personalkartei. Wir erkennen leicht verschiedenartige Gruppen von Informationen. Einige enthalten nur Wörter, andere nur Zahlen. Die Vielfalt der Informationen ist in einigen Gruppen begrenzt. So kann die Eintragung für Geschlecht entweder nur männlich oder weiblich, die für Familienstand nur ledig, verheiratet oder verwitwet sein.

Man kann auch eine andere Liste mit Wörtern und Zahlen mit dem Titel „Beschäftigung“

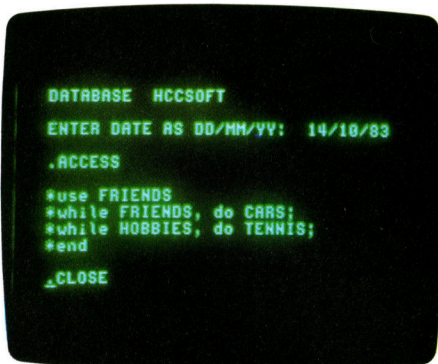


- VERFÜGBAR
- TENNISPIELER
- KRICKETSPIELER
- LESERATTE
- FUSSBALLER
- KÜNSTLER
- DART-SPIELER
- WEINFREUND
- AUTOBESITZER
- ÖFFENTLICHE VERKEHRSMITTEL
- RADFAHRER

anlegen, die Beruf, Firmenname, Geschäftsadresse, Telefonnummer in der Firma und Name des Vorgesetzten enthält. Eine weitere Liste „Transportmittel“ könnte zusätzlich Marke und Alter der Privatautos enthalten. Außerdem wäre es möglich, alle Wohnorte in einer 4. Liste zusammenzufassen.

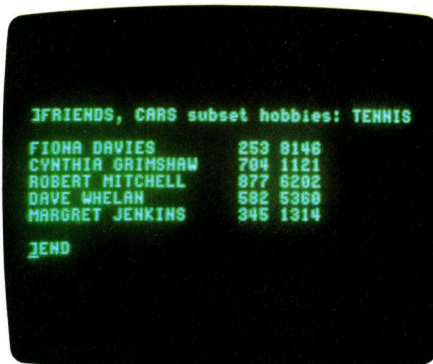
Den Familienstand könnte man um den Namen des Ehepartners erweitern, falls dies sinnvoll erscheint. Dafür wäre nur ein Wort nötig. Da es sich aber auf eine Person bezieht, die einen eigenen Datensatz haben kann,

Dieser Rubik-Würfel dient als anschauliches Beispiel für die Grundform einer Datenbank. Sie enthält zwar alle Daten, ist aber noch ungeordnet. Alle Tennisspieler sollen aussortiert werden (Tennisschläger-Symbol), ebenso die Autobesitzer (Auto-Symbol), die an einem bestimmten Tag (rotes Quadrat) Zeit haben.

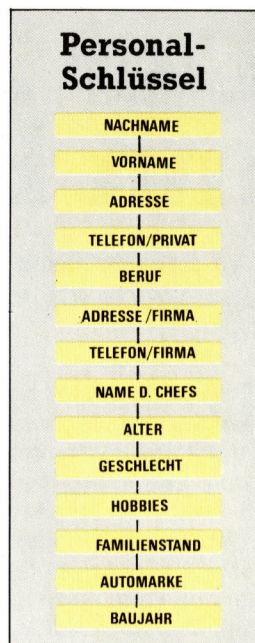
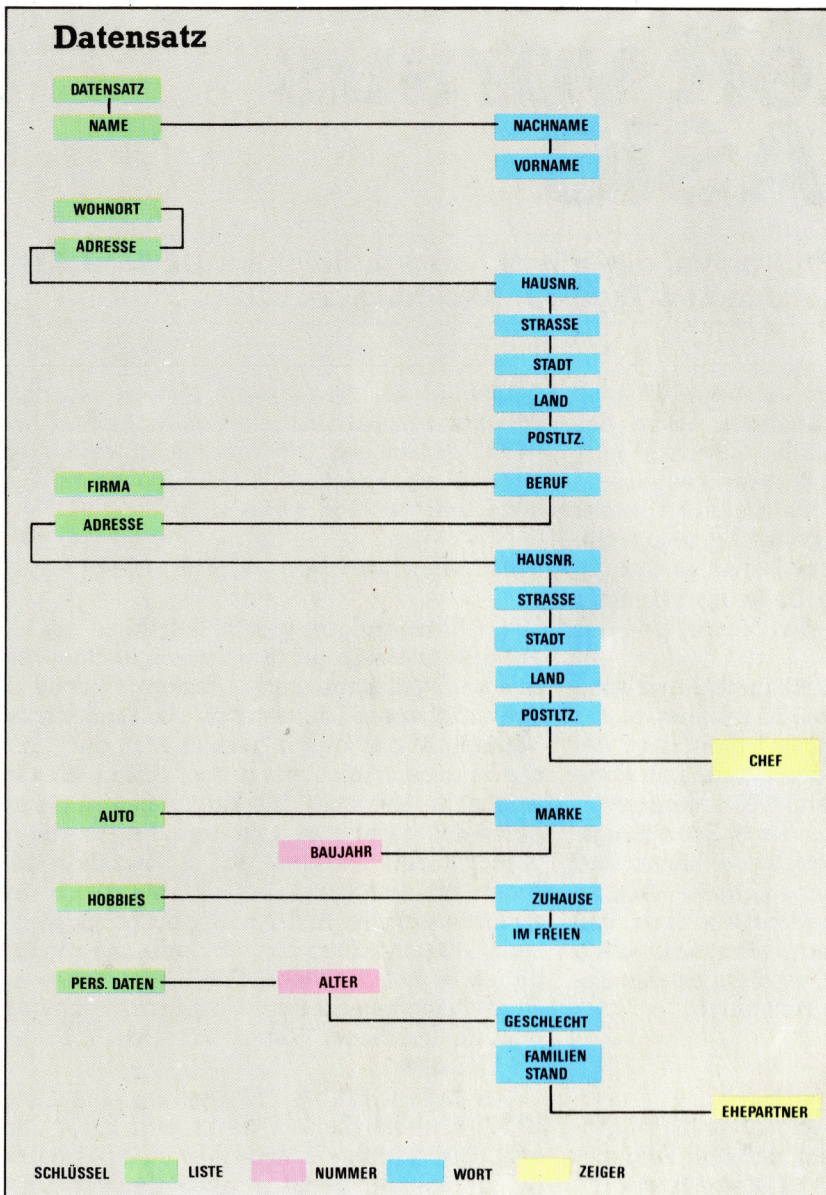


Die richtigen Fragen

Einige typische Datenbank-Befehle zeigen die Abbildungen. „ACCESS“ sagt dem Computer, daß Fragen über eine vorher aufgebaute Datei gestellt werden. „USE“ zeigt an, daß die Datei „Freunde“



benutzt werden soll. Alle Datensätze, die eine Eintragung im Feld „Auto“ und in der Unterdatei „Hobbies“ die Eintragung „Tennis“ haben, sollen aussortiert werden. Auf diese Weise erhält man Namen und Telefonnummern der Freunde, die ein Auto besitzen und Tennis spielen.



Hierarchisch strukturierte Datenbanken führen den Benutzer durch die Masse von Informationen und bieten Wahlmöglichkeiten an allen Abzweigungen. Eine besondere Kenntnis des Datenbank-Inhalts ist nicht erforderlich.

wäre ein einfacher Querverweis sinnvoll. Dies kann mit der Nummer geschehen, die jeder Datensatz erhält. In diesem Fall wird nicht der Name des Ehepartners, sondern die Nummer des Datensatzes angegeben. In der Datenbank-Fachsprache wird dies „Zeiger-Datensatz“ genannt.

Im Vergleich zu einer Computer-Datenbank hat eine normale Kartei gewöhnlich nur eine alphabetische Ordnung. Um den Arbeitgeber einer Person festzustellen, mag dies ausreichen. Was aber, wenn die Namen aller Arbeitnehmer einer bestimmten Firma aufgelistet werden sollen? Mit einer normalen Kartei wäre dies ein zeitraubendes Verfahren, denn man müsste jede Karte einzeln prüfen und die gesuchten Daten aufschreiben. Bei Verwendung einer Datenbank dagegen wird der Computer beauftragt, sich die Datensätze nacheinander anzusehen und die gewünschten Daten geordnet auszudrucken. Eine normale Kartei kann für die Kartenordnung (Datensatz-Ordnung) nur ein Schlüsselfeld haben, beispielsweise den Nachnamen einer Person. Bei einer Datenbank dagegen kann jedes Feld (Spalte für Eintragung) des Datensatzes das Schlüsselfeld sein. Zwar ändert sich bei einer Neuordnung der Datensätze durch Änderung des Schlüsselfeldes der Datenbestand nicht, die Datenbank selbst aber bekommt ein neues „Gesicht“. So kann man das Feld mit den Firmennamen zum Schlüsselfeld machen und alle Datensätze mit dem gleichen Firmennamen in eine gemeinsame Gruppe sortieren.

Nach diesem Rezept ist die Datenbank immer wieder neu zu ordnen. Die Datensatzfelder mit den Markennamen der Privatautos oder dem Namen des Wohnortes könnten ebenso zum Schlüsselfeld erklärt werden. Diese enorme Flexibilität ist der große Vorteil von Computer-Datenbanken.

Dienstleistungsangebote wie der Bildschirmtext der Post machen umfangreiche Datenbanken über den eigenen Fernseher zugänglich. Die Eingabe erfolgt über eine spezielle Tastatur, die über die Telefonleitung den Zugang zu einem Großrechner ermöglicht. Der TV-Bildschirm dient als Monitor. Über ein Menü kann der Teilnehmer die für ihn interessanten Teile der Datenbank auswählen.

Prestel LTD



Der Zeit voraus mit BASIC

Ein interessantes Programm, das mit den neuen Befehlen DIM, READ, und DATA die verbleibenden Tage bis Weihnachten zählt.

Das Programm beginnt mit der Auflistung der verwendeten Variablen. Wenn das Programm läuft, erscheinen auf dem Bildschirm als erstes die PRINT-Anweisungen, die in Zeile 230 beginnen. Sie sagen in kurzer Form, was das Programm bezweckt, und veranlassen den Anwender, das Datum in der gezeigten Form einzutippen. Dabei sind Kommas zum Trennen des Tages, des Monats und des Jahres zu verwenden.

Die erste bisher nicht bekannte Anweisung ist in Zeile 300. Sie lautet DIM (abgeleitet von Dimension = Abmessung) und dient dazu, die Anzahl von Positionen oder Elementen abzugrenzen, die mit der Variablen X dimensioniert werden. Die Feldvariable, auch Zuordnungs- oder Dimensionierungsvariable genannt, wird im Programmablauf wie eine normale Variable eingesetzt. Der Unterschied besteht darin, daß der Variablen in Klammern eingeschlossene Werte folgen. In Zeile 300 wird ein solcher Kasten X mit 13 Abteilungen konstruiert.

```
310 INPUT D, M$, Y
```

Diese Zeile ist eine gewöhnliche INPUT-Anweisung mit der Ausnahme, daß drei Eingabewerte erwartet werden. D ist eine numerische Variable, die das Tagesdatum zum Inhalt hat, Y eine weitere numerische Variable für die Jahreszahl. M\$ dagegen stellt eine kleine Besonderheit dar; es handelt sich hier um eine sogenannte Zeichenfolgevariable, deren Erkennungsmerkmal das Dollarzeichen \$ ist. Eine solche Variable akzeptiert sowohl Buchstaben als auch Zahlen. Tippen wir 23, JANUAR, 1984 ein, so wird die Zahl 23 der Variablen D, die Buchstabenfolge JANUAR der Variablen M\$ und die Zahl 1984 der Variablen Y zugeordnet.

```
330 GOSUB 560 REM 'MONATS-NR.'-ROUTINE
```

Mit dieser Anweisung wird das Programm veranlaßt, auf die in Zeile 560 beginnende Subroutine überzugehen. Hier ist zu bemerken, daß REM auf derselben Zeile steht. Dies ist zulässig, wenn die Zeile genügend Platz dafür bietet. Es ist also nicht unbedingt erforderlich, für REM immer eine neue Zeile zu wählen.

Die Subroutine, so einfach sie auch ist, zeigt sehr überzeugend, wie gut BASIC mit Buchsta-

benketten umzugehen weiß. Nehmen wir zum Beispiel den JANUAR als den Monatsteil der INPUT-Anweisung. Die Variable M\$ wird dann der Buchstabenkette JANUAR zugewiesen. Die erste Zeile der Subroutine lautet:

```
560 IF M$ = "JANUAR" THEN LET M = 1
```

Diese Anweisung vergleicht den Inhalt von M\$ mit der innerhalb der Anführungszeichen stehenden Buchstabenkette. Besteht Gleichheit, wie in unserem Fall, wird für die numerische Variable M der Wert 1 gesetzt. Man darf aber die Variable M nicht mit der Variablen M\$ verwechseln. Nur eine der Variablen kann eine Zeichenfolgevariable sein, und dies ist diejenige mit dem Dollarzeichen \$. Nachdem das Programm festgestellt hat, daß M\$ gleich der Buchstabenkette JANUAR ist, geht es zur nächsten Zeile und überprüft, ob der Inhalt von M\$ gleich ist mit dem Wort FEBRUAR. Erst wenn Übereinstimmung besteht, wird für M ein Wert gesetzt, und dieser Wert ist 1 für JANUAR, 3 für MAERZ usw.

