

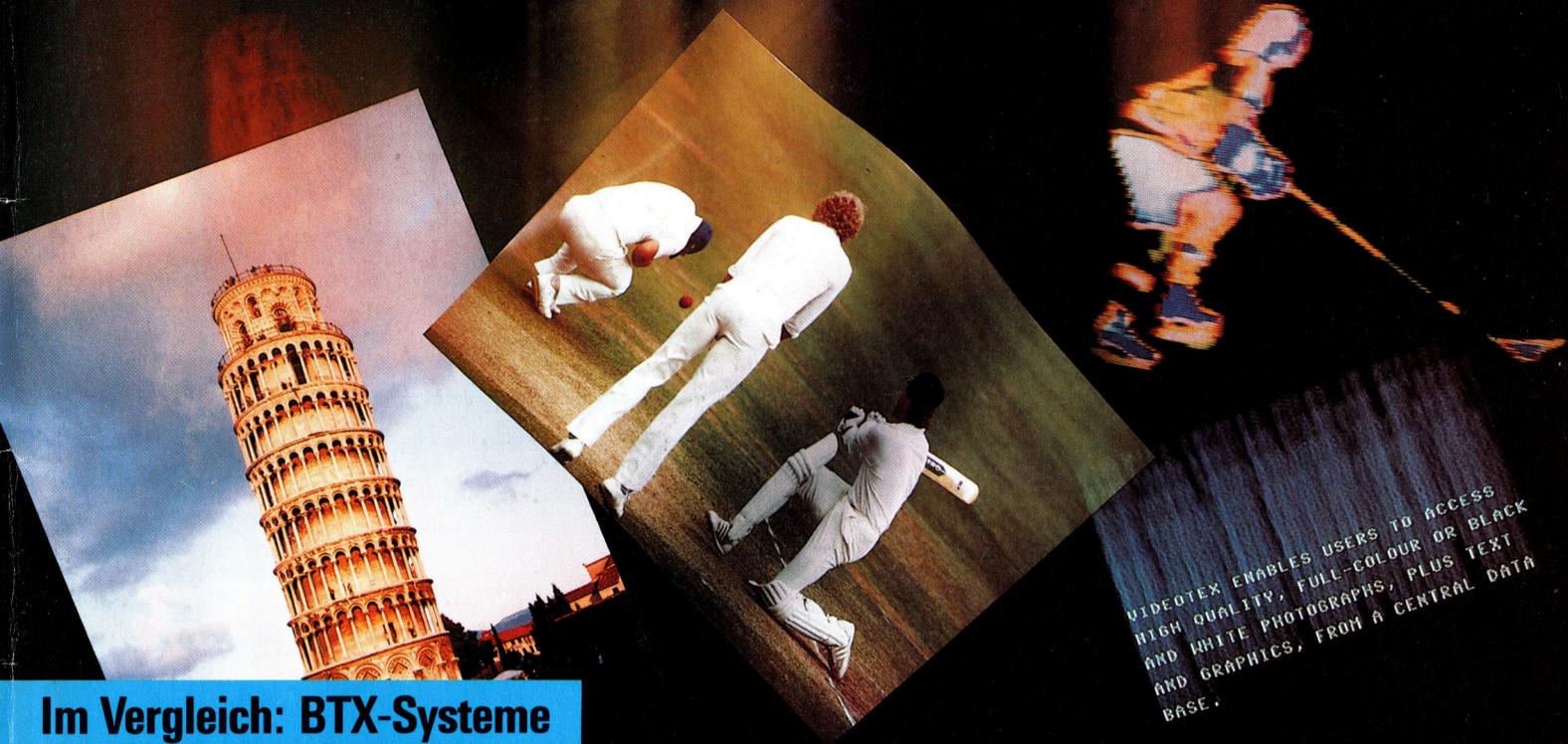
Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Heft **74**

Ein wöchentliches Sammelwerk



VIDEOTEK ENABLES USERS TO ACCESS
HIGH QUALITY, FULL-COLOUR OR BLACK
AND WHITE PHOTOGRAPHS, PLUS TEXT
AND GRAPHICS, FROM A CENTRAL DATA
BASE.

Im Vergleich: BTX-Systeme

Serienbriefe mit Format

Ein Tip für schnelles Messen

Nützliches MS-DOS



computer kurs

Heft 74

Inhalt

Computer Welt



Kabelsalat 2045

BTX im internationalen Maßstab

Buntes Treiben 2055

Die mutigen Taten des Hauses Prism

Sprache



Das Dateiprinzip 2048

Ein Trick sorgt für Übertragbarkeit

Software



MS-DOS: der Boss 2050

Höchst nützliche Directory-Befehle

Die Textmaus 2063

MacWrite stellt seine Talente unter Beweis

Bits und Bytes



Seriell gesteuert 2053

Das Interface 1 schlägt Haken

Spectrum am Netz 2066

Kleiner Rechner in der großen Telekommunikation

Programmier-Service

Briefe mit Format 2056

Serienbriefe – mit dem Heimcomputer kein Problem

Tips für die Praxis



Feine Zutaten 2061

Pinbelegung des Multimeter-Chips

Signalabtastung 2071

Ein Tip für schnelles Messen

BASIC 74



Für Strategen 2068

Dem Go-Programm wird Intelligenz verliehen

Fachwörter von A–Z

WIE SIE JEDE WOCHEN IHR HEFT BEKOMMEN

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

Zurückliegende Hefte

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

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

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

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

Abonnement

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

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.

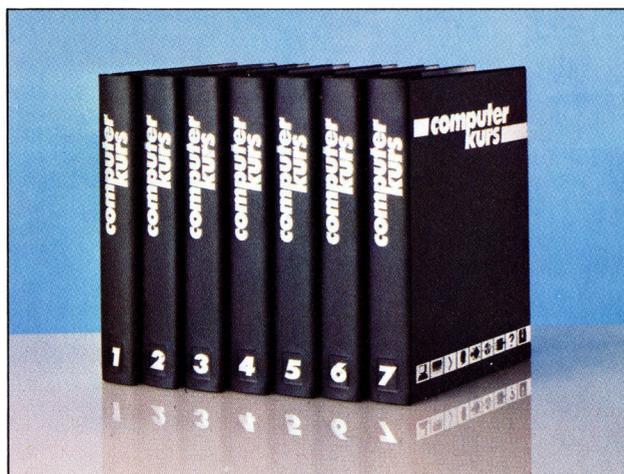
Schweiz: Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

INHALTSVERZEICHNIS

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

Redaktion: Winfried Schmidt (verantw. f. d. Inhalt), Peter Aldick, Holger Neuhaus, Uta Brandl (Layout), Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

Vertrieb: Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1



© APSIF, Copenhagen, 1982, 1983; © Orbis Publishing Ltd., 1982, 1983; © Marshall Cavendish Ltd., 1984, 1985, 1986; **Druck:** E. Schwend GmbH, Schmollerstraße 31, 7170 Schwäbisch Hall



Kabelsalat

BTX-Systeme senden qualitativ hochwertige Bilder über Telefonleitungen oder Funksignale an Computer. Wir zeigen die prinzipielle Funktion des BTX und einige gute Systeme.

Selbst im Bereich der Homecomputer sind die Benutzer inzwischen an perfekte Computergrafiken gewöhnt. Qualitativ hohe Bildübertragung gehört zu den technischen Möglichkeiten des als „BTX“ bekannten Dienstes der Deutschen Bundespost. Das artverwandte britische System heißt „Videotex“. In den USA ist BTX eine Mischung aus Videobildern und Textinformationen, die ausschließlich per Kabel zum Endverbraucher gelangt.

Das Manko in jedem qualifizierten BTX-System ist die Geschwindigkeit, mit der Bilder auf dem Bildschirm des Endgerätes erscheinen. Während der englische Standard „Prestel“ etwa eine Sekunde für den Aufbau einer Datenseite benötigt, kann derselbe Vorgang bei anderen Standards über zehn Sekunden dauern – eine Frage der Datenmenge.

Der Bildschirm ist in als Pixel bezeichnete Punkte unterteilt. Eine typische Prestel-Seite umfaßt 32 Zeilen zu je 234 Pixel. Zum Vergleich: Eine Seite des britischen Foto-Videotext besteht aus 240 Zeilen mit jeweils 270 Pixel. Es muß jedoch für diese sehr hohe Auflösung eine entsprechend höhere Speicherkapazität zur Verfügung stehen (128 kByte gegenüber den lediglich 5 kByte einer Prestel-Seite).

Daten-Kompression wichtig

Eine zunehmend wichtige Rolle beim Einsparen von Speicherplatz kommt daher der Daten-Kompression zu. Foto-Videotext beispielsweise verdichtet Bilddaten von theoretisch 128 kByte bis auf die Hälfte. Eine weitere Einsparung könnten intelligenter Systeme erbringen, die in der Lage sind, auch Bildschirmausschnitte zu verändern.

Bildschirmtext-Systeme sind für jede Anwendung geeignet, in der Text, gemischt mit farbigen Illustrationen, übermittelt werden soll. Daß auch das BTX-System der Deutschen Bundespost noch erheblich mehr leisten könnte, zeigen Anwendungsbeispiele aus England.

Dort nutzen viele Sicherheits-Systeme diese Übertragungstechnologie, zum Beispiel um Namen mit Fotos zu verbinden. So können



etwa von der Polizei Daten und Bilder von zentralen Datenbanken abgerufen werden.

BTX eignet sich gut als Werbemedium. PR-Agenturen, Vertreter und Makler machen regen Gebrauch von Informationssystemen wie Prestel und Compunet, um Kaufinteressierte und Verkäufer zusammenzuführen. Eine weitere Entwicklung in dieser Richtung führt zu Monitoren, die in Schaufenstern aufgestellt sind und fortwährend Angebote der Geschäfte aus einer Datenbank abrufen und anzeigen. Aktuelle Informationen erreichen den Passanten, ohne daß er das Geschäft betreten muß, sich in eine Reihe wartender Kunden zu stellen hat und nach langer Wartezeit erfährt, daß der Artikel ausverkauft ist.

BTX eignet sich wegen des interaktiven Dialogs sogar für Vertragsabschlüsse, etwa Buchungen bei Reisebüros. Wählt man als Verbindung die Telefonleitung oder ein Computernetzwerk, sind Abschlüsse auch über die Grenzen der Länder hinweg möglich.

Der private BTX-Kunde kann ebenfalls per Computer aus dem Sortiment verschiedener Versandhäuser einkaufen. Das britische Einkaufssystem heißt bezeichnenderweise „Armchair Grocer“, was zu Deutsch etwa Lehnstuhl-Laden bedeutet. Über den heimischen Fernseher rufen die Benutzer Kataloge ab und bestellen sofort unter Angabe ihrer Kreditkarten-Nummer.

Die technische Weiterentwicklung eröffnet

Informationssysteme wie BTX ermöglichen die Übertragung von Fotografien, Texten und Computergrafiken an externe Abfrageterminals, wobei sich unterschiedliche Anwendungen im privaten und kommerziellen Bereich ergeben. Lokalnachrichten, klassifizierte Offerten und Produktinformationen werden in interaktiven Systemen gespeichert, wo sie einer großen Anwenderzahl zur Verfügung stehen. Direkte Rückmeldungen erleichtern dem Kunden Bestellungen jeder Art.



immer neue Anwendungsgebiete der digitalen Datenübermittlung. Das amerikanische Unternehmen Datapoint bietet ein Paket mit Namen „Minx“ für Video-Konferenzen per Datennetz an. Minx funktioniert auch in Verbindung mit dem IBM PC und beinhaltet eine Videokamera, ein spezielles Telefon, einen Farbmonitor und einen kleinen Bildschirm. Nach Aufbau der Verbindung über ein öffentliches oder auch firmeninternes Datennetz sehen sich die Gesprächspartner gegenseitig auf den Monitoren. Die Leitung überträgt Videosignal, Ton und Daten simultan. So wird der Austausch von Dateien zwischen den Computern während der Konferenz möglich. Ein solches System erfordert eine Übertragungsrate von zwei Megabytes pro Sekunde.

Grundsätzlich besteht ein BTX-System aus einer Mainframe-Anlage oder zumindest einem Minicomputer, der Daten speichert und Anrufe entgegennimmt. Mit einem Heimcomputer ist diese Aufgabe kaum zu leisten. Die Firma „Soft Machinery“ setzt den Acorn B für ein als „The Gnome at Home“ benanntes, sehr kleines System ein. Erst die Verbindung mehrerer Acorn B über das Econet-Netzwerk ermöglicht die simultane Benutzung durch bis zu acht Anrufer.

Communitel-Version

Die Software besteht aus einer modifizierten Version des Programms „Communitel“. Diese Software setzt den Acorn B als Viewdata-Hauptrechner ein. Communitel ist erfreulicherweise kompatibel zum Prestel-Standard.

Apricot bietet ein Paket mit der Bezeichnung „Apricot Viewdata“ an. Es enthält ein privates Viewdata Hostsystem für acht bis 16 simultane Anwender. Zur Datenübermittlung stehen RS232-Schnittstellen zur Verfügung, an die Mo-

dem oder ein lokales Netzwerk angeschlossen werden können. Da das System Prestel-kompatibel ist, kann es über ein „Gateway“ mit einem größeren Prestel-System verbunden werden. (Gateway ist die gebräuchliche Bezeichnung für einen feststehenden Verbindungsweg zwischen Großrechnern.)

Sehr hohe Bildqualität

Der British Telecom Foto Videotex basiert ebenfalls auf einem Microcomputer – dem IBM PC –, der schwarz/weiße oder farbige Vorlagen durch eine Videokamera aufnimmt und mit Text kombiniert. Die Bildqualität entspricht der einer Fotografie. Sowohl die Host- als auch die Abfrage-Programme sind für den IBM PC ausgelegt.

So oder so

In Großbritannien, wie in Deutschland, gibt es zwei verschiedene Systeme nebeneinander: das mit unserem BTX eng verwandte „Viewdata“ und „Teletext“, ein Gegenstück zu unserem Videotext. Sam Fedida entwickelte die Grundlagen für Viewdata. Auf seinen Arbeiten basierend, entstand 1979 das weltweit erste kommerzielle Prestel Viewdata System.

Der Prestel Standard war überaus erfolgreich und eroberte auch extrem konkurrenzstarke Märkte wie die USA oder Deutschland. Prestel ist natürlich nicht das einzige Viewdata-System in England.

Während BTX- und Viewdata-Systeme die Telefonleitungen als Übertragungsmedium benutzen, sind bei Teletext und unserem Videotext die Daten dem Fernsehsignal zugemischt. Daraus ergibt sich wieder eine Mixtur aus Text und Grafik. Oftmals findet sich in Programmzeitschriften der Hinweis „Untertitel auf Tafel 150“. In das normale



Verdrahteter Weitblick

Obwohl Fernsehkabel ein ideales Medium für Informationsdienste ist, reagieren die englischen Verbraucher nur zögernd darauf. Im Jahre 1985 waren erst 127 000 Haushalte an das Kabelnetz angeschlossen. Wenn sich die neuen Kabelgesellschaften, wie Westminster Cable, am Markt etablieren, erwartet man einen schnellen Zuwachs bis auf 300 000 Anschlüsse.

1984 starteten British Telecom und Micronet 800 ein gemeinsames Experiment, genannt „Gamestar“. Über Kabel versandte diese Gesellschaft Computer-Software direkt in die Haushalte. Die Kunden mieteten einen Sinclair Spectrum und einen modifizierten Modem, um Programme via Teletext zu laden. „Rediffusion“ übernahm diesen neuen Service, doch das Projekt scheiterte an technischen Problemen.



Privatfernsehen

Das British Telecom Foto Videotex-System benutzt IBM PCs und offeriert umfangreiche Editierfunktionen zum Überarbeiten, Vergrößern und Verkleinern der aufgenommenen Bilder sowie deren Kombination mit anderen Bildern, Texten und Grafiken. Die Anwendungen erstrecken sich von lokalen Nachrichten über Annoncen und jeder Art Werbung bis zu Personal-Dateien und interner Kommunikation für größere Firmen.

Das System verbindet Benutzerterminals, Editierstationen und ein zentrales Computersystem und ermöglicht Kommunikation zwischen den Terminals untereinander oder im Dialog mit dem Rechnersystem. Die Bildqualität entspricht generell dem des Fernsehprogramms.

Verbindungen

BTX-Informationen können durch verschiedene Kabelarten übertragen werden. Die physikalische Verbindung kann aus einem verdrehten Leitungspaar (Telefonleitung), einem Leitungsstrang mit mehreren Adern (wie ein RS232-Anschluß für Modems und Drucker) oder auch einem Koaxialkabel (gleich dem Antennenkabel des Fernsehsenders) bestehen.

Die Deutsche Bundespost nutzt für ihr System das normale Telefonnetz als Zugang zu den BTX-Rechnern. Dieses Leitungsnetz, in Großbritannien PSTN (Public Switched Telephone Network) genannt, kommt auch bei British Telecom zum Einsatz. Über das PSTN kann der Benutzer mit Hilfe eines Modems auch auf das Datennetz PSS (Packet Switch-Stream) zugreifen. Dieses Netz ist mit dem deutschen Datex-P vergleichbar. Ein Vorrechner (in als Deutschland „PAD“, in Großbritannien als „NODE“ bezeichnet) verschickt die gesammelten Daten des Benutzers in „Paketen“ über Standleitungen mittels des X25-Protokolls an den Adressaten. Der Empfänger kann mit seinem Rechner direkt oder über einen weiteren PAD mit dem Datennetz verbunden sein. Diese Datennetze sind zu IPSS (International Packet SwitchStream), einem internationalen Verbundnetz, zusammengeslossen und arbeiten intern mit Übertragungsraten von 48 000 Baud. Wenn allerdings die Telefon-Verbindung zum PAD durch Fremd-Geräusche sehr gestört ist, sollte die Datenübertragung per Protokoll geschehen, was natürlich die Geschwindigkeit wesentlich vermindert.

Aus diesem Grund eignen sich vom normalen Netz und dessen Störungen isolierte Leitungen besser zur Übertragung. Die deutsche Bundespost bietet solche Leitungen für private Datenverbundnetze an. Auch die „X-stream services“ der British Telecom benutzen separate Leitungen oder auch eigene Nachrichtensatelliten. Die Kosten solcher Leitungen richten sich nach der Übertragungsgeschwindigkeit, sind aber im Verhältnis zu den Kosten eines ungenauen oder zu langsamen Datenaustausches relativ gering.

Durch eine flächendeckende Verkabelung sind Systeme realisierbar, die Datenmengen von 64 KBit bis zwei Megabytes pro Sekunde übertragen.

So sind dann ideale Voraussetzungen für Informations-Systeme geschaffen.



Fernsehbild blendet der Videotext-Empfänger dann die empfangenen Daten ein. Dieser kostenlose Service kann nur mit speziell ausgestatteten Fernsehgeräten genutzt werden. Die drei wichtigsten Videotext-Anbieter in England sind Ceefax von BBC, Oracle von ITV und 4-Tel von Channel 4.

An dem Namens-Wirrwarr ist bereits ersichtlich, daß bisher nur wenig auf Kompatibilität der verschiedenen nationalen Systeme geachtet wurde. Videotext heißt in England Teletex, darf aber nicht mit dem erweiterten Telex-Dienst „Teletex“ verwechselt werden. Videotex oder Viewdata trägt in Deutschland die Bezeichnung BTX. Gemeint ist die gleiche Idee, doch eignen sich unsere BTX-Decoder nicht zum Empfang englischer BTX-Informationen. Die großen englischen Anbieter British Telecom und BBC/IBA trafen erste Vereinbarungen, um einen allgemeingültigen Standard zu schaffen. Der vielversprechende Terminus „Welt-System“ wird seitdem auf alles angewandt, was mit den Prestel- oder Oracle-Ceefax-Standards konform ist.

Das Dateiprinzip

Bei vielen Sprachen gibt es Schwierigkeiten, das beste Verhältnis zwischen völlig freien und exakt definierten Befehlsformaten zu finden. In C wurde dieses Problem auf sehr praktische Weise gelöst.

Wie bei vielen modernen Programmiersprachen ist auch in C die Ein- und Ausgabe (E/A) nicht fest in die Sprache eingebaut, da umfassende E/A-Definitionen (wie etwa bei COBOL und FORTRAN) verhindern würden, daß Anwendungen sich auf andere Maschinen und Systeme übertragen lassen.

So kann COBOL ohne Spracherweiterungen (die außerhalb des Standards liegen) nicht einmal Zeichen direkt von der Tastatur lesen. Wenn jedoch überhaupt keine E/A definiert ist, lassen sich Programme auf unterschiedlichen Betriebssystemen und Maschinen nur unter Schwierigkeiten zum Laufen bringen. Die Tastatureingabe bei PASCAL, die in den einzelnen Sprachversionen verschieden ausgelegt wurde, ist dafür ein gutes Beispiel.

Das Modul `stdio.h` mit seinen vorgefertigten E/A-Definitionen auf hoher und maschinennaher Sprachebene sorgt dafür, daß C eine Art Standard erhält. Da der Inhalt dieser Routinen von einer Anwendung zur anderen variieren kann, verwendet man am besten nur die Funktionen auf hoher Ebene.

Die Standardfunktionen für Bildschirmausgabe (`printf`) und Tastatureingabe (`scanf`) wurden bereits erläutert. Es gibt eine ganze Reihe weiterer Funktionen zur Steuerung von Bildschirm, Tastatur, Speicher und anderen Geräten. C behandelt Dateien und Geräte gleich und vereinfacht damit das gesamte Konzept der E/A. Für C ist eine Datei ein Bytestrom, aus dem sich die Eingabefunktionen ein Byte oder eine Bytegruppe herausnehmen und auf dem die Ausgabefunktionen ihre Daten aussenden. Ist der Strom mit einer Diskette verbunden, dann kann man damit Adressen ansprechen und wahlfrei darauf zugreifen. Der logische Unterschied zwischen Datei und Gerät besteht somit nicht mehr.

