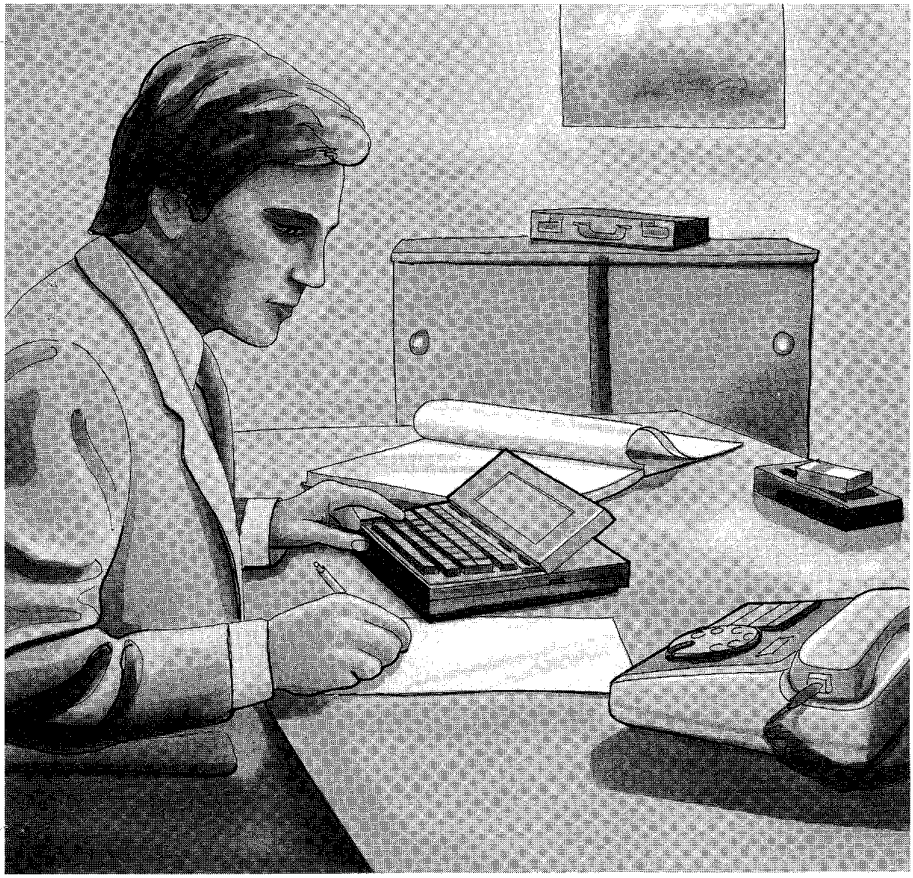
M 10

Benutzer-Handbuch



olivetti

Das Handbuch dient der Information, sein Inhalt ist ohne ausdrückliche schriftliche Vereinbarung nicht Vertragsgegenstand. Technische Änderungen behalten wir uns vor. Die angegebenen Daten sind lediglich Nominalwerte.

VORWORT

Der Olivetti M10 ist ein kleiner, mobiler Personal Computer von einer solchen Vielseitigkeit, daß er in idealer Weise für unterwegs, das Büro und den Hausgebrauch geeignet ist.

Unter den vielen herausragenden Qualitäten des M10 sind besonders hervorzuheben die leichte Bedienung, eine 8-Zeilen-Flüssigkristall-Anzeige, eine professionelle Tastatur, die Kompatibilität zu vielen unterschiedlichen Peripherie-Geräten, ein weiter Bereich von Textverarbeitungsfunktionen, eine umfassende Möglichkeit, ihn in BASIC zu programmieren und ein direkter Anschluß an andere Computer oder an Hostrechner über das Telefonnetz.

Zur bequemen Benutzung ist das Handbuch in zwei Abschnitte aufgeteilt.

Abschnitt 1 (Kapitel 1-9), das "M10 Bediener-Handbuch", beschäftigt sich mit den wesentlichen Bedienungsfunktionen und gibt einen Gesamtüberblick des M10 und seine Peripherie, Beschreibung seiner verschiedenen Einrichtungen und Anwendungsmöglichkeiten und Einzelheiten darüber, wie man aus den fünf fest gespeicherten Programmen, mit denen der M10 ausgerüstet ist, Nutzen zieht. Mit anderen Worten: Dieses Handbuch führt den M10 Benutzer in alle Aspekte der Bedienung ein und liefert alle notwendigen Informationen.

Abschnitt 2 beschäftigt sich ausschließlich mit BASIC, (Beginners Allpurpose Symbolic Instruction Code), der erweiterten intelligenten Programmiersprache des M10. Dieser Teil ist betitelt "M10 BASIC-Handbuch". Die Absicht dabei ist nicht, Ihnen das Programmieren in BASIC zu zeigen, sondern dieser Teil ist vielmehr eine Zusammenfassung der BASIC Struktur, der Syntax und der Befehlsbeschreibung, auf die der Anwender im Zweifelsfalle zurückgreifen kann. Der Abschnitt 2 enthält insbesondere ein alphabetisches Nachschlage-Verzeichnis aller BASIC-Befehle, Meldungen und Funktionen, wobei ihre Anwendungen und ihr Format mit vielen Beispielen dargestellt werden.

INHALTSVERZEICHNIS

ABSCHNITT 1

M10 BEDIENER-HANDBUCH

ABSCHNITT 2

M10 BASIC-HANDBUCH

Anmerkung:

Die folgende Schreibweise ist durchgängig in diesem Handbuch benutzt, um Verwirrung zu vermeiden:

1. Wörter, die in Großbuchstaben eingegeben werden, stellen Befehle oder Anweisungen an den Computer dar und sollten in dem angegebenen Format eingegeben werden. Dies ist nicht immer notwendig, da der M10 alle Buchstaben in Großbuchstaben umwandelt (außer im Text-Modus). Z. B.

   PRINT TIME$ kann eingegeben werden als
   print time$ oder
   PrinT tIMe$ aber nicht
   PRINT TIME $ wo ein Leerschritt hinzugefügt worden ist.

   Zum Beispiel haben Symbole eine genau definierte Funktion und müssen eingegeben werden wie angezeigt.

   Eine Liste aller Symbole, die beim Programmieren in BASIC benutzt werden, findet sich im Teil 2 dieses Handbuches am Anfang.

2. Wörter, die in Kleinbuchstaben gedruckt sind, sind Platzhalter, für die der Anwender den entsprechenden Wert einsetzen muß.

   PRINT Dateiname

   In diesem Beispiel stellt PRINT eine Anweisung dar, die genauso eingegeben werden muß, wie sie aufgeführt wird. Der Anwender muß dann nur noch den wirklichen Dateinamen der zu druckenden Datei angeben.

3. Die folgende Liste zeigt die Bedeutungen verschiedener Gruppen von Wörtern und Symbolen.

   A|B|C wählen Sie entweder A, B oder C

   [ ] Ausdrücke in eckigen Klammern sind optional (wahlweise anzugeben oder nicht)

... drei Punkte geben an, daß weitere Werte bzw. Parameter in der Anweisung angegeben werden können.

DATA constant-1, constant-2 ..., constant-n

⟨**KEY**⟩ Die Taste, deren Name in Fettdruck erscheint, sollte gedrückt werden, z. B.

⟨**ENTER**⟩ bedeutet "drücken Sie die Taste ENTER"

⟨**CTRL** + **x**⟩ bedeutet "drücken Sie die Tasten **x** und **CONTROL** gleichzeitig"

⟨**GRPH** + **SHIFT**⟩ bedeutet "drücken Sie die Tasten **GRPH** und **SHIFT** gleichzeitig"

4. Texte und Meldungen, die auf dem Bildschirm erscheinen, werden fett gedruckt, z. B.

File to edit?

