

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Heft 12

Roboter für zu Hause

BASIC-Kurs: Datenverwaltung

Zeichen generieren

Zubehör für den ZX 81

Verkehrscomputer

NEU: Der Sinclair QL

Ein wöchentliches Sammelwerk

computer kurs

Heft 12

Inhalt

Computer Welt

Verkehrscomputer 309

Rechnergestützte Transport- und Fahrzeugkontrolle

Präzise Greifarme 328

Roboter für Heimcomputer

Herman Hollerith 330

Software

Schwarzes Brett 312

Kommunikation per Computer und Telefon

Fernspiele 314

Hardware

Sinclair QL 315

Tips für die Praxis

Leichte Muse und Grundmuster 318

VC 20-Grafik und Spectrum-Sound

Peripherie

Dem DOS auf der Spur 320

Die Aufgaben eines Diskettenverwaltungssystems

ZX 81-Zubehör 322

BASIC 12

Datenverwaltung 324

Computer-Adreßbuch selbst programmiert

LOGO 12

Turtle-Variationen 331

Interessante Grafik-Puzzles

Bits und Bytes

Ändern und Löschen 334

Zeichengenerierung 335

Wie Buchstaben, Symbole und Ziffern entstehen

Schritt für Schritt 336

Gray-Code contra Binärsystem

Fachwörter auf einen Blick

WIE SIE JEDE WOCHE IHR HEFT BEKOMMEN

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

Zurückliegende Hefte

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

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

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

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

Abonnement

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

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

SAMMELORDNER

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

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

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

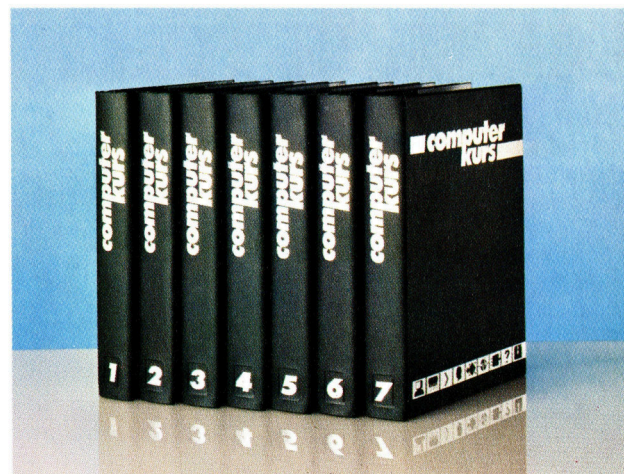
Schweiz: Der Sammelordner kostet sfr 15. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Sammelordner Computer Kurs, und notieren Sie Ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

INHALTSVERZEICHNIS

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

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

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





Ebenso wie bei vielen Flughäfen in den Vereinigten Staaten gibt es beim Gatwick Airport südlich von London ein beachtenswertes Transportsystem zwischen den Terminals, das die Richtungsstabilität und die Fähigkeit zu einem führerlosen, automatischen Betrieb eines Zuges mit dem Komfort und den Annehmlichkeiten eines Busses oder Taxis verbindet. Der People Mover wurde von der Westinghouse Corporation entworfen und gebaut, die durch ihre Zug-Signalsysteme bekannt ist. Der People Mover kann bis zu 100 Personen zur gleichen Zeit transportieren.

Verkehrscomputer

Der Transport von Gütern und die Personenbeförderung machen etwa 20 Prozent des Welthandelsvolumens aus. Computer sorgen für einen reibungslosen und fehlerfreien Ablauf.

Noch vor einhundertfünfzig Jahren benötigte man drei Monate, um von Europa nach Australien zu reisen. Heute kann diese Strecke in weniger als einem halben Tag zurückgelegt werden. Ohne die hochentwickelte, computergestützte Fahrzeugkontrolle wäre dies unmöglich.

Die höchsten Anforderungen werden dabei an den Flugverkehr gestellt. Auf dem Flughafen Heathrow bei London beispielsweise werden täglich mehr als 1000 Flüge pro Tag abgefertigt, in Spitzenzeiten sogar bis zu 120 pro Stunde. Ohne koordinierenden Computereinsatz könnte dieses System nicht reibungslos funktionieren. Eine genauere Betrachtung zeigt dies auf anschauliche Weise: Moderne Passagierflugzeuge sind in der Anschaffung und Unterhaltung sehr teuer. Die Airlines sind verständlicherweise bemüht, einen einwandfreien Zustand der Jets so lange wie möglich zu erhalten. Der Schlüssel hierzu liegt in der geregelten Wartung der Maschinen. Nach einer festgelegten Anzahl von Betriebsstunden kehrt das Flugzeug zu dem Flughafen zurück, an dem das technische Personal Zugang zu computergestützten Aufzeichnungen über das Flugzeug hat (beginnend mit dem ersten Konstruktionstag bis hin zur Seriennummer jedes Einzelteils). Die Computerberichte enthalten ferner genaue Aufzeichnungen über alle Re-

paraturen, das von Flugingenieuren und anderen Crewmitgliedern beobachtete Flugverhalten, den Treibstoffverbrauch und alle andere Daten, die von Interesse sein könnten.

Nur wenn der Wartungsplan erfüllt und auf den neuesten Stand gebracht ist, wird das Flugzeug wieder eingesetzt. Es wird dann Teil eines anderen Computersystems, der Einsatzplanung der Luftverkehrsgesellschaften. Hier werden die Maschinen auf die Flugrouten verteilt, der Treibstoff für die Zwischenlandungen geordert, die Crewwechsel vorbereitet oder die Bordverpflegung bestellt.

Flughafen-Kontrollsystem

Ein weiteres Kontrollsystem mit Computerunterstützung findet auf Flughäfen Verwendung, wo die Angestellten mit ihren oft begrenzten Mitteln die Forderungen seitens der Fluglinien bewältigen müssen. Selbst wenige Minuten Verspätung können den Verlust einer großen Menge Geldes bedeuten. Nur mit Hilfe computerisierter Fahrpläne ist es möglich, diese Aufgabe zu bewältigen. Das Flughafen-Computersystem achtet auch auf das rechtzeitige Einchecken der Passagiere und steuert die Anzeigetafeln mit den Flugzeiten.

Bevor die Fluggäste überhaupt am Flughafen eintreffen, hat der Pilot schon einen detail-



lierten Flugplan mit der Flugüberwachung aufgestellt. Überraschenderweise ist die Flugüberwachung aber nur zum Teil computergestützt. Dank eines Radar-Transponder-Systems müssen sich die Fluglotsen nicht länger darauf verlassen, die Position vom Piloten mündlich übermittelt zu bekommen. Jeder Punkt, den sie auf dem Radarschirm sehen, wird durch eine Flugnummer, sowie durch die computererrechneten Höhe- und Ziel-Codes des Bestimmungsortes identifiziert, der vom Flugzeug automatisch gesendet wird. Zusätzliche Computerunterstützung erhält der Fluglotse in Form von Ausdrucken, die jeweils einen Teil des vom Piloten aufgestellten Flugplanes darstellen. Diese Ausdrücke, die Angaben über Flugweg und -höhe, Fracht sowie den Flugzeugtyp enthalten, helfen dem Lotsen, den Flug durch sein Kontrollgebiet auf dem schnellsten und kostengünstigsten Wege zu leiten.

Ein anderes Einsatzgebiet für Computer ist der Bahnverkehr. Die Streckenleiter haben die Aufgabe, den Passagier- und Frachtverkehr sicher und so kostengünstig wie möglich durch ihren Bereich zu leiten. Die britische Eisenbahn beispielsweise verwendet seit Mitte der 70er Jahre computergestützte Kontrollsysteme und folgt damit der amerikanischen Southern Pacific Railroads Gesellschaft. Deren vollintegriertes Operationssystem berücksichtigt alle Aspekte der Frachtsteuerung, angefangen bei einer genauen und aktuellen Einsatzübersicht

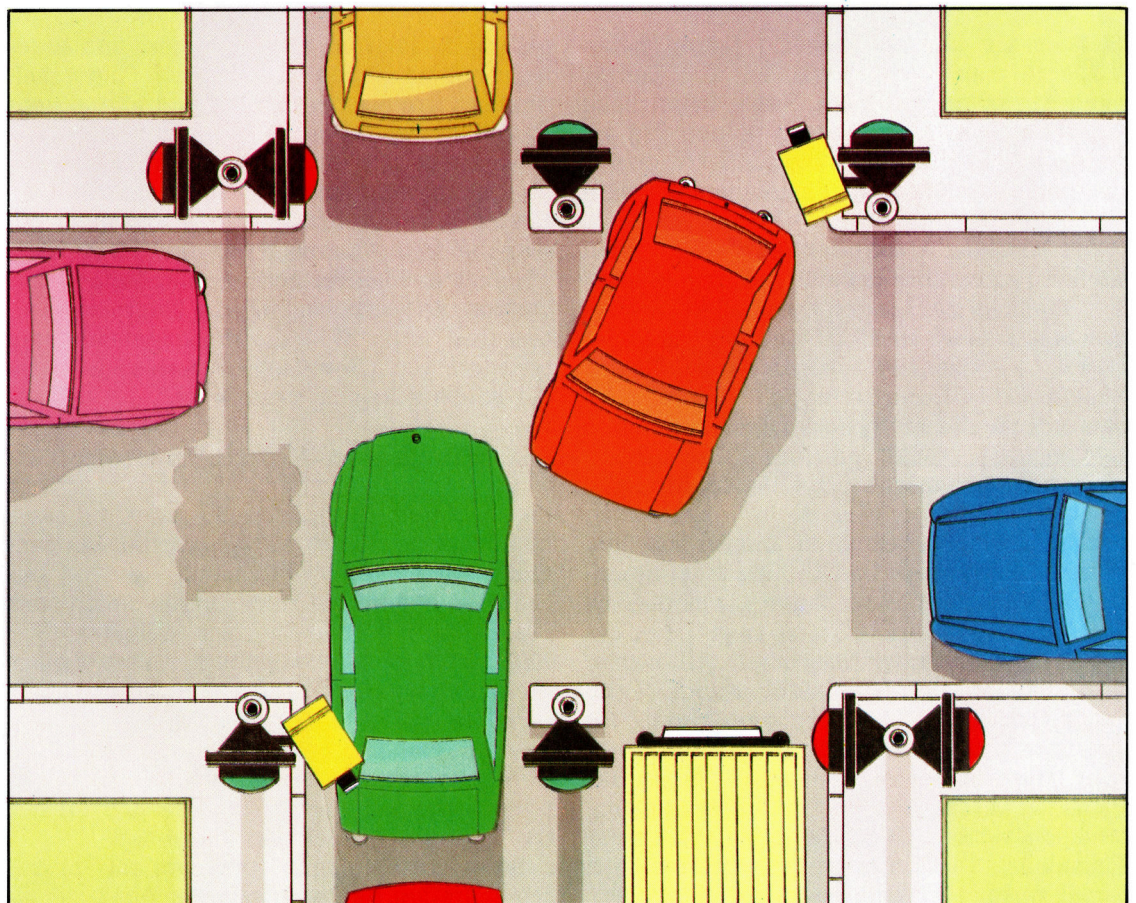
für jede Lokomotive und jeden Waggon bis hin zur Zusammenstellung der Einheiten zu vollständigen Zügen mitsamt der Routenplanung.

Jeder Güterwagen hat eine eigene Identifikationsnummer, die zusammen mit dem jeweiligen Stützpunkt in dem Zentralcomputer gespeichert wird. Benötigt man einen solchen Waggon an einem anderen Ort, so wird dieser an einen geeigneten Zug angekoppelt. Gleich nach der Ankunft wird der neue Wagenstatus wieder in den Rechner eingegeben, so daß eine Neuzuteilung erfolgen kann. In Anbetracht des heutigen Frachtvolumens der Eisenbahnen sind derartige Computerkontrollsysteme unentbehrlich geworden.

Automatische Fahrplanauskunft

Im Schienenverkehr haben sich weitere Möglichkeiten des Computereinsatzes eröffnet. Beispielsweise bei unbesetzten Haltestationen. Es sind schon verschiedene Versuche unternommen worden, Fragen von Reisenden per Microcomputer zu beantworten, und in einigen Fällen wurde für die Fahrplanauskunft schon eine synthetische Stimme verwendet. Computer ermöglichen es auch, Züge ohne Lokomotivführer einzusetzen. Die Londoner U-Bahn der Victoria-Linie bietet diese Möglichkeit, doch in der Praxis wird dies nicht durchgeführt, weil führerlose, durch Computer gesteuerte Züge von den Passagieren eher mit Arg-

Die Ampelphasen wurden von den Verkehringenieuren für einige Zeit eingesetzt, um den Verkehrsfluß besonders an Verkehrsknotenpunkten der Städte zu steuern. Heute können einzelne Ampeln die Verkehrsdichte in ihrer unmittelbaren Umgebung mit Hilfe eines Dopplerradars aufzeichnen und diese Daten an ein Computersystem übertragen. Die Häufigkeit des Lichtwechsels kann dann dem jeweiligen Fahrzeugaufkommen angepaßt werden.





Die Anzeigetafeln für Ankunft und Abflug sind die vielleicht am stärksten ins Auge fallenden Computersysteme eines Flughafens. Diese elektromechanischen Geräte werden von dem Zentralrechner, sobald neue Informationen verfügbar sind, fortlaufend bereitgestellt.

wohn betrachtet werden.

Eine Parallele zu den führerlosen Zugsystemen findet man bei höherentwickelten Modelleisenbahnen. Jede einzelne Lokomotive besitzt einen Identifikationscode, der in einem kleinen Microcomputer auf nur einem Chip gespeichert ist. Der Controller sendet seine Signale zu jedem Zug, indem er die Stromversorgung über die Schienen so moduliert, daß nur eine Lokomotive in der Lage ist, die Signale zu entschlüsseln. So können sehr viele Züge zur gleichen Zeit auf derselben Anlage unter der direkten Kontrolle des zentralen Microcomputers fahren. Es ist jedoch unwahrscheinlich, daß sich dieses Konzept auch bei Straßenfahrzeugen durchsetzen wird. Trotzdem werden Computer auch im Straßenverkehr eingesetzt, besonders bei öffentlichen Verkehrsmitteln und im Gütertransport. Hier helfen sie, Fahrpläne aufzustellen, was bei einer Großstadt eine äußerst komplexe Aufgabe ist. So ist es notwendig, Busse, Züge und U-Bahnen zu einem gemeinsamen öffentlichen Verkehrsverbund zusammenzuführen.

Ein interessantes Experiment im Computerbereich für den öffentlichen Nahverkehr, ein Bus-Ruf-System, wird zur Zeit in den Randbezirken von Hannover durchgeführt. Basierend auf vielen Minibussen, die keiner festen Route folgen, erlaubt das System dem Reisenden, von der Haltestelle aus eine zentrale Kontrollstelle anzurufen und sein Fahrziel anzugeben. Ein Mini-Terminal druckt dann die voraussichtliche Ankunftszeit des Busses, die Fahrtdauer sowie den Tarif aus.

Der Transport von Gütern und die Personenbeförderung erreichen ungefähr 20 Prozent des Welthandelsvolumens. Der Einsatz von Computern ist in diesem Bereich weiter entwickelt als in anderen und hat zweifellos zu dem immensen Wachstum beigetragen. Während die meisten der hier vorgestellten Beispiele mit Großcomputern ausgeführt werden, sind viele jedoch auch für Heimcomputer geeignet. Mittlerweile werden eine ganze Reihe von Programm-Paketen für zahlreiche Anwendungsgebiete im Handel angeboten.

Die Schifffahrts-Industrie setzt Computer weit weniger ein als andere Verkehrsbetriebe, doch ein bedeutendes Anwendungsgebiet ist der Container-Service. Micro-Computer spielen eine wichtige Rolle in der Belegungsabwicklung (mehrere Versender teilen sich einen Container), der Organisation, der Auslegung des Containerterminals und in der Beladung des Containerschiffes, um die Gewichtsverteilung zu überprüfen.

Routenplanung

Ein weiteres Anwendungsgebiet, welches aus dem Einsatz statistischer Verfahren Nutzen zieht, ist die Routenplanung für Lieferwagen. Hier müssen die Leerfahrten zwischen der Be- und Entladung so kurz wie möglich gehalten werden. Eine interessante Variante ist die Einsatzplanung besonders von Taxis und Streifenwagen. Jede Fahrzeugposition wird als Straßenname in den Rechner eingegeben, der diese dann in Koordinaten umwandelt. Bei der Anforderung eines Taxis oder Notarzwagens erfolgt die Eingabe auf die gleiche Weise, und die Aufgabe, die vorhandenen Fahrzeuge optimal dem Bedarf anzupassen, findet über einen Abgleich der beiden Koordinaten nach vorgegebenen Regeln statt.





Schwarzes Brett

Über ein Modem kann sich jeder Computerbesitzer Zugang zu einem elektronischen „schwarzen Brett“ verschaffen, das wie das Anschlagbrett eines Clubs funktioniert.

Mit einem Heimcomputer allein hat man nur wenige Möglichkeiten, mit anderen auf elektronische Weise in Verbindung zu treten. Per Telefon aber hat man Zugang zu einer großen Zahl anderer Computer – Großcomputern von Universitäten, öffentlichen Datenbanken oder dem Heimcomputer des Nachbarn.

Auf einem elektronischen „schwarzen Brett“ können Nachrichten für die Besitzer anderer Heimcomputer hinterlassen werden, Gegenstände zum Verkauf angeboten oder Programme getauscht werden. Es handelt sich dabei um die Version des Nachrichtenbretts eines Clubs. Es wird meist von mehreren Heimcomputerbesitzern, die über Geräte mit großen Speicherkapazitäten verfügen, betrieben. Elektronische schwarze Bretter sind jedem zugänglich, vorausgesetzt, man kennt die Telefonnummer sowie das korrekte Paßwort.

Verbindungen nach Übersee

Entfernung ist dabei kein Hindernis: Maschinen der Heimatstadt lassen sich ebenso anwählen wie Geräte am anderen Ende des Landes, und auch Verbindungen nach Übersee bereiten keine Schwierigkeiten – wenn man nichts gegen hohe Telefonrechnungen hat.

Bevor Ihr Computer allerdings „Telefongespräche“ führen kann, brauchen Sie ein Gerät, das die Verbindung mit der Postleitung herstellt. Am besten eignet sich dafür ein Modem, das über den seriellen (RS232) Ein- und Ausgang des Computers angeschlossen wird. Manche Heimcomputer benötigen eine zusätzliche Schnittstellenkarte, da sie keinen Eingang dieser Art haben. Ein Modem wird entweder durch Direktanschluß oder über den auf die Hörschalen gelegten Telefonhörer mit dem Telefonnetz verbunden. Auch ein Akustikkoppler kann die Verbindung herstellen. Normalerweise werden zwei Übertragungsgeschwindigkeiten (Baudraten) verwendet: 300 Baud und 1200/75 Baud. Kommerzielle Systeme bieten zumeist die höhere Geschwindigkeit von 1200/75 Baud, während private Anbieter ihre Systeme (speziell für Heimcomputer) auf 300 Baud eingestellt haben. 1200/75 Baud bedeutet, daß die Informationen des Anbieters mit 1200 Baud gesandt werden, während der Anrufer Nachrichten an den Zentralcomputer mit 75 Baud übermittelt. Natürlich sollte ein Modem verstellbare Baudraten haben. Da der Preis für Modems stark gefallen

ist, erhält man heute ein Gerät, das für 300 Baud ausgelegt ist, schon für etwa 300 DM.

Für den Betrieb eines Modems gibt es zwei Arten von Programmen, mit denen die Maschine wie ein Computerterminal funktionieren kann – „intelligente“ und „unintelligente“. Ein intelligentes Terminal kann Programme laden und die Nachrichten anderer Computer speichern. Stationen ohne eigene Intelligenz haben diese Möglichkeiten nicht, sind aber einfacher zu bedienen und daher für die ersten Übertragungsversuche besser geeignet. Für einige Computer wird entsprechende Software von den Computerclubs oft kostenlos zur Verfügung gestellt. Es ist aber nicht schwierig, sich ein eigenes „Emulationsprogramm“ zu schreiben, falls diese Möglichkeit besteht. Das Programm braucht nur alle auf der Tastatur getippten Zeichen an den RS232-Ausgang weiterzugeben und die dort ankommenden Daten auf den Bildschirm zu bringen. Seit einiger Zeit werden jedoch Terminal-Programme mit zusätzlichen Funktionen von verschiedenen Software-Firmen angeboten.

Um den Gebrauch eines Modems anschaulich darzustellen, folgen nun die einzelnen Schritte, die für die Kommunikation mit einem „schwarzen Brett“ erforderlich sind. Natürlich können die Befehle sowie die Reihenfolge variieren. Zunächst brauchen Sie die Telefonnummer und die technischen Einzelheiten des Computers, zu dem Sie Zugang haben möchten (beispielsweise die Baudzahl). Sie starten das Übertragungsprogramm und wählen die Nummer. Ist der Computer am anderen Ende der Leitung eingeschaltet, ertönt im Hörer ein hoher Ton – die „Trägerfrequenz“. Bei einem Akustikkoppler setzen Sie den Hörer jetzt auf die Gummimuscheln, bei einem Modem betätigen Sie den Schalter „LINE“ oder „DATA“ und legen den Hörer auf. Beide Computer stehen nun miteinander in Verbindung.

Ohne „Paß“ kein Zugriff

Das erste, was auf Ihrem Bildschirm erscheinen wird, ist vermutlich die „Begrüßungszeile“ des Computers am anderen Ende der Leitung. Dann werden Sie nach Ihrer Identität und/oder einem Paßwort gefragt. Haben Sie kein Paßwort und wird der angewählte Dienst von einer Privatfirma betrieben, werden Sie kaum weiter kommen, es sei denn, Sie sind gut im Raten oder sehr geduldig. Es gibt jedoch viele Com-

TELECOM GOLD

Telecom Gold ist ein System für elektronische Post in England. Die Nachrichten von anderen Teilnehmern können gelesen und auch in den Computer des Teilnehmers kopiert werden. Mit Telecom Gold ist es möglich, mit elektronischen Netzwerken auf der ganzen Welt in Verbindung zu treten, wobei die technischen Unterschiede der einzelnen Systeme untereinander automatisch von den Computern des Netzwerks ausgeglichen werden.



puter, die „Gästen“ begrenzten Zugang geben. Probieren Sie deshalb ruhig ein paar Paßwörter wie NEWUSER, GAST oder HELP aus.