Die Zeilen 350 bis 370 ergeben eine FOR-NEXT-Schleife, die den Wert I von 1 bis 13 steigert. Die Variable I dient als Index-Variable (auch Subskript genannt) der X-Reihe in Zeile 360 und muß genau beachtet werden.

```
360 READ X(I)
```

READ ist eine neue Anweisung, die immer in Verbindung mit einer entsprechenden DATA-Anweisung steht. Die DATA-Anweisung hierzu findet sich in Zeile 510:

```
DATA 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 24, 0
```

Diese Zahlen, ausgenommen der beiden letzten, sind die Tage in den einzelnen Monaten des Jahres. Beide Zeilen, 360 und 510, sind das Äquivalent zu den 13 LET-Anweisungen

```
LET X(1) = 31
LET X(2) = 28
LET X(3) = 31
LET X(4) = 30
LET X(5) = 31
LET X(6) = 30
```




```
LET X(7) = 31
LET X(8) = 31
LET X(9) = 30
LET X(10) = 31
LET X(11) = 30
LET X(12) = 24
LET X(13) = 0
```

Die in Zeile 350 eingerichtete Schleife läßt I von 1 bis 13 aufwärtszählen, so daß X(I) für X(1), X(2), X(3) usw. eingesetzt werden kann. Hier nun ein sehr viel einfacheres Programm:

```
10 READ A, B, C
20 LET D = A+B+C
30 PRINT D
40 DATA 5, 10, 20
```

Die READ-Anweisung in Zeile 10 liest die erste DATA-Position in Zeile 40 und „schreibt“ den Wert in die erste Variable. Mit anderen Worten weist sie den Wert 5 der Variablen A zu. Dann liest READ die nächste DATA-Position und weist sie der nächsten Variablen zu. So wird mit diesem Programm A=5, B=10 und C=20; diese Werte werden dann addiert und das Ergebnis der Variablen D zugewiesen. Das Ergebnis, in diesem Fall 35, wird dann in Zeile 30 PRINT ausgewiesen.

Doch nun zurück zum Weihnachtsprogramm. Beim ersten Durchgang, der in Zeile 350 beginnenden Schleife, wird als Wert I die 1 gesetzt und Zeile 360 daher gleichwertig der Anweisung READ X(1). Die entsprechende DATA-Position in Zeile 510 ist 31 (die erste Position) und folglich wird X(1) gleich 31 gesetzt.

Beim zweiten Durchlauf der Schleife wird I zu 2, so daß Zeile 360 gleichwertig ist der Anweisung READ X(2). Die nächste DATA-Position in Zeile 510 ist 28 und damit wird X(2) zu 28. Auf diese Weise werden alle 13 „Abteilungen“ des imaginären Variablenkastens X mit der jeweiligen Anzahl der Monatstage aufgefüllt, mit Ausnahme der zwölften Abteilung, die nur 24 Tage enthält, und der Abteilung 13 mit dem Inhalt 0.

```
390 GOSUB 750 REM 'SCHALTJAHR'-ROUTINE
```

Diese Zeile dirigiert das Programm zu einer Subroutine, die nachprüft, ob das eingegebene Jahr ein Schaltjahr ist oder nicht.

```
750 LET Y = Y/4
760 LET Z=INT(Y)
770 IF Y-Z=0 THEN GOTO 790
780 RETURN
790 LET X(2)=X(2)+1
800 RETURN
```

Als Schaltjahr wird jedes Jahr betrachtet, dessen Jahreszahl sich ohne Rest durch 4 teilen läßt. Eine Jahrhundertzahl muß sich außerdem durch 400 teilen lassen, um als Schaltjahr qua-

```
100 REM AUFLISTUNG DER VARIABLEN
110 REM
120 REM D = HEUTIGES TAGESDATUM
130 REM M$ = MONATSNAME
140 REM Y = JAHR
150 REM I = INDEX 1
160 REM X = REIHE DER MONATSTAGE
170 REM R = RESTTAGE
180 REM M = LAUFENDE MONATSNUMMER
190 REM L = INDEX 2
200 REM Z = GANZZAHL VON Y/4
210 REM
220 REM
230 PRINT "DIESES PROGRAMM ERRECHNET"
240 PRINT "DIE ZAHL DER VERBLEIBENDEN TAGE"
250 PRINT "BIS WEIHNACHTEN"
260 PRINT
270 PRINT "GEBE TAGESDATUM, MONAT UND JAHR EIN"
280 PRINT "Z. B. 12, JULI, 1984"
290 PRINT
300 DIM X(13)
310 INPUT D, M$, Y
320 REM
330 GOSUB 560 REM 'MONATS-NR.'-ROUTINE
340 REM
350 FOR I = 1 TO 13
360 READ X(I)
370 NEXT I
380 REM
390 GOSUB 750 REM 'SCHALTJAHR'-ROUTINE
400 REM
410 LET R = X(M) - D
420 FOR L = M TO 11
430 LET M = M + 1
440 LET R = R + X(M)
450 NEXT L
460 REM
470 IF R = 1 THEN GOTO 500
480 PRINT "ES VERBLEIBEN"; R;"TAGE BIS WEIHNACHTEN"
490 GOTO 520
500 PRINT "ES VERBLEIBT 1 TAG BIS WEIHNACHTEN"
510 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 24, 0
520 END
530 REM
540 REM
550 REM
560 IF M$ = "JANUAR" THEN LET M = 1
570 IF M$ = "FEBRUAR" THEN LET M = 2
580 IF M$ = "MAERZ" THEN LET M = 3
590 IF M$ = "APRIL" THEN LET M = 4
600 IF M$ = "MAI" THEN LET M = 5
610 IF M$ = "JUNI" THEN LET M = 6
620 IF M$ = "JULI" THEN LET M = 7
630 IF M$ = "AUGUST" THEN LET M = 8
640 IF M$ = "SEPTEMBER" THEN LET M = 9
650 IF M$ = "OKTOBER" THEN LET M = 10
660 IF M$ = "NOVEMBER" THEN LET M = 11
670 IF M$ = "DEZEMBER" THEN LET M = 12
680 RETURN
690 REM
700 REM ANMERKUNG: DIESE ROUTINE PRUEFT NICHT
710 REM SCHALTJAHRE AM ENDE
720 REM EINES JAHRHUNDERTS
750 LET Y = Y/4
760 LET Z = INT (Y)
770 IF Y - Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```




lifiziert zu werden. Zeile 750 dividiert Y (Jahreszahl) durch 4. Der neue Wert Y wird eine ganze Zahl sein, wenn diese Jahreszahl ohne Rest durch 4 teilbar ist, anderenfalls erhält man einen Dezimalbruch.

Zeile 760 verwendet die Funktion INT zum Auffinden des Integer-Wertes von Y, worunter man eine ganze Zahl versteht. Die Funktion INT, die auf ganze Zahlen keine Auswirkung hat, rundet Bruchzahlen auf die nächste ganze Zahl ab. Die abzurundende Zahl wird nach INT in Klammern gesetzt. Alternativ kann ein variabler Name in die Klammern gesetzt werden. LET Z = INT(496.25) wird auf diese Weise für Z den Wert 496 setzen.

Zeile 770 subtrahiert Z von Y und überprüft, ob das Ergebnis 0 ist. Wenn ja, bedeutet dies, daß das eingegebene Jahr ein Schaltjahr ist (weil der neue Wert Y kein Dezimalbruch ist). In diesem Fall geht das Programm aufgrund der Anweisung GOTO zu Zeile 790, die eine 1 zur zweiten Position der X-Reihe addiert (dies ist die 28, gleich der Tage im Monat Februar eines gewöhnlichen Jahres).

Ist das Subtraktionsergebnis in Zeile 770 nicht 0, bleibt X (2) wie es ist und die Subroutine kehrt auf das Hauptprogramm zur Zeile 400 zurück. In Zeile 400 dient ein REM wiederum lediglich dazu, das Programm übersichtlicher zu gestalten. Die nächste Zeile, in der etwas geschieht, ist die Zeile 410, worin R die den verbleibenden Tagen entsprechende Variable ist. Die Resttage ergeben sich aus der Anzahl der Tage des eingegebenen Monats minus dem eingegebenen Tagesdatum. Gibt man beispielsweise 12, Februar, 1984, ein, wird D zu 12 und M zu 2. Demzufolge wird X(M)

gleich sein X(2), was der zweiten Position in der X-Reihe mit 28 entspricht (da 1984 ein Schaltjahr ist, wird eine 1 hinzuaddiert). Folglich wird R zu 29 - 12 gesetzt, was 17 ergibt und die Anzahl der im laufenden Monat Februar verbleibenden Tage darstellt.

Mit Zeile 420 beginnt eine weitere Schleife, die dazu dient, den Wert M schrittweise zu steigern. Jetzt ist ersichtlich, warum es FOR L=1 TO 11 heißt und nicht FOR L = 1 TO 12. Ist M die Zahl 2, weil Februar eingegeben wurde, so wird Zeile 430 diese Zahl auf 3 erhöhen. Zeile 440 ergänzt dann R (Anzahl der Resttage) zu R plus X(M). Dieser Wert ist gleich X(3), weil M um den Wert 1 erhöht worden ist. X(3) entspricht 31, nämlich der Anzahl der Tage im März. In Zeile 440 wird demzufolge der neue Wert R als 17+31 gesetzt (17 war das Ergebnis von 29-12). Beim nächsten Schleifendurchgang wird M auf 4 gesteigert und die Anzahl der Apriltage, X(4), dem alten R-Wert hinzugefügt. Damit wird die Variable R zu 16+31+30.

Der letzte Durchgang durch die Schleife findet statt, wenn L=11 und der Wert für X(12), nämlich 24, zu R addiert wird. Was geschieht nun, wenn ein Dezemberdatum eingegeben wird, also M zu 12 wird? Einige Computer werden die Schleife vollständig überspringen, während andere einen Schleifendurchlauf vollziehen, so daß X(13) zu R addiert wird. Um das korrekte Ergebnis zu erhalten, wurde X(13) gleich 0 gesetzt.

```
470 IF R = 1 THEN GOTO 500
```

Diese Zeile hat lediglich die Aufgabe, festzustellen, ob es nur noch ein Tag bis Weihnachten ist, so daß ein grammatikalisch richtiger Satz auf dem Bildschirm erscheint. Ist R nicht 1, muß die bis Weihnachten verbleibende Tageszahl größer als 1 sein und die PRINT-Anweisung in Zeile 480 wird grammatikalisch richtig erscheinen.

BASIC ist in der Verarbeitung der LET-Anweisung recht widersprüchlich; wenn Zeilen wie IF M\$ = "SEPTEMBER" THEN LET M=9 Ihrem Computer Probleme bereiten, kann die Subroutine auch folgendermaßen geschrieben werden:

```
560 IF M$ = "JANUAR" THEN GOSUB 900
570 IF M$ = "FEBRUAR" THEN GOSUB 910
580 IF M$ = "MAERZ" THEN GOSUB 920
```

```
900 LET M = 1
905 RETURN
910 LET M = 2
915 RETURN
920 LET M = 3
(... usw.)
```

Diese Lösung ist aufwendiger und, mit all ihren GOSUBs und RETURNS, unübersichtlicher.

BASIC-Dialekte

READ DATA

Diese Befehle gibt es nicht auf dem ZX81: Deshalb sind die Zeilen 300, 350 bis 370 und 510 zu löschen und folgende hinzuzufügen:

```
10 DIM X(13)
20 FOR K = 1 TO 13
30 PRINT „GEBE POSITION NR. EIN“; K
40 INPUT X(K)
50 NEXT K
60 STOP
```

Jetzt das Programm laufen lassen (RUN) und die Daten eingeben. Nun die Zeilen 10 bis 60 löschen und das Programm speichern (SAVE), wobei auch der Inhalt der X-Reihe gesichert wird. Bei einem späteren Laden (LOAD) des Programms GOTO 100 und nicht RUN verwenden, um die Variablen zu erhalten.

INPUT

Beim ZX81 ist für jede Position ein INPUT-Befehl zu geben. Hinzuzufügen ist: 285 PRINT „WENN GEFRACHT“, und ebenso:

```
310 PRINT „GEBE TAG EIN“
312 INPUT D
```

mit ähnlichen Zeilen für Monat und Jahr.

REM

Beim BBC Micro, Commodore 64 und VC-20 muß vor REM-Anweisungen am Ende einer Programmzeile ein Doppelpunkt (:) gesetzt werden.

Auf zwei Rädern

Bodenroboter können Hindernisse geschickt umgehen. Doch sie sind nicht nur ein beliebtes Spielzeug, sondern vermitteln auf anschauliche Weise programmierte Bewegungsabläufe.

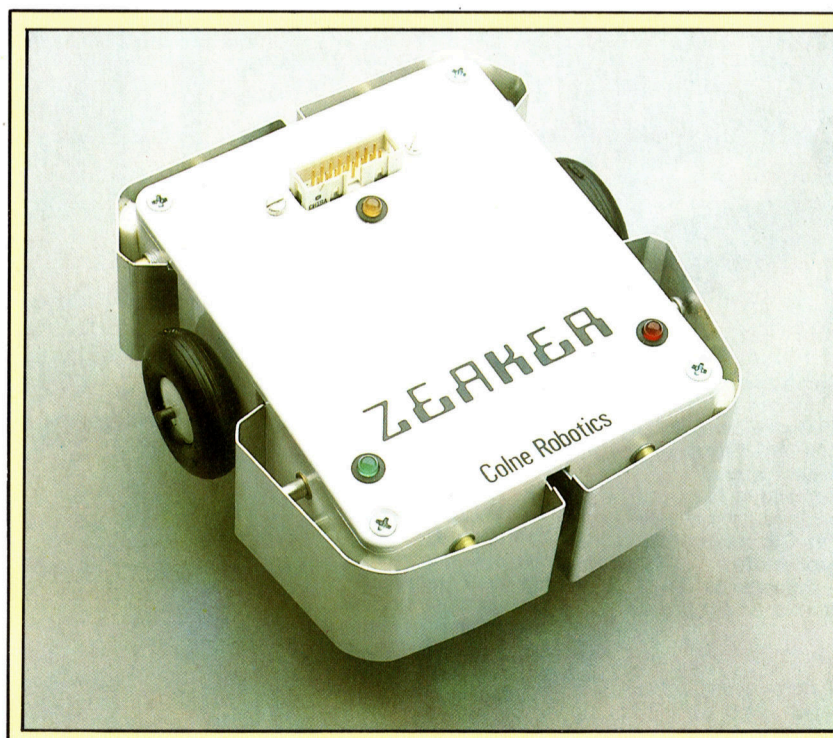
Mit Bodenrobotern und sogenannten „Schildkröten“ erhält man nicht nur einen Einblick in die Robotertechnologie, man kann auch viel Spaß damit haben. Eine kleine Maschine wie der BBC Buggy funktioniert auf die gleiche Weise wie ein großer industrieller Fertigungsroboter. Zwar können diese Roboter noch keine Hausarbeit erledigen, sie werden aber mit Sicherheit einen großen Einfluß auf die nächste Generation von Haushaltsgeräten ausüben.

Roboter müssen präzise Bewegungen ausführen und werden daher meist mit Schrittmotoren angetrieben. Schrittmotoren rotieren nicht wie normale Motoren, sondern drehen ihre Nabe bei jedem Stromimpuls um einen exakt festgelegten Bruchteil eines Vollkreises. Die Konstruktion der einzelnen Motorentypen bestimmt, wieviele Impulse ein Schrittmotor für eine ganze Umdrehung benötigt. Die Umdrehungsrichtung kann ebenfalls gesteuert werden. Über den unabhängigen Betrieb der Antriebsmotoren werden die Bewegungen eines Bodenroboters präzise kontrolliert. So kann er auf der Stelle rotieren, indem sich die entsprechenden zwei Räder einfach im gleichen Takt in gegensätzliche Richtungen drehen.

Ebenso wichtig wie die präzise Kontrolle der Bewegungen ist die Entdeckung von Hindernissen in der Bahn des Bodenroboters. Zusammenstöße werden generell über Stoßdämpfer entdeckt, die um den Roboter herum angeordnet sind und per Microschalter die Stöße in Signale umwandeln. Diese sind wiederum über eine Schnittstelle mit dem Computer verbunden, der jede Bewegung der Schalter als Wechsel des Bitstatus erkennt.

Der Roboter kann auch einer weißen Linie auf dem Boden folgen. Diese Fähigkeit wird durch eine Lichtquelle ermöglicht, die auf der Unterseite des Roboters auf den Boden scheint und deren Reflektionsstärke gemessen wird. Fährt der Roboter über unterschiedliche Oberflächen, werden die Schwankungen der Lichtintensität analog erfaßt und an den Steuercomputer weitergegeben. Der BBC Micro verfügt über eine analoge Schnittstelle, mit der diese Art der Steuerung direkt erfolgen kann. Bei den meisten anderen Systemen muß das analoge Signal erst in ein digitales umgewandelt werden, bevor es an den Computer weitergegeben werden kann.

Für die Steuerung von Bodenrobotern verwendet man auch Lichtsignale, Töne und ma-



Der Zeaker

Diese „Micro-Schildkröte“ besitzt Kollisionsfühler und verfügt über eine „Teilintelligenz“. Der Zeaker kann sich vor, zurück und nach beiden Seiten bewegen. Außerdem hebt und senkt er auf Befehl einen Zeichenstift. Der Zeaker wird über eine speziell angepasste Version von LOGO (LOGO SNAIL) kontrolliert, die Befehle wie FORWARD und BACKWARD direkt versteht. Andere Computer als der BBC Micro benötigen zur Steuerung ein Spezialinterface.



Schrittmotor

Beide Antriebsräder des Buggy verfügen über einen eigenen Schrittmotor. Ein Impuls, der an den Motor gesandt wird, dreht ein Rad um genau ein Grad weiter.

Steuerplatine

Alle Impulse von und zu dem Computer werden vor ihrer Weitergabe mit dieser Platine umgesetzt.

Antriebsrad

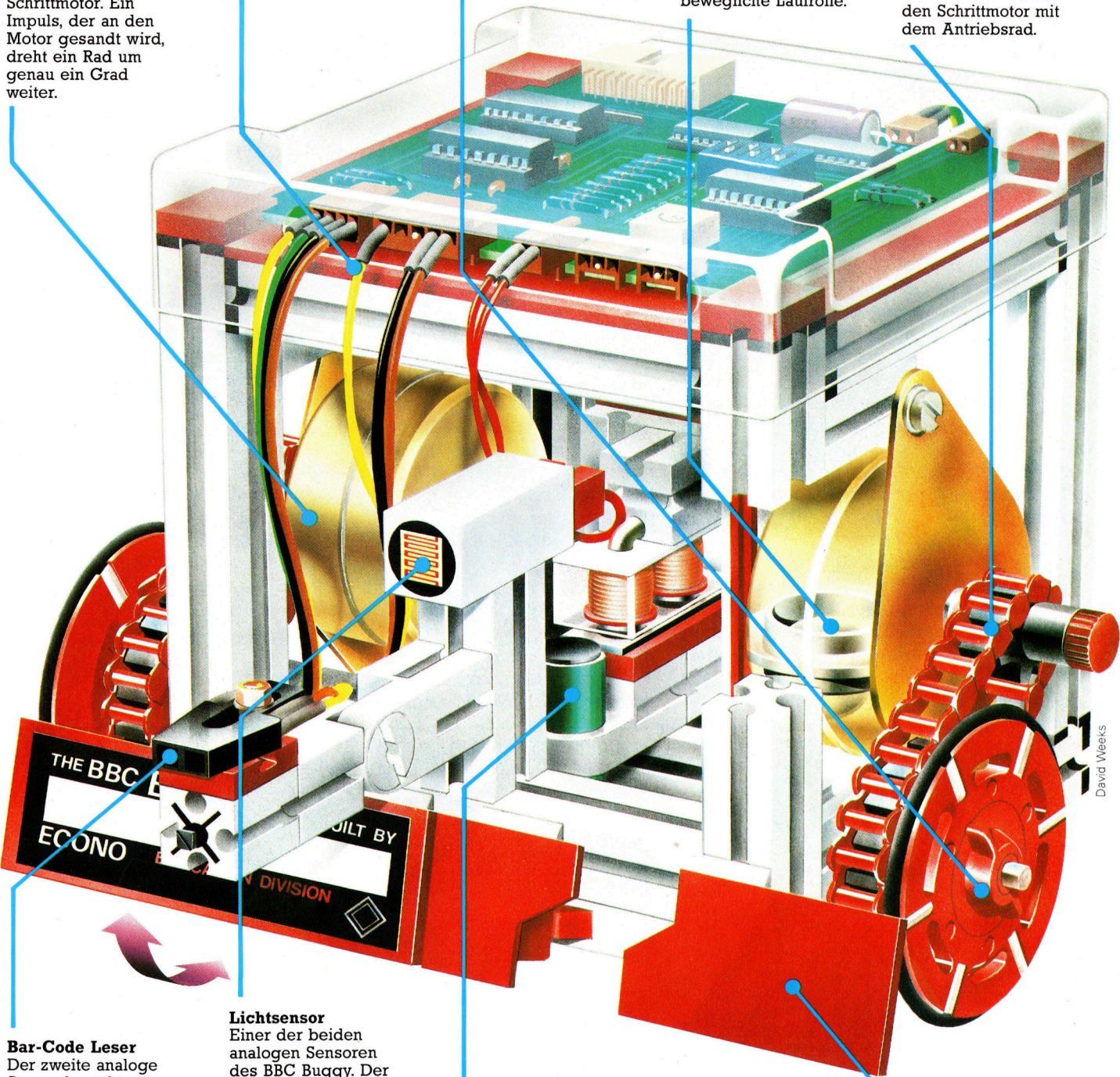
Gummibeschichtete Räder auf beiden Seiten des Buggy bewegen den Bodenroboter.

Hintere Laufrolle

Am Heck des dreieckigen Buggys befindet sich eine frei bewegliche Laufrolle.

Antriebskette

Die Kette verbindet den Schrittmotor mit dem Antriebsrad.



David Weeks

Bar-Code Leser

Der zweite analoge Sensor besteht aus einer Leuchtdiode (LED) und einer Photodiode (LDD). Das Licht der Leuchtdiode wird von dem Boden unter dem Buggy reflektiert und über die Photodiode empfangen, so daß der Buggy Linien auf dem Boden folgen kann.

Lichtsensor

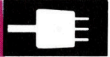
Einer der beiden analogen Sensoren des BBC Buggy. Der Sensor sendet Impulse an die analoge Schnittstelle des Computers – je größer die Lichtintensität, desto stärker ist der Impuls.

Halterung für einen Zeichenstift

Der BBC Buggy kann mit einer Halterung für einen Zeichenstift ausgerüstet werden. Damit können Linien auf die überquerten Flächen gezeichnet werden.

Vordere Kollisionsfühler

Der geteilte Stoßfänger ist auf jeder Seite mit Microschaltern ausgerüstet, die ein ON (Zusammenstoß) oder ein OFF (kein Zusammenstoß) an den Computer geben.



gnetische Felder. Die Industrie nutzt oft eine Kombination dieser Möglichkeiten, wenn Roboter festgelegten Wegen durch Lager- oder Fertigungshallen folgen müssen. Spezialkabel, die unter dem Boden verlegt sind, definieren einen bestimmten Pfad, dem Roboter mit ihren Sensoren folgen können.

Spezialroutinen

Bodenroboter werden meist mit Programmen kontrolliert, die speziell auf diese Art von Computern abgestimmt sind. Spezialroutinen steuern das Senden und Empfangen von Informationen, die der interne Computer über besondere Schnittstellen an die Maschine übermittelt. Im Falle des BBC Buggy wird ein 4 Bit-Code für die Kontrolle der Motoren verwandt. Über dasselbe Kabel empfängt der Computer auch die Daten der äußeren Sensoren. Die analogen Signale des Lichtsensors und des Bar-Code-Lesers werden über den analogen Eingang eingespeist und die Kollisionsfühler sind mit zwei Leitungen des Benutzer-Ein- und Ausgangs verbunden.

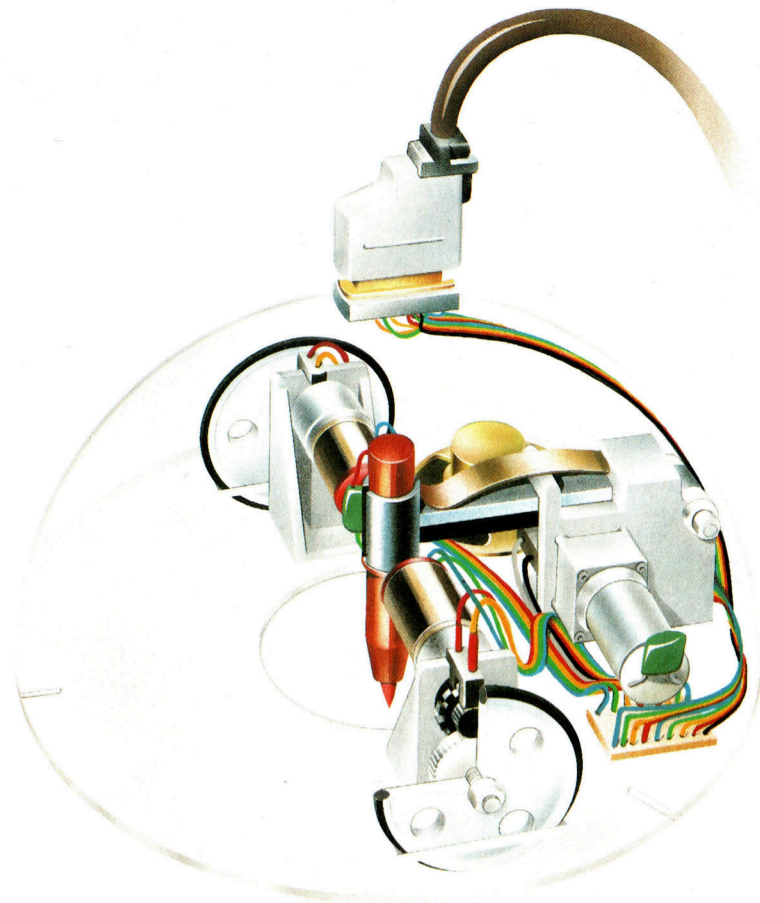
Die Signale einer Ein- und Ausgabeschnittstelle sind als Code in bestimmten Speicheradressen des Computers abgelegt. Der BASIC-Befehl PEEK z. B. zeigt den aktuellen Inhalt dieser Speicheradressen an. Mit dem BASIC-Befehl POKE kann ein Speicherinhalt verändert werden, beispielsweise wenn die Bewegungsrichtung der Schrittmotoren umgekehrt werden soll, damit sich der Roboter umdreht und in die andere Richtung fährt.

Analoge Informationen können auf ähnliche Weise untersucht werden, wenn in den Computer ein Analog/Digital-Wandler (A/D-Wandler) eingebaut ist. Verfügt der Computer nicht über einen internen A/D-Wandler, muß diese Wandlung über ein zusätzliches Interface erfolgen.

Eine „Schildkröte“ ist eine Art Bodenroboter, der in Verbindung mit der Programmiersprache LOGO verwendet wird. Es gibt jedoch keine klare Unterscheidung mehr zwischen Bodenroboter und „Schildkröte“.

Fahrendes Zeichengerät

Neue Schildkrötenmodelle sind häufig mit Kollisionsfühlern ausgerüstet; wie auch viele Bodenroboter inzwischen über Halterungen für Zeichengeräte verfügen. Die Idee, große Zeichnungen von Maschinen anfertigen zu lassen, die eigentlich ein mit Rädern versehenes Zeichengerät sind, stammt aus Lehrprogrammen, die die Beziehungen zwischen Abstand, Winkel und Form illustrieren. Wenn eine Person sich 10 Einheiten vorwärts bewegt, dann weitere 10 Einheiten nach links, wiederum 10 Einheiten nach links und nochmals 10 Einheiten nach links, dann ist die Person in einem Quadrat gewandert. Für die Darstellung dieser



David Weeks

Beziehung zwischen Bewegung und Form wird einfach ein Zeichenstift an einen Roboter angebaut, der den Weg auf Papier zeichnet.

Ist ein Roboter aktiv, muß das Programm ständig die Sensoren überprüfen, um zu „sehen“, ob eine Linie auf dem Boden (starke Lichtreflektion) existiert, oder ob sich ein Hindernis in seinem Weg befindet. Wird ein Hindernis entdeckt, muß der Roboter sofort reagieren, damit kein Schaden entsteht. Programme dieser Art werden „Echtzeitprogramme“ genannt, da Reaktionen unmittelbar erfolgen müssen.

Die Techniken, die im Spiel mit Bodenrobotern gelernt werden, vertiefen auch das Verständnis für die Funktionsweise der künstlichen Intelligenz. Es können Programme geschrieben werden, die einen Bodenroboter veranlassen, bestimmte Aufgaben so lange auszuführen, bis seine Batterien fast erschöpft sind. Der Roboter sucht sich dann automatisch eine Steckdose, über die er seine Batterien wieder auflädt und fährt danach mit seiner Tätigkeit fort.

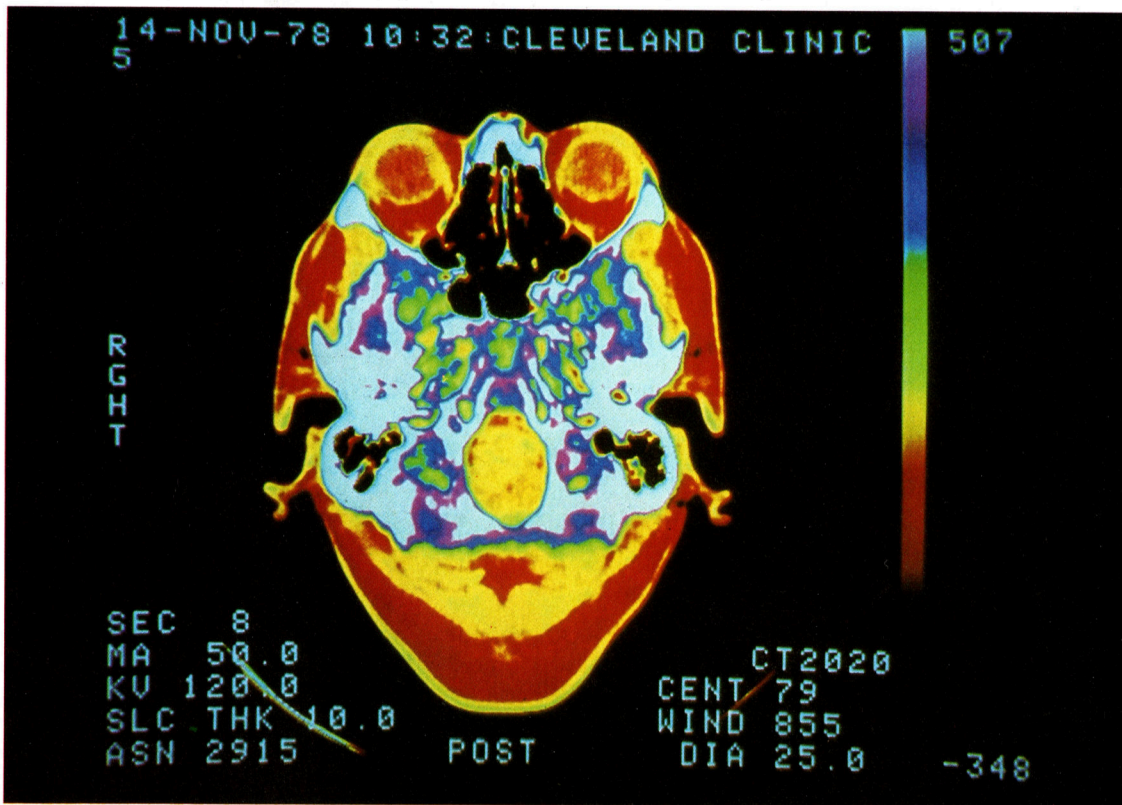
Die nächste Generation von Bodenrobotern wird über weitere erstaunliche Fähigkeiten verfügen. Die Roboter werden voraussichtlich mit Greifarmen ausgerüstet sein, mit denen sie leichte Lasten tragen können. Miniaturkameras werden die einfachen Lichtsensoren ersetzen und der Roboter wird „sehen“ können, wohin er sich bewegt.

Die Schildkröte ist eine besondere Konstruktionsart des Bodenroboters. Sie kann per Computersteuerung mit einem beweglichen Zeichenstift Linien auf den Boden zeichnen. Schildkröten werden meist über die Programmiersprache LOGO gesteuert, können aber auch mit BASIC programmiert werden.



Micro-Medizin

Auch im medizinischen Bereich haben Computer längst Einzug gehalten. Sowohl dem Arzt als auch dem Patienten kommt die neue Technologie auf vielfältige Weise zugute.



Konventionelle Röntgenbilder produzieren ein flaches, zweidimensionales Bild, auf dem sich alle Organe überlagert darstellen. Ein weitaus genaueres Bild kann in Form von Schichten durch den Körper aufgebaut werden, indem man den Körper schichtweise mit einem Röntgenstrahl abtastet. Die Daten werden dann von einem Computer verarbeitet, um die gespeicherten Signale als Bild auf dem Monitor darzustellen.



Das Monitorsystem zeigt das derzeitige Befinden des Patienten, kann jedoch auch Informationen für spätere Analysen speichern. Hier bekommt ein Arzt gerade Informationen über den Zustand des Patienten während der Nacht.

Computer sind im medizinischen Bereich von fast unschätzbarem Wert. Kurz nach der generellen Einführung von Mikroprozessoren hat sich zum Beispiel die Arbeitsweise auf Intensivstationen beträchtlich verändert. Microcomputer mit Monitoren werden zur Registrierung von Blutdruck, Puls und Atemfrequenz eingesetzt, und sie ermöglichen den sofortigen Abruf aller gewünschten Daten.

Neuerdings werden Mikroprozessoren in Arztpraxen, innerhalb der Krankenhausverwaltung bei der Patientenregistrierung, Terminvergabe sowie zur Überprüfung der Arzneimittelbestände eingesetzt.

Immer öfter kommen besonders hochentwickelte Programme zum Einsatz. „Mickie“ beispielsweise ist ein simples aber erfolgreiches Expertensystem, das praktischen Ärzten Überblick über die momentane körperliche Verfassung ihrer Patienten gibt. Es versorgt den Arzt allerdings mit sehr allgemeinen Informationen und liefert keine endgültigen Diagnosen. Die Fragen werden vom Computer so gestellt, daß sie mit „ja“, „nein“, „Ich weiß

nicht“ oder „Ich verstehe Sie nicht“ beantwortet werden können. Lautet die Antwort „Ich verstehe Sie nicht“, so bemüht sich der Computer, dem Patienten zu einer Antwort zu verhelfen. Der Patient wird nicht mit einer kompletten Schreibmaschinentastatur verwirrt. Ein Apparat mit vier Knöpfen bietet die entsprechenden Antwortmöglichkeiten an.

Das System läuft sehr langsam, um der durchschnittlichen Lesegeschwindigkeit zu entsprechen. Die nächste Entwicklungsstufe des Expertensystems für praktische Ärzte wird daraus bestehen, neu gewonnene Informationen mit bereits bekannten Fakten zu kombinieren. Wenn ein Patient über dieselben Beschwerden wie beim letzten Arztbesuch klagt, so kann es sein, daß der Arzt zur Diagnosefindung nur noch die Frage „Fühlen Sie sich wie am . . .?“ stellen muß.

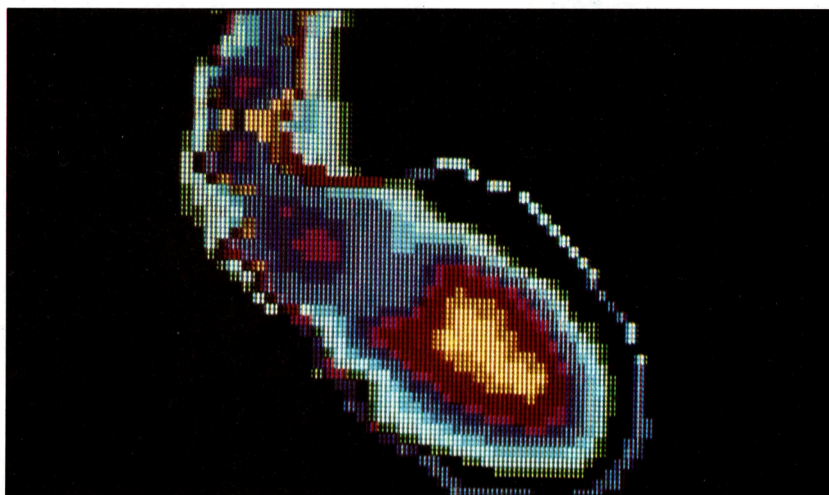
Andere Vorteile, die in Computern gespeicherte medizinische Daten bieten, liegen im Bereich des Datenschutzes. Es ist viel schwerer, von einer Floppy Disk Informationen abzulesen, als von einem Blatt Papier, selbst wenn



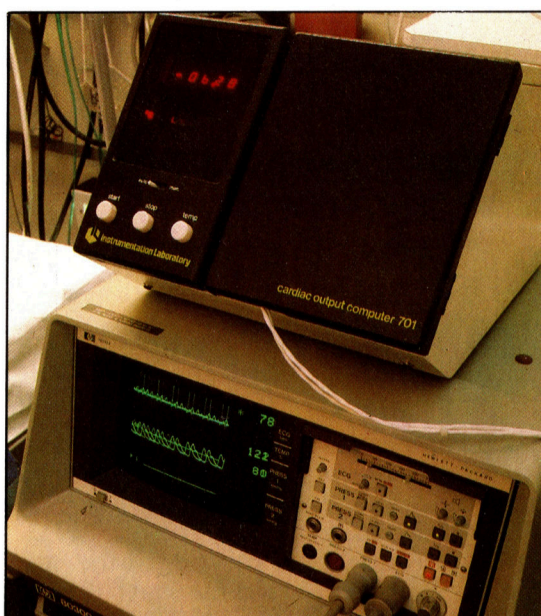
man etwas von Computern versteht. Die signifikantesten Vorteile liegen jedoch im Diagnostikbereich. Bis vor kurzem noch war die Untersuchung innerer Organe nicht ungefährlich. Man denke an Röntgenbilder (Strahlenbelastung), Endoskopie (mögliche innere Verletzungen von Weichteilgewebe) und explorative Operationen. Ein bedeutender Fortschritt, der sowohl die Genauigkeit als auch die Sicherheit betrifft, ist die Entwicklung von computergespeicherten Schichtaufnahmen (CT), die sich einer dünnen Schicht von Röntgenstrahlen bedienen, anstatt den Körper ungefilterten Strahlen auszusetzen. Absolut ungefährliche Wege, innere Untersuchungen durchzuführen, bieten alternative Methoden wie der Ultraschall.

Auch die medizinische Notversorgung profitiert von der Computerisierung. Unvorstellbar ist zum Beispiel, daß das britische Organ-Transplantations-Programm ohne Computer-Datenbanken so weit fortgeschritten wäre. Andere spektakuläre Fortschritte sind bei der Behandlung körperlich und geistig Behinderter erzielt worden. Obwohl noch in den Anfängen, ist gerade in diesem Bereich eine enorme Entwicklung abzusehen.

Man stelle sich zum Beispiel die Freude eines stummen querschnittgelähmten Menschen vor, der über eine Computerregistrierung der Augenbewegungen, die zu Sprache umgesetzt wird, zum ersten Mal in seinem Leben sprechen kann. Bei der Messung der Augenbewegung wird entlang einer Linie die Be-

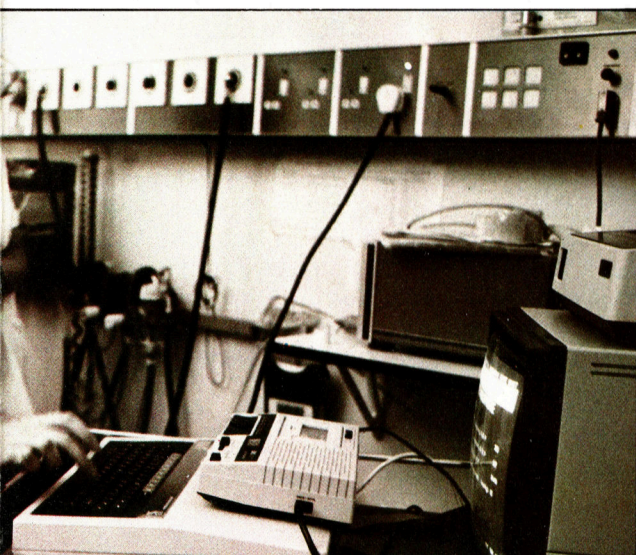


Ian Dobbie



Kernspintomographen, wie dieser hier, sind ebenso gut wie die Computertomographie. Da sie keine Strahlen verwenden, sind sie wesentlich sicherer. Momentan befindet sich diese Technik noch im Experimentierstadium.

Microprozessoren, die über Sensoren am Körper Informationen über den Zustand des Patienten erhalten, setzen diese Daten um und stellen sie auf einem speziell modifizierten Oszillographen dar, ähnlich denjenigen, die von Elektroingenieuren verwendet werden.



Ian Dobbie

wegung des Augapfels beobachtet. Von einem Referenzpunkt aus ist es möglich, die momentane Position zu bestimmen und so den Informationsgehalt abzulesen. Sprachsynthesizer-Chips enthalten die Sprachbausteine im ROM, die Phoneme genannt werden. Unter Programmkontrolle werden individuelle Phoneme zu Worten zusammengesetzt. Dies kann als Grundlage eines computerisierten Kommunikationssystems dienen, beispielsweise bei Behinderten, die sich überhaupt nicht bewegen

können. Der bekannteste Versuch auf diesem Gebiet ist, eine Standard-Tastatur durch eine den jeweiligen Anforderungen entsprechende Einheit zu ersetzen. Zum Beispiel können für Patienten mit multipler Sklerose oder lokomotorischer Ataxie sehr große Bedienungsknöpfe konstruiert werden.

Die Maschine kann eine vorbestimmte Anzahl von Möglichkeiten anbieten, zum Beispiel die Zusammenstellung eines Menüs. Indem man den jeweils geeigneten Knopf drückt, teilt der Bediener dem Computer mit, was er essen möchte. Wenn keine Reaktion mehr folgt, nimmt der Computer eine negative Antwort an. In diesem Fall wird das Keyboard aus nur einer einzigen Taste bestehen, die vielleicht nur durch eine leichte Kopfbewegung ausgelöst werden kann. Als neueste Errungenschaft bietet diese Möglichkeit dem Anwender den Einsatz eines Textverarbeitungssystems, so daß beispielsweise contergangeschädigte Menschen selbst schreiben können – ein enormer Fortschritt, der ohne den Computer nicht hätte erreicht werden können.



Bogen und Listenverarbeitung

Interessante Computergrafiken mit Bogen- und Kreisprozeduren. Raffinierte Wortspielereien mit Listenverarbeitung.

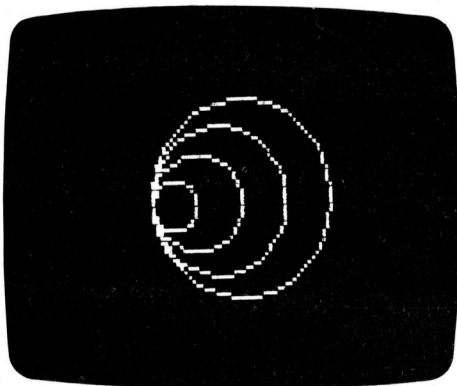
Die Prozedur KREIS :GROESSE haben wir bereits besprochen – doch dabei gibt es leider den Nachteil, daß der Kreis nur rechts herum gezeichnet wird. Was also ist zu tun, um einen Kreis in der entgegengesetzten Richtung zu zeichnen? Für diesen Zweck ist es sinnvoll, zwei sogenannte Kreisprozeduren zu definieren – eine, die mit einem rechten Bogen, und eine andere, die mit einem linken Bogen zeichnet:

```
TO KREIS.RE :GROESSE
  REPEAT 36 [FD :GROESSE RT 10]
END
```

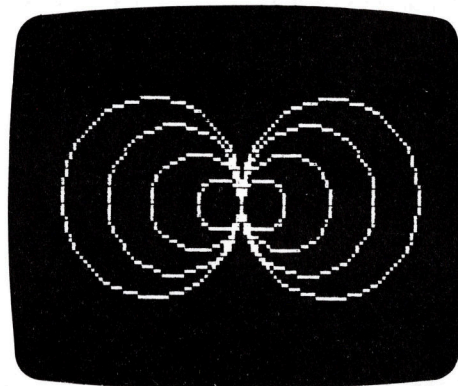
```
TO KREIS.LI :GROESSE
  REPEAT 36 [FD :GROESSE LT 10]
END
```

Wichtig – innerhalb einer Prozedurdefinition darf kein Leerraum auftauchen. Deshalb der Punkt zwischen KREIS und RE bzw. LI.

Erinnern Sie sich an die Prozedur SUPERVIERECK. Sie sollten jetzt in der Lage sein, eine SUPERVIERECK-Prozedur unter Verwendung der Prozedur KREIS :GROESSE zu schreiben. Die SUPERVIERECK-Prozedur ruft dabei KREIS :GROESSE auf. Ist Ihnen das gelungen, so erscheint auf dem Bildschirm folgendes Muster:



Sie sehen, der Superkreis unter Verwendung von KREIS.RE :GROESSE und KREIS.LI :GROESSE funktioniert. Deshalb wagen wir uns nun an einen doppelten SUPERKREIS, der so aussieht:



Es gibt mindestens zwei Möglichkeiten, diesen doppelten Superkreis zu zeichnen:

1. Zeichnen Sie zuerst die Kreise auf der rechten und dann die auf der linken Seite.
2. Zeichnen Sie erst die kleinsten Kreise und dann die nächst größeren.

Bogenprozeduren

Eigentlich ist es sehr einfach, die Bogenprozeduren von den Kreisprozeduren abzuleiten. Hilfreich dabei ist, eine präzise Vorstellung für die Teile einer Kreisprozedur und deren Funktion zu entwickeln. Nehmen Sie z. B. KREIS.RE :GROESSE. Was passiert, wenn Sie hier eine Eingabe verändern? Welche Eingabe ist für die Rückkehr des Igels an den Ausgangspunkt zuständig? Welche Eingabe bestimmt die Größe des Kreises?

Halbieren Sie einfach die Eingabe 36 für REPEAT, während alle anderen Eingaben konstant bleiben. Zum Ausprobieren geben Sie dem Igel erst einmal direkte Befehle:

```
REPEAT 36 / 2 [FD 5 RT 10] oder
```

```
REPEAT 18 [FD 5 RT 10]
```

Sie sehen jetzt einen Halbkreis auf dem Bildschirm? Nun versuchen wir uns an einem Viertelkreis:

```
REPEAT 36 / 4 [FD 5 RT 10]
```

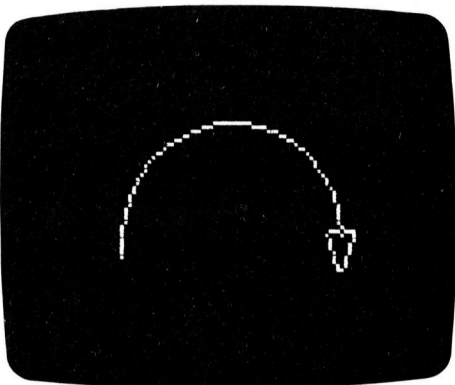
Jetzt werden wir mit Variablen eine Prozedur so definieren, daß eine unbegrenzte Anzahl



verschiedener Bögen entsteht. Haben Sie sich dann mit dem Zeichnen von Bögen angefreundet, so werden Sie schnell den Nutzen von Bögen erkennen, die sowohl nach links als auch nach rechts gezeichnet werden. Am besten, Sie beginnen Ihre Versuche mit einem Halbkreis, da dieser auch später von Ihnen am meisten genutzt wird. Hier zwei Prozeduren für das Zeichnen von Halbkreisen:

```
TO BOGEN.RE :GROESSE
  REPEAT 18 [FD :GROESSE RT 10]
END
```

```
TO BOGEN.LI :GROESSE
  REPEAT 18 [FD :GROESSE LT 10]
END
```



Nun können Sie eine Anzahl unterschiedlich großer Bögen abrufen und mit diesen entsprechende Muster zeichnen. Hier ein Vorschlag:

```
REPEAT 2 [BOGEN.RE 2 RT 180]
```

Listenverarbeitung

Die grafischen Fähigkeiten von LOGO waren bislang das Hauptthema unseres Kurses. In diesem Abschnitt wollen wir Sie in die Welt der Listenverarbeitung einführen.

Die grundlegende Einheit einer LOGO-Struktur wird als „Wort“ bezeichnet; mehrere „Worte“ wiederum bilden „Listen“. Unter dem Begriff Listenverarbeitung versteht man die Art, wie diese Strukturen bearbeitet und verändert werden. Ein LOGO-Wort ist aus einer Folge von Zeichen aufgebaut, mit Ausnahme des Leerzeichens, sowie der linken und rechten eckigen Klammer. Beispielsweise sind folgende Worte für LOGO gültig:

```
EINGABE
234567
+
STEFAN
NETTO.PREIS
```

Die oben angeführten Worte ähneln auf den

ersten Blick einer LOGO-Prozedur, einfachen Zahlen bzw. arithmetischen Operationen. Um aber zwischen einem Wort und anderen Elementen unterscheiden zu können, verlangt LOGO ein Anführungszeichen vor dem Wort. Geben Sie ein:

```
PRINT STEFAN
```

LOGO antwortet mit der Fehlermeldung:

```
THERE IS NO PROCEDURE NAMED STEFAN
```

Deshalb versuchen Sie jetzt:

```
PRINT "STEFAN
```

Dieser Befehl gibt den Text STEFAN auf dem Bildschirm aus. Die Anweisung PRINT " stellt eine Leerzeile auf dem Screen dar. Bislang konnten Sie nur ein Wort pro Zeile ausgeben. Was kann getan werden, um mehrere Worte in eine Linie zu bekommen? Der Befehl

```
PRINT1
```

schreibt das nachfolgende Wort und belässt den Cursor an der jetzigen Stelle. Die Anweisung