ALLGEMEINE BESCHREIBUNG

Der M10 ist ein selbständiger, leicht zu benutzender, voll tragbarer Personal Computer. Er ist in Abbildung 1-1 unten dargestellt.

Abbildung 1-1 Der M10 Personal Computer

Der M10 ist in idealer Weise geeignet für eine große Anzahl von Anwendungsbereichen. Er bietet Möglichkeiten zum Programmieren, für die Textverarbeitung, Datenübertragung, für die Führung eines elektronischen Adreßbuches und für die Terminplanung. Der M10 kann entweder von seiner eigenen Batterie betrieben werden (vier 1,5 Volt Trockenzellen) oder durch Anschluß an das Netz, wobei man ein Netzteil benutzen muß. Er wiegt nur 1900 Gramm (4 lb, 3 oz.). Seine Abmessungen sind 30 x 22 x 6 cm (11 3/4 x 8 1/2 x 2 3/8 in.), und er paßt leicht in eine normale Aktentasche, worduch er in idealer Weise für den Gebrauch auf der Reise, im Außendienst, im Büro wie auch zu Hause geeignet ist. Der M10 wird in einem attraktiven Etui geliefert, das ihn vor Beschädigung und Staub schützt.

Ins Auge fallende Einrichtungen des M10 sind seine Tastatur und sein Bildschirm. Eine der am häufigsten vorgebrachten Kritiken an tragbaren Computern ist, daß die Tastatur nicht leicht zu bedienen ist oder daß die Anzeige schwierig zu lesen ist. Solch eine Kritik erwarten wir schwerlich dem M10 gegenüber mit seiner soliden Tastatur in Originalgröße und seinem Flüssigkristall-Display von 8 Zeilen und 40 Zeichen.

TECHNISCHE BESCHREIBUNG

Der M10 ist ausgestattet mit einem ROM (Read only Memory) von 32 KB Kapazität, das fünf integrierte Anwendungsbereiche ermöglicht. Diese sind:

- BASIC Interpreter, der es Ihnen erlaubt, Programme zu erstellen, zu speichern und auszuführen.

- TEXT zur Erstellung von Texten und zur Textverarbeitung.

- TELCOM für die Datenübertragung.

- ADDRSS zur Erstellung eines elektronischen Adreß-Verzeichnisses und Telefonbuches.

- SCHEDL zur leichten Erstellung und jederzeitigem Abruf von Terminplänen und anderen Aktivitäten.

Die anderen Haupteigenschaften sind:

- 8KB Arbeitsspeicher (RAM, Random Access Memory), ausbaubar in 8KB-Schritten bis auf 32k.

- OKI 80C85 Mikroprozessor, kompatibel mit dem Intel 8085

- Die Tastatur mit 188 alphanumerischen und graphischen Zeichen. Es stehen sowohl eine deutsche Tastatur als auch englische, französische und italienische Versionen zur Verfügung.

- Zusatzfunktionstasten für spezielle Anwendungsbereiche.

- Neigbarer Flüssigkristall-Bildschirm (LCD), mit einstellbarem Bildschirmkontrast.

- Schnittstellen für Peripherien (paralleler und serieller Anschluß möglich).

Der M10 hat drei Inhaltsverzeichnisse im Speicher. Dateinamen erhalten einen Anhang (Typbezeichnung) entsprechend dem Verzeichnis, in dem sie gespeichert werden.

Dateinamen mit dem Anhang '.BA' sind in Binärform gespeicherte BASIC-Programme.

Dateien mit dem Anhang '.CO' sind Programme in Maschinensprache.

Dateien mit dem Anhang '.DO' sind solche, die mit dem Programm 'TEXT' erstellt wurden, oder es sind BASIC-Programme, die im ASCII-Format abgespeichert wurden oder BASIC-Dateien. (Dieses Handbuch beschäftigt sich nur mit '.DO'-Dateien, die mit Hilfe des Programmes 'TEXT' erstellt wurden).

SCHNITTSTELLEN UND PERIPHERIE

Der M10 hat Schnittstellen für die Verbindung mit einer Vielzahl von Peripheriegeräten. Neben Schnittstellen für Drucker, Kassetten-Rekorder und Bar-Code-Leser verfügt der M10 über die Möglichkeit, mit Hilfe eines Akustik-Kopplers Datenübertragung zu betreiben. Für alle Schnittstellen stehen spezielle Kabel zur Verfügung.

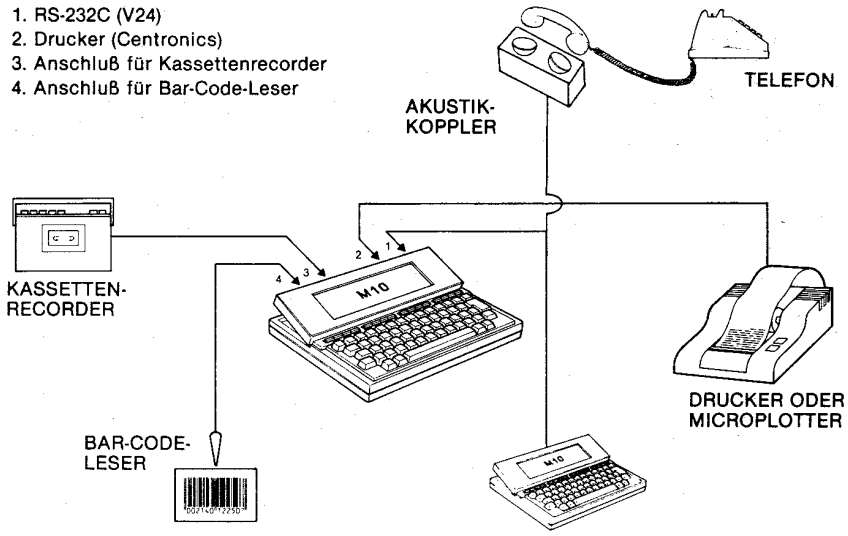


Abbildung 1-2 Typische System-Konfiguration

Wesentliche Bausteine der Peripherie sind:

Kassettenrekorder

An den M10 kann man jedes Kassettenrekordermodell anschließen, vorausgesetzt, es verfügt, zusätzlich zu den Standardeinrichtungen, über einen Eingang (MIC), einen Ausgang (EAR), einen Eingang für Fernsteuerung (REM) und ein Bandzählwerk. Siehe auch Kapitel 9 für weitere Einzelheiten.

Drucker

Jeder Drucker oder Mikroplotter, der dem Industriestandard für parallele Schnittstellen entspricht, kann an den M10 mit Hilfe von Olivetti Mikroplotter-Kabeln angeschlossen werden.

Telefon

Der M10 kann über RS232 und einen Akustik-Koppler an das öffentliche Telefonnetz zur Datenfernübertragung angeschlossen werden.

Bar-Code-Leser

Der M10 hat eine Schnittstelle zur Verbindung mit einem Hewlett-Packard HEDS-3050 oder HEDS-3000 Bar-Code-Leser für Lagerbestandsverwaltung.

Der M10 kann über die Schnittstelle RS-232C an einen anderen Computer direkt angeschlossen werden.

DIE TASTATUR UND DER BILDSCHIRM

Das Modell M10 ist mit einer der vier nationalen Tastaturen - Deutschland, Italien, Frankreich oder Großbritannien - ausgerüstet. Alle diese Tastaturen verfügen über standardmäßige Zeichensätze. Beispiele siehe Abbildungen 1-3 und 1-4.