Ein schwarzes Brett steht jedem offen und kostet nichts (außer den Telefongebühren). Sie geben einfach Ihren Namen und die Stadt an, aus der Sie anrufen (wenn danach gefragt wird), und „loggen“ sich auf diese Weise ein. Der Computer fragt dann vermutlich nach dem Zeilenformat Ihres Bildschirms oder Ihrem Gerätetyp. Mit diesen Informationen kann er Sie in Zukunft identifizieren und sein System mit Ihrem abstimmen.

Nach dieser Anmeldung erhalten Sie meist Systeminformationen wie Betriebszeiten, technische Einzelheiten und die maximale Anrufdauer. Erst jetzt kommen Sie an das Hauptmenü heran, das Ihnen eine Anzahl von Befehlen zur Auswahl anbietet, die Sie durch Drücken der entsprechenden Anfangsbuchstaben auswählen können. Um sich zum Beispiel als neuer Teilnehmer informieren zu können, tippen Sie N für NEU und zur Beendigung des Anrufs geben Sie G für GOODBYE ein. Oftmals erscheint nach Aufruf einer Funktion ein weiteres Menü, aus dem Sie wiederum die entsprechende Option auswählen, bis Sie dort sind, wo Sie sein wollen. Ein elektronisches schwarzes Brett hat eine Baumstruktur. Sie fangen beim Stamm an und verzweigen mit jedem Untermenü in die gewünschte Richtung.

Mit der Option „neuer Teilnehmer“ können Sie sich in dem schwarzen Brett eintragen. Ihr Name und Ihre Adresse werden registriert, und Sie können eine Benutzerkennung für zu-

künftige Anrufe angeben. Die Option „Information“ gibt Ihnen Einzelheiten über das System, mit dem Sie verbunden sind. Über „Utilities“ (Hilfsprogramme) können Sie abfragen, wie lange Ihr Anruf gedauert hat, und auch Informationen einholen.

Privatnachrichten

Unter der Option „schwarzes Brett“ finden Sie schließlich die Nachrichten, die andere Teilnehmer dort zum öffentlichen Aushang gespeichert haben. Hier können Sie auch Ihre eigenen Informationen „aushängen“. In der Abteilung „Privatnachrichten“ finden Sie Botschaften, die dort für bestimmte Teilnehmer abgelegt sind, und auch hier können Sie Informationen hinterlegen. Oft befinden sich an dieser Stelle auch Nachrichten für ganze Teilnehmergruppen.

Ein etwas anders gearteter Dienst in England, genannt REWTEL, bietet eine Spezialdatenbank für Informationen über elektronische Bauteile. Eingetragene Mitglieder können von ihrer Tastatur aus damit sogar Teile bestellen. Anders als beim schwarzen Brett geben Sie hier eine Kennung für den technischen Bereich ein, der Sie interessiert. Mit dem Paßwort HELP REWSHOP erhalten Sie Unterstützung bei einer Bestellung, und mit CHALK können Sie auf einer Art schwarzem Brett Nachrichten hinterlassen. Nichtmitglieder haben die Möglichkeit, den Dienst acht Minuten lang zu benutzen. Vorteil ist eine „Öffnungszeit“ rund um die Uhr.

Eine der interessantesten Einsatzmöglichkeiten für ein Modem ist der Zugang zu einem elektronischen schwarzen Brett. Nicht viel anders als am Nachrichtent Brett eines Clubs können darauf Botschaften „angeschlagen“ werden, die für jeden bestimmt sind, oder auch „verschlüsselte Briefe“, die nur mit dem richtigen Paßwort „geöffnet“ werden können. Es werden dort Zusammenkünfte angekündigt und gebrauchte Computerausrüstungen zum Verkauf angeboten. Sogar Spiele und Programme können damit auf eigene Disketten oder Cassetten überspielt werden.



Fernspiele

Nicht jedes Computerspiel erfordert sekundenschnelle Reaktionen. In Fernspielen kann ein Zug bis zu sechs Wochen dauern.

'Starlord' war in Großbritannien das erste Fernspiel, das von einem Computer verwaltet wurde. Es wurde von Mike Singleton koordiniert, der dafür den Computer Pet 3032, ein Plattenlaufwerk mit 7,5 Megabyte Kapazität und einen Integrex Farb-Matrixdrucker einsetzte. Ziel jedes Spielers war, den Thronstern zu finden und galaktischer Kaiser zu werden. Nach jedem Zug erhielt der Teilnehmer eine Karte, die das Gebiet in unmittelbarer Umgebung seiner Truppen zeigte, und eine Liste, wer die umliegenden Planeten besaß. Die hohe Speicherkapazität war notwendig, um die große Anzahl Programme unterzubringen, die das Spiel steuerten. Außerdem mußten die Daten von mehr als 700 Teilnehmern verwaltet werden.

O bwohl die meisten Käufer von Heimcomputern behaupten, sie wollten darauf BASIC lernen, werden doch zweifellos die meisten Geräte nur für interessante und lehrreiche Spiele eingesetzt. Durch den Computer sind völlig neue Arten wie Abenteuer-, Lehr- und Brettspielvarianten entstanden, die auf unterschiedliche Schwierigkeitsgrade eingestellt werden können.

Ein Spieltyp wird Ihnen jedoch vermutlich unbekannt sein – das Fernspiel. Vergleicht man es mit Arcadespielen, die Reaktionszeiten von Sekundenbruchteilen erfordern, dann handelt es sich hierbei um ein Spiel im „Schneckentempo“. Zwischen einzelnen Zügen können durchaus Wochen vergehen. Einige Dutzend Spieler nehmen gleichzeitig teil, sind dabei allerdings über ein ganzes Land oder im Fall von internationalen Spielen über die ganze Welt verteilt. Jeder Zug wird innerhalb eines bestimmten Zeitraums schriftlich fixiert und per Post an den Koordinator des Spiels gesandt. Dieser gibt sie in einen Micro-

computer ein. (Bei den meisten Fernspielen braucht der Teilnehmer keinen Heimcomputer zu besitzen.) Der Koordinator schickt dem Spieler dann einen Computerausdruck mit relevanten Informationen, dem aktuellen Stand und den neuen Positionen der Mitspieler.

Spiele dieser Art können sich entweder monatelang fortsetzen oder auch überhaupt nicht beendet werden, wenn die Regeln es erlauben, daß jederzeit neue Teilnehmer einsteigen können. Normalerweise wird eine Eintrittsgebühr verlangt, für die man die Spielregeln zugesandt bekommt. Danach wird pro Zug ein geringer Betrag (meist um vier Mark) entrichtet. Findet jede Woche ein Zug statt, hat man es mit einem extrem schnellen Spiel zu tun – bei internationalen vergehen oft bis zu sechs Wochen zwischen den einzelnen Zügen. Melden sich zu viele Interessenten, fängt der Koordinator ein weiteres Spiel mit anderen Disketten, aber dem gleichen Programm, an.

Fernspiele gab es schon lange, bevor Computer gebaut wurden. Beispiele dafür sind Fernschach oder das Brettspiel „Perplomatie“, bei dem die Spieler sieben europäische Nationen darstellen, die versuchen, auf diplomatischem Wege Europa zu erobern. Durch den Einsatz von Computern für Berechnungen und Verwaltung sind die Spiele phantasievoll verfeinert worden. In einigen gibt es riesige Galaxien, durch die Raumflotten bewegt werden; andere handeln in mystischen Ländern, in denen sich Königreiche bekämpfen; und natürlich gibt es auch Computerversionen des Spiels „Diplomatie“.

Zehn und mehr Spieler

Einzigartig an diesen Spielen ist der Kontakt der Teilnehmer untereinander. Das unterscheidet sie wesentlich von dem einsamen Erforschen in einem Abenteuerspiel. Nicht selten gibt es dabei Bündnisse zwischen zehn und mehr Spielern, die jede Woche per Brief oder per Telefon in Verbindung stehen. Die Qualität der Fernspiele kann daran erkannt werden, daß das Altersspektrum der Teilnehmer breiter ist als bei den meisten anderen Computerspielen.

Per Modem und Akustikkoppler haben auch bundesdeutsche User die Möglichkeit Fernspiele zu starten. In England ist man schon weiter: Zeitschriften wie 'Flagship' befassen sich ausschließlich mit Spielen dieser Art.





Der Sinclair QL

Der neueste Sinclair-Computer bietet viel, ist aber nicht unumstritten.



Der Name Sinclair steht auf einer ganzen Palette von Produkten: Von Tonverstärkern bis zu der berühmten „Black Watch“, von Taschenrechnern und Computern bis zum flachen Fernsehschirm und zum Elektroauto. Technische Findigkeit, neue Ideen, hochtechnisiertes Design und ehrgeizige Marketingstrategien waren von Anfang an das Markenzeichen von Sinclair. Böse Zungen behaupten jedoch, daß Effekthascherei, nur für die Presse konstruierte Geräte und überzogene Liefertermine die Firma besser charakterisieren würden.

Als Clive Sinclair seinen ZX Spectrum vorstellte, verglich er ihn mit dem Acorn B, der im Preis doppelt so hoch lag. Zwei Jahre später, bei der Ankündigung des QL, behauptete er sogar, sein Gerät leiste das gleiche wie der IBM PC oder der Macintosh von Apple, die fünfmal soviel kosten wie der QL.

Über Leistung und Einsatzgebiet des QL gibt es noch immer widersprüchliche Meinungen. Bei der Vorstellung des Gerätes gab es einen großen Wirbel in den Medien. Der QL sollte der modernste Heimcomputer überhaupt sein, ausgerüstet mit dem 32-Bit-Prozessor 68008, vier integrierten, kommerziell einsetzbaren Programmen und einem BASIC, das jede andere bekannte BASIC-Variante in den Schatten stellt.

Rückschläge

Als dann die ersten Liefertermine nicht erfüllt werden konnten, ließ die Begeisterung schnell nach, und bald kam man zu dem Schluß, daß Sinclair wohl ein wenig übertrieben habe. Ein echter Rückschlag kam, als klar wurde, daß die Maschine – trotz gegenteiliger Beteuerungen – noch weit von der Serienreife entfernt

war: Das Geld für vorbestellte und bezahlte Maschinen lag auf Sinclairs Konten, während seine Kunden warten mußten und sich schmerzlich an frühere Erfahrungen mit der Auslieferung von Sinclair-Produkten erinnerten.

Trotzdem bestand zwischen den Einführungen des Spectrum und des QL ein großer Unterschied: Unmittelbar nach der Vorstellung des Gerätes wurden Journalisten funktionierende Testgeräte zur Verfügung gestellt, die mit den Maschinen fast identisch waren, die später zum Verkauf freigegeben werden sollten. Sinclair mußte allerdings zugeben, daß das versprochene „Super-BASIC“ nicht in die dafür abgestellten 32 KByte ROM paßte, sondern 48 KByte benötigte. Für den zusätzlichen Chip war jedoch kein Platz mehr vorhanden.

Statt Geld und Zeit für den Umbau der Platine zu verwenden, kam Sinclairs Team die Idee zu dem – inzwischen berühmt-berüchtigten – „dongle“, ein kleiner schwarzer Kasten, der aus dem Eingang für Cartridges herausragte und die fehlenden Teile des BASIC und des Betriebssystems enthielt.

Mit dieser Lösung konnte Sinclair erst einmal ein paar funktionierende Maschinen aus-



liefern – spätere verbesserte Versionen sollten dann ohne diese externe Erweiterung funktionieren. Inzwischen waren aus den versprochenen 28 Tagen Lieferfrist drei Monate geworden. Nun wurden innerhalb eines kurzen Zeitraums mehrere Versionen des Betriebssystems gegeneinander ausgetauscht – jedes mit Fehlern behaftet, bis sich Sinclair für die Version „AH“ entschied, die dann auch in größeren Stückzahlen verfügbar war.

Die Öffentlichkeit reagierte nicht sehr positiv auf den „dongle“. Als sichtbarer Beweis dafür, daß die Maschine nicht durchkonstruiert war, ließ er Zweifel an der Gesamtzuverlässigkeit des QL aufkommen. Um dieses Manko zu beseitigen, kam Sinclair auf eine andere Idee: Aus besagten Platzgründen wurde der zusätzliche 16 K ROM-Chip „Huckepack“ auf einen der bereits installierten Chips gesetzt und alle Kontakte bis auf einen einzeln an die darunterliegenden Chips gelötet. Ein freies Kabel verband den letzten Kontakt mit einem anderen Teil der Platine, so daß der zusätzliche Chip unabhängig von seinem Träger angesprochen werden konnte. Der „dongle“ war jetzt zwar beseitigt, aber an der Fehlkonstruktion selbst hatte sich nichts Wesentliches geändert.

Gute Figur

Alle frühen Versionen des QL enthielten EPROMs statt ROMs. Sinclair sparte sich damit zwar die Zeit für die Konstruktion der ROMs, verlor aber eine Menge Geld, da jeder QL drei EPROMs brauchte, die pro Stück etwa 300 Mark kosteten. Obwohl Sinclair mit Sicherheit Mengenrabatt in Anspruch nehmen konnte, machte dieser Preis doch einen zu hohen Anteil der Gesamtkosten aus und veranlaßte ihn vermutlich, sich trotz bestehender Fehler für die „AH“-Version des Betriebssystems zu entscheiden. Nun konnten die teuren EPROMs endlich gegen ROMs ausgetauscht werden.

Wie läßt sich vor diesem Hintergrund der QL nun überhaupt beurteilen? Einerseits ist er ein hochentwickelter Heimcomputer, andererseits ein begrenzt kommerziell einsetzbares Gerät, wobei diese Kombination allein schon ein völlig neues Konzept darstellt. Zieht man jedoch alles in Betracht, entspricht der QL wohl doch eher dem herkömmlichen Bild eines Heimcomputers: Er ist klein, verfügt über ein eingebautes BASIC, hochauflösende Farbgrafik, Anschlüsse für Joysticks und wird über Kaufhäuser und Versandgeschäfte vertrieben. Zwei wesentliche Eigenschaften unterstützen jedoch den Anspruch auf einen kommerziellen Einsatz: Die eingebauten Microdrives sind (im Vergleich mit Cassetten) ein durchaus brauchbares Speichermedium, und die Maschine wird standardmäßig mit vier Anwenderprogrammen geliefert: Text- und Kalkulationssystem, Datenbank und Grafik.

Die Konkurrenz

QL und Acorn B

Vergleicht man die beiden Geräte, so fällt auf, daß beim Acorn B keine Programme mitgeliefert werden, das Gerät keinen Massenspeicher oder erweiterbaren Arbeitsspeicher besitzt und nicht mit einem modernen Microprozessor arbeitet. Andererseits fehlt dem QL die breite Palette an qualitativ hohen Zusatzgeräten anderer Hersteller, die große Anzahl verfügbarer Programme und die Möglichkeit, die Maschine mit einem zweiten Prozessor weiter ausbauen zu können.



QL und Macintosh

Mit der Verwendung des gleichen Microprozessors suchte Sinclair den Vergleich mit einer Maschine, die vier- bis fünfmal so teuer ist. Doch obwohl beide Rechner eine Anzahl gleicher Eigenschaften besitzen, wie die Größe des Arbeitsspeichers, Taktgeschwindigkeit und integrierte Programme, ist der Vergleich damit auch schon beendet. Der QL ist zwar technisch gesehen eine hervorragend konstruierte Maschine, bietet aber keine wesentlichen Neuerungen. Apples Macintosh dagegen ist mit seinem völlig neuen Betriebssystem für Symbolsteuerung, Fenster-technik und der brillanten Integration der Maus eine Maschine, die weit aus dem Computermarkt herausragt.



Als Heimcomputer macht der QL eine gute Figur, da diese Geräte selten mit eingebautem Speichermedium sowie Small-Business-Programmen ausgerüstet sind. Das integrierte SuperBASIC ist einer der besten existierenden BASIC-Dialekte und damit eine echte Bereicherung. Wegen der Problematik in der Konstruktion der Maschine wird für den QL zur Zeit noch wenig kommerzielle Software angeboten. Erschwerend für die Herstellung neuer Programme kommt hinzu, daß die Microdrives des Spectrum und des QL unterschiedliche Formate haben, daß ihre Betriebssysteme nicht

Alle vier Programme verfügen über einen ähnlichen Bildschirm-aufbau, identische Befehlssätze und knappe und klare Darstellungen. Mit Hilfe der Microdrives können Daten von einem Programm zum anderen transferiert werden. Das Textprogramm Quill erlaubt die Bildschirmdarstellung mit 40, 64 und 80 Zeichen pro Zeile. Abacus ist ein neuartiges Kalkulationsprogramm mit vielen integrierten Funktionen und der Möglichkeit, eine Anzahl von Feldern als Gruppe zu bezeichnen und anzusprechen. Leider reicht der nach dem Laden des Programms noch verfügbare Arbeitsspeicher von 15 KByte für ernsthafte Anwendungen nicht aus. Die Archive der Datenbanken verfügen über eine Reihe integrierter Befehle zum Speichern und Suchen von Daten. Leider wird sie durch die geringe Geschwindigkeit der Microdrives relativ langsam. Mit Easel können numerische Daten als Balken-, Kurven- oder Tortendiagramme dargestellt werden.



SINCLAIR QL

PREIS

ca. 1.600 Mark

ABMESSUNGEN

472 x 138 x 46 mm

ZENTRALEINHEIT

Motorola 68008; 7,5 MHz

SPEICHERKAPAZITÄT

128 K RAM (erweiterbar bis 640 K) 48 K ROM

BILDWIEDERGABE

25 Zeilen mit je 80 Zeichen (Monitor); hochauflösende Grafik: 512 x 256 Pixel (4 Farben); 256 x 256 Pixel (8 Farben)

SCHNITTSTELLEN

Seriell RS232 (2), Joysticks (2), Microdrives, LAN, TV, RGB Monitor

PROGRAMMIERSPRACHE

SuperBASIC

TASTATUR

Pseudo-Schreibmaschinentastatur; 65 Tasten mit Leerzeichentaste und fünf Funktionstasten. Keine Löschtaste

DOKUMENTATION

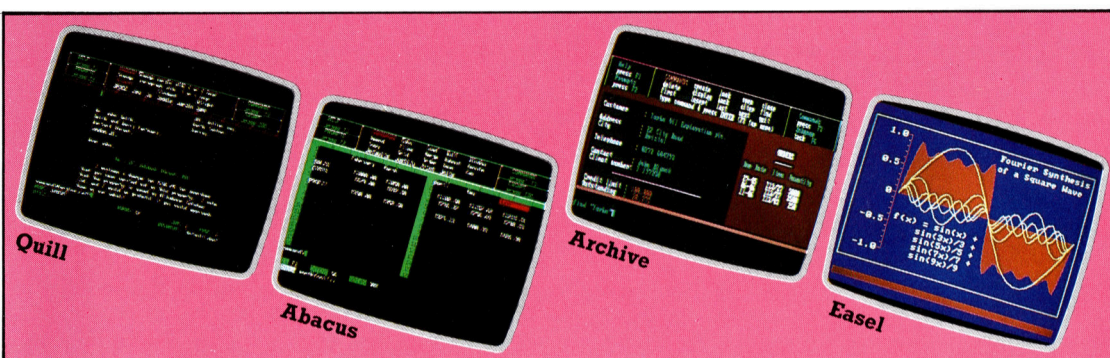
Benutzerhandbuch im Ringbuchformat enthält Anleitungen für SuperBASIC und die Anwendungsprogramme

STÄRKEN

Extrem schnelle Zahlenverarbeitung durch die 68008 CPU

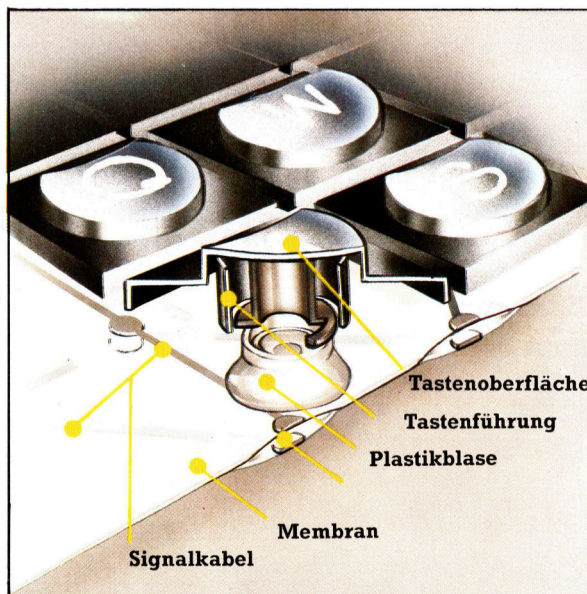
SCHWÄCHEN

Die eingebauten Microdrives arbeiten recht langsam



kompatibel sind und ihr Bildschirmaufbau unterschiedlich gehandhabt wird. Die Grafikfähigkeiten des QL sind ausgezeichnet, sein Tongenerator erzeugt jedoch enttäuschende Klänge.

Als kommerziell einsetzbare Maschine überzeugt der QL nicht so sehr. Das mitgelieferte Programmpaket ist zwar gut, es fehlt aber eine Finanzbuchhaltung. Weiterhin stehen nach dem Laden des Kalkulationssystems nur 15 Kbyte RAM zur Verfügung, und das reicht für ernsthafte Anwendungen bei weitem nicht aus. Die Langsamkeit und Unzuverlässigkeit der Microdrives stellen die Speicherfähigkeiten des QL in Frage, da es keine weiteren Schnittstellen für Disketten gibt. Die Tastatur macht nicht den Eindruck, als würde sie der täglichen Büroarbeit lange standhalten, und man kann sich nur schwer vorstellen, daß erfahrene Schreibkräfte sich mit deren Eigenheiten abgeben werden.



