

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Heft **25**

Portables von Kyocera

Bildplatten als Speicher

Umgang mit Maschinensprache

Tips: Selbstbau-Gatter

Dateien-Zugriff

Ein wöchentliches Sammelwerk

computer kurs

Heft 25

Inhalt

Computer Welt

- Laservisionen** 673
Massenspeicher mit hoher Datendichte
- Verkabelte Klänge** 695
MIDI-Hard- und Software
- Tops auf Band** 700
Virgin Games erstellt Spielprogramme

Software

- Daten-Ablage** 676
Zugriffsarten verschiedener Dateien
- Gut zuhören!** 684
Sprache und Geräusche für eigene Computerspiele

Hardware

- Nachbauten** 678
Portables von Kyocera

Bits und Bytes

- Register und Speicher** 681
Umgang mit dem Maschinencode

Peripherie

- Camera Obscura** 685
EV1 zum Aufzeichnen von beliebigen Motiven
- Bewegter Arm** 698
Ein preiswerter Roboter für Heimcomputer

BASIC 25

- Das Spectrum des ZX-BASIC** 688

Tips für die Praxis

- Transistor-Logik** 690
Selbstgebaute AND- und NOT-Gatter

LOGO 25

- Neue Strukturen** 692
Kontrollen werden integriert

Fachwörter von A—Z

WIE SIE JEDE WOCHE IHR HEFT BEKOMMEN

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

Zurückliegende Hefte

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

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

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

Schweiz: Das einzelne Heft kostet sfr 3,80. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

Abonnement

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

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

SAMMELORDNER

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

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

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

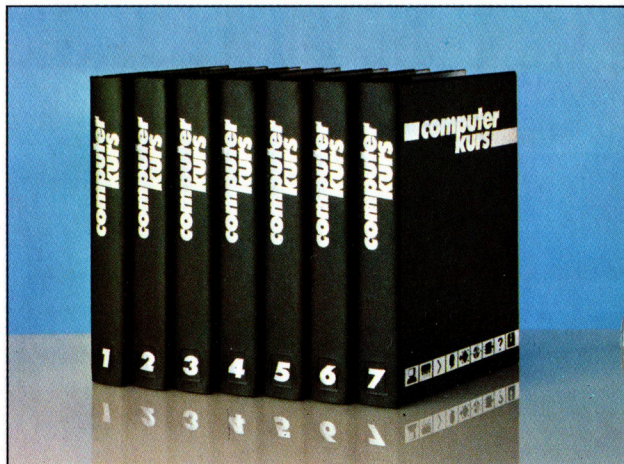
Schweiz: Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

INHALTSVERZEICHNIS

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

Redaktion: Winfried Schmidt (verantw. f. d. Inhalt), Elke Leibinger, Susanne Brandt, Uta Brandt (Layout), Sammelwerk RedaktionsService GmbH, Paulstraße 3, 2000 Hamburg 1

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





Laservisionen

Videoplatten und Laser-Discs sind Massenspeicher von hoher Datendichte, zu denen man direkten Zugriff hat. Die gesunkenen Produktionskosten und der industrielle Wettbewerb haben einen kontinuierlich fallenden Preis zur Folge.

Viele Leser werden überrascht sein, wenn sie erfahren, daß Laser-Disc-Spieler, die vor kurzem noch als Luxusgüter galten, jetzt weniger kosten als Videorecorder, obwohl sie weitaus bessere Bilder liefern. Ein Fernsehbild ist ein dynamisches, fortlaufendes Bild, das auf einem Videoband als ununterbrochene Sequenz festgehalten wird. Die einzige Möglichkeit, einen bestimmten Teil dieser Sequenz zu finden, besteht darin, das Band entsprechend weit vorlaufen zu lassen, – sei das nun, indem man es „normal“ abspielt oder es mit Hilfe des Bandzählwerkes bis zu einer bestimmten Zahl „schnell“ vorlaufen läßt. Auf Disketten bzw. Discs werden Bilder als einzelne Felder gespeichert. So ist ein direkter, genauer und schneller Zugriff zu jedem einzelnen Bild möglich. Die Lage eines Feldes auf einer Diskette

kann durch Spur und Sektor genau beschrieben werden. Ein Microprozessor verwaltet die genauen Positionen. Er steuert den Zugriff zu den Feldern und bietet die Möglichkeit von Standbild oder Zeitlupe und Stereoton.

Wer je bewußt die hochauflösende Grafik seines Microcomputers eingesetzt hat, weiß, daß ein Fernsehbild aus vielen einzelnen Lichtpunkten besteht. – Je mehr Punkte auf dem Bildschirm dargestellt werden, desto besser ist das Bild. Für die Speicherung dieser hochauflösenden Grafik ist viel Speicherplatz erforderlich. So verfügt beispielsweise der Acorn B über eine maximale Auflösung von 1280x1224 Punkten pro Bild, womit genau 20 KByte Speicher benötigt werden. Hätte ein Fernsehbild auch keine bessere Auflösung, würde das Speichern einer Video-Sekunde

Laser-Disc-Spieler werden bald ebenso preiswert wie Heimcomputer sein, und man wird sich an die Verbindung der beiden miteinander gewöhnen. Durch entsprechende Discs und Software hat man Zugang zu Bild-Datenbanken und ausgefeilten Ausbildungsprogrammen und kann perfekte Abenteuerspiele „durchleben“.



Neue Speicher

35 cm-Laser-Discs und die kleineren Compact-Discs erlauben die Speicherung von Video- und Audio-Informationen sowie digitaler Daten. Nachteil ist, daß die Discs nicht beschrieben werden können und technisch als ROMs zu betrachten sind. Eigene Programme oder Informationen können also vorerst nicht darauf gespeichert werden.

(mit 25 Bildern pro Sekunde) 25x20 KByte Speicherplatz erfordern. Eine aufgezeichnete Fernsehminute beanspruchte bereits 30 Mega-Byte (30 000 000 Byte), und die Folge einer Fernsehserie beanspruchte gar mehr als ein Giga-Byte (1000 MegaByte) Speicher. Über 6000 Disketten mit single density wären dafür erforderlich. Die Aufzeichnungszeit betrüge über eine Woche.

Berücksichtigt man diese Zahlen, scheint die Aufzeichnung auf Videoband schon attraktiver. Denn das Speichern einer halben Fernsehstunde auf Diskette stellt sich als unlösbares Problem dar.

Doch es ist zu bewältigen, indem man den Platz für die beschreibenden Daten klein hält. Bei einem Laser-Disc-Recorder werden die Daten mittels eines Laserstrahls auf eine metallische Platte geätzt, die mit einer unempfindlichen Schutzschicht überzogen ist. Mittels eines schwachen Laserstrahls werden diese Daten von der Disc gelesen. Man benutzt Laser zum Schreiben/Lesen dieser Discs, da sie eine hohe Auflösung bei niedriger Toleranz bieten. Mit keiner anderen Technik ist das Lesen und Schreiben von Daten auf so geringem Platz möglich.

Das Speicherformat ist eine Kombination der Techniken, die bei Schallplatten und Disketten

Verwendung finden. Die Spuren einer herkömmlichen Schallplatte bilden eine Spirale, wobei die Spuren eine geätzte Darstellung der Wellenformen der aufgezeichneten Tonsignale sind. Die Spuren auf einer Diskette sind konzentrische Kreise, die Information wird magnetisch auf die Diskette geschrieben und anschließend digital als Muster aus Einsen und Nullen festgehalten. Bei der Laser-Disc verwendet man Spuren in Spiralforn, die jedoch nicht als Rillen erkennbar sind. Die Information wird optisch als Einsen/Nullen-Muster auf die Disc geätzt, kann aber nicht gelöscht werden. Die Einsen und Nullen werden auf die Platte gebracht, indem der Laser zur Darstellung der Einsen winzige Löcher in die Metallbeschichtung brennt. Bei der Darstellung von Nullen bleibt die Schicht unberührt. Ein solches Loch ist einen halben Micrometer (0,0005 mm) breit und einen Zehntelmicrometer (0,0001 mm) tief. Auf einem Quadratcentimeter lassen sich somit 400 000 000 dieser Löcher festhalten.

Diese erstaunliche Miniaturisierung erfüllt die Voraussetzungen für die Speicherung von Videofilmen völlig. Auf der Seite einer Disc von 35 Zentimetern Durchmesser befinden sich 54 000 Einzelbilder. Das entspricht in etwa einer Spielzeit von 36 Minuten. Die Berechnung der Feldgröße, die an anderer Stelle dieses Beitrages erfolgte, basierte auf der Schwarzweiß-Auflösung bei grafischer Computerdarstellung. Dagegen muß auf der Laser-Disc für jeden Bildschirmpunkt eine Farbaufzeichnung enthalten sein und ferner eine Ton-„Spur“. Ein Farbfeld mit entsprechender Tonspur erforderte 100 KByte Speicher. Für 54 000 solcher Felder wären rund 5 400 000 KByte oder 5,4 GigaByte erforderlich.

Ist das Problem der Speicherplatzbegrenzung gelöst, liegen die Vorteile der Laser-Disc auf der Hand. Ein Grundproblem der Datenverarbeitung, nämlich Datenaufzeichnung und -zugang, entfällt. Normalerweise stehen zwar fertige Informationen zur Verfügung, doch zuvor muß jemand all diese Daten in codierter Form eingeben. Das ist zeitraubend – und teuer. Erfäßt man statt dessen die Daten mit einer Kamera, wird der Aufwand erheblich reduziert.

Die Ausstattung der bisher angebotenen Laser-Disc-Spieler ist unterschiedlich. Preiswerte Geräte gibt es bereits. Sie finden Verwendung zum Abspielen von Filmen wie sonst mit dem Videorecorder, haben aber den entscheidenden Vorteil der besseren Bilddarstellung. Durch Eingabe bestimmter Feldnummern über die Tastatur ist die Ansteuerung der einzelnen Felder möglich. Dabei wird der Bildschirm für einen Augenblick leer, danach erscheint das gewünschte Bild.

Die echten Möglichkeiten werden aber erst erkennbar, wenn man bestimmte Felder mit Hilfe von Computerprogrammen wählen kann. Firmen wie Pioneer und Philips bieten solche



Laser-Disc-Spieler an. Sie sind noch sehr teuer und für die professionelle Anwendung gedacht. Beim einfachsten System verwendet man ein IEEE- oder RS232-Interface, um mittels Computer ein bestimmtes Feld nach der „Adresse“ zu wählen. Mit entsprechender Software lassen sich einzelne Felder erfassen.

Philips hat diese Idee weiterentwickelt und einen einfachen Microcomputer in seine neueren Modelle integriert. Somit kann ein Programm entweder von einem EPROM-Steckmodul oder von einer Laser-Disc „geladen“ werden. Jede Laser-Disc enthält zwei Audiospuren und eine Videospur. So können beispielsweise auf einer Disc zwei Sprachen enthalten sein. Wird die zweite Audiospur nicht benötigt, läßt sich darauf statt dessen ein Computerprogramm speichern.

Mischen von Bild und Text

Computergesteuert stehen also 54 000 hochwertige Bilder zur Verfügung. Letztes Stadium wäre folglich, die Bilder der Laser-Disc mit Text zu kombinieren. Das kann über zwei Monitore erfolgen oder durch Mischen über den Video-Eingang oder – Möglichkeit drei – durch Verwendung eines Monitors mit eigenem Teletext-Generator. Damit wäre als neues Medium das interaktive Video geschaffen. Anwender und Software steuern ergänzend die Bilddarstellung, Handlungen und Standbilder durch einfaches Lesen der Discs.

Eine der interessantesten Anwendungen wäre eine optische Datenbank. Der Anwender

würde eine Frage an den Computer stellen, eine entsprechende Information bekommen und über den Laser-Disc-Spieler das adäquate Videofeld erreichen. Das ließe sich in Bibliotheken und Schulen hervorragend einsetzen, ob man nun nach verschiedenen Blumen sucht oder Artikel in einem Katalog finden will.

Interaktives Video würde den Anwender in das Bildschirmgeschehen noch weiter einbeziehen. Ein Ausbildungsprogramm wäre ideal, da sich einzelne Sequenzen stets wiederholen lassen. Man könnte sogar Filme produzieren, in denen der Anwender selbst Entscheidungen treffen und so den Ablauf beliebig beeinflussen kann. Für Freunde von „Abenteuerspielen“ auf dem Computer böten sich faszinierende neue Möglichkeiten.

Doch im Augenblick stehen dem noch einige Marktprobleme im Wege. Neben den Kosten für die Ausstattung und die Herstellung von Laser-Discs müssen neue Wege und Möglichkeiten sowohl der Gestaltung als auch der Produktion solcher Discs gefunden werden. Seien das nun die speziellen Anforderungen der Computer-Software oder das Beschreiben und Abbilden von optischen Bildern. Kleinere Unternehmen setzen sich mit diesen Anforderungen bereits auseinander. Eine Gesellschaft namens Computer Assisted Televideo (CAT) ist derzeit Marktführer. Sie bietet alles, von der Hardware bis zur Entwicklung und Produktion geeigneter Software. Allerdings ist der Markt von CAT aus Kostengründen auf die Schaffung von Ausbildungsprogrammen für den Geschäftsbereich „Großfirmen“ reduziert.

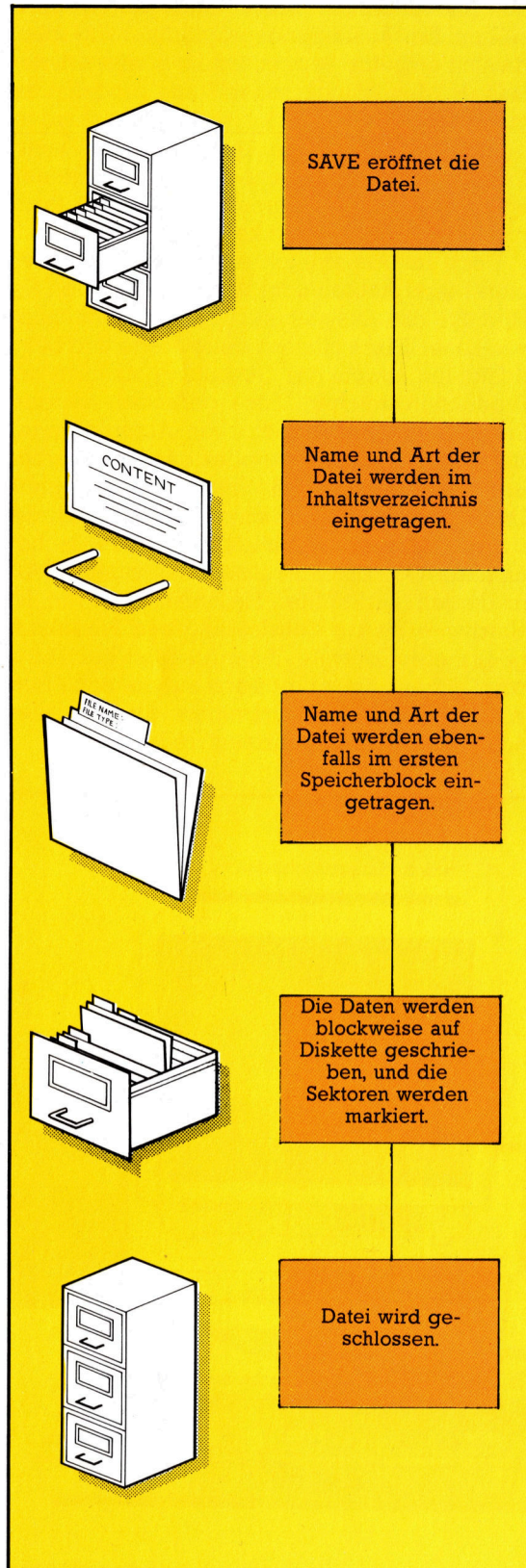
Viele Laser-Disc-Spieler sind mit einer IEEE- oder RS232-Schnittstelle ausgestattet, wodurch eine Steuerung über Microcomputer möglich ist. Eine typische Konfiguration könnte so aussehen, daß ein Computer eine auf Laser-Disc gespeicherte optische Datenbank kontrolliert. Der Spieler wird über den Computer durch Angabe einer Feldzahl angesteuert und zeigt dann das gewünschte Bild. Die Ausgabe des Computers und die Laser-Disc-Bilder können auf dem Bildschirm kombiniert werden.



Daten-Ablage

Hier beginnt eine neue Serie, in der wir genauer auf die Methoden eingehen, mit denen Daten auf Disketten untergebracht werden.

Binäre Dateien sind direkte Kopien des Arbeitsspeichers. Sie können Programme, Bildschirmhalte etc. enthalten. Der Speichervorgang ist einfach: Nachdem im Inhaltsverzeichnis der Diskette der Dateiname eingetragen wurde, werden die Daten als eine Folge von miteinander verbundenen Blöcken auf dem Speichermedium untergebracht. Das DOS unterhält ebenfalls eine Liste, in der die Belegung der Speicherblöcke verzeichnet ist. Damit wird verhindert, daß neue Dateien bereits belegte Blöcke überschreiben.



Das Wort Datei wird deshalb so oft für Speichervorgänge der EDV eingesetzt, weil es direkte Vergleiche mit den Ablagesystemen von Dokumenten in Karteischränken zuläßt. Wir werden daher zunächst erklären, welche Aufgaben Dateisysteme auf Heimcomputern überhaupt ausführen können.

Für den Ablauf unseres täglichen Lebens ist es notwendig, wichtige Informationen, wie z. B. Kontostände oder Termine, so geordnet wie möglich aufzuzeichnen und aufzubewahren. Diese Informationsmenge läßt sich mit Terminkalendern, Adreßbüchern und Notizblocks leicht bewältigen. Bei großen Datenmengen muß die Speicherung und besonders die schnelle Verfügbarkeit der Informationen jedoch genauer geplant werden, besonders, wenn Datenmengen sich ständig verändern und Auskunft über eine Vielzahl von Menschen, Projekten, Orten und Umständen gegeben werden soll. Je größer ein Unternehmen wird, desto komplexer wird die Verwaltung von Informationen. Viele Fehler in schnell wachsenden Betrieben lassen sich auf ungenügende Informationsstrukturen oder auf Falschinterpretation von Daten zurückführen. Jedes gute Managementsystem muß daher über Methoden verfügen, mit denen Daten einfach und effektiv gespeichert, indiziert und abgerufen werden können. Erfahrene Manager setzen zu diesem Zweck Dateisysteme ein, die sich nach Art, Menge und Änderungshäufigkeit der anfallenden Datenmengen strukturieren lassen.

Selbst wenn für die Informationsverwaltung Computersysteme eingesetzt werden, die mit unglaublicher Geschwindigkeit große Datenmengen speichern und verarbeiten können, bleiben die Grundlagen gleich. Das Zentrum eines solchen Systems bildet der „Manager“: das Betriebssystem – auch „DOS“ (Disk Operating System) genannt. Ein DOS ist bestens für die Verwaltung von Daten geeignet – wenn es von Ihrem Programm die richtigen Anweisungen erhält. Computersysteme können immer nur so effektiv sein wie die im Betriebssystem eingebaute Datenstruktur und die Art, wie das DOS eingesetzt wird.

Die Methoden, mit denen Microcomputer ihre Dateiverwaltung aufbauen, wurden ursprünglich für Geräte der Groß-EDV und der mittleren Datentechnik entwickelt. Es gibt dabei drei große Gruppen: binäre, sequentielle und Random-Access-(wahlfreie-)Dateien.



Das Inhaltsverzeichnis einer Diskette

Auf dem Bildschirm wird das Inhaltsverzeichnis einer Diskette über das CP/M Modul STAT angezeigt. Dieses Modul liefert ausführlichere Informationen als viele andere Anzeigeprogramme.

Recs
Eine Datei kann eine unterschiedliche Anzahl von Records haben; hier sind es 128 Bytes.

Bytes
Die Länge der Datei in KBytes.

Ext
Der Extent ist eine weitere Möglichkeit, die Länge des Speicherplatzes anzuzeigen.

Acc
Access: Eine Datei kann für Lesen und Schreiben R/W oder nur für Lesen R/O markiert sein.

Filename
Der volle Dateiname beginnt mit der Angabe der Laufwerkskennung (A: oder B:). Dahinter erscheint der eigentliche Dateiname (z. B. AUTOST), gefolgt von dem Dateikürzel (z. B. .COM). Das Kürzel gibt Aufschluß über den Dateiinhalt.