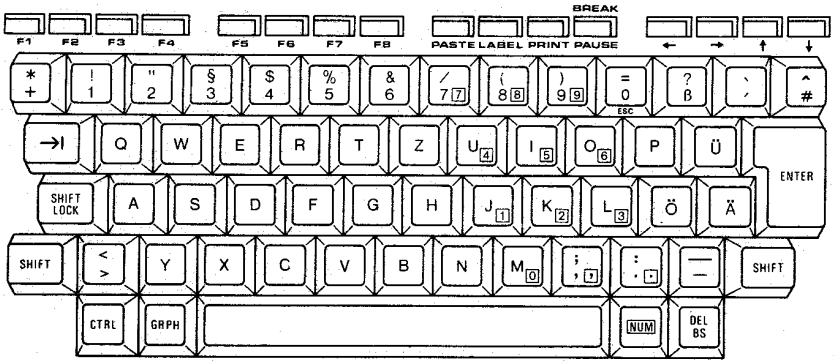


Abbildung 1-3 Deutsche Tastatur

Zusätzlich zu den auf dem Tasten angegebenen Zeichen stehen weitere 47 Zeichen zur Verfügung, wenn man die Taste ⟨**GRPH**⟩ drückt, wie in Abbildung 1-5 dargestellt. Nochmals werden weitere 47 Zeichen generiert, wenn man ⟨**GRPH + SHIFT**⟩ drückt, wie in Abbildung 1-6 dargestellt ist. Die Zeichensätze, die durch ⟨**GRPH**⟩ und ⟨**GRPH + SHIFT**⟩ erzeugt werden, sind in allen nationalen Versionen identisch.

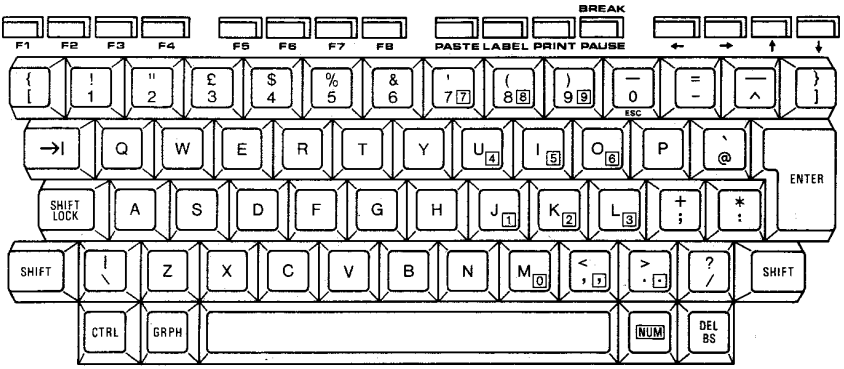


Abbildung 1-4 Englische Tastatur

Beachten Sie, daß alle Tasten mit einer automatischen Wiederholfunktion ausgestattet sind. Dies gilt auch für die Tasten Delete und Backspace (DEL/BS), die Tabulations-Taste (⟶|) und die Tasten für die Cursor-Steuerung.

Die **SHIFT**-Taste hat dieselbe Funktion wie an einer normalen Schreibmaschine - sie wandelt Kleinbuchstaben in Großbuchstaben um und erzeugt anstelle der Ziffern die Sonderzeichen.

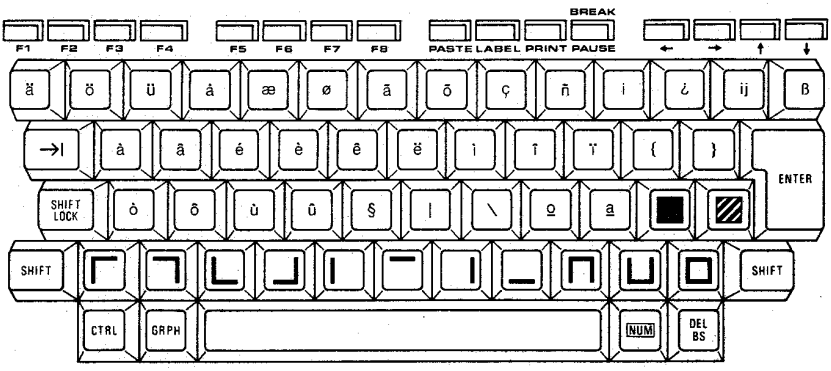


Abbildung 1-5 Tastatur, wenn ⟨**GRPH**⟩ gedrückt wird.

Die Umschalttaste **SHIFT LOCK** bewirkt die dauerhafte Umschaltung auf Großschreibung. Die Umschaltung wird durch erneutes Drücken der **SHIFT LOCK**-Taste wieder aufgehoben.

Die Taste für Tabulation ist durch das Symbol → gekennzeichnet. Sie bewegt den Cursor von einem Tabulatorstop zum nächsten. Diese Tabulationsstops sind in Abständen von acht Zeichen gesetzt und können nicht geändert werden.

CTRL ist eine Taste, mit deren Hilfe Computer spezielle Steuerzeichen eingegeben werden können. Sie wird in Verbindung mit anderen Tasten für bestimmte Funktionen benutzt. Diese werden später noch eingehend besprochen.

GRPH ermöglicht die Erstellung eines weiteren Zeichensatzes, wie schon beschrieben wurde.

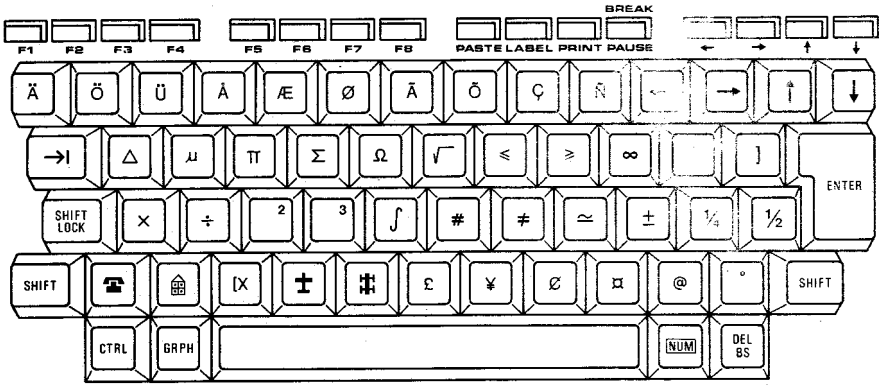


Abbildung 1-6 Die Tastatur, wenn ⟨**GRPH** + **SHIFT**⟩ gedrückt werden.

An der rechten Seite der Tastatur befinden sich 12 Tasten, die zusätzlich zu ihrem normalen Zeichensatz noch die Zahlen 0 - 9 aufweisen, und einen Punkt, der in einem kleinen Quadrat steht, wie in Abbildung 1-7 gezeigt ist. Diese numerische Tastatur wird in Betrieb genommen, indem man **NUM** drückt. Alle anderen Tasten werden außer Funktion gesetzt, wenn die Taste **NUM** gedrückt wurde, und bleiben solange außer Funktion, bis **NUM** wieder gedrückt wird. Wenn Ihre Tastatur blockiert ist und keine Buchstaben, sondern nur Zahlen produziert, liegt es meistens daran, daß Sie die **NUM**-Taste gedrückt haben.

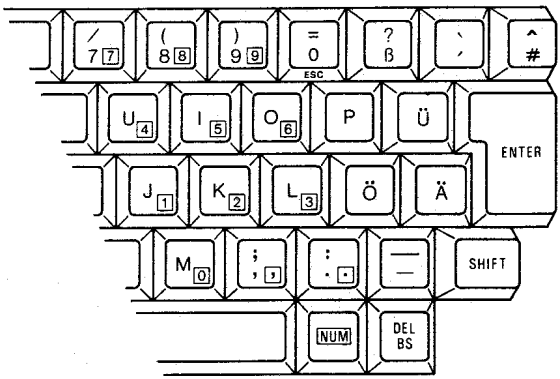


Abbildung 1-7 Die alphanumerische Tastatur auf dem M10

DEL/BS ist die Lösch- und Rücktaste. Wenn diese Taste gedrückt wird, wird das Zeichen, das sich vor dem Cursor befindet, gelöscht. Der Cursor wird gleichzeitig um eine Zeichenbreite zurückgesetzt. Im TEXT-Modus löscht ⟨**DEL/BS + SHIFT**⟩ dasjenige Zeichen, auf dem der Cursor steht, und bewegt den auf den Cursor folgenden Text um ein Zeichen nach links.