Normalerweise müssen Dateien vor ihrem Einsatz eröffnet werden. In jedem C-Programm sind jedoch drei Ströme ständig offen: `stdin` (normalerweise an die Tastatur angeschlossen), `stdout` und `stderr` (normalerweise mit dem Bildschirm verbunden). In vielen Betriebssystemen (z. B. Unix) läßt sich die Ein- und Ausgabe umkehren und so die Zuordnung verändern. Die Funktionen `printf` und `scanf` arbeiten mit `stdout` und `stdin`, sind aber nur Spezialfälle von `fprintf` und `fscanf`.

```
printf( . . . );
fprintf(stdout, . . . );
```

haben die gleiche Wirkung. Zwei ähnliche Funktionen – `sprintf` und `sscanf` – führen im Hauptspeicher E/A-Vorgänge mit Strings aus.

Der Pointer `stdout` zeigt auf ein Datenelement vom Typ `FILE`, das normalerweise über ein Makro (`#define`) in `stdio.h` definiert ist. Die Eröffnung einer Datei (für eine Diskette oder ein Gerät) erledigt die Funktion `fopen` (dateiname, `dateiart`). Der String `dateiname` kann sich je nach Betriebssystem auf ein Gerät oder eine Datei beziehen. `dateiart` kann „r“ für Lesen (aus der Datei), „w“ für Schreiben oder „a“ für das Anhängen von Daten (an die Datei) sein. Wenn eine mit „a“ oder „w“ eröffnete Datei nicht existiert, wird sie neu angelegt. Bereits bestehende, mit „w“ eröffnete Dateien werden überschrieben. Der Dateipointer (eine Ganzzahl vom Typ „long integer“, die die aktu-

Ernstfall

Es gibt viele Programmbibliotheken mit interessanten Funktionen und Makros, auf die wir aus Platzgründen nicht eingehen können. In der Datei `ctype.h` finden Sie eine Reihe Funktionen zur Bearbeitung einzelner Zeichen. Die Datei läßt sich mit

```
#include (ctype.h)
```

in ein Programm einschließen. Die erste Tabelle zeigt, wie getestet werden kann, ob Zeichen bestimmte Eigenschaften haben. Ein Wert ungleich Null bedeutet, daß das Zeichen die Eigenschaft besitzt (`true`).

Name	,True' falls
<code>isalpha(c)</code>	c = Buchstabe
<code>isupper(c)</code>	c = Großbuchstabe
<code>islower(c)</code>	c = Kleinbuchstabe
<code>isdigit(c)</code>	c = Zahl
<code>isxdigit(c)</code>	c = Hexadezimalzahl
<code>isspace(c)</code>	c = Leerzeichen
<code>isalnum(c)</code>	c = Buchstabe oder Zahl
<code>ispunct(c)</code>	c = Interpunktionszeichen
<code>isprint(c)</code>	c = druckbares Zeichen
<code>isctrl(c)</code>	c = Steuerzeichen
<code>isascii(c)</code>	c = ASCII-Code

Die zweite Funktionsgruppe führt Umwandlungen aus:

Name	wandelt
<code>toupper(c)</code>	c in Großbuchstaben
<code>tolower(c)</code>	c in Kleinbuchstaben
<code>toascii(c)</code>	c in ASCII-Code

Charakterbewertung

/* Dieses Programm zaeht die Zahl der Worte und Zeichen in der Datei, deren Name in der Befehlszeile angegeben ist */

```
# include<stdio.h>
# include<ctype.h>
main(argc,argv)
int argc;
char *argv;
{
    int char_count=0, word_count=0,c,
    inword =0; FALSE
    FILE *in_file, *fopen();
    /* Beachten Sie, dass der Dateiname und die
    Funktion fopen als Pointer vom Typ FILE deklariert sind */
    /* Testen, ob die korrekte Argumentenzahl vorhanden ist */
    if (argc !=2)
    {
        fprintf(stderr, "/nusage is %s filename/n",
        *argv);
    }
    /* Denken Sie daran, dass der Dateiname der erste
    Eintrag im Array argv ist */
    exit (1);
}
/* Datei eroeffnen und testen, ob sie existiert.
++argv zeigt auf den Dateinamen */
if ((in_file=fopen(*++argv,"r"))==NULL)
{
    fprintf(stderr, "/ncannot open &s/n",
    *argv);
}
/* Denken Sie daran, dass argv nun auf den
Dateinamen zeigt */
exit(1);
}
while ((c=getc(in_file)) !=EOF)
{
    ++char_count;
    if(inword)
    {
        if(isalnum(c))
        /* ein Leerbefehl */
        else
        {
            inword=0;
            ++word_count;
        }
    }
    else
    if (isalnum(c))
        inword=1;
}
if (inword)
    ++word_count;
printf("/nzeichenzahl=%d",char_count);
printf("/nzahl der worte=%d/n",word_count);
fclose(in_file);
}
```

elle Position im Bytestream angibt) wird an den Anfang einer mit „r“ oder „w“ eröffneten Datei gestellt. Die Funktion fopen liefert einen Pointerwert, den andere Funktionen verarbeiten können. Steht er auf Null, ist der Zugriff auf die Datei nicht möglich. Die Funktion fclose(pointer_auf_die_datei) schließt eine offene Datei.

E/A-Standardfunktionen

In der Standard-Programmbibliothek von C gibt es folgende weitere E/A-Funktionen:

getc(pointer_to_file) Holt das nächste Byte aus der Datei, die mit „r“ eröffnet wurde. Das Ergebnis hat die Form einer Ganzzahl. Bei Dateiende erscheint ein spezieller EOF-Wert. Die Funktion läßt sich als Makro einsetzen.

getchar() Entspricht **getc(stdin)**

fgetc(pointer_to_file) Setzt das Zeichen c wieder in die Datei zurück, aus der es gelesen wurde. Das Ergebnis hat die Form einer Ganzzahl. Bevor das erste Zeichen in die Datei zurückgeschrieben werden kann, muß zumindest ein Zeichen von dort gelesen worden sein.

putc(c,pointer_to_file) Übergibt das Zeichen c an die Ausgabedatei und liefert den Ganzzahlenwert von c. Als Makro einsetzbar.

putchar(c) Entspricht **putc(c,stdout)**

fputc(c,pointer_to_file) Entspricht **putc**, wird aber nur als Funktion eingesetzt

gets(s) Der String s ist Pointer auf char und liest bis zum nächsten Zeilenvorschub Zeichen aus stdin. Der Zeilenvorschub wird nicht mehr an den String übergeben und der String nun mit „/0“ abgeschlossen. Zurückgegeben wird der Wert von s.

fgets(s,n,pointer_to_file) Liest Zeichen aus der Datei und speichert sie im String s, bis entweder n-1 Zeichen gelesen wurden oder der nächste Zeilenvorschub erscheint. Der Zeilenvorschub wird an s übergeben und s mit „/0“ abgeschlossen. Zurückgegeben wird der Wert von s.

puts(s) Gibt den String s an stdout aus und hängt einen Zeilenvorschub an.

fputs(s,pointer_to_file) Gibt den String s ohne Zeilenvorschub an die Datei aus.

fseek(pointer_to_file,offset,place) Verschiebt den Dateipointer auf dem Bytestrom bis zum Offset oder von der angegebenen Position an um eine bestimmte Anzahl Bytes. Place kann folgende Werte haben: 0 für den Dateianfang, 1 für die aktuelle Position und 2 für das Dateiende. Offset sollte eine Ganzzahl vom Typ „long int“ sein.

rewind(pointer_to_file) entspricht:

fseek(pointer_to_file,OL,0)

ftell(pointer_to_file) Liefert das aktuelle Offset (Typ „long int“) vom Dateianfang aus.

unlink(filename) Löscht die angegebene Datei im Directory. Liefert -1, wenn es die Datei nicht gibt, und 0, wenn sie existiert.

exit(status) Beendet ein Programm und liefert den Ganzzahlenwert von Status an das Betriebssystem oder das aufrufende Programm. 0 zeigt ein normales Ende an.



MS-DOS: der Boss

Die Befehle des Betriebssystems MS-DOS erfüllen fast alle normalen Anforderungen – vom Durchsuchen des Inhaltsverzeichnisses bis zum Drucken einer Datei wird nahezu alles möglich.

Beim Einschalten eines Systems mit Festplatte wird MS-DOS automatisch geladen (auf einem Diskettensystem durch Einlegen der Systemdiskette). Viele Geräte, die mit einer IBM-ähnlichen Tastatur ausgerüstet sind, können einen „Warmstart“ (oder Reset) mit der herkömmlichen Technik des IBM PC „Control-Alt-Delete“ auslösen. Dabei werden die Tasten CTRL und ALT links auf der Tastatur untergehalten und gleichzeitig die DEL-Taste (rechts) gedrückt. Nach dem Laden von MS-DOS und der Systeminitialisierung wird dann geprüft, ob sich auf dem angemeldeten Laufwerk die „Batch“-Datei AUTOEXEC.BAT befindet. Wenn es diese Datei gibt, werden alle darin gespeicherten Befehle automatisch wie Tastatureingaben ausgeführt.

Sie können beispielsweise eine einfache AUTOEXEC.BAT-Datei mit dem Inhalt

```
MENU
```

anlegen, wobei MENU.COM (oder MENU.EXE) der Name des gewünschten Anwendungsprogramms ist. Das Programm wird nun bei jedem Booten von dieser bestimmten Systemdiskette automatisch ausgeführt. Diese Technik wird „Turnkey“ – auf gut deutsch soviel wie schlüsselfertiges System genannt.

Besonders die Versionen 2 und 3 des MS-DOS (und PC-DOS) sind außerordentlich flexibel. Sie beinhalten alle Diskettenbefehle als eigenständige EXE- oder COM-Dateien. In dieser Folge beschäftigen wir uns jedoch nur mit den Befehlen, die in allen Versionen des MS-DOS und PC-DOS identisch sind und über den DOS-Befehlszeileninterpreter bzw. das Rahmenprogramm COMMAND.COM (den einzigen Teil des DOS, den der Anwender „sehen“ kann) angesprochen werden.

Auf allen Systemen wird der Befehl „Directory“ vermutlich am häufigsten eingesetzt. Er zeigt alle Dateinamen einer Diskette oder – von DOS 2 an aufwärts – ein „Directory“ (d. h. einen bestimmten Teil des Disketteninhalts). Hier das Grundformat:

```
dir
```

Wie bei CP/M (an das sich DOS 1 eng anlehnt) listet dieser Befehl alle Dateien der angemeldeten Diskette. Das Directory anderer Disketten läßt sich über zusätzliche Angaben abrufen. Auch hier besteht Ähnlichkeit zu CP/M, da sich auch „Wildcards“ verwenden las-

sen: Dateigruppen werden mit ? (stellt ein einzelnes Zeichen dar) und * (stellt eine Zeichengruppe oder fehlende Zeichen dar) in Kurzform angesprochen.

Dateinamen können (wie bei CP/M) ein bis acht Zeichen enthalten und die Dateierweiterungen von null bis drei Zeichen. Das MS-DOS Directory liefert jedoch weit mehr Informationen und hat ein anderes Format. MS-DOS listet nicht nur Dateinamen, sondern auch die Dateilänge (in Bytes) sowie Datum und Zeit, zu der die Datei auf die Diskette geschrieben wurde. Hier ein Beispiel:

```
A>dir
```

```
Volume in drive A has no label
Directory of A:
```

```
COMMAND.COM  22672  9-03-85  11:36a
AUTOEXEC.BAT  128    22-07-85  5:15p
WP            EXE  73156  12-10-84  12:07p
WPMMSG       OVR  38269  27-09-84  12:02p
WPINST       EXE  56385  3-10-84  12:04p
WPCOUNT      PAS  4986   1-08-85  3:11p
WPCOUNT      OBJ  3874   1-08-85  3:12p
WPCOUNT      EXE  8385   1-08-85  3:14p
WPCOUNT      BAK  4732   31-07-85 11:23p
MARY1        TXT  634    7-08-85  11:05a
INVOICE1     885   477    21-08-85 9:36a
MARY2        TXT  938    15-08-85 12:47p
```





Früher hatte das Datum amerikanisches Format (Monat–Tag–Jahr), neuere DOS-Versionen enthalten jedoch inzwischen das europäische Format und die speziellen Zeichensätze der europäischen Sprachen. Obwohl die Dateigröße in Bytes angegeben ist, belegen die Dateien auf der Diskette mehr Platz, da sie nur sektorenweise (normalerweise 512 Bytes) geschrieben werden.

DOS gibt weiterhin Auskunft über die Zahl der Dateien und den freien Platz auf der Diskette. Viele Funktionen des CP/M-Dienstmoduls STAT.COM wurden in den dir-Befehl integriert, der nun weit mehr Informationen liefert als die CP/M-Version. Der Ausbau des Befehls wurde möglich, da bei MS-DOS die Grenze von 64 KByte nicht besteht.

CP/M-Systeme waren fast immer auf diese maximale Speichergöße ausgerichtet, wenn sie nicht mit der Technik des „Paging“ arbeiten. Um Speicherplatz einzusparen, wurde nur das Allernotwendigste in die eingebauten Befehle übernommen. Informationen über Dateigrößen und zusätzliche Eigenschaften lieferten Diskettenbefehle (wie STAT).

Auch MS-DOS besitzt viele Module dieser Art. Da 16-Bit-Systeme über einen weit größeren RAM-Bereich verfügen (üblich sind 256 KBytes und mehr), konnten auch die eingebauten Funktionen umfangreicher und flexibler gestaltet werden. Die Version 3 des DOS belegt über 60 KByte des Arbeitsspeichers – die genaue Bytezahl hängt von der (OEM) Firma ab, die das System zusammenstellt. MS-DOS arbeitet mit einer Systemuhr – selbst auf Systemen ohne eine echte batteriegebufferte Uhr, die auch nach Abschalten des Computers weiterläuft. Für diesen Fall gibt es eine Softwarelösung, die mit Interrupts arbeitet.

Die „Softwareuhr“ wird standardmäßig (auf z. B. 1-07-86 12:00:00) initialisiert und muß nach dem Booten mit den DOS-Befehlen date und time per Hand gesetzt werden. Die Befehle zeigen Datum und Zeit an und lassen den Ein-

trag neuer Werte per Tastatur zu. Die Eingaben müssen das gleiche Format haben, brauchen aber nicht vollständig zu sein.

time 14

setzt die Systemuhr auf 14:00:00 (die letzten beiden Ziffern geben Hundertstelsekunden an, selbst wenn die Zeitmessung nicht in dieser hohen Auflösung arbeitet). Jede neu angelegte Datei wird mit den aktuellen Systemwerten gekennzeichnet und erscheint im Directory daraufhin als

NEUDATEI TXT 512 21-08-86 2:05p

Alle MS-DOS-Programme erhalten die Dateierweiterung EXE (EXEcutable – ausführbar) oder COM (COMmand – Befehl). Einziger Unterschied sind die Speicherbereiche, in die die Dateien geladen und von wo sie ausgeführt werden. EXE-Dateien sind im allgemeinen flexibler und orientieren sich an den Grenzen der „Paragrafen“ (Computerworte), während COM-Dateien immer an einer „Segmentgrenze“ anfangen müssen (d. h. am Anfang eines 64 KByte Speicherbereiches – wie von Intel für die 8086-Familie festgelegt). Weitere Namensweiterungen können Spezialdateien bezeichnen oder aus anderen Gründen vergeben werden. So erhalten Texte oft die Erweiterung .TXT (oder .DOC), Maschinencodeteilen im Standard Intel Format (verschiebbar) .OBJ und Quelltexte für Programme in PASCAL, FORTRAN und C .PAS, .FOR oder .C.

Quellennachweise

Das Directory gibt weiterhin darüber Auskunft, daß das Programm WCOUNT.EXE auf einen Quelltext in PASCAL zurückgeht, der am 1. August um 3 Uhr 11 nachmittags angelegt wurde. Der Text wurde eine Minute später in eine Datei mit dem Intel Format .OBJ kompiliert und zwei Minuten später zu dem endgültigen .EXE-Programm gelinkt. Die vorhergehende Quelltextversion (WCOUNT.BAK), die von der Textverarbeitung (WPEXE) als Sicherheitskopie (.BAK) angelegt wurde, zeigt, daß der Programmierer damit vermutlich Probleme hatte, da sie erst spät in der Nacht des 31. Juli gespeichert wurden. Aus dem Namen des Programms könnten wir schließen, daß damit die Wörter einer Textdatei gezählt werden sollen – all diese Information liefert ein verhältnismäßig kurzer Blick auf das Directory.

Der Befehl dir hat aber noch andere Tricks auf Lager, die ihn weiter von der CP/M-Version abheben. So lassen sich an den Befehl Optionen anhängen. Option P blendet jeweils nur einen Bildschirminhalt auf (oder mehrere nacheinander) und wartet auf einen Tastendruck, bevor er mit dem Listing fortfährt. Das ist besonders bei Disketten praktisch, auf denen Hunderte von Dateien untergebracht sind. Die Option W unterdrückt die Darstellung der

Glück gehabt

Hier die Geschichte, wie Bill Gates von Microsoft den Auftrag erhielt, IBM ein 16-Bit-Betriebssystem (MS-DOS) zu liefern. Nach den ersten Anfragen von IBM war Microsoft unsicher, ob es den Vertrag für das OS annehmen sollte. IBM entschied sich daher, Digital Research aufzusuchen, um den Auftrag mit Gary Kildall zu besprechen, der durch die Entwicklung von CP/M bereits in Computerkreisen bekannt geworden war.

An diesem Tag war Kildall jedoch nicht in seinem Büro. Nachdem die IBM-Delegation zwei Stunden auf ihn gewartet hatte, reiste sie wieder ab und unterzeichnete den Vertrag mit Microsoft. So wurde ein Industriestandard von einer Firma entwickelt, die nur durch eine BASIC-Version bekannt geworden war und kaum Erfahrung mit einem OS hatte.

Zusatzinformationen und listet nur die Dateinamen: fünf pro Zeile. Viele interne und externe DOS-Befehle haben Optionen, die von einem Schrägstrich angeführt werden:

```
dir c:*.exe/p/w
```

bringt beispielsweise die Namen von 115 (5 x 23) .EXE-Dateien auf eine Bildschirmseite.

Ein weiterer Befehl (den es auch unter CP/M gibt) stellt den Inhalt einer Textdatei dar. Hier das Format:

```
type{laufwerk:}dateiname
```

Der folgende Befehl bringt das MODULA-2-Programm MOUSE.MOD (in Laufwerk B) auf den Schirm:

```
type b:mouse.mod
```

Wie bei CP/M aktiviert ein - vor dem Return eingegebenes - Control-P den Drucker für einen Ausdruck.

MS-DOS besitzt jedoch eine alternative Möglichkeit, die in einiger Hinsicht besser, in jedem Fall aber flexibler ist. Während CP/M den Diskettenbefehl PIP.COM für die Daten-

übertragung einsetzt, wurde in MS-DOS der sehr praktische COPY-Befehl eingebaut. Hier das einfachste Format:

```
copy{laufwerk:}quellename{ziel}
```

Ziel kann jeder andere Dateiname oder eine Systemkomponente wie CON (Console), LST oder PRN (Listenausgabe oder Drucker) sein. Einen Ausdruck wie type erzeugt auch

```
copy b:mouse.mod prn
```

Hier brauchen Sie kein Control-P anzugeben und verhindern weiterhin, daß außer der Datei noch andere Zeichen auf dem Papier erscheinen. Mit dem type-Befehl wird oft ein Systemprompt (z. B. A>) angehängt.

Die Datei macht's möglich

Der Befehl copy hat drei Optionen: /a kopiert ASCII-Dateien und setzt dabei die Dateiendmarkierung (Control-Z) ein; /b nimmt das physische Dateiende und ignoriert alle Control-Z; /v verifiziert jede Datenübertragung (diese Option läßt sich auch mit „verify on“ permanent anschalten). All diese Funktionen werden durch eine Datei möglich, in der die Länge jeder Datei des Directories eingetragen ist. Wenn kein Ziel angeboten wird, nimmt copy an, daß eine Datei mit dem gleichen Namen wie die Quelle(n) auf dem angemeldeten Laufwerk gemeint ist (z. B. copy c:*.txt/a/v).

Dieser integrierte Befehl bietet noch viele andere Möglichkeiten, darunter auch die Dateizusammenstellung:

```
copy b:*.txt bigtext.txt
```

stellt alle .TXT-Dateien auf Laufwerk B zu der Datei bigtext.TXT auf dem angemeldeten Laufwerk in Reihenfolge ihres Directoryeintrags zusammen. Einzelne Dateien lassen sich mit + zusammenschließen, wie in

```
copy c:x1.wr+k2.dat+b:x3.frm xxx.neu
```

Wenn Sie Daten an eine bestehende Datei anhängen wollen, brauchen Sie nur die Zielangabe wegzulassen:

```
copy seite 1.txt+seite2.txt+seite3.txt
```

Der Inhalt von seite2.txt und seite3.txt wird dabei an das Ende von seite1.txt angehängt und unter diesem Namen gespeichert.

MS-DOS hat auch einen Systemeditor (EDLIN.EXE), den wir uns zusammen mit anderen Diskettenbefehlen in der nächsten Folge ansehen. Sie können Textdateien jedoch auch ohne jede Textverarbeitung durch

```
copy con neudatei.txt
```

anlegen. Nach der Eingabe wird die Datei mit Control-Z (Dateiendmarkierung) geschlossen und gespeichert. Im Gegensatz zu pip (CP/M) muß danach jedoch noch ein ENTER eingegeben werden.