Recs	Bytes	Ext	Acc	Filename
0	0K	1	R/W	B:ACNTLIST.DTA
11	2K	1	R/W	B:ANT
10	2K	1	R/W	B:ANT.BAK
64	8K	1	R/W	B:ASM.COM
16	2K	1	R/O	B:CODELIST.DTA
16	2K	1	R/W	B:AUTOST.COM
1	1K	1	R/W	B:COMPANY.DTA
2	1K	1	R/W	B:CONTROL.DTA
34	5K	1	R/W	B:COPY.COM
40	5K	1	R/W	B:DDT.COM
4	1K	1	R/W	B:DUMP.COM
250	32K	2	R/W	B:INSTALL.COM
4	1K	1	R/W	B:JUNK
16	2K	1	R/W	B:LOAD.COM
6	1K	1	R/W	B:MICROLIN
40	5K	1	R/W	B:ML.COM
86	11K	1	R/W	B:MOVCPM.COM
58	8K	1	R/W	B:PIP.COM
2	1K	1	R/W	B:SCREEN.ASM
1	1K	1	R/W	B:SCREEN.COM
4	1K	1	R/W	B:SCREEN.DOC
42	6K	1	R/W	B:STAT.COM

Bytes Remaining On B: 85K

Binäre Dateien

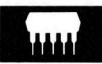
Binäre Dateien sind direkte Kopien des Arbeitsspeichers. Ein gutes Beispiel dafür sind Programme, die mit SAVE gespeichert wurden. Stellen Sie sich Ihren Arbeitsspeicher einfach als Notizblock vor. Seiten mit wichtigen Notizen oder interessanten Zeichnungen reißen Sie ab und heben Sie auf. Binäre Dateien arbeiten mit der gleichen Methode. Bei Eingabe des Befehls SAVE schreibt das DOS zunächst den Dateinamen auf die Diskette, markiert den Eintrag dann als binäre Datei und kopiert schließlich Byte für Byte den Inhalt des entsprechenden RAM-Bereiches. Die Daten (oder das Programm) werden dabei als eine Folge von miteinander verbundenen Blöcken gespeichert. Das Ende eines Blockes enthält die Information, wo der nächste zu finden ist, und im letzten Block befindet sich eine Endmarkierung. Bei einem Notizblock hätten wir entsprechend jeder Seite eine Überschrift gegeben, sie in einem Karteikasten abgelegt und die Überschrift in dessen Inhaltsverzeichnis eingetragen.

Eine BASIC-Zeile wird nach der Eingabe automatisch komprimiert und in ein Tokenformat

umgesetzt. Dabei ersetzt der BASIC-Interpreter einzelne Schlüsselworte durch Zahlen von einem Byte Länge, die das System leichter verarbeiten kann. Für eine Programmliste werden die Token wieder in das Textformat rückgewandelt. Da binäre Dateien direkte Kopien des Arbeitsspeichers sind, können sie auch ASCII-Codes und binäre Daten speichern. Weiterhin läßt sich damit der Inhalt des Bildschirms auf Diskette ablegen und später wieder zurückladen. Manche Betriebssysteme und BASIC-Interpreter bieten außerdem die Möglichkeit, BASIC-Programme im ASCII-Format ohne Token abzuspeichern. Die Programme können dann als Textdateien mit einem Texteditor bearbeitet werden.

Binäre Dateien sind zwar leicht zu handhaben, haben aber zwei Nachteile: Zunächst lassen sich Informationen nur als zusammenhängender Datenblock speichern, der auch nur vollständig in den Speicher zurückgeladen werden kann, und weiterhin ist die maximale Länge einer Datei auf die Größe des Arbeitsspeichers begrenzt.

In der nächsten Folge werden wir uns mit sequentiellen und Random-Access-Dateien beschäftigen.



Nachbauten

Wir untersuchen drei Geräte, die nicht größer als Telefonbücher („Lap-size“) sind – darunter der Tandy Modell 100.

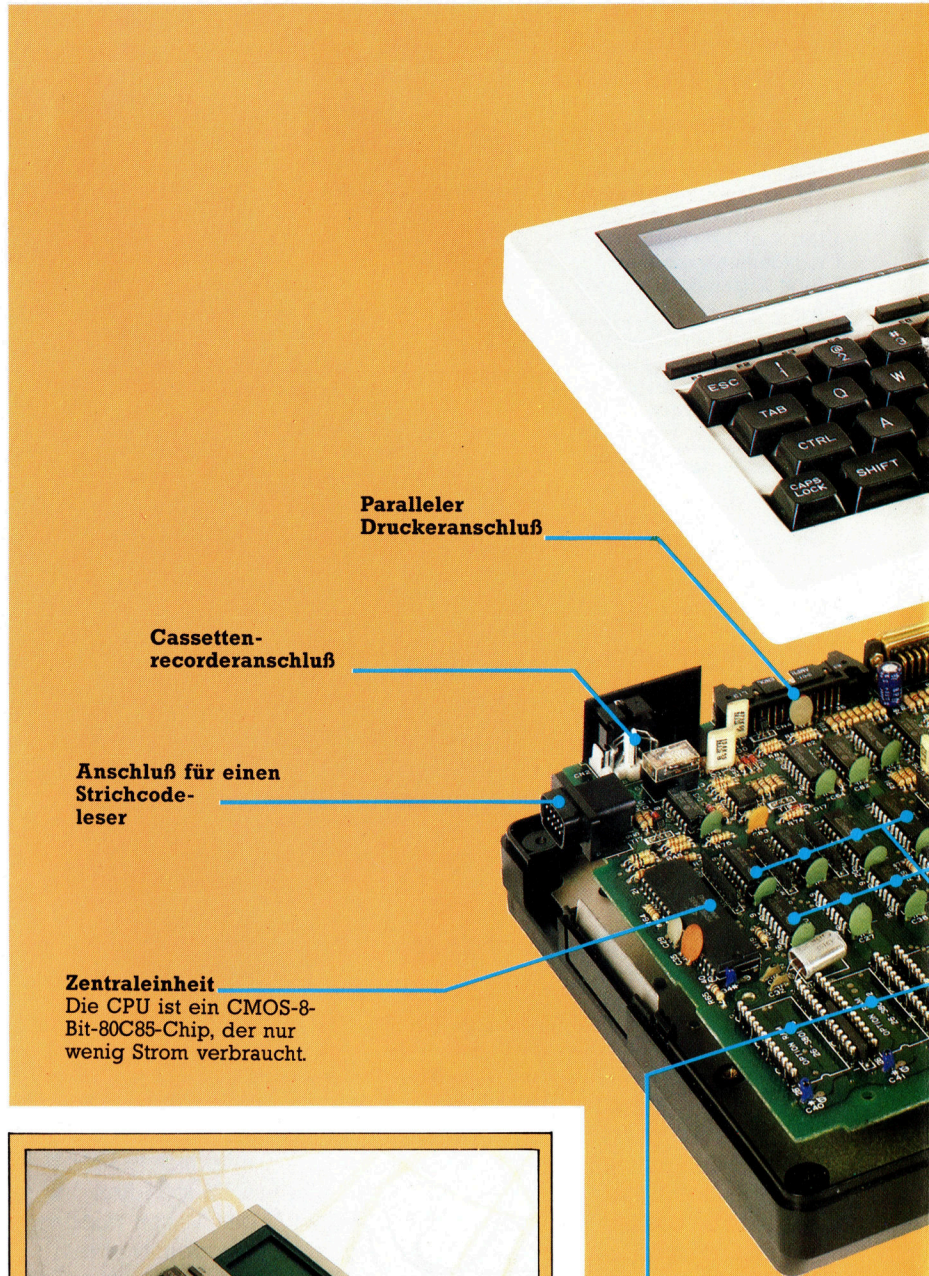
Nachbauten sind Produkte, bei denen ein Hersteller einen im Markt eingeführten Artikel kauft, ihm ein neues Aussehen gibt und als „eigene Schöpfung“ anbietet. Auch auf dem Heimcomputermarkt wenden mehr und mehr Firmen diese Methode an. So entstanden drei bekannte tragbare Computermodelle – der Tandy Modell 100, der NEC PC8201A und der Olivetti M10 – als Nachbauten desselben Modells. Alle drei Maschinen werden von der japanischen Firma Kyocera hergestellt, von Tandy, NEC und Olivetti geringfügig verändert und unter verschiedenen Namen angeboten.

Mit einem Gewicht von 1,8 kg gehören der Tandy, NEC und Olivetti in die Kategorie der kleinen tragbaren Maschinen (Lap-size). Das Modell 100 hat eine vollwertige QWERTY-Tastatur, im ROM integrierte Software und eine Flüssigkristallanzeige. Das Gerät ist im Batteriebetrieb voll funktionsfähig. Der RAM-Speicher verliert beim Abschalten seinen Inhalt nicht. Auch können Dateien in bestimmten RAM-Bereichen untergebracht werden, die sich wie eine Cassette oder Diskette ansprechen lassen. Es lassen sich jedoch auch Cassettenrecorder oder Diskettenlaufwerke als externe Speicher anschließen.

Zeichendarstellung

Die Flüssigkristallanzeige stellt acht Zeilen zu 40 Zeichen dar und kann Text und Grafik mischen. Das LCD besteht aus 15360 Bildpunkten, die sich einzeln ansprechen lassen. Die Zeichen sind aus einer Matrix von 6x8 Punkten zusammengesetzt und umfassen Groß- und Kleinbuchstaben. Das Modell 100 bietet einen vollen internationalen Zeichensatz und – im Gegensatz zu den drei Grafikzeichen der NEC-Maschine – einen Spezialzeichensatz für Grafik. Die LCD-Schirme der Modelle von NEC und Tandy sind flach in die Oberfläche der Geräte eingebaut, während Olivetti eine bewegliche Anzeige bietet, deren Winkel verstellbar ist. Auf dem Tandy und dem NEC läßt sich der Anzeigekontrast regulieren.

Die Tastatur des Tandy enthält Spezialtasten, über die die Grafikfähigkeiten angesprochen werden können und sich ein Teil der Ta-



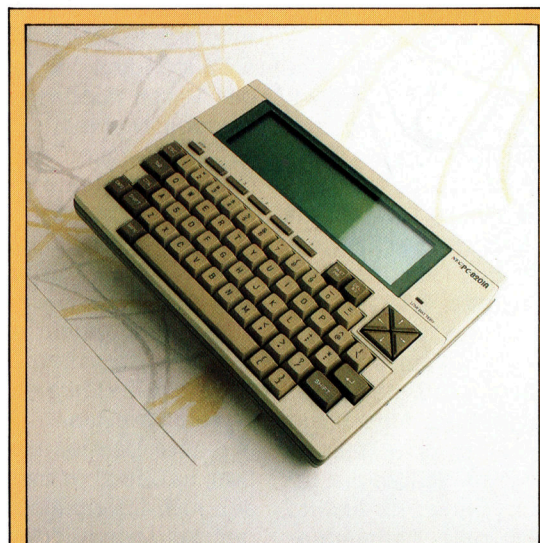
Paralleler Druckeranschluß

Cassettenrecorderanschluß

Anschluß für einen Strichcodeleser

Zentraleinheit
Die CPU ist ein CMOS-8-Bit-80C85-Chip, der nur wenig Strom verbraucht.

RAM-Erweiterung
Der Tandy 100 läßt sich hier mit weiteren RAM-Chips auf 32 K aufrüsten.



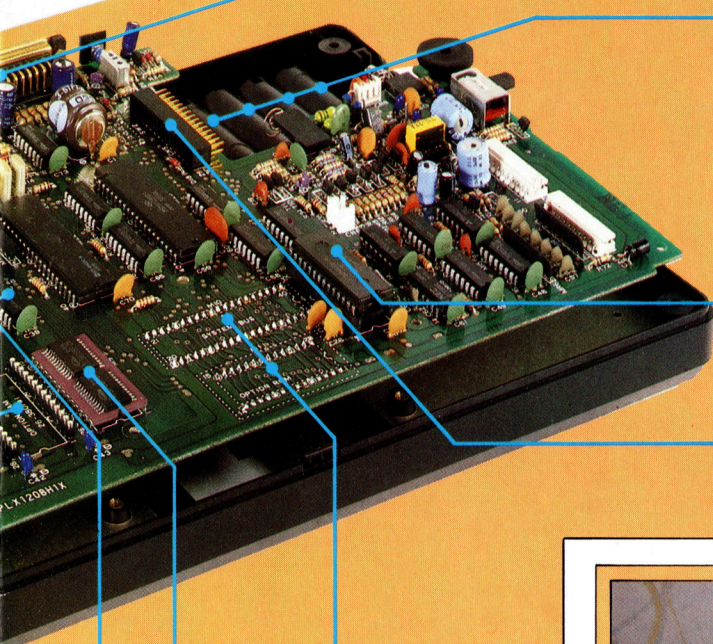
NEC PC8201A

Obwohl der NEC PC8201A genauso groß ist wie seine Nachbauten, unterscheidet sich seine Tastatur wesentlich von der anderer Maschinen. Die Cursorsteuerung ist in einem Viererblock zusammengefaßt, er hat nur fünf Funktionstasten, und auch die Aufteilung der Tastatur ist anders. Im ROM befinden sich nur drei Standardprogramme.



Modemanschluß

Die Maschine verfügt über die serielle Standardschnittstelle RS232. Die Software für eine Datenfernübertragung ist fest in die Maschine eingebaut.



Stromversorgung

Das Modell 100 kann mit vier Batterien bis zu 20 Stunden betrieben werden. Der Inhalt des Arbeitsspeichers wird von aufladbaren Nickel-Cadmium-Batterien bis zu 30 Tage lang erhalten. Die N/C-Batterien werden bei einem Anschluß an das Netz automatisch aufgeladen.

Standard-ROM

Dieser Chip enthält das Microsoft-BASIC und weitere Software.

LCD-Anschluß

Ein Flachkabel verbindet die Flüssigkristallanzeige mit der Systemplatine.

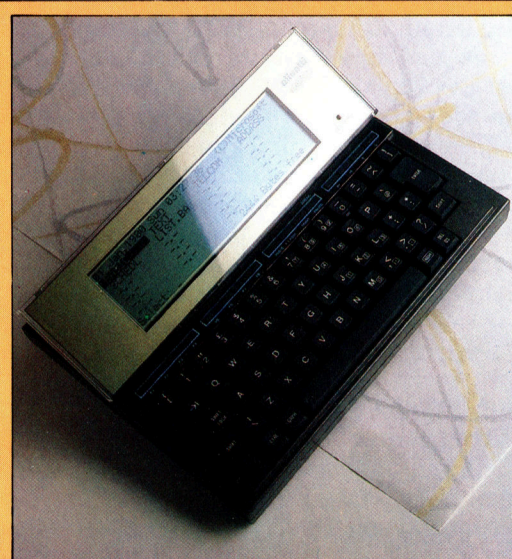
Systembus und Steckleisten für ROM-Erweiterungen

Über diese Steckleisten lassen sich zukünftige Erweiterungen des ROM-Speichers und der Ein- und Ausgabesteuerung anschließen.

Standard-8-K-RAM

Tastatursteuerung

Dieser Chip steuert die Tastatureingabe und enthält die Zeichensätze.



Olivetti M10

Der Olivetti hat ein interessantes Merkmal: Die Flüssigkristallanzeige kann in einem Winkel bis zu 40 Grad aufgestellt werden und läßt sich damit leichter lesen. Die Tastatur und die Standardsoftware sind die gleichen wie beim Tandy 100.

Tandy Modell 100

PREIS

etwa 1700 Mark

ABMESSUNGEN

300x215x50 mm

ZENTRALEINHEIT

8-Bit-80C85-CMOS, 2,4 MHz

SPEICHERKAPAZITÄT

8 K oder 24 K RAM erweiterbar in 8-K-Schritten auf insgesamt 32 K. 32 K ROM enthalten Software und das Microsoft-BASIC.

BILDSCHIRMDARSTELLUNG

Flüssigkristallanzeige; 8 Zeilen mit je 40 Zeichen; 240x64 einzelnen ansprechbare Bildpunkte; ASCII und internationale Zeichensätze, 39 Grafikzeichen.

SCHNITTSTELLEN

Für einen parallelen Drucker, Cassettenrecorder, RS232, Strichcodeleser, Systembus.

PROGRAMMIERSPRACHEN

Microsoft-BASIC

TASTATUR

Standard-Schreibmaschinentastatur mit 56 Tasten und umstellbarem Zehnerblock, 8 programmierbare Funktionstasten, 4 Befehlstasten und 4 Tasten für die Cursorsteuerung.

HANDBÜCHER

BASIC-Kurzhandbuch mit 48 Seiten; Betriebshandbuch mit 200 Seiten.

STÄRKEN

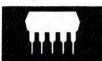
Der Tandy 100 ist klein, besitzt aber alle Eigenschaften, die für ernsthafte Programmierung notwendig sind. Durch den permanenten Arbeitsspeicher und den Batteriebetrieb ist das Gerät für mobile Datenverarbeitung geeignet.

SCHWÄCHEN

Durch den kleinen Arbeitsspeicher läßt er sich nicht mit den kommerziell einsetzbaren Portables vergleichen.

ABWEICHUNGEN

Olivetti M10: Tastatur mit 57 Tasten; 47 Grafikzeichen; bewegliche Anzeigeeinheit, ein Benutzerhandbuch.
NEC PC8201A: Tastatur mit 57 Tasten; 5 Funktionstasten, 16 K RAM erweiterbar auf 96 K; nur drei Grafikzeichen.



Obwohl drei verschiedene Firmen das gleiche Gerät als Tandy Modell 100, Olivetti M10 und NEC PC8201A verkaufen, werden alle drei Varianten von der japanischen Firma Kyocera hergestellt. Die Maschinen unterscheiden sich zwar in einigen kleinen Merkmalen, aber selbst das ungebübte Auge kann sehen, daß diese drei tragbaren Computer den gleichen Ursprung haben.



statur in einen Zehnerblock für Zahleneingaben umwandeln läßt.

Bei Tandy und Olivetti liegen die vier Tasten zur Cursorsteuerung in einer Reihe rechts über der Tastatur, während der NEC PC8201A einen quadratischen Steuerblock mit vier dreieckigen Tasten besitzt.

Alle Maschinen sind mit programmierbaren Funktionstasten ausgestattet, die z. B. für die Datenübertragung zwischen den Programmen eingesetzt werden. Das Modell 100 von Tandy hat acht Funktions- und vier Zusatz Tasten, die interne Aufgaben ausführen. PASTE überträgt Daten von einem Programm zum anderen; LABEL ordnet den Funktionstasten Namen zu, die dem Anwender leichter im Gedächtnis bleiben; PRINT sendet Daten direkt an den Drucker, und die BREAK-Taste unterbricht das augenblicklich laufende Programm. Die Tastatur des Olivetti ist ähnlich ausgelegt, während der NEC eine Pausen- und fünf Funktionstasten besitzt, die zweifach belegt werden können.

Speicherkapazitäten

Das Modell 100 und der M10 werden mit Arbeitsspeichern von acht oder 24 KByte geliefert. Der Einbau eines zusätzlichen RAM-Paketes erweitert sie jedoch auf 32 KByte. Der NEC ist standardmäßig mit 16 KByte ausgerüstet, kann aber über ein internes Zusatzmodul bis auf 64 KByte und über eine Steckleiste bis auf 96 KByte ausgebaut werden.

Das Modell 100 enthält das Microsoft-BASIC und ein System für die Verwaltung der eingebauten Software. Beim Anschalten des Gerätes werden der verfügbare Speicher und die Na-

men der eingebauten Programmmodule auf dem LCD angezeigt.

Unter den mitgelieferten Programmen gibt es das „Text“-System, mit dem sich kurze Briefe oder Aufstellungen schreiben lassen. Das Programm kann gut für kurze Notizen eingesetzt werden und ist daher besonders für Journalisten, Studenten und Geschäftsleute geeignet. „Schedule“ ist ein kleines Datenbanksystem für die Aufzeichnung von Terminen, Ausgaben und anderen Informationen. Mit der eingebauten Suchfunktion können Daten schnell wiedergefunden werden. Die ähnlich strukturierte Datenbank „Address“ ist eigentlich überflüssig, da Schedule die gleichen Aufgaben erfüllt. Schließlich gibt es noch „Telecom“, ein Kommunikationsprogramm, das per RS232-Schnittstelle und Modem auf telefonischem Wege Daten an andere Computer senden oder von dort empfangen kann. Der NEC PC8201A wird mit BASIC, Text und Telecom geliefert.