Membran und Blasen

Der QL hat eine Membrantastatur. Dabei hält eine in eine Plastikmembrane eingelassene Blase zwei Signalkabel auseinander. Wird eine Taste gedrückt, verflacht sich die Blase und die beiden Kabel haben Kontakt. Tastenwiderstand und Tastenrückführung werden ebenfalls von dem kleinen Plastikballon geregelt. Die flache Plastikastatur in fast standardmäßiger QWERTY Auslegung und mit geformten, voll beweglichen Tasten ist ein großer Fortschritt.



Leichte Muse

Klangerzeugung auf dem Sinclair Spectrum

Der Spectrum von Sinclair bietet für wenig Geld ausgezeichnete Farbgrafik und brauchbare Speicherkapazitäten. Um den Preis niedrig halten zu können, mußte bei diesem Gerät auf großen Komfort verzichtet werden. Das betrifft hauptsächlich die Tastatur und das nicht standardmäßige BASIC, mit Sicherheit aber ist die Klangerzeugung sein schwächster Punkt.

Bevor überhaupt ein Ton erklingen kann, muß das Gerät zunächst entweder an eine externe HiFi-Anlage angeschlossen oder der interne Lautsprecher mit dem folgenden direkten Befehl aktiviert werden:

```
POKE 23609,100
```

Damit vergrößert sich ebenfalls die Lautstärke des „Klick“, das beim Drücken einer Taste zu hören ist.

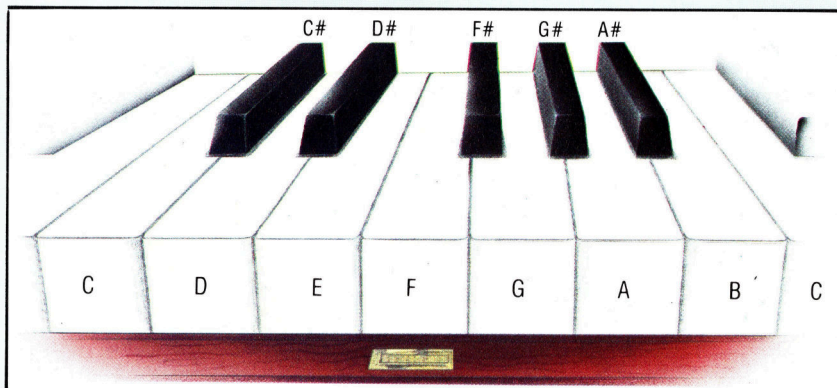
Tonsteuerung

Zur Erzeugung eines bestimmten Tones benutzt man den BEEP Befehl mit folgendem Format:

```
BEEP,d,n
```

wobei „d“ die Länge des Tones bezeichnet und „n“ die Tonhöhe. Die Tondauer kann zwischen 0,00125 und 10 Sekunden eingestellt werden; die Höhe eines Tones wird als die Zahl der Halbtöne ausgehend vom mittleren C (das den Wert 0 hat) dargestellt. Die Zahlen reichen von -60 bis 69. Im folgenden Beispiel erklingt die Note „A“ mit 440 Hz. Der Ton liegt neun Halbtöne über dem mittleren C und dauert eine halbe Sekunde:

```
BEEP,5,9
```



Die notwendigen Pausen zwischen den einzelnen Tönen einer Tonfolge lassen sich mit PAUSE exakt angeben. Der Befehl hat folgendes Format:

```
PAUSE ms
```

wobei „ms“ die Zeit in Einheiten von 0,001 Sekunden (eine Millisekunde) darstellt. Das folgende Programm spielt die Oktave in C-Dur vom mittleren C an aufwärts. Jeder Ton dauert eine halbe Sekunde mit Pausen von je einer Viertelsekunde zwischen den Tönen:

```
10 FOR I = 1 TO 8
20 READ N
30 BEEP .5,N
40 PAUSE 250
50 NEXT I
60 DATA 0,2,4,5
70 DATA 7,9,11,12
```

Mit diesem Programm läßt sich auch leicht das Format einer Tonleiter darstellen. Eine Oktave fängt bei dem Grundton an (in diesem Fall das mittlere C) und hört bei der nächsten Note mit der gleichen Buchstabenbezeichnung auf. Sie besteht aus acht Tönen und umfaßt einen Bereich von 12 Halbtönen.

Der Spectrum als Klavier

Mit dem Befehl INKEY\$ kann der Spectrum in eine Klaviertastatur umgewandelt werden:

```
10 REM *****
20 REM *OCTAVE PIANO*
30 REM *****
40 IF INKEY$="Q" THEN BEEP1,0
50 IF INKEY$="2" THEN BEEP1,1
60 IF INKEY$="W" THEN BEEP1,2
70 IF INKEY$="3" THEN BEEP1,3
80 IF INKEY$="E" THEN BEEP1,4
90 IF INKEY$="R" THEN BEEP1,5
100 IF INKEY$="5" THEN BEEP1,6
110 IF INKEY$="T" THEN BEEP1,7
120 IF INKEY$="6" THEN BEEP1,8
130 IF INKEY$="Y" THEN BEEP1,9
140 IF INKEY$="7" THEN BEEP1,10
150 IF INKEY$="U" THEN BEEP1,11
160 IF INKEY$="I" THEN BEEP1,12
170 GOTO 40
```

Dieses Programm muß natürlich noch verfeinert werden, wenn damit auf dem Spectrum ein „Tasteninstrument“ simuliert werden soll.



Grundmuster

Die grafischen Fähigkeiten des Commodore VC 20

Wie der Commodore 64 und der PET, ist auch der VC 20 ein gut durchkonstruierter Heimcomputer. Der BASIC-Befehlssatz enthält allerdings keine speziellen Grafikkommandos, so daß ein Anwender ohne genaue Kenntnisse der internen Funktionsweise nur schwer Grafikprogramme schreiben kann. Natürlich kann man dafür Zusatzprogramme kaufen, aber es gibt auch eine andere Möglichkeit. Der Commodore ist mit einem Satz von Spezialzeichen ausgerüstet, die mit ein wenig Erfindungsgabe zu interessanten Formen zusammengesetzt werden können.

Auf dem VC 20 lassen sich 16 Farben darstellen, von denen jedes Spezialzeichen vier enthalten kann. Das Bildschirmformat besteht standardmäßig aus 23 Zeilen mit je 22 Zeichen, von denen jedes aus achtmal acht Bildpunkten besteht. Die Zeichen lassen sich jedoch auch rechteckig im Format von 16 x 8 Bildpunkten darstellen. Sogar hochauflösende Grafik ist auf dem VC 20 möglich, die Programmierung ist aber kompliziert.

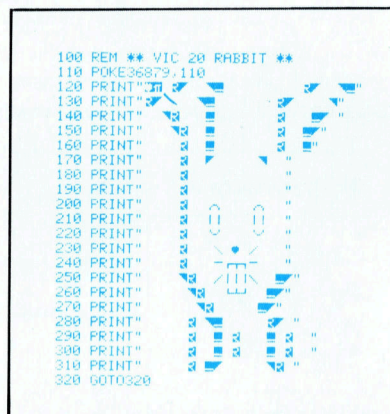
Grafik mit geringer Auflösung

Der VC 20 verfügt außer über Groß- und Kleinbuchstaben noch über mehr als 60 der speziellen Grafikzeichen des PET. Auf vielen Tasten des VC 20 befinden sich zwei kleine Quadrate, von denen jedes das entsprechende Grafikelement enthält. Neben den Halb- und Viertelzeichen gibt es Spielkartensymbole und Schachbrettmuster, Kreise und viele andere Symbole, mit denen auf dem Bildschirm Kurven gezeichnet, Tabellen und Großbuchstaben dargestellt sowie viele andere Effekte erzeugt werden können. Da sich jedes Zeichen invertieren läßt (schwarz auf weiß statt weiß auf schwarz), vervielfachen sich die Möglichkeiten nochmals. Mit Geduld und ein wenig Kreativität kann man durchaus qualitativ hochwertige Darstellungen auf den Schirm zaubern.

Die Zeichen lassen sich entweder mit dem PRINT-Befehl auf den Bildschirm bringen oder durch POKE mit entsprechenden Zahlenfolgen in den Bildschirm- und Farbspeicher des VC 20 ablegen. Commodore hat in den PRINT-Befehl die Möglichkeit eingebaut, jedem einzelnen Zeichen Farbwerte zuordnen zu können. Da sich auch der Cursor über den PRINT-Befehl leicht steuern läßt, sind Bewegungen einfach zu programmieren. Zwar können Zeichen über POKE nicht so schnell dargestellt werden wie mit PRINT, bei manchen Programmen ist dieser Befehl aber sehr brauchbar.

Schon in der Grundversion des VC 20 ist hochauflösende Grafik möglich, aber leider reicht der Speicher nur für die Hälfte des Bildschirms aus. „Bit Mapping“ (Bitmuster) wird die Technik genannt, mit der ein Programmierer jeden Bildpunkt in einem bestimmten Bereich des Schirms steuern kann. Jedes Zeichenfeld in der Darstellungsart von 23 Zeilen mit 22 Spalten besteht aus 64 Bildpunkten (Pixeln), die in acht Reihen mit je acht Pixeln angeordnet sind. Zwischen den Koordinaten eines Bildpunktes (x, y) und dem entsprechenden Bit der Zeichenmatrix besteht ein mathematischer Zusammenhang. In Verbindung mit dem POKE-Befehl kann damit ein Bildschirminhalt aus einzelnen Bildpunkten aufgebaut werden.

Es ist sehr aufwendig, hochauflösende Grafik nach der Bitmustermethode herstellen zu



Dieses VC 20 Programm gibt einen Eindruck von den vielen Einsatzmöglichkeiten der Spezialzeichen von Commodore. Der Hasenkopf ist nur aus diesen Zeichen aufgebaut. Bewegt man den Cursor über das fertige Bild, lassen sich die einzelnen Zeichen besser erkennen.

wollen. Man kann sich aber die Super Expander Cartridge von Commodore zulegen, die das BASIC mit Befehlen für hochauflösende Grafik, Farbdarstellung und Klangerzeugung erweitert. Der Befehlssatz für hochauflösende Grafik enthält die Kommandos GRAPHIC, mit dem diese Art der Darstellung eingeschaltet wird, POINT bringt einen einzelnen Bildpunkt auf den Schirm, und PAINT füllt ein Feld mit Farbe.

Die Cartridge-Version weist zwei gravierende Nachteile auf: Es gibt kein UNPOINT, mit dem ein Bildpunkt gelöscht werden kann, und PAINT kann keine zwei Seiten einer diagonalen Linie einfärben, ohne daß dadurch die Linie selbst in Mitleidenschaft gezogen wird. Die niedrige grafische Auflösung des VC 20 ist gut aufgebaut und flexibel. Ohne die zusätzliche Cartridge ist die Darstellung mit hochauflösender Grafik allerdings äußerst kompliziert zu programmieren.

Dem DOS auf der Spur

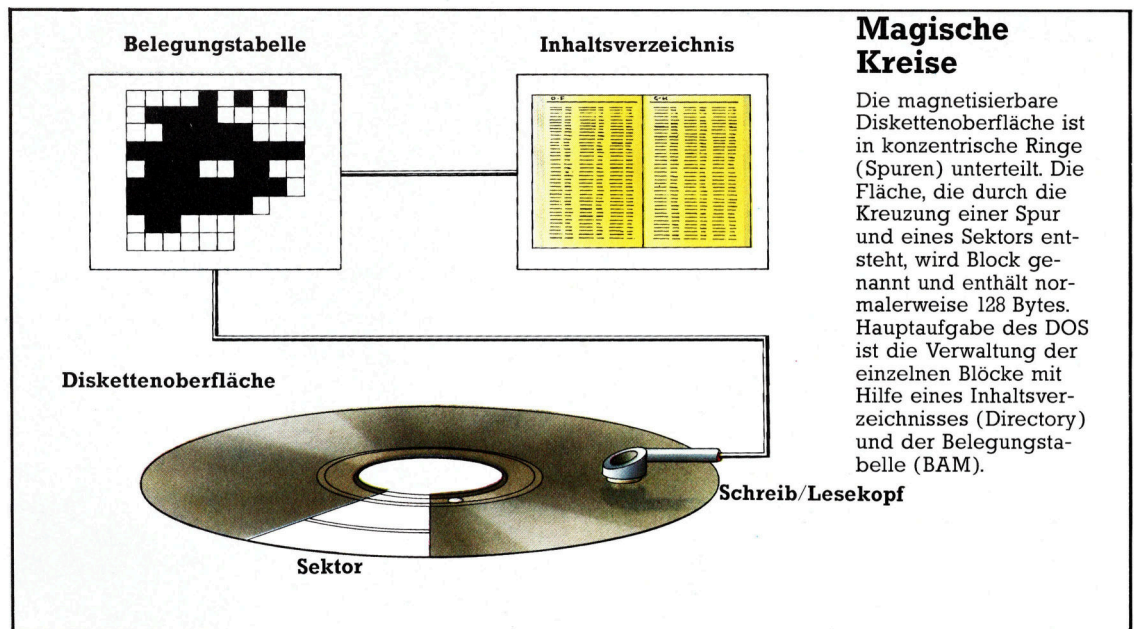
Hauptaufgabe des Diskettenverwaltungssystems (Disk Operating System) ist, den Platz auf der Diskette zu verwalten und sich zu merken, wo welche Informationen gespeichert sind.

Bevor ein Computer ein Programm ausführt, benötigt er einen internen Befehlssatz, um die verschiedenen Teile seines Systems aufeinander abzustimmen und die einzelnen Anweisungen des Programms verstehen zu können. Dieser interne Befehlssatz heißt Betriebssystem und ist in den meisten Heimcomputern als ROM-Chip fest eingebaut. Normalerweise ist nicht zu merken, daß das Betriebssystem arbeitet: Es ist „unsichtbar“.

Wenn Sie neben Ihrem Computer auch eine Diskettenstation arbeiten lassen, dann muß sich ein Großteil des Betriebssystems mit dessen Steuerung beschäftigen. Dieser Teil wird als DOS oder Diskettenverwaltungssystem bezeichnet. Der Name DOS taucht auch in einigen bekannten Produktbezeichnungen auf, zum Beispiel MSDOS, dem Betriebssystem der

Diese Methode erfordert „intelligente“ Diskettenstationen (wie die von Commodore). Solche Laufwerke verfügen über eigene Microprozessoren. Sie sind zwar teuer, haben aber etliche Vorteile gegenüber „unintelligenten“ Geräten: Sie belegen keinen wertvollen Speicherplatz und können mit der Diskettenverwaltung beschäftigt sein, während im Computer das eigentliche Anwenderprogramm weiterläuft. Bei der dritten Möglichkeit ist das DOS im Arbeitsspeicher des Computers untergebracht. Diese Technik ist besonders in kommerziellen Systemen weit verbreitet, bei denen die Diskettenlaufwerke in den Computer integriert sind und ein großes RAM zur Verfügung steht. Der Hersteller muß so nicht bei jeder kleinen Änderung einen neuen Satz ROMs produzieren.

Wie aber wird das DOS in diesem Fall beim



Firma Microsoft.

Es gibt drei Hauptmethoden, das DOS in einem Computersystem unterzubringen. Zum einen kann es als ROM-Chip im Rechner eingebaut sein, wie bei dem Spectrum von Sinclair, dessen Befehlssatz zur Steuerung des Microdrives auf diese Weise fest installiert ist. Zum zweiten kann das DOS als ROM direkt in das Diskettenlaufwerk eingebaut werden.

Einschalten des Systems überhaupt von der Diskette in den Arbeitsspeicher geladen, wenn darin noch kein DOS vorhanden ist? Kein Programm kann sich selbst ohne ein kleines Hilfsprogramm in das RAM laden. Deshalb wird beim Einschalten des Gerätes automatisch das „Bootstrap“ aktiviert. Dieses Programm hat seinen Namen von dem englischen Ausdruck „pull itself by its own bootstraps“ =

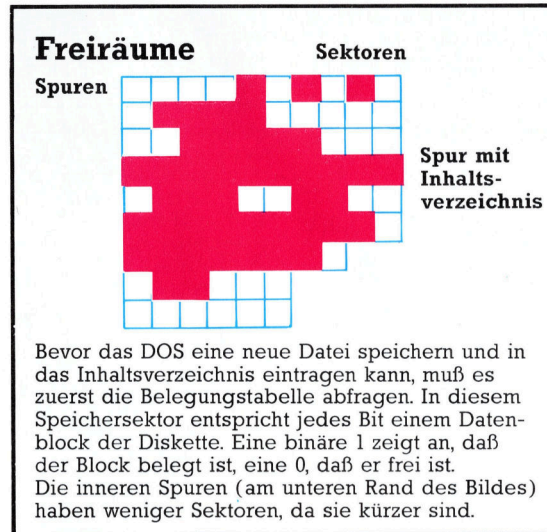


„sich an den eigenen Schnürsenkeln herausziehen“, vergleichbar mit Münchhausens „am eigenen Zopf aus dem Moor ziehen“. Bootstraps enthält eine Minimalversion des DOS. Seine Aufgabe ist, das eigentliche DOS auf der Diskette zu finden und es in den RAM-Bereich zu kopieren. Dieser Vorgang, der beim Anschalten des Computers automatisch abläuft, wird „booten“ oder „Urstart“ genannt. Danach zeigt ein Informationstext und/oder prompt auf dem Bildschirm an, daß der Computer jetzt arbeitsbereit ist.

Nun zur eigentlichen Funktion des DOS. Es wurde bereits erwähnt, daß eine Diskette in konzentrische Ringe (Spuren) und Sektoren unterteilt ist. Die Fläche, die durch die Kreuzung einer Spur und eines Sektors entsteht, wird Block genannt. Sie enthält normalerweise 128 Bytes. Ein Block ist die kleinste Einheit, in der Daten auf einer Diskette untergebracht werden können. Eine der Hauptgründe für die Existenz des DOS ist die Notwendigkeit, genau zu wissen, wo auf der Diskette welche Informationen gespeichert sind. Ein Beispiel: Die Diskette hat eine Speicherkapazität von 320 KByte – genug Platz für 20 Programme mit je 16 KByte. Ohne DOS müßten für das Laden eines

oder Belegungstabelle genannt). In dieser Liste wird jeder Block von einem einzelnen Bit dargestellt, das anzeigt, ob er bereits vergeben ist oder nicht. Wird ein Block beschrieben, ändert sich auch der Wert des Bits von 0 auf 1.

Das DOS speichert seine Dateien nicht –



wie zu erwarten wäre – in fortlaufenden, benachbarten Blöcken ab, sondern auf eine besondere Weise. Nehmen wir einmal an, eine Spur besteht aus zwölf Sektoren, die im Uhrzeigersinn von eins bis zwölf durchnummeriert sind. Die ersten 128 Bytes eines Programms würde man im Sektor eins finden, die zweiten in Sektor sieben, die dritten im Sektor zwei und so weiter. Der Grund für diese eigenartige Schreibmethode liegt darin, daß die Daten, bevor sie auf Diskette geschrieben werden können, erst in einem Buffer blockweise bereitge-

Inhaltsverzeichnis

Dateiname	Art	Speicherort (Spur-Sektor)
Invaders	Prog	20-1, 20-7, 20-2...
Temperat	Prog	25-11, 26-5, 26-12...
Plan	Prog	23-12, 24-3, 24-9...
Plandat	Daten	27-1, 27-7, 27-2...

Das Inhaltsverzeichnis einer Diskette liegt normalerweise auf der mittleren Spur. Es enthält die Namen aller Einträge, ihre Art (Programm, Daten oder anderes) und die Nummern der Spuren und Sektoren, auf denen die Daten gespeichert sind.

einzigsten Programms die Spuren- und Sektorennummern von 128 verschiedenen Blöcken à 128 Bytes angegeben werden.

Als „Gedächtnisstütze“ legt sich das DOS ein Verzeichnis (Directory) vom Inhalt der Diskette an. Es liegt normalerweise auf der mittleren Spur der Diskette, da es oft gebraucht wird und diese Position dem Schreib/Lesekopf den kürzesten Weg ermöglicht. Die Schnelligkeit eines Diskettenlaufwerks hängt weniger von der Umdrehungsgeschwindigkeit einer Diskette ab, als davon, wie schnell sich der Schreib/Lesekopf von einer Spur zu anderen bewegen kann. Das Inhaltsverzeichnis enthält eine Liste aller Daten (und/oder Programme), die auf der Diskette gespeichert sind, mit den Namen der Einträge, Angaben über die Art der Daten sowie einer Auflistung aller Blöcke (mit Spur- und Sektorennummer), in denen sich diese Daten befinden.

Soll eine neue Datei angelegt werden, liest das DOS zunächst die Liste der freien Sektoren (auch Block Availability Map – BAM –



Einige Diskettenstationen arbeiten mit eigenem Mikroprozessor und Arbeitsspeicher. Sie werden als „intelligent“ bezeichnet und enthalten das DOS in Form eines ROM-Chips. Bei „unintelligenten“ Laufwerken ist das DOS im Computer untergebracht.

stellt werden. Das DOS müßte zwischen jedem Schreibvorgang eine vollständige Diskettenumkehrung abwarten, wenn die Sektoren fortlaufend beschrieben werden sollten. Dadurch würde sich der ganze Speichervorgang stark verlangsamen. Auf einer Diskette, deren Dateien oft gelöscht und neu geschrieben wurden, sieht die Belegungstabelle wie ein „Schweizer Käse“ aus, bei dem sich neue Dateien auf die verbleibenden „Löcher“ verteilen müssen.



ZX 81-Zubehör

Mit entsprechender Peripherie wird der Sinclair-Micro zum hochentwickeltesten System.