Integrierte MS-DOS-Befehle

Befehl	Funktion	Optionen	Beispiel
 copy	Datei kopieren	/a/b/v	copy {Laufwerk:}Wildcard{Ziel}
 date	Datum anzeigen/ setzen		date{dd-mm-jj} date{mm/dd/jj} (US-Version)
 del	Datei löschen		del{Laufwerk:} {Wildcard}
 dir	Dateien listen	/p/w	dir{Laufwerk:} {Wildcard}
 erase	siehe del		
 ren	Dateien umbenennen		ren{Laufwerk:} Name1 Name2
 rename	siehe ren		
 time	Zeit anzeigen/ setzen		time{ss:mm:ss}
 type	Textdateien listen		type{Laufwerk:} Dateibezeichnung

Zeichenerklärung:

{ } Angaben in geschweiften Klammern sind optional
| ein vertikaler Strich trennt alternative Möglichkeiten

Zeichen: alle ASCII-Zeichen außer ",?*<>:;[]\|
Dateiname: Zeichen{Zeichen} (maximal acht)
Dateierweiterung: {Zeichen} (maximal drei)
Dateibezeichnung: Dateiname{Dateierweiterung}
Wildcard: Dateibezeichnung enthält Wildcardzeichen (* und ?)
Gerät (Device): CON | LST | PRN | AUX | LPT1 etc.
Ziel: Dateibezeichnung | Gerät
Laufwerk: A | B | etc. (normalerweise bis P)



Seriell gesteuert

Das Interface 1 des Spectrum bietet Maschinencodeprogrammierern viele interessante Routinen. In diesem Abschnitt untersuchen wir, wie die RS232-Schnittstelle mit „Hakencodes“ gesteuert wird.

Sehen wir uns zunächst die neunpolige RS232-Buchse der seriellen Schnittstellen genauer an (unser Bild zeigt die Funktionen der einzelnen Kontakte). Wieviele Leitungen Sie benötigen, hängt im wesentlichen vom Peripheriegerät ab. Beim Betrieb der RS232 müssen jedoch zumindest die Leitungen 2, 3 und 7 verbunden sein und die Leitungen 4 und 5 mit 9 „gebrückt“ werden, um auf logisch Eins zu stehen. Die Brücken verhindern, daß sich der Spectrum „aufhängt“, wenn das Empfangsgerät am anderen Ende des Kabels nicht bereit ist, Daten entgegenzunehmen. Da dabei Daten verloren gehen, sollten Sie zur Vorbeugung versuchen, immer eine vollgültige RS232-Verbindung herzustellen.

Über die RS232-Schnittstelle lassen sich etwa die verschiedensten Drucker an den Spectrum anschließen. Ist die Kabelverbindung einmal hergestellt, übernimmt die Software die Steuerung. Wenn Sie Text zum Drucker senden wollen, müssen Sie zunächst die Baudrate mit dem BASIC-Befehl FORMAT setzen und dann folgenden Code ausführen:

```
CLOSE #3
OPEN #3;"T"
```

Damit wird Strom 3 dem Kanal T zugeordnet, der den Textkanal der RS232-Schnittstelle darstellt. Die Unterschiede zwischen „Text-“ und „Binärströmen“ sind im Handbuch des Interface 1 beschrieben. LLIST und LPRINT senden nun Daten zu dem Gerät, das an das Interface 1 angeschlossen ist.

Die Daten werden mit einem Startbit, acht Datenbits und zwei Stopbits übertragen – die entsprechende Formatierung erledigt das Betriebssystem. Wichtiger ist jedoch die Baudrate, also die Sendegeschwindigkeit der Daten. Die Baudrate wird normalerweise mit dem Befehl FORMAT gesetzt, dem aber nur bestimmte Geschwindigkeiten zur Verfügung stehen. Obwohl bei RS232-Schnittstellen nur selten Daten verloren gehen, sollten Sie die Baudrate mit wachsender Entfernung zwischen Sender und Empfänger heruntersetzen.

Bevor Sie die folgenden Techniken einsetzen, müssen Sie die Interfacevariablen mit dem Hakencode 49 anlegen. Die RS232-Schnittstelle wird über zwei Hakencodes und drei Systemvariablen angesprochen. Zunächst die Variablen:

● **BAUD** (bei 23747 und 23748) ist eine Variable im Zwei-Byte-Format, deren Wert die Baud-

rate für das Senden und Empfangen von Daten bestimmt. Hier stoßen wir jedoch auf eine der Grenzen der seriellen Schnittstelle des Spectrum – es gibt keine Möglichkeit, mit unterschiedlichen Baudraten zu senden und zu empfangen.

● **SERFL** (bei 23751) ist ein Flag-Byte für den Empfang. Wenn es auf Eins steht, ist ein weiteres Byte zum Lesen bereit.

● **SERBT** (bei 23752) ist ein Empfangsbuffer im Ein-Byte-Format. Gelegentlich wird hier auch nach Beendigung der Datenübertragung noch ein Byte empfangen und gespeichert.

Eine vierte Systemvariable – IOBORD (bei 23750) – ist nicht ausschließlich auf die serielle Schnittstelle ausgerichtet. Sie gibt die Farbe des Bildschirmrahmens während der E/A-Vorgänge an. Es ist also möglich, während der Ein- und Ausgabe die Hintergrundfarbe beliebig zu verändern.

Die Baudraten des Interface 1

Der RS232-Standard wird hauptsächlich für die Datenübertragung über größere Entfernungen eingesetzt. Die Baudrate gibt an, wieviel Datenbits pro Sekunde übermittelt werden. Die im Artikel angegebene Formel liefert bei den üblichen Baudraten folgende Ergebnisse:

Baudrate	Wert
50	2690
110	1221
300	446
600	222
1200	110
2400	54
4800	28
9600	12
19200	5

Der Wert der gewünschten Baudrate wird mit POKE in die Systemvariable BAUD (bei 23747 und 23748) gesetzt. Da dafür nur eine Variable zur Verfügung steht, gilt für Senden und Empfangen die gleiche Baudrate.

Der folgende Code zeigt, wie die Baudrate von 300 Baud gesetzt wird:

```
LD HL,446
LD (23747),HL
RET
```

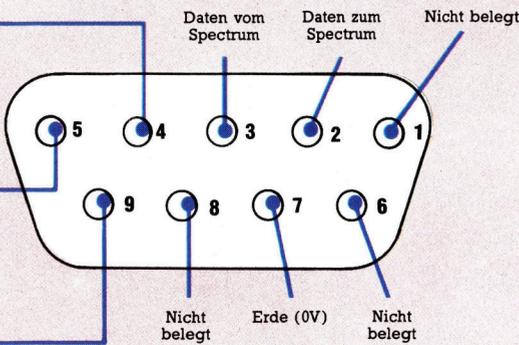


Die serielle Schnittstelle RS232

DTR (sollte vom angeschlossenen Computer auf logisch Eins geschaltet werden)

CTS (Sollte vom angeschlossenen Computer auf logisch Eins geschaltet werden, wenn er bereit ist, Daten zu empfangen.)

Zeigt mit logisch Eins an, daß der Spectrum „On-Line“ ist



Über den seriellen Port des Interface 1 können Daten mit anderen seriellen Geräten ausgetauscht werden. Die Schnittstelle arbeitet mit einer standardmäßigen neunpoligen Buchse.

Für den Einsatz der RS232-Schnittstelle wird zunächst mit dem Befehl `FORMAT` die Variable `BAUD` auf einen Wert gesetzt, der sich folgendermaßen errechnet:

$$\text{Wert} = (3500000/26 * \text{Baudrate}) - 2$$

Mit dieser Gleichung lassen sich auch nicht standardmäßige Baudraten errechnen, die vom Maschinencode aus eingestellt werden. Außer für die Kommunikation mit einem anderen Spectrum ist das jedoch kaum von Nutzen. Unsere vorseitige Tabelle zeigt Baudraten und die Routine, mit der sie gesetzt werden.

Wie werden nun die Daten über die Leitungen gesandt? In BASIC erzeugt das Eröffnen eines Stroms (`OPEN`) für die RS232-Schnittstelle einen Kanal mit einer bestimmten Struktur (siehe Bild). Wenn der Kanal indirekt angelegt wurde (z. B. mit `SAVE*`, `MOVE`, `FORMAT` etc.), wird die Zahl 128 zu dem Inhalt von Kanalbyte 4 addiert.

Wenn Sie die RS232-Routinen vom Maschinencode aus einsetzen, brauchen Sie darauf nicht zu achten. Beachten Sie, daß derartigen Kanälen kein Buffer zugeordnet ist. Die Schreib- und Leseroutinen werden über einen Hakencode aufgerufen, der die Adresse in Kanalbyte 5 und 6 (Senden) oder 7 und 8 (Empfangen) anspricht. Hier die beiden RS232-Hakencodes:

● **Hakencode 29** stellt – falls vorhanden – ein Byte der seriellen Schnittstelle in das A-Register. Das Übertragsflag wird in diesem Fall auf Eins gesetzt. Ist kein Byte vorhanden, enthält A den Wert Null, und das Übertragsflag wird entsprechend gelöscht.

● **Hakencode 30** sendet das im Register A gespeicherte Byte zur Datenübertragung an die serielle Schnittstelle. Beim Einsatz dieses Codes gibt es auf Maschincodeebene keinen Unterschied zwischen Strom B und T, da alle Bytes unverändert übertragen werden.

An dieser Stelle können jedoch Probleme mit `SERBYT` und `SERFL` auftreten. Wenn die Hakencoderoutine feststellt, daß `SERFL` auf Eins steht, holt sie ein Byte von `SERBYT`, selbst wenn die serielle Kommunikation bereits

beendet wurde und das Byte daher nicht mehr gültig ist. Dieses „Fehlerbyte“ muß abgefangen werden, da es die Bedeutung der darauffolgenden Bytes verändert, sobald es sich am Anfang von Empfangsdaten befindet. Wenn Sie `SERFL` vor jedem Lesevorgang (aus der seriellen Schnittstelle) auf Null setzen, wird der Inhalt von `SERBYT` ignoriert. Hier der entsprechende Code:

```
XOR A
LD (23751),A
```

Hier liegt eigentlich die einzige Schwierigkeit, die beim seriellen Interface auftreten kann. Wenn Sie überhaupt keine Daten erhalten, sollten Sie die Leitungen im Stecker vertauschen, die zu Kontakt 2 und 3 führen – oft liegt hier der Fehler. Die erzeugte Baudrate ist zwar nicht immer exakt, liegt aber im Rahmen des RS232-Standards.

Der folgende Code überträgt 255 Bytes an die RS232-Schnittstelle:

```

;send 255 bytes of data down serial interface
;
CF          rst #8           ;set up...
31          defb 49         ;...interface 1 variables
216E00     ld hl,110       ;value for 1200 Baud
22C35C     ld (23747),hl   ;set baud rate
06FF      ld b,255        ;set counter
C5          loop: push bc   ;save counter
3E00      ld -a,DATA      ;user supplies routine..
CF          rst #8         ;...to get data in a reg
1E          defb 30       ;send the byte in a
C1          pop bc        ;restore counter
10FB      djnz loop
C9          ret

```

In vielerlei Hinsicht ist die RS232-Verbindung die einfachste Funktion des Interface 1. Zwei Routinen für Empfangen und Senden erledigen die Arbeit, ohne daß der Maschincode Kanalinformationen steuern muß.



Die Struktur des RS232-Kanals

Beachten Sie, daß diesem Kanal kein Buffer zugeordnet ist. Die Daten werden mit den Hakencodes 29 und 30 über das A-Register geleitet.



Buntes Treiben

Das Unternehmen Prism beteiligte sich einst an der Entwicklung von „Micronet“ und vertrieb Computer der Marken Sinclair, Oric und Wren sowie Roboter für den Heimgebrauch. Doch die erfolgversprechende Zukunft der Firma fand ein jähes Ende.

Die meisten Firmen auf dem Heimcomputermarkt arbeiten nicht mit langfristiger Perspektive – sie haben genug damit zu tun, die schnell wechselnden Forderungen des Marktes nach Hard- oder Software zu erfüllen. Anders das Unternehmen Prism: Es hatte nicht nur entscheidenden Anteil an der Entwicklung von Micronet, der ersten englischen Datenbank, die auch Heimcomputer-Anwendern offensteht, sondern bewarb sich um Marktanteile beim Verkauf von preiswerten Robotern.

Das Unternehmen wurde 1982 für die Micronet-Entwicklung gegründet. Leiter wurden Richard Heath und Bob Denton. Micronet ermöglicht den englischen Heimcomputer-Freunden den Austausch von Programmen, schafft Zugang zu Informationen und erlaubt die elektronische Verbreitung von Nachrichten. Prisms Mutterfirma, die Verlagsgesellschaft ECC, brachte bereits das Magazin „Sinclair User“ heraus, als man die Rechner dieser Marke noch ausschließlich im Postversand bestellen konnte. Das Blatt hatte großen Erfolg, obwohl der damalige Prism-Vertriebsdirektor Terry Cartwright anfangs sehr skeptisch war: „Ich dachte, der Sinclair wäre eine Eintagsfliege – aber auf der ersten Sinclair-Messe haben uns die Leute 8000 Bestellformulare innerhalb weniger Stunden aus den Händen gerissen.“

Sinclair entschloß sich daraufhin, den Rechner über den normalen Handel zu verkaufen und machte mit Prism einen Vertrag über den Vertrieb des ZX81 und des neu erschienenen Spectrum.

Im März 1983 wurde Micronet von Prism zusammen mit der British Telecom & Telemap

gestartet. Prism übernahm durch den Vertrieb verschiedener Modems (Hersteller O E Ltd und Thorn EMI) für die bekannteren Rechner-Typen die Hardware-Seite des Geschäftes. Als letztes Gerät dieser Reihe erschien auch ein Modem für den Commodore C 64.

Frühes Ende

Micronet verfügt heute bereits über ca. 10 000 Kunden, Prism verkaufte jedoch seinen Anteil an diesem Datenpool und beschränkte sich auf den Vertrieb von Computer-Hardware. Neben dem Spectrum bot das Unternehmen auch Oric- und Atmos-Rechner sowie den transportablen Bürocomputer Wren an. Nach Anfangsschwierigkeiten, die das Erscheinen des Rechners um viele Monate verzögerten, wird dieser von Thorn EMI hergestellte Computer in England inzwischen vertrieben.

Aber Prism wagte sich noch auf ein weiteres Feld vor – den Vertrieb von Heimrobotern. Das Interesse an diesem Bereich blieb jedoch hinter den Erwartungen zurück. Prism bot unter anderem „Topo“ an, einen aus den USA importierten Roboter.

Terry Cartwright sah in den Robotern einen Wachstumsbereich. „Roboter erwecken unglaublich starkes Interesse. Ich weiß zwar nicht, was die Leute mit ihnen machen werden, aber das wußte man 1976 bei den Apple-Computern auch nicht.“

Leider wurden all diese Zukunftspläne durch Liquiditätsschwierigkeiten und schließlich den Konkurs beendet, bevor sie Prism in Firmenrealität umsetzen konnte.



Unter der Leitung von Richard Heath wurde Prism als Tochter des britischen Verlags- und Medienunternehmens ECC-Publications gegründet.

Miete statt Kauf

Das Unternehmen plante die Vermietung dieser Modems. Den Zugriff auf Prestel und Micronet gab es als Jahresabonnement dazu.



Z80 im Betrieb

Der tragbare „Wren“ arbeitet mit einem Z80. Schon die Grundversion enthält zwei Diskettenlaufwerke und ein eingebautes Modem für die Verbindung mit Prestel.



Briefe mit Format

Es kommt vor, daß man gleichlautende Briefe an verschiedene Adressaten verschicken muß. Ein Heimcomputer und ein leistungsfähiger Drucker machen diese aufwendige Arbeit leicht – mit unserem Texteditor-Programm.

Viele gleichartige Briefe zu schreiben, zum Beispiel Bewerbungen, ist sehr ermüdend, wenn man diese Arbeit per Hand zu erledigen hat. Der Computer kann hier wesentliche Erleichterung schaffen – mit unserem Texteditor, den wir in diesem PROGRAMMIER-SERVICE für Commodore, Atari, Sinclair und Schneider vorstellen.

Das Programm dient dazu, einen Leitbrief zu erstellen, von dem beliebig viele Kopien gemacht werden können. Um aber überhaupt mit Texten richtig umzugehen, bedarf es einer brauchbaren Tastatur – wir haben deshalb auf ein Programm für den ZX 81 mit seinen Folientasten verzichtet.

Im Gegensatz zu einer „echten“ Textverarbeitung, für die man etliche Eingabestunden benötigen würde, kann man hier den fertigen Brief nicht auf dem Bildschirm sehen; aber das stört nicht, denn das

Programm vermeidet automatisch Worttrennungen am Zeilenende. Es setzt sogar Leerzeilen zwischen einzelne Absätze und ist überdies für eigene Ideen offen.

Bitte beachten Sie, daß für wichtige Briefe, etwa Bewerbungen, ein guter Drucker mit frischem Farbband unabdingbar ist!

Tippen Sie nun zuerst das Hauptprogramm ohne DATA-Befehle ein und speichern Sie es ebenfalls gleich ab, damit es für weitere Briefe zur Verfügung steht. Bevor ein Brief ausgedruckt werden kann, muß zunächst das Programm an den jeweiligen Drucker angepaßt werden: Dazu müssen Sie herausfinden, wieviele Zeichen pro Zeile Ihr Drucker akzeptiert. Diese Zahl ordnen Sie der Variablen TL zu. Die Variable LL ist die Zeilenlänge des Briefes. Sie sollten LL immer etwas kleiner als TL wählen, damit ausreichend Rand an beiden Seiten bleibt. Selbstverständlich können die Werte auch noch kurz vor dem Ausdrucken geändert werden, um die letzte Anpassung vorzunehmen.

Atari

Die Variablen TL und LL werden in Zeile 30 festgelegt. Dabei steht TL

für die Druckerbreite und LL für die Zeilenlänge.

Schneider

In Zeile 30 finden sich alle wichtigen Parameter. Druckerbreite (tl) und Textbreite (ll) können Sie hier gemäß den Erfordernissen einstellen. Wenn Sie den Brief auf dem Schirm sehen wollen, ändern Sie den Wert von s auf 0. Vor dem Drucken aber auf keinen Fall vergessen, wieder s = 8 zu setzen!

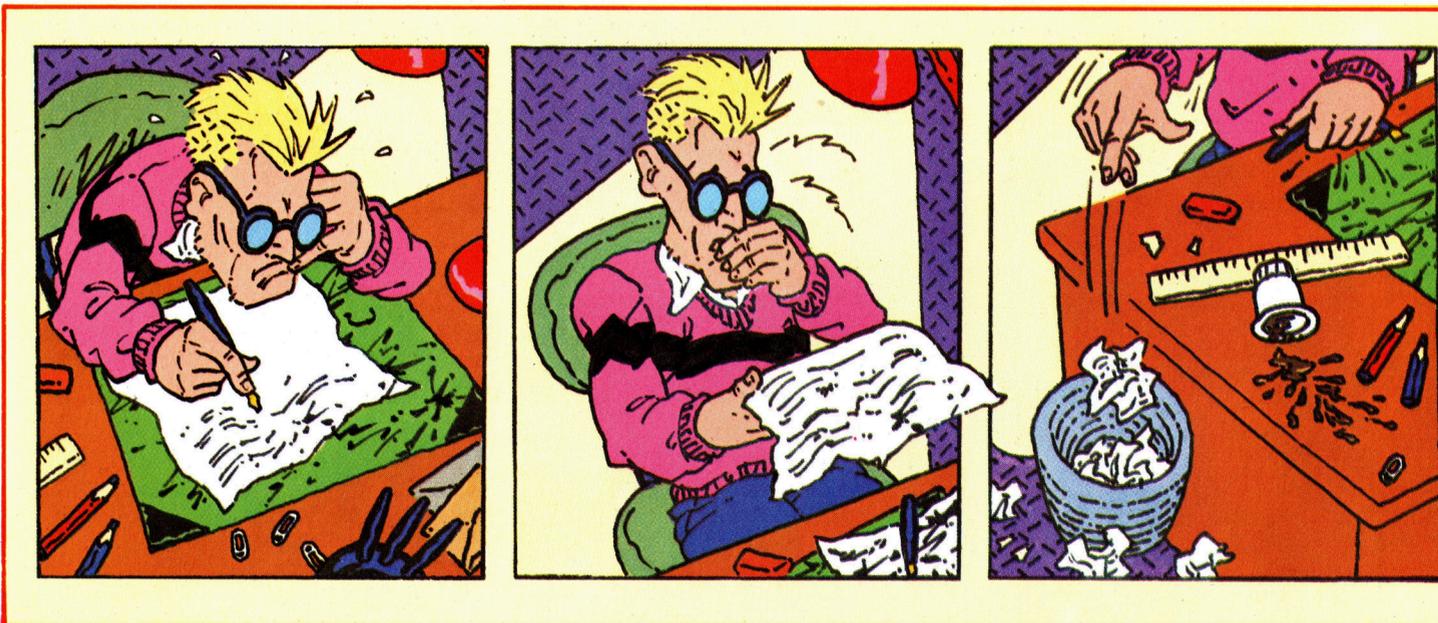
Spectrum

Die Werte für TL und LL sind für den ZX-Drucker eingestellt. Sie sind in Zeile 10 problemlos zu ändern.

Commodore

Dieses Programm ist auf einem Commodore-Drucker getestet worden. Die Anpassung an andere Drucker erfolgt durch die Änderung von TL und LL in Zeile 10.

Ihr Brief besteht aus einer Reihe von DATA-Befehlen. Geben Sie sie so ein, wie es unser Beispiel auf der letzten Seite des PROGRAMMIER-SERVICE zeigt. Beginnen Sie in Zeile 1000; am Anfang steht Ihre Adresse, wie zu Beginn eines jeden Briefes. Zuerst durchsucht der Com-



puter die Zeilen, die die Adresse enthalten, nach der längsten Zeile. Dann setzt er Ihre Adresse an den rechten Rand. Alle weiteren Zeilen werden dann an den linken Rand gedruckt – so sehen moderne Geschäftsbriefe normalerweise aus.

Beachten Sie aber, daß jede Zeile des Briefes (außer beim Atari) in Anführungszeichen stehen muß! Hier nun eine Liste mit den Bedeutungen der übrigen Symbole:

- Das Doppelkreuz (#) bedeutet: Diese Zeile ist ein Teil meiner Adresse; setze sie an den rechten Rand des Briefes.
- Das Dollarzeichen (\$) bedeutet: Drucke eine Leerzeile und starte dann einen neuen Absatz.
- Das kaufmännische Und (&) meint: Beginne einen neuen Absatz, aber diesmal ohne Leerzeilen davor. So was kann man zum Beispiel für die Anschrift gebrauchen.
- Das Multiplikationszeichen (*) verlangt: Zentriere diese Zeile.