Die Taste **ENTER** wird dazu benutzt, ein Programm oder eine TEXT-Datei aus dem Hauptmenü auszuwählen. Wird ein Programm über seinen Programmnamen aufgerufen bzw. der Cursor auf den Programmnamen positioniert, so wird die betreffende Eingabe mit der Taste **ENTER** abgeschlossen. Sie dient ebenso dazu, eine Dateneingabe abzuschließen bzw. den Cursor auf die nachfolgende Zeile zu positionieren.

Direkt unterhalb des Displays befinden sich vier Gruppen von je vier blauen Tasten.

Die ersten zwei Gruppen, bezeichnet als F1 - F8, sind Funktionstasten, die entsprechend ihrem Anwendungsbereich unterschiedliche Funktionen ausüben können. Diese Funktionen werden in den folgenden Kapiteln beschrieben.

Die dritte Gruppe umfaßt die Befehlstasten **PASTE**, **LABEL**, **PRINT** und **PAUSE/BREAK**. Die Funktionen dieser Tasten sind in allen Programmen die gleichen.

- **PASTE** wird für die Funktionen ZERLEGEN, MONTIEREN und Kopieren verwendet. Dieses sind Textverarbeitungsfunktionen. Eine detaillierte Beschreibung hiervon wird in Kapitel 5 gegeben.

- **LABEL** zeigt am Fuß des Bildschirms die Funktionen der Tasten F1 - F8, und zwar für jedes der fünf Anwenderprogramme, das Sie im Moment im Gebrauch haben. Wenn Sie **LABEL** während der Arbeit in einem Programm eine Sekunde lang gedrückt halten, werden diese Definitionen im Bildschirm überblendet.

- **PRINT** wird benutzt, wenn an den M10 ein Drucker angeschlossen ist. Wenn Sie diese Taste drücken, wird der gegenwärtige Display-Inhalt ausgedruckt. ⟨**SHIFT** + **PRINT**⟩ bewirkt, daß der gesamte Inhalt der Datei ausgedruckt wird, wobei dem Anwender speziell die Möglichkeit eingeräumt wird, die Anzahl der Zeichen pro Zeile, die gedruckt werden sollen, zu definieren.

- **PAUSE/BREAK** hat zwei Funktionen. (Wenn diese Taste in der BASIC-Ebene gedrückt wird, hat sie die Funktion einer nicht umgeschalteten Taste und unterbricht die Ausführung eines Programmes.) Im BASIC ist die Funktion '**PAUSE**' aktiviert. Wird diese Taste gedrückt, so wird die Ausführung des gerade laufenden Programms unterbrochen. Der Programmablauf wird wieder aufgenommen, wenn diese Taste wieder gedrückt wird. Die obere Funktion ⟨**BREAK**⟩ wird aktiviert, indem der Benutzer die Tastenkombination aus den Tasten ⟨**SHIFT**⟩ und ⟨**PAUSE/BREAK**⟩ drückt. Diese Umschaltung erlaubt es dem Benutzer, ein Programm anzuhalten oder die normalen Funktionen neu zu starten, wenn das System sich "aufgehängt" hat. Das ist z. B. dann immer der Fall, wenn ein LPRINT-Befehl gegeben wurde, ohne daß ein Drucker an den M10 angeschlossen ist. Im TEXT-Modus hebt das Drücken der Taste ⟨**BREAK**⟩ einen Vor-gang der Art "Select, Find, Load, Save oder PRINT" auf.

Die letzten vier mit Pfeilen versehenen Tasten sind die Cursor-Steuerungstasten. Diese werden im TEXT-Modus zur Steuerung (Positionierung) des Cursors auf dem Bildschirm benutzt. Ihre Funktionen sind in der Abbildung 1-8 zusammengefaßt.

TASTE	CURSOR STEUERUNG
→	Eine Zeichenbreite nach rechts
⟨SHIFT⟩ + →	Bis zum Beginn des nächsten Wortes
⟨CTRL⟩ + →	Bis zum Ende der aktuellen Zeile
←	Ein Zeichen nach links
⟨SHIFT⟩ + ←	Bis zum Beginn des letzten Wortes (oder des aktuellen Wortes)
⟨CTRL⟩ + ←	Bis zum Beginn der aktuellen Zeile
↑	Eine Zeile höher
⟨SHIFT⟩ + ↑	Bis zur obersten Zeile des Bildschirms
⟨CTRL⟩ + ↑	Bis zum Beginn der Datei
↓	Eine Zeile nach unten
⟨SHIFT⟩ + ↓	Bis zum unteren Rand des Bildschirms
⟨CTRL⟩ + ↓	Bis zum Ende der Datei.

Abbildung 1-8 Cursor-Steuerungstasten

Zusätzlich zu den in dieser Tabelle angegebenen Funktionen sind die Cursor-Steuerungstasten mit einer Dauerfunktionseinrichtung ausgestattet, die bei einem längeren Drücken dieser Taste aktiviert wird. Wenn Sie die Taste mit dem Pfeil nach rechts gedrückt halten, wandert der Cursor von links nach rechts und von Zeile

zu Zeile bis ans Ende der Datei. Wenn Sie die Taste mit dem Pfeil nach links gedrückt halten, bewirkt das das Gegenteil. Wenn Sie die Taste mit dem Pfeil nach unten gedrückt halten, wandert der Cursor von oben nach unten durch die gesamte Datei und rollt den Bildschirm auf. Die gegenteilige Funktion kann natürlich durch Drücken der Taste mit dem Pfeil nach oben erzielt werden.

Der Bildschirm, auf dem die Zeichen erscheinen, besteht aus einer Flüssigkristallanzeige (LCD). Er besteht aus 8 Zeilen zu je 40 Zeichen, und jedes Zeichen wird in einer 6 x 8 Punkt Matrix dargestellt. Die Spalte auf der rechten Seite wird im TEXT-Modus automatisch freigelassen, um den Leerraum zwischen Wörtern der folgenden Zeile anzugeben.

Um den bestmöglichen Blickwinkel zu erreichen, kann der Bildschirm von Hand von der horizontalen Position bis zu einem Winkel von ungefähr 30 Grad verstellt werden. Der Bildschirmkontrast kann durch einen Drehknopf an der rechten Seite justiert werden.

Bis zu acht Zeilen Text können auf dem Bildschirm gleichzeitig angezeigt werden. Wenn man sich in dem TEXT-Modus befindet, kann der Bildschirm durch die Cursor-Steuerungstasten von oben oder von unten gerollt (SCROLLING) werden. In diesem Modus gibt es eine automatische Zeilenschaltung, so daß ein Wort automatisch in die nächste Zeile transportiert wird, wenn es am Zeilenende zu lang ist. Auf diese Weise vermeidet man unbeabsichtigte Worttrennungen.

Obwohl der M10 Bildschirm nur 40 Zeichen in einer Zeile zeigt, ist es möglich, bis zu 132 Zeichen pro Zeile auf einem Drucker ausdrucken zu lassen.

DIE STECKER AUF DER RÜCKSEITE

An der Rückseite des M10 gibt es eine Anzahl von Steckkontakten für periphere Anschlüsse, die in Abbildung 1-9 dargestellt werden.