Mit den entsprechenden Peripheriegeräten kann der ZX 81 zu einem kompletten Microcomputersystem ausgebaut werden. Verfügbar sind hochauflösende Farbgrafik, Sprachsynthese und Datenfernübertragung. Das Grundgerät hat einige Nachteile, die jedoch mit Zusatzgeräten ausgeglichen werden können; so beispielsweise mit einer professionellen Standardtastatur, Erweiterungen des Arbeitsspeichers (RAM) und programmierbaren Joystickcontrollern.

RAM-Erweiterung

Standardmäßig verfügt der ZX 81 nur über ein KByte RAM, von denen bereits 123 Bytes für Systemvariablen reserviert sind. Von Sinclair selbst wird ein Erweiterungsmodul mit 16 KByte angeboten (hier im Bild). Andere Hersteller bieten jedoch größere Kapazitäten, mit denen sich der ZX 81 bis auf 64 KByte (Cheetah-Version) aufstocken läßt.

Forth-ROM

Die ZX-Microcomputer von Sinclair verfügen über eine eigene BASIC-Version. Obwohl es nicht möglich ist, diesen BASIC-Dialekt gegen einen anderen auszutauschen, kann man die Programmiersprache durch eine andere ersetzen, beispielsweise Forth. Zwei Möglichkeiten bieten sich an: Die neue Sprache kann von einer Cassette in den RAM-Speicher geladen werden – mit dem Nachteil, daß bei jedem RESET oder nach jedem Abschalten der Computer wieder auf seine BASIC-Version zurückspringt. Oder man tauscht das BASIC-ROM gegen ein anderes aus. David Husband entwickelte diese ROM-Version von Forth, mit der zehn oder mehr Programme ablaufen können.

Weiterhin lieferbar...

Es gibt noch eine Reihe weiterer Möglichkeiten, den ZX 81 auszubauen. Eine Farbkarte ermöglicht die Programmierung von bis zu 16 Farben, und ein zusätzlicher Tongenerator läßt drei programmierbare „Stimmen“ erklingen. Bidirektionale Ein- und Ausgänge unterstützen die Steuerung von bis zu 16 Ein- und Ausgabegeräten. Als kleinster Sinclair Micro läßt sich der ZX 81 bis auf das volle Potential seines Z80-Microprozessors ausbauen.

Akustikkoppler

Es gibt zwei Arten von Modems: Einige benötigen einen zusätzlichen Steckkontakt für die direkte Einschaltung in das Telefonnetz, während Akustikkoppler wie der hier abgebildete MicroMyte 60 die Impulse unmittelbar über den Hörer senden und empfangen. Direkt angeschlossene Modems empfangen ihre Daten als elektronische Impulse, die Nullen und Einsen darstellen. Sie sind im allgemeinen teurer. Akustikkoppler dagegen übersetzen die Nullen und Einsen ihrer Daten in hörbare Töne, die über das Telefonnetz gesandt oder empfangen werden.



computer kurs

Computer Welt

Heft	Seite
1 Die Micro-Revolution	1
2 Vom Abakus zum Microchip Micros ganz groß im kleinen Betrieb	29
3 Expertensysteme Millionen von Schaltern	52
4 Computerkarrieren Sir Clive Sinclair	57
Strich-Codes und Geldautomaten	80
5 Miniaturtechnik auf Sand gebaut Chips im Haus	85
Micro-Medizin	102
6 Die Klangzauberer Stephen Wozniak	109
Chips im Auto	113
7 Computerpiloten John von Neumann	116
8 Im Spielfieber Analoge Systeme	132
9 Computerfilme 3-D-Computerspiele	141
10 Verkabelte Nachbarn Schöne Aussichten Der Computerpionier Chuck Peddle	144
11 „Kollege“ Roboter Pinball Wizard	160
12 Verkehrscomputer Präzise Greifarme Herman Hollerith	169
	190
	197
	216
	225
	246
	253
	272
	274
	281
	304
	309
	328
	330

Hardware

Heft	Seite
1 System komplett Ein Boom ohne Ende? Das Kleingedruckte Commodore 64	4
2 Kommunikation per Fingerdruck Atari 400 & 800	6
3 Texas Instruments 99/4A Monitore im Blickpunkt	8
4 Die Reisecomputer Sinclair Spectrum	11
5 Auf die Perspektive kommt es an BBC Modell B	32
6 Apple IIe	35
7 Dragon 32	59
8 Lokale Netzwerke Sinclair ZX 81	62
9 Speichertechnik im Umbruch Sord M5	88
10 Daten im Warteraum VC 20	93
11 Epson HX-20 Magischer Kristall	118
12 Sinclair QL	119
	145
	172
	200
	202
	228
	230
	256
	258
	287
	290
	315

Tips für die Praxis

Heft	Seite
1 Computerkauf mit Köpfchen	9
2 Tasten und Typen	34
3 Gestochen scharf und flimmerfrei Welches Programm ist das richtige?	64
4 Es muß nicht der Teuerste sein	71
5 Kein Anschluß ohne FTZ!	90
6 Vielseitigkeit durch Steckkarten	138
7 Schreibmaschine oder Drucker?	148
8 Der „Mini“ mit den vielen Extras	174
9 Wir stellen vor: Sound und Grafik	204
10 Sound-Begriffe und Grafik-Spiele	240
11 Der tönende VC Der zeichnende Dragon	264
12 Leichte Muse und Grundmuster	292
	293
	318

Software

Heft	Seite
1 Ohne Software läuft nichts	14
2 Phantastische Reise durch neue Welten Ein Mosaik aus tausend Punkten	38
3 Auftrag verstanden, Programm läuft!	40
Text perfekt	66
4 Malen nach Zahlen Der Code-Knacker	68
5 Der Weg zur Lösung Daten auf Abruf	96
6 Spritzige Sprites	98
7 Budgetplanung am Bildschirm Digitaler Dialog	122
8 Dolmetscher für Maschinen- sprache Software-Piraten	124
9 Geordnete Dateien	150
10 Modellverhalten Gut sortiert ist halb gewonnen	176
11 Das Programm-Genie Micro-Labyrinth	179
12 Schwarzes Brett Fernspiele	206
	208
	234
	260
	262
	284
	300
	312
	314

BASIC

Heft	Seite
1 Auf die Anrede kommt es an	18
2 Loopings mit der FOR. . .NEXT- Schleife	42
3 Auf Punkt und Komma genau	72
4 Reine Routine	100
5 Der BASIC-Kalender	126
6 Spiel mit Elementen	157
7 Ordnung ins Programm bringen	186
8 Arithmetische und String- Funktionen	212

9 Dem puren Zufall ausgeliefert	242
10 Neue Dimensionen	268
11 Unter Kontrolle	296
12 Datenverwaltung	324

Peripherie

Heft	Seite
1 Cassetten als Datenspeicher	22
2 Die Tempomacher	46
3 Zeile für Zeile sauber gedruckt	74
4 Diskettenstationen	104
5 Auf zwei Rädern	129
6 Plotten mit Pfiff	154
7 Schnittstellen	181
8 Klein, aber oho!	210
9 Das Grafiktablett	238
10 Mäuse im Einsatz	266
11 Der Zauberstab	294
12 Dem DOS auf der Spur ZX 81-Zubehör	320
	322

LOGO

Heft	Seite
1 Aller Anfang ist schwer — dieser nicht	24
2 Programmieren mit Prozeduren	49
3 Farben, Formen, Zahlenspiele	77
4 Eine Wiederholung, die es in sich hat	106
5 Bogen- und Listenverarbeitung	134
6 Input/Output und Joystickabfrage	162
7 Entwickeln Sie eigene Prozeduren	192
8 Bauen Sie den „Turm von Hanoi“	219
9 Farb-„DEMO“ und LOGO- Telefonbuch	248
10 Relativ oder absolut?	275
11 Stück für Stück . . .	302
12 Turtle-Variationen	331

Bits und Bytes

Heft	Seite
1 Keine Angst vor Bits und Bytes!	27
2 Ein Gedächtnis, das nichts vergißt	55
3 Computerlogik	83
4 Gatter und Addierer	111
5 Wenn 1 + 1 = 10 . . .	139
6 Der Sprachkünstler Richtig aufbewahrt	165
7 Von Minus zu Plus	167
8 Die Schaltzentrale	195
Multiplizieren im Binärsystem	222
9 Gesetze des Denkens PEEK und POKE	224
10 Richtig adressiert Rechnen mit Basis 16	250
11 Detektivarbeit	252
Auf Fehlersuche	278
12 Ändern und Löschen Zeichengenerierung Schritt für Schritt	280
	306
	308
	334
	335
	336

Index Band 1

A	Seite
Abakus	29—30, 280
Ablaufskizze	66
Acorn B	119—21, 316
ACT Apricot	89
Addierer und, Gatter	111—12
Adreßbuch-Programm	324—7
Adresse: Adressierung	278—9
-Decoder	279
-leitungen	179—80, 222
ADSR, siehe Hüllkurve	
AI, siehe Künstliche Intelligenz	
Ajile Hyperion	89
Akkumulator	223
Aktenkoffer-Computer	88—9
Akustikkoppler	3, 138
mit Sinclair ZX-81	322
und Modem	185, 312
Algorithmus	226
Sortier-	262—3
ALU (Arithmetik-Logik-Einheit)	222
Analog/Digital-Wandler	131, 179
analoger Computer	216—17
analoger Eingang	182
Animation	150—2, 225—7
Anwenderfreundlichkeit	305
Apple	144, 274
Analogsynthesizer	143
Apple II	148—9
Apple IIe	145—7, 148
Lisa	284—6, 305
Macintosh	316
Thermodrucker	70
Visicalc	176
Arcadespiele	197—8
Arithmetik-Logik-Einheit (ALU)	222
ASCII	8, 298
Assembler	14, 98, 150
Atari: 400	35—7
600 XL	6
800	35—7, 70, 143
800 XL	6
Peripherie	36
Spielcomputer	198
Sprite-Grafik	150, 152
Ausgabe, siehe E/A	
Auto: Bordcomputer	160—1

B	Seite
Babbage, Charles	29, 30—1
Bahnverkehrscomputer	310—11
Bardeen, John	81
BASIC	8
Anfängersprache	18—21
Befehle	21, 100
Datenverwaltung	324—7
FOR...NEXT-Schleife	42—4
Funktionen	212—15, 242—5
Interpunktio	72
Kontroll-Strukturen	296—9
Microsoft (MBASIC)	242, 293
Programmaufbau	186—9
Subroutinen	100—1, 186
Synthese	143
Variablen	19, 20, 157—9
Weihnachtsprogramm	126—8
Baud-Rate	23, 184
und Modem	312
und Drucker	76
BBC Buggy	129, 131
Modell B	6, 70, 119—21
Peripherie	120
und Bodenroboter	129
Bedienungspersonal	86
Befehle	278
Betriebssystem	98, 320
Bildschirm, siehe Monitor	
Bildschirm-Editor	334
Bildwiederholtspeicher	41
Binärsystem	27—8, 139—40, 224
-code	336
Bit	27—8
Paritäts-	308
Bitmuster (Bit mapping)	319
Bit-Pad-Digitiser	227
Boole, George	250
Bootstrap	320—1

Brattain, Walter	81
Briefkasten, siehe Schwarzes Brett	
Bubble Sort	262
Bubble-Speicher, siehe Magnetblasen-Speicher	
Buchhaltung	10, 17, 54
Budge, Bill	304
Buffer	153, 256—7
FIFO-	256
Hardware-	257
Signal-	257
Büro, Computer im	3, 15, 16, 52—4, 284—6
Bushnell, Nolan	197
Busse	181
Byte	27—8

C	Seite
Cartridge	14
Casio FP200	91
FX700P	88
Synthesizer	143
Casiotone VL1	143
Cassette	14
Digital-	211
Cassettengerät: Atari 410	36
Ikon Hobbit	22, 211
Micro-	287, 288
Cassetteninterface	137, 182, 184
Cassettenrecorder	2, 4, 5, 9, 12
als Datenspeicher	22—3
Centronics Schnittstelle	76, 82, 181, 184
Cheetah Sweet Talker	323
Chip	1
6809E	173
7408	84
8255PPI	180
E/A Schnittstellen-	180
Herstellung	113—15
im Haus	116—17
Colne Robotics Zeaker	129
Colossus	31
Colour Genie	6
Commodore	274
-LOGO	275—7
Personal Electronic Transactor (PET)	258, 274
Commodore 64	6, 11—13
Peripherie	12
Sprite-Grafik	11, 150, 152
Synthese	143
Tastatur	34
Textverarbeitung	70
Ton	264
Commodore VC 20	6, 7, 258—9
Grafik	319
Peripherie	258
Sounderzeugung	292
Super Expander Cartridge	319
Textverarbeitung	70
Compiler	98, 103, 206—7
Computer Aided Design (CAD)	3
mit Lichtgriffel	295
Computer Generated Imaging (CGI)	170, 171
Computerfilme	225—7
Computerpiloten	169—71
Computerspiele, siehe Spiele	
Computertomographie	133
CP/M (Control Program for Microcomputers)	10
CPU (Zentraleinheit)	4, 8, 222—3
Adressierung	278—9
Cursor und, Maus	266—7

D	Seite
Darstellung	8
Data latch	179—80
Dateien: Direktzugriff	234
sequentielle	234
Daten: Aufbereitung	334
-bus	222
-Fernübertragung	137—8
Ordnen von	234—7, 262—3
Speicherung von	4, 22—3
-verwaltung	15, 17, 324—7
Datenbank	17
-Programme	124—5
Dateneingabe-Personal	86

Datensysteme: Sicherheit	103
Datenübertragung	233
Fehlersuche	308
Hamming-Codes	306—7
parallele	181
Difference Machine	30
Digitalcassette	211
Digitiser	238
Disk Operating System (DOS)	320—1
Diskette, siehe Floppy Disk	
Disketten-Interface	182
Disketten-Laufwerk	2, 3, 4, 5, 104—5
Apple IIe	146
Atari 810	36
BBC Modell B	120
Commodore-1540	12
Dragon 32	172
intelligente	200
Lesekopf	105
Winchester	200
Diskettenverwaltungssystem (DOS)	320—1
Dodgson, Charles	250
Dongle	209, 315
DOS, siehe Diskettenverwaltungssystem	
Dragon 32	6, 7, 172—3
Grafik	293
Synthese	142
Textverarbeitung	70
Drahtgittermodell	118
Drucker	2, 3, 5, 15, 74—6
Ball-Pen-	76
-Buffer	256—7
-Interface	182
Matrix-	75, 204
Nadel-	174—5
Papierzuführung	233
Textverarbeitung	70
Thermo-	76
Tintenstrahl-	75, 76
Typenrad-	75, 200
Zeilen-	75
Duplex	138
-Betrieb	233

E	Seite
E/A (Ein- und Ausgabe)	179—80
Adresse	279
-Schnittstellen-Chip	180
Econet	201, 254
Editor	334
Eingabe, siehe E/A	
Eingang, analoger	182
Einsatz-Sort	262—3
Elektronenröhre	81
Elektronik-Ingenieur	86, 87
Emulationsprogramm	312
ENIAC	31, 80, 190
EPROM (Erasable Programmable Read Only Memory)	146, 168, 201, 316
Epson HX-20	6, 7, 88, 89, 211, 287—9, 290
Tabellenkalkulation	176
Textverarbeitung	70
Epson PX-8	92
Exatron Microdrive	211
Expansion-Box	60
Expertensysteme	57—8

F	Seite
Fairlight Synthesizer	142
Fakturierung	54
Farbe	226
Fehler: -suche	308
Hamming-Code	306—7
Felder, zweidimensionale	268—71
Fenster	233
Fernseher als Monitor	64
Kabel-	253—5
TV-Ausgang	183
Video-Ausgang	183
Firmware	16
Flip-Flop	229
Floppy Disk	5, 10, 14, 104, 320
DOS	320—1
Inhaltsverzeichnis	321
Micro-Floppy	153
RAM-Disks	90, 149
Floppy tape	211

Index Band 1

Flugzeug: Autopiloten	217
Flughafen-Kontrollsystem	309—10
Flugsimulator	169—71
Wartung	309
Flußdiagramme	122—3
Flüssigkristallanzeigen (LCD)	290—1
FORTH	18, 153
für Jupiter Ace	6
-ROM	322
FTZ-Nummer (Fernmelde-Technische-Zulassung)	138
Fünfte Generation	82
Funktionen, numerische	212—13
String-	213—15

G	Seite
Gates, Bill	274
Gatter, Exklusiv-Oder-	306
-Schaltungen	83—4
und Addierer	111—12
Gedächtnis	55—6, 167—8
siehe auch Speicher	
Geldautomaten	110
Generationen	82
Geschichte	1—3, 29—31
Grafik	40—1, 96—7
Block-	96—7, 241
Computerfilme	225—7
3D-Effekte	151
Erzeugung von	241
Gitter-	118
hochauflösende	97, 241
mit Commodore VC 20	319
mit Dragon 32	293
mit Joystick	218
mit LOGO	24—6
mit PEEK und POKE	265
Plotter	154—6
Sprite-	11, 97, 150—2, 225—6, 241
Vektor-	118
Grafiktablett	3, 97, 218, 238—9
Gray-Code	336
Greifarme	328—9

H	Seite
Hacker	103
Hamming-Codes	306
Handheld-Computer	89, 90—2
Hardcopy-Routine	218
Hardware, siehe Peripherie	
Haus, Mikroprozessor im	116—17
Hewlett Packard Sweetlips Plotter	154
Heimcomputer-Systeme	2—3
Hexadezimalzahlen	98, 280
Hobbit	22, 211
Hollerith, Herman	330
Horoskope, Computer-	103
Hüllkurve (ADSR)	143, 264

I	Seite
Ikon Hobbit	22, 211
integrierte Schaltung	29, 31, 82
Intel-2114	168
-2364	167
Interface, siehe Schnittstelle	
International Business Machines (IBM)	330
Interpreter	67, 98, 206
Interrupts	180

J	Seite
Jobs, Steve	144
Joystick	2, 5, 15, 46—8
-Abfrage mit LOGO	164
analoger	47
Anschluß für	182
digitaler	46
-Interface	48
Paddle	48
Zeichnen mit	218
Jupiter Ace	6, 7

K	Seite
Kabelfernsehen	253—5
Karrieren in der EDV	85—7
Kelvin, Lord	216
Kernspintomographen	133
Kilby, Jack	115
Klang, siehe Ton	
Koaxialkabel	65, 254, 255
Konstrukteure	87
Kontroll-Block	222
-summe	306
künstliche Intelligenz (AI)	3, 57—8, 131

L	Seite
LED-Anzeige	179
Leibniz, Gottfried Wilhelm	224
Lichtgriffel (Lightpen)	2, 294—5
Lochkarten	330
Logik	83—4
Gatter-Schaltungen	83—4
Gesetze	250—1
Transistor-Transistor- (TTL)	257
Wahrheitstabellen	84
LOGO: Befehle	25, 77, 107
Begriffe	51
Commodore-	275—7
Grafik	24—6, 77—8, 134—5, 248—9
Hilfsprogramme	220—1
Input/Output	162—3
Joystick-Abfrage	164
Listenverarbeitung	135—6
Prozeduren	49—51, 192—4
Recursionen	106—8, 219
Tangram-Puzzle	302—3
Telefonbuch	249
Variablen	50—1, 331—3
Zahlenspiele	79
Lovelace, Ada	30

M	Seite
M-Coder	205
Magnetblasen-Speicher	89, 90, 229
Maschinencode	67, 98—9
Interpreter & Compiler	206—7
M-Coder	205
Matrix	235, 271
Maus	3, 33, 48, 266—7, 284
Medizin, Computer in der	132—3
Menü	294
-gesteuertes Programm	325
Microcassette	211, 287, 288
Microchip-Revolution	1—3
Microdrive	9
Sinclair	210—11
Microsoft-BASIC (MBASIC)	242
Flugsimulator	170, 171
Microwriter	89
Modelle	260—1
Modem	3, 137, 312
als Akustikkoppler	185
Halbduplex	138
Voll duplex	138
Monitor	2, 3, 5, 41, 62—3
Auflösung	40
Composit-Standard	62
Flüssigkristallanzeige	290—1
Funktionsweise	40
Qualität	64—5
RGB	62
MOS Technology-6502	274
MSX (Microsoft Extended BASIC)	218, 232
Musiksynthese	141—3

N	Seite
NCR Decision Mate	54
NEC PC-8201A	92
Netzwerk, lokal	200—1
Econet	201, 254
Signalverlust	253
Neumann, John von	31, 190—1
New Brain: Textverarbeitung	70

O	Seite
Olivetti M10	91
Oric I	7, 143
MCP-40	154
Osborne I	89, 176
Executive	89
Oszillatoren	240
Oughtred, William	216
Overlay-Verfahren	176

P	Seite
Pantograph	239
PASCAL	18
Pascal, Blaise	30
Peddle, Chuck	274
PEEK	252
und Grafik	265
Peripherie	8
Cassettenrecorder	22—3
Disketten-Laufwerk	104—5
Drucker	74—6
Grafiktablett	238—9
Joystick	46—8
Lichtgriffel	294—5
Maus	266—7
Modem	137, 312
Monitor	62—3
Plotter	154—6
System komplett	4—5
Tastatur	32—3
Track Ball Controller	5, 47, 48
und E/A	179—80
Philips Micro-Digitalcassette	211
Phoneme	133, 165
Piraten	208—9
Pixel	97, 151, 241
Planimeter	216
Planungslisten	17, 176—8
Planzeichner, siehe Plotter	
Platine	29, 138
Plotter	3, 154—6
Flachbett-	154, 156
Rollen-	154
Tintenstrahl-	156
Trommel-	154
POKE	252
und Grafik	265
Portables: Aktenkoffer-Computer	88—9
Epson HX-20	287—9
Handheld-Computern	90—2
Portico Miracle	89
Potentiometer	46
-schaltkreis	47
Prestel-Mailbox	253
Programm: Aufbau von	186—9
Echtzeit-	131
Emulations-	312
integrierte Pakete	185
Kalkulations-	176—8
-Pakete	16
Raubkopien	208—9
Roboter-Steuerungs-	329
-Zähler	223
Programmierbare Peripherieschnittstelle (PPI)	180
Programmierer	85—7
Programmiersprache	14
Assembler	14, 98, 150
FORTH	6, 18, 153
PASCAL	18
Script	142
Simulation mit	261
siehe auch BASIC; LOGO	
Programmierung	44, 66—7
Flußdiagramme	122—3
objektorientierte	284—6, 305
strukturierte	66
Subroutinen	100—1
Top-Down-	324
siehe auch BASIC; LOGO	
Punktmatrix	40