Wenn Sie zusätzliche Leerzeilen benötigen, etwa zwischen „Hochachtungsvoll“ und Ihrer Unterschrift, müssen Sie nur einige weitere Dollarzeichen einfügen. Sie sollten einzeln in Anführungszeichen stehen und durch Kommata getrennt sein, damit der Rechner ausreichend instruiert ist.

Der Atari läßt sich allerdings nicht durch Anführungszeichen „überlisten“ und wertet Kommas in Texten

als Trennung zwischen zwei DATA-Werten. Um nicht auf Kommas im Text verzichten zu müssen, hilft folgender Trick: Wenn ein Komma im Text vorkommt, drücken Sie gleichzeitig auf CONTROL und Komma. Auf dem Bildschirm erscheint dann ein Herz, und das Druckerprogramm wandelt in Zeile 140 dieses Herz später wieder in ein Komma zurück. Die DATA-Zeilen sehen dadurch vielleicht etwas merkwürdig aus, doch kann man so auf die Anführungszeichen verzichten.

Spaces beim Atari

Ähnliches geschieht beim Atari mit den Leerfeldern – der Rechner nimmt sie nicht an. Der Zeilen-Editor „räumt“ eine eingegebene BASIC-Zeile auf und fügt dort Leerzeichen ein, wo welche ausgelassen wurden, oder er löscht gar überflüssige Leerfelder. Doch auch hier hilft ein Trick: Wenn zu Anfang einer Zeile Leerfelder stehen sollen, muß als erstes Zeichen CONTROL-A eingegeben werden. Das Grafikzeichen kann der Drucker nicht ausgeben, aber die Leerfelder werden vom Atari als Teil der DATA-Zeile anerkannt.

Da alles in BASIC geschrieben ist, können Sie die Briefe – oder die Entwürfe – einfach als Programm abspeichern. In diesem Fall besteht der erste Schritt zur Korrektur eines Briefes darin, das Programm mit den

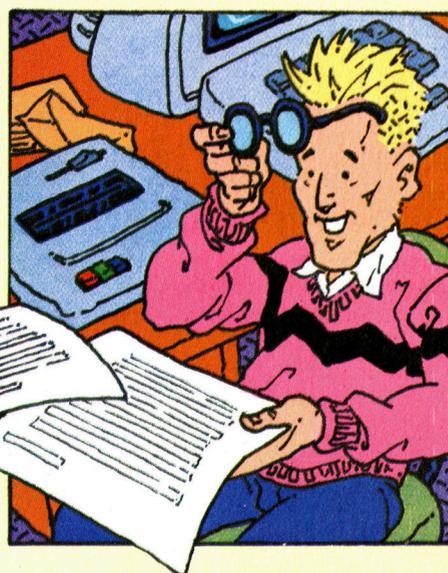
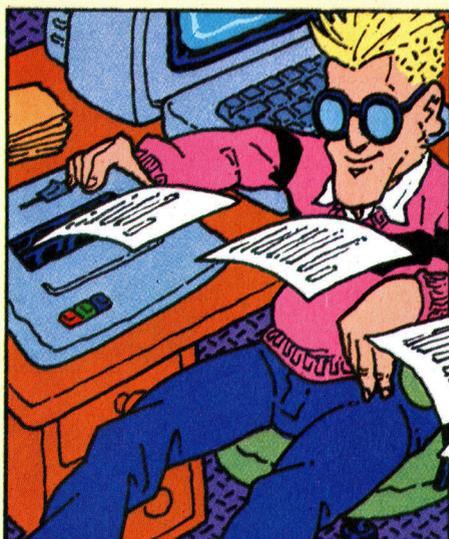
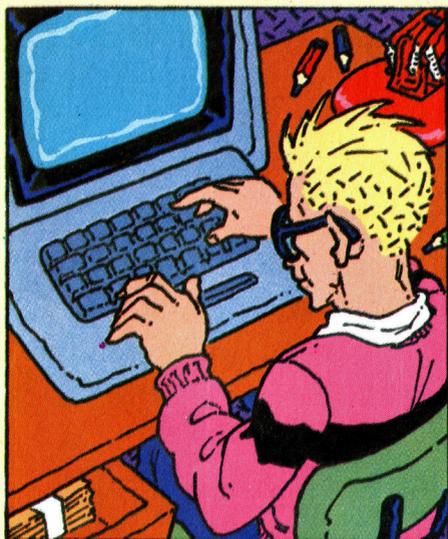
Daten erneut zu laden. Dann kann der Brief in der gewohnten Weise geändert werden.

Falls Sie vergessen haben, den ersten Brief abzuspeichern, können Sie trotzdem Kopien produzieren – allerdings müssen dann zuerst alle DATA-Befehle eingegeben werden, statt der Korrekturen.

Um den fertigen Brief auszudrucken, muß nur das Programm gestartet werden. Die Zeilen werden dann automatisch an den Drucker geschickt. Sollten Sie vergessen haben, den Drucker anzuschließen, geschieht folgendes: Der Anfang des Briefes erscheint auf dem Schirm; dann wartet der Rechner so lange, bis gedruckt wird, was allerdings nicht funktionieren kann. Damit der Computer das Gewünschte ausführt, muß das Programm unterbrochen und der Drucker an das System angeschlossen werden.

Die Papiersorte, die verwendet werden kann, hängt von der Art des Druckers ab. Bei einigen kann man zwischen Endlospapier und einzelnen Blättern wählen, wobei lose Blätter für Korrespondenz besser geeignet sind. Wenn Sie aber Endlospapier verwenden, sollten Sie die Seiten mit einer Schere nachschneiden.

Wenn Sie Papier mit eigenem Briefkopf haben, lassen Sie Ihre Adresse natürlich nicht nochmals ausdrucken. Dennoch müssen die DATA-Statements in Zeile 1000 mit



dem Adressaten beginnen.

Um den Brief zu beenden, benötigt der Commodore das Wort „Ende“ in einer letzten, eigenen DATA-Zeile.

Atari

```

10 DIM A$(100),B$(100),C$(100),D$(100)
20 B$=" ":B$(100)=B$:B$(2)=B$
30 LL=70:TL=80
40 LF=INT((TL-LL)/2)
90 GOSUB 410
100 IF ASC(A$)=36 THEN GOSUB
    400:LPRINT:A$=A$(2)
110 IF ASC(A$)=38 THEN GOSUB
    400:A$=A$(2)
120 IF ASC(A$)=42 THEN 200
130 IF ASC(A$)=35 THEN 300
140 FOR I=1 TO LEN(A$):C=ASC(A$(I)):IF
    C=0 THEN C=44
150 C$(LEN(C$)+1)=CHR$(C):IF C=32
    THEN 180
160 IF LEN(C$)>LL THEN PRINT
    CHR$(253);"FORMATFEHLER - WORT ZU
    LANG!":END
170 NEXT I:GOTO 90
180 IF LEN(D$)+LEN(C$)<LL THEN D$
    (LEN(D$)+1)=C$:C$="":GOTO 170
190 GOSUB 400:D$=C$:C$="":GOTO 170
200 L=LEN(A$)-1:IF L>LL THEN PRINT
    CHR$(253);"FORMATFEHLER - NICHT
    ZENTRIERBAR!":END
210 SL=INT((TL-L)/2)
220 LPRINT B$(1,SL);A$:GOTO 90
300 SL=LEN(A$)-1:RESTORE
310 GOSUB 410:IF ASC(A$)<>35 THEN
    RESTORE :GOTO 350
320 L=LEN(A$)-1:IF L>SL THEN SL=L
330 IF L>LL THEN C$(100)="":GOTO 160
340 GOTO 310
350 GOSUB 410:IF ASC(A$)<>35 THEN 90
360 LPRINT B$(1,LL-L);A$(2):GOTO 350
400 LPRINT B$(1,LF);D$:D$="":RETURN
410 TRAP 450:READ A$:A$(LEN(A$)+1)=
    "":RETURN
450 IF PEEK(195)=6 THEN GOSUB 400:END
460 EL=PEEK(186)+256*PEEK(187):PRINT
    "FEHLER - ";PEEK(195);" IN ZEILE";EL-
    END

```

Spectrum

```

10 LET LL=32:LET TL=32
15 LET LL=LL+1:LET T=(TL-LL)/2
20 LET D=0
30 READ A$:LET L=LEN A$
40 LET C=0
50 IF C=L THEN GOTO 30
60 LET C=C+1:LET D=D+1:IF C>1
    THEN GOTO 100
70 IF A$(C)="#" THEN GOTO 500

```

```

80 IF A$(C)="*" THEN GOTO 700
85 IF A$(C)="&" THEN GOTO 850
90 IF A$(C)="$" THEN LPRINT
    CHR$ 13;CHR$ 13;LET D=0:
    GOTO 900
95 LET A$="□"+A$:LET L=L+1
100 IF A$(C)="□" THEN GOTO 800
110 LPRINT A$(C);
115 IF D>LL THEN LET D=0
120 GOTO 50
500 LET NL=0:LET TA=LL:LET BE=0
510 LET LE=LEN A$-1:IF LE>LL THEN
    PRINT FLASH 1;"FORMATFEHLER -
    ADRESSE ZU LANG":STOP
520 IF LE>BE THEN LET BE=LE

```

```

530 LET NL=NL+1:READ A$:
    IF A$(1)="#" THEN GOTO 510
540 RESTORE 1000
550 LET TR=T+LL-BE:FOR G=1 TO NL:
    FOR H=1 TO TR:LPRINT "□";NEXT H:
    READ A$:LPRINT A$(2 TO ):NEXT G
560 GOTO 30
700 LET TA=(LL-L)/2+T:IF TA<T
    THEN LPRINT CHR$ 13:
    PRINT FLASH 1;"FORMATFEHLER -
    WORT ZU LANG":STOP
710 LPRINT CHR$ 13;FOR N=1 TO
    TA:LPRINT "□";NEXT N:LPRINT
    A$(2 TO L):GOTO 20
800 LET SL=LL-D-1:LET CC=C+1:

```





```

LET X=1
810 IF CC=L THEN GOTO 825
820 IF A$(CC) <> "□" THEN LET
  CC=CC+1: LET X=X+1: GOTO 810
825 IF X>=LL THEN LPRINT CHR$ 13:
  PRINT FLASH 1;"FORMATFEHLER –
  NICHT ZENTRIERBAR":STOP
830 IF SL>=X THEN GOTO 110
850 LPRINT CHR$ 13: LET D=0
900 FOR B=1 TO T: LPRINT "□";
  NEXT B: GOTO 50
  
```

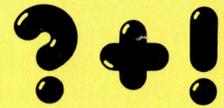
Schneider

```

20 MODE 2
30 II=60: tl=80: s=8
  
```

```

40 c$=""
50 II=II+1
60 d$=STRING$((t1-II)/2,"□")
80 PRINT #s,CHR$(13);d$;
90 READ a$
100 a$=a$+"□"
110 IF ASC(a$)=35 THEN
  GOSUB 330 'adresse
120 IF ASC(a$)=36 THEN PRINT #s:
  PRINT #s: PRINT #s,d$;
  a$=RIGHT$(a$,LEN(a$)-1): lp=0
130 IF ASC(a$)=38 THEN PRINT #s: PRINT
  #s,d$;: a$=RIGHT$(a$,LEN(a$)-1):
  lp=0
140 IF ASC(a$)=42 THEN GOSUB 270:
  GOTO 90 'zentrieren
150 c$=""
160 FOR t=1 TO LEN(a$)
170 b$=MID$(a$,t,1)
180 c$=c$+b$
190 IF LEN(c$)>II THEN PRINT: PRINT
  "FORMAT FEHLER – WORT ZU LANG":
  END
200 lp=lp+1
210 IF lp<=II THEN 240
220 PRINT #s: PRINT #s,d$;:
  IF ASC(c$)=32 THEN
  c$=RIGHT$(c$,LEN(c$)-1)
230 lp=LEN(c$)
240 IF b$="□" AND lp<=II THEN
  PRINT #s, c$;: c$=""
250 NEXT
260 GOTO 90
270 ' zentrieren
280 a$=RIGHT$(a$,LEN(a$)-1)
290 IF LEN(a$)>II THEN PRINT: PRINT
  "ZEILE ZU LANG – KANN NICHT
  ZENTRIERT WERDEN": END
300 PRINT #s: PRINT #s,d$;
  STRING$((II-LEN(a$))/2,"□");a$;
310 lp=0
320 RETURN
330 ' adresse
340 x=LEN(a$)
350 RESTORE
360 READ a$: IF ASC(a$) <> 35 THEN 390
370 IF LEN(a$)>x THEN x=LEN(a$)
380 GOTO 360
390 IF x>II THEN PRINT: PRINT "FORMAT
  FEHLER – ADRESSE ZU LANG": END
400 e$=STRING$((II-x),"□")
410 RESTORE
420 READ a$
430 WHILE ASC(a$)=35
440 a$=RIGHT$(a$,LEN(a$)-1)
450 PRINT #s,e$a$: PRINT #s,d$;
460 READ a$
470 WEND
475 a$="□"
480 RETURN
  
```



Wie sollte eine formgerechte Bewerbung aussehen?

Gestaltung und Aussehen einer Bewerbung sind extrem wichtig, denn sie ist der erste Eindruck, den eine Firma von Ihnen bekommt. Ein unordentlicher oder fehlerhafter Brief hinterläßt einen schlechten Eindruck, gleichgültig, wie qualifiziert Sie sind. Die Stile verändern sich im Laufe der Zeit; hier sind die neuesten Regeln.

Ihre Adresse kommt oben rechts auf das Blatt. Jede Zeile der Adresse sollte direkt unter der anderen liegen, ohne sie zu verschieben, also linksbündig.

Das Datum steht unter der Adresse nach einer Leerzeile.

Danach kommen Name und Adresse der angesprochenen Person. Sie gehören an den linken Rand des Blattes, entweder auf gleicher Höhe mit oder unterhalb Ihrer Adresse. Die Anschrift wird ganz ausgeschrieben, denn sie dient als Notiz, an wen Sie geschrieben haben.

Falls die Arbeitsvermittlung eine Kennziffer mitgegeben hat, kommt diese hinter „Ihre Kennziffer“ über die Anschrift der Firma. Nach einer Leerzeile folgt hinter „Betr.“ das Anliegen, also z. B. „Bewerbung um einen Ausbildungsplatz als ...“.

Nun kommt der eigentliche Brief. Eine persönliche Anrede wirkt immer am besten. Kennen Sie den Namen jedoch nicht, wird die allgemeine Anrede „Sehr geehrte Damen und Herren“ gewählt. Der nun folgende Teil sollte eine kurze Begründung enthalten, warum Sie sich für diesen Beruf und bei dieser Firma bewerben. Sie sollten auch erwähnen, woher das Angebot stammt. Dies alles muß linksbündig stehen.

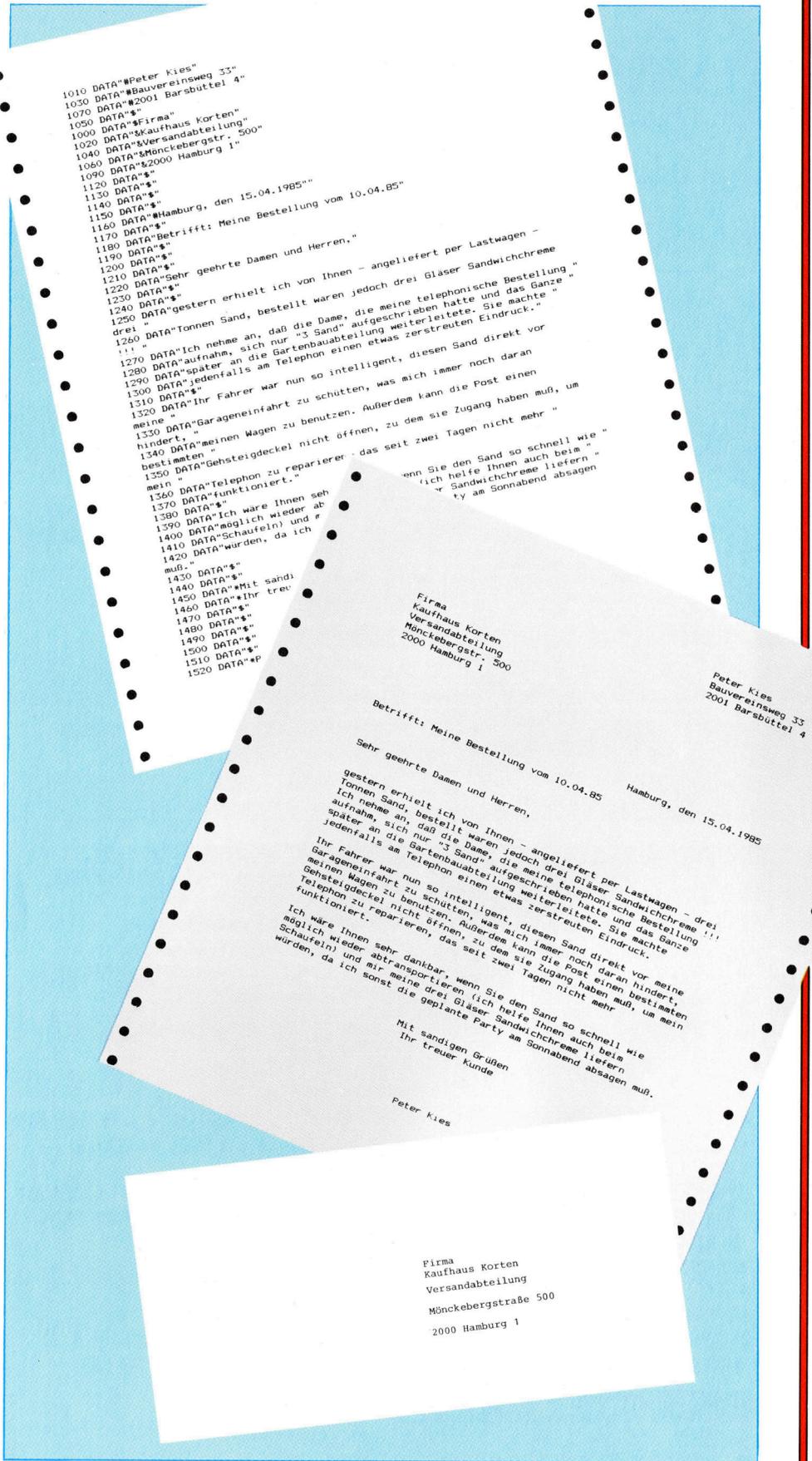
Sie beenden den Brief mit „Mit freundlichem Gruß“ und Ihrer Unterschrift. Beides sollte in der Zeilenmitte stehen.

So sieht ein – hier nicht ganz ernster – Brief aus: Einmal als Folge von DATA-Befehlen und einmal als fertiger Ausdruck.

Commodore

```

10 CLR:LL=60:TL=80:QQ=(TL-LL)/2:
  DIM A$(20):PRINT"☐" CR$=CHR$(13)
15 FOR Z=1 TO
20 OPEN 4,4:PRINT #4, CR$
100 READ K$:IF K$="ENDE☐" THEN
  PRINT #4,CR$:CLOSE 4:END
102 K$=K$+"☐":IF LEFT$(K$,1)="*" THEN LE=0:GOGO 600
115 IF LEFT$(K$,1)="&" THEN LE=0:PRINT #4,CR$:GOTO 400
120 IF LEFT$(K$,1)="$" THEN PRINT #4, CR$:LE=0:GOTO 400
130 GOTO 452
400 K$=RIGHT$(K$, LEN(K$)-1)
450 IF LEN(K$)=0 THEN 100
452 IF LE=0 THEN PRINT #4, LEFT$(K$,QQ);
455 FOR L=1 TO LEN(K$)
460 IF MID$(K$,L)="☐" THEN KK$=LEFT$(K$,L):K$=RIGHT$(K$, LEN(K$)-L):GOTO 500
470 NEXT:GOTO 100
499 REM*****WORTTRENnung*****
500 IF LEN(KK$)>LL+1 THEN 950
505 IF LE+LEN(KK$)>LL THEN PRINT #4,CR$+LEFT$(K$,QQ); LE=0
510 LE=LE+LEN(KK$):PRINT #4, KK$:GOTO 450
599 REM*****ZENTRIEREN*****
600 PRINT #4, CR$+LEFT$(K$,QQ):IF LEN(K$)>LL THEN 960
610 PRINT #4, LEFT$(K$,LL-(LEN(K$)+1/2)RIGHT$(K$, LEN(K$)-1)):GOTO 100
799 REM*****ADRESSE*****
800 READ X$:PRINT X$:P=P+1:IF LEFT$(X$,1)<>"#" THEN 900
810 A$(P)=RIGHT$(X$, LEN(X$)-1):IF LEN(A$(P))>HL THEN HL=LEN(A$(P))
815 IF HL>LL THEN 950
820 GOTO 800
900 IF P=1 THEN RESTORE:GOTO 100
910 P=P-1:FOR Z=1 TO P
920 PRINT #4, LEFT$(K$,QQ)+LEFT$(K$,LL-HL)+A$(Z):NEXT Z:K$=X$:GOTO 102
949 REM ***** FEHLERMELDUNGEN ***
950 PRINT #4,"☐":PRINT "☒FORMAT FEHLER-WORT ZU LANG":CLOSE 4:END
960 PRINT #4, CR$:PRINT "☒FORMATFEHLER-NICHT ZENTRIERBAR":CLOSE 4:END
  
```





Feine Zutaten

Bevor wir mit dem Zusammenbau unseres Digitalvoltmeters beginnen, wollen wir den 7135 A/D-Wandler-Chip etwas genauer studieren: Wir erläutern die Pinbelegung und die Fähigkeiten des Chips.

Wir beziehen uns zunächst auf den Schaltplan von Seite 2027, der die Pinbelegung und die Bezeichnung der Anschlüsse darstellt. Die Erklärungen erfolgen in der Reihenfolge der Pin-Nummern.

1. V-: Der 7135 braucht eine positive und eine negative Versorgungsspannung. Die positive Versorgungsspannung darf +6 Volt nicht überschreiten, auf der negativen Seite ist -9 Volt das Maximum. Zur Vereinfachung werden wir mit +5/-5V arbeiten. Die Stromaufnahme für die negative Versorgung beträgt nur circa 0,8 mA. Dadurch kann die Schaltung der Spannungsquelle sehr einfach aufgebaut sein. In unserem Beispiel verwenden wir ein Netzteil, es geht aber auch mit einer 6-Volt-Batterieversorgung (sowohl TTL- als auch CMOS-ICs arbeiten bei sechs Volt ohne Probleme).