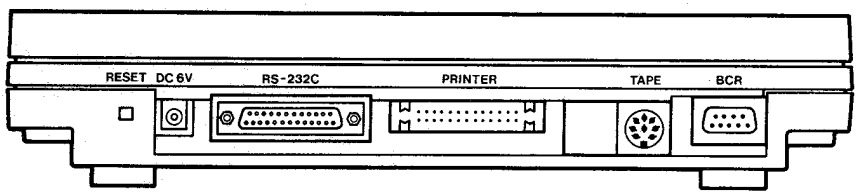


Abbildung 1-9 Steckerverbindungen an der Rückseite

Von links nach rechts sind es die folgenden Verbindungen:

RESET

Wie der Name schon sagt, ist dieser eingelassene Knopf dazu bestimmt, den Computer wieder neu in Gang zu setzen, wenn er sich "aufgehängt" hat, und wenn alle Funktionen blockiert sind. Dieses ist bei Einsatz eines der eingebauten Anwenderprogramme unwahrscheinlich, aber solch ein Fall könnte doch einmal eintreten, wenn Sie eines Ihrer eigenen Programme laufen lassen.

DC 6V

Diese Buchse nimmt den Stecker am Ende des Netzteils auf, wenn der M10 an die Netzspannung angeschlossen ist.

RS-232C (V24)

Dies ist eine Schnittstelle entsprechend dem EIA Standard mit Ausnahme der Data Carrier Detect (DCD) Line, welche nicht geschaltet ist. Typische Anwendungen der RS-232C Schnittstelle sind:

- den M10 an ein externes Modem oder einen Modem/Akustik-Koppler anzuschließen.

- den M10 an ein Gerät anzuschließen, das als Datenterminal fungiert, z. B. die elektronische Olivetti Schreibmaschine ET351, oder den Typenraddrucker Olivetti PR430.

- um den M10 mit einem anderen M10 oder einem anderen Personal Computer zu verbinden.

DRUCKER

Diese parallele Schnittstelle ist der Ausgang von dem M10 zu einem Drucker oder einem Mikroplotter.

PHONE

Diese Verbindung existiert nur bei dem M10 MODEM. Sie stellt das Bindeglied her zwischen dem M10 und einer Telefonleitung, entweder zum Austausch von Informationen mit einem Gastcomputer oder zum automatischen Wählen einer Nummer (siehe Kapitel 8).

TAPE

Wenn ein Cassettenrecorder zum Abspeichern der Daten von dem M10 (wie empfohlen) vorgesehen ist, wird der Recorder mit dieser Buchse verbunden. Mehr darüber finden Sie in Kapitel 9.

BCR

Diese Buchse liefert die Verbindungsmöglichkeit zwischen dem M10 und einem Bar-Code-Leser. Sie ist speziell zum Anschluß des Hewlett-Packards HEDS-3050 oder HEDS-3000 Bar-Code-Leser vorgesehen. Damit kann dann z. B. Lagerhaltung und allgemeine Lagerkontrolle einfach erledigt werden.

DIE UNTERSEITE DES M10

An der Unterseite sieht der M10 Computer aus wie in der Abbildung 1-10 dargestellt.

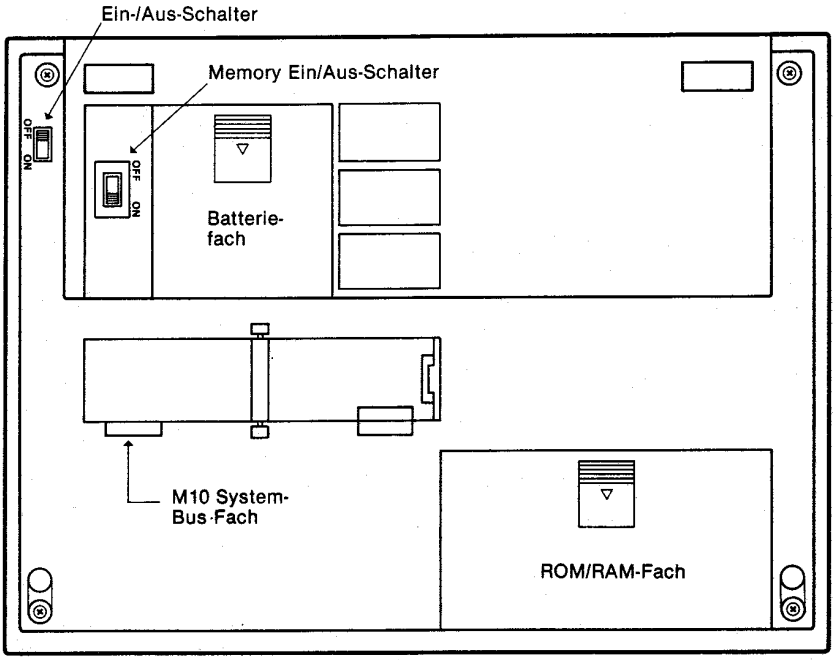


Abbildung 1-10 Die Unterseite

Die einzelnen Bereiche haben die folgenden Funktionen:

1. Ein-/Aus (On/Off)-Schalter

Mit diesem Schalter wird der Computer ein- und ausgeschaltet.

2. Memory-Ein-/Aus (On/Off)-Schalter

Dieser Schalter versorgt den Arbeitsspeicher mit Strom, auch wenn der Computer ausgeschaltet ist. Bei Lieferung des M10 ist dieser Schalter in der Stellung OFF. Während des Startvorganges wird er in die Position ON gestellt und wird in dieser Position bleiben, mit Ausnahme des Falles, daß der Computer für eine längere Zeit nicht benutzt wird.

Es ist sehr wichtig, sich klarzumachen, daß in der Stellung OFF im Arbeitsspeicher abgelegte Daten ver-

lorengehen. Alle wichtigen Dateien sollten auf Kassette abgespeichert werden, bevor dieser Schalter ausgeschaltet wird. Weiterhin arbeitet der Computer nicht, wenn dieser Schalter ausgeschaltet ist. Er befindet sich in einer versenkten Position, um sicherzugehen, daß er nicht aus Versehen betätigt wird.

Abhängig von der installierten Speichererweiterung hält die wiederaufladbare Nickel-Cadmium-Batterie den Speicherinhalt zwischen 8 bis 30 Tage aufrecht, wenn der M10 ausgeschaltet wird. Es ist am besten, den Computer regelmäßig alle paar Tage zu benutzen, um die Batterie aufgeladen zu halten.

3. Batteriefach

Der M10 wird durch vier 1,5 V AA Trockenzellen, die in dieses Fach eingelegt werden, mit Strom versorgt. Sie sollten entsprechend der Abbildung 2-1 eingelegt werden. Diese Zellen haben eine Lebensdauer von ungefähr 20 Stunden, unabhängig von der installierten Speicherkapazität. Wenn die Batterien leer sind, leuchtet die Anzeige mit der Bezeichnung 'Battery Low' an der Frontseite auf. Sie haben dann noch etwa 20 Minuten Zeit, bis die Batterie keinen Strom mehr liefert. Schalten Sie dann den Computer aus, und ersetzen Sie die Batterie.

4. ROM/RAM Fach

Dieses Fach bietet Raum für eine Erweiterung sowohl des ROM als auch des RAM. Der M10 hat 32KByte ROM als Standardausrüstung und kann mit einem einzelnen Extrablock bis auf 64k aufgerüstet werden. Die Standard RAM Kapazität ist 8k Byte, die bis auf 32k in Schritten von 8k aufgerüstet werden kann.

5. M10 System-Bus Fach

Dieses Fach ist abgedeckt und enthält einen Stecker mit 40 Kontakten für den M10 System-Bus.

2. DAS EINSCHALTEN DES M10

Der M10 Computer bezieht seine Stromversorgung aus einer 6V DC, die von einer Batterie oder vom Netz mit Hilfe eines Netzteils geliefert wird. Dieses Netzteil ist ein Zubehörteil und kann von Olivetti bezogen werden. Dabei stehen verschiedene Modelle zur Verfügung, alle mit einer Ausgangsspannung von 6 Volt, jedoch für unterschiedliche Netzspannungen von 240 V, 220 V oder 110 V (für Großbritannien, Europa und USA).