Q	Seite
Quantels Paint Box-System	226
Quecksilberverzögerungsleitung	228

Index Band 1

R	Seite
RAM (Random Access Memory)	8, 55, 167—8, 229
-Disk	90, 149
Erweiterung	322
Raubkopien	208—9
Rauschen, siehe Ton	
Receive Data-Leitung	183—4
Rechenmaschine	29—30, 111, 216
Rechner, siehe Computer	
Redundanz	307
Register	222
Relais	80
Roboter	45, 82
Boden-	129—31
Fertigungs-	281—3
Greifarm-	328—9
-Steuerungsprogramm	329
Tisch-	328
Zeichen-	24, 39, 131, 277
ROM (Read Only Memory)	8, 14, 55, 167—8
EPROM	146, 168
Rushent, Martin	143

S	Seite
Sampling	141
Schaltkreise	1
integrierte Schaltung	31, 115
Schichtaufnahme, computergespeicherte (CT)	133
Schildkröte	129, 131
Schnittstelle	8, 174, 181—4
analoge	129—31
Cassetten-	137, 182, 184
Centronics-	76, 175, 181, 184
Disketten-	182
E/A-	180
IEEE	183
intelligente	76
parallele	181, 182
programmierbare Peripherie-	180
RGB-	183
RS-232/V24	138, 183, 184
Schreibmaschinen-	175
serielle	183
Schreibmaschine, elektronische	174—5
Schrittmotor	129, 155, 282
schwarzes Brett	253, 312—13
Scrolling	334
Sequenz	142
Shannon, Claude	80
Sharp CE-150	155
MZ-700	6
PC-1251	88, 211
PC-5000	89, 90, 92, 290
Shockley, William	81
Silicon Valley	45
Silizium	113—14
siehe auch Chip	
Simplex-Betrieb	233
Simulation	260—1
Sinclair, Sir Clive	94, 102
Sinclair Microdrive	210—11
Sinclair QL	102, 315—17
Sinclair Spectrum	33, 93—5
als Portable	89
Peripherie	93, 94, 153
Textverarbeitung	69—70
Tonerzeugung	318
Sinclair ZX-81	6, 7, 202—3
Akustikkoppler	322
FORTH-ROM	322
RAM-Erweiterung	322
Soundbox	204
Sprachsynthese	323
Tastatur	32, 205, 323
Textausgabeprogramm	69
Zubehör	204—5, 322—3
Singleton, Mike	247, 314
Sirius-1	166, 274
Software	14—17, 45
Definition	4
-Piraten	208—9
Programm-Paket	16
Selbstschreiben von	324
siehe auch Programm	
SORD IS11	92

SORD M5	150, 152, 230—2
-Grafik	231
-ROM-Cartridge	230
Sortiersysteme	262—3
Sound, siehe Ton	
Speicher	4, 8, 9
Adresse	55—6, 179, 278—9
Bildwiederhol-	41
Bubble-	89, 90, 229
CPU	222
Erweiterungsanschluß	182
Flip-Flop	229
Kathodenstrahlröhre	228—9
Kern-	228
kryogenische	229
Kurzzeit-	256—7
Laser-	229
Quecksilberöhre	228
RAM- und ROM-	167—8
-zelle	28, 55—6
Zwischen-	153, 179
Speicherung von Daten	4
Geschichte	228—9
Spiel	17, 38—9
Arcade-	197—9
Animation	150—2
3-D-	246—7
Fern-	314
Flugsimulator	170
Labyrinth-	300—1
Pinball Construction Set	304—5
-system	2
Verpackungen	15
Spielautomaten	118
Spielzeug-Computer	9, 198
Spooler	153
Spracherkennung	218
Sprach-Synthesizer	60, 165—6
digitalisierte Sprache	166
im Auto	160
Phoneme	133, 165
Sweet Talker	323
Spreadsheet	176—8
Sprite-Grafiken	11, 97, 150—2
Steckkarten	148—9
Steuernknüppel, siehe Joystick	
Straßenverkehrscomputer	310, 311
Stich-Code	109—10, 142, 143
Stringy floppies	211
Strobe 100 Plotter	154
Subtraktion mit Zweierkomplement	195—6
Synclavier Synthesizer	142
Synthese, Musik-	141—3
Sprach-	165—6
Synthesizer	141—3
Systemanalytiker	66, 86, 87
System, offen und geschlossen	260

T	Seite
Tabellen	176—8, 268—71
Tachometer	216
Tandy	172
CGP-115	154
Modell 100	92
TRS-80	211, 290
Taschenrechner	88
Tastatur	4, 8, 32—3, 34
Folien-	32
Membran-	33
QWERTY/QWERTZ	184
Schreibmaschinen-	32
Taschenrechner-	32
technische Daten	8
Telefax	137, 153
Teletex	153
Teletype-Maschine	266
Telex	153
Texas Instruments Sprach-Synthesizer	165
Spracherkennung	218
TI 99/4A	59—61, 97, 150, 152
Textverarbeitung	10, 15, 16, 54, 68—70
Microwriter	89
mit Handheld-Computern	90
-Programme	70, 71
Ton: mit Sinclair Spectrum	318
mit VC 20	292
Rauschen	264
Sound-Begriffe	264

Sound-box	204
Sound-Erzeugung	240
Synthesizer	141—3
Track Ball Controller	5, 47, 48
Track bouncing	142
Transistor	80—1
Transmit-Data-Leitung	183—4
Trickfilm-Techniken,	
computergesteuerte	170, 171
Computerfilme	225—7
TTL (Transistor-Transistor-Logik)	257
Turing, Alan	31
Turtle	129, 131
Hebot-	323

U	Seite
Uhr, interne	99
ULA (Uncommitted Logic Array)	202
Unterbrecherschaltungen (interrupts)	180

V	Seite
Variablen: einfache	268
Index-	157—9
mit BASIC	19, 20
mit LOGO	50—1
verschachtelte	268
Vectrex	118
Venn, John	250
Venn-Diagramme	251
Verkehrsregelung	14, 309—11
Verwaltung, siehe Daten	
Video	8
-Controller	294—295
-Spielautomaten	118
Viewdata System	253
Visicalc	176
Votrax	166

W	Seite
Wahrheitstabellen	84
Wettervorhersage	272—3
Whitted, Turner	246
Williams, F. C.	288
Windowing	170
Wozniak, Stephen	30, 144

Y	Seite
Yamaha-Synthesizer	142

Z	Seite
Zahlen, Binäre	27—8, 139—40, 224
Hexadezimale	98, 280
randomisierte	140, 191
Zufalls-	191, 242—5
Zeaker	129
Zeichengenerator	151, 335
Zentrale Steuereinheit (CPU)	4, 8, 222—3
Adressierung	278—9
Zugriff, direkt und sequentiell	234—5
zweidimensionale Felder	268—71
Zweierkomplement	195—6
Zwischenspeicher	153

Alle zwölf Hefte erscheint ein solcher Teilindex. Der Gesamtindex erscheint mit dem letzten Heft von COMPUTER KURS — darin einbezogen sind Kreuzverweise auf die Artikel, die mit dem gesuchten Stichwort in Verbindung stehen.



Joystick-Controller und Joystick

Sinclair bietet keine eigenen Joysticks und geeignete Controller an. Es gibt auf dem Markt aber eine breite Palette dieser Steuerungen, die entweder mit Steuerzeichen programmierbar sind oder die ausgegebenen Zeichen vorschreiben. Das abgebildete Modell der Firma AGF Hardware wird durch das Umstecken von Verbindungskabeln programmiert. An den ZX 81 lassen sich Track Balls und Joysticks, die nach dem Schalterprinzip funktionieren, anschließen.

ZX-Drucker

Der ZX-Drucker von Sinclair benötigt aluminiumbeschichtetes Papier, das auf Elektrizität reagiert. Entgegen der herkömmlichen Druckweise entfernt der Druckkopf die Aluminiumschicht, so daß die dunklere Papieroberfläche darunter zum Vorschein kommt. Obwohl die Druckgeschwindigkeit akzeptabel ist, stellen die Verfügbarkeit des ungewöhnlichen Papiers und die Papierbreite ein Problem dar. Über zusätzliche Schnittstellenkarten können jedoch auch normale Drucker angeschlossen werden. Es gibt Karten für die Schnittstellen RS232 und Centronics.

Hebot-Turtle

Standardmäßig mit Steuersoftware ausgerüstet, kann die Schildkröte mit Zusatzgeräten ausgestattet werden, wie Fotosensoren, die es dem Roboter ermöglichen, über ein reflektierendes Klebeband einem vorbestimmten Weg zu folgen.

Sprachsynthese

Ein weiteres interessantes Zusatzgerät ist der „Sweet Talker“ für Sprachsynthese von Cheetah. Der Sweet Talker funktioniert nach dem allophonischen System und ist daher einfacher zu programmieren als Module, die mit dem phonemischen System arbeiten. Allophone sind Gruppen gleichklingender Phoneme.

Tastatur

Die Membrantastatur ist eines der negativen Merkmale des ZX 81. Es überrascht daher nicht, daß mehrere Firmen dafür herkömmliche Tastaturen anbieten. Die hier abgebildete Mapsoft-Tastatur von Maplin Electronics ist als Bausatz oder komplett erhältlich. Die Mapsoft-Tastatur besitzt außer dem normalen Zeichensatz noch drei weitere Funktionstasten. Eine Alternative dazu bietet die Aufsatzastatur, die auf das Originalgerät gesetzt wird. Damit ist die Bedienung der Tastatur zwar einfacher, zusätzliche Funktionen sind aber nicht verfügbar.

Datenverwaltung

Die ersten Schritte zur Entwicklung eines Datenverwaltungsprogramms unter Einsatz aller bisher erworbenen Programmierkenntnisse.

Die bisher besprochenen Programme lösten jeweils eine bestimmte Aufgabe und dienten zur Demonstration der verschiedenen Möglichkeiten von BASIC. Doch es waren keine Programme, die man tagtäglich zur Arbeit verwenden konnte. Sie zeigten lediglich, wie man die einzelnen „Zahnräder“ von BASIC zusammensetzen kann, um einen einfachen Mechanismus zu erstellen.

Nun zu einem sinnvollen Einsatz Ihrer BASIC-Fähigkeiten. Ein herkömmliches Adreßbuch besteht meistens aus einem alphabetischen Index, der das Auffinden jedes beliebigen Namens gestattet. Die Namen werden normalerweise nach Bedarf eingetragen, wodurch eine exakte alphabetische Reihenfolge nie eingehalten werden kann. Ihre erste Eintragung unter P kann zum Beispiel Dirk Peterson sein. Zu einem späteren Zeitpunkt fügen Sie Karin Peters oder Elke Patel hinzu. Die Eintragungen sind also nicht alphabetisch geordnet.

Adreßbuch-Eintragungen

Es sind lediglich alle Namen mit dem gleichen Anfangsbuchstaben zusammengefaßt. Die anderen üblichen Eintragungen in einem Adreßbuch sind die Adresse, die Telefonnummer und eventuell noch einige personenspezifische Daten. Trotzdem kann Ihnen ein konventionelles Adreßbuch keine zusammengefaßte Liste aller Personen geben, die beispielsweise in Wiesbaden wohnen, oder den Namen nennen, zu dem die Telefonnummer 4 13 63 14 gehört.

Selbstverständlich braucht man solche Informationen als Privatperson kaum. Doch stellen Sie sich vor, Sie wären Besitzer eines kleinen Versandhandels. In einem solchen Fall kann es von Vorteil sein, wenn man derart spezifische Informationen aus der Kundenkartei abrufen kann. Nehmen Sie an, Sie hätten als neues Produkt Nachthemden für Kinder. Sie könnten nun Werbebriefe an alle Ihre Kunden schicken. Dies ist jedoch recht kostenintensiv. Wenn Sie die Werbung nur an Kunden mit Kindern schicken würden, könnten Sie erhebliche Portogebühren sparen und hätten denselben Werbeerfolg. Wie Sie sehen, sind dies Erwägungen, die angestellt werden müssen, bevor man entscheiden kann, ob es gerechtfertigt ist, ein Problem mit dem Computer zu lösen.

Ist es sinnvoll, fertige Programme zu kaufen? Beim flüchtigen Durchblättern einer Computer-

zeitschrift entsteht der Eindruck, daß jedes nur mögliche Problem bereits durchdacht und gelöst wurde. Eine nähere Betrachtung zeigt jedoch, daß ein kommerzielles Programm nicht das vollbringt, was Sie wollen, oder es ist für Ihren Computer nicht erhältlich, oder es ist viel zu teuer. Der Preis eines Programms spiegelt im allgemeinen die Entwicklungskosten wider. Ein Textverarbeitungsprogramm kann bis zu 1000 DM kosten.

Der Vorteil des Selbstschreibens von Software liegt jedoch klar auf der Hand: Sie können exakt bestimmen, was das Programm machen soll. Die Entwicklung eines Programms erstreckt sich über viele einzelne Abschnitte. Das erste ist, daß man sich über das Problem völlig im Klaren sein muß. Dies erfordert eine eindeutige Definition der Problemstellung.

Der zweite Schritt ist, daß Sie einen Lösungsweg finden müssen. Dies erfordert eine Beschreibung der Form, in der Eingaben und Ausgaben erwartet werden. Die Probleme und ihre Lösungen sollten in allgemein abgefaßten Sätzen festgehalten werden, die dann fortlaufend genauer ausformuliert werden, bis die Programmierung beginnen kann.

Der dritte Abschnitt ist die Codierung selbst. Hier wird die Programmiersprache BASIC verwendet, aber genauso gut könnte es jede andere Programmiersprache sein. Bis zu dem Zeitpunkt, an dem das Programm endgültig in BASIC programmiert wird, verwendet man eine Pseudo-Sprache, die die Flexibilität und Freiheit von Englisch bzw. Deutsch gewährt, gleichzeitig aber bereits die Struktur einer Computersprache (wie BASIC) berücksichtigt.

Die Art der Programmierung, die soeben beschrieben wurde, wird „Top-Down“-Programmierung genannt. Sie beginnt beim höchstmöglichen Niveau und führt von einer generellen Beschreibung der Ziele über fortlaufende Verbesserungen hin zu den endgültigen, notwendigen Details des Programms.

Die einzelnen Schritte zur Entwicklung eines Programms lassen sich wie folgt zusammenfassen:

1. Eine klare Definition des Problems
2. Die Form der Ein- und Ausgaben
 - 2.1 Verbesserungen
 - 2.2 Weitere Verbesserungen
3. Codierung in die gewählte Programmiersprache

Bevor man sich mit einem großen Software-Projekt beschäftigt, ist es unbedingt notwendig, das Problem klar zu fixieren. Diese Aufgabe ist gar nicht so einfach. Für das Computer-Adreßbuch starten Sie mit einer Liste aller wünschenswerten Funktionen. Zu einem späteren Zeitpunkt kann entschieden werden, welche dieser Funktionen in das Programm integrierbar sind.

1. Suchen einer Adresse, einer Telefonnummer und Anmerkungen durch Eingabe eines Namens über die Tastatur
2. Erstellung einer Liste von Namen, Adressen und Telefonnummern durch Eingabe eines Teils eines Namens (beispielsweise nur des Vornamens)
3. Erstellen einer Liste von Namen, Adressen und Telefonnummern für eine bestimmte Stadt oder ein Gebiet
4. Erstellen einer Liste aller Namen, die mit einem bestimmten Buchstaben beginnen
5. Erstellen einer Liste aller Namen des Adreßbuchs, alphabetisch sortiert
6. Hinzufügen beliebiger neuer Eingaben
7. Ändern beliebiger Eingaben
8. Löschen beliebiger Eingaben

Stellen Sie sich vor, das Adreßbuch-Programm wäre bereits geschrieben. Wie sollte Ihrer Meinung nach das Programm aus der Sicht des Anwenders gesehen arbeiten? In einem „menügesteuerten“ Programm wird dem Anwender immer dann, wenn er eine Entscheidung zu treffen hat, eine Liste (das sogenannte Menü) der Möglichkeiten zur Verfügung gestellt. Der Anwender kann dann normalerweise durch Drücken einer einzigen Taste seine Wahl treffen. In „befehlsgesteuerten“ Programmen muß der Anwender spezielle Befehls- worte oder -sätze eingeben. Einige Programme verwenden beide Techniken. Der Vorteil eines menügesteuerten Programms ist, daß ein Anfänger kaum Probleme haben wird – das Programm ist „anwenderfreundlich“. Einem versierten Anwender wird allerdings ein befehlsgesteuertes Programm lieber sein, da es meistens schneller arbeitet. In unserem Beispiel werden wir eine Menü-Steuerung vorsehen. Trotzdem können Sie für sich entscheiden, ob Sie nicht lieber entsprechende Befehls-Routinen einbauen wollen.

Da das Programm auf die Verarbeitung einer Liste von Namen ausgerichtet ist, müssen Sie zunächst einmal festlegen, welche Form diese Namen haben sollen. Sollte der Computer alle folgenden Formate akzeptieren?

. J. P. Schneider
Leonardo da Vinci
Gisela MAHRENHOLZ
Peter B.
P. O'Toole

e. e. fischer
K Bogmaier
Bobey
WALTER KALAUCH
Graf Wolfram von Eichborn

Stellen Sie sich vor, was passieren würde, wenn Sie K Bogmaier eingegeben hätten und dann das Programm nach K. Bogmaier suchen ließen. Sicherlich ahnen Sie schon, daß der Computer mit der Meldung NAME NICHT GEFUNDEN antworten würde. Zur Lösung dieses Problems gibt es zwei Wege: Zum einen könnten Sie beliebige Eingaben zulassen, wobei Namen in jeder denkbaren Form eingegeben werden können. Allerdings müssen dann intelligente Routinen erstellt werden, die das Suchen möglich machen. Zum anderen könnten Sie einen festen Rahmen zur Eingabe der Namen vorgeben. Jede Namenseingabe, die nicht genau der Vorgabe entspricht, hätte dann eine Fehlermeldung wie NAMENSFORMAT NICHT AKZEPTABEL zur Folge. Wir haben uns hier für eine unbeschränkte Eingabe entschieden und werden das Programm die Umwandlung in eine Standardform selbst vornehmen lassen.

Wenn man das Ganze vom Standpunkt eines alphabetischen Suchens aus betrachtet, können Namen aus zwei Teilen bestehen – dem Familien- und dem Vornamen. Ein Familienname scheint recht einfach definierbar zu sein: Jede beliebige Zeichenkette von großen und kleinen Buchstaben, die durch ein RETURN abgeschlossen und durch eine Leerstelle begonnen wird (ASCII 32). Sofort zeigt sich ein Problem: Was würde passieren, wenn der Name „Bobey“ ohne vorgestellte Leerstelle eingegeben wird? Sehr wahrscheinlich würde das Programm die Eingabe nicht akzeptieren. Die Definition muß folglich noch verbessert werden.

Eingabe von Namen

Ein Name kann einen Familiennamen oder einen Familiennamen und einen Vornamen umfassen. Der Name kann sowohl aus Groß- als auch aus Kleinbuchstaben bestehen, kann Punkte, Apostrophe und Bindestriche enthalten. Er beginnt immer mit einem Buchstaben und wird mit einem RETURN abgeschlossen (Punkte als Abschluß sind nicht gestattet). Be- findet sich vor dem RETURN eine Leerstelle, so wird die letzte Gruppe von Zeichen – eingeschlossen aller Apostrophe und Bindestriche – als der Familienname und die restlichen Zeichen, einschließlich der Leerstelle, als der Vorname angesehen. Gibt es keine Leerstelle, so wird der gesamte Name wie ein Familienname behandelt.

Der Familienname erfordert eine spezielle Handhabung, da er bei jedem alphabetischen Suchen eine vorrangige Stellung gegenüber

den Vornamen hat. So muß Albert Schmidt nach Zoltan Samberg aufgeführt werden. Besteht ein Name nur aus einer Gruppe von Zeichen, so wie Fexl, Bobey oder ein Rufname wie Susi, so kann er für den Zweck unseres Programms als Familienname behandelt werden.

Welcher Name würde bei einem alphabetischen Suchvorgang zuerst aufgeführt – A. J. P. Schneider oder Alfred Schneider? Die Entscheidung könnte willkürlich getroffen werden, doch die einfachste Lösung ist, alle Punkte und Leerstellen vor der letzten Leerstelle zu ignorieren und die Namen in die Form AJP SCHNEIDER und ALFRED SCHNEIDER umzuwandeln. Danach können sowohl AJP als auch ALFRED als Vornamen behandelt werden, wodurch AJP SCHNEIDER zuerst aufgeführt werden muß.

Zeichen-Umwandlung

Ein Teil unseres Programms akzeptiert als Eingabe einen Namen und gibt einen Namen mit Adresse und Telefonnummer aus. Was erwarten Sie von der Ausgabe, wenn beliebige Eingaben zugelassen sind und danach eine interne Umwandlung in ein Standard-Format erfolgt?