2. Reference: Die Referenzspannung für den 7135 muß sehr stabil sein. Der Meßbereich eines Dual-Slope-Wandlers ist genau doppelt so groß wie der Wert der Referenzspannung – für einen Meßbereich von 2V brauchen wir also eine Referenzspannung von genau 1,000 Volt. Woher also nehmen?

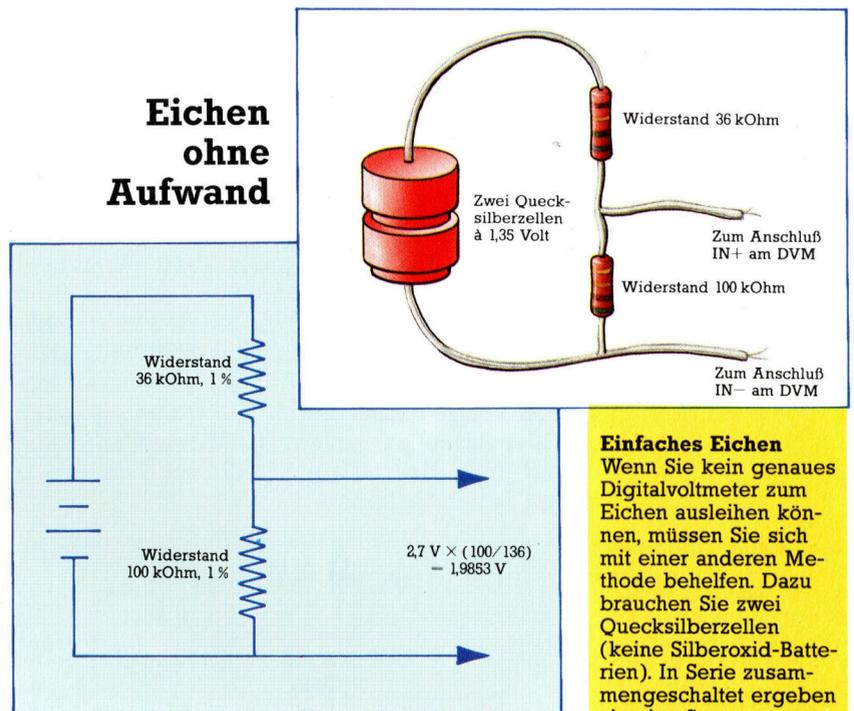
Präzision gefordert

Um eine 1-Volt-Referenzspannung zu erzeugen, gibt es mehrere Verfahren. Wir empfehlen die Verwendung einer Präzisions-Zenerdiode 1,22 Volt (Typ 9491). Die erforderlichen 1 Volt werden mit einem Potentiometer abgegriffen. Als Potentiometer sollte aber keinesfalls ein billiger Lautstärkereglern verwendet werden, sondern ein 10- oder mehrgängiges Trimmerpoti von möglichst hoher Qualität. Leihen Sie sich ein gutes Digitalvoltmeter (DVM), und stellen Sie damit am Trimmer die 1 Volt genau ein. Noch besser wäre es, am DVM mit einem Spannungsteiler die Anzeige auf 1,9999 Volt zu bringen und diese Spannung dann an das selbstgebaute Voltmeter anzuschließen: Danach wird auch hier der Anzeigewert ganz exakt auf 1,9999 Volt eingestellt.

Wenn Sie keinen Zugang zu einem genauen Digitalvoltmeter haben, können Sie die Eichspannung auch aus zwei in Serie geschalteten Quecksilberzellen mit einem Spannungsteiler erzeugen. Den Spannungsteiler bauen Sie mit einem 36 kOhm- und einem 100 kOhm-Widerstand auf. Sie müssen dazu möglichst eng tolerierte Widerstände (ein Prozent oder

weniger) und selbstverständlich nagelneue Quecksilberzellen verwenden.

3. Analogue Ground: Dieser unschuldig wirkende Pin kann mehr Schwierigkeiten hervorrufen, als es auf den ersten Blick scheint. Das liegt daran, daß es zwei unterschiedliche Masseleitungen gibt: die Analog-Masse und die Digital-Masse (Pin 24). Die analogen Signale,



Einfaches Eichen
Wenn Sie kein genaues Digitalvoltmeter zum Eichen ausleihen können, müssen Sie sich mit einer anderen Methode behelfen. Dazu brauchen Sie zwei Quecksilberzellen (keine Silberoxid-Batterien). In Serie zusammengeschaltet ergeben sie eine Spannung von 2,7 Volt, die mit dem Spannungsteiler aus den beiden Widerständen auf 1,9853 Volt reduziert wird. Die Abweichung der Spannung liegt bei dieser Anordnung im Bereich von 1–2 Prozent, sofern Metallschicht-Widerstände verwendet werden. Sie sollten aber nicht versuchen, den 36 kOhm-Widerstand aus mehreren Einzelbauteilen zusammenzulöten – Sie würden die Meßgenauigkeit dabei verschlechtern. Mit dem Trimmerpoti stellen Sie nach dem Anschluß der Eichspannungsquelle die Anzeige exakt auf 1,9999 Volt.

+ und - Versorgungsspannung und die Digital-Leitungen müssen strikt voneinander getrennt sein. Wir brauchen dazu einen Schaltungsaufbau, der verhindert, daß digitale Signale in den Analogteil und umgekehrt gelangen, wobei der zweite Fall weniger gravierende Folgen hat.

Alle Punkte des Analogteils, die auf Massepotential liegen (die Verbindung zwischen der Referenzdiode und dem Poti-Anschluß mit Pin 3), und die Masseseite des an der Diode angeschlossenen Widerstandes (der mit dem Integrationskondensator und dem Auto-Nullstellungskondensator an Pin 4 und 5 verbunden ist) sollten an einem einzigen Lötunkt zusammentreffen. Auch die Anode der Referenzdiode und den 100 kOhm-Widerstand (der mit dem Pin Int.Out über eine Diode verbunden ist) sollten Sie so zusammenfassen.



Genauso wichtig ist es, die digitalen Massepunkte auf einer Lötstelle zu vereinigen. Im Schaltbild haben wir zwei verschiedene Symbole für die beiden Massepunkte verwendet: für die analoge und für die digitale Masse. Beide Massepunkte werden mit einem möglichst dicken Kupferdraht verbunden.

Wenn Sie hier Fehler machen, treten vielfältige Komplikationen auf: Die Anzeige im Display wird ungenau, das Display könnte mehrere verschiedene (falsche) Anzeigen erzeugen, das Voltmeter könnte das Messen aber auch ganz verweigern, ins Schwingen geraten oder gar beschädigt werden.

Die Stromversorgung des Taktgenerators, des Display-Treibers und des Displays selbst sollte durch den Einsatz von Elektrolyt-Kondensatoren von hoher Kapazität entkoppelt und durch zusätzliche keramische Scheibekondensatoren gegen hochfrequente Störungen geschützt werden. Genauso wichtig ist es, daß die analogen Schaltungsteile auf der Platine möglichst weit von den digitalen Signalleitungen (etwa CLOCK, STROBE und BUSY) entfernt sind.

Die Regel lautet, digitale und analoge Schaltungsteile physikalisch zu trennen und auf die Führung der Masseleitungen zu achten – Erdschleifen müssen um jeden Preis vermieden werden.

Weiter gilt, daß man möglichst keine unbenutzte digitale Signal-Leitung (STROBE, UNDER-RANGE, RUN/HOLD u. ä.) undefiniert lassen sollte. Besser ist es, die freigebliebenen Anschlüsse mit einem 4,7 KOhm-Widerstand auf High oder mit einem 440 Ohm-Widerstand auf Low zu halten.

4. Int. Out. (Integrator-Ausgang): Integrierende A/D-Wandler wie der 7135 beruhen auf der Annahme, daß der Spannungsanstieg über dem integrierenden Kondensator dem Stromfluß proportional ist. Der absolute Wert dieses Kondensators ist nicht so wichtig – zwischen 0,4 und 0,47 µF (Mikrofarad) geht alles bestens. Worauf es ankommt, sind möglichst geringe Verluste im Kondensator. Am besten eignen sich daher Polypropylen-Kondensatoren, es geht aber auch mit Polystyren-Kondensatoren oder Polycarbonat-Typen. Nur Elektrolyt-Kondensatoren funktionieren nicht.

5. A-Z IN: Der hier verwendete Kondensator darf von jeder Art und Güte sein. Nicht einmal sein Wert ist kritisch – je größer, desto besser. Aber auch hier keinen Elko einsetzen!

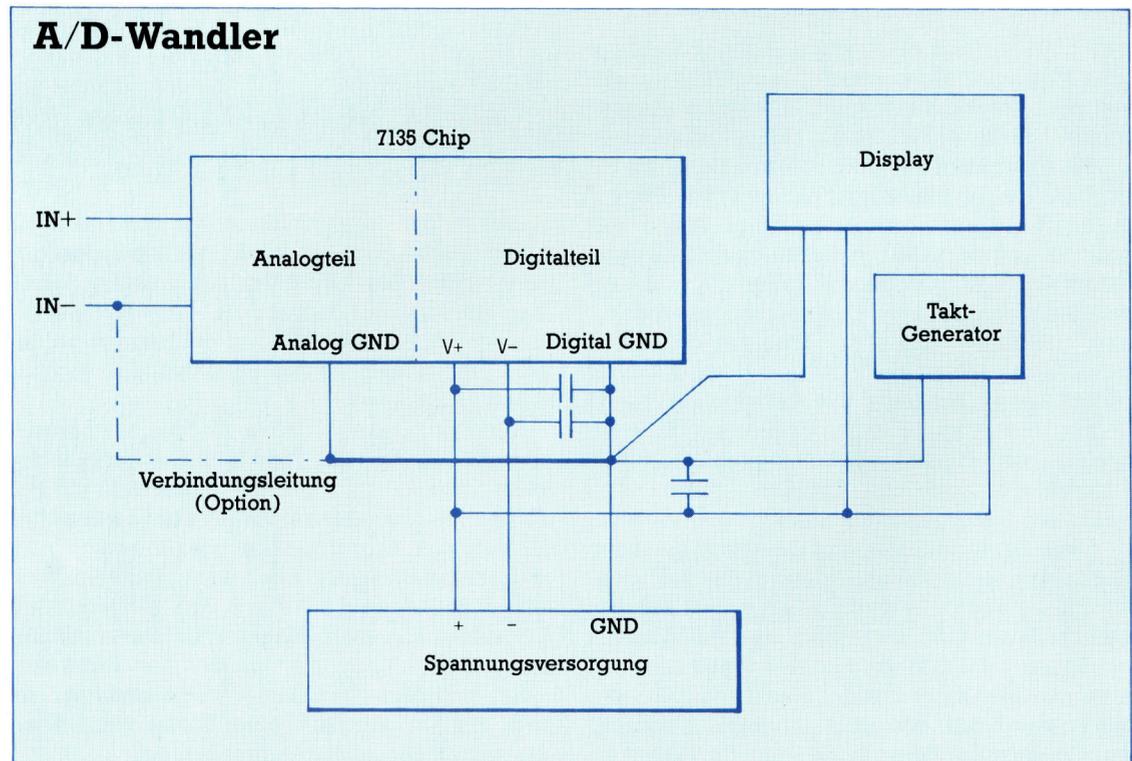
6. BUF OUT (Bufferausgang): An diesem Punkt wird der Integrator-Widerstand angeschlossen. Den Wert des Widerstandes ergibt:

$$R = \frac{\text{Spannung für Vollausschlag}}{20 \mu\text{A}}$$

Dafür kann jeder 100 KOhm-Widerstand verwendet werden. Ein Metallschichtwiderstand mit einer Abweichung unter einem Prozent eignet sich aber besser als ein billiger Kohleschicht-Widerstand.

Die Diode am Integrator-Ausgang dient zur Korrektur kleiner roll-over-Fehler. Hier kann jedes Bauteil zum Einsatz kommen. Bastler mit geringen Erfahrungen wünschen sich zwar meist eine Typenbezeichnung, es geht aber wirklich mit jeder verfügbaren Diode – mit Siliziumdioden allerdings besser als mit Germanium-Typen und mit Signaldioden besser als mit Gleichrichtern. Geeignete Dioden kann man tütenweise für wenig Geld bekommen.

Beim Aufbau der Schaltung muß sehr genau darauf geachtet werden, daß der Analogteil des Voltmeters sicher von den digitalen Elementen des Gerätes getrennt ist. Die einzelnen Segmente dürfen keinen Kontakt zueinander haben. Das abgebildete Schaltungsschema minimiert die Möglichkeit von Masseschleifen und einer eventuellen Rückkopplung von Digitalsignalen in den Analogteil.





Die Textmaus

Der Apple Macintosh bietet mit seiner mausgesteuerten Piktogramm- und Fenstertechnik ein reiches Angebot an grafischen Möglichkeiten, aber außerdem eine sehr wirkungsvolle Textverarbeitung. Auch bei ihr erfolgt die Befehlseingabe ausschließlich mit Hilfe der Maus, so daß die gewohnten Steuerzeichen entfallen.

Im Unterschied zu den Programmen in den voraufgegangenen Artikeln arbeitet das beim Macintosh mitgelieferte Textsystem „MacWrite“ nicht mit Textmenüs und Steuerzeichen – es macht statt dessen ausgiebig von Fenstermenüs, Piktogrammen und der Maus Gebrauch.

Nach dem Booten von MacWrite erscheint ein Begrüßungsschirm mit einem großen Feld für die Texteingabe. Darüber liegt ein Streifen mit Piktogrammen und eine Maßstabszeile, noch eins höher eine Überschriftleiste mit dem Namen des gerade bearbeiteten Schriftstücks, und ganz am oberen Bildrand stehen sechs Menütitel zur Auswahl.

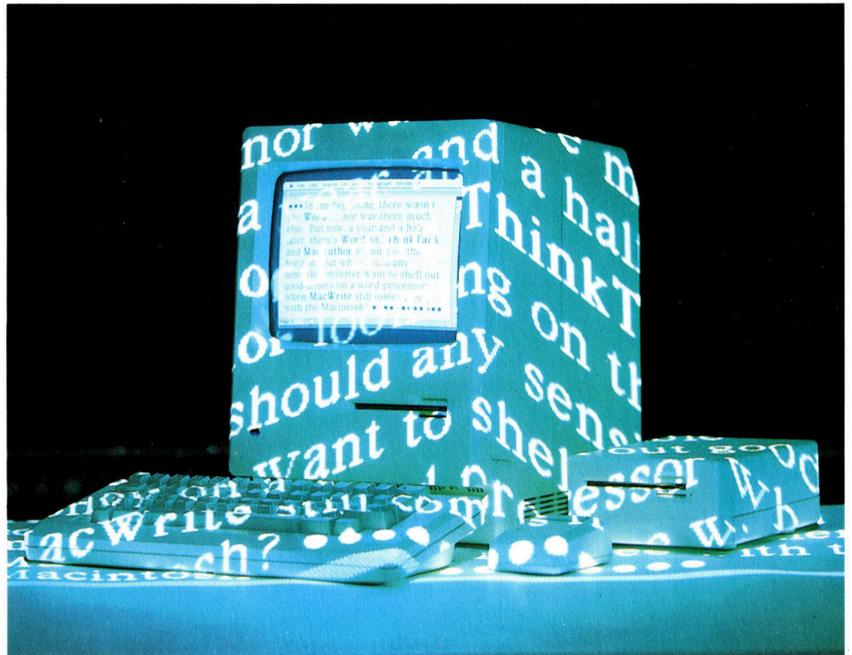
Anders als bei vergleichbaren Textsystemen wird bei MacWrite der Cursor nicht durch Steuertasten, sondern nur mit der Maus geführt. Und während der Cursor sonst nur die Funktion der „Schreibmarke“ hat und allenfalls noch zum Einfügen, Löschen oder Definieren von Blöcken dient, wird er bei MacWrite außerdem für die Softwaresteuerung eingesetzt, zum Beispiel für das Verschieben von Tabulatorstopps oder für die Menüeinblendung und die Auswahl der Funktionen.

Nicht alle Varianten, die im Menü zu sehen sind, lassen sich unmittelbar ansprechen. Die direkt verfügbaren erscheinen in schwarzer Schrift, die anderen dagegen in Grau. Die Menüs bieten Standardeinstellungen an; die gerade gültige ist durch ein Häkchen gekennzeichnet. Wenn Sie zu einer anderen Option übergehen, wandert die Markierung entsprechend und zeigt die neue Vorauswahl an.

Menüs à la minute

Das Datei-Menü enthält DOS-Kommandos zum Eröffnen und Schließen von Dateien wie auch zum Speichern auf Diskette. Hier finden Sie auch den Befehl „Drucken“.

Das zweite Menü auf der Kopfleiste heißt „Bearbeiten“ und umfaßt eine Reihe von Befehlen für das Editieren von Text. Für den Anfänger ist davon vielleicht der Undo (Widerruf-) Befehl am wichtigsten, der alles löscht, was seit der letzten Betätigung der Maus-Taste eingegeben wurde. Dieses Menü enthält auch die Blockbearbeitungs-Kommandos wie Verschieben, Duplizieren oder Löschen.

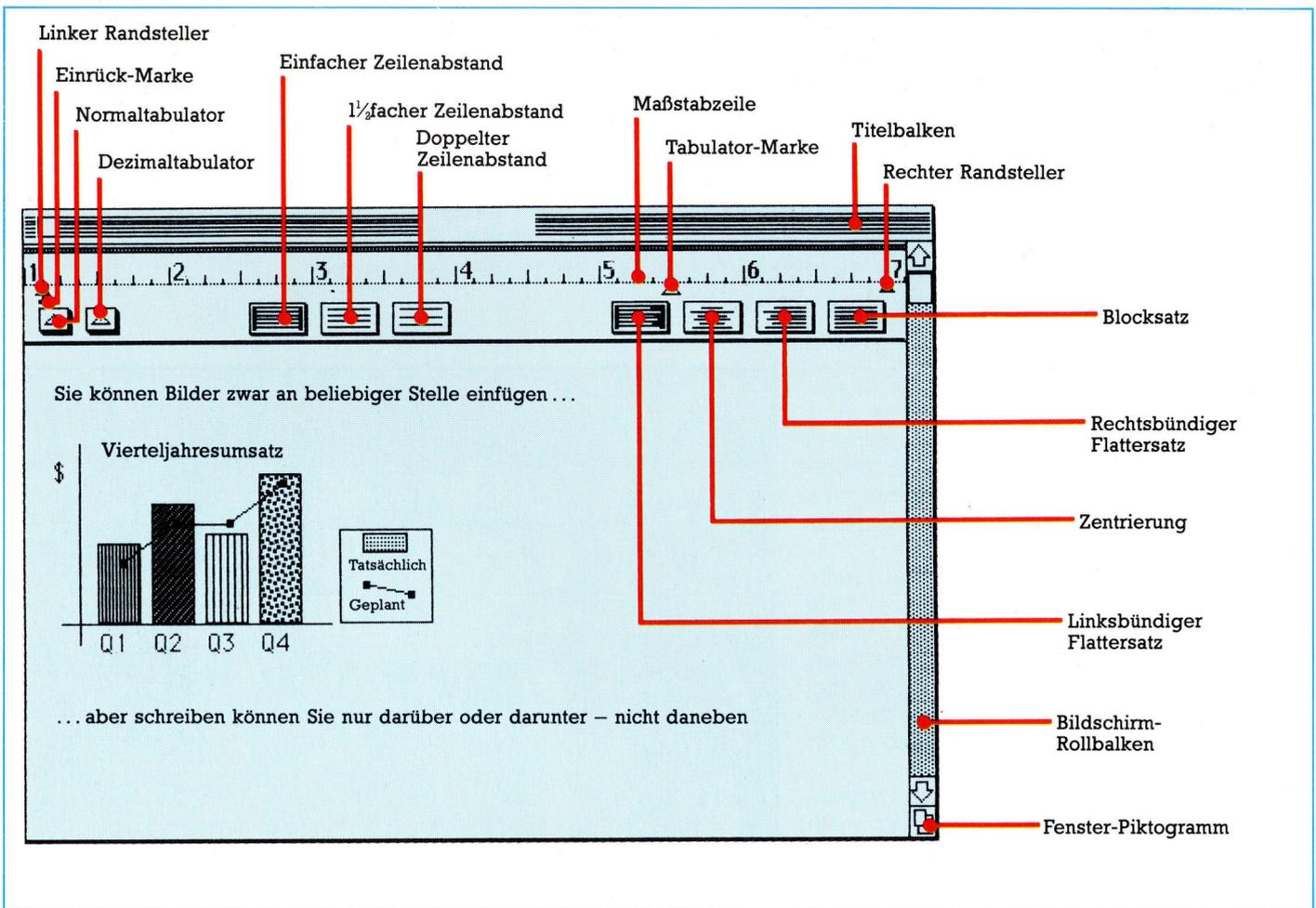


Wie bei WordStar können durch Abgrenzung von Passagen Blöcke definiert werden. Nachdem der Cursor mit der Maus auf Blockanfang bzw. -ende gesetzt und die Maus-Taste betätigt ist, verwandelt sich der Cursor in eine „Einfügungsmarke“, von deren Position aus eine Textergänzung erfolgen könnte. Wenn Sie den Cursor von dort aus nach oben oder unten verschieben, erscheint der entsprechende Textabschnitt in Negativschrift, und ein zweiter Druck auf die Maus-Taste legt dann die jeweils andere Blockgrenze fest.

Nach der Definition sind verschiedene Blockoperationen möglich. Zum Beispiel kann man ins Bearbeitungs-Menü zurückkehren und dort die „Clipboard“-Option wählen, die den Block zunächst in einen Zwischenspeicher stellt. In diesem Buffer wird der Block während der Bearbeitung abgelegt. Er läßt sich mit dem Befehl „Zwischenspeicher einblenden“ im Bearbeitungs-Menü jederzeit ausgeben. Nach Wahl des neuen Einfügungspunktes können Sie die Kopier- oder Einsetzfunktion aufrufen; mit der Übernahme des Blockes ist dann automatisch eine NeufORMATIERUNG verbunden.