Alle drei Maschinen besitzen eine ganze Reihe von Schnittstellen: je einen RS232-Anschluß, einen parallelen Druckerausgang, eine Cassettenrecorderbuchse und einen Eingang für Strichcodeleser. Die Modelle von Tandy und Olivetti verfügen außerdem noch über einen Systembus, während der NEC zwei weitere serielle Anschlüsse hat.

Mit dem Einsatz eines einzigen Grundmodells, das von Hersteller zu Hersteller nur wenige Unterschiede aufweist, konnten die Entwicklungskosten auf drei Firmen aufgeteilt werden. Das Ergebnis ist ein Qualitätsprodukt, das von allen drei Herstellern für unter 2000 Mark angeboten wird.



Register und Speicher

Hier wird untersucht, wie im Speicher ein Platz für Maschinencodeprogramme gefunden werden kann. Weiterhin sehen wir, wie die Zentraleinheit den Inhalt des Akkumulators behandelt.

In unserer letzten Folge haben wir ein einfaches Assemblerprogramm entwickelt, es in den Maschinencode übersetzt (assembliert), in den Speicher geladen und ausgeführt. Die beiden letzten Aufgaben wurden dabei von einem Monitorprogramm übernommen. Ein höher entwickeltes Programm mit integriertem Assembler hätte auch die Assemblierung ausführen können. Wenn Sie erst die Grundlagen des Vorgangs verstanden haben und Ihre Programme länger werden, ist die Handübersetzung eine lästige und fehleranfällige Aufgabe; zu dem Zeitpunkt sollten Sie sich ein Assemblerprogramm für Ihre Maschine anschaffen.

Unser kurzes Programm im Maschinencode enthält einige wesentliche Merkmale: Mit einem CPU-Register haben wir Daten verschoben, weiterhin brauchten wir eine Entscheidung, in welchem Bereich des Speichers wir den Code unterbringen wollen, und außerdem mußte der Microprozessor veranlaßt werden, den Code auszuführen. All diese Aspekte der Assemblerprogrammierung können einen Anfänger verwirren. Wir werden daher genau darauf eingehen und uns zunächst mit dem für Maschinencode reservierten Speicherplatz beschäftigen.

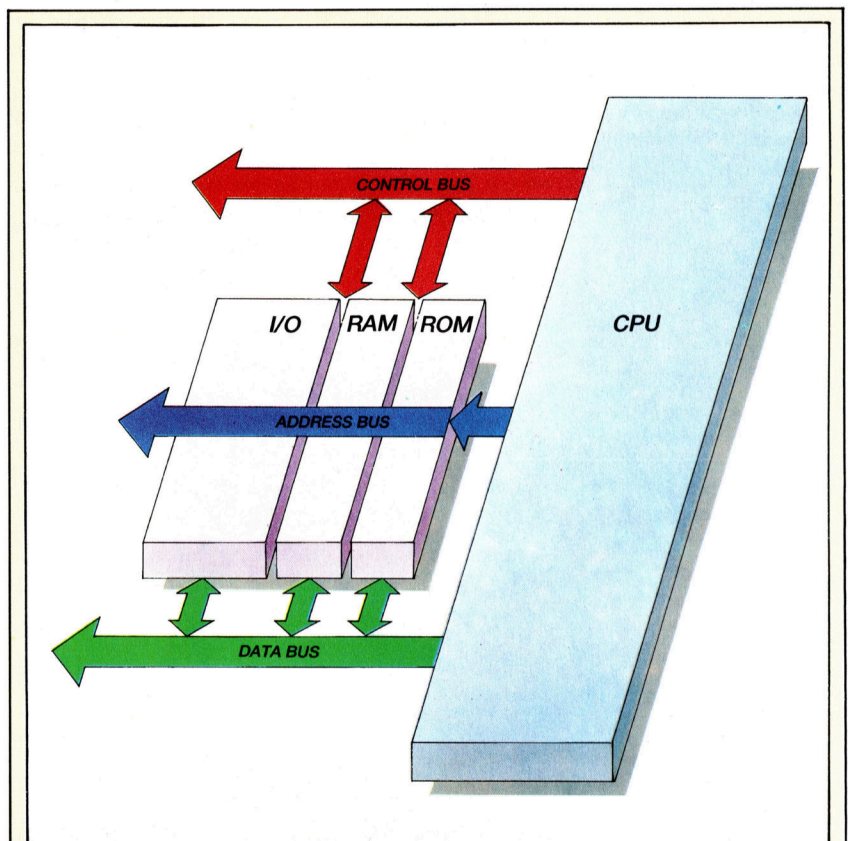
Speicherbereiche für Maschinsprache

Zwar wäre es möglich, überhaupt kein BASIC einzusetzen und damit den gesamten dafür reservierten Bereich zur Verfügung zu haben, doch normalerweise wird zumindest ein Teil des Programms in BASIC geschrieben und der Maschinencode nur für Zwecke eingesetzt, für die BASIC zu langsam ist (z. B. Bewegungsabläufe auf dem Bildschirm). Wenn BASIC und Maschinenprogramme aber gleichzeitig im RAM vorhanden sein sollen, dann müssen wir dem BASIC einige Bereiche wegnehmen und dem Maschinencode zur Verfügung stellen. Wir können dafür die Grenzen des Textbereiches für BASIC-Programme verändern oder vom BASIC oder Betriebssystem nicht genutzte Teile belegen. Auf dem Acorn B lassen sich die Grenzen des BASIC leicht verschieben, da die entsprechenden Adressen in den System-

variablen PAGE, TOP, LOMEM und HIMEM enthalten sind. So zeigt PAGE z. B. auf den Anfang des Bereiches für BASIC-Programmtexe – normalerweise \$1200. Wenn Sie folgenden Befehl eingeben:

```
PAGE=PAGE+500
```

speichert das Betriebssystem alle BASIC-Programme 500 Bytes höher im Speicher und stellt



Der Aufbau kleiner Systeme

In seiner einfachsten Form besteht ein typisches Computersystem aus einem Speicher und einer CPU. Der Speicher enthält ROM-Chips mit Systemprogrammen, RAM-Chips und Spezialchips für die Steuerung von Ein- und Ausgabevorgängen.

Busse tragen Daten und Steuersignale von einem Systemelement zum anderen. Sie sind Flachkabeln ähnlich und können ein Byte oder auch mehr Daten gleichzeitig übertragen. Einige Busse können Daten nur in eine Richtung übermitteln, z. B. Adreßbusse, während andere in beiden Richtungen funktionieren, wie z. B. Datenbusse. Ein Steuerbus überträgt die internen Schaltimpulse des Systems und öffnet und schließt damit Gatter, die den Datenfluß steuern.



Ihnen damit einen Bereich von 500 Bytes für Maschinencodeprogramme zur Verfügung. Bei anderen Systemen kann die gleiche Wirkung erzielt werden, indem die Systemzeiger (siehe Speicherdiagramme) über POKE mit höheren Adressen belegt werden. Sie können auch die obere Grenze des Bereiches für BASIC-Programmtexte nach unten verlegen. Auf dem Spectrum erzielt der Befehl CLEAR gefolgt von einer Adresse diese Wirkung. Bei all diesen Grenzverschiebungen muß nur darauf geachtet werden, daß noch genügend Platz für die BASIC-Programme übrigbleibt.

Auch im für das Betriebssystem reservierten RAM lassen sich kleine Blöcke mit freiem Speicherplatz finden. Ein typisches Beispiel dafür ist der Cassettenbuffer des Commodore 64. Er besteht aus 192 RAM-Bytes (von \$033C bis \$03FB) und wird von dem Betriebssystem nur beim Einsatz eines Cassettenrecorders angesprochen. Dieser Platz reicht vielen Pro-

grammierern völlig aus.

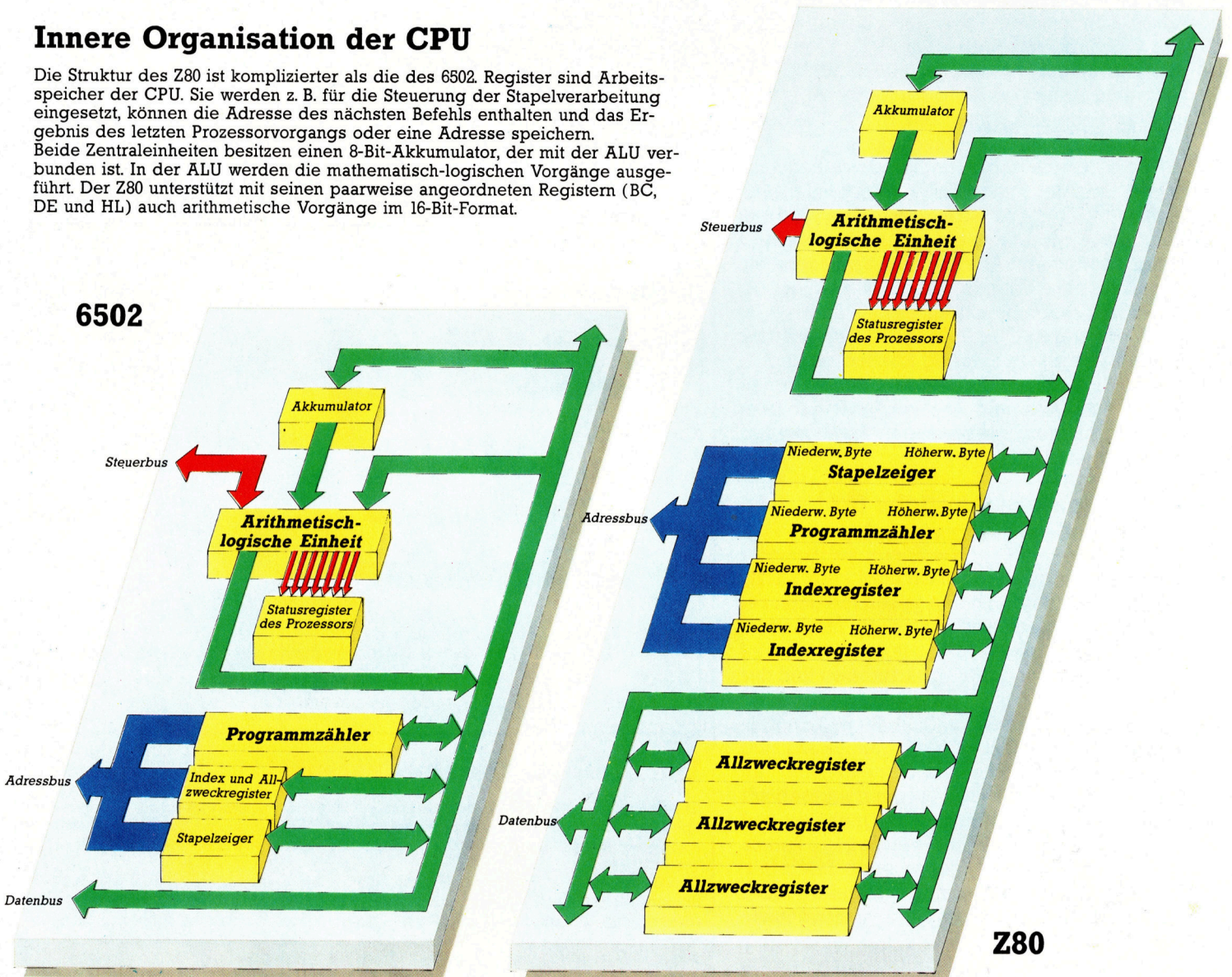
Doch selbst in BASIC-Programmen lassen sich kleinere freie Speicherbereiche finden, z. B. in REM-Zeilen. Der Speicherbereich für BASIC-Text enthält in der Zeile

```
10 REM*****
```

25 Bytes mit der Zahl \$2A (ASCII-Code für "*"). Diese Bytes werden vom Betriebssystem oder vom Interpreter nie angesprochen, da sie den Befehl REM als „ignoriere den Rest dieser Zeile“ interpretieren. In einer derartigen Programmzeile können die Sterne ungestört vom Maschinencode überschrieben werden. Diese scheinbar unsaubere Methode (oft für Programme auf dem erweiterten ZX81 eingesetzt) hat einen großen Vorteil: Wenn Sie den BASIC-Programmtext mit SAVE speichern und dann mit LOAD wieder aufrufen, bleibt der Maschinencode erhalten. Bei anderen Methoden muß der Maschinencode normalerweise getrennt

Innere Organisation der CPU

Die Struktur des Z80 ist komplizierter als die des 6502. Register sind Arbeitsspeicher der CPU. Sie werden z. B. für die Steuerung der Stapelverarbeitung eingesetzt, können die Adresse des nächsten Befehls enthalten und das Ergebnis des letzten Prozessorvorgangs oder eine Adresse speichern. Beide Zentraleinheiten besitzen einen 8-Bit-Akkumulator, der mit der ALU verbunden ist. In der ALU werden die mathematisch-logischen Vorgänge ausgeführt. Der Z80 unterstützt mit seinen paarweise angeordneten Registern (BC, DE und HL) auch arithmetische Vorgänge im 16-Bit-Format.





Akkumulator

Der Akkumulator ist das zentrale Register der CPU. Alle arithmetischen Vorgänge und auch die Datenübertragung werden über dieses Register geleitet – der 6502 macht davon jedoch mehr Gebrauch als der Z80.

Arithmetisch-logische Einheit

Sie enthält ein Additionsmodul für Binärzahlen und logische Gatter, durch die der Zugang zu den individuellen Bits der Register und des Datenbusses möglich wird. Mit der entsprechenden Steuerung können damit Addition, Subtraktion und Boolesche Vergleiche ausgeführt werden.

Statusregister des Prozessors

Bei fast jedem Vorgang in der CPU erhalten die einzelnen Bits dieses Registers Informationen über Teilaspekte des Vorgangs, z. B. ein Übertragungsbit aus dem Ergebnis einer Addition.

Programmzähler

Dieses Register enthält die Speicheradresse mit dem nächsten Op-Code, den die CPU ausführen soll. Die BASIC-Funktion USR (Adresse) bewirkt, daß die angegebene Adresse direkt in den Programmzähler geladen wird und die CPU den von dieser Adresse an gespeicherten Code aufrufen und ausführen kann.

Stapelzeiger

In diesem Register ist die Adresse des nächsten freien Bytes des für die CPU reservierten RAM-Bereiches gespeichert. Wenn die CPU Daten auf dieser Adresse ablegt, wird der Stapelzeiger automatisch auf das nächste freie Byte gesetzt.

Index- und Allzweckregister

Indexregister werden von der CPU eingesetzt, um den Speicher auf mehrere verschiedene Arten ansprechen zu können. Allzweckregister werden als allgemeine Ablage oder für Spezialaufgaben der CPU eingesetzt.

gespeichert werden. Dennoch hat die REM-Methode einen großen Nachteil: Beim Anzeigen eines Programmtextes mit LIST interpretiert das Betriebssystem den Maschinencode als ASCII-Zeichen und kann damit Chaos auf dem Bildschirm hervorrufen. Hier wird auch deutlich, warum in den Beispielprogrammen des Acorn B und des Spectrum Warnungen enthalten sind. Der Commodore 64 lädt seinen Maschinencode in den Cassettenbuffer, der Acorn und der Spectrum jedoch in die erste REM-Zeile – daher die Warnung, diese Programmversionen nicht mit LIST anzeigen zu lassen.

Ausführung starten

Wenn Sie ein Assemblerprogramm geschrieben, in den Maschinencode übersetzt und in den RAM-Bereich Ihrer Wahl geladen haben, können Sie es ausführen lassen. Dieser Vorgang wird mit den BASIC-Befehlen CALL (nur Acorn), SYS (nur Commodore 64) oder USR (alle drei Maschinen) in Gang gesetzt. Allen drei Befehlen folgt die Adresse des ersten Bytes im Maschinencode. Für den Interpreter bedeutet das: „Führe den Maschinencode aus, der an der angegebenen Adresse anfängt, und gehe bei einem Rücksprungbefehl des Maschinencodes (RET oder RTS) auf den nächsten Befehl des BASIC-Programmes.“ Der Ablauf ähnelt dem BASIC-Befehl GOSUB.

In unserer letzten Folge haben wir ein Programm geschrieben, das den Inhalt eines Bytes in ein anderes kopiert. Wir haben dazu den Akkumulator mit dem Inhalt der ersten Adresse geladen und dann den Inhalt des Akkumulators in die andere Adresse übertragen. Dieser Vorgang zeigt die zentrale Rolle der CPU: Alle Daten und die Steuerung müssen vom Speicher in die CPU und von dort wieder in den Speicher fließen. Während im BASIC LET X=Y (d.h. „kopiere den Inhalt von Y direkt nach X“) möglich ist, können wir in der Assemblersprache nur vom Speicher in die CPU und von der CPU in den Speicher kopieren. Die Register der CPU (siehe Bild) sind RAM-Bytes, die sich im Prozessor selbst befinden. In diesen Registern werden Daten des Speichers abgelegt oder bearbeitet. Sowohl der Z80 als auch der 6502 besitzen ein Register, das Akkumulator heißt und von den meisten Befehlen der Assemblersprache angesprochen wird. In ihm finden fast alle Rechenoperationen statt.

Wenn Sie z. B. die Zahlen \$42 und \$07 addieren wollen (das \$-Symbol kennzeichnet die Ziffern als Hexadezimalzahlen), brauchen Sie nur eine dieser Zahlen in den Akkumulator zu laden und die andere darauf zu addieren – die Summe wird sich in diesem Register regelrecht „ansammeln“ (englisch: accumulate). Im folgenden Kasten finden Sie die Befehle des Z80- und des 6502-Microprozessors für diesen Vorgang.

Z80	6502
LD A,\$42 ADC A,\$07	LDA #\$42 ADC #\$07

Beide Befehle des Z80 geben die Zahlen direkt an, während den Zahlen der Version des 6502 ein # vorangestellt werden muß, damit sie als Zahlen und nicht als Adressen interpretiert werden. LDA #\$65 bedeutet daher: „lade die Zahl \$65 in den Akkumulator“, während LDA \$65 die Bedeutung hat: „lade den Inhalt des Bytes mit der Adresse \$65 in den Akkumulator“. Auf die gleiche Weise funktioniert der Additionsbefehl (das mnemotische Kürzel ist in beiden Versionen zufällig identisch). ADC bedeutet: „addiere einen festgelegten Operanden zu dem Inhalt des Akkumulators.“ Die Zahlen \$42 und \$07 sind in unserem Beispiel sogenannte „Konstante“, so daß LDA #\$42 als „lade den Akkumulator mit der Konstanten #\$42“ verstanden wird.

Nach der Ausführung der Befehle steht die Summe beider Zahlen im Akkumulator. Da Sie dies nicht „sehen“ können, muß der Inhalt des Akkumulators in ein RAM-Byte übertragen werden, das sich abfragen läßt. Das vollständige Programm sieht folgendermaßen aus:

Maschinencode	Assemblersprache
Z80	
A7 3E 42 CE 07 32 ?? ?? C9	AND A LD A,\$42 ADC A,\$07 LD BYTE1,A RET
6502	
18 A9 42 69 07 8D ?? ?? 60	CLC LDA #\$42 ADC #\$07 STA BYTE1 RTS

Wir werden für den Augenblick nicht auf die Bedeutung des ersten Befehls beider Programme eingehen. Der vierte Befehl enthält statt einer direkten Adresse das Symbol BYTE1. Da der Wert von BYTE1 von Maschine zu Maschine variiert, haben wir lediglich ein Symbol eingesetzt.

Jetzt müssen Sie sich überlegen, wo Sie den Maschinencode abspeichern wollen und welche Adresse BYTE1 zugeordnet werden soll. Wählen Sie einen Speicherbereich für Ihren Code aus und setzen Sie an die Stelle von BYTE1 die Adresse des Bytes, das dem Programm unmittelbar folgt. Diese Adresse muß im Maschinencode das lo-hi-Format haben. Danach laden, starten und untersuchen Sie den Maschinencode mit dem Monitorprogramm. Der Inhalt des Ergebnisbytes sollte \$49 sein.



Gut zuhören!