Setzen Sie voraus, daß der Name einem String NAME\$ zugeordnet wurde und daß zwei weitere Strings mit den Namen VORNAME\$ und FAMNAME\$ zur Verfügung stehen. Wie ordnen Sie nun die entsprechenden Teile von NAME\$ den Strings VORNAME\$ und FAMNAME\$ zu? Das Programm könnte ungefähr so aussehen:

Wandle alle Zeichen in Großbuchstaben um
Eliminiere alle nicht alphabetischen Zeichen
– Ausnahme: die letzte Leerstelle
Ordne alle Zeichen nach der Leerstelle
FAMNAME\$ zu
Ordne alle Zeichen, die vor dieser Leerstelle
kommen, VORNAME\$ zu

Vor dem eigentlichen Programmieren in BASIC hier ein Programm-Überblick. Beachten Sie, daß außer recht langen Variablenamen (VORNAME\$) auch Befehlswörter wie BEGIN (STARTE), LOOP (SCHLEIFE) und ENDLOOP (ENDE DER SCHLEIFE) verwendet werden. Dies sind „Erfindungen“, die zum besseren Verständnis dienen. Im letzten Entwicklungsstadium werden sie gegen entsprechende BASIC-Befehle ausgetauscht.

ERSTE ZUSAMMENFASSUNG DER ZIELE

INPUT (EINGABE)

Ein Name (in jeder beliebigen Form)

OUTPUT (AUSGABE)

1. Ein Vorname

2. Ein Familienname

ERSTE VERBESSERUNG

1. Lese NAME\$
2. Wandle alle Buchstaben in Großbuchstaben um
3. Finde letzte Leerstelle
4. Lese FAMNAME\$
5. Lese VORNAME\$
6. Entferne alle Zeichen außer Buchstaben aus dem VORNAME\$

ZWEITE VERBESSERUNG

1. Lese NAME\$
2. (Wandle alle Buchstaben in Großbuchstaben um)
BEGIN (STARTE)
LOOP (SCHLEIFE) während nicht untersuchte Zeichen im NAME\$ sind
Lese der Reihe nach Zeichen aus NAME\$
IF (WENN) Zeichen ist Kleinbuchstabe
THEN (DANN) wandle um
ELSE (SONST) mache nichts
ENDIF (WENN FERTIG — WEITER)
Ordne Zeichen temporärer String-Variablen zu
ENDLOOP (ENDE DER SCHLEIFE)
LET NAME\$=temporäre String-Variable
END (ENDE)
3. (Finde letzte Leerstelle)
BEGIN (STARTE)
LOOP (SCHLEIFE) während nicht untersuchte Zeichen im NAME\$ sind
IF (WENN) Zeichen=" "
THEN (DANN) speichere Position in einer Variablen
ELSE (SONST) mache nichts
ENDIF (WENN FERTIG — WEITER)
ENDLOOP (ENDE DER SCHLEIFE)
END (ENDE)
4. (Lese FAMNAME\$)
BEGIN (STARTE)
Ordne Zeichen nach der letzten Leerstelle in NAME\$ bis zur letzten Leerstelle
FAMNAME\$ zu
END (ENDE)
5. (Lese VORNAME\$)
BEGIN (STARTE)
LOOP (SCHLEIFE) während nicht untersuchte Zeichen bis zur letzten Leerstelle im NAME\$ sind
SCAN (ÜBERPRÜFE) Zeichen
IF (WENN) Zeichen kein Buchstabe des Alphabetes ist
THEN (DANN) mache nichts
ELSE (SONST) ordne Zeichen
VORNAME\$ zu
ENDIF (WENN FERTIG — WEITER)
ENDLOOP (ENDE DER SCHLEIFE)
END (ENDE)
6. (Entferne alle Zeichen außer Buchstaben aus dem VORNAME\$)
(Ist bereits eben in Punkt 5 ausgeführt worden)

Diese zweite Verbesserung ist bereits sehr nahe an dem Stadium, an dem man mit der Codierung in die Programmiersprache beginnen könnte. Lassen Sie uns Punkt 2 (Umwandlung in Großbuchstaben) in einer dritten Verbesserung weiterentwickeln und dann in BASIC programmieren.

DRITTE VERBESSERUNG

```
2. (Wandle alle Buchstaben in Großbuchstaben um)
BEGIN (STARTE)
READ (LESE) NAME$
LOOP (SCHLEIFE)
FOR L=1 TO Länge des Strings
  READ (LESE) Zeichen L
  IF (WENN) Zeichen ein Kleinbuchstabe ist
    THEN (DANN) subtrahiere 32 vom ASCII-
      Wert des Zeichens
  ELSE (SONST) mache nichts
ENDIF (WENN FERTIG — WEITER)
LET TEMPSTRING$=TEMPSTRING$+
  Zeichen
ENDLOOP (ENDE DER SCHLEIFE)
LET NAME$=TEMPSTRING$
END (ENDE)
```

Dieses Programmfragment in unserer Pseudo-Sprache ist nun so weit fortgeschritten, daß Sie es in einer Programmiersprache codieren können. Microsoft BASIC gestattet leider nicht die Verwendung vollständiger Namen für String-Variablen. Deshalb werden sie gegen einzelne Buchstaben ausgetauscht. So wird beispielsweise aus NAME\$ jetzt N\$.

```
1000 REM UMWANDLUNGSRoutine GROSS-
      BUCHSTABEN
1010 INPUT "GEBE NAMEN EIN"; N$: REM NUR
      ZU TESTZWECKEN
1020 LET P$=" ": REM STELLT SICHER, DASS
      DER STRING LEER IST
1030 FOR L=1 TO LEN(N$): REM BEGINN DER
      SCHLEIFE
1040 LET T$=MID$(N$,L,1): REM SUCHT ZEI-
      CHEN HERAUS
1050 LET T=ASC(T$): REM BERECHNET ASCII-
      WERT DES ZEICHENS
1060 IF T >= 97 THEN LET T=T-32: REM UM-
      WANDLUNG KLEIN/GROSS
1070 LET T$=CHR$(T)
1080 LET P$=P$+T$: REM P$ IST TEMPSTRING
1090 NEXT L: REM ENDE DER SCHLEIFE
1100 LET N$=P$: REM N$ BESTEHT NUN NUR
      NOCH AUS GROSSBUCHSTABEN
2000 PRINT N$: REM NUR ZU TESTZWECKEN
2010 END: REM NUR ZU TESTZWECKEN
2020 REM NORMALERWEISE STEHT HIER IM
      FERTIGEN PROGRAMM EIN RETURN
```

Dieser Programmteil würde normalerweise als Unteroutine verwendet und vom Hauptprogramm aufgerufen. Es ist zu Testzwecken mit

einer INPUT-Anweisung, einer PRINT-Anweisung, vielen REM-Anweisungen und einem END-Befehl versehen worden, da das Hauptprogramm ja noch nicht existiert. Diese Anweisungen müssen selbstverständlich vor Einbau der Unteroutine in das eigentliche Programm entfernt werden.

Übungen

- Schreiben Sie ein Programm, das die Eingabe eines einzelnen Zeichens über die Tastatur zuläßt und Ihnen dann den entsprechenden ASCII-Wert des Zeichens mitteilt (in dezimal).
- Überprüfen Sie Ihren Computer daraufhin, ob den ASCII-Werten von 128 bis 255 Zeichen (oder spezielle Grafik) zugewiesen sind. Verwenden Sie dafür eine Schleife und die CHR\$-Funktion.
- Entwickeln Sie ein Programm, bei dem zwei Worte (Strings) zwei String-Variablen zugeordnet werden. Die dritte String-Variable soll nun die beiden Worte zusammenfassen und auf dem Bildschirm ausgeben.

BASIC- Dialekte



Der Lynx verfügt über eine Funktion, UPC\$(A\$), die die Buchstaben in A\$ in Großbuchstaben umwandelt. Unser Programm reduziert sich daher auf folgende Zeilen:

```
1010 INPUT "GEBEN SIE
      EINEN NAMEN EIN";
      N$
1020 LET N$=UPC$(N$)
2000 PRINT N$
2010 REM IM FERTIGEN
      PROGRAMM STEHT
      HIER EIN RETURN
```

Beim Spectrum sieht das Programm so aus:



```
1010 INPUT "GEBEN SIE
      EINEN NAMEN EIN";
      N$
1020 LET P$=" "
1030 FOR L=1 TO LEN N$
1040 LET T$=N$(L)
1050 LET T=CODE T$
1060 IF T >= 97 THEN
      LET T=T-32
1070 LET T$=CHR$ T
1080 LET P$=P$+T$
1090 NEXT L
2000 LET N$=P$
2010 PRINT N$
2020 REM IM FERTIGEN
      PROGRAMM STEHT
      HIER EIN RETURN
```

Dieses Programm läuft auch auf dem Dragon, aber da die Kleinbuchstaben allein für den Drucker reserviert sind, taucht das Problem eigentlich gar nicht auf. Ersetzen Sie Zeile 1010 durch:



```
1010 INPUT "GEBEN SIE
      EINEN NAMEN EIN",
      N$
```




Präzise Greifarme

Kleine Greifarm-Roboter erleichtern den Einstieg in die Steuerung per Heimcomputer. Sie sind über ein paralleles Interface an jedes Gerät anschließbar.

Hatten Sie nicht schon mal den Wunsch, Ihr Rechner könnte Ihnen kleine Handreichungen abnehmen, beispielsweise Tee aufgießen? Bei Vorhandensein der richtigen Schnittstelle ist es einfach, einen Kocher über ein Programm ein- und auszuschalten. Aber wenn wirklich hantiert werden muß, wie beim Eingießen des Wassers in die Kanne, benötigt das Gerät einen Greifarm. Mittlerweile gibt es sie auch für den Heimcomputer-Besitzer. Diese Greifarme sind Miniaturausgaben der Industrieroboter, die in den Fahrzeugfabriken als Schweiß- oder Lackierautomaten am Fließband stehen. Solche „Tischroboter“ sind etwa seit 1980 unter Namen wie ‚Minimover‘, ‚Arm-droid‘ oder ‚Cobra‘ auf dem Markt. Obgleich sie ortsfest sind (wenn man sie nicht auf einen Bodenroboter montiert), können sie erstaunlich vielseitig und präzise arbeiten.

Die wichtigsten Bestandteile des Tischroboters sind die Schrittmotoren, die eine exakt definierte Bewegung des Arms ermöglichen. Es gibt sechs Motoren: einen zur Drehung des „Oberkörpers“ auf dem festen Unterteil, je einen für „Schulter“- und „Ellenbogen“-Gelenk und drei für die Bewegung der „Hand“; alle können sehr einfach vom Rechner gesteuert werden.

Für den Anschluß muß lediglich eine frei programmierbare 8-Bit-Parallelschnittstelle

Greifhand

Die „Knöchelgelenke“ der drei Greiferfinger öffnen unter Federspannung; Gummipolster erleichtern das Festhalten von Gegenständen.

vorhanden sein. Ein Bit bestimmt, ob Informationen vom Computer zum Roboter übertragen werden oder umgekehrt. Zur Auswahl des gewünschten Motors sind drei Adreß-Bits erforderlich, und mit den übrigen 4 Bits werden Drehrichtung und Drehzahl gesteuert. Außerdem werden Taktimpulse übertragen, um Bewegungs- und Programm-Ablauf zu synchronisieren. Für komplizierte Bewegungen ist der Roboter mit Signalspeichern ausgestattet, die den gleichzeitigen Betrieb mehrerer Schritt-

Ellenbogengelenk

Dieses Gelenk hat einen Bewegungsspielraum von 270°.

Unterarm

Zugausgleich

Diese Spannrolle sorgt dafür, daß alle drei Greiferfinger beim Halten eines Gegenstands den gleichen Andruck ausüben, auch wenn es sich um einen Körper von unregelmäßiger Gestalt handelt.

Seilzugrollen

Sie sind so angeordnet, daß das Ellbogenstück bei einer Drehung des Schultergelenks automatisch nachgeführt wird, damit der Unterarm mit dem Greifer die ursprüngliche Winkelseinstellung zum Werkstück behält.

Oberarm

Spannfeder

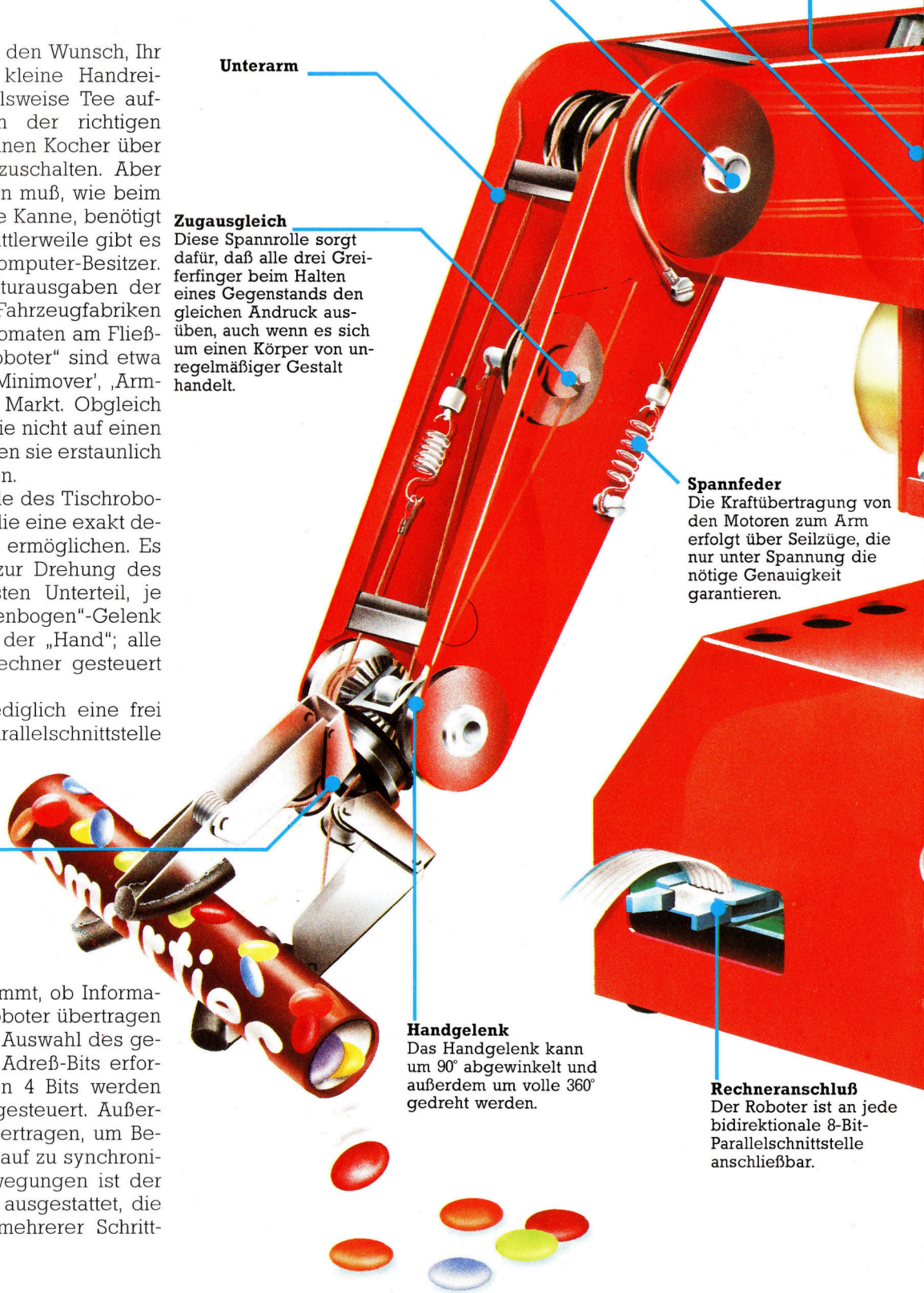
Die Kraftübertragung von den Motoren zum Arm erfolgt über Seilzüge, die nur unter Spannung die nötige Genauigkeit garantieren.

Handgelenk

Das Handgelenk kann um 90° abgewinkelt und außerdem um volle 360° gedreht werden.

Rechneranschluß

Der Roboter ist an jede bidirektionale 8-Bit-Parallelschnittstelle anschließbar.





Zahnriemengetriebe

Der stark untersetzte Zahnriemenantrieb ermöglicht eine hohe Wiederholungsgenauigkeit der Greiferpositionierung (± 1 mm bei Geschwindigkeiten bis zu 1 m/s).

motoren ermöglichen. Ein gegebenes Kommando bleibt für den einen wirksam, während die Anweisungen für einen anderen Motor übertragen werden.

Die Positionierung des Arms und der eigentliche Greifvorgang müssen zunächst in eine Folge einachsiger Bewegungen zerlegt werden. Die einzelnen Motoren müssen dann jeweils die Befehle für „ihre“ Bewegungsanteile erhalten. Alle Steuerinformationen werden abgespeichert, so daß der Rechner den Ablauf

Schultergelenk

Der „Oberarm“ des Roboters ist um 180° um dieses Gelenk drehbar.

Schrittmotoren

Sie gewährleisten eine exakte Steuerbarkeit der Roboterbewegungen. Bei jedem Steuerimpuls dreht sich die Motorachse um einen bestimmten Schritt (beispielsweise $7,5^\circ$).

Körperachse

Das ganze Oberteil ist um 360° drehbar; die Kabel sind durch die hohle Achse geführt.

Leiterplatte

Die Elektronik umfaßt lediglich einfache logische Schaltkreise zum Decodieren der Computersignale. Tischroboter benötigen keinen eigenen Prozessor, kein ROM oder RAM.

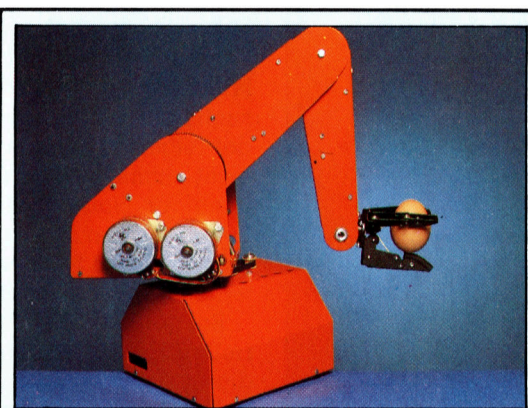
beliebig oft vom Roboter wiederholen lassen kann. Die mitgelieferten Betriebssysteme bieten auch meist die Möglichkeit, den Roboter die Gesamtbewegung durch Abspeichern einer handgesteuerten Ausführung „lernen“ zu lassen („Follow Me“-Verfahren).

Beim Hantieren mit empfindlichen Dingen – gängiges Testobjekt ist ein rohes Ei – muß der Computer die Greifkraft überwachen: Das Ei darf weder herunterfallen, noch zerdrückt werden. Die Informationen können dem Rechner auf verschiedene Art vermittelt werden; meistens verwendet man Microschalter. Damit kann das Erreichen einer vorgegebenen Greifkraft oder auch der Bewegungsspielraum des Arms eingegrenzt werden.

Bei größeren Robotern werden statt der Microschalter meist „Sensoren“, wie beispielsweise Dehnungsmeßstreifen, benutzt. Dabei wird die durch eine Kraft verursachte Widerstandsänderung eines elektrischen Leiters gemessen. Dies Verfahren ist zwar aufwendiger, aber äußerst exakt.

Solange kein Informationsfluß vom Roboter zurück zum Rechner erfolgt, spricht man von einer „Steuerung“. Bei einfacher Steuerung würde im obigen Beispiel das Ei zweifellos zerbrechen. Wenn dagegen eine Rückkopplung (feedback) wirksam ist, liegt eine „Regelung“ vor. Dabei würde das Kraftmeßsignal die Greifer-Schließbewegung anhalten, wenn das Ei fest, aber noch unzerdrückt, im Griff ist.

Hochentwickelte Industrieroboter besitzen oft vielfältige Sensoren – neben den erwähnten Druckmessern beispielsweise noch visuelle Systeme oder Temperaturfühler.



Die Sprache des Roboters

Ein Roboter-Steuerungsprogramm ist recht einfach zu erstellen. In BASIC müßte hauptsächlich die Übernahme der Tastatur-Kommandos und ihre Ausgabe an den Roboter mit Hilfe von POKE programmiert werden. Entsprechend müßten die Rückmeldungen des Roboters mit Hilfe von PEEK eingelesen werden. Höhere Geschwindigkeiten werden mit dem Maschinencode oder auch mit der Sprache FORTH erreicht. Mit ihr, die den gleichen Komfort wie BASIC bietet, wird Echtzeit-Programmierung möglich. Ein komplexer Greifvorgang könnte bei FORTH beispielsweise durch einen einzigen „GREIF“-Befehl aufgerufen werden.



Herman Hollerith

1860

Geboren in Buffalo im Staate New York

1879

Universitätsabschluß an der Columbia University und Annahme einer Assistentenstelle bei der amerikanischen Volkszählungsbehörde

1883

Arbeit im Patentamt in Washington

1884

Anmeldung der ersten Patente für die Übertragung von Informationen auf Endlos-Papierstreifen

1887

Sein System wird zum ersten Mal in Baltimore, Maryland, für die Sterblichkeitsstatistik eingesetzt

1889

Das System wird im Büro des Oberstabsarztes der Armee für Krankheitsstatistiken eingesetzt; Hollerith meldet die Lochkarte zum Patent an

1890

Erhält den Zuschlag, mit seinen Maschinen die Auswertung der Volkszählung von 1890 durchzuführen; die Columbia University verleiht ihm den Dokortitel für seine Arbeit über Datenverarbeitung

1901

Die neue Maschinen- generation wird für landwirtschaftliche Statistiken eingesetzt

1911

Gründung einer Dachgesellschaft, der Tabulating Recording Company

1924

Die Firma ändert ihren Namen in International Business Machines (IBM)

1929

Stirbt in Washington DC



Der geniale Erfinder, der als erster mit Lochkarten arbeitete, gründete eine Firma, die sich zum größten Computerhersteller der Welt entwickelte.

Herman Hollerith wurde 1860 in Amerika geboren. Nach einem Studium an der Columbia University arbeitete er als Assistent in der Volkszählungsbehörde der Vereinigten Staaten und wertete die Daten der Zählung im Jahre 1880 statistisch aus. Da alle Daten mühsam per Hand ausgewertet wurden, lagen bei der nächsten Zählung, zehn Jahre später, noch nicht einmal alle Ergebnisse der letzten Befragung vor. Hollerith glaubte fest daran, dieses Problem lösen zu können, und er ließ sich deshalb von dem National Census Office zum Patentamt in Washington versetzen.