Das „Search“-Menü enthält die Befehle zum Lokalisieren und Austauschen von Buchstabenkombinationen.

Apple stieß mit dem Anspruch, der Macintosh sei ein ernstzunehmender Bürorechner, zunächst auf einigen Widerstand. Zweifellos lag das wesentlich am Fehlen eines leistungsfähigen Textverarbeitungsprogramms, ohne das ein erfolgreicher Einsatz im Büro kaum denkbar ist. Inzwischen gibt es sehr leistungsfähige Pakete, wenn auch nicht in einer solchen Vielfalt wie für den IBM PC.



Wie alle Macintosh-Softwarepakete macht auch MacWrite bei der Steuerung ausgiebig Gebrauch von Piktogrammen, die am Rand des Bildschirms angeordnet sind. Viele davon - z. B. der Bildschirm-Rollbalken - sind mit denen der sonstigen Macintosh-Programme identisch, während andere eine Besonderheit von MacWrite darstellen.

Bei vielen Textverarbeitungsprogrammen können innerhalb eines Schriftstücks zur Hervorhebung nur Kursiv- und Fettdruck gewählt werden. Außerdem ist es im allgemeinen nicht möglich, den Text druckgetreu (mit Unterstreichungen und veränderten Schriftarten) auf dem Bildschirm darzustellen, weil das Textprogramm meist nur auf den Zeichensatz zurückgreift, der im System-ROM des Rechners steht.

MacWrite bietet dagegen eine ganze Reihe verschiedener Schriftarten, und mit jeder neuen Software-Version kommen weitere hinzu. Da der Macintosh-Bildschirm bei der Textwiedergabe im Bitmustermodus betrieben und nicht wie üblich zeichenweise angesprochen wird, können alle Typen direkt dargestellt werden. Trotzdem ist die Sache nicht perfekt: Detailreiche Schriften wie „London“ sind auf dem Bildschirm fast nicht lesbar.

Der Bitmustermodus verträgt sich besser mit dem „Stil“-Menü, das die gebräuchlicheren Hervorhebungsarten wie Unterstreichen und Fettdruck enthält. So etwas ist auf dem Bildschirm ohne weiteres wiederzugeben, um den Textindruck vor der Ausgabe auf Papier beurteilen zu können.

Die meisten Funktionen für das Seitenlayout sind im „Format“-Menü zu finden, die aktuelle Seitendimensionierung steht jedoch im „Datei“-Menü. Auf jede Seite können Standardtitel in Kopf- und Fußzeilen sowie Seiten-

zahl, Datum und Uhrzeit gesetzt werden. Die letztgenannten drei Angaben sind als Piktogramme aus ihrer Menüleiste „herauszugreifen“ und irgendwo über oder unter dem Text zu plazieren.

Direkt an der Maßstabzeile oben auf dem Schirm befinden sich ein paar kleine Pfeil-Piktogramme zum Setzen von Randstellern, Absatz-Einrückungen und Tabulatorstopps. Der Text wird bei jeder Änderung der Vorgaben automatisch neu ausgerichtet. Zwei Kästchen links unterhalb der Maßstabzeile dienen als „Tabulator-Quellen“, aus denen sich weitere Pfeilmarken hervorzaubern und auf die Skala setzen lassen. Rechts davon befindet sich eine Anzahl von Seiten-Piktogrammen, jedes mit einem Liniensystem zur Verdeutlichung der Funktion. Die ersten drei legen einfachen, anderthalbfachen und doppelten Zeilenabstand fest, die übrigen vier rechts außen sind für das Ausrichten des Satzes zuständig. Hier kann man zwischen links- und rechtsbündigem Textausdruck, Blocksatz und zentriertem Text ganz nach Belieben wählen. Ein wichtiger Punkt für die Textgestaltung.

Beim Macintosh können Text und Grafik gleichzeitig dargestellt werden. Auch hierbei leistet MacWrite Hilfe - der Entwurf der Bilder oder Zeichnungen erfolgt allerdings mit MacPaint bzw. MacDraw. Beide Programme sind Bestandteil des mitgelieferten Pakets. Der Zu-

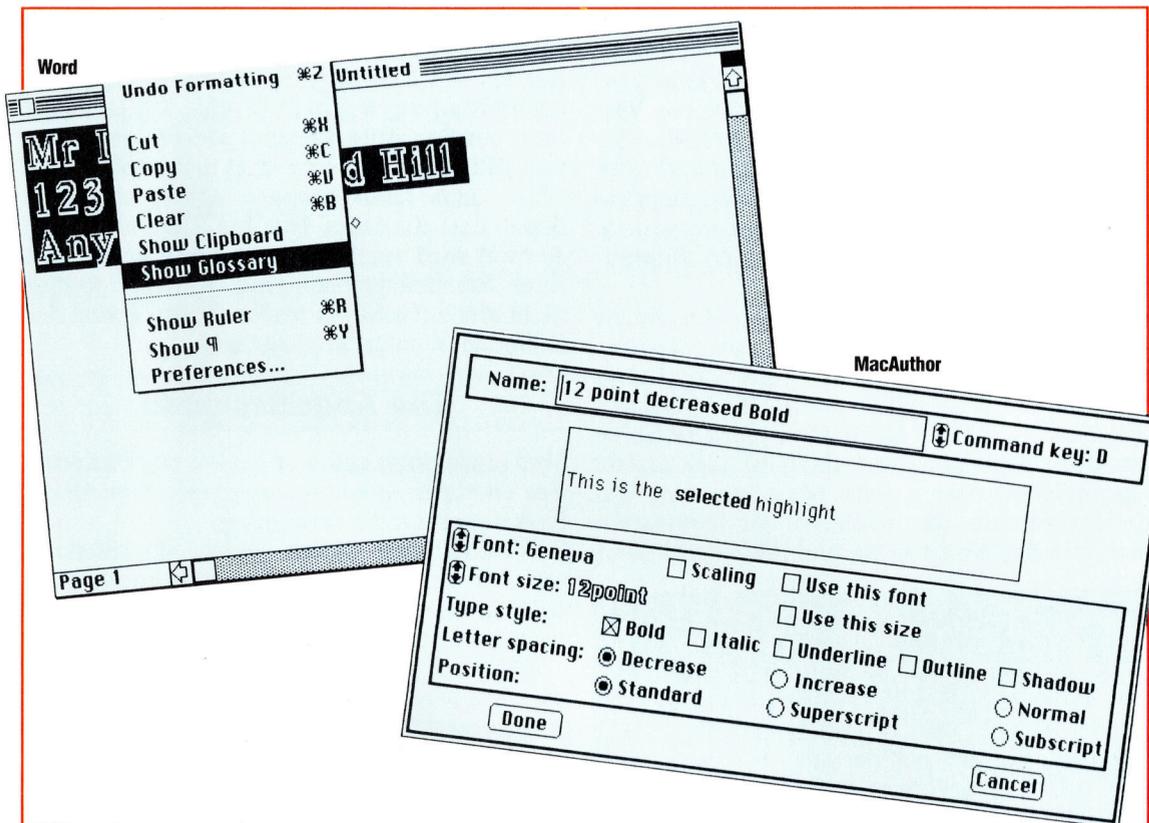


griff auf die Bilder kann dann über das beim Macintosh integrierte Schreibtisch-Organisationsprogramm erfolgen; dort ist ein Speicherbereich speziell für die Übertragung von Text und Grafik an beliebige Stellen eines Schriftstücks reserviert. Aufgrund der Flexibilität des Rechners können die Bilder dabei auf jede gewünschte Größe gebracht werden.

Trotz seiner bemerkenswerten Benutzerfreundlichkeit hat MacWrite auch ein paar Haken. Vor allem ist bei der Speicher-Grundausstattung im RAM nur noch wenig Raum für Text. Das ist angesichts einer Kapazität von 256 KByte zwar kaum vorstellbar, aber das Betriebssystem und der Bitmustermodus verschlingen arg viel Platz. Außerdem bleibt der Entwurf bei MacWrite vollständig im Arbeitsspeicher und wird nicht wie bei anderen modernen Textsystemen kontinuierlich auf Diskette ausgelagert.

MacWrite ist also in fast jeder Hinsicht ungewöhnlich und genauso eigenwillig wie der ganze Macintosh. Die Art und Weise, wie die Piktogramm- und Fenstertechnik mit der Maussteuerung hier auf einem Gebiet Anwendung findet, auf dem bislang nur mit der Tastatur gearbeitet wurde, erscheint dabei ebenso sinnreich wie raffiniert.

Darüber hinaus besticht die Textverarbeitung durch ihre Vielseitigkeit, die schon dann von großem Nutzen ist, wenn man etwa im Text nur einige Statistiken von einigermaßen anspruchsvollem Aussehen integrieren will. Ein Plus, das sich erst beim Einsatz von hochwertigen Druckern richtig bemerkbar macht, sind die vielen verfügbaren Schrifttypen, die auch den Ästheten unter den Computernutzer etwas bieten. So steht der MacWrite auf halbem Wege zu professionellem Textdesign etwas besser da als eine einfache Textverarbeitung.



Konkurrenten

MacWrite hat durchaus Konkurrenz. Die beiden wichtigsten Alternativen sind die Textprogramme „Word“ von Microsoft und „MacAuthor“ von Icon Technology.

Auf den ersten Blick erinnern die beiden Pakete mit ihren Piktogramm-Befehlen und dem Titel- sowie Rollmodus-Balken am Bildschirmrand stark an MacWrite. Sie bieten aber zusätzliche Editiermöglichkeiten, die dem Einsatz von Macintosh in der professionellen Textgestaltung wesentlich entgegenkommen.

Das Paket Word beinhaltet eine Reihe von Maßnahmen zur Erhöhung der Flexibilität und Leistungsfähigkeit, so daß bis zu vier Entwürfe

gleichzeitig am Bildschirm bearbeitet werden können.

Word ist als Textverarbeitungssystem fürs Büro gedacht; dagegen richtet sich MacAuthor vorwiegend an „schriftstellernde“ Textgestalter. Das Programm wurde entwickelt, um größtmögliche Flexibilität beim Gebrauch der Schrifttypen zu erreichen. Der Benutzer kann z. B. durch Kombination der vorhandenen Zeichensätze neue erzeugen und nach Wunsch das Aussehen des Textes verändern. Wegen dieser Freiheiten ist MacAuthor etwas schwerer in den Griff zu bekommen als Word oder MacWrite, doch viele professionelle Anwender werden feststellen, daß sich die Mühe lohnt.

MacWrite

ZEILENUMBRUCH

Zeilenumbruch und Ausrichten des rechten Rands erfolgen automatisch.

BLOCK-BEARBEITUNG

Text und Grafik können beliebig verschoben werden.

BILDSCHIRMHILFE

Nicht vorgesehen.

80-ZEICHEN-BILDSCHIRM

An sich erlaubt MacWrite nur die Darstellung von 60 Zeichen pro Zeile; bei der kleinsten Type (9 typografische Punkte) passen aber auch 80 Zeichen.

WORTZÄHLER

Nicht vorhanden.

SUCHEN UND ERSETZEN

Innerhalb des aktuellen Fensters lassen sich Kombinationen von max. 44 Zeichen lokalisieren.

DRUCKGETREUES TEXTBILD

Eine besondere Stärke von MacWrite liegt darin, daß auf dem Bildschirm die verschiedensten Schriftarten und auch Grafiken dargestellt werden können.

SERIENBRIEFE

Nicht vorgesehen.

RECHTSCHREIBKONTROLLE

Bei MacWrite nicht integriert; allerdings gibt es Programme aus anderer Quelle.

SCHRIFTARTEN

MacWrite bietet eine Fülle verschiedener Schriften.

DATEIKETTUNG

Obwohl einige Funktionen (z. B. Ausschneiden und Einfügen) dateiübergreifend eingesetzt werden, ist echte Dateikettung unmöglich.



Spectrum am Netz

Das Interface 1 des Spectrum wird hauptsächlich zur Steuerung der Microdrives oder als serielle Schnittstelle RS232 eingesetzt. Es kann Ihnen aber auch Zugang zu lokalen Netzwerken (LAN) verschaffen.

Wir beschrieben bereits, wie das Interface 1 die Microdrives und die serielle Schnittstelle steuert. In diesem Artikel wollen wir untersuchen, wie sich die Netzwerkfähigkeiten des Interface 1 einsetzen lassen. Die Arbeitsweise von Netzwerken wurde in einem früheren Teil des Kurses schon genauer erklärt. Im folgenden Text bezeichnet SELF unsere Maschine und IRIS die Maschine im Netzwerk, mit der wir kommunizieren wollen.

Programme und Daten lassen sich mit zwei verschiedenen Techniken über ein Netzwerk senden. Beim „Broadcasting“ kann jede Maschine des Netzwerks „mithören“. Dieses Verfahren verteilt die Daten schnell an viele Maschinen. Bei der „Exklusivübertragung“ gibt SELF an, daß nur IRIS die Information empfangen soll. Wenn IRIS keine Daten entgegennimmt, versucht SELF mehrfach eine Kommunikation herzustellen.

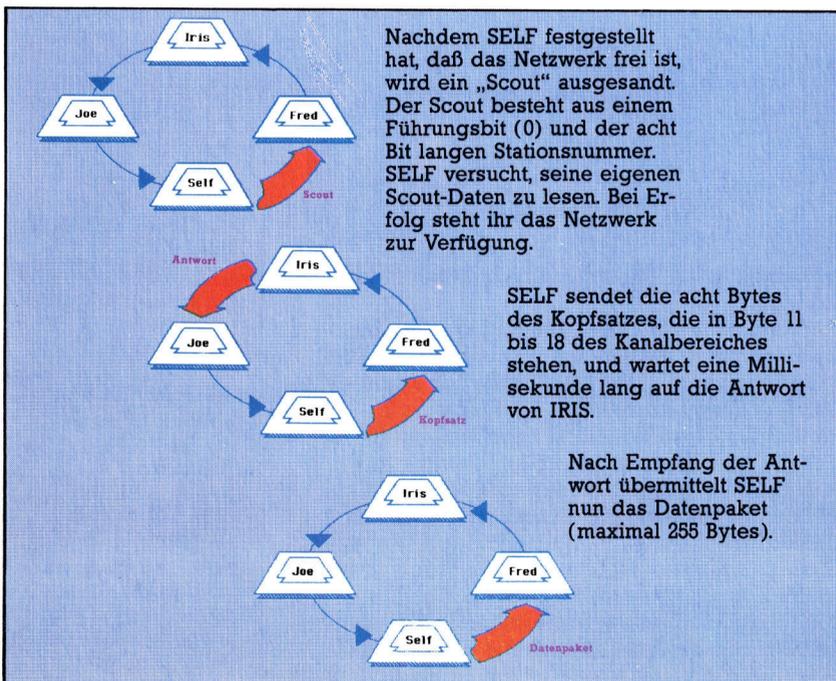
Auch die Abläufe, mit denen die Maschinen eines Netzwerks untereinander Daten austauschen, sind interessant. Ein Gerät, das Daten senden will, prüft zuerst, ob das Netzwerk frei ist. Damit nicht alle Maschinen gleichzeitig zu senden beginnen (und dadurch ihre Daten vermischen), wartet jede Maschine eine zufallsgesteuerte Zeit, nachdem sie festgestellt hat, daß das Netz frei ist, und nimmt erst dann

das Netz für sich in Anspruch. Wenn es zu diesem Zeitpunkt bereits wieder aktiv ist, muß das Gerät nochmals warten.

In das freie Netz sendet SELF zunächst einen „Scout“. Der Scout enthält die Stationsnummer der Maschine, zu der SELF Verbindung aufnehmen will. Ist sie damit erfolgreich, hat SELF das Netz „unter Kontrolle“.

Nun wird ein „Datenpaket“ gesandt, das aus einem Kopfsatz (mit Informationen über den darauffolgenden Datenblock) und 255 Datenbytes besteht. Der nächste Vorgang hängt von der Art der Übertragung ab. Bei einer Exklusivübertragung wartet SELF eine Millisekunde lang auf das Antwortsignal der angesprochenen IRIS. Trifft dieses Signal nicht ein, sendet SELF acht Millisekunden später nochmals Scout und Kopfsatz. Nur bei Empfang einer Antwort wird auch der Datenblock abgesandt. Eine Broadcastübertragung wartet dagegen nicht erst auf eine Antwort, sondern speist den Datenblock sofort ins Netz ein.

Ein Netzvorgang „belegt“ zuerst das Netz für die Sendestation und übermittelt dann die Daten (angeführt von einem Kopfsatz). Das Bild zeigt, wie in einem Netzwerk mit vier Geräten die Station SELF Daten exklusiv an die Station IRIS übermittelt.



Die Angriffspunkte

Vom Maschinencode aus stehen vier Hakencodes und zwei wichtige Systemvariablen für die Netzsteuerung zur Verfügung.

- **NTSTAT** (bei 23749) enthält die Stationsnummer der Maschine. Wenn der Befehl FORMAT in einem Netzwerk verwandt wird, setzt er diese Variable. Ihr Standardwert ist Eins.
- **DSTR1**: Diese Variable kennen Sie bereits von den Microdrives. Sie liegt bei 23766 und 23767 und enthält die Stationsnummer der IRIS, an die SELF Daten senden will. Wenn die Variable auf Null steht, werden alle Daten per Broadcasting übertragen.

Die Hakencodes für das Netzwerk funktionieren ähnlich wie die der Microdrives. Zunächst wird mit Hakencode 45 ein Kanal mit 276 Bytes angelegt (das Bild zeigt seine Struktur), der sich wie jeder andere Kanal im Speicherbereich für Kanaldaten befindet.

Der für das Netzwerk bestimmte Kopfsatz enthält die Bytes von NCIRIS bis einschließlich NCHCS. Der Datenblock reicht von Kanalbyte 21 bis 275. Die Kanalbytes 19 und 20 haben nur Bedeutung, wenn über den Kanal Daten aus dem Netzwerk gelesen werden. Wie beim Microdrivesystem läßt sich leicht in den Kanal schreiben oder daraus lesen. Dabei wird die Kanaladresse an CURCHL übergeben und



RST #10 oder CALL #15E6 aufgerufen. Bei einem Netzwerk löst man jedoch automatisch eine Übertragung von Kopfsatz und Daten aus, wenn versucht wird, das 256te Byte in den Datenbereich zu schreiben.

● **Hakencode 45** eröffnet einen Netzwerkkanal. Für seinen Einsatz wird NISTAT auf die Stationsnummer von SELF gesetzt und DSTR1 auf die von IRIS (beim Lesen werden die Daten von IRIS an SELF übermittelt). Sie sollten auch hier sicherstellen, daß die Systemvariablen des Interface 1 gesetzt sind. Der Hakencode wird auf die übliche Art eingesetzt. Beim Rücksprung enthält das IX-Register die Anfangsadresse des Kanals.

Wie bereits erwähnt, sendet RST #10 Bytes über das Netzwerk, und CALL #15E6 empfängt Daten. Dabei kann jedoch der Inhalt des IX-Registers zerstört werden. Sie sollten daher vor Aufruf dieser Routinen den Inhalt von IX sichern, da Sie ihn – oder zumindest die Kanaladresse – für das Schließen des Kanals unbedingt benötigen.

● **Hakencode 46** schließt einen Netzwerkkanal. Bei Aufruf der Routine muß sich die Anfangsadresse des Kanals im IX-Register befinden. Wenn der Kanal zum Schreiben eröffnet war, werden alle Daten, die sich noch im Datenbereich des Kanals befinden, in das Netzwerk übertragen. Nach Schließen des Kanals verschiebt sich der Speicher zwischen Kanäle und STKEND nach unten.

● **Hakencode 47** kann ein bestimmtes Paket aus dem Netzwerk empfangen. Die Kanalvariable NCNUMB muß dabei die gewünschte Blocknummer enthalten. Es wird vorausgesetzt, daß auch in NCSELF und NCIRIS die korrekten Werte gespeichert sind (die normalerweise gleich bei Eröffnung des Kanals gesetzt werden). Die Hakencoderoutine versucht nun, das Paket zu empfangen. Wenn ein „Time Out“ eingetreten ist, die Daten nicht sauber empfangen wurden, oder die Prüfsummen nicht übereinstimmen, steht das C-Flag nach dem Ablauf auf Eins. Anhand des Status der Flags läßt sich feststellen, ob Sie ein bestimmtes Paket nochmals aus dem Netzwerk anfordern müssen. Ist das Paket „unbeschädigt“ eingetroffen, wird – falls nötig – eine Bestätigung gesandt und NCNUMB inkrementiert.

● **Hakencode 48** übermittelt ein Datenpaket über das Netzwerk und ist damit die wichtigste Übertragungsroutine. Der Code arbeitet mit einem offenen Kanal, dessen Adresse per IX-Register übergeben wird. Register A gibt den Dateityp des im Kanal gespeicherten Datenblocks an. Wenn das Register A auf Eins steht, enthält der ausgesandte Block das Dateiende. Für alle normalen Datenblöcke ist der Registerwert Null.