Farbe und Bewegung bilden den Schwerpunkt bei fast allen gängigen Computerspielen, Sprache und Geräusche führen dagegen eher ein Schattendasein. Hier sind Vorschläge, wie Sie witzige und interessante „Audio-Spiele“ selbst programmieren können.

```

10 REM*****AUDIO GAME*****
***
20 PRINT"A SIMPLE AUDIO GAM
E":PRINT:PRINT"MOVE YOUR SPACE
SHIP TO THE BEACON BEFORE
YOUR FUEL RUNS OUT"
30 PRINT:PRINT" THE NEA
RER YOU ARE, THE HIGHER
THE BEACON'S PITCH"

40 PRINT:PRINT" C
ONTROLS ARE:":PRINT" I (UP) M
(DOWN) J (LEFT) K (RIGHT)"
50 PRINT:PRINT" *****PRE
SS ANY KEY TO START*****"
70 AS=INKEY$(0):IF AS="" TH
EN GOTO 70
80 P=INT(RND(1)*500)+1:G=IN
T(RND(1)*500+1):F=3*(P+G)/2:X=
0:Y=0
90 XD=0:YD=0
100 PRINT "FUEL=";F," DISTAN
CE=";
110 IF (ABS(P-X)<2 AND ABS(G
-Y)<2) THEN PRINT"YOU MADE IT
-WITH ";F;" FUEL UNITS LEFT":
END
120 AS=INKEY$(0):IF AS="" TH
EN GOTO 120
130 IF (AS="I" AND YD<3) TH
EN YD=YD+1
140 IF (AS="M" AND YD>-3) TH
EN YD=YD-1
150 IF (AS="J" AND XD<-3) TH
EN XD=XD-1
160 IF (AS="K" AND XD<3) TH
EN XD=XD+1
170 X=X+XD:Y=Y+YD
180 D=SQR((P-X)*(P-X)+(G-Y)*
(G-Y))
190 PRINT D
200 SOUND 1,-8,(255-D),2
210 F=F-ABS(XD)-ABS(YD):IF F
<0 THEN GOTO 90
230 PRINT TAB(10);"*****CR
ASH*****":PRINT TAB(11);"***
OUT OF FUEL***"
240 PRINT:PRINT"POSITION IS
";D;" SPACE UNITS FROM BASE"
260 END

```

In den professionellen Videospiele werden Geräuscheffekte meist nur eingesetzt, um das Geschehen auf dem Bildschirm realistischer und packender zu gestalten. Dabei gibt es keinen vernünftigen Grund, sich bei Spielen auf die Optik zu beschränken – warum nicht auch einmal das Gehör in den Mittelpunkt eines Computerspieles rücken?

Sicher ist das hier vorgestellte „Hör-Spiel“ nicht gerade eines der aufregendsten – es macht aber trotzdem Spaß. Was zuerst eine recht einfache Aufgabe zu sein scheint, erweist sich nach und nach als faszinierende Herausforderung – schon deshalb, weil Sie sich plötzlich auf ein anderes Sinnesorgan als Ihre Augen verlassen müssen, wie sie es vom Umgang mit der farbenprächtigen Bildschirmdarstellung gewöhnt sind.

In diesem Spiel sollen Sie ein Raumschiff steuern, dem gerade der Treibstoff ausgeht. Nur durch rechtzeitiges Anlegen am nahegelegenen Treibstoff-Depot können Sie sich noch retten. Eine Computer-Störung verhindert aber die visuelle Überwachung des „Rendezvous-Manövers“ – nur ein Funkfeuer-Signal steht zur Verfügung. Je höher die Frequenz des Signals ist, um so näher sind Sie dem Ziel. Sie müssen also sehr genau hinhören.

Wie ein reales Raumschiff fliegt ihre Rakete so lange in der einmal eingeschlagenen Richtung weiter, bis Sie gegensteuern: Geht es durch den doppelten Druck auf „U“ schnell aufwärts, kommt das Raumschiff auch erst nach einem zweifachen Druck auf „D“ wieder zum Stillstand – eine zusätzliche Klippe beim genauen Navigieren!

Sie werden wahrscheinlich einige Zeit üben müssen, bis das Ziel zum ersten Mal erreicht

wird. Mit ein paar zusätzlichen Programmzeilen können Sie sich die Aufgabe natürlich auch erleichtern: Die Anzeige der Variablen D gibt etwa den Abstand des Raumschiffs vom Treibstoffdepot an. Ebenso hilfreich ist es, einen Tip für den nächsten Tastendruck zu bekommen – $SGN(p-x)=-1$ würde etwa angeben, daß man nun besser nach links steuern sollte. Mit diesen kleinen Tricks können Sie Ihre Überlebenschance im Weltraum stark verbessern!

Wenn Sie unser Programm einmal ausprobieren haben, bekommen Sie vielleicht Lust, selbst einmal ein kleines Spiel zu erfinden. Dabei eignen sich Töne besonders gut zum Suchen und zum Umgehen von Hindernissen. Wie wär's mit einem Navigationsspiel für Unterseeboote, bei dem Sie nach den Warntönen eines Minendetektors steuern müssen? Auch das aus alten Filmen vertraute „Safeknacken nach Gehör“ ließe sich als Übung verwenden.

BASIC-Dialekte

Dieses Programm wurde für den Acorn B in Mode 7 erstellt, damit entspricht es fast vollständig dem Standard-Microsoft-BASIC. Die „PRINT“-Kommandos sind für eine 40-Zeichen-Darstellung eingerichtet worden. Das „SOUND“-Kommando in Zeile 200 entspricht dem besonderen Acorn-BASIC: Der Wert des Parameters (255-D) gibt die Tonhöhe an, die anderen Parameter beziehen sich auf die Lautstärke, die Tondauer und den angesprochenen Tonkanal. Die Verwendung von INKEY\$(0) in Zeile 120 und von RND in Zeile 80 ist bei anderen Rechnern nicht zulässig – die Kommandos müssen verändert werden:

Spectrum

Bei allen Variablendefinitionen LET hinzufügen. INKEY\$(0) durch INKEY\$ ersetzen. RND(1) durch RND ersetzen. Zeile 200 ändern: 200 BEEP 0.4, (255-D) Die Programmzeilen 15 und 195 einfügen:

```
15 RANDOMIZE
```

```
195 IF D > 254 THEN LET D=D-254
```

Commodore 64/VC-20

INKEY\$(0) durch GET A\$ ersetzen. Tonbefehle nach dem Commodore-Handbuch verändern. Zusätzliche Zeile 75 einfügen: 75 X=RND(-TI)

Dragon

INKEY\$(0) gegen INKEY\$ austauschen. RND(1) gegen RND(0) austauschen. Zeile 200 ändern:

```
200 SOUND (255-D),10
```

Zeile 195 ergänzen: 195 IF D>254 then D=D-254

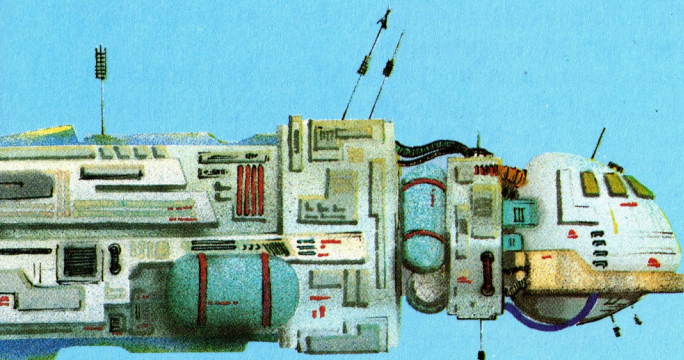
Oric Atmos

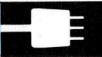
INKEY\$(0) gegen KEY\$ austauschen.

Zeile 200 ändern:

```
200 SOUND 1,(255-D),9:WAIT 40:PLAY0,0,0
```

Zeile 195 einfügen: 195 IF D>254 THEN D=D-254





Camera Obscura

Der Acorn B bietet die verschiedensten Möglichkeiten der Grafik-Wiedergabe, die den Aufbau komplexer Bilder zulassen. Unter der Bezeichnung „EV1“ wird außerdem eine elektronische Kamera angeboten, die Sie einfach auf das gewünschte Motiv richten, um es auf den Bildschirm des Rechners zu bringen.

Die elektronische Kamera EV1 – auch als „Snap“ (Schnappschußkamera) bekannt – ist auch für Heimcomputer-Besitzer erschwinglich. Sie kostet mit der dazugehörigen Software unter 450 Mark, knapp halb so viel wie ein vergleichbares Video-Interface. Die kleine Kamera wird über ein Flachbandkabel mit der Benutzerschnittstelle des Acorn B verbunden. Die mitgelieferte Software (auf Cassette oder Floppy) erlaubt es, das Kamerabild durch einen einfachen Rechnerbefehl auf den Bildschirm zu übernehmen. Das System eignet sich nicht nur für Spielereien, sondern auch für ernsthafte Anwendungen, beispielsweise für Alarmanlagen oder automatische Bildererkennung, vielleicht sogar als „Auge“ für einen Roboter.

Die EV1 arbeitet mit einem ungewöhnlichen 32-K-RAM-Baustein als Bildempfänger hinter dem Objektiv. Dieser Chip ist oben mit einem Fenster versehen. Die 256x128 winzigen Speicherzellen, die auf dem Siliziumplättchen integriert sind, werden durch den Lichteinfall langsam entladen. Die Geschwindigkeit ist der Lichtintensität proportional. Bei der „Snap“ werden zunächst alle Speicherzellen mit einem bestimmten Wert geladen, etwa mit „Einsen“. Dadurch sind sie alle im Zustand „an“. Nach kurzer Belichtung wird durch Lesen jeder Zelle der Entladungszustand festgestellt. Der eingetragene Wert ist nur dort erhalten geblieben, wohin kein Licht fiel. Die vom Licht getroffenen Speicherzellen sind „umgekippt“. Sie sind „aus“. Die Software sorgt dafür, daß auf dem Bildschirm überall dort ein heller Punkt erscheint, wo sich der Speicherinhalt verändert hat. Auf diese Weise wird das Bild vom Chip auf den Bildschirm übertragen.

Die Zeit, die zwischen der Aufladung der Speicherzellen und dem Lesen vergeht, entspricht der „Belichtungszeit“ bei der üblichen Fotografie. Unter günstigen Lichtverhältnissen bewältigt die Snap viele Bilder pro Sekunde. Und auch bei normaler Raumhelligkeit bleibt



Mit der „Snap“-Kamera wird unter der Bezeichnung „EV1“ ein Software-Paket geliefert, das verschiedene Wiedergabemöglichkeiten vorsieht. Sie können auf dem Bildschirm die von der Kamera erfaßten Motive darstellen lassen, ein Bild auf dem Schirm „einfrieren“, auf Diskette abspeichern oder über einen Drucker ausgeben. Die Belichtungssteuerung erfolgt automatisch durch den Rechner, kann aber bei Bedarf auch von Hand mit Hilfe der Cursortasten beeinflusst werden.

die Belichtungszeit unter einer Sekunde.

Die Auflösung ist, bedingt durch das Speicherraster von 256x128 Zellen, nicht besonders hoch. Sie entspricht etwa den Bildschirm-Betriebsarten „Mode 2“ und „Mode 5“ des Acorn B. Die Bildqualität ist jedoch durchaus brauchbar, ungefähr vergleichbar mit Zeitungsfotos. Stärker als die begrenzte Auflösung stört die Anordnung der Speicherzellen auf dem Chip in zwei großen Blöcken. Dazwischen ist nämlich ein toter Streifen, und das dort einfallende bzw. nicht einfallende Licht wird nicht registriert und somit nicht dargestellt. Erstaunlicherweise fällt das meist nicht besonders auf – bei vielen Bildern ist es überhaupt nicht zu merken.

Das Kunststoffgehäuse der Kamera hat etwa die Größe einer Zigaretenschachtel. Vorn ist das Objektiv, an der Unterseite finden Sie ein Normgewinde zum Einschrauben eines Kame-

rastativ, um etwa lange Belichtungszeiten zu ermöglichen. Das zwei Meter lange Flachbandkabel der Kamera (achtadrig) wird am User-Port des Acorn B eingesteckt, und damit ist das System auch schon betriebsbereit.

Das Objektiv der EV1 (in einer Standard-Bajonettfassung) stammt von der kleinen Pentax-110-Spiegelreflexkamera. Passende Wechselobjektive gibt es in Fotogeschäften mit den verschiedensten Brennweiten, auch als Zoom-Linsen. Allerdings entspricht der Abstand der Objektivfassung vom RAM-Chip bei der „Snap“ nicht dem Filmebenenabstand bei der Pentax, so daß die Entfernungsskala nicht mehr stimmt. Es lohnt sich, die Skala nachzueichnen; denn das Scharfstellen der Snap kann ziemlich langwierig sein.

Was letztlich auf dem Bildschirm erscheint, kann über die Software weitgehend beeinflusst werden. Mit dem System erhalten Sie eine Reihe von Programmen und eine 50seitige Anleitung (mit einem Apple Macintosh geschrieben), in der die Software und die Funktionen der Kamera im einzelnen erklärt sind.

Harte oder weiche Aufnahmen

Sie können mit der EV1 auf zweierlei Weise arbeiten: Eine Einzelaufnahme liefert nur ein „hartes“ Schwarzweißbild (ohne Halbtöne). – Aus einer Serie von Aufnahmen mit unterschiedlicher Belichtung läßt sich dagegen ein Bild mit abgestuftem Kontrast aufbauen. Das Programm für das Einzelbildverfahren ist das erste des mitgelieferten Softwarepakets. Es arbeitet in der Betriebsart „Mode 4“ mit einem kleinen Bildschirmausschnitt, wobei fortlaufend das zuletzt aufgenommene Bild dargestellt wird. Die Belichtungszeit wird mit zwei Cursortasten eingestellt. Die Bildausgabe kann auch auf einem EPSON- (oder EPSON-kompatiblen) Drucker erfolgen.

Das Programm für die realistischere Wiedergabe mit abgestuftem Kontrast baut das Bild durch Überlagerung aus einer Belichtungsreihe von acht Aufnahmen im „Mode 0“ auf. Dabei werden die Hellzonen in den Einzelbildern um so intensiver dargestellt, je kürzer die Belichtungszeit war. Das Ergebnis ist ein Bild mit acht Graustufen zwischen Schwarz und Weiß. Das wirkt naturgetreuer, erfordert aber bei normaler Raumhelligkeit Aufnahmezeiten bis zu zehn Sekunden – ein bißchen viel für Schnappschüsse.

Das Programm „Secure“ macht aus dem EV1-System eine Computer-Alarmanlage. Es achtet nur auf Unterschiede zwischen aufeinanderfolgenden Aufnahmen desselben Objekts. Wenn die Veränderungen eine bestimmte Schwelle überschreiten, wird der Alarm ausgelöst. Bei richtiger Einstellung ignoriert die Anlage, beispielsweise auf einen Hauseingang gerichtet, Blätterbewegungen im Wind, spricht aber bei Erscheinen eines Besu-

chers vor der Tür an.

Das Programm „Movie“ speichert eine kurze Folge harter Einzelbilder, die dann trickfilmähnlich rasch abgespielt werden. Außerdem gibt es ein Programm namens „Animal“, eine Videofassung eines bekannten Computerspiels. Dabei wird zunächst ein Bild mit mehreren Objekten vor die Kamera gehalten. Das Programm analysiert das Bild, merkt sich die Umrisse der einzelnen Objekte und fordert Sie auf, den Gegenständen Namen zu geben. Wenn Sie die Kamera dann auf eine andere Szene richten, versucht das Programm, Objekte zu finden, die bereits auf dem ersten Bild zu sehen waren.

Den größten Reiz bietet vermutlich das Programm „Arty“, mit dem sich aus Einzelaufnahmen komplexe Grafiken zusammensetzen lassen. Dabei können Sie auf dem ganzen „Mode 1“-Schirm mit dem Joystick Kameraaufnahmen an beliebiger Stelle plazieren – auch vergrößern.

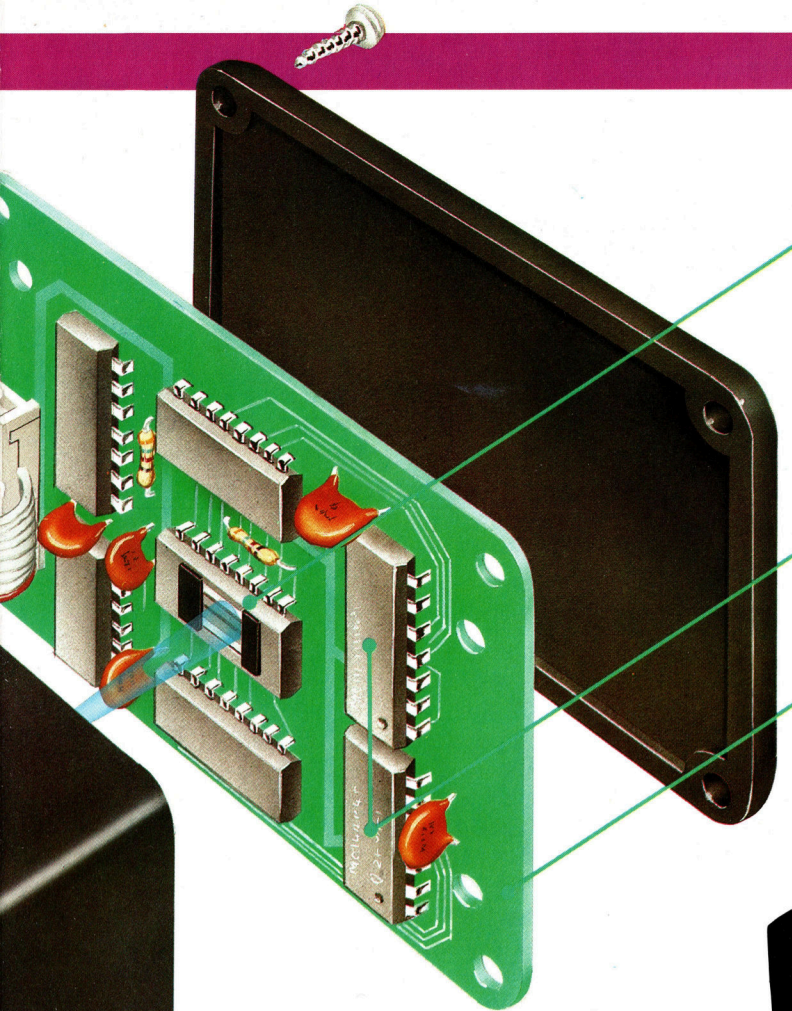


Kabel zum Rechner
Die Kamera wird mit einem Flachbandkabel an die Benutzer-Schnittstelle des Acorn B angeschlossen.

Kameragehäuse

Scharfeinstellung

Objektiv
Die Kamera wird mit einem 18-mm-Weitwinkelobjektiv (Lichtstärke 1:2,8) oder mit einem 24-mm-Normalobjektiv geliefert.



Lichtempfindliches RAM
 Als Bildempfänger dient ein dynamisches RAM mit einem Fenster an der Vorderseite. Die Speicher-Zellen verlieren bei Belichtung ihre Ladung.

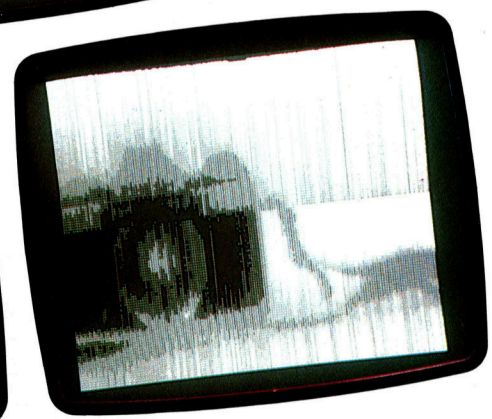
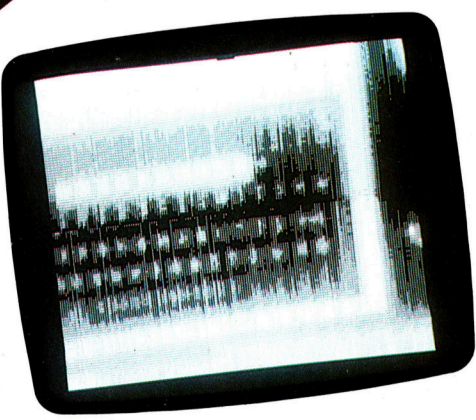
Steuerbausteine

Leiterplatte

Snap/EV1 Video Preis:
 etwa 450 Mark
Abmessungen:
 70x50x25 mm
Objektive: 24 mm oder 18 mm Brennweite
Anschluß: 20-poliger Stecker für das User-Port des Acorn B
Anleitung: für Kamera und Software, 50-seitig
Stärken: Kameraaufnahmen können im Bildspeicher oder auf Diskette festgehalten werden.
Schwächen: Die Auflösung ist dürftig, was manchmal auch die Einstellung der Kamera schwierig macht.