```
PRINT1 "HALLO PRINT1 "LEUTE
```

schreibt den Text HALLOLEUTE auf den Bildschirm. Leider fehlt noch ein Leerzeichen zwischen den beiden Worten, außerdem kostet es sehr viel Zeit, immer PRINT1 schreiben zu müssen. LOGO bietet daher die Möglichkeit, mehrere Worte mit nur einem PRINT-Befehl zu schreiben, wenn der gesamte Ausdruck in eine Klammer gesetzt wird:

```
(PRINT "HALLO "LEUTE)
```

gibt wie gewünscht den Text HALLO LEUTE auf einer Zeile aus, auch der notwendige Zwischenraum wird eingefügt. LOGO bietet außerdem eine Prozedur, die zwei separat eingegebene Worte zu einem zusammenfügt. Beispielsweise erzeugt der Befehl

```
PRINT WORD "HA "LLO
```

das gleiche Ergebnis wie die Anweisung

```
PRINT "HALLO
```

Das „Aneinanderreihen“ von Texten wird auch als Verknüpfung bezeichnet. Versuchen Sie jetzt die Eingabe:

```
PRINT (WORD "UN "GE "HOR "SAM)
```

und auf dem Bildschirm erscheint das zusammengefügte Wort UNGEHORSAM.

LOGO verarbeitet aber nicht nur Zahlen und Worte, sondern auch „Listen“, die vielseitig einsetzbar sind. Eine Liste ist eine Zusammenstellung von Zahlen oder Worten. Die Reihenfolge der Zahlen und Worte ist sehr wichtig. In LOGO werden Listen durch eckige Klammern definiert, das heißt, alle Elemente einer Liste müssen innerhalb dieser Klammern stehen, wie zum Beispiel:

```
[[ 0 1 2 3 4] [4 3 2 1 0]]
```

Diese Liste besteht aus zwei separaten Listen, nämlich [01234] und [43210]. Beide enthalten fünf Elemente. Ebenso wie LOGO-Worte, können auch diese leer sein. Die folgende Anweisung setzt die Variable E.LIST als eine sogenannte leere Liste:

```
MAKE "E.LIST [ ]
```

Nun ein Beispiel, wie Listen Variablen zugeordnet werden:

```
MAKE "SATZ [JETZT DRUCKE ICH EINEN SATZ]
PRINT :SATZ
```

Erinnern Sie sich noch an das REPEAT-Kommando? Mit REPEAT können natürlich auch Listen bearbeitet und wiederholt werden. Voraussetzung dafür ist, daß man zuerst einer Variablen die Befehle zuweist, die durch REPEAT-Befehl wiederholt werden sollen.

```
MAKE "BEFEHL [FD 20 RT 90 FD 60]
REPEAT 2 :BEFEHL
```

So wie wir neue Worte aus anderen erzeugen können, erlaubt LOGO, neue Listen zu definieren. Um neue Listen erstellen zu können, benutzt man die LOGO-Prozeduren LIST und SENTENCE. LIST verarbeitet normalerweise zwei Eingaben und Ausgaben:

```
input
LIST "GUTEN "TAG
LIST [GUTEN TAG] [MEIN HERR]
(LIST 1 2 3 4 5)
```

```
output
[GUTEN TAG]
[[GUTEN TAG] [MEIN HERR]]
[1 2 3 4 5]
```

SENTENCE (Kurzform = SE) verarbeitet ebenfalls zwei Eingaben und gibt eine Liste aus. Wenn die Eingaben ihrerseits Listen sind, werden sie zunächst wieder in ihre ursprünglichen Elemente zerlegt:

```
input
SENTENCE "MICKY "MOUSE
SENTENCE [GUTEN TAG] "CHEF
SENTENCE [GUTEN TAG] [MEIN HERR]
```

```
output
[MICKY MOUSE]
[GUTEN TAG CHEF]
[GUTEN TAG MEIN HERR]
```

Wie der Name schon andeutet, ist SENTENCE mehr gebräuchlich bei Listen, die Text-Zeichenketten verwenden. Der LOGO-Befehlssatz enthält auch Möglichkeiten, Listen zu verändern, etwa mit:

FIRST

Diese Prozedur verarbeitet als Eingabe eine Liste und gibt das erste Element aus. Sinnvollerweise sollte die Eingabe keine leere Liste darstellen. Bei

```
PRINT FIRST [EINS ZWEI DREI]
```

wird EINS auf dem Bildschirm ausgegeben.

BUTFIRST

(abgekürzt BF) dagegen gibt die eingegebene Liste mit Ausnahme des ersten Elementes aus. Ist nur ein Element vorhanden, bekommt man als Ausgabe eine leere Liste.

```
PRINT BUTFIRST [EINS ZWEI DREI]
```

zeigt ZWEI DREI auf dem Bildschirm.

LAST

macht genau das Gegenteil von FIRST. Es wird also nur das letzte Element der Liste auf dem Bildschirm ausgegeben.

BUTLAST

Diese Prozedur verhält sich entgegengesetzt von BUTFIRST und verhindert die Ausgabe des letzten Elementes einer Liste. Um die Effizienz der Listenverarbeitung zu sehen, geben Sie nun unser Beispielprogramm QUIZ ein:

```
TO QUIZ
FRAGE [WER IST DER VATER DER LOGO
PHILOSOPHIE] [SEYMOUR PAPERT]
FRAGE [WAS ERGIBT 4+4] [8]
FRAGE [WIE HEISST DAS CHEMISCHE
ZEICHEN FUER WASSERSTOFF] [H]
END
```