DAS EINLEGEN DER BATTERIE

Da der M10 ein tragbares Gerät ist, ist er für vier 1,5 V DC Trockenzellen, Größe AA, vorgesehen. Das Batteriefach befindet sich an der Unterseite des Chassis, wie in Abbildung 1-10 dargestellt ist. Abbildung 2-1 zeigt, wie die Batterien eingelegt werden und wohin der Plus- bzw. Minuspol zeigen muß. Die Batterien haben eine durchschnittliche Lebensdauer von ungefähr 20 Stunden, unabhängig davon, ob eine Speichererweiterung installiert wurde. Wenn die Batterien zu schwach werden, leuchtet die "Battery Low"-Lampe an der Frontseite des Gerätes auf. Wenn dieser Fall eintritt, bleiben noch ungefähr 20 Minuten zur Bedienung, bevor die Spannung nicht mehr ausreicht, so daß die Batterien sofort gewechselt werden sollten.

Zusätzlich zu der 6 Volt Batterie befindet sich in dem Gerät eine Nickel-Cadmium-Zelle, die für den Arbeitsspeicher Spannung liefert, auch wenn der M10 ausgeschaltet ist. Diese Zelle lädt sich wieder auf, wenn der Computer wieder in Betrieb ist, und kann im aufgeladenen Zustand den Speicherinhalt zwischen 8 und 30 Tagen aufrechterhalten, abhängig von der installierten Speichererweiterung. Diese Zelle hat eine durchschnittliche Lebensdauer von ungefähr 2 Jahren. Nach dieser Zeit sollten Sie sie bei Ihrem Olivetti-Händler austauschen lassen.

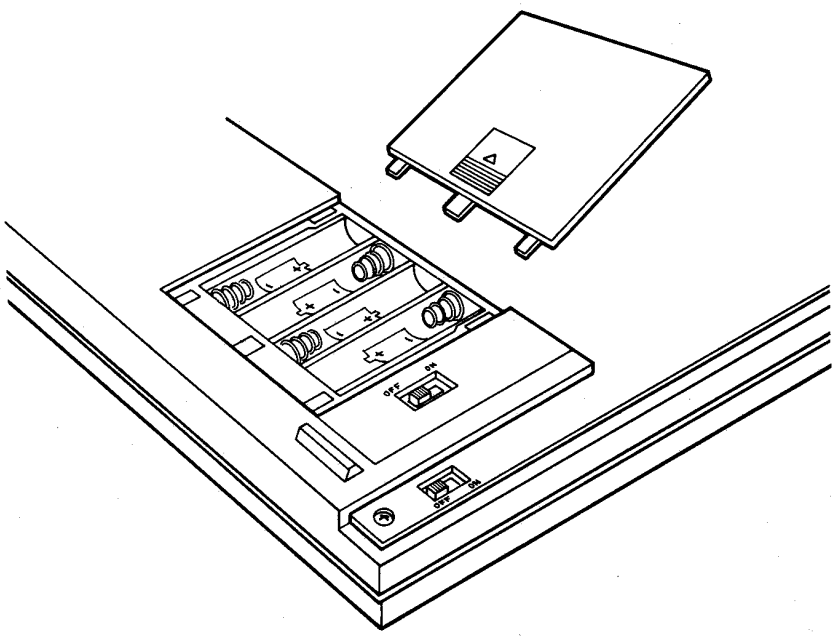


Abbildung 2-1 Das Einlegen der Batterien

DAS ERSTE EINSCHALTEN

1. Entfernen Sie die Abdeckung des Batteriefachs an der Unterseite des M10 Chassis.

2. Legen Sie vier 2,5 V AA Trockenbatterien ein, wie in der Abbildung 2-1 gezeigt wird. Sie können auch das Ende des AC-Netzteils mit der Buchse verbinden, die die Bezeichnung DC 6 V an der Rückseite des M10 trägt. Dann stecken Sie den Stecker des Netzteils in eine Netzsteckdose.

   Es ist sehr wichtig, daß der M10 und alle Peripheriegeräte ausgeschaltet sind, bevor das Netzteil mit dem M10 verbunden wird.

3. Schalten Sie den ON/OFF-Schalter in die Position ON (an der Unterseite, siehe Abbildung 1-10).

4. Bringen Sie den Speicherschalter in die Position ON (an der Unterseite, siehe Abbildung 1-10).

Wenn Sie diesen letzten Schritt nicht tun, kann der Computer nicht arbeiten.

Wenn Sie den M10 ausschalten, müssen Sie zuerst alle Peripheriegeräte, die mit ihm in Verbindung stehen (z. B. ein Drucker), ausschalten und dann erst den M10. Schalten Sie nicht den Memory-Schalter ab, da dieses den Inhalt Ihres Arbeitsspeichers zerstört.

Wenn diese Startprozedur einmal durchgeführt ist, wird der M10 nur noch durch den Ein-/Ausschalter an der Unterseite in Betrieb genommen.

Achten Sie darauf, daß der M10 sich selbst nach etwa 10 Minuten ausschaltet, wenn keine Eingaben vorgenommen werden. Dieses dient dem Schutz der Batterie. Um den Computer wieder in Betrieb zu nehmen, schalten Sie ihn kurz aus und dann wieder an. Dieses automatische Ausschalten kann mit Hilfe des BASIC Befehls POWER CONT außer Kraft gesetzt werden. Es kann aber auch die Zeitspanne, nach der sich der M10 automatisch ausschaltet, durch den BASIC Befehl POWER modifiziert werden. Einzelheiten dieser Befehle werden im BASIC-Handbuch zum M10 erläutert (Band 2).

Beachten Sie bitte, daß Sie nach dem Einschalten unter Umständen im Bildschirm das angezeigt bekommen, was beim letzten Ausschalten dort stand. Dann können Sie entweder einfach weiterarbeiten oder durch Drücken von **F8** zum Menü zurückkehren.

Die Funktion des Menüs wird im nächsten Kapitel beschrieben.

3. DAS HAUPTMENÜ

Wenn der M10 zum ersten Mal eingeschaltet wird, sieht der Bildschirminhalt aus wie in Abbildung 3-1. Dieses ist das Hauptmenü des Computers.

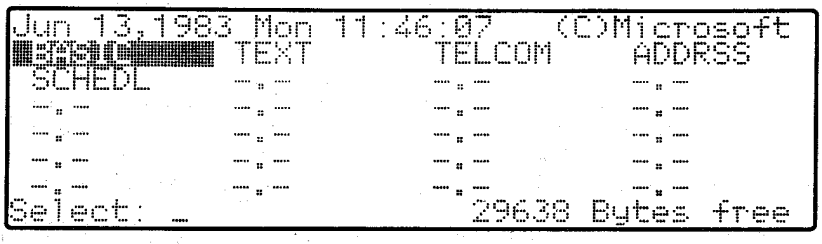


Abbildung 3-1 Das Hauptmenü

Wie man sieht, enthält die Anzeige 8 Textzeilen. Das dunkle Rechteck, das über BASIC positioniert ist, ist der Cursor, der durch die Corsor-Steuerungstasten bewegt werden kann, wie in Kapitel 1 beschrieben. Das Wort, auf dem der Cursor steht, erscheint "negativ". Die Bildschirmhelligkeit kann mit dem Drehknopf an der rechten Seite verändert werden.

In der ersten Zeile des Bildschirms wird das Datum und die Uhrzeit angezeigt. Bei dem ersten Start wird folgendes angezeigt:

Jan 01, 1900 Sun 00:00:00 (C) Microsoft

Die nächsten sechs Zeilen sind reserviert für eine Auflistung der Programme und Dateien, die in dem Computer (Arbeitsspeicher = RAM) gespeichert sind. Zu Anfang sind nur die folgenden fünf Anwendungen bzw. Programme gespeichert:

- BASIC
- TEXT
- TELCOM
- ADDRSS

- SCHEDL

Im Inhaltsverzeichnis ist für weitere 19 Namen Platz vorhanden. Diese erscheinen nach dem Anlegen bzw. Speichern im Menu.