Auflösungsverhältnisse
 Die Aufnahmen werden entweder als „hartes“ Schwarzweißbild aus 256x128 Punkten oder in einer Graustufendarstellung wiedergegeben. Die Auflösung genügt bescheidenen Ansprüchen.



Sanfte Schattierungen
 Das Programm „Grey“ arbeitet mit einer differenzierten Kontrastwiedergabe in acht Graustufen (Halbtönen) von Schwarz bis Weiß. Das Bild wird dabei unter Nutzung des vollen Schirmformats aus hellen und dunklen vertikalen Linien zusammengesetzt, wobei der Abstand der hellen Linien entsprechend dem Bildinhalt variiert wird.

bert oder verkleinert. Vorder- und Hintergrundfarben sind über die Funktionstasten frei wählbar. Sie brauchen für diese Grafiken etwas Geduld, aber Sie werden durch detailreiche farbige Bilder aus einer Vielfalt lebendiger Motive belohnt.
 Die mitgelieferte Software ist zwar umfangreich und gut durchdacht, deckt aber nur einen

begrenzten Anwendungsbereich ab. Beispielsweise sind Graustufenbilder nur im „Mode 0“ darstellbar. Es wäre günstiger, wenn sich wahlweise „Mode 2“ (unter Nutzung der Farbspeicherkapazität für die Graustufen) verwenden ließe. Die Anleitung informiert ausführlich über die in der Software enthaltenen Maschinencode-Programme, aber diese Informationen setzen umfangreiche Kenntnisse der Materie voraus.

Das Spectrum des ZX-BASIC

BASIC ist zur Standard-Sprache der Microcomputer geworden, doch arbeitet fast jeder Computer mit einer anderen Variante bzw. einem anderen Dialekt. In der folgenden Serie werden einige dieser Dialekte und deren Funktionen vorgestellt.

Als erstes wird die unterschiedliche Schreibweise der Variablennamen aufgezeigt. Im Sinclair-BASIC dürfen String-Variablennamen nur einen Buchstaben umfassen. Dabei ist es gleichgültig, ob es sich um einen Groß- oder Kleinbuchstaben handelt. Dies bedeutet, daß sich die Variablen a\$ und A\$ auf dieselbe Speicherstelle beziehen.

Numerische Variablennamen müssen mit einem Buchstaben beginnen und aus Buchstaben oder Zahlen bestehen. Ihre Länge ist nicht von Bedeutung. Sie dürfen Leerstellen enthalten, und man kann Klein- und Großbuchstaben beliebig miteinander mischen. Einige gültige numerische Variablennamen sind:

qwert, ub40, computer kurs

Die folgenden Namen haben die gleiche Bedeutung:

QWERT, UB40, Computer Kurs

Die Namen für numerische Bereiche müssen aus einzelnen Buchstaben bestehen. Dabei können auch die Namen normaler numerischer Variablen noch einmal verwendet werden. Die Variable für den Bereich v(8) unterscheidet sich völlig von der einfachen numeri-



SuperBASIC

Obwohl Sinclairs SuperBASIC gegenüber dem ZX-BASIC einen erweiterten Befehlssatz bietet, ist die bemerkenswerteste Tatsache das nicht mehr vorhandene „single-key reserved word entry system“, das noch beim ZX80, ZX81 und Spectrum verwendet wurde. Dieses System beinhaltet, daß Sie mit einem einzigen Tastendruck eine ganze Anweisung auf einmal eingeben können, erfordert jedoch eine Vielzahl verschiedener Modi, um zwischen einer Eingabe einfacher Zeichen und Schlüsselwörtern zu differenzieren.

schen Variablen v. Für FOR . . . NEXT-Schleifen müssen normale numerische Variablen wie v verwendet werden, zum Beispiel FOR V=1 TO 9 . . . NEXT V.

Der Hauptunterschied zwischen dem Sinclair-Dialekt und anderen BASICs liegt in der Handhabung von Stringlängen. Beginnen wir mit der DIM-Anweisung. Beim Sinclair-BASIC werden nach Ausführung der Anweisung DIM a\$(12) zwölf Bytes des Speichers für die Variable a\$ reserviert. Diese Speicherstellen werden zuerst mit Leerzeichen gefüllt. Jedes dieser Bytes kann als einzelne, untergeordnete Variable angesehen werden, oder die zwölf Bytes werden zusammen als a\$ bezeichnet. Beim Versuch, der Variablen mehr als zwölf Zeichen zuzuordnen, werden alle überzähligen Zeichen ignoriert. Bei der Anweisung

```
DIM a$(12):LET a$="123456789"
```

enthält a\$ die Zeichen 123456789 sowie drei Leerzeichen, um eine Gesamtzahl von zwölf Zeichen bzw. Bytes aufzufüllen.

```
DIM a$(12):LET a$="ABCDEFGHIJKLMN"
```

a\$ enthält jetzt die Zeichen ABCDEFGHIJKL. Die gesamte Länge des Strings ABCDEFGHIJKLMN wurde auf der rechten Seite abgeschnitten, damit sie in die DIMensierte Länge von a\$ paßt. Bei

```
LET a$(2 TO 5)="1234"
```

enthält a\$ jetzt die Zeichen A1234FGHIJKL. Dies verdeutlicht die Stärke des Sinclair-String-Handlings. – Alle Strings werden als eindimensionale String-Bereiche behandelt, und es kann jederzeit Zugriff auf individuelle Elemente eines Bereiches genommen werden. Dadurch wird ein gravierender Unterschied zu einigen anderen BASIC-Versionen deutlich, bei denen DIM a\$(12) zwölf separate String-Variablen bildet.

Bei anderen BASIC-Versionen erfordert diese Art des String-Handlings diverse String-Funktionen, wie LEFT\$, RIGHT\$, MID\$ und INSTR, um Teilstring-Manipulationen und String-Aufspaltung in der eben gezeigten Form zu ermöglichen. Beim Sinclair-BASIC ist das nicht



nötig. Die Sinclair-Äquivalente dieser String-Funktionen sind:

`LEFT$(A$,N) = A$(TO N)`

(wobei mit N die am weitesten links stehenden Zeichen von A\$ gemeint sind),

`RIGHT$(A$,N) = A$(LEN A$-N+1 TO)`

(wobei mit N die am weitesten rechts stehenden Zeichen von A\$ gemeint sind) und

`MID$(A$,P,N) = A$(P TO P+N-1)`

(wobei mit N die Zeichen von Position P bis zum Ende von A\$ gemeint sind).

`LET S=INSTR(A$,"teststring")`

(was soviel bedeutet, daß die Startposition des Teilstrings „teststring“ innerhalb von A\$ gefunden werden soll). Diese Anweisung kann wie folgt ersetzt werden:

`LET Y$=A$:LET Z$="teststring":GO-SUB 9900:LET S=POSN`

`9900 LET ZL=LEN Z$: LET SL=LEN`

`Y$-ZL+1:LET POSN=0`

`9910 FOR K=1 TO SL`

`9920 IF Y$(K TO K+ZL-1)=Z$ THEN LET`

`POSN=K:LET K=SL`

`9930 NEXT K:RETURN`

Beachten Sie in dieser Unterroutine, daß die String-Variable Y\$ als eine Teilstring-Variablen behandelt wird, obwohl sie nicht DIMensioniert wurde. Da im Sinclair-BASIC alle String-Variablen als Bereichs-String-Variablen gehandhabt werden, ist eine nicht DIMensionierte String-Variable automatisch ein unterschiedlich langer, eindimensionaler Bereich von einzelnen Zeichen.

Die große Aufmerksamkeit, die Sinclair-BASIC der Länge von DIMensionierten String-Variablen schenkt, bewirkt, daß einfache Anweisungen sehr unterschiedlich arbeiten können, je nachdem ob DIMensioniert wurde oder nicht. Wenn a\$ eine einfache String-Variable ist, bewirkt `LET a$=""`, daß der Inhalt von a\$ zu einem Null-String ("") und die Länge von a\$ zu 0 wird. Wenn jedoch zuvor die Anweisung `DIM a$(7)` ausgeführt wurde, bewirkt `LET a$=""`, daß der Inhalt von a\$ sieben Leerzeichen entspricht und die Länge den Wert 7 hat. Geben Sie diese Zeile ein:

`IF a$="" THEN PRINT "null-String"`

Dieser Test wird negativ ausfallen, und es wird nichts ausgegeben werden, da a\$ sieben Leerzeichen enthält und somit kein Null-String ist.

Wenn Sie String-Bereichs-Elemente in dieser Form testen wollen, ist es ratsam, eine String-Variable für diesen Zweck bereitzustellen. Sie sollte auf die Länge der längsten im Programm verwendeten Bereichs-Variable DIMensioniert werden, als Beispiel:

`100 DIM a$(12,34)`

50 DIM A\$(12)

100 LET A\$ = "ABCDEFGHIJKLMN"

150 LET A\$ = "*?!+"

200 LET A\$ = ""

DIM A\$(12) definiert beim Spectrum-BASIC eine String-Variable mit festgelegter Länge und dem Namen A\$. Sie enthält nach der Initialisierung zwölf Leerzeichen.

Wenn Sie versuchen, A\$ einen String von mehr als zwölf Zeichen zuzuordnen, werden die überzähligen Zeichen auf der rechten Seite ignoriert.

Wenn Sie A\$ einen kürzeren String zuordnen, wird er linksbündig in A\$ gespeichert. Die restlichen Zeichen von A\$ werden dabei nicht verändert.

Wenn Sie A\$ auf "", den sogenannten Null-String, setzen, sollten Sie die Länge auf 0 reduzieren, und der Inhalt sollte gleich "" sein. Bleibt seine Länge bei zwölf Zeichen, wird er mit Leerzeichen gefüllt.

`120 DIM b$(7,56)`

`140 DIM N$(56)`

`150 REM N$ wird als leerer String verwendet`

`580 IF b$(3)=N$(TO 56) THEN PRINT "Leer"`

`590 IF a$(11)=N$(TO 34) THEN PRINT "Leer"`

Hier wird N\$ nur als Leerstring eingesetzt. Würde man ihn nicht verwenden, so müßten die Test-Zeilen 580 und 590 so aussehen:

`580 IF b$(3)=" " THEN PRINT "Leer"`

`585 REM 56 Leerzeichen zwischen den Anführungsstrichen`

Diese Methode ist nicht sehr effizient und zudem fehleranfällig. Eine Alternative gegenüber N\$ wäre, alle Bereichs-Variablen mit einem Element mehr zu DIMensionieren als unbedingt notwendig. Dieses letzte Element kann dann als Leerstring zum Test des entsprechenden Bereiches verwendet werden. Zeile 590 sähe dann wie folgt aus:

`590 IF a$(11)=a$(12) THEN PRINT "Leer"`

Dabei wird davon ausgegangen, daß a\$(12) nie verwendet wird und deshalb nur Leerzeichen beinhalten kann.



Transistor als Verstärker

An jedem Transistor finden sich drei Anschlußdrähte, die mit den im Gehäuse verborgenen Elementen des Bauteils (Basis, Emitter und Kollektor) verbunden sind. Wenn die an der Basis angelegte Spannung wechselt, ändert sich der Widerstand der Kollektor-Emitter-Strecke im gleichen Takt. Mit einem vergleichsweise geringen Strom zur Basis lassen sich sehr große Ströme zwischen Kollektor und Emitter regeln – der Transistor verstärkt den kleinen Basis-Strom.

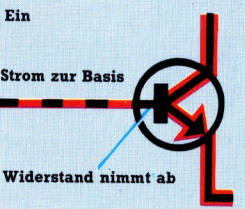
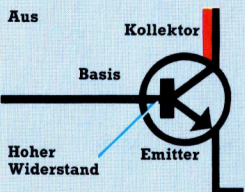
Kleines Eingangssignal



Großes Ausgangssignal

Transistor als Schalter

Bei einem bestimmten Sättigungspunkt führt auch ein weiteres Ansteigen des Basisstromes nicht mehr zu einem vergrößerten Fluß vom Kollektor zum Emitter. Ist der Basisstrom hingegen sehr klein, „sperrt“ der Transistor, das heißt, der Widerstand zwischen den beiden anderen Anschlüssen wird groß.



Transistor-Logik

In diesem Kursabschnitt werden einfache Logik-Schaltungen mit Transistoren aufgebaut.

Transistoren werden für zwei ganz unterschiedliche Funktionen gebraucht: Sie dienen entweder als Verstärker für ein elektrisches Signal oder zum Ein- und Ausschalten eines elektrischen Stromes. In Computern setzt man sie vorwiegend als Schalter ein. Eine Transistorschaltung läßt sich so aufbauen, daß sie bestimmte Muster von Ein- und Ausschaltzuständen speichert. Der Computer behandelt ein solches Muster wie eine Dualzahl. Logische „Gatter“ sind andersartige Schaltkreise, mit denen sich zwei bestimmte Ein/Aus-Muster beispielsweise addieren lassen.

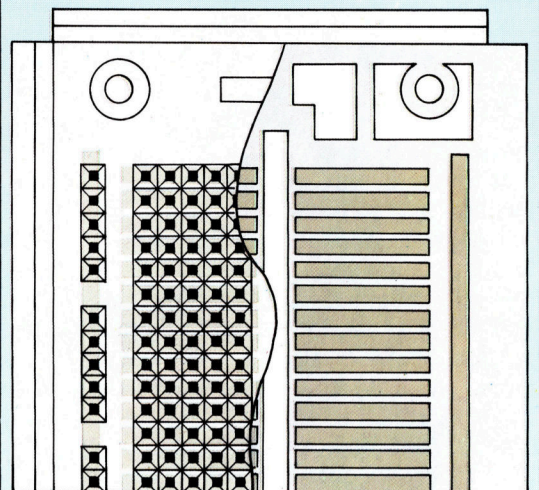
Beim Zusammenbau der auf der nächsten Seite beschriebenen Logikschaltungen wird Ihnen auffallen, daß ein aus Einzel-Transistoren hergestellter Computer nicht nur sehr groß, sondern auch extrem teuer wäre. Das war früher tatsächlich so – Computer zu heutigen Preisen wurden erst durch die Entwicklung von „ICs“ möglich. Ein Integrated Circuit besteht aus Hunderten von Transistoren, die gemeinsam auf einem winzigen, plastikummantelten Silizium-Chip sitzen. Für die meisten Anwendungen kann man die Chips in Standardausführung (TTL) einsetzen. Vier AND-Gatter in einem IC-Gehäuse kosten nur wenige Groschen. Die Chips für schwierigere Aufgaben, etwa Binärzähler, sind etwas teurer: Ab 2 Mark sind Sie dabei.

Die Methode, immer mehr Schaltkreise auf einem einzigen Chip zu vereinen, führt zum VLSI-Bauteil (Very Large Scale Integration). Damit sind sehr komplexe Rechen- und Steueroperationen möglich. VLSI-Chips bestehen aus Tausenden von Schaltungen in einem vergleichsweise winzigen Gehäuse. Der Microprozessor Ihres Computers ist ebenso ein

VLSI-Chip wie die ICs für die Bildschirmsteuerung, Tonerzeugung und die Schnittstellen-Controller. Das Arbeitsprinzip dieser Chips ist jedoch grundsätzlich das gleiche wie bei den drei nebenstehenden Logik-Schaltungen – nur die Anzahl der Schaltkreise und die Packungsdichte sind tausendfach höher.

Es gibt verschiedene Techniken, VLSI-Chips herzustellen. Sie variieren daher stark – sowohl im Preis als auch in der Arbeitsgeschwindigkeit und beim Stromverbrauch. In stationären Computern finden sich meist MOS-Chips (Metall-Oxid-Silizium), während man für batteriebetriebene Geräte CMOS-Typen (Complementary Metall-Oxid-Silizium) verwendet. Sie arbeiten langsamer, aber energiesparend.

Experimentier-Stecksocket



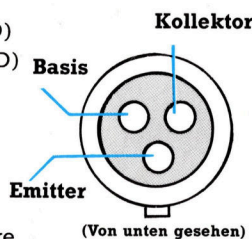
Oberfläche innenliegende Leiterbahnen

Mit „Breadboards“ können Sie eine Schaltung sehr schnell und ohne zeitraubende Lötarbeiten aufbauen. Dabei handelt es sich um wiederverwendbare Platinen, in die Bauteile nur eingesteckt werden. Die Anschlußdrähte werden von metallischen Krallen festgehalten. Jeweils fünf Anschlußpunkte sind miteinander elektrisch verbunden. Durch diese Aufteilung in Felder ist es recht einfach, eine Schaltung auf der Platine zu arrangieren. Mit kurzen Drahtstücken lassen sich die unterschiedlichen Anschlußfelder verbinden. Wählen Sie keinen zu kleinen Experimentiersocket!

Das benötigen Sie:

- 1 Experimentier-Stecksocket (z. B. Experimentor 300)
- 2 Transistoren BC 109
- 2 rote Leuchtdioden (LED)
- 1 grüne Leuchtdiode (LED)
- 3 Widerstände 500 Ohm
- 2 Widerstände 15 kOhm
- 2 Taster (Schließer)
- 1 Trockenbatterie, 9 Volt
- 1 Batterie-Anschlußclip
- ein paar kurze Drahtstücke

BC 109



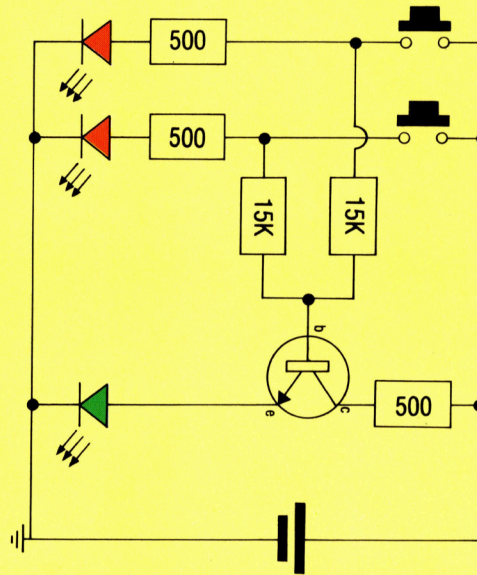
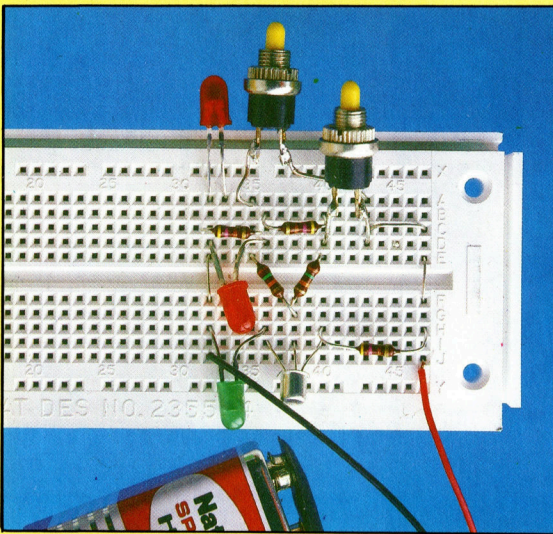
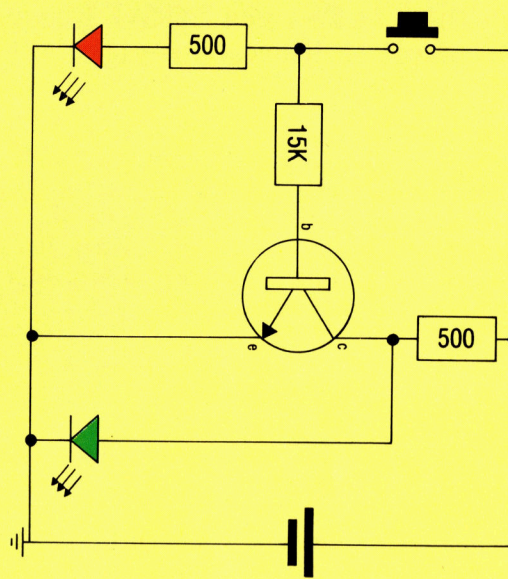
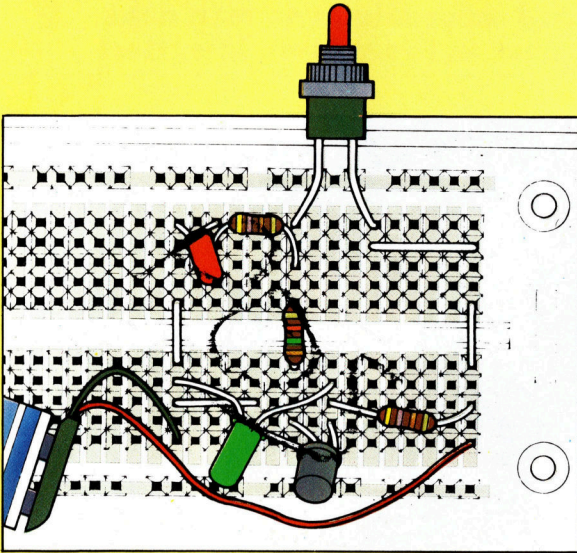
Aufbau logischer Gatter

Die hier abgebildeten Baupläne zeigen Schalt-Transistoren in „logischen Gattern“. Sie können die Schaltungen nacheinander aufbauen und dabei immer dieselben Bauteile und einen Experimentiersockel verwenden. Die Funktionen der hier vorgestellten Schaltungen sind „solid state“ (fest verdrahtet). Sie lassen sich nur durch eine andere Leitungsführung abwandeln. Zur Eingabe in die logi-

schen Gatter dienen Taster mit Leuchtdioden (LEDs). Anders in einem Computer: Hier ist der Eingang eines logischen Gatters meist mit dem Ausgang eines anderen verbunden. Wenn Sie mit den Gattern vertraut geworden sind, bekommen sie vielleicht Lust, einmal etwas Komplizierteres herzustellen – „füttern“ Sie doch einmal ein Gatter mit dem Ausgangssignal eines anderen!