Bei seinem ersten Lösungsversuch übertrug Hollerith die Daten zunächst auf Papierbänder, die in verschiedene „Felder“ unterteilt waren. Jedes einzelne symbolisierte eine bestimmte Kategorie wie z. B. männlich/weiblich oder schwarz/weiß. Die kreisrunden Löcher wurden später maschinell „gelesen“. 1884 erhielt Hollerith seine ersten Patente. Während der nächsten Jahre arbeitete er an der Verbesserung seines Systems, das zunächst für die Krankheitsstatistiken der Armee und der

schnell anwachsenden amerikanischen Städte eingesetzt wurde.

Fünf Jahre später führte Hollerith eine wesentliche Verbesserung ein: Statt eines gelochten Papierstreifens wurde für jede befragte Person eine eigene Karte angelegt. Er konstruierte einen Speziallocher, der 6 mm große quadratische Löcher ausstanzte. Eine einzige Lochkarte konnte auf diese Weise eine Reihe unterschiedlicher Informationen und Daten enthalten.

1889 stellte sich auch der kommerzielle Erfolg ein: Die Volkszählungsbehörde verkündete eine Ausschreibung für Maschinen, die die nächste Zählung statistisch auswerten sollten. Hollerith machte mit Abstand das Rennen. Da alle seine Maschinen patentiert waren, nutzte er seine Monopolstellung und berechnete der amerikanischen Regierung 65 Cents für die Verarbeitung von je tausend Karten. Obwohl es für jede gezählte Person eine eigene Lochkarte gab, benötigte Hollerith für seine Auswertung nur zwei Jahre. Danach stand fest, daß Amerika 56 Millionen Einwohner hatte, und Hollerith sandte der Regierung eine entsprechend hohe Rechnung. Bis zur Volkszählung im Jahre 1900 hatte er seine Maschinen wiederum wesentlich verbessert, weilerte sich aber, seinen Preis zu senken. Als seine Patente schließlich abliefen, suchte die amerikanische Regierung nach anderen Firmen, die diese Arbeit billiger erledigen konnten. Hollerith gründete jedoch eine eigene Firma und schlug auch diesmal die Konkurrenz aus dem Feld. Seine Firma wurde später in „International Business Machines“ umbenannt und ist heute mit 20 Milliarden Dollar Jahresumsatz der größte Computerhersteller der Welt.

Noch heute verwendet man Holleriths Methode zur Darstellung von Informationen. Geändert hat sich nur das Format der Karten: Die modernen Lochkarten haben 12 Zeilen mit je 80 Spal-

ten. Tabelliermaschinen arbeiten nach dem Dezimalsystem und können auf jeder Karte 80 Zahlen darstellen. Mit Computern können aber auch binär-codierte Karten verarbeitet werden.





Turtle-Variationen

Wenn Variablen durch verschiedene Eingabewerte verändert werden können, so lassen sich auf dem Bildschirm interessante Grafiken erzeugen.

In LOGO kann ein Wort, zum Beispiel GROESSE, auf drei unterschiedliche Arten eingesetzt werden. Allein die Schreibweise (entweder GROESSE oder :GROESSE oder "GROESSE") gibt dem Computer Aufschluß darüber, welche Bedeutung das Wort hat und wie das Programm ablaufen soll. Trifft LOGO auf den Ausdruck GROESSE in dieser Schreibweise, also ohne vorangehendes Interpunktionszeichen, wird angenommen, daß es sich hierbei um einen Prozedurnamen handelt und die unter dieser Prozedur abgelegten Befehle ausgeführt werden sollen. :GROESSE dagegen wird als Variable benutzt, der die Eingabe eines Wertes folgen muß. Der Ausdruck "GROESSE (ein Wort, das mit Anführungsstrichen beginnt) wird verwendet, um ihm Variablen oder Prozeduren zuzuordnen.

Bei manchen LOGO-Versionen sind die Anführungsstriche vor den eigentlichen Prozedurnamen bei Befehlen wie EDIT, ERASE und PO nicht notwendig. In diesen Fällen lautet die korrekte Eingabe EDIT QUADRAT, obwohl der vollständige Prozedurnamen "QUADRAT heißt.

Wie Sie bereits gelernt haben, wird ein LOGO-Programm durch Eingabe des entsprechenden Prozedurnamens gestartet – und zwar nach genau dem gleichen Schema, das auch für Anweisungen wie DRAW oder HIDE-TURTLE gilt. Andere Befehle, wie etwa FORWARD, können nur ausgeführt werden, wenn zusätzliche Informationen vorhanden sind. Ohne die Angabe, wie weit die Turtle vorwärts laufen soll, ist der Befehl sinnlos. Statt fester Werte lassen sich aber auch Variablen in das Programm einsetzen, deren Werte man anschließend beliebig verändern und somit unterschiedliche Effekte erzielen kann. Dazu ein bekanntes Beispiel:

```
TO QUADRAT
  REPEAT 4 [FD 50 RT 90]
END
```

Das Programm zeichnet ein Quadrat mit einer Seitenlänge von 50 Schritten. Ziel ist, die Prozedur so umzuschreiben, daß der Anwender auch andere Werte für die Seiten eingeben kann. Für diesen Zweck muß das Programm durch EDIT "QUADRAT bzw. EDIT QUADRAT aufgerufen, der Wert 50 durch eine Variable ersetzt sowie in der ersten Zeile eine Eingabemöglichkeit für den Variablenwert vorgesehen

werden. Danach sieht die Prozedur so aus:

```
TO QUADRAT :GROESSE
  REPEAT 4 [FD :GROESSE RT 90]
END
```

Nachdem das Programm aktiviert wurde, muß der Variablen :GROESSE ein Wert zugewiesen werden. Versuchen Sie es mit GROESSE 40, GROESSE 10 usw.

Überlegen Sie, in welcher Reihenfolge LOGO die Befehle verarbeitet, wenn Sie zum Beispiel GROESSE 30 eingeben. Die erste Zeile besagt, daß eine Eingabe zu tätigen ist, die "GROESSE zugeordnet wird. Nachdem die eingetippte Zahl, in diesem Fall 30, in "GROESSE abgelegt wurde, können die weiteren Befehle ausgeführt werden. Man kann sich den Ablauf so vorstellen: Sobald LOGO die Zeile FORWARD :GROESSE erreicht, wird der Wert, den diese Variable enthält, herausgenommen und FORWARD zugewiesen. Die Zahl, die für :GROESSE eingegeben wurde, kann jedoch nur in dieser einen Prozedur verwendet werden. Erscheint der gleiche Variablenname in einer anderen Prozedur, so muß man einen neuen Wert definieren. Es handelt sich hier also um eine „lokale“ Variable.

Man kann Eingaben ebenfalls in Unterprozeduren integrieren. Zur Veranschaulichung das HAUS-Programm:

```
TO HAUS :WERT
  QUADRAT :WERT
  FD :WERT RT 30
  TRI :WERT
  LT 30 BK :WERT
END
```

```
TO QUADRAT :GROESSE
  REPEAT 4 [FD :GROESSE RT 90]
END
```

```
TO TRI :SEITE
  REPEAT 3 [FD :SEITE RT 120]
END
```

In diesem Beispiel wurden drei Variablen eingesetzt – "WERT, "GROESSE und "SEITE. Da es sich ausschließlich um lokale Variablen handelt, hätte man genausogut statt drei verschiedener Namen einen, der alle repräsentiert, wählen können.





Geben Sie nun HAUS 30 ein, um das Programm zu starten. LOGO liest anschließend die Eingabezeile und ordnet 30 der Variablen "WERT" zu. Dadurch wiederum wird anschließend den Variablen "GROESSE" und "SEITE" ebenfalls der Wert 30 zugewiesen, da diese in der Hauptprozedur HAUS nacheinander definiert werden. So wandelt sich die QUADRAT-Anweisung FD :GROESSE in FD 30 usw.

Auch beim nächsten Programm, das ein Vieleck auf den Bildschirm zeichnet, ist eine Eingabemöglichkeit für die Anzahl der Seiten vorgesehen:

```
TO VIELECK :SEITEN
  REPEAT :SEITEN [FD 50 RT 360 / :SEITEN]
END
```

Die Angabe VIELECK 3 bewirkt, daß ein Dreieck gezeichnet wird; bei VIELECK 4 entsteht ein Quadrat... Bislang konnten Sie nur die Seitenanzahl selbst bestimmen – die Länge der Seiten (50) blieb immer konstant. Um die Prozedur flexibler zu gestalten, muß also der Wert 50 durch eine Variable ersetzt werden, der man beliebige Zahlen zuweisen kann. Und so sieht die verbesserte Prozedur aus:

```
TO VIELECK :SEITEN :LAENGE
  REPEAT :SEITEN [FD :LAENGE RT 360 /
    :SEITEN]
END
```

Vergessen Sie nicht, daß bei diesem Programm zwei Werte einzugeben sind. VIELECK 10 20 zum Beispiel läßt eine zehneckige Figur mit einer Seitenlänge von jeweils 20 Einheiten auf dem Schirm entstehen.

Globale Variablen

Im Gegensatz zu „lokalen“ verarbeitet LOGO auch „globale“ Variablen, die von mehreren Prozeduren aufgerufen werden und Informationen von einem Programm zum nächsten transferieren. Der Nachteil bei dieser Variablenart ist, daß die Fehlersuche wesentlich komplizierter wird. Der Befehl MAKE wird benutzt, um globalen Variablen Werte zuzuweisen. MAKE "SEITE 3 bedeutet: Ordne "SEITE die Zahl 3 zu, während MAKE "SEITE :SEITE + 1 den Inhalt von "SEITE um eins erhöht. Grundsätzlich benötigt MAKE immer zwei Angaben – den Wert, der, wie oben gezeigt, auch aus einer arithmetischen Operation bestehen kann, und die Variable, der dieser Wert bzw. das Ergebnis der Berechnung zugewiesen wird.

Aus der erlernten Anweisung werden wir nun eine Prozedur konstruieren, die eine Spirale zeichnet. Zur Berechnung und Ausführung benötigt die Hauptprozedur EQSPI drei Werte: Die Länge der ersten zu zeichnenden Linie,

den Drehungswinkel nach jeder ausgeführten Linie und den Berechnungsfaktor, um den die Länge der Linien erweitert wird, damit der Spiraleffekt entsteht. Versuchen Sie zum Beispiel die Eingaben 70 283 0.95, 70 143 0.95 oder 20 143 1.05.

NOWRAP ist ein neuer Befehl. Er verhindert, daß die Turtle „aus dem Bildschirm“ läuft. Sobald die Spirale den Schirmrand erreicht, stoppt das Programm und gibt eine Fehlermeldung aus. Der Programmablauf läßt sich wie folgt beschreiben: Zuerst wird eine Linie dargestellt (die jeweilige Länge ergibt sich aus dem Berechnungsfaktor), danach dreht sich die Turtle um den eingegebenen Winkel und anschließend wird der Faktor entsprechend geändert. Der ganze Prozeß wiederholt sich tausendmal, es sei denn, Sie unterbrechen den Ablauf mit Control-G oder BREAK. Je nachdem, ob ein Berechnungsfaktor größer oder kleiner als Eins eingegeben wurde, werden die Linien während des Zeichenvorgangs länger oder kürzer. Alle verwendeten Variablen sind lokal, mit Ausnahme von „SCALE“, die den Datenaustausch zwischen den beiden Prozeduren ermöglicht.

Spiral-Programm

```
TO EQSPI :SIZE :ANGLE :FACT
  SETUP
  REPEAT 1000 [S.FORWARD :SIZE RIGHT
    :ANGLE GROW :FACT]
END
TO SETUP
  DRAW NOWRAP MAKE "SCALE 1
END
TO GROW:NUMB
  MAKE "SCALE:SCALE 1 :NUMB
END
TO S.FORWARD :DIST
  FORWARD :SCALE 1:DIST
END
```

LOGO-Dialekte

In einigen LOGO-Versionen ist die Anweisung NOWRAP durch FENCE zu ersetzen. Bei den Dialekten, die weder FENCE noch NOWRAP als Befehl vorsehen, z. B. Atari-LOGO, wird die Anweisung WINDOW verwendet. Abweichungen bestehen auch bei dem Befehl DRAW. Bitte ersetzen Sie diesen gegebenenfalls durch CS. Manche LOGO-Versionen verarbeiten keine Befehle in abgekürzter Form. Hinweise darauf finden Sie in Ihrem LOGO-Handbuch.

Übungen

1. Entwickeln Sie eine Prozedur, die einen Kreis mit dem Radius von 50 zeichnet. Ersetzen Sie anschließend die 50 durch eine Eingabevariable.
2. Schreiben Sie ein Programm, das fünf konzentrische Kreise auf den Bildschirm zeichnet.

Lösungen

1) Tangram-Prozeduren

Nachfolgend sind die Prozeduren für einen Läufer, einen sitzenden Mann, einen sich beugenden Mann und eine Katze aufgeführt. Die Einzelteile, aus denen die Figuren zusammengesetzt sind, wurden bereits in der HUND-Prozedur definiert und müssen nur statt rechtsherum nach links gedreht werden.

```
TO PAR
  REPEAT 2 [FD 25 RT 45 FD 35 RT 135]
END
```

Nun die Drehungen nach links:

```
TO PAR1
  REPEAT 2 [FD 25 LT 45 FD 35 LT 135]
END
```

Die anderen Befehlsabfolgen beziehen sich ausschließlich auf die bereits erklärten Prozeduren.

Man Running

```
TO RUNNING
  MOVE1 TRI1 MOVE2 PAR MOVE3 TRI3 MOVE4
  TRI3 MOVE5 SQUARE MOVE6 TRI1 MOVE7 TRI2
  MOVE8
END
TO MOVE1
  LT 45
END
TO MOVE2
  PU FD 25 RT 135 FD 17.5 LT 45 PD
END
TO MOVE3
  PU FD 75 RT 90 PD
END
TO MOVE4
  PU RT 90 FD 25 RT 90 PD
END
TO MOVE5
  PU FD 50 RT 135 FD 50 LT 135 PD
END
TO MOVE6
  PU RT 135 FD 21 RT 135 FD 25 LT 90 FD 50
  LT 90 FD 25 RT 90 PD
END
TO MOVE7
  PU FD 25 RT 135 FD 71 RT 45 BK 35 PD
END
TO MOVE8
  PU FD 35 LT 90 FD 25 RT 45 FD 17.5 LT 45
  FD 25 RT 135 PD
END
```

Man Sitting

```
TO SITTING
  MOVE1 TRI1 MOVE2 TRI2 MOVE3 TRI3 MOVE4
  TRI1 PAR1 MOVE5 SQUARE MOVE6 TRI3 MOVE7
END
TO MOVE1
  LT45
END
TO MOVE2
  PU FD 25 LT 45 FD 17.5 RT 90 PD
END
TO MOVE3
  PU BK 15 LT 90 PD
END
TO MOVE4
  PU FD 50 RT 45 FD 25 RT 90 PD
END
TO MOVE5
  PU FD 25 LT 45 FD 35 LT 45 PD
END
TO MOVE 6
  PU BK 50 LT 90 PD
END
TO MOVE7
  PU BK 21 RT 135 BK 50 RT 90 FD 35 LT 90
  PD
END
```

Man Bowing

```
TO BOWING
  MOVE1 TRI1 MOVE2 PAR1 MOVE3 TRI3 MOVE4
  TRI3 MOVE5 TRI1 MOVE6 TRI2 MOVE7 SQUARE
  MOVE8
END
TO MOVE1
  LT 90
END
TO MOVE2
  PU FD 25 RT 135 FD 30 PD
END
TO MOVE3
  PU LT 45 FD 35 LT 135 BK 50 PD
END
TO MOVE4
  PU RT 90 FD 50 LT 135 PD
END
TO MOVE5
  PU RT 90 FD 50 LT 135 FD 5 RT 90 PD
END
TO MOVE6
  PU RT 90 FD 25 RT 45 FD 7.5 RT 45 BK 35
  PD
END
TO MOVE7
  PU FD 35 RT 135 FD 7.5 LT 45 FD 55 RT 45
  PD
END
TO MOVE8
  PU LT 45 FD 36 RT 45 FD 56 LT 135 FD 5
  LT 45 BK 25 PD
END
```

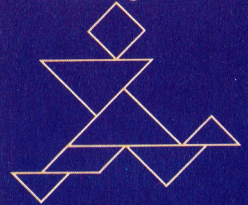
Cat

```
TO CAT
  MOVE1 TRI3 MOVE2 SQUARE MOVE3 TRI1 MOVE4
  TRI1 MOVE5 TRI3 MOVE6 PAR1 MOVE7 TRI2
  MOVE8
END
TO MOVE1
  PU FD 50 RT 90 PD
END
TO MOVE2
  RT 170
END
TO MOVE3
  PU RT 90 FD 25 LT 90 PD
END
TO MOVE4
  RT 180
END
TO MOVE5
  PU RT 90 FD 25 LT 80 FD 50 RT 45 FD 50
  RT 90 PD
END
TO MOVE6
  LT 155
END
TO MOVE7
  PU LT 160 FD 35 PD
END
TO MOVE8
  PU FD 35 LT 45 FD 21 RT 135 PD
END
```

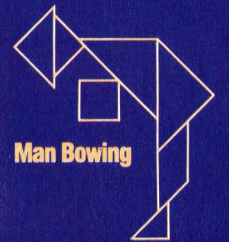
Und so könnten Ihre Antworten auf die Fragen 2 bis 4 aussehen:

<p>2)</p> <pre>TO HOUSE SQUARE FD 50 RT 30 TRI LT 30 BK 50 END TO SQUARE REPEAT 4 [FD 50 RT 90] END TO TRI REPEAT 3 [FD 50 RT 120] END</pre>	<p>3)</p> <pre>TO BOARD REPEAT 5 [LINE FD 10] BK 50 END TO LINE REPEAT 5 [SQUARE RT 90 FD 10 LT 90] LT 90 FD 50 RT 90 END TO SQUARE REPEAT 4 [FD 10 RT 90] END</pre>	<p>4)</p> <pre>TO SIX.STAR TRI MOVE TRI MOVE. BACK END TO TRI RT 30 REPEAT 3 [FD 50 RT 120] END TO MOVE PU FD 29 RT 60 PD END TO MOVE.BACK PU LT 60 BK 29 PD END</pre>
--	--	--

Man Running



Man Sitting



Man Bowing



Cat

Ändern und Löschen

Die Möglichkeit, Daten auf dem Bildschirm aufzubereiten, wie sie von den meisten Microcomputern geboten wird, hilft viel Zeit und Mühe zu sparen.

Beim Eingeben von Daten in den Computer sind Tippfehler unvermeidlich. Unvermeidlich ist es auch, bereits eingegebene Daten inhaltlich zu ändern, zu ergänzen oder auf den neuesten Stand zu bringen. Wir brauchen darum eine Möglichkeit, auf dem Bildschirm angezeigte Daten „aufzubereiten“. Diese Möglichkeit bietet der „Editor“ wie die Aufbereitungsfunktion genannt wird.

Den Editor gibt es in irgendeiner Form in den meisten Anwenderprogrammen. In einem Textverarbeitungssystem beispielsweise ermöglicht der Editor eine Textaufbereitung in der Form wie sie beim Überarbeiten von Berichten und Briefentwürfen gebraucht wird.

Bei fast allen Heimcomputern ist im Betriebssystem ein Editor eingebaut, der das Korrigieren und Ändern von Programmlistings ermöglicht. Gerade bei der Programmeingabe passieren häufig Eingabefehler; Syntaxfehler treten mit ziemlicher Regelmäßigkeit auf, und beim Programmablauf ist mit Sicherheit noch der eine oder andere „Mißgriff“ zu beseitigen. Dabei darf nicht vergessen werden, daß ein guter Programmierer bereits vor dem Eingeben eine Menge Zeit dafür aufgewendet hat, sein Programm schon auf dem Papier auf eventuelle Logikfehler zu überprüfen. Es ist eine schlechte Arbeitsweise, ein Programm „aus dem Kopf“ einzutippen und sich dann erst mit einem vielfach größeren Zeitaufwand auf die Fehlersuche zu begeben.

Es gibt zwei Editor-Arten: den Bildschirm-Editor und den Zeilen-Editor. Der Bildschirm-Editor erlaubt ein freizügiges und leichtes Ar-

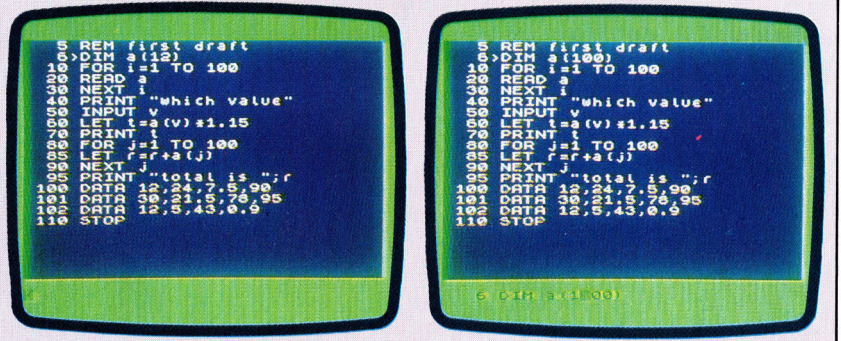
beiten; bei zahlreichen Heimcomputern steht jedoch nur der Zeilen-Editor zur Verfügung. Bei diesem hat der Programmierer zwar die Möglichkeit, mit Hilfe der Anweisung LIST das ganze Programm auszudrucken, wenn jedoch in einer Zeile eine Korrektur notwendig wird, muß die ganze Zeile neu eingetippt werden. Bei einigen Systemen kann man mit EDIT und nachfolgender Zeilennummer diese eine Zeile separat auf dem Bildschirm darstellen und dann Änderungen und Löschungen mit den Tasten „backspace“ (Rücktaste) und „rubout“ durchführen. Andere Befehle wie DELETE machen es möglich, bestimmte Zeilen zu löschen.