Die Fußzeile zeigt:

Select : _ 29638 Bytes free

Die Anzeige 'Select' ermöglicht es Ihnen, ein Programm oder ein anderes File anzugeben, um darauf zuzugreifen. Dieses kann auch geschehen, indem Sie den Cursor auf den Filenamen im Menu bringen und ⟨**ENTER**⟩ drücken.

Die Angabe **29638 Bytes free** bezieht sich auf den Ihnen zur Verfügung stehenden Arbeitsspeicher (RAM). Die Anzahl der beim Start angezeigten Bytes hängt von der Anzahl der installierten RAM-Bausteine ab. Die unterschiedlichen Werte sind in der folgenden Tabelle angegeben:

Speicherkapazität	freie Bytes
32KB	29638
24KB	21446
16KB	13254
8KB	5062

Diese Werte vermindern sich natürlich, wenn Sie den Arbeitsspeicher des Computers mit Daten belegen.

Beim Start steht der Cursor auf BASIC. Er kann mit Hilfe der Cursor-Steuerungstasten in jede andere Position gebracht werden. Um den Anfang unserer Erklärung nicht zu komplizieren, gehen wir jedoch zunächst nur auf BASIC, TEXT und TELCOM ein. Mit den Programmen ADDRSS und SCHEDL beschäftigen wir uns dann in den Kapiteln 6 und 7.

SETZEN DER UHRZEIT

Wie schon oben erwähnt, erscheinen in der ersten Zeile des Hauptmenus die Angaben für das Datum, den Tag und die Uhrzeit. Wenn Sie den Computer zum erstenmal einschalten, wird:

Jan 01, 1900 Sun 00:00:00

angezeigt, und die Uhrzeit läuft dann automatisch weiter.

Wie Sie nun die richtige Ausgangszeit genau einstellen, wird im folgenden Abschnitt beschrieben.

NEUFESTSETZUNG DER UHRZEIT

Bringen Sie den Cursor über BASIC und drücken Sie ⟨**ENTER**⟩. Der Cursor erscheint als blinkendes Rechteck direkt unter der BASIC-Anzeige **Ok**.

Geben Sie nun ein:

TIME$="Stunde:Minute:Sekunde"

und drücken Sie ⟨**ENTER**⟩. Für die Platzhalter 'Stunde', 'Minute' und 'Sekunde' sind folgende Werte zulässig:

'Stunde' als zweistellige Zahl von 00 bis 23

'Minute' als zweistellige Zahl von 00 bis 59

'Sekunde' als zweistellige Zahl von 00 bis 59

genauso wie Sie es bei dem Einstellen einer Digitaluhr machen. Die Anführungszeichen mit Doppelpunkten müssen mit eingegeben werden.

In dem Moment, in dem Sie ⟨**ENTER**⟩ drücken, wird der Computer Ihre Zeitangabe als Anfangszeit akzeptieren und die Zeit von diesem Wert an weiterlaufen lassen.

Wenn Sie die Zeitangabe nicht richtig durchführen, erscheint eine Fehlermeldung auf dem Bildschirm. Um den Wert, den Sie angegeben haben, zu prüfen, geben Sie ein:

PRINT TIME$

und drücken Sie ⟨**ENTER**⟩. Die von Ihnen angegebene Zeit erscheint dann auf dem Bildschirm (ohne Anführungszeichen).

NEUFESTSETZUNG DES DATUMS

Geben Sie das Datum auf dem Bildschirm in der folgenden Form an:

Date$="Tag/Monat/Jahr"

und drücken Sie ⟨**ENTER**⟩. Für die Platzhalter 'Tag', 'Monat' und 'Jahr' sind folgende Wert zulässig:

'Monat' als zweistellige Zahl von 01 bis 12

'Tag' als zweistellige Zahl von 01 bis 31

'Jahr' als zweistellige Zahl von 00 bis 99

Die Anführungszeichen und das Zeichen / müssen mit eingegeben werden.

Bei einer Fehlermeldung gehen Sie analog zur Zeiteingabe vor (PRINT DATE$ statt PRINT TIME$).

Wenn Sie z. B. eingegeben haben:

DATE$="14/06/83", entspricht das dem 14. Juni 1983.

Wenn Sie den Befehl PRINT DATE$ geben, wird das von Ihnen eingegebene Datum auf dem Bildschirm angezeigt.

NEUFESTSETZUNG DES TAGES

Geben Sie den Tag auf dem Bildschirm wie folgt an:

DAY$="Tag"

und drücken Sie die Taste ⟨**ENTER**⟩. Für den Platzhalter 'Tag' müssen Sie eine der in der folgenden Tabelle aufgeführten Abkürzungen eingeben:

Monday (Montag)	- Mon
Tuesday (Dienstag)	- Tue
Wednesday (Mittwoch)	- Wed
Thursday (Donnerstag)	- Thu
Friday (Freitag)	- Fri
Saturday (Samstag)	- Sat
Sunday (Sonntag)	- Sun

Die Anführungszeichen müssen mit eingegeben werden.

Wiederum können die eingegebenen Werte kontrolliert werden, wenn Sie PRINT DAY$ tippen und ⟨**ENTER**⟩ drücken.

Um zum Hauptmenu zurückzukehren, drücken Sie die Funktionstaste ⟨**F8**⟩ . Der Tag, das Datum und die Uhrzeit, die Sie eingegeben haben, erscheinen nun in der ersten Zeile des Menus.

4. DER BASIC-INTERPRETER

Der BASIC-Interpreter ist eines der fünf eingebauten Programme, mit dem der M10 ausgerüstet ist. Die meisten Informationen, die Sie benötigen, um seine Möglichkeiten voll auszuschöpfen, sind in Band 2, dem BASIC-Handbuch, beschrieben. Bei allen Fragen, die sich auf das Programmieren, die Syntax und die Struktur von BASIC beziehen, sollte der Benutzer Gebrauch von diesem Nachschlageverzeichnis machen, in dem er auch eine zusammenfassende Liste aller BASIC-Befehle mit Erklärung ihrer Funktion und ihrer Anwendung findet. Da jedoch BASIC das Herzstück des M10 ist, werden in diesem Kapitel einige Hauptaspekte von BASIC behandelt, um den Umgang mit dem M10 ausführlich zu beschreiben.

BASIC steht für "Beginners All-purpose Symbolic Instruction Code" und ist die Programmiersprache des M10. BASIC ist die an Mikrocomputern weitverbreitete Programmiersprache. BASIC existiert in unterschiedlichen Versionen. Die am M10 verwendete Version ist eine Variante von Microsoft-BASIC, eines besonders leistungsfähigen BASIC-"Dialekts". Der BASIC-Interpreter dient dazu, Ihnen die Erstellung eigener Progmme in BASIC zu ermöglichen, diese zu speichern und ablaufen zu lassen.

DER BILDSCHIRM UND DIE VORGEHENSWEISE IN BASIC

Um BASIC aus dem Hauptmenü zu wählen, positionieren Sie den Cursor über den Namen BASIC und drücken ⟨**ENTER**⟩. Der Bildschirm erscheint dann wie in Abbildung 4-1 dargestellt. Das blinkende Rechteck unmittelbar unter der BASIC-Anzeige **Ok** ist der Cursor.