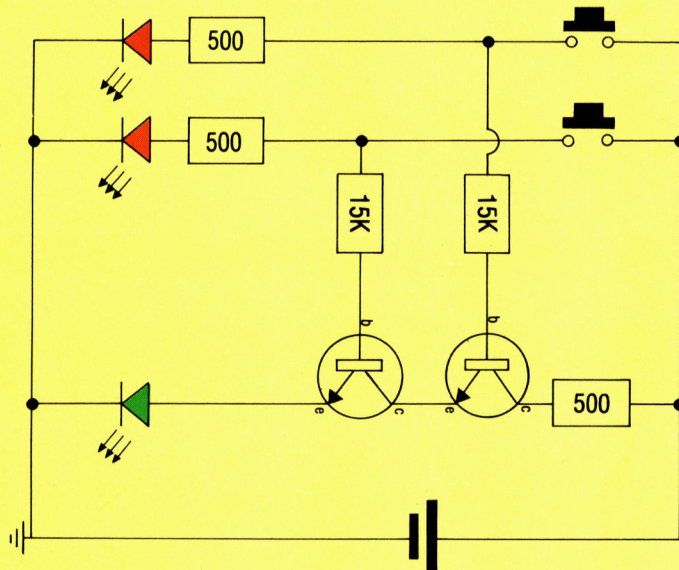
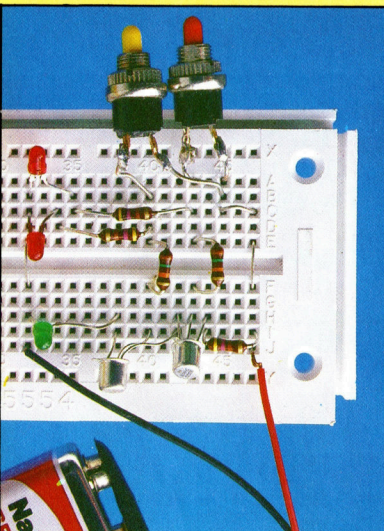
NOT-Gatter

Dies ist das einfachste aller logischen Gatter. Es hat nur einen einzigen Eingang (rote LED und Taster) und auch nur einen Ausgang (grüne LED). Bei geöffnetem Taster fließt kein Strom zur Basis des Transistors – also bleibt der Widerstand zwischen Kollektor und Emitter hoch, und der Strom nimmt seinen Weg durch die grüne LED. Ist also der Taster nicht gedrückt (Eingabe 0), leuchtet die grüne Diode. Bei gedrücktem Taster fließt ein Strom zur Basis des Transistors, der dadurch leitend wird (Widerstand zwischen Kollektor und Emitter niedrig). Jetzt nimmt der Strom seinen Weg durch den Transistor hindurch, ohne die LED zu passieren (leuchtet nicht).



OR-Gatter

Die Schaltung für das NOT-Gatter läßt sich recht einfach zu einem OR-Gatter umbauen. Zuerst wird die Ausgangs-LED so plaziert, daß sie von einem Strom durch den Transistor angesteuert wird. Das OR-Gatter hat zwei Eingänge, beide mit Taster und LED. Ist einer der beiden Taster gedrückt, fließt ein Strom zur Basis des Transistors. Er wird leitend, und die Diode strahlt Licht aus.



AND-Gatter

Für ein AND-Gatter werden zwei Transistoren hintereinander in die Ausgangsleitung geschaltet. Jeder Transistor bekommt aber einen eigenen Eingangstaster. Nur wenn beide Taster gedrückt sind (also beide Eingänge 1 sind), kann der Strom nacheinander die beiden Transistoren passieren und die Ausgangs-LED zum Leuchten bringen.



Neue Strukturen

In diesem Teil des LOGO-Kurses wird aufgezeigt, wie man neue Kontrollstrukturen entwickelt und diese in Prozeduren integriert.

Der LOGO-Befehl RUN hat eine Liste als Eingabe und bringt die Anweisungen in der Liste zur Ausführung. Mit Hilfe von RUN lassen sich neue Kontrollstrukturen bauen und bei Bedarf aufrufen. Die nächste Prozedur arbeitet mit diesem Befehl:

```
TO WHILE :BEDINGUNG :HANDLUNG
  IF NOT (RUN :BEDINGUNG) THEN STOP
  RUN :HANDLUNG
  WHILE :BEDINGUNG :HANDLUNG
END
```

Das folgende Programm berechnet die Potenzen der Eingaben unter 1000:

```
TO POTENZ :X
  MAKE "P :X
  WHILE [:P < 1000] [PRINT :P MAKE
    "P :P * :X]
END
```

Im Gegensatz zu anderen Programmiersprachen sind Kontrollstrukturen wie WHILE, REPEAT und FOR bei LOGO nicht unbedingt notwendig. Die POTENZ-Prozedur kann man auch so formulieren:

```
TO POTENZ :P
  IF NOT :P < 1000 THEN STOP
  PRINT :P
  POTENZ P * :P
END
```

In allen LOGO-Versionen gibt es die Anweisung REPEAT. Die nächste Prozedur zeigt eine Möglichkeit, wie Sie die REPEAT-Funktion selber definieren können.

```
TO REPT :NR :LIST
  IF :NR = 0 THEN STOP
  RUN :LIST
  REPT :NR - 1 :LIST
END
```

Der Befehl RUN bietet vor allem bei der weiterführenden LOGO-Programmierung vielseitige Einsatzmöglichkeiten. Um eine davon zu demonstrieren, wird zuerst ein Dreieck auf bekannte Weise definiert.

```
TO DREI
  FD 50 RT 120 FD 50 RT 120 FD 50 RT 120
END
```

Bei der Eingabe von PRINT TEXT "DREI erscheint folgendes Resultat:

```
[] [FD 50 RT 120 FD 50 RT 120 FD 50 RT 120]
```

Der Text der Prozedur wird als eine aus Listen bestehende Liste dargestellt, wobei jede „innere“ Liste eine Zeile der Prozedur repräsentiert. Das nächste Beispiel zeigt, warum bei der obenstehenden Ausgabe eine leere Liste vorgestellt wurde.

```
TO ADD :A :B
  PRINT :A + :B
END
```

Bei PRINT TEXT "ADD erscheint:
[:A :B] [PRINT :A + :B]

Es stellt sich heraus, daß die erste Liste die Eingaben enthält. Mit TEXT lassen sich also die Definitionen bestimmter Befehle ermitteln. Die Anweisung DEFINE dagegen ermöglicht es, eine Prozedur anhand von Listen zu erstellen, ohne den Editor-Mode aufrufen zu müssen. Geben Sie DEFINE "L [[:A] [FD :A] [RT 90] [FD :A / 2]] ein und anschließend L 30. Mit Hilfe dieser direkten Eingabe läßt sich innerhalb einer Prozedur eine zweite definieren.

Wir werden nun ein Programm entwickeln, das die Größe eines Objektes untersucht. Die wichtigsten Befehle sind FRAGE, der nach der Form des Gegenstandes fragt, und GROESSE, der die Größe des Objektes verändert. Zum Beispiel zeichnet FRAGE "QUADRAT ein Quadrat, und GROESSE [* 10] löscht das Quadrat und zeichnet ein neues, das jedoch um den Faktor 10 vergrößert ist.

Um das Programm so einfach wie möglich zu gestalten, müssen allerdings einige Einschränkungen in Kauf genommen werden. Aus diesem Grund sollte bei der FRAGE-Prozedur auf REPEAT sowie Unterroutinen verzichtet werden. Bei negativen Werten bricht die Programmausführung ab.

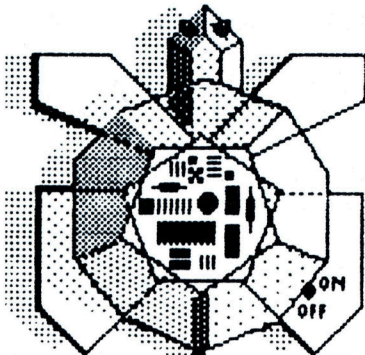
FRAGE weist zuerst das gewählte Objekt der globalen Variable "WAHL zu, ruft dann die Liste des Prozedurnamens auf und startet das Programm durch RUN.

```
TO FRAGE :OBJEKT
  HIDETURTLE
  MAKE "WAHL :OBJEKT
  RUN ( LIST :OBJEKT )
END
```



GROESSE löscht zu Beginn die Originalzeichnung (beim Commodore-LOGO lautet der Befehl zum Löschen PENCOLOR-1), ruft DEFINE auf, um das modifizierte Objekt zu definieren, und zeichnet dann die neue Figur.

```
TO GROESSE :OPLIST
  PENCOLOR-1
  RUN ( LIST :WAHL )
  DEFINE :WAHL MODI.PROZ TEXT :WAHL
  PENCOLOR 1
  RUN ( LIST :WAHL )
END
```



MODI.PROZ unterteilt den Text in Zeilen und übermittelt diese nacheinander an MODI.ZEILE:

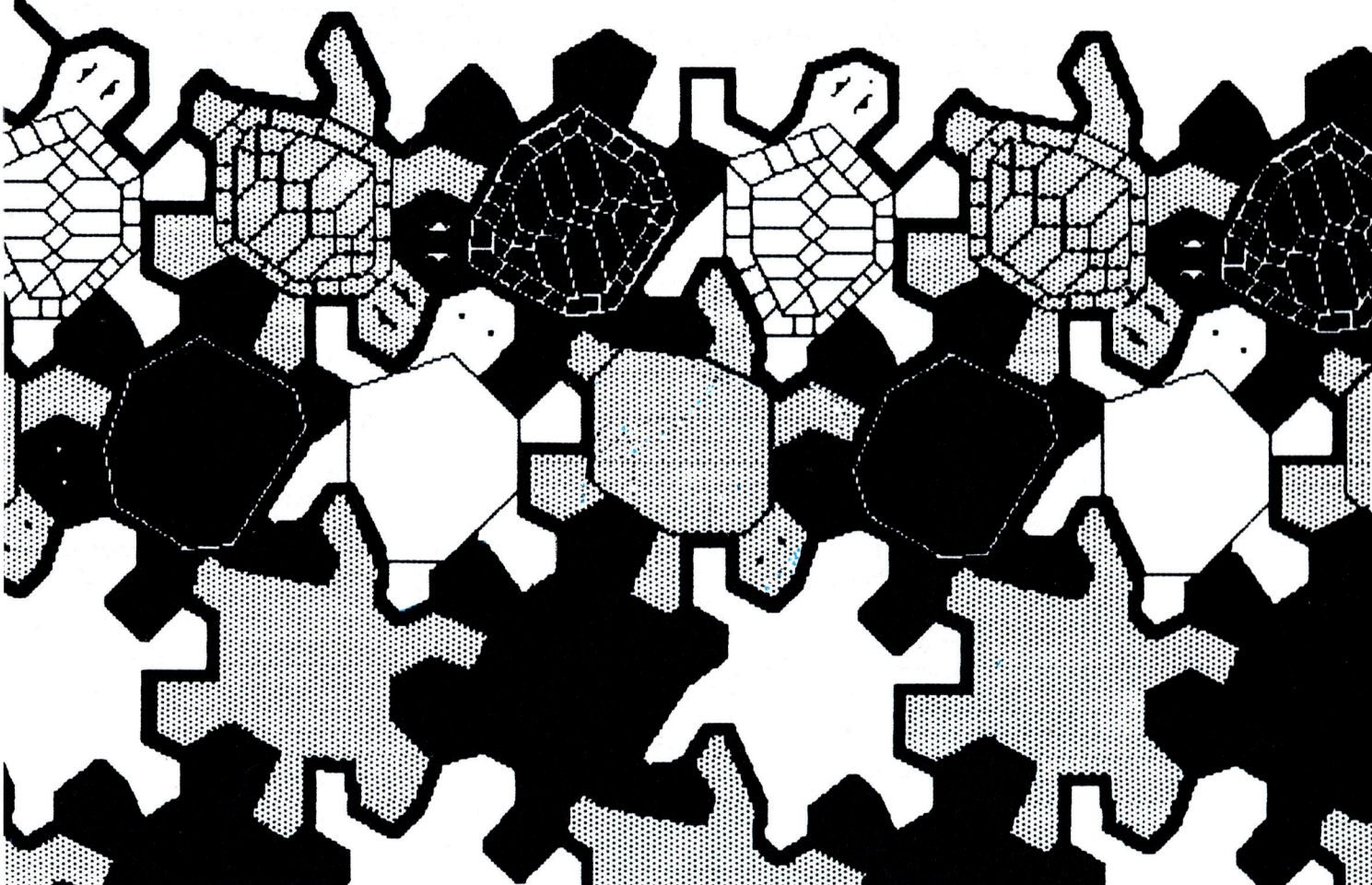
```
TO MODI.PROZ :TEXT
  IF EMPTY? :TEXT THEN OUTPUT []
  OUTPUT FPUT MODI.ZEILE FIRST :TEXT
  MODI.PROZ BUTFIRST :TEXT
END
```

MODI.ZEILE untersucht die Zeile nach einem FD oder FORWARD. Wurde dieser Befehl gefunden, wird der Rest der Zeile an AENDERN zur weiteren Bearbeitung übermittelt.

```
TO MODI.ZEILE :ZEILE
  IF EMPTY? :ZEILE THEN OUTPUT []
  IF ANYOF FIRST :ZEILE = "FD FIRST
  :ZEILE = "FORWARD THEN OUTPUT
  AENDERN BUTFIRST :ZEILE
  OUTPUT FPUT FIRST :ZEILE MODI.ZEILE
  BUTFIRST :ZEILE
END
```

Die Prozedur AENDERN beschäftigt sich mit der modifizierten Zeile. Das erste Element von LIST – die Eingabe zu AENDERN – könnte der FORWARD-Wert der Originalprozedur sein. Angenommen, dieser betrüge 50 und OPLIST (* 2), dann ergäbe SENTENCE FIRST :LIST :OPLIST die Formel (50 * 2). Anschließend ruft AENDERN den Befehl RUN auf, der die Be-

Wie Sie wissen, sind Recursionen, die in sich selbst definiert werden, bei LOGO unentbehrlich. Es wurden solche Beispiele gegeben: Prozeduren, die sich selbst aufrufen, Listen, die als Listen definiert werden, und schließlich Prozeduren, die Prozeduren schreiben. Mit etwas Phantasie ist es leicht, eine Zeichnung im Stil von MC Escher in LOGO zu schaffen, indem man eine Turtle dazu benutzt, eine Turtle zu generieren, die eine Turtle zeichnet...





LOGOs Vorteile

LOGO wird wie BASIC interpretiert. Im Gegensatz zu den meisten BASIC-Versionen ist es strukturiert und bietet fertige Prozeduren. LOGO ist wie FORTH erweiterbar, da man neue Worte definieren kann, die dann Bestandteil des Vokabulars werden. Es bietet wie LISP die Möglichkeit der Listenverarbeitung und eignet sich deshalb für die Erforschung solcher Bereiche wie „künstliche Intelligenz.“ LOGO wurde für die Verarbeitung komplexer Algorithmen entwickelt. Die meisten der vorgestellten Programme beginnen mit einer einfachen Prozedur, die nur einen Teil der Aufgabe durchführt. Danach wird die Prozedur auf verschiedene Weise geändert und verbessert. Am Ende dieses Prozesses steht ein perfekter Algorithmus zur Verfügung.

LOGOs Nachteile

Die Hauptprobleme liegen in der Begrenzung des Speicherplatzes und der langsamen Ausführungsgeschwindigkeit. Zusätzliche Möglichkeiten wären sinnvoll. So gibt es beim LOGO keine Fehlersuchmöglichkeiten, keine Datenfelder und Datenverwaltungen. Einige LOGO-Versionen bieten diese Möglichkeiten, aber eben nicht alle.

LOGO-Dialekte

Bei einigen LOGO-Versionen verwendet man

- NUMBERP statt NUMBER?
- LISTP statt LIST?
- WORDP statt WORD?
- EMPTYP statt EMPTY?

Im Spectrum-LOGO stehen sowohl COPYDEF als Befehl zur Verfügung, als auch PRIMITIVE? (entsprechend unserem PRIMITIVE?) und DEFINEDP (entsprechend unserem PROCEDURE?). Beim Atari-LOGO steht PC für PENCOLOR, SE für SENTENCE und HT für HIDETURTLE. DEFINE und TEXT gibt es hier nicht, allerdings werden im Handbuch Definitionsmöglichkeiten dafür gegeben. Die zur Verfügung stehenden Farben hängen vom Rechner ab. PC-1 wird hier zum Löschen von Zeilen genutzt. In einigen LOGO-Versionen wird statt dessen der Befehl PE verwendet.

rechnung ausführt. Zum Schluß wird eine Liste aus FD, dem neu berechneten Wert und dem Rest der modifizierten Zeile gebildet.

```
TO AENDERN :LIST
  OUTPUT (SENTENCE "FD (RUN
    SENTENCE FIRST :LIST :OPLIST)
    MODI:ZEILE BUTFIRST :LIST)
END
```

In einigen Fällen ist es sinnvoll, bestimmte Routinen zu kopieren. Mit der Prozedur COPYDEF und der Eingabe COPYDEF "NEU "ALT definieren Sie NEU als eine Kopie von ALT, wobei die Prozedur ALT weiterhin in der ursprünglichen Form erhalten bleibt.

```
TO COPYDEF :NEU :ALT
  DEFINE :NEU TEXT :ALT
END
```

Das Problem bei dieser Definition ist: Existiert ALT nicht, geht die Prozedur weiter und definiert NEU ohne Inhalt. Eine Lösung wäre das folgende Programm:

```
TO COPYDEF :NEU :ALT
  IF NOT PROCEDURE? ALT THEN (PRINT
    [ES GIBT KEINE PROZEDUR] :ALT)
  STOP
  DEFINE :NEU TEXT :ALT
END
```

Hier wird der Befehl PROCEDURE? verwendet, der TRUE ausgibt, wenn seine Eingabe eine Prozedur ist, andernfalls FALSE. PROCEDURE? und das Gegenstück PRIMITIVE? sind hilfreiche Tests, leider gibt es sie aber nicht in allen LOGO-Versionen. Deshalb haben wir Abwandlungen von PROCEDURE? und PRIMITIVE? entwickelt, die sowohl beim Apple- als auch beim Commodore-LOGO angewendet werden können.

```
TO PROCEDURE? :NAME
  IF NUMBER? :NAME THEN OUTPUT
    "FALSE
  IF LIST? :NAME THEN OUTPUT "FALSE
  TEST WORD? :NAME
  IF TRUE IF WORD? TEXT :NAME THEN
    OUTPUT "FALSE ELSE IF NOT (TEXT
      :NAME = []) THEN OUTPUT "TRUE
  OUTPUT "FALSE
END
```

```
TO PRIMITIVE? :NAME
  IF NUMBER? :NAME THEN OUTPUT
    "FALSE
  IF LIST? :NAME THEN OUTPUT "FALSE
  TEST WORD? :NAME
  IF TRUE IF WORD? TEXT :NAME THEN
    OUTPUT "TRUE ELSE OUTPUT
    "FALSE
END
```





Verkabelte Klänge

In diesem Artikel beschäftigen wir uns mit der MIDI-Hard- und Software, die dem Heimcomputerbesitzer zur Verfügung steht.