Der Bildschirm-Editor dagegen ist flexibler und einfacher zu handhaben. Er gestattet, Text- oder Zeichensymbole ohne Mühe auf dem Bildschirm hin- und herzubewegen. Wird die Taste RETURN betätigt, liest der Editor die Programmzeile, auf der der Cursor steht, in den Interpreter.

Ein Bildschirm-Editor muß in Maschinencode geschrieben sein, um ausreichend schnell zu arbeiten. Ein komfortabler Editor verfügt über zusätzliche Funktionen, wie z. B. das „scrolling“, d. h. die Möglichkeit, ein ganzes Programmlisting von oben nach unten oder umgekehrt über den Bildschirm wandern zu lassen. So kann man sowohl ganze Zeilen als auch einzelne Zeichen einfügen oder löschen oder sogar, wie bei einem Textverarbeitungssystem, bestimmte Textpassagen (Namen, Zeichenfolgen usw.) bei jedem Auftreten im Text oder der Liste herausfinden und ändern. Mit jeder neuen Computergeneration werden die Editor-Leistungen vielseitiger und die Handhabung einfacher. Mit der Entwicklung der „Maus“ und der Verfügbarkeit von Software, die das manuelle Schneiden und Zusammenstellen von Textteilen „nachahmt“, wird der Zeitaufwand immer geringer, den man benötigt, um ein Schriftstück oder eine Liste vom ersten Entwurf bis zur endgültigen Form „aufzubereiten“.

Zeile um Zeile

Der Spectrum bietet bessere Editiermöglichkeiten als die meisten anderen Zeilen-Editoren. Um eine bestimmte Zeile in einem Programm zu ändern, muß der Positionszeiger (»), der zwischen der Zeilennummer und der Zeile selbst erscheint, mit Hilfe der Cursorsteuerungstasten zu der entsprechenden Zeile bewegt werden.





Zeichengenerierung

Im Zeichengenerator, einem festgelegten Sektor im Speicher des Computers, wird definiert, wie ein Zeichen auf dem Bildschirm aussieht. Bei manchen Systemen ist es auch möglich, eigene Zeichen zu „konstruieren“.

Im BASIC-Kurs haben Sie gelernt, daß jedes alphanumerische und grafische Zeichen als 8-Bit-Code (gewöhnlich im ASCII-Format) fest gespeichert ist. Sollen Zeichen (Buchstaben, Ziffern, Symbole) auf dem Bildschirm erzeugt werden, müssen die Codes für diese Zeichen im Bildwiederholungsspeicher (Video-RAM), einem reservierten Speicherbereich, abgelegt werden. Ein Beispiel: Der Buchstabe A soll in der oberen linken Ecke des Bildschirms erscheinen. In diesem Fall enthält das erste Byte im Video-RAM den Code 65 (ASCII für A). Kommt ein C unter den Buchstaben A, ist der Code 67 (ASCII für C) an 41ster Stelle im Video-RAM abgelegt, vorausgesetzt, der Computer stellt 40 Zeichen pro Zeile dar, und so fort. Wie verwandelt nun der Computer den Code 65 in ein Punktraster, das auf dem Bildschirm den Buchstaben A darstellt? Die Antwort liegt in einer Einrichtung, die „Zeichengenerator“ genannt wird.

Punktmuster-Sammlung

Dieser Generator ist im Prinzip lediglich eine Sammlung von Punktmustern, die als Bits gespeichert sind. Bei Heimcomputern ist der Zeichengenerator im ROM gespeichert. Dabei kann er in den ROMs untergebracht sein, die den BASIC-Interpreter und das Betriebssystem enthalten, oder aber in einem separaten ROM-Chip. Ist der Zeichengenerator in einem separaten Chip untergebracht, besteht die Möglichkeit, auf dem Markt erhältliche Austausch-ROMs einzusetzen, mit denen sich Zeichen in anderer Typographie oder auch spezielle Symbole erzeugen lassen, wie sie beispielsweise von Ingenieuren oder Mathematikern verwendet werden. Bei vielen Computern besteht die Möglichkeit, den Zeichengenerator in den RAM-Speicher zu übertragen.

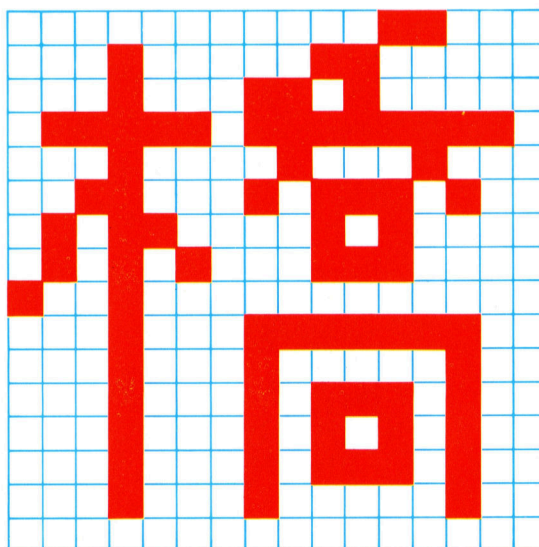
Jedes Zeichen wird aus einzelnen Punkten zusammengesetzt, deren Position auf einem Raster definiert wird. Die meisten Heimcomputer verwenden ein 8×8 -Raster. Besser ist jedoch ein größeres Raster, weil durch die damit verbundene höhere Auflösung die Lesbarkeit wesentlich verbessert und der Bereich der darstellbaren Zeichen erweitert wird, wie die Darstellung mit dem 16×16 -Raster deutlich macht. Die Zeichen entstehen durch Ausfüllen

von Feldern in dem das Raster bildenden Gitter. Jedes ausgefüllte Feld wird durch ein 1-Signal, jedes leere Feld durch ein 0-Signal definiert. Für ein Zeichen, das in ein 8×8 -Punktraster zerlegt ist, wird ein Speicherplatz von acht Bytes ($8 \times 8 = 64$ Bits) benötigt.

Das erste Byte im Zeichengenerator ist der Code für die oberste Rasterzeile des ersten Buchstabens. Bei einem Computer, der nach dem ASCII-Code mit seinen Werten zwischen 0 und 127 arbeitet, benötigt der Zeichengenerator 128×8 Bytes.

Wenn nun die Rasterabtastung des Bildschirms die oberste Zeile der Anzeige generiert, muß der Computer die oberste Zeile der Punktmatrix für den Buchstaben erzeugen, der oben links auf dem Bildschirm steht, dann die oberste Punktzeile des rechts daneben stehenden Buchstabens und so fort bis an das Bildschirmende.

Kehren wir zu unserem Beispiel mit dem Buchstaben A zurück, dessen ASCII-Code 65 ist. Wir können ausrechnen, daß die erste Rasterzeile in Byte Nummer 520 ($65 \times 8 + 0$) gespeichert ist, die zweite in Byte Nummer 521 ($65 \times 8 + 2$) und so weiter. Der Video-Schaltung bleibt jetzt nur noch, diese acht Bits in eine Spannungsfolge umzuwandeln, die den abtastenden Elektronenstrahl ein- und ausschaltet, um die Anzeige des Zeichens auf dem Bildschirm zu erzeugen.



Japanische Schriftzeichen sind sehr kompliziert. Eine gewöhnliche 8×8 -Matrix kann nicht genügend Einzelheiten darstellen, um diese Schriftzeichen lesbar zu machen. Das hier gezeigte japanische Schriftzeichen für „Brücke“ ist erst auf einer 16×16 -Matrix lesbar. Noch deutlicher werden Schriftzeichen mit einer 24×24 -Punktmatrix.

橋



Schritt für Schritt

Beim Messen linearer oder winkelförmiger Bewegungen mit Hilfe optischer Sensoren erweisen sich Binärcores als fehlbar. Darum wurde der Gray-Code entwickelt.

Nur ein Bit ändert sich
Die Tabelle zeigt den Gray-Code im Vergleich zum Binär-Code für Dezimalzahlen von 0 bis 15.

Dezimal	Binär	Gray-Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Bei zahlreichen Anwendungen ist es notwendig, die genaue örtliche Position eines beweglichen Gegenstandes festzustellen und die Werte an einen Computer zu übermitteln. Beispiele dafür sind Roboter, bei denen der Computer die Positionen und Bewegungsrichtungen aller Elemente kennen muß, oder Werkzeugmaschinen, wo die Fräsbewegungen von einem Computer gesteuert werden. Wie aber wird eine Position in eine Binärzahl verwandelt, die vom Computer verarbeitet werden kann?

Eine Lösung wäre die Verwendung eines Analogsystems. Das sich bewegende Element wird an einen Regelwiderstand angeschlossen und die resultierende Spannung in einen Analog-Digital-Umsetzer eingelesen (oder direkt in den Analogeingang des Computers). Ein solches System ist jedoch nicht sehr genau und seine mechanischen Elemente unterliegen dem Verschleiß.

Ein anderer Weg ist, einen Binär-Code auf das entsprechende Element zu drucken und diesen direkt in den Computer einzulesen. Der Code besteht gewöhnlich aus einer Reihe schwarzer und weißer Felder und wird mit Hilfe eines auf die Felder gerichteten Lichtstrahls in Verbindung mit Fotozellen gelesen.

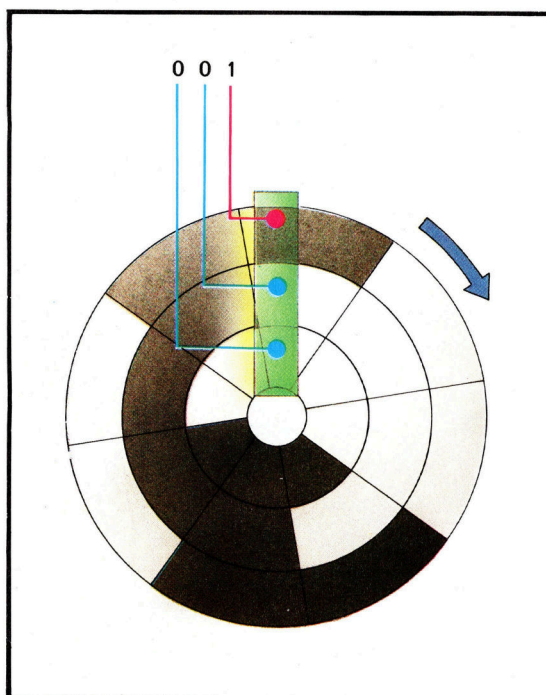
Je eine Fotozelle ist dabei für eine Stelle des Binärmusters verantwortlich. Bewegt sich das Teil, verändert sich das Binärmuster unter den Fotozellen und erzeugt dadurch ein Binär-Signal, das die gegenwärtige Position des Elementes bestimmt. Neben linearen Binärmustern gibt es auch kreisförmige zur Erfassung von Drehbewegungen, wie z. B. in einem Roboter-Armgelenk.

Korrekte Signale

Schwierigkeiten entstehen hierbei, wenn sich der Gegenstand von dem einen Binär-Code zum anderen bewegt oder wenn er gar zwischen zwei Codes zum Stehen kommt, denn dann können die Fotozellen sowohl den einen, als auch den anderen Code lesen. Bleibt das Element beispielsweise zwischen der Binärposition 11 (1011) und 12 (1100) stehen, kann nur das höchstwertige Bit (die am weitesten links stehende 1) ein korrektes Signal liefern, wogegen die anderen drei Bits unzuverlässige Werte wiedergeben. In bestimmten Situationen ändern sich alle Bits, wie beim Übergang von binär 7 (0111) auf binär 8 (1000), so daß selbst kleine Ungenauigkeiten im Muster zu unkorrekten Werten aller drei Fotozellen führen können. Die Positionsangabe wäre möglicherweise völlig falsch, und der Computer würde dies nicht erkennen. Die Folgen könnten verhängnisvoll sein.

Um diese Schwierigkeiten zu beheben, wird ein Zählsystem benötigt, in dem sich bei jedem Schritt nur ein Bit ändert. Dies stellt sicher, daß beim Übergang von einem Code zum anderen nur ein Bit fehlerhaft sein und sich ein Positionsfehler nur im Rahmen eines Schritts bewegen kann. Ein solches Zählsystem ist der sogenannte Gray-Code, der unmißverständlich durch folgende Anforderungen bestimmt wird: Bei jedem Bewegungsschritt ändert sich nur ein einziges Bit, und dies soll das am weitesten rechts stehende Bit sein, das noch ein unverwechselbares Muster ergibt. Beginnen wir, wie im Binärsystem, mit 0000, wird die 1 im Gray-Code durch 0001 repräsentiert. Um die 2 darzustellen, muß nun das zweite Bit geändert werden, um 0011 zu erhalten. Zur Darstellung der 3 ist es jetzt möglich, das erste Bit zu ändern, um 0010 zu erhalten. Demgegenüber wäre die Binärfolge für diese Zahlen 0000, 0001, 0010 und 0011.

Die Winkelstellung eines Gegenstandes kann in den Computer mit Hilfe einer Scheibe eingelesen werden, auf die ein Code gedruckt ist. Das durch einen Lichtstrahl angeleuchtete Rasterbild wird durch eine Reihe Fotozellen abgetastet. Bewegt sich der Gegenstand, verändert sich das Rasterbild unter den Fotozellen, und es wird ein digitales Signal erzeugt. Bei Verwendung eines Binär-Codes kann es zu fehlerhaften Ausgangssignalen kommen, wenn die Scheibe am Übergang von einem Rasterbild zum anderen stehen bleibt. Diese Schwierigkeit wird durch den Gray-Code überwunden.



Fachwörter auf einen Blick

Antwortzeit

Zeitspanne zwischen dem Senden einer Anfrage bei einer Datenstation und dem Empfang einer Antwort bei derselben Station

BAM

Block Availability Map oder Belegungstabelle; Liste der freien Sektoren auf einer Diskette

Emulationsprogramm

Programme, die auf einem bestimmten Rechner lauffähig sind, werden mit Hilfe eines Emulationsprogramms einem anderen Computer angepaßt, so daß sie auch dort arbeiten können



Hilfskanal

Bestimmte Modems für serielle Datenübertragung sehen einen Hilfskanal in der Gegenrichtung zum Datenkanal vor; er hat im allgemeinen eine wesentlich geringere Übertragungsgeschwindigkeit als der Datenkanal

Identifikation

Ein Zeichen oder eine Zeichenfolge, mittels der eine Dateneneinrichtung ihre eigene Identität kenntlich macht, auch Password genannt

Mainframe

Englische Bezeichnung für Großrechner

Mehrfachbetätigung

(rollover) Gleichzeitiges Drücken von zwei oder mehr Tasten

Multiplexer

Gerät, das mehrere Nachrichtenkanäle auf einen oder wenige Kanäle höherer Geschwindigkeit zusammenfaßt, wobei jeder eingehenden Leitung ein entsprechender Anteil der abgehenden Leitung ständig reserviert ist

Notation

Schreib- und Darstellungsweise von Zahlen in der Computertechnik; die am meisten verwendeten Notationen sind binäre, dezimale und hexadezimale; gemeint ist die Schreib- bzw. Darstellungsweise im jeweiligen Zahlensystem

OCR-Leser

Ist die Abkürzung für Optical Character Recognition; sie werden für die maschinelle Erfassung von Standard-Schriften verwendet; OCR-Leser arbeiten zeilen- und seitenorientiert; sie finden Anwendungen bei Schecks, Kreditkarten etc.

Overlay-Technik

Gewährleistet eine ausgezeichnete Nutzung der Speicherkapazitäten durch das Aufteilen von Programmen in einzelne Segmente und den damit zusammenhängenden Überlagerungsmöglichkeiten

Password

siehe Identifikationscode

Piezoelektrisch

Auf dem von Madame Curie entdeckten Effekt beruhend: bestimmte Kristalle laden sich unter Druck elektrisch auf

PL/M

(Programming Language Microprocessor) Von der Firma Intel entwickelte höhere Programmiersprache für Mikroprozessoren; PL/M ist eine Art Dialekt der Programmiersprache PL/1, die die Firma IBM aus ALGOL 60 und COBOL abgeleitet hat; PL/1 wird fast ausschließlich auf Großrechnern eingesetzt

Prompt

Auf dem Bildschirm erscheinende Information, die den Bediener zur Eingabe von Daten auffordert

Referenz-Auflistung

Liste, die von einem Übersetzerprogramm erstellt wurde und mit Befehls- und Speicherplatzangaben ausgestattet ist

RGB-Monitor

In einem RGB-Monitor werden die einzelnen Signale (rote, blaue und grüne Farben sowie Zeilen und Bildschirmsynchronisation) über eine eigene Leitung übertragen

Splitmodem

Datenübertragungseinrichtung mit Multiplexer kombiniert; erlaubt die Benutzung eines Übertragungswegs für mehrere Datenverbindungen; auch: Modem mit Aufteilung in Unterkanäle

Tetrad

Eine Gruppe von vier Bits zum Darstellen von Dezimalziffern in binärer Form

Verdichten

Bezeichnet den Vorgang, der Informationen auf kleinerem Speicherplatz darstellt; Zahlen können auf diese Weise von der ASCII-(1 Ziffer/Byte) in gepackte Darstellung (2 Ziffern/Byte) verwandelt werden

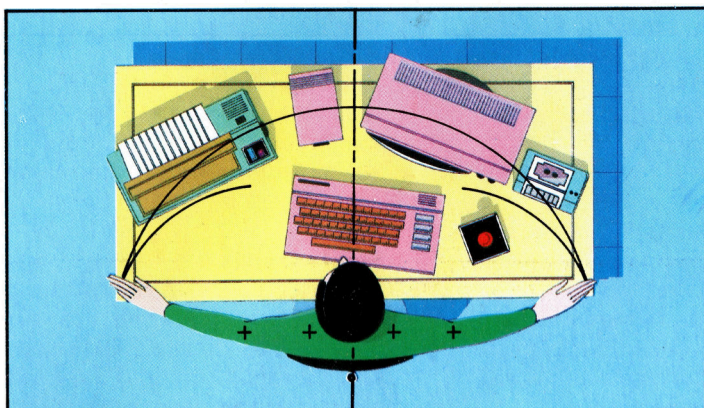
Bildnachweise

309: YRM Architects & Planners
310: Roy Ingram
311: British Airports Authority
311: The Motor Ship
312: British Telecom
313: Tony Lodge
314: Starlord
315, 317, 329, 335: Ian McKinnell
317: Rosalind Buckland
319: Gary Marsh
320, 336: Kevin Jones
321: Chris Stevens
322: Marcus Wilson-Smith
329: David Weeks
330: IBM UK Ltd.
Fachwörter: Peter Fink

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

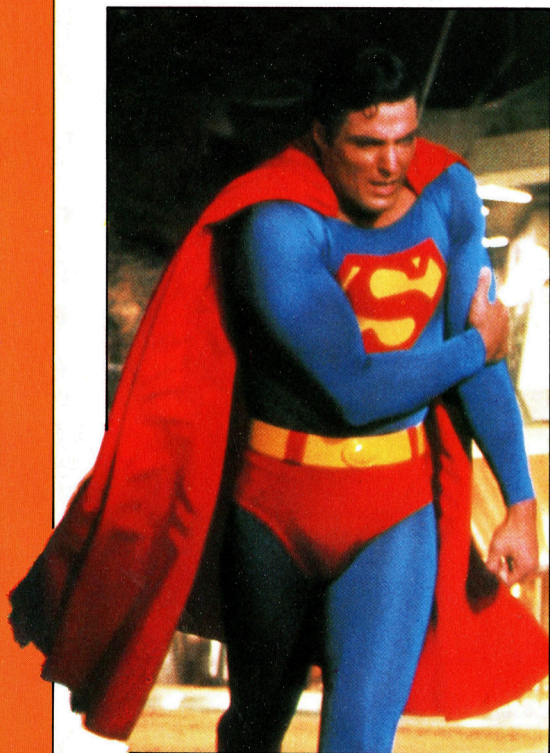
computer kurs

Heft 13



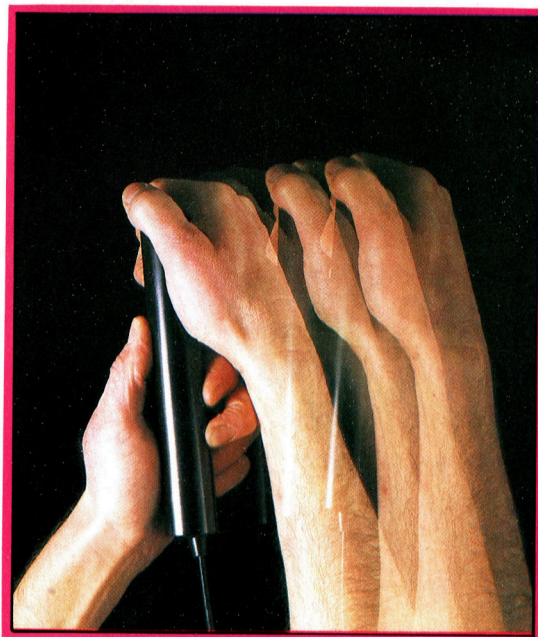
Arbeitsplatz

Wichtig ist nicht nur, daß man sich den richtigen Computer für seine Zwecke auswählt, sondern auch, wie der „Arbeitsplatz“ eingerichtet ist.



Allmächtige Rechner

spielen in der Science-Fiction-Literatur eine große Rolle. Im Film „Superman 3“ hat es der Held mit Computer-Kriminalität zu tun.



Ansteuerung

durch neuartige Joysticks: Diese „Feuer-Knüppel“ arbeiten ohne Mechanik.

+++ Computer und Astronomie +++ Speicherstrukturen +++ Klänge mit dem VC 20
+++ Der Sharp MZ-711 +++ Spannende Spiele +++ BASIC: Datei-Aufbau +++ Der Erfinder Konrad Zuse +++ LOGO: Recursion