```
TO FRAGE :QUESTION :ANSWER
TYPE :QUESTION
IF RL = :ANSWER
[PRINT [RICHTIG]] [PRINT [FALSCH.
DIE RICHTIGE ANTWORT IST] PRINT
:ANSWER]
END
```

Weiter geht es mit Bedingungen.

LOGO-Befehle

FIRST
Bei Eingabe einer Liste bewirkt FIRST, daß das jeweils erste Element ausgegeben wird. Gibt man dagegen ein Wort ein, dann veranlaßt FIRST, daß das erste Zeichen des Wortes ausgegeben wird.

LAST
LAST bewirkt genau das Gegenteil von FIRST. Dies bedeutet, daß jeweils nur das letzte Element der Liste, bzw. das letzte Zeichen des Wortes, ausgegeben wird.

BUTFIRST
Bei der Eingabe einer Liste bewirkt BUTFIRST, daß alle Elemente bis auf das erste ausgegeben werden. Wird aber ein Wort eingegeben, veranlaßt BUTFIRST, daß alle Zeichen bis auf das erste ausgegeben werden.

BUTLAST
BUTLAST macht das Gegenteil von BUTFIRST. Alle Elemente, bzw. alle Zeichen des Wortes, mit Ausnahme des jeweils letzten werden ausgegeben.

SENTENCE
SENTENCE verarbeitet Eingaben und gibt sie als Liste aus.

„Reise“ per Telefon

Ein Modem kann Computer-Informationen über Tausende von Kilometern übertragen. Es ermöglicht weltweiten Datentransfer.



Ian Dobbie

Die meisten Modems verbinden einen Computer per Stecker direkt mit dem Telefonnetz. Die Post fordert für Geräte, die an ihr Netz angeschlossen werden, einen hohen technischen Standard, der die Herstellung teuer macht.

Der technische Begriff „Modem“ setzt sich aus den Wörtern „Modulator“ und „Demodulator“ zusammen. Obwohl schon seit über sieben Jahren auf dem Markt, setzen Besitzer von Microcomputern Modems erst seit kurzer Zeit in größerem Umfang ein, da sie nicht einsehen, warum sie Geld für ein Gerät ausgeben sollten, das ihre Maschine an das Telefonnetz anschließen kann.

Über ein Modem kann der Computer mit anderen Computern in der ganzen Welt kommunizieren. Der Rechner am anderen Ende der Leitung muß nur ebenfalls an ein Modem angeschlossen sein. Beim Empfangscomputer kann es sich auch um einen Großrechner in einer Universität oder in einer Firma handeln. Eine Verbindung mit Großcomputern kann den Zugang zu Datenbanken, zu Informationsdiensten oder auch zu den aktuellen Börsenkursen bedeuten, während beim Zusammenschluß mit dem Gerät eines Freundes beispielsweise Programme ausgetauscht werden können.

Modems funktionieren ähnlich wie ein Cassetteninterface, das heute in den meisten Heimcomputern standardmäßig eingebaut ist. Modem und Cassetteninterface formen die Nullen und Einsen des Computers in analoge Audiosignale um. Im Falle eines Cassetteninterfaces werden diese Signale wie normale Töne auf Band aufgezeichnet. Bei Modems werden die Audiosignale über eine Telefonleitung gesendet und von dem Modem am anderen Ende wieder in die Binärzahlen des Computers zurückverwandelt.

Cassetteninterfaces brauchen binäre Signale entweder nur in Audiosignale umzuwandeln (modulieren), um sie auf Band aufzu-



Telefax-Maschinen (Kurzform für Faksimile) werden in den Büros der Vereinigten Staaten und Europas zunehmend populärer. In Japan verwenden schon fast alle Firmen und viele Privathaushalte diese praktischen Geräte. Telefax-Maschinen können Dokumente, Bilder und Zeichnungen in Sekundenschnelle an andere Telefaxgeräte senden. Dabei wird nur das eingebaute Modem und die gewöhnliche Telefonleitung benötigt.

zeichnen, oder die Audiosignale des Cassettenbandes wieder in binäre Impulse zurückzuwandeln (demodulieren). Modems sind jedoch für die simultane Kommunikation in beide Richtungen gedacht und benötigen dazu zwei Frequenzbänder und vier individuelle Frequenzen. Ein weit verbreiteter Standard verwendet zum Senden die Frequenz 1070 Hz für 0 und 1270 Hz für die 1 und für den Empfang die Frequenz 2025 Hz für 0 und 2225 Hz für 1. Die Kennzeichnungen liegen damit sehr dicht beieinander.

Ein Cassetteninterface dagegen verwendet für die Kennzeichnung einer 1 etwa die doppelte Frequenz wie für die 0. Die Dekodierung der dicht nebeneinanderliegenden Signale eines Modems benötigt deshalb eine vergleichsweise komplexe und teure Elektronik.



Kein Anschluß ohne FTZ!

Beim Einstieg in die Welt des Datenverkehrs müssen die Vorschriften der Post und der richtige Systemaufbau beachtet werden.

Da es in der Bundesrepublik verboten ist, an Anlagen der Post selbst Veränderungen vorzunehmen, muß ein Modem akustisch gekoppelt werden. Mittels Mikrophon und Lautsprecher im Gehäuse wird der Telefontörer mit dem Modem verbunden. Für den Datenverkehr sind somit zwei Modems erforderlich, die aufeinander abgestimmt sein müssen und sowohl auf Senden als auch auf Empfang gestellt werden können. Befindet sich das eine Modem auf Senden, dann „steht“ das andere auf Empfang. Auf diese Weise ist ein Zweiwegeverkehr möglich, aber die Übertragung kann nur in einer Richtung zur Zeit erfolgen. Diese Betriebsart wird als Halbduplex bezeichnet, im Gegensatz zu einem Vollduplex-Verkehr, der für simultanes Senden und Empfangen zwei verschiedene Frequenz-Bänder verwendet.

Das Modem selbst kommuniziert mit dem Computer über eine RS-232/V24-Schnittstelle und übersetzt die binären Zahlen in die benötigten Frequenzen. Bevor dies aber möglich ist, muß eine halbduplexfähige serielle Schnittstelle überhaupt vorhanden sein, die auf die erforderliche Übertragungsgeschwindigkeit einstellbar ist. Von der Post wird bislang nur die recht langsame Geschwindigkeit von 300 Baud genehmigt, und dies auch nur für Modems mit FTZ-Zulassung. Liegen beide Vor-

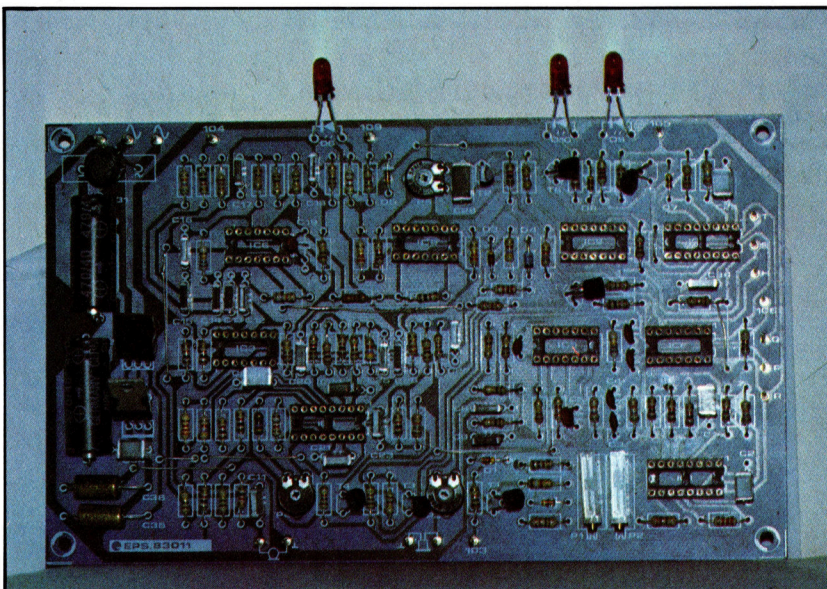
aussetzungen vor, ist die Tür zur Fernübertragung aber noch nicht geöffnet: Die Anschlußbelegung der Stecker ist keineswegs genormt, und so kann es passieren, daß nach dem Anschluß zunächst überhaupt nichts passiert. Und dann bleibt die Frage, wie die zu übertragenden Programme und Daten überhaupt an die Schnittstelle gelangen. Unbedingt notwendig ist hierfür eine dem Betriebssystem angepaßte Übertragungssoftware.

Direkt vom Hersteller

Viele Portables geben Modemfähigkeit vor, doch die Übertragungssoftware ist manchmal nur schwer zu bekommen. Für Heimcomputer gilt in Einzelfällen das Gleiche, obwohl in den Computer-Clubs schon viele Lösungen entwickelt wurden. Ein sinnvoller Weg zu der Datenfernübertragung liegt also in einem ausgesteteten System, das man möglichst direkt vom Hersteller bezieht. Dies ist bei einigen Handheld- und tragbaren Computern kein Problem, bei anderen Rechnern muß mit verschiedenen Fabrikaten gearbeitet werden. Es empfiehlt sich, nicht bei einem Computerversand zu bestellen, sondern einem örtlichen Händler die gesamte Systemzusammenstellung zu übertragen, mit einer Garantie für die Funktionsfähigkeit.

Sind die technischen Voraussetzungen für die Datenfernübertragung geschaffen, bleibt die Vereinbarung einer gemeinsamen Übertragungsart: Neben den reinen Datenbits gibt es nämlich noch Startbits, Stop- und Prüfbits. Um dem Empfänger mitzuteilen, wo im Strom der übertragenen Bits ein Byte beginnt, sendet man als erstes Bit auf jeden Fall eine Null, bezeichnenderweise Startbit genannt. Das Ende eines Wortes wird entsprechend durch eine oder zwei Einsen gekennzeichnet – die Stopbits. Sowie noch ein weiteres Bit an das Datenwort angehängt wird, spricht man vom Parity-Bit, das eine einfache Prüfung der übertragenen Daten ermöglicht. Dazu dient ein Vergleich mit der Summe der Einsen des Datenwortes, die entweder eine gerade oder ungerade Zahl ergibt. Die Parität wird dementsprechend als gerade (even) oder ungerade (odd) vereinbart, der Empfänger kann so feststellen, ob ein Wort falsch übermittelt wurde.

Diese Platine eines Akustikkopplers wurde von einem Hobby-Programmierer nach einem Schaltplan bestückt. Selbstgebaut kostet das Gerät etwa 140 Mark.

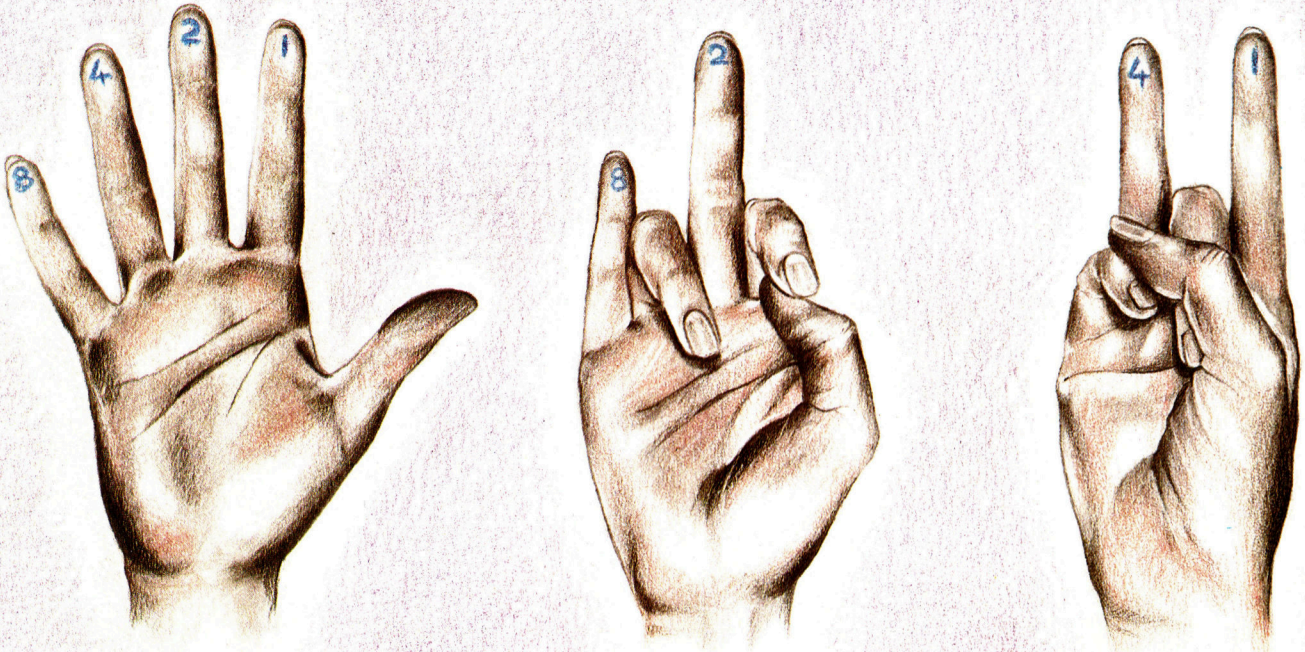




Wenn $1+1=10$. . .

Mit den Binärziffern 0 und 1 kann der Computer jede Dezimalzahl darstellen. So entspricht zum Beispiel 11111010 der Zahl 506.

Die Umwandlung ins Binäre



Ein einfacher Weg, vierstellige binäre Zahlen in Dezimalzahlen zu verwandeln, ist dieser: Auf die Finger der rechten Hand schreibt man (z. B. per Aufkleber) die Stellenwerte der Binärzahlen (8, 4, 2, 1). Finger, die binär Eins sind, streckt man aus, Finger, die binär Null sind, bleiben gekrümmt. Wenn wir dies für binär 1010 ausführen, erhalten wir 8 und 2, was zusammen dezimal 10 ergibt. Für binär 0101 erhalten wir auf gleichem Wege dezimal 5.

Versuchen Sie es einmal selbst. Welche Dezimalzahlen erhalten Sie für binär 1110 und 0110? Wir können diese Methode bis auf achtstellige Binärzahlen ausdehnen, wenn wir die Finger der linken Hand mitverwenden. Sie repräsentieren dann die Stellenwerte 16, 32, 64 und 128, wobei wir dem kleinen Finger 16 und dem Zeigefinger 128 zuordnen. Wie stellen Sie mit Hilfe aller Finger die Zahl 180 dar?

Elaine Keenan