Der Micon oder auch MIDI controller wird von „XRI-Systems“ hergestellt und ist für die von kleineren Software-Häusern angebotenen MIDI-Programme typisch. Er wurde für den Sinclair Spectrum entwickelt und besteht aus einem Interface nebst dazugehöriger Software. Seine Sequenzer-Möglichkeiten reichen für acht verschiedene Spuren, die jeweils 2951 „Ereignisse“ (Noten, Pausen etc.) aufzeichnen können. Die Musikeingabe erfolgt über die angeschlossene Tastatur des Synthesizers. Mittels der Leertaste des Spectrum wird die Anzahl der zu spielenden Ereignisse definiert.

Bei Benutzung des Synthesizers „rollt“ die Musik in recht roher Form, dargestellt in herkömmlicher Notation – mit fünf Linien – über den Bildschirm. Die normalen Schlüssel (Baß- und Violin-), die Dauer (einschließlich punktierter Noten), Erhöhen und Erniedrigen von Notenwerten sowie Staccato werden ebenfalls dargestellt. Die Notation ist aber recht dürftig, etwa bei Pausen, wenn keine Notationsereignisse stattfinden. Dazu kommt, daß die Notenstriche immer nach oben zeigen, obwohl deren Richtung von ihrer Platzierung auf den Notenlinien abhängen sollte. Abfolgen kurzer Notenwerte werden nicht in Gruppen zusammengefaßt, die das tatsächliche musikalische Taktmaß wiedergeben.

Beim Editieren einer Komposition wird die Bedeutung der Bildschirmdarstellung deutlich. Die Musik ist in Takte unterteilt. Sobald eine Takt Nummer eingegeben wird, können einzelne Noten gelöscht, eingefügt oder verändert werden. Ganze Takte lassen sich herausnehmen und Taktgruppen innerhalb der Sequenz lassen sich beliebig wiederholen. Bis zu zehn Sequenzen – das entspricht fast 24 000 Noten – sind neben detaillierten musikalischen Anweisungen auf Band speicherbar. Das Wiedergabetempo kann man bis auf vier Millisekunden genau definieren oder wahlweise durch die Verwendung einer Rhythmusmaschine kontrollieren.

Die Sequenzierung bei Micon erfolgt in Echtzeit. Das bedeutet: Er „hört“ die Darbietung auf einem MIDI-kompatiblen Synthesizer und gibt alle Daten in den Speicher des Spectrum ein. Zunächst kann ein Keyboard-Musiker die Aufnahme seiner eigenen Darbietung hören, ohne sich gleichzeitig um die Aussteuerung bei der Aufnahme kümmern zu müssen, was auch für die musikalische Ausbildung interessant ist. Da ferner die dargebotene Musik „unter den Fingern“ liegt, wird das taktge-

rechte Sequenzieren erleichtert. Dennoch benötigt der Künstler einen Taktgeber (Metronom) bzw. den Takt einer Rhythmusmaschine, um ein gleichmäßiges Tempo zu halten. Wie bei jedem MIDI-System hängen die Ergebnisse des Micon vom angeschlossenen Synthesizer und seinen Möglichkeiten ab.

Das MIDI-Paket von Jellinghaus Music Systems wurde für mehrere Rechner konzipiert. Es ist für den Apple II, den Commodore 64 und den Spectrum erhältlich. Im Prinzip stimmt es mit dem Micon-Angebot überein: Musikalische Information wird via Keyboard eingegeben. Der Hauptunterschied besteht darin, daß die Sequenzen nur in Echtzeit eingegeben werden, und die Bildschirmdarstellung erfolgt nicht in der traditionellen Notation.

Kein 12/8-Takt

Die Taktaufteilung geschieht durch Spezifikation der Zeitwerte – also die Viertel- oder Achtelnoten pro Takt. Vier Zeitwerte stehen zur Verfügung: drei, vier oder fünf Viertel pro Takt (3/4, 4/4 oder 5/4 Takt) oder sieben Achtel (7/8). Weicht die Echtzeit-Darbietung vom Tempo ab, werden die Eingabedaten als Kette ungetakteter Noten erfaßt. Bei Wiedergabe normaler Notengruppen ist das unproblematisch. Sobald andere Werte als 3/4, 4/4 oder 5/4 bzw. 7/8 laufen, versagt der Sequenzer.

Klangkontrolle

Durch Nutzung der Soundmöglichkeiten des Commodore 64 kann ein BASIC-Programm geschrieben werden, das mittels Joystick eingegeben und dann gespielt wird. Im hier gezeigten Beispiel erscheint ein Lichtpunkt auf einem grafisch erzeugten Keyboard, das durch Joysticksteuerung bewegt werden kann. Ist die gewünschte Note erreicht, wählt man durch Joystickbewegung die Notendauer. Durch Druck auf den Feuerknopf wird der Notenwert gespeichert, und die nächste Note kann eingegeben werden.



1 Computer-Kontrolle
Durch Software lassen sich auf dem Commodore 64 Speicherverwaltung und menügesteuerter Betrieb des Musiksystems realisieren. Diskettenstation oder Cassettenrecorder werden zur digitalen Aufzeichnung von Melodien verwendet.

Das bedeutet auch, daß ein 12/8-Takt nicht eingegeben werden kann, selbst wenn der Rhythmus bzw. das Tempo stimmt. Das ist ein entscheidender Nachteil, da der 12/8-Takt häufig bei Jazz, Rock, Funk und Reggae verwendet wird.

Vorausgesetzt, eine Tonfolge wurde korrekt eingegeben, erfolgt die Datendarstellung auf dem Bildschirm in Tabellenform unter Angabe der Oktave, in der das Notenereignis stattfindet. Dazu werden die Höhe der Note innerhalb der Oktave, ihre Dauer (in Viertel- bzw. Achtelnoten) und die „Gate“-Zeit (ein Parameter, der die Phrasierung einer Note, d. h. ihre Betonung, ermöglicht) angegeben.

Diese Art der Darstellung gibt mehr Informationen, als es bei der traditionellen Darstellung des Micon der Fall ist, da dort weder „Gate“-Zeiten noch Tempo-Stufen angegeben werden. Die Möglichkeit der Spezifizierung solcher Parameter ist erst durch die Elektronik möglich geworden.

Allerdings haben viele Musiker auch ohne herkömmliche Ausbildung die Fähigkeit entwickelt, die Abfolge einer Standardnotation zu erkennen. Aus dem Notenanstieg bzw. -fall auf den Notenlinien läßt sich eine ungefähre Klangfolge bzw. Melodie erkennen. Ferner kann man daraus schließen, wie Notengruppen als Akkord klingen. Mit etwas Übung können solche Notationen gelesen werden.

Das Hauptproblem des musikinteressierten

Heimcomputer-Besitzers liegt weniger darin, das richtige MIDI-Paket zu finden. Viel entscheidender ist die Wahl des geeigneten MIDI-kompatiblen Synthesizers, der einerseits die entsprechenden Möglichkeiten bietet, andererseits aber nicht zu teuer sein sollte. Nehmen wir einmal an, wir benötigten einen Synthesizer, der etwa 50 Prozent der Backings spielen kann, wie sie bei einer Pop-Single-Produktion des Jahres 1984 Verwendung fanden, so kämen wir auf einen Preis zwischen 3200 und 6000 Mark. Synthesizer, die weniger kosten, sind in ihren Möglichkeiten extrem beschränkt.

Synthesizer-Konzepte

Eine interessante Variante ist, was als „Synthesizer-Konzept“ (bzw. „conceptual synthesizer“) bezeichnet wird. Typisch dafür ist ein ausgefeiltes Software-Paket, ergänzt um eine periphere Tastatur, die die Klangerzeugungsmöglichkeiten des Microcomputers nutzt.

Ein Beispiel dafür ist der von Clef Products hergestellte PDSC (programmable digital sound generator), der an den Acorn B angeschlossen werden kann. Zum Lieferumfang gehören ein anschlagesempfindliches Keyboard sowie zwei Pedale. Dieses „empfindliche“ Keyboard verdeutlicht den Unterschied zwischen phrasierungsloser oder „toter“ Darbietung und den ausdrucksvollen Daten, die sich aus dem



2 Sequential Circuits Six Trak

Dieser leistungsfähige Synthesizer ist mit einem Mehrspur-Recorder ausgestattet. Wie der Name schon sagt, kann auf sechs verschiedenen Spuren aufgezeichnet werden. Anders als vergleichbare Synthesizer kann beim Six Trak jede Stimme anders eingestellt werden, womit man komplexe Stücke sofort erzeugen kann. Der Synthi kostet etwa 3200 Mark.

3 Sequential Circuits Model 64

Durch Anschluß des Model 64 an den Erweiterungsport des Commodore 64 werden ein MIDI-kompatibler Synthesizer, der Speicher des Computers, Cassetten- oder Diskettenspeicher und Bildschirmausgabe zu einem kompletten Musiksystem verbunden. Bis zu 4000 Noten können gespeichert werden. Bei Wiedergabe sendet der Model 64 über das Interface entweder ein digitales Signal oder ein modifiziertes – ganz nach Wunsch. Model 64 kostet rund 750 Mark.



Echtzeitspiel ergeben und für Aufnahme wie Wiedergabe gespeichert werden können. Statt Oszillatoren finden 32 soundgenerierende Chips Anwendung, bei denen sich jeweils elf Charakteristika definieren lassen. Gegebenenfalls können alle 32 Einheiten zur Erzeugung einer einzigen Note verwendet werden. Damit verfügt das PDSG über weit mehr Möglichkeiten als jeder Synthesizer. Ein talentierter Anwender kann aus dem Vollen schöpfen.

Wird für die Erzeugung jeder Note ein Ton-generator verwendet, kann eine programmierte Sequenz aus bis zu 32 individuellen Werten bestehen. Alternativ lassen sich einige Generatoren zur Modifizierung des sequenzierten Materials verwenden, und die restlichen könnten in Echtzeit dazu spielen. Die Wellenform-Charakteristika werden auf dem Bildschirm dargestellt und geben so die Möglichkeit visueller Klanganalyse – eine wertvolle Ergänzung zur Hörkontrolle, die das PDSG ideal für die musikalische Ausbildung macht. Das Soundgenerierungs-Paket allein – passend für Nicht-Echtzeit-Einspielung, Wellenform-Erzeugung und Analyse – kostet etwa 800 Mark. Dazu kommt ein etwa ebenso hoher Preis für die Tastatur.

Ein Nachteil des PDSG ist allerdings die Klangumwandlung auf Digital-Analog-Basis. Der Mensch kann Klänge in einer Bandbreite von 20Hz bis 20kHz wahrnehmen. Natürliche Klänge, einschließlich der durch akustische

Musikinstrumente erzeugten, liegen in dieser Bandbreite und sogar darüber. Das PDSG dagegen vermag nur Töne bis zu 12 kHz darzustellen. Das Ergebnis entspricht also einem minderwertigen HiFi-System. Der Hersteller verweist aber darauf, daß durch Anschluß eines HiFi-Verstärkers und entsprechender Boxen eine Verbesserung bzw. Erweiterung möglich ist. Clef Products plant neue PDSG-Versionen, die auch mit anderen Microcomputern verbunden werden können.

MIDI wird als Durchbruch betrachtet, da der Microcomputer als vollwertiges Musikinstrument benutzt werden kann. Das PDSG-System ist weit entwickelt und ist eine echte Alternative zu Synthesizern.

Musiksysteme

Dank der MIDI-Interface-Entwicklung haben Microcomputer-Besitzer ungeahnte Möglichkeiten. Das birgt zugleich die Gefahr, viel Geld auszugeben – für das Interface, den Synthesizer und die erforderliche Software – , nur um festzustellen, daß der effektive Einsatz der Geräte zu kompliziert ist. Dazu kommt gegebenenfalls der Frust, wenn's nicht funktioniert. Es gibt jedoch preiswerte Alternativen, um sich mit den Grundlagen elektronischer Musik vertraut zu machen. Ein solches System muß natürlich musikalisch befriedigend sein und Interesse an weiteren Möglichkeiten wecken. Das Programm „Ultisynth 64“ von Quicksilva (auf Cassette) ist ein interessantes Anfänger-Programm, das die Möglichkeiten der C-64-SID-(sound interface device-)Chip und seiner drei Oszillatoren nutzt. Jede Taste des Commodore hat dabei eine unabhängige Kontrollfunktion für Klang-Generierung und Definition. Auch die vier Grundwellenformen – Sinus, Rechteck, Dreieck und Sägezahn – sind verfügbar. Ferner lassen sich Attack, Sustain, Decay und Release definieren. Die Klänge können gefiltert werden (eine bestimmte Bandbreite von Frequenzen kann bei der Ausgabe weggenommen werden). So ist der Klang detaillierter formbar. Ringmodulation etwa für glockenähnliche Töne ist ebenfalls möglich. Insgesamt können 2048 Noten festgehalten werden.

Die Möglichkeiten des Ultisynth stimmen weitgehend mit dem VCS3-Synthesizer überein, einem spannungskontrollierten Synthesizer, der Ende der sechziger bis Anfang der siebziger Jahre Verwendung fand. Ultisynth kostet etwa 50 Mark, wogegen der VCS3 heute eher Museumswert hat. Ebenfalls für Anfänger interessant ist Romiks „Multisound“, das dem Ultisynth von den Möglichkeiten her ähnlich ist, jedoch außerdem eine grafische Darstellung eines Keyboards beinhaltet.

Bewegter Arm

Immer mehr preiswerte Roboter sind nach und nach auch für den Heimcomputer-Markt erhältlich. Beasty ist einer der ersten.

Beasty wird als Bausatz geliefert und kann vom Anwender mit zwei Schraubendrehern zusammengebaut werden. Mit dem Bausatz wird Software (auf Cassette) geliefert, auf der das Robol-Betriebssystem enthalten ist, mit dem der Roboter-Arm gesteuert wird.

Zwei Handbücher werden mitgeliefert. Die Bauanleitung enthält einen umfangreichen Teil zur Geschichte der Robotik und kommt erst spät auf die eigentlichen Konstruktions-Details. Für den Anfänger mag die Vielzahl von Teilen und die stellenweise etwas komplizierte Bauanweisung etwas entmutigend sein. Dennoch wird es jedem gelingen, den Arm zusammenzubauen, wenngleich es etwas Zeit in Anspruch nehmen wird.

Der „Körperbau“

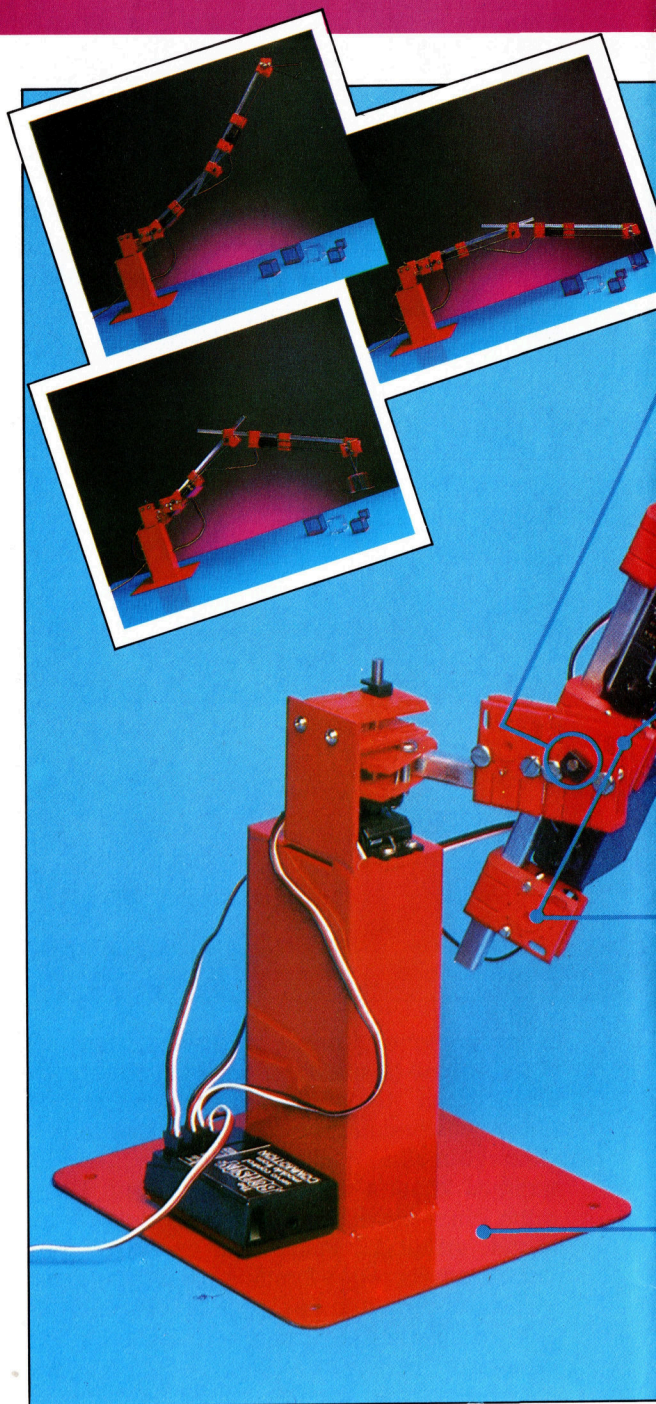
Beasty besteht aus einem Sockel und einem darin befindlichen Gelenk, das seitliche Bewegungen erlaubt. Mit dem Gelenk ist ein kurzer Aluminiumstutzen verbunden, um den sich über ein zweites Gelenk der „Oberarm“ bewegt. Der „Unterarm“ ist durch ein drittes Gelenk bewegbar. Die Gelenke werden durch Servomotoren betrieben. Diese bewegen kurze Drahtstücke, die mit dem „Skelett“ des Arms verbunden sind. Durch Drehung einer Übertragungsscheibe wird ein Draht verkürzt, während der andere Motor die andere Scheibe in entgegengesetzter Richtung dreht. So erfolgen Gelenk- und Armbewegung. Ein Servomotor arbeitet, indem er digitale Schwingungen in Bewegungen umsetzt. Der Motor empfängt eine Reihe von Schwingungen in einer bestimmten Frequenz. Diese Schwingungen werden vom Prozessor als Bewegungswinkel interpretiert. Solange die Frequenz konstant bleibt, hält der Motor den Arm in der erreichten Position. Ein Schwingungswechsel der Frequenz vermittelt dem Prozessor einen neuen Winkel, der Arm wird bewegt.

Die beim Beasty verwendeten FP-128-Servomotoren erzeugen eine maximale Zugkraft von 3,5 kg pro Zentimeter. Das bedeutet: Über eine Strecke von einem Zentimeter kann ein Servomotor ein Gewicht von 3,5 kg heben, über eine Strecke von zehn Zentimetern jedoch nur ein

Gewicht von 350 Gramm.

Der Servo-Prozessor befindet sich in einem kleinen schwarzen Gehäuse. Dieses Gehäuse hat Anschlußbuchsen für insgesamt vier Servomotoren. Die vierte Buchse ist für einen Options-Motor gedacht, der zum Betrieb eines „Greifers“ oder ähnlichen Instruments am Ende des Unterarms verwendet werden kann. Ferner gibt es einen Stecker, der als Schnittstelle zum Userport des Acorn B dient, sowie eine Stromzuführung, die in die Hilfsstrombuchse des Computers paßt.