Beim Öffnen eines Kanals bestimmt der Wert von DSTR1, ob die Daten per Broadcast oder per Exklusivübertragung gesandt werden. Wenn die Empfangstation die Daten an-

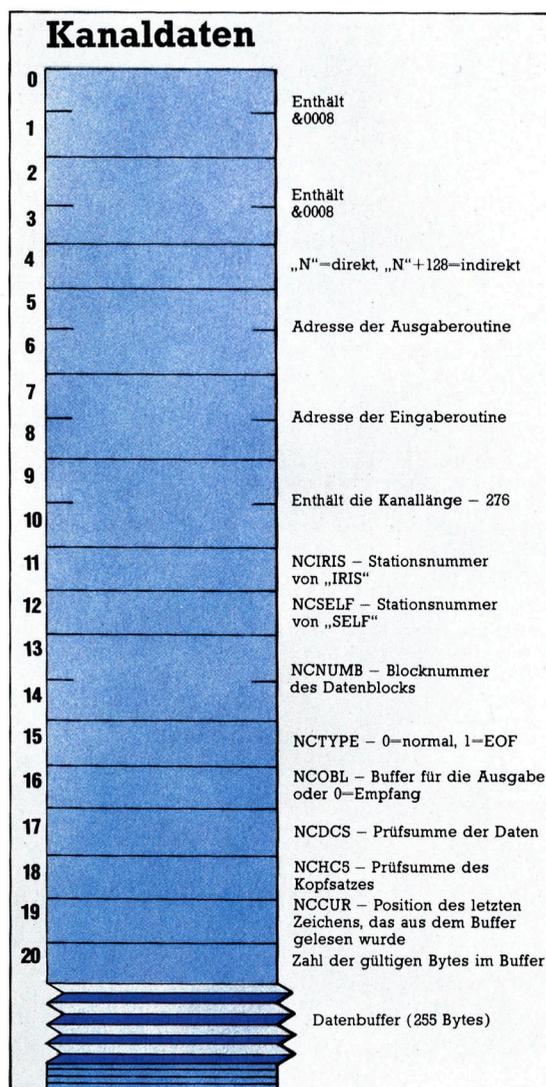
genommen hat, wird zunächst der Kopfsatz in den Kopfbereich des Kanals gestellt. NCOBL enthält die Anzahl der übermittelten Bytes und NCTYPE die Information, ob das empfangene Paket das Dateiende markiert.

Bedarfsgerecht

Bei einer erfolgreichen Übertragung erhöht sich die Kanalvariable NCNUMB um Eins.

In einem LAN-Netzwerk können Sie die Übermittlung Ihrer Daten mit diesen zwei Methoden genau auf Ihren Bedarf einstellen. In einem kleinen Netz mit nur zwei Maschinen sind die Hakencodes 48 und 47 allerdings weniger geeignet als die einfache Kommunikation mit RST #10 und CALL #15E6.

Damit endet unsere Untersuchung der Datenübertragungsmöglichkeiten des Interface 1. In der nächsten Folge sehen wir uns die neuen BASIC-Befehle des Interface 1 an. Darüberhinaus machen wir Sie damit vertraut, wie Sie mit Hilfe des Interface 1 eigene BASIC-Befehle definieren können, damit das BASIC Ihres Spectrum noch leistungsfähiger wird und Ihre eigenen Programme noch professioneller.



Hakencode 45 legt einen Netzwerkkanal von 276 Bytes an, in dem 20 Bytes für die Informationen des Kopfsatzes und 255 Bytes als Datenbuffer zur Verfügung stehen. Das Bild zeigt die Byte-Struktur des Kopfsatzes.

Für Strategen

Mit diesem Artikel endet das Go-Programm. Wir fügen die Zugsbewertungsroutinen ein, die auf der bekannten Analyse von Gruppen und Mustern basieren.

Einer der Gründe, warum Go so schwierig zu programmieren ist, liegt darin, daß Computer menschlichem Geist nicht ebenbürtig sind. Zur Zugauswahl kann ein Spieler aus Fleisch

und Blut intuitiv Gruppen und Muster analysieren. Da Micros derzeit nur sequentiell – also Schritt für Schritt – arbeiten, ist eine derartige Analyse problematisch. Dennoch können wir Routinen einfügen, die Verbindungen zwischen zusammenliegenden Gruppen untersuchen, so daß der Computer angemessen reagieren kann, indem er eigene Gruppen verbindet und verteidigt oder gegnerische angreift. Hier die „intelligenten“ Programmteile für den Schneider CPC, Commodore 64 und Spectrum.

Fünftes Modul

Commodore 64:

```

330 GOSUB 790
780 :
790 REM READ-PATTERNS ROUTINE
810 PAT=BOARD+556
830 FOR L=0 TO 71
840 READ P%:POKE PAT+L,(256+P%) AND 255
850 NEXT
860 DATA 32,17,16,2,-15,1
870 DATA -32,-17,-16,-2,15,-1
880 DATA 33,16,17,31,16,15
890 DATA -33,-16,-17,-31,-16,-15
900 DATA -14,1,-15,18,1,17
910 DATA 14,-1,15,-18,-1,-17
920 DATA 48,33,32,48,17,16
930 DATA -48,-33,-32,-48,-17,-16
940 DATA 3,-14,2,3,-15,1
950 DATA -3,14,-2,-3,15,-1
960 DATA 64,33,32,4,-14,2
970 DATA -64,-33,-32,-4,14,-2
980 RETURN
990 :
1000 REM*****
2580 IF LOCAT%=0 THEN GOSUB 2900:T$="DEF"
"
2590 IF LOCAT%=0 THEN GOSUB 3070:T$="ATT"
"
2600 IF LOCAT%=0 THEN GOSUB 3230:T$="SAT"
"
2610 IF LOCAT%=0 THEN GOSUB 3380:T$="SCN"
"
2890 :
2900 REM DEFEND-CONNECTION
2910 HI=-9999
2920 BV%=BLACK%:GOSUB 4420
2930 FOR A=1 TO 255:DCS%=PEEK(BOARD+A):I
F DCS%<>BLACK% GOTO 3010
2940 FOR P=PAT TO PAT+70 STEP 3
2950 B=PEEK(BOARD+((A+PEEK(P))AND 255)):
C=PEEK(BOARD+((A+PEEK(P+1))AND 255))
2960 IF B<>BLACK% OR C<>WHITE% GOTO 3000
2970 D=(A+PEEK(P+2))AND 255:SCR=RND(O)+P
EEK(WEIGHT+D)
2980 IF (D AND 240)=0 OR (D AND 15)=0 OR
SCR<=HI GOTO 3000
2990 LP%=D:LC%=BLACK%:GOSUB 3890:IF LL%=
0 AND CLIB%>2 THEN HI=SCR:LOCAT%=D
3000 NEXT
3010 NEXT
3020 BV%=0:GOSUB 4420
3030 RETURN
3040 :
3050 REM*****
3060 :
3070 REM ATTACK-CONNECTION ROUTINE
3080 HI=-9999
3090 BV%=WHITE%:GOSUB 4420
3100 FOR A=1 TO 255:ACS%=PEEK(BOARD+A):I
F ACS%<>WHITE% GOTO 3180
3110 FOR P=PAT TO PAT+70 STEP 3
3120 B=PEEK(BOARD+((A+PEEK(P))AND 255)):

```

```

C=PEEK(BOARD+((A+PEEK(P+1))AND 255))
3130 IF B<>WHITE% OR C<>BLACK% GOTO 3170
3140 D=(A+PEEK(P+2))AND 255:SCR=RND(O)+P
EEK(WEIGHT+D)
3150 IF (D AND 240)=0 OR (D AND 15)=0 OR
SCR<=HI GOTO 3170
3160 LP%=D:LC%=BLACK%:GOSUB 3890:IF LL%=
0 AND CLIB%>2 THEN HI=SCR:LOCAT%=D
3170 NEXT
3180 NEXT:BV%=0:GOSUB 4420
3190 RETURN
3200 :
3210 REM*****
3220 :
3230 REM START-ATTACK ROUTINE
3240 HI=-9999
3250 FOR A=1 TO 255:SAS%=PEEK(BOARD+A):I
F SAS%<>WHITE% GOTO 3330
3260 FOR P=PAT TO PAT+70 STEP 3
3270 B=PEEK(BOARD+((A+PEEK(P))AND 255)):
D=PEEK(BOARD+((A+PEEK(P+2))AND 255))
3280 IF B<>WHITE% OR D<>0 GOTO 3320
3290 C=(A+PEEK(P+1))AND 255:SCR=RND(O)+P
EEK(WEIGHT+C)
3300 IF (C AND 240)=0 OR (C AND 15)=0 OR
SCR<=HI GOTO 3320
3310 LP%=C:LC%=BLACK%:GOSUB 3890:IF LL%=
0 AND CLIB%>2 THEN HI=SCR:LOCAT%=C
3320 NEXT
3330 NEXT
3340 RETURN
3350 :
3360 REM*****
3370 :
3380 REM START-CONNECTION ROUTINE
3390 HI=-9999
3400 FOR A=1 TO 255:SCS%=PEEK(BOARD+A):I
F SCS%<>BLACK% GOTO 3470
3410 FOR P=PAT TO PAT+70 STEP 3
3420 C=PEEK(BOARD+((A+PEEK(P+2))AND 255)
):IF C>0 GOTO 3460
3430 B=(A+PEEK(P))AND 255:SCR=RND(O)+PEE
K(WEIGHT+B)
3440 IF (B AND 240)=0 OR (B AND 15)=0 OR
SCR<=HI GOTO 3460
3450 LP%=B:LC%=BLACK%:GOSUB 3890:IF LL%=
0 AND CLIB%>2 THEN HI=SCR:LOCAT%=B
3460 NEXT
3470 NEXT
3480 RETURN
3490 :
3500 REM*****
4410 :
4420 REM BOUNDARY ROUTINE
4430 FOR BX=0 TO 15
4450 BY=16*BX
4460 POKE BOARD+BX,BV%
4470 POKE BOARD+BY,BV%
4480 NEXT
4490 RETURN
4500 :
4510 REM*****
4520 REM*****END OF PROGRAM*****

```



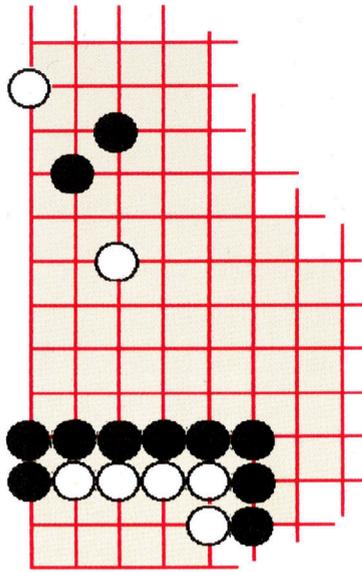
Schneider CPC 464/664:

```

280 GOSUB 790:REM read patterns
780 :
790 REM read patterns routine
810 pat=board+800
820 RESTORE 860
830 FOR 1%=0 TO 72
840 READ p%:POKE (pat+1%),(256+p%)AND 25
5
850 NEXT 1%
860 DATA 32,17,16,2,-15,1
870 DATA -32,-17,-16,-2,15,-1
880 DATA 33,16,17,31,16,15
890 DATA -33,-16,-17,-31,-16,-1
900 DATA -14,1,-15,18,1,17
910 DATA 14,-1,15,18,-1,-17
920 DATA 48,33,32,48,17,16
930 DATA -48,-33,-32,-48,-17,-16
940 DATA 3,-14,2,3,-15,1
950 DATA -3,14,-2,-3,15,-1
960 DATA 64,33,32,4,-14,2
970 DATA -64,-33,-32,-4,14,-2
980 RETURN
990 :
1000 REM *****
2580 IF location%=0 THEN GOSUB 2900:T$="
DEF"
2590 IF location%=0 THEN GOSUB 3070:T$="
ATT"
2600 IF location%=0 THEN GOSUB 3230:T$="
SAT"
2610 IF location%=0 THEN GOSUB 3380:T$="
SCN"
2900 REM defend connection routine
2910 hi=-9999
2920 bv%=black%:GOSUB 4420:REM boundary
2930 FOR a%=1 TO 255:dcs%=PEEK(board+a%)
:IF dcs%<>black% THEN 3010
:IF dcs%<>black% THEN 3010
2940 FOR p%=pat TO pat+70 STEP 3
2950 b%=PEEK(board+((a%+PEEK(p%))AND 255
))
c%=PEEK(board+((a%+PEEK(p%+1))AND 255
))
2960 IF b%<>black% OR c%<>white% THEN 30
00
2970 d%=(a%+PEEK(p%+2)) AND 255:score=RND
D(1)+PEEK(weight+d%)
2980 IF (d% AND 240)=0 OR (d% AND 15)=0
OR score<=hi THEN 3000
2990 lp%=d%:lc%=black%:GOSUB 3890:IF 11%
=0 AND clib%>2 THEN hi=score:locatio
n%=d%
3000 NEXT p%
3010 NEXT a%
3020 bv%=0:GOSUB 4420:REM boundary
3030 RETURN
3040 :
3050 REM *****
3060 :
3070 REM attack connection routine
3080 hi=-9999
3090 bv%=white%:GOSUB 4420:REM boundary
3100 FOR a%=1 TO 255:acs%=PEEK (board+a%
):IF cs%<>white% THEN 3180
3110 FOR p%=pat TO pat+70 STEP 3
3120 b%=PEEK(board+((a%+PEEK(p%))AND 255
))
c%=PEEK(board+((a%+PEEK(p%+1))AND 255
))
3130 ifb%<>white% OR c%<>black% THEN 317
0
3140 d%=(a%+PEEK(p%+2))AND 255:scr%=RND(
1)+PEEK(weight+d%)
3150 IF (d% AND 240)=0 OR (d% AND 15)=0
OR score<=hi THEN 3170
3160 lp%=d%:lc%=black%:GOSUB 3890:IF 11%
=0 AND clib%>2 THEN hi=score:loaction%
=
3170 NEXT p%
3180 NEXT a%
3190 RETURN
3200 :
3210 REM *****
3220 :
3230 REM start attack routine

```

Hohe Dichte



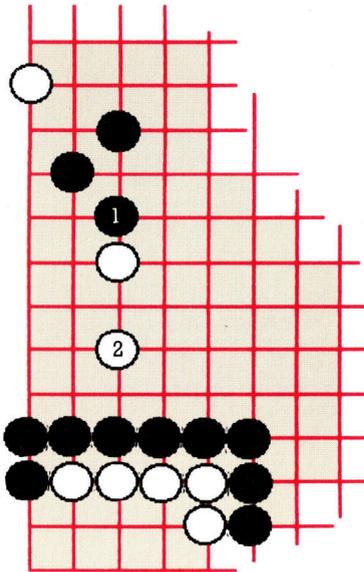
Dicht gepackt

Bei fest geformten Gruppen, wie der hier gezeigten schwarzen Gruppe, spricht man davon, daß sie eine „hohe Dichte“ besitzen. Da es keine schwachen Punkte gibt, ist ein Angriff nicht sehr vielversprechend. Eine wichtige Strategie ist daher: „Halte dich fern von diesen dicht gepackten Gruppen“.

```

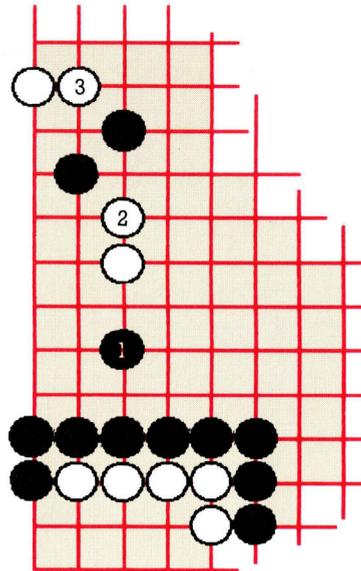
3240 hi=-9999
3250 FOR a%=1 TO 255:sas%=PEEK(board+a%)
:IF sas%<>white% THEN 3330
3260 FOR p%=pat TO pat+70 STEP 3
3270 b%=PEEK(board+((a%+PEEK(p%))AND 255
))
d%=PEEK(board+((a%+PEEK(p%+2))AND 255
))
3280 IF b%<>white% OR d%<>0 THEN 3320
3290 c%=(a%+PEEK(p%+1))AND 255:score=RND
(1)+PEEK(weight+c%)
3300 IF (c% AND 240)=0 OR (c% AND 15)=0
OR score<=hi THEN 3320
3310 lp%=c%:lc%=black%:GOSUB 3890:IF 11%
=0 AND clib%>2 THEN hi=score:location=c%
3320 NEXT p%
3330 NEXT a%
3340 RETURN
3350 :
3360 REM *****
3370 :
3380 REM start connection routine
3390 hi=-9999
3400 FOR a%=1 TO 255:scs%=PEEK(board+a%)
:IF scs%<>black% THEN 3470
3410 FOR p%=pat TO pat+70 STEP 3
3420 c%=PEEK(board+((a%+PEEK(p%+2))AND 2
55)):IF c%>0 THEN 3460
3430 b%=(a%+PEEK(p%))AND 255:score=RND(1
)+PEEK(weight+b%)
3440 IF (b% AND 240)=0 OR (b% AND 15)=0
OR score<=hi THEN 3460
3450 lp%=b%:lc%=black%:GOSUB 3890:IF 11%
=0 AND clib%>2 THEN hi=score:location%=b
%
3460 NEXT p%
3470 NEXT a%
3480 RETURN
3490 :
3500 REM *****
4410 :
4420 REM boundary routine
4440 FOR x%=0 TO 15
4450 y%=16-x%
4460 POKE (board+x%),bv%
4470 POKE (board+y%),bv%
4480 NEXT x%
4490 RETURN
4500 :
4510 REM *****
4520 REM ***** end of program *****

```



Ausnutzung hoher Dichte

Im allgemeinen ist es erfolgversprechend, eigene dichte Gruppen bei einem Angriff mitzuverwenden. In der hier gezeigten Situation setzt Schwarz über den einzelnen weißen Stein, wodurch er Weiß in Richtung der dichten schwarzen Gruppe am unteren Brettrand treibt. Weiß kann zwar versuchen, durch Setzen von Stein 2 die Situation zu verteidigen, doch ist das auf lange Sicht nicht erfolgversprechend, da der Zug noch näher an die recht dichte schwarze Gruppe führt.



Ebenfalls ein Fehlschlag

Spielt Schwarz unterhalb des weißen Steines, wird der Angriff für Schwarz ein Fehlschlag, da Weiß gleichzeitig den eigenen Stein verteidigen und die beiden benachbarten schwarzen Steine durch Setzen von Stein 2 angreifen kann. Setzt Weiß dann auch noch Stein 3, muß sich Schwarz sowohl dem Angriff als auch der Verteidigung stellen.

Sinclair Spectrum:

```

280 GO SUB 790
790 REM read-patterns routine
810 LET pat=board+556
820 RESTORE 860
830 FOR l=0 TO 71
840 READ p: POKE pat+1,p
850 NEXT l
860 DATA 32,17,16,2,-15,1
870 DATA -32,-17,-16,-2,15,-1
880 DATA 33,16,17,31,16,15
890 DATA -33,-16,-17,-31,-16,-1
5
900 DATA -14,1,-15,18,1,17
910 DATA 14,-1,15,-18,-1,-17
920 DATA 48,33,32,48,17,16
930 DATA -48,-33,-32,-48,-17,-1
6
940 DATA 3,-14,2,3,-15,1
950 DATA -3,14,-2,-3,15,-1
960 DATA 64,33,32,4,-14,2
970 DATA -64,-33,-32,-4,14,-2
980 RETURN
1000 REM *****
2580 IF location=0 THEN GO SUB
2900: LET t$="DEF"
2590 IF location=0 THEN GO SUB
3070: LET t$="ATT"
2600 IF location=0 THEN GO SUB
3230: LET t$="SAT"
2610 IF location=0 THEN GO SUB
3380: LET t$="SCN"
2900 REM defend-connection routine
2910 LET hi=-9999
2920 LET bv=black: GO SUB 4420
2930 FOR a=1 TO 255: LET dcs=PEEK
K (board+a): IF dcs<>black THEN
GO TO 3010
2940 FOR p=pat TO pat+70 STEP 3
2950 LET b=a+PEEK p: LET c=a+PEE
K (p+1)
2952 IF b>255 THEN LET b=b-256:
GO TO 2952
2954 IF c>255 THEN LET c=c-256:
GO TO 2954
2956 LET b=PEEK (board+b): LET c
=PEEK (board+c)
2960 IF b<>black OR c<>white THE
N GO TO 3000
2970 LET d=a+PEEK (p+2)
2972 IF d>255 THEN LET d=d-256:
GO TO 2972
2974 LET score=RND+PEEK (weight+
d)
2980 IF INT (d/16)=0 OR d-16*INT
(d/16)=0 OR score<=hi THEN GO
TO 3000
2990 LET lp=d: LET lc=black: GO
SUB 3890: IF ll=0 AND clib>2 THE
N LET hi=score: LET location=d
3000 NEXT p
3010 NEXT a
3020 LET bv=0: GO SUB 4420
3030 RETURN
3070 REM attack-connection routi
ne
3080 LET hi=-9999
3090 LET bv=white: GO SUB 4420
3100 FOR a=1 TO 255: LET acs=PEE
K (board+a): IF acs<>white THEN
GO TO 3180
3110 FOR p=pat TO pat+70 STEP 3
3120 LET b=a+PEEK p: LET c=a+PEE
K (p+1)
3122 IF b>255 THEN LET b=b-256:
GO TO 3122
3124 IF c>255 THEN LET c=c-256:
GO TO 3124
3126 LET b=PEEK (board+b): LET c
=PEEK (board+c)
3130 IF b<>white OR c<>black THE
N GO TO 3170
3140 LET d=a+PEEK (p+2)
3142 IF d>255 THEN LET d=d-256:
GO TO 3142
3144 LET score=RND+PEEK (weight+
d)
3150 IF INT (d/16)=0 OR d-16*INT
(d/16)=0 OR score<=hi THEN GO
TO 3170
3160 LET lp=d: LET lc=black: GO
SUB 3890: IF ll=0 AND clib>2 THE
N LET hi=score: LET location=d
3170 NEXT p
3180 NEXT a: LET bv=0: GO SUB 44
20
3190 RETURN
3230 REM start-attack routine
3240 LET hi=-9999
3250 FOR a=1 TO 255: LET sas=PEE
K (board+a): IF sas<>white THEN
GO TO 3330
3260 FOR p=pat TO pat+70 STEP 3
3270 LET b=a+PEEK p: LET d=a+PEE
K (p+2)
3272 IF b>255 THEN LET b=b-256:
GO TO 3272
3274 IF d>255 THEN LET d=d-256:
GO TO 3274
3280 LET b=PEEK (board+b): LET d
=PEEK (board+d)
3280 IF b<>white OR d<>0 THEN G
O TO 3320
3290 LET c=a+PEEK (p+1)
3292 IF c>255 THEN LET c=c-256:
GO TO 3292
3294 LET score=RND+PEEK (weight+
c)
3300 IF INT (c/16)=0 OR c-16*INT
(c/16)=0 OR score<=hi THEN GO
TO 3320
3310 LET lp=c: LET lc=black: GO
SUB 3890: IF ll=0 AND clib>2 THE
N LET hi=score: LET location=c
3320 NEXT p
3330 NEXT a
3340 RETURN
3380 REM start-connection routi
ne
3390 LET hi=-9999
3400 FOR a=1 TO 255: LET scs=PEE
K (board+a): IF scs<>black THEN
GO TO 3470
3410 FOR p=pat TO pat+70 STEP 3
3420 LET c=a+PEEK (p+2)
3422 IF c>255 THEN LET c=c-256:
GO TO 3422
3424 LET c=PEEK (board+c): IF c>
0 THEN GO TO 3460
3430 LET b=a+PEEK p
3432 IF b>255 THEN LET b=b-256:
GO TO 3432
3434 LET score=RND+PEEK (weight+
b)
3440 IF INT (b/16)=0 OR b-16*INT
(b/16)=0 OR score<=hi THEN GO
TO 3460
3450 LET lp=b: LET lc=black: GO
SUB 3890: IF ll=0 AND clib>2 THE
N LET hi=score: LET location=b
3460 NEXT p
3470 NEXT a
3480 RETURN
4420 REM boundary routine
4440 FOR x=0 TO 15
4450 LET y=16*x
4460 POKE board+x,bv
4470 POKE board+y,bv
4480 NEXT x
4490 RETURN
4500:
4510 REM *****
4520 REM *****END OF PROGRAM*****

```



Signalabtastung

Wir beschließen unsere detaillierte Untersuchung der einzelnen Anschlüsse des 7135-Wandlers und bauen eine Zusatzschaltung für einen zweiten Taktgenerator-Kreis auf, mit der sich Messungen schneller durchführen lassen.