Für viele Menschen sind Dezimalzahlen etwas so Selbstverständliches, daß sie sich ein anderes Zahlensystem gar nicht vorstellen können. Dabei erdachten sich die Römer ein System, in dem Zahlen durch Buchstaben ausgedrückt werden. So stellte X die Zahl 10, L die Zahl 50, C die Zahl 100, D die Zahl 500 dar, um nur einige zu nennen. Für die Aufzeichnung einfacher Zahlen funktionierte dieses System ganz gut, zum Rechnen aber taugte es nicht viel. Selbst einfache Additionen fallen mit römischen Zahlen schwer, weil eine Zuordnung von Stelle und Wert fehlt.

Sieht man sich aber die beiden Dezimalzahlen 506 und 56 an, so fällt als erster Unterschied die Null in 506 auf. Sie sagt uns, daß 506, neben fünf Hundertern und sechs Einern,

keine Zehner hat. In unserem Zahlensystem hat jede Ziffer ihren Stellenwert. In der ersten Stelle von rechts stehen die Einer, in der nächsten nach links stehen die Zehner, dann die Hunderter, die Tausender und so weiter. Die Ziffer gibt also nur an, wieviele „Werte“ in einer Stelle stehen.

Was aber hat dies alles mit Computern und dem binären Zahlensystem zu tun? Nun, Computer sind elektronische Maschinen, die mit Zahlen gut umgehen können, sobald sie durch Spannungspegel repräsentiert werden. So kann 5 Volt binär Eins und 0 Volt binär Null darstellen. Sie wissen, daß Einsen und Nullen im Binärsystem völlig ausreichen, um jede beliebige Zahl, egal wie groß, darzustellen.

Allerdings, so kompakt wie unser Dezimal-



oder Zehnersystem ist das binäre System nicht. So ist die Dezimalzahl 506 in binärer Form 11111010 für uns wesentlich unhandlicher. Das liegt daran, daß im Dezimalsystem die Stellenwerte Potenzen von 10 und im Binärsystem Potenzen von 2 sind. Hinzu kommt, daß im Dezimalsystem pro Stellenwert zehn Ziffern, von 0 bis 9, zur Verfügung stehen, im Binärsystem dagegen pro Stellenwert nur zwei, nämlich 0 und 1. Der geringere Ziffernvorrat führt also im Binärsystem zu mehrstelligeren Zahlen. Benötigt der Wert Zehn dezimal zwei Stellen, so sind es im Binärsystem bereits vier: 1010.

Obwohl zwischen beiden Zahlensystemen grundsätzliche Ähnlichkeiten bestehen (beide sind Stellenwert-Systeme) und der kleinste ganzzahlige Wert bei beiden Eins ist, sind die Ergebnisse in Ziffern doch sehr unterschiedlich. Wir wollen dies am Beispiel der Dezimalzahl 506 weiter verdeutlichen:

Im Dezimalsystem:

Hunderter	Zehner	Einer	
5	0	6	
= 5×100 + 0×10 + 6×1			= 506

Im Binärsystem:

256er	128er	64er	32er	16er	8er	4er	2er	1er	
1	1	1	1	1	1	0	1	0	
= 1×256+1×128+1×64+1×32+1×16									
+1×8+0×4+1×2+0×1									= 506 dezimal

Die uns vom Dezimalsystem her vertrauten Rechenregeln gelten genauso im Binärsystem.

Wir müssen nur beachten, daß hier die größte Ziffer 1 und nicht, wie gewohnt, 9 ist. Zum Beweis einige Additionen. (Zur Unterscheidung stehen Dezimalzahlen in Klammern):

$$\begin{array}{r} (3) \quad 11 \\ + (5) \quad +101 \\ \hline (8) \quad 1000 \end{array}$$

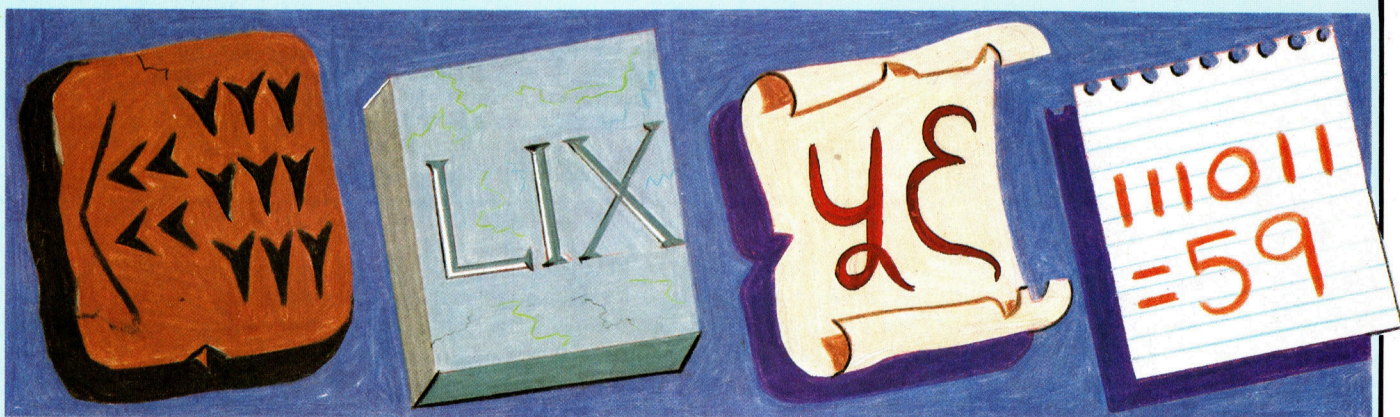
1+1=0 und Übertrag 1
 Übertrag 1+0+1 = 0 und Übertrag 1
 Übertrag 1+1 = 0 und Übertrag 1
 Übertrag 1+0 = 1

Wir haben gerade gesehen, daß binär Eins und Eins gleich Null mit Übertrag Eins ergibt (dies ist vergleichbar zu 9 und 1 gleich 10, „Null hin Übertrag Eins im Sinn“). Im Binärsystem ist Eins die größte Ziffer, vergleichbar zu Neun im Dezimalsystem. Hier nun die nächste Addition und zwei weitere Beispiele zum Ausprobieren:

(4) 100	(7) 111	(3) 11	
+ (6) +110	+ (2) + 10	+ (12) +1100	
(10) 1010	(?) ?	(?) ?	(?) ?

Spätestens jetzt haben Sie feststellen können, daß binäre Zahlen viel länger als ihre gleichwertigen dezimalen Partner sind. Versuchen Sie doch einmal, binär 11010110 und 1101101 zu addieren. Vergessen Sie aber nicht, gleiche Stellenwerte untereinander zu schreiben, genau wie beim Rechnen mit Dezimalzahlen.

Die Geschichte der Zahlensysteme



Babylonisch

Die alten Babylonier entwickelten vor mehr als 4000 Jahren ein Zahlensystem, das, im Gegensatz zu unserem heutigen Dezimalsystem, die Zahl 60 als Basis verwendete. Die Illustration zeigt die Zahl 59 in babylonischer Keilschrift. Das römische Zahlensystem war im Vergleich zum Babylonischen ein Rückschritt. Buchstaben repräsentierten Zahlen. Weil aber die Stellung der Buchstaben-Zahl keinen Hinweis auf ihren Wert gab, waren selbst einfache Rechenoperationen fast nicht ausführbar. Die Hindus verwendeten neun Zeichen für Zahlen von 1 bis 9 und fügten später ein Zeichen für 0 hinzu. Ihr Beitrag zur Entwicklung

Römisch

Hindustanisch

der Zahlensysteme war die Einführung des Stellenwertes: Der Wert eines Zeichens (Ziffer) richtete sich nach seiner Stellung. Das System wurde von den Arabern übernommen und breitete sich von dort nach Europa aus. Einer der führenden arabischen Mathematiker war Al-Huwarizmy. Aus der latinisierten Form seines Namens entstand der Begriff Algorithmus und aus der Abkürzung des Titels seines Lehrbuches „Al-Gabr Wal-Muqabala“ entstand der Begriff Algebra. Die Vorliebe der Computer für das Binärsystem hat ihren Grund darin, daß sich jede beliebige Zahl mit Einsen und Nullen darstellen läßt.

Binär

Fachwörter auf einen Blick

A/D Wandler

Analog-Digital-Wandler; Gerät, das eine analoge Eingabe in digitale Werte umwandelt

Akustikkoppler

ermöglichen die Datenübertragung über weite Entfernungen.

Akustikkoppler sind passende Telefon-Sprechmuscheln, die an den Computer angeschlossen werden können

Directory

Verzeichnis, das angibt, an welcher Stelle sich abgelegte Programme oder Daten auf Diskette oder Cassette befinden

Duplex

Übertragungsart, bei der Daten gleichzeitig in beide Richtungen übertragen werden können; hierzu wird ein Übertragungssystem mit zwei Leitern verwendet

Elektron

Leicht negativ geladenes, stabiles Elementarteilchen

Flußdiagramm

Grafische Darstellung eines Programmablaufs; wird zur Erstellung von Rechnerprogrammen benutzt

FTZ-Nummer

FTZ ist die Abkürzung für „Fernmelde-Technische Zulassung“, die für alle Geräte erforderlich ist, die in die Posthoheit eingreifen; ein Modem oder ein Akustikkoppler bedient sich des Telefonnetzes und muß daher von der Post mit einer FTZ-Nummer zugelassen sein

Halbleiter

Kristalline Stoffe, die sich bei niedrigen Temperaturen wie Isolatoren verhalten; ihre Leitfähigkeit nimmt bei steigender Wärme zu

LED

(Light Emitting Diode) Leuchtdiode, die als Anzeigeelement dient; LEDs existieren in mehreren Farben

Mailbox

Elektronischer „Briefkasten“ in einem größeren Computer, der über Telefon oder Datex-Leitungen mit Hilfe eines Modems angewählt werden kann; in der Mailbox können Nachrichten für andere abgelegt oder aber Informationen abgefragt werden

Microcassetten

Für Diktiergeräte und Handheld-Computer verwendete Mini-Cassetten

Platine

Leiterplatte, auf der verschiedene Bausteine befestigt werden; besteht in der Regel aus Kunststoff

Raster

Senkrechte und waagerechte Einteilung des Bildschirms; Monitorbilder sind gerastert

Silizium

Chemisches Element, das für die Halbleiterelektronik sehr wichtig ist; es wird in einem aufwendigen Fertigungsprozeß aus Quarzsand gewonnen

Sonderzeichen

Zeichen und Symbole, die zusätzlich im Zeichensatz vorhanden sind

Start-Bit

In der Datenfernübertragung werden die seriell übertragenen Bytes mit einem Start- sowie mit einem Stopbit versehen, damit der empfangende Rechner erkennen kann, wann ein Byte beginnt und wann es endet; zusätzlich werden oft noch Prüfbits angehängt, um die Übertragung zu überprüfen

Telefax

Telekopierer, der über die normale Telefonleitung Texte und Grafiken übermitteln kann; benötigt wird dazu ein druckerähnliches Gerät



David Higham

Überschreiben

Eingabe von Daten in den Speicher, wobei die bereits abgespeicherten Daten gelöscht und zerstört werden

User

Anwender eines Rechners werden in der Computersprache User genannt (englisch „to use“)

Vektor

Höherstehende Zahl, die neben dem Zahlenwert (Größe) auch durch eine Richtung gekennzeichnet ist

Vollduplex

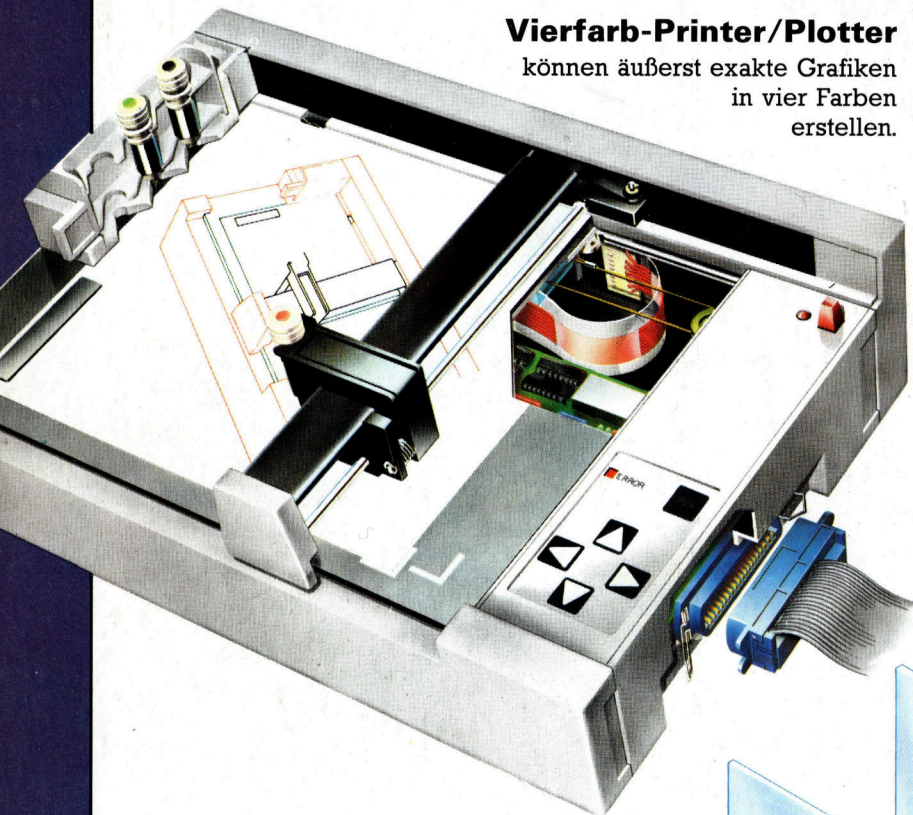
Art der Datenübertragung, bei der Leiter für alle Übertragungsrichtungen vorhanden sind

Wafer

Siliziumscheibe, auf der integrierte Schaltkreise aufgetragen sind; wenn integrierte Schaltkreise montiert sind, werden die einzelnen Plättchen in ein Gehäuse eingebaut; der Microprozessor eines Computers ist ein Wafer

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

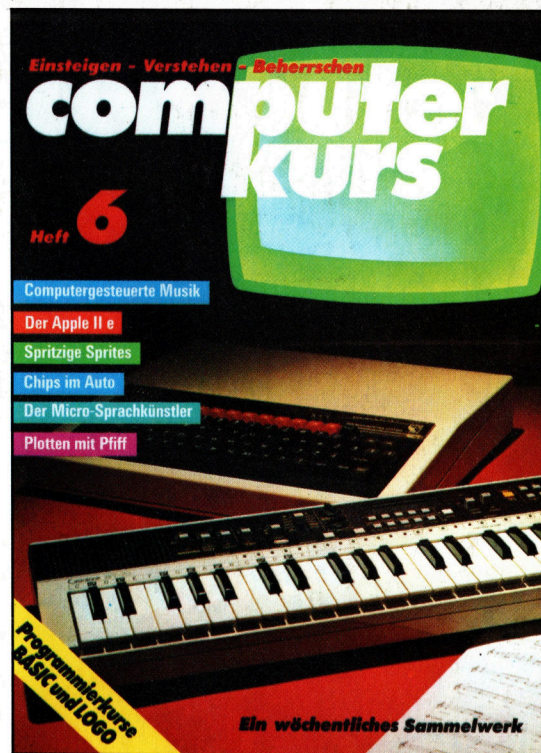
computer kurs Heft 6



Vierfarb-Printer/Plotter
können äußerst exakte Grafiken
in vier Farben
erstellen.

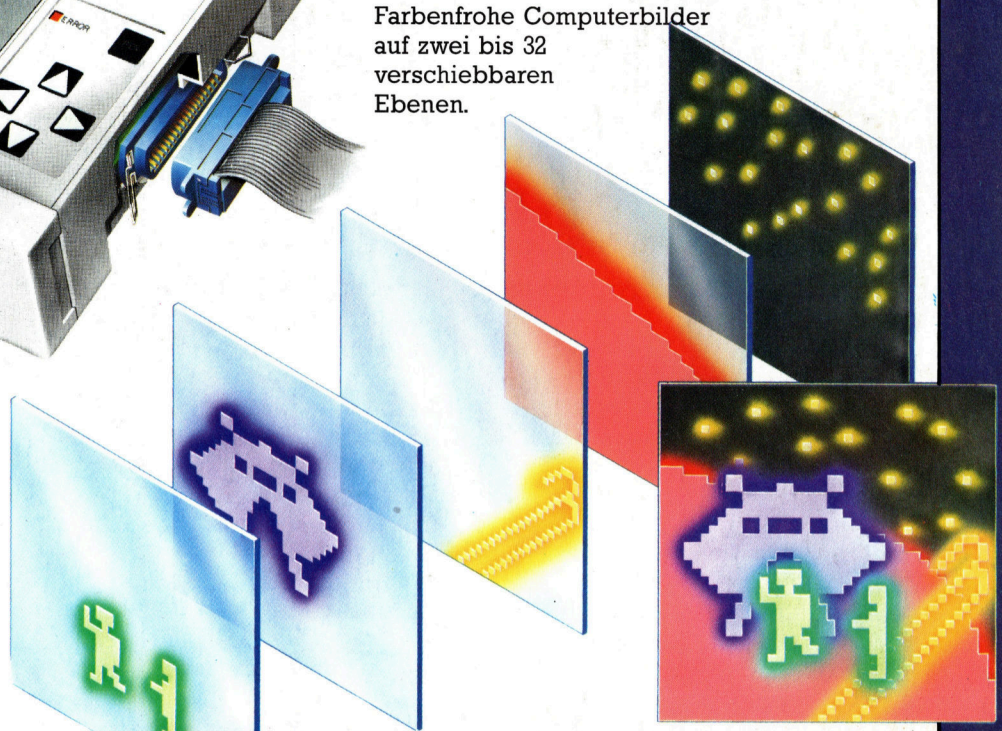
Klangzauberer

Auf der Musikszene längst etabliert, sind Synthesizer nun auch auf dem Heimcomputer-Markt ganz groß im Kommen. Mit einem Keyboard sind Sie Komponist, Musiker aller Instrumente und Dirigent in einer Person. Aber auch Ihre Programmierkenntnisse sind gefragt.



Sprites

Farbenfrohe Computerbilder
auf zwei bis 32
verschiebbaren
Ebenen.



+++ Chips im Auto +++ Der Computer als
Sprachkünstler +++ Wie Daten richtig ge-
speichert werden +++ Stephen Wozniak und
der Apple II e +++ Tips für die Praxis +++