Nach dem Laden des Programms erfolgt eine Meldung auf dem Bildschirm, die den Anwender darauf hinweist, daß er sich im Edit-Mode befindet. Eine Programmzeile in Robol besteht aus der Zeilennummer, einem Befehl und einer Reihe von Zahlen, die jeweils mit





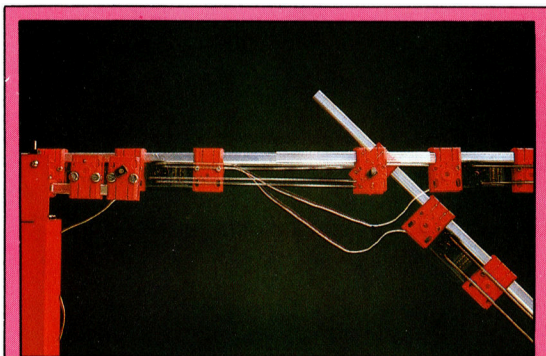
Drehpunkte
Sie entsprechen den Gelenken des menschlichen Arms.

Verbindungsdrähte
Sie verbinden die Servomotoren mit den Drehpunkten. Wird ein Draht verkürzt, verlängert sich der andere in entgegengesetzter Richtung.

Servo-Motoren
Mittels der Motoren wird der Freiheitsgrad der verschiedenen Achsen verändert.

Greifer (als Option)
Nur einer dieser Greifer ist mit dem Motor verbunden. Der Zugriff erfolgt durch Bewegung des freien Greifers gegen den starren.

Grundplatte
Die Basis des Armes besteht aus einer Metallplatte. Beim Heben von Gegenständen ist jedoch Vorsicht geboten, da Beasty leicht umkippt.



Servo-Motoren
Beasty wird mit drei Servo-Motoren geliefert, die mit einer Schnittstelle verbunden sind. Servo-Motoren finden häufig in der Robotik Anwendung, da sie entsprechend der Frequenz eines digitalen Signals eine gleichbleibende Kraft ausüben.

einer der vier Servo-Motor-Optionen korrespondieren. Enthält die Zeile den Befehl MOVE, entsprechen die Zahlen der Schwingungsfrequenz, die die Servos in der gegenwärtigen Position halten. Diese Werte können durch den Anwender verändert werden. Solange man sich im Edit-Modus befindet, wird sich der Servo-Motor, bedingt durch die Korrekturen, bewegen. So ist es möglich, den Arm nach den Wünschen des Anwenders zu positionieren. Hat man die gewünschte Position erreicht, drückt man RETURN. Daraufhin erfolgt die Darstellung einer neuen Robol-Zeile, und die nächste Bewegungsabfolge kann programmiert werden.

Nach Drücken der F0-Funktionstaste kann der Arm eine komplette Bewegungssequenz ausführen. Durch Drücken von F1 gefolgt von F0 (womit das Programm auf den Start zurückgesetzt wird) kann das Programm von jeder Zeile gestartet werden. – Ebenfalls dadurch, daß die derzeitige Zeilennummer durch Drücken der Cursor-Steuerungstasten verändert wird. Als Abschluß wird die Bewegungssequenz automatisch wiederholt. Zum Stoppen des Programms muß der Befehl MOVE in STOP verändert werden.

Verzögerungen

Bei der Ausführung von MOVE-Anweisungen läßt sich der Arm durch Einfügen eines WAIT-Befehls, gefolgt von einer Zahl, stoppen. Dies geschieht mit Hilfe des TIMER-1-Pins des Userports, über den eine Unterbrechung erzeugt wird. Da der Timer in Hundertstelsekunden arbeitet, hat WAIT 100 eine Verzögerung von einer Sekunde zur Folge, bevor die nächste Anweisung ausgeführt wird.

Die Armbewegung kann durch Veränderung des Befehls MOVE in JUMP erheblich beschleunigt werden. Zwei Zeitablaufbefehle sind ebenfalls möglich – JDELAY und MDELAY. Beasty ist mit einer Verzögerungsschaltung ausgestattet, die vor Ausführung jeder Programmzeile wirksam wird. Die Verzögerung beträgt 20, was etwa 1/5 Sekunde entspricht. Der Wert kann jedoch durch JDELAY (bei den JUMP-Befehlen) und MDELAY (bei MOVE-Anweisungen) geändert werden.

Der Umgang mit dem Robol-Betriebssystem ist einfach. Ebensoleicht läßt sich der Arm so programmieren, daß er komplexe Bewegungen ausführen kann. Das Betriebshandbuch ist einfach, doch perfekt, wengleich fortgeschrittene Programmierer es unbefriedigend finden mögen. Mit Hilfe des Driver-Programms kann Beasty durch BASIC gesteuert werden.

Commotion hat die Software so ausgestattet, daß Backups von Robol gemacht werden können. Da aber fast alle Acorn-Diskettenstationen an die Hilfsstrombuchse des Acorn B anzuschließen sind, lassen sich Beasty und Diskettenstation nicht gleichzeitig einsetzen.

Beasty

PREIS

etwa 450 Mark

SCHNITTSTELLEN

Zum Userport und zur Hilfsstromversorgung des Acorn B.

SOFTWARE

Robol und Betriebssystem auf Cassette.

DOKUMENTATION

Beasty wird mit Bauanleitung und einem Programmierhandbuch geliefert.

STÄRKEN

Leicht verständlich, preiswert.

SCHWÄCHEN

Bedienungsanleitung mangelhaft, keine praktischen Nutz-anwendungen für den Arm.



Tops auf Band

Virgin Games ist eine Tochtergesellschaft der Plattenfirma Virgin Records. Die Verbindungen zwischen der Promotion für Pop-Musik und Computer-Spiele liegen auf der Hand: Software-Hitparaden sind ebenso bedeutend wie die Top Forty.

Als der Markt für Software Anfang der Siebziger Jahre noch eine nicht rechenbare Größe war, bot er jugendlichen Unternehmern reichlich Möglichkeiten, mit Programmen auf Cassetten Geld zu machen. Wer immer witzige Unterhaltungsprogramme in BASIC schreiben konnte und Zugang zu schnellen Audiocassetten-Kopiermaschinen hatte, stieg mit kleinen Anzeigen ins Software-Versandgeschäft ein – und hatte auch Erfolg.

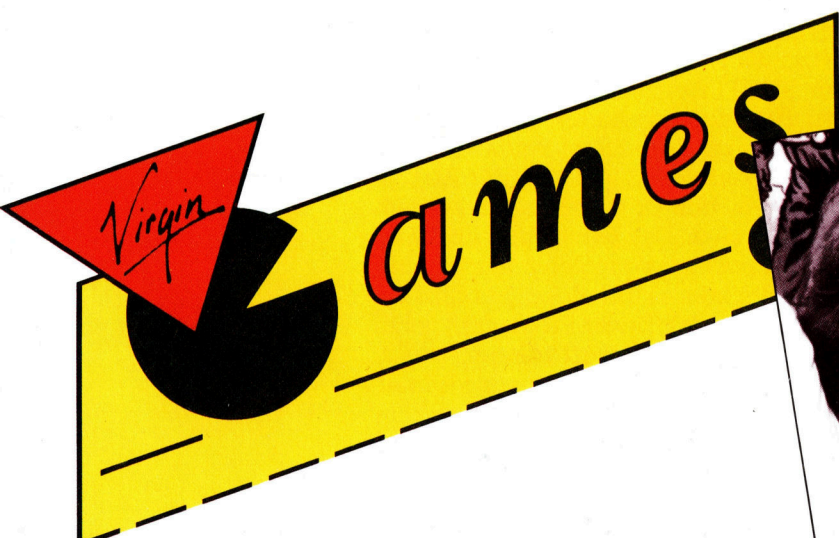
Heute sind die Dinge nicht mehr so einfach. Wie gut ein Programmierer auch sein mag – Umsatz setzt Originalität und Kreativität in der Programmgestaltung voraus. Die verschiedenen Software-Häuser beziehen ihre „Ware“ aus verschiedenen Quellen. Imagine-Software z. B. beschäftigte viele hauseigene Programmierer, die die Ideen ihrer Brötchengeber umsetzten.

Nick Alexander, der 28jährige Boß von Virgin Games, sagt: „Oft gilt: Je besser der Programmierer, desto weniger Ideen hat er. Ein guter Programmierer zu sein, setzt voraus, daß man sehr logisch, sehr methodisch, sehr sorg-

fältig an Dinge herangeht. Und mit diesen Qualitäten sind sogenannte Kreative üblicherweise nicht ausgestattet.“ Aus diesem Grunde sind nur sehr wenige Leute bei Virgin als Programmierer direkt angestellt. Man arbeitet auf der Basis, daß intern lediglich die kleinen Fehler ausgemerzt werden, die in den vielen, wöchentlich eingehenden Programmen hoffnungsvoller Jungprogrammierer enthalten sind, und dazu einen technischen wie kreativen Service parat hat. Talentierten Programmierern wird bei der Entwicklung von Ideen geholfen. Und kreative Leute bekommen technische Unterstützung.

Das Salär für eine Veröffentlichung mag auf den ersten Blick bei Virgin geringer ausfallen als bei anderen Gesellschaften. Pro Spiel zahlt Virgin zwischen 1000 und 3000 englische Pfund an den Autor, zu verrechnen gegen eine Umsatzbeteiligung von 7,5 Prozent vom Nettopreis. Zum Vergleich: Andere Softwarehäuser bieten im Schnitt 25 Prozent Umsatzbeteiligung. Doch Alexander argumentiert, daß die Verkäufe bei Virgin (und damit auch das Einkommen des Autors) durch die umfangreiche Werbung ungleich größer sind.

Virgin machte sich im Musikgeschäft durch erfolgreiche Politik und entsprechendes Auftreten einen Namen (und Umsatz). Was der 31jährige Virgin-Boß Richard Branson dabei gelernt hat, wirkt sich auch im Software-Geschäft aus. Spielautoren werden wie Stars vorgestellt: Die Cassettexte zeigen nicht nur den Namen des Programmierers, sondern enthalten zusätzlich sein Foto und eine Mini-Biographie. Seit dem Start von Virgin Games im Februar 1983 (damals liefen in Heimcomputer-Magazinen die ersten Anzeigen), kamen 500 Einsendungen ins Haus. Daraus wurden bis heute 46 Titel ausgewählt, die Katalogbestandteile sind. Mit diesem Angebot sind acht Computersysteme abgedeckt. Die besten Verkäufe erzielten bislang die Spectrum-Programme. Auf Platz zwei liegt der Acorn B, gefolgt vom Commodore 64.



Nick Alexander



Richard Branson



Fachwörter von A bis Z

Carrier Tone = Trägerfrequenz

Zum Datenaustausch über das Fernsprechnet ist ein Akustikkoppler oder ein Modem erforderlich, weil Telefonleitungen nur einen bestimmten Frequenzbereich übertragen. Wenn Verbindung zwischen zwei Rechnern besteht, liegt ständig die „Trägerfrequenz“ auf der Leitung. Diese ist als hoher Ton hörbar.

An den meisten Modems und Akustikkopplern befindet sich ein LED mit der Bezeichnung „Carrier Detect“, die erlischt, wenn die Trägerfrequenz unterbrochen wird. Die Trägerfrequenz ist mitten ins Fernsprech-Übertragungsband gelegt; die Daten werden nach einem bestimmten Schema moduliert. Eine „1“ könnte z. B. durch einen etwas über, eine „0“ durch einen etwas unter dem Träger liegenden Ton darge-



stellt werden. Das Modulationschema bestimmt die maximale Übertragungsrates – bei den meisten Modems nicht mehr als 1200 Baud, bei einigen Geräten aber auch bis zu 9600 Baud.

In dem Film „Wargames“ dringt ein Computer-Freak in das militärische Rechnernetz der USA ein, wobei er mit einem Programm Computer ausfindig macht, die telefonisch anzuzapfen sind. Das Programm, das er entwickelt hat, wählt einfach eine Nummer nach der anderen in einem Ortsnetz. Sobald eine Trägerfrequenz übermittelt wird, speichert das Programm die Nummer auf Diskette ab. Die meisten Datenbanken sind jedoch vor unrechtmäßiger Abfrage geschützt.

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

Carry = Übertrag

Ohne „Übertrag“ von einer Stelle zur benachbarten funktioniert das Rechnen in Stellenwertsystemen nicht – auch Ihr Computer macht beim Rechnen im Binärsystem davon Gebrauch. Eine einfache Halbaddierer-Schaltung, die zwei einstellige Binärzahlen addiert, hat z. B. außer dem Summen- noch einen Übertrags-Ausgang. Ein Volladdierer hat noch einen dritten Eingang für etwaige Überträge aus der nächstniedrigeren Ziffernstelle. Acht gekoppelte Volladdierer können zwei Acht-Bitzzahlen addieren, u. U. mit neunstelligem Ergebnis. Das Rechenwerk (ALU = Arithmetic and Logical Unit) in der Zentraleinheit (CPU) Ihres Computers hat in einem solchen Fall dafür zu sorgen, daß das „Carry Bit“ in einem der internen Register als Überlaufanzeige gesetzt wird.

CCD = CCD

Charge-Coupled Devices (Ladungsgekoppelte Bauelemente) ersetzen zunehmend die klassischen Bildröhren in Fernsehkameras, die Computern als optische Eingabegeräte eine Art „Gesichtssinn“ verleihen können. Dies ist besonders für den Bereich der Automatisierung interessant, wo man erreichen möchte, daß Industrieroboter Bauteile identifizieren.

Verglichen mit konventionellen Fernsehkameras ist ein CCD-System kleiner, leichter und in großen Stückzahlen billiger. Als Bildempfänger dient dabei ein Raster aus z. B. 256x256 winzigen Ladungsspeichern, die auf einem Chip integriert sind.

Darauf wird mit der üblichen Optik das Bild projiziert, nachdem jedes Speicherelement aufgeladen wurde (z. B. auf die „1“). Die Zellen entladen sich nun entsprechend der örtlichen Bildhelligkeit, und das Bild kann durch Ladungsübertragung ausgelesen werden – bei digitaler Auswertung wären jetzt dunkle Bereiche durch „1“, helle durch „0“ gekennzeichnet. Die Lage der Hell/Dunkel-Schwelle, die die Zuordnung eines Bildteils zur 1 oder 0 bestimmt, ist von der Belichtungszeit abhängig.

Cell = Tabellenfeld

Als Felder bezeichnet man bei der Tabellenkalkulation die einzelnen Sektoren im Schnittpunkt von Zeilen und Spalten. Jedes Feld ist über die Zeilen- und Spaltenkoordinaten individuell adressierbar und kann nach Wahl mit Text oder Zahlenwerten belegt werden, aber auch mit einem arithmetischen Ausdruck wie „84+82“, wobei der Inhalt des Feldes durch den zweier anderer definiert wird.

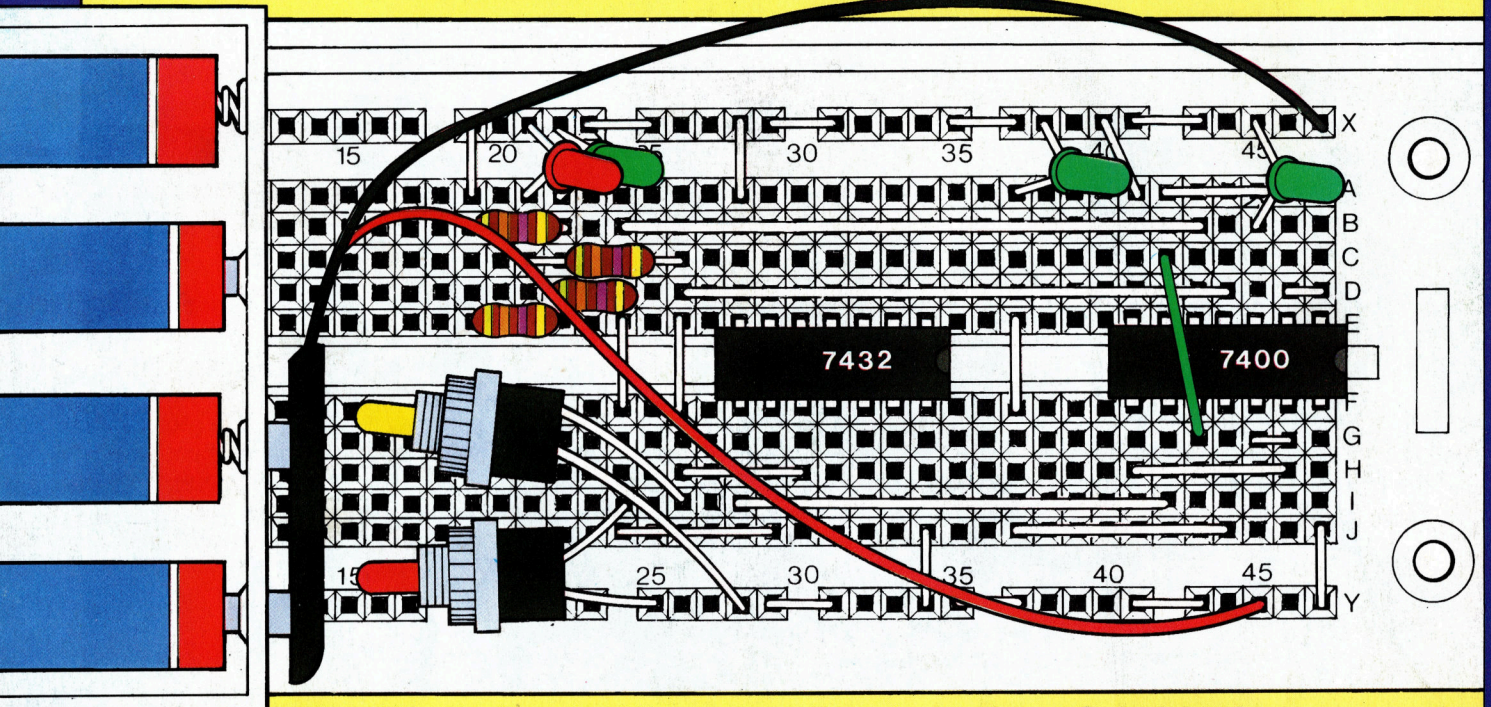
Centronics = Centronics

Die amerikanische Firma Centronics hat für ihre Drucker eine Parallel-Schnittstelle eingeführt, die sich in der ganzen Branche durchgesetzt hat. Auf acht Leitungen werden die zu einem Byte gehörigen Bits gleichzeitig übertragen. Der Centronics-Standardstecker ist 36-polig, es sind aber auch kleinere Stecker verwendbar. Geräte mit Centronics-Schnittstelle sind problemlos zu koppeln. Die Schnittstelle ist jedoch nicht bidirektional, d. h. nicht für Geräte geeignet, die zugleich Sender und Empfänger sind wie z. B. Modems.

Bildnachweis

673: Ian McKinnell, Teletape Video Ltd.
674, 680, 685, 689, 693, 696, 697:
Ian McKinnell
675, 686, 687: Steve Cross
678, 679: Ian McKinnell, CL Projects
681, 682, 690: Kevin Jones
684: Dave Cooper-Smith
691: Ian McKinnell, Kevin Jones
695: Liz Heaney
698, 699: Ian McKinnell, Commotion Ltd.

computer kurs Heft 26



Tips für die Praxis: Versuchsaufbau integrierte Schaltkreise



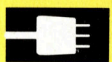
Hardware: Tatumg Einstein

Dieses Gerät ist serienmäßig mit einer Disketten-Station ausgerüstet. Betriebssystem und Basic werden von der Diskette geladen – der Speicherplatz des ROM steht bei anderem Einsatz vollständig zur Verfügung.



Maschinencode

Programme, die im Assemblerformat geschrieben werden, müssen am Anfang bestimmte Anweisungen enthalten. Diese Folge zeigt, welche Funktionen sie ausführen.



Peripherie: Commodore Music Maker

Um mit dem C 64 Musik zu machen, empfiehlt sich das Zubehör „Music Maker“.



Software: Entschlüsseln

Eine der ersten Aufgaben für Computer war das Entschlüsseln von Geheimcodes. Auch mit BASIC läßt sich eine Geheimsprache entwickeln.



Sehende Roboter

Maschinen mit „Augen“ zu versehen, ist prinzipiell nicht schwer. Probleme bereitet dagegen die Verarbeitung des „Gesehenen“.