Die Bauteile im einzelnen: 7/8. REF.CAP: Die Kapazität und auch die Qualität dieses Bauteils ist nicht sehr kritisch. Wir schlagen einen Kondensator von 1µF vor. Sie können aber auch eine geringere Kapazität wählen – das hat nur zur Folge, daß Sie nach einer Überschreitung des Meßbereiches einige Sekunden länger warten müssen, bis sich das Gerät wieder „erholt“.

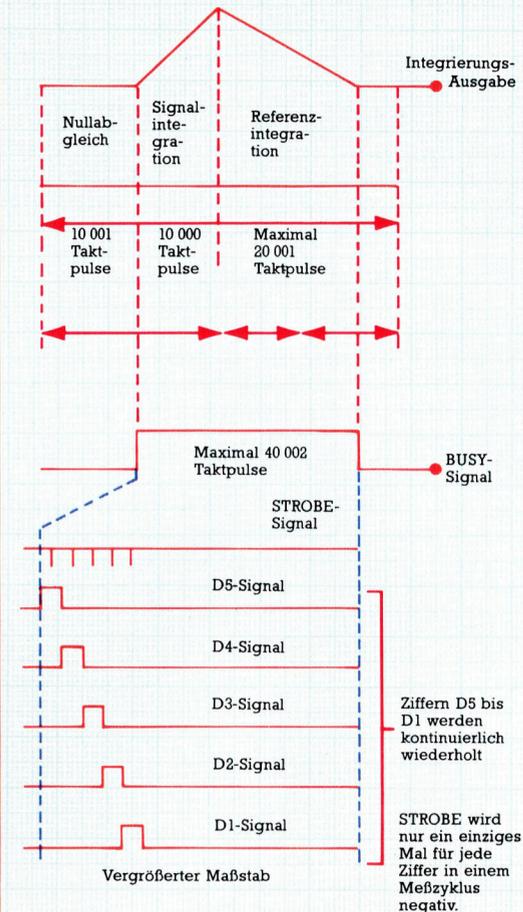
9/10. IN–/IN+: Diese beiden Anschlüsse gehören zum analogen Masseanschluß (Pin 3). Es können Spannungen zwischen –4 Volt und +4,5 Volt verarbeitet werden. Diese Spannungen können erdfrei sein, oder der IN-Pin kann mit der (analogen) Masse verbunden werden.

11. V+: Zur Versorgung des ICs ist hier der Anschluß einer 5-Volt-Spannungsquelle erforderlich, die allerdings nur 3mA liefern muß.

12. D5: Es gibt fünf digitale enable-Ausgänge (D1–D5). (Die weiteren Ausgänge liegen auf den Pins 17–20.) Sie werden dafür genutzt, die einzelnen Stellen des LED-Displays nacheinander zu aktivieren, was für den Multiplex-Betrieb erforderlich ist. Jedes Ziffern-enable wird nacheinander für 200 Taktpulse High. Dabei wird von D5, dem höchstwertigen Bit (MSB), zu D1 (niedrigstes Bit) gezählt. Wenn einer der Anzeigentreiber-Ausgänge aktiv wird, steigt seine Spannung von etwa 0,25 Volt auf rund 4 Volt an. Der angeschlossene Transistor wird durchgesteuert, und der Strom gelangt zum Anzeigeelement der entsprechenden Dezimalstelle. Die Display-Treiberanschlüsse aktivieren die Segmente nur, die Ansteuerung erfolgt mit BCD-Signalen (B1, B2, B4 und B8), die von dem Siebensegment-Decoder/Treiber 7447 geliefert werden. Mit logischen Schaltungen, die ein Vierleitungs-BCD-Signal zum Ansteuern einer Sieben-Segment-Anzeige umwandeln, haben wir uns bereits befaßt.

13. B1 (LSB): Es gibt vier BCD(Binär codierter Dezimalwert)-Ausgangsleitungen: B1 auf Pin 13, B2 auf Pin 14, B4 auf Pin 15 und B8 auf Pin 16. Gemeinsam liefern sie die darzustellende, binär codierte Dezimalzahl. Die Ziffern-enable-Signale sind so synchronisiert, daß immer nur die gerade aktive Ziffer mit dem für sie richti-

Zeitdiagramm



Nach dem Abschluß einer A/D-Wandlung beginnt das IC 7135 mit der Zifferausgabe im Multiplexverfahren. Der Chip stellt auch ein STROBE-Signal bereit, das zusammen mit den ersten fünf BCD-Ziffernsignalen ausgegeben wird. Das Signal kann zur Exterübertragung der Daten genutzt werden.

gen Anzeigewert belegt wird.

Alle fünf Ziffern scheinen auf dem Display gleichzeitig dargestellt zu werden, in Wahrheit sind sie jedoch im Multiplex-Verfahren angesteuert und erscheinen in schneller Folge nacheinander. Dadurch leuchten die LED-Ziffern schwächer als bei dauerhafter, gleichmäßiger Ansteuerung. Das ist jedoch kein Fehler: Der Stromverbrauch der LEDs sinkt auf ein Fünftel ab und hilft Batterien sparen.

Für die gleichzeitige Steuerung aller Ziffern wären normalerweise 20 BCD-Ausgangsleitungen erforderlich. Damit bräuchten wir auch fünf 7447-Treiber ICs. Das Multiplex-Verfahren erspart also nicht nur Leitungen, sondern auch eine Vielzahl von Bauteilen.

21. BUSY: Diese Leitung ist besonders dann wichtig, wenn das Digitalvoltmeter mit dem



Computer in Verbindung treten soll. Wenn die Signalleitung BUSY auf High-Potential liegt, bedeutet das, daß der A/D-Wandler gerade mit Berechnungen beschäftigt ist, also die Daten auf Ziffern- und BCD-Ausgängen momentan keine definierte Bedeutung haben. Zu Beginn jeder Signal-Integrationsphase wird der BUSY-Ausgang High und bleibt bis zum ersten Taktimpuls nach dem Nulldurchgang in diesem Zustand. Erst wenn er wieder auf Low liegt, befinden sich die Signalleitungen in einem stabilen Zustand mit gültigen Werten, die vom Computer gelesen werden können.

22. CLOCK IN: Zur Steuerung des 7135 ist ein externer Taktgenerator erforderlich. Die Taktschwindigkeit sollte 1,2 MHz nicht überschreiten – das ist allerdings schon recht schnell, die gemessene Spannung wird also nach sehr kurzer Zeit von neuem überprüft. Eine so hohe Frequenz erzeugt allerdings zusätzliche Probleme bei der Auswahl von Bauteilen. Der von uns verwendete Chip 555 kann die Grenzen der Geschwindigkeit des Wandlers nicht voll ausnutzen: Er erzeugt nur Frequenzen bis maximal 100 kHz.

Da ein vollständiger Meßzyklus maximal 40 002 Taktpulse dauert, bestimmt die Taktfrequenz die Geschwindigkeit der Messung. Bei einer Taktfrequenz von 100 kHz wären das etwa 2,5 Messungen pro Sekunde. Wenn Sie schneller messen möchten, können Sie die Frequenz auf 1,2 MHz steigern, dazu brauchen Sie allerdings einen zweiten Schaltkreis zur Erzeugung der Taktsignale. Die Kondensatoren für automatische Nullstellung und Referenz müssen zudem von extrem guter Qualität sein. Die Meßgenauigkeit wird bei hoher Geschwindigkeit schlechter. Der 7135-Chip arbeitet mit Taktfrequenzen bis herunter zu 1 Hz – dann müßten Sie allerdings 40 002 Sekunden auf Ihr Ergebnis warten!

23. POLARITY: Dieser Pin gibt ein Signal aus, das die Polarität der gemessenen Spannung angibt. Bei allen positiven Meßwerten einschließlich der 0,0000 ist das Signal positiv. Es

wird genutzt, um das Plus-Zeichen in der linken, 1/2stelligen Ziffernanzeige des Displays zu aktivieren.

24. DIGITAL GROUND: (Digitale Masse): Siehe Pin 3 (Analoge Masse=ANALOGUE GROUND)

25. RUN/HOLD: Liegt dieser Eingang auf High, dann befindet sich der Wandler-Chip im Freilaufbetrieb und gibt seine Meßwerte nach jeweils 40 002 Taktimpulsen aus. Wird er auf LOW geschaltet (entweder durch ein HOLD-Signal vom Rechner oder durch einen Schalter am Voltmeter), wird die aktuelle Anzeige festgehalten. Nach dem Anhalten startet ein mindestens 300 Nanosekunden andauernder positiver Wert an diesem Pin einen neuen Meßzyklus. Für den Anschluß über langsame Parallelschnittstellen ist dieser Pin ideal, wenn nämlich nicht alle fünf BCD-Ausgangssignale gleichzeitig verarbeitet werden können.

Eine Vereinfachung

26. STROBE: In jedem Meßzyklus geht das STROBE-Signal fünfmal auf Minus, für jede gültige Ziffer des Meßwertes einmal. Die STROBE-Pulse erscheinen in der Mitte jedes Ziffern-enable-Ausgangssignals (siehe Ablaufplan). Das STROBE-Signal soll die Verbindung zu externen Mikroprozessoren vereinfachen und kann genutzt werden, um die BCD-Daten in Zwischenspeicher zu schaffen. STROBE wird nur für einen halben Taktzyklus lang negativ (aktiv) und zwar für jede Ziffer nur einmal. Die Displaytreiber und BCD-Ausgänge wiederholen ihre Ausgabe während eines Meßzyklus mehrfach (um eine dauerhafte Anzeige des Wertes zu ermöglichen).

27. OVER RANGE (Meßbereichsüberschreitung): Dieses Signal zeigt eine Überschreitung des zulässigen Meßbereiches (mehr als 20 000 Impulse) an. In komplizierteren Multimetern könnte man es für eine automatische Bereichswahl einsetzen – eine Schaltung, die selbsttätig ohne mechanische Schalter die Eingangsempfindlichkeit des Instruments verändert. In unserer einfachen Anwendung soll es nur zur Steuerung einer Leuchtdiode eingesetzt werden, die uns warnt, wenn die Abschwächer-schaltung am Eingang des Multimeters auf einen niedrigeren Wert gestellt werden muß.

28. UNDER RANGE (Meßbereichsunterschreitung): Dieses Signal geht auf HIGH, wenn die Eingangsspannung zu niedrig ist. Wie das OVER RANGE-Signal ist es zuerst einmal dafür gedacht, eine automatische Bereichsumschaltung zu steuern. Voltmeter mit automatischer Bereichswahl nutzen diese beiden Signale für die Selektion des Meßbereiches. Für uns wäre das etwas zu schwierig. Wir verzichten daher auf diese Funktion.

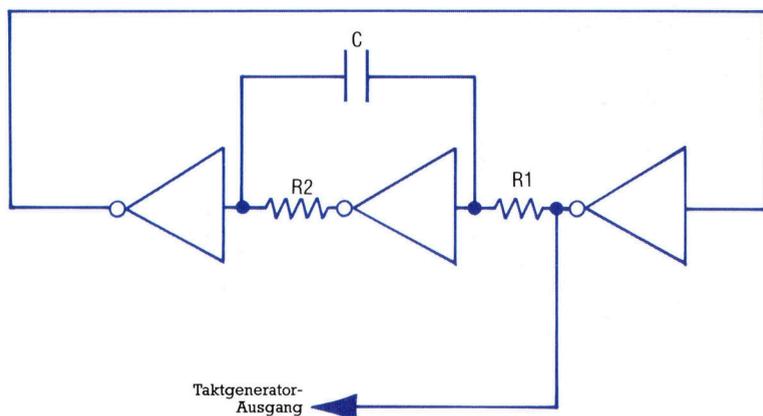
Die Grundlagen der Schaltung des Digitalvoltmeters sind damit recht ausführlich dargestellt worden. Im nächsten Abschnitt geht es an den konkreten Aufbau des Gerätes.

Diese Schaltung kann alternativ zu dem von uns vorgeschlagenen Timer-Chip 555 eingesetzt werden. Sie besteht aus drei TTL-Invertierern, zwei Widerständen und einem Kondensator. Die Formel zur Berechnung der gewünschten Frequenz lautet:

$$F = 2 C (R1 + R2)$$

Da der 7135-Wandler mit recht hoher Geschwindigkeit arbeitet, können Sie diesen Taktgeber einsetzen, wenn es mehr auf hohe Meßraten als auf exakte Ergebnisse ankommt.

Taktgenerator für höhere Frequenzen



Fachwörter von A bis Z

Password = Paßwort

Im kommerziellen Bereich wird bei Computersystemen zur Benutzeridentifikation ausgiebig von ‚Paßwörtern‘ Gebrauch gemacht. Das sind Buchstaben- und Zeichenkombinationen, die beim ‚Einloggen‘ eingegeben werden müssen, um zu verhindern, daß Unbefugte kostbare Rechenzeit stehlen oder Zugang zu vertraulichen Dateien haben.

Wer sich bei einem derartig geschützten System einloggen (anmelden) will, wird vom Rechner zunächst nach dem Paßwort gefragt. Es ist im Speicher hinterlegt, und der Benutzer erhält nur Zutritt, wenn seine Eingabe mit dem Muster exakt übereinstimmt. Darüber hinaus können bestimmte Systembereiche noch durch Paßworte gesperrt sein.

Bei vielen Datenbanken lassen sich die Paßwörter auch vom Benutzer selbst definieren. Für das erste Einloggen wird zwar eine Kombination zugeteilt, aber danach ist ein beliebiger Austausch möglich.

Manche User haben offensichtlich Schwierigkeiten, sich geeignete Paßwörter einfallen zu lassen. Vielfach werden schlicht Name und Geburtstag des Benutzers genommen, unter Umständen auch der angesteuerte Datenbank-Untertitel. Der verbreitete Phantasiemangel ist ein Segen für die ‚Hacker‘ und kommt ihnen beim Knacken von Computersystemen außerordentlich zustatten.

Patch = Flicker

Wenn ein Programm nicht richtig laufen will, ist ein nachträglich angehefteter ‚Patch‘ (Flicker) aus ein paar Korrekturbefehlen manchmal unentbehrlich. Professionelle Programmierer runzeln bei solchem Verfahren allerdings die Stirn und lassen einen ‚Patch‘ höchstens zu, wenn nur eine Kleinigkeit in einem fertigen Programm auszubessern ist. Der bessere Weg ist natürlich, gleich sauber strukturiert und zuverlässig zu programmieren. Das erspart meist die Flickarbeit; andernfalls führt die Korrektur der Korrekturen leicht zum totalen Chaos.

Hardwaremäßige ‚Patches‘ sind

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

die fliegenden Drähte, die zuweilen einer Schaltung auf die Beine helfen müssen, wenn die Platine einen kleinen Fehler enthält. Gebräuchlich ist der Begriff auch für die Kabelbrücken, mit denen sich auf sogenannten ‚Patchboards‘ (Steckbrettern) kurzfristig Verbindungen herstellen lassen.

Pattern Recognition = Mustererkennung

Die Mustererkennung ist ein wichtiges Forschungsgebiet im Bereich der Künstlichen Intelligenz. Der Begriff umfaßt eigentlich jede Form der



Der Apricot Portable ist als einer der ersten Rechner von vornherein für die Spracheingabe eingerichtet. Einfache Wörter wie Befehle und Ziffern nimmt der Rechner über das eingebaute Mikrofon zur Kenntnis.

automatischen Auswertung von Informationsmustern, wird aber meist für die visuelle und die Spracherkennung verwendet. Da der Computer weder Augen noch Ohren hat, müssen für ihn optische oder akustische Eindrücke zunächst mit Hilfe von Kamera bzw. Mikrofon in elektrische Signale umgesetzt werden. Der Erkennungsprozeß beruht dann auf der Deutung dieser Information anhand eines Vergleichs mit im Rechner gespeicherten Vorgaben.

Wenn dabei versucht wird, das unbekannte Objekt mit vorrätigen Schablonen zur Deckung zu bringen, spricht man vom ‚Klassifikationsverfahren‘. Dem Rechner wird beispielsweise die Form eines Quadrats einprogrammiert, mit der sich dann mehr oder weniger direkt ein Vergleich durchführen läßt. Obwohl diese Methode am einfachsten erscheint, führt sie bei komplexen Mustern zu erheblichen Problemen.

Beim ‚analytischen‘ Verfahren wird das Muster nicht als Einheit betrachtet, sondern es wird auf seine Komponenten hin analysiert. Im genannten Beispiel könnte der Rechner das Muster daraufhin untersuchen, ob seine Ecken rechtwinklig und alle Kanten gleich lang sind, so daß es sich als Quadrat einordnen läßt. Dieses Verfahren erfordert zwar wesentlich mehr Programmieraufwand, ist aber universeller.

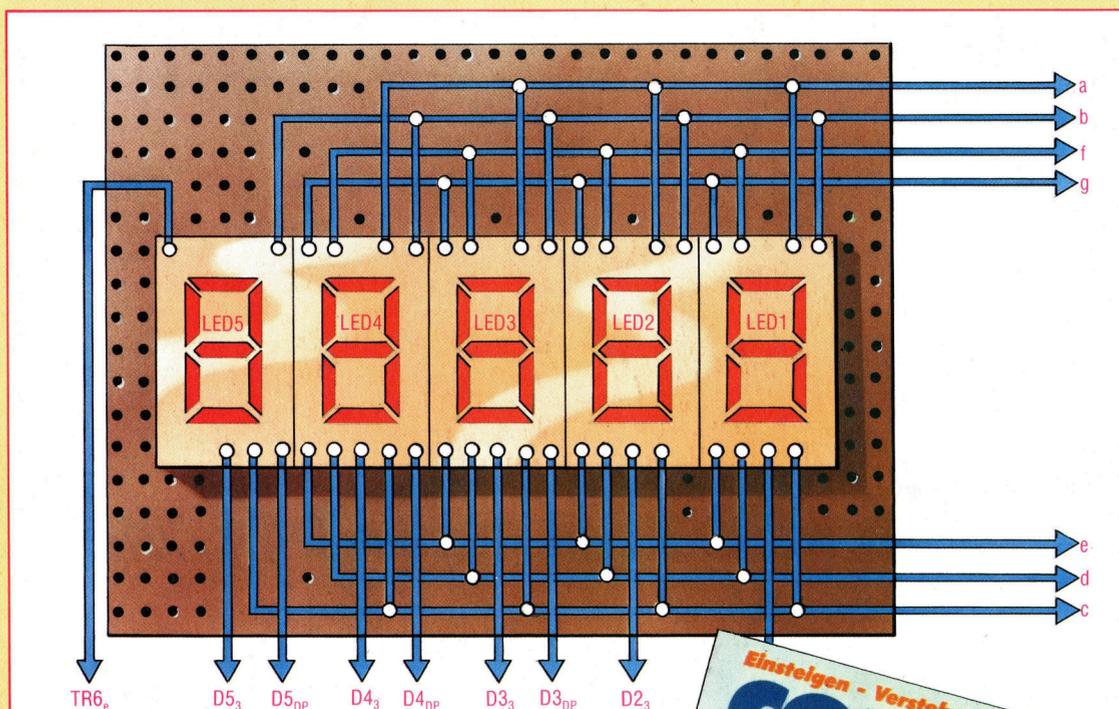
Wenn der Rechner überhaupt erst einmal bestimmte Muster identifizieren kann, ist als nächstes dafür zu sorgen, daß er sie auch bei gewissen Abweichungen von der starren Schablone noch wiedererkennt. Das spielt besonders bei der Spracherkennung eine entscheidende Rolle. Bei der Sprachanalyse wird das Frequenzspektrum des akustischen Signals untersucht.

Bildnachweise

2043: Crispin Thomas
2046: MTV
2047: British Telecom
2049, 2051, 2054, 2066,
2069: Caroline Clayton
2050: Hil Rudge
2054, 2061, 2067, 2071: Kevin Jones

computer kurs

Heft 75



Unter Spannung

Zur Komplettierung unseres Digitalmeßgerätes erfolgt nun die Montage der Segment-Anzeige, einem wichtigen Bauteil.



Letzte Feinheiten

Das Go-Programmier-Projekt endet mit einigen ergänzenden Überlegungen zur Antwortzeit des Rechners.



Robuster Rechner

Research Machines 380Z stellt sich als stabiles und belastbares Gerät vor.



In der Bibliothek

C ist auch in Verbindung mit dem Betriebssystem Unix verwendbar.



Computer und Beruf

Industrie und Handel im Computer-Bereich bieten viele Arbeitsplätze.