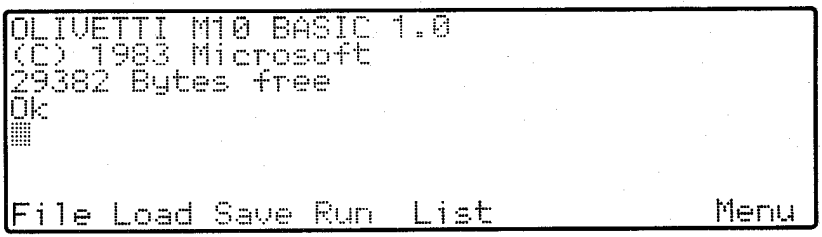


Abbildung 4-1 Bildschirm bei Eingabe von BASIC

Es gibt drei Arten, in BASIC vorzugehen:

- Direct

- Execute

- Text

Es ist nicht nötig, daß der Bediener aus den ersten beiden Möglichkeiten eine Auswahl trifft; der Computer wählt selber diejenige aus, die aufgrund der auszuführenden Operation nötig wird.

DIRECT-MODUS

Dieses ist der automatisch hergestellte Modus, wenn die BASIC-Anzeige **Ok** auf dem Bildschirm erscheint. Der Direct-Modus wird benutzt, um Programme einzugeben oder Befehlszeilen unmittelbar zur Ausführung zu bringen (siehe die Erklärung von Befehlszeilen unten). Wenn ⟨**ENTER**⟩ gedrückt wird, um ein Programm in Gang zu setzen, wechselt der M10 in den Execute-Modus und kehrt in den Direct-Modus zurück, nachdem die Abarbeitung beendet ist.

EXECUTE-MODUS

Der M10 befindet sich in diesem Modus, wenn er gerade ein Programm ablaufen läßt oder wenn eine Befehlszeile sofort ausgeführt wird. Wenn dieser Vorgang beendet ist, kehrt der Computer in den Direct-Modus zurück, und die **Ok**-Anzeige erscheint wieder auf dem Bildschirm.

TEXT-MODUS

Der Befehl EDIT (mit anschließender Betätigung von ⟨**ENTER**⟩) ruft diesen Modus auf, wenn die BASIC-Anzeige Ok auf dem Bildschirm erscheint. Dieser Modus wird benutzt, um ein Programm zu ändern. Wenn der M10 sich in diesem Modus befindet, funktionieren die Cursor-Steuerungstasten, wie in Kapitel 5 beschrieben wird. Um diesen TEXT-Modus zu verlassen, drücken Sie ⟨**F8**⟩.

Im folgenden ist das Konzept für Befehls- oder Zeichenzeilen dargestellt:

- Eine logische Zeile ist eine Zeile von Befehlen oder Anweisungen, die im allgemeinen durch Drücken der Taste ⟨**ENTER**⟩ abgeschlossen wird. Sie kann bis zu 255 Zeichen (Zahlen, Buchstaben, Leertasten oder Sonderzeichen) lang sein.

- Eine physische Zeile ist eine Textzeile auf dem Bildschirm und kann deshalb nicht mehr als 40 Zeichen lang sein.

- Eine Befehlszeile zur sofortigen Eingabe ist ein Befehl oder eine Serie von Befehlen, die sofort von dem Computer ausgeführt werden, wenn man die Taste ⟨**ENTER**⟩ drückt, z. B.

PRINT TIME$

Werden mehrere Befehle oder Anweisungen in einer Befehlszeile zusammengefaßt, muß dazwischen immer ein : (Doppelpunkt) stehen, z. B.

PRINT TIME$:PRINT DATE$

DIE FUNKTIONSTASTEN IN BASIC

Die Tabelle unten faßt den Gebrauch der Funktionstasten **F1 - F8** und **PASTE**, **LABEL**, **PRINT** und **PAUSE/BREAK** zusammen, wenn man im Direct-Modus von BASIC ist:

TASTENNAME	FUNKTION
F1 File	Listet alle Programme und Dateien auf dem Bildschirm auf.
F2 Load	Wird benutzt, um ein BASIC-Programm von einem externen Speicher (z. B. einem Kassettenrekorder) oder aus dem RAM zu laden
F3 Save	Speichert das augenblickliche Programm auf einen externen Speicher wie z. B. einem Kassettenrekorder oder in RAM
F4 Run	Bringt das gerade im RAM befindliche Programm zum Laufen

F5 List	Gibt alle Befehlszeilen des gerade im RAM befindlichen Programms, beginnend mit der ersten, auf den Bildschirm aus
F6	Nicht in Gebrauch
F7	Nicht in Gebrauch
F8 Menu	Kehrt zum Hauptmenü zurück
PASTE	Fügt den Inhalt des PASTE-Puffers an der Position des Cursors ein
LABEL	Gibt die Funktion von **F1** bis **F8** auf dem Bildschirm aus
PAUSE	Hält das laufende Programm an (Pausenfunktion); wird erneut **PAUSE** gedrückt, läuft das Programm weiter
BREAK oder **SHIFT** + **PAUSE**	Beendet das Programm vorzeitig (auch z. B. eine Ausgabe durch **LIST**-Funktion)

Abbildung 4-2 Die Funktionstasten im Direct-Modus

Die Funktionstasten sind frei belegbar, d. h., der Anwender kann ihnen besondere, von ihm selbst definierte Funktionen übertragen. Einzelheiten dieser Prozedur werden im Band 2 gegeben, welcher sich ausschließlich mit BASIC beschäftigt.

Folgende Standardbedeutungen sind nach erstmaligem Einschalten im TEXT-Modus gültig:

TASTENNAME	FUNKTION
F1 Find	Erlaubt es Ihnen, eine Zeichenfolge zu spezifizieren und ihn innerhalb des zu ändernden Programms zu suchen
F2 Load	Lädt ein Programm von einem Kassettenrekorder
F3 Save	Speichert ein Programm auf einen Kassettenrekorder
F4	Nicht in Gebrauch
F5 Copy	Speichert eine ausgewählte Zeichenfolge im PASTE-Puffer, um ihn zu kopieren und an die vom Cursor angegebene spezielle Position zu placieren
F6 Cut	Speichert eine ausgewählte Zeichenfolge in den PASTE-Puffer und entfernt das Original aus der Datei
F7 Sel	Ermöglicht es Ihnen, ein String für die Funktionen Cut, Copy und Paste auszuwählen
F8 Exit	Verlassen des TEXT-Modus
PASTE	Fügt den Inhalt des PASTE-Puffers an diejenige Stelle ein, an der der Cursor steht
LABEL	Zeigt die Funktionen der Tasten **F1** bis **F8** auf dem Bildschirm an

Abbildung 4-3 Funktionstasten im TEXT-Modus

5. DAS TEXTVERARBEITUNGSPROGRAMM

Das zweite der fünf eingebauten Programme, die in dem Menü aufgelistet werden, ist das TEXT-Programm. Mit Hilfe dieses Programms können Sie beliebige Texte eingeben, modifizieren und abspeichern. Zur schnellen und einfachen Handhabung bietet Ihnen das Programm viele hilfreiche, arbeits- und zeitsparende Textverarbeitungshilfen an. Sie können Textstellen suchen lassen, jederzeit beliebige Zeichen löschen oder einfügen, ganze Absätze innerhalb des Textes löschen oder an eine andere Stelle kopieren.

Den ersten Kontakt mit der Erstellung von TEXT-Dateien werden Sie bei der Anwendung der Programme ADDRSS und SCHEDL haben, für die Sie die TEXT-Dateien ADRS.DO und NOTE.DO erstellen.

Das Programm TEXT starten Sie, indem Sie im Hauptmenü den Cursor über den Namen Text positionieren und ⟨**ENTER**⟩ drücken. Sie können aber auch den Befehl

TEXT ⟨**ENTER**⟩

über die Tastatur geben. Ihre Eingabe erscheint dann am Fuß des Bildschirms unmittelbar rechts von Select:. Nachdem Sie das TEXT-Programm aufgerufen haben, sieht der Bildschirm aus wie in Abbildung 5-1 dargestellt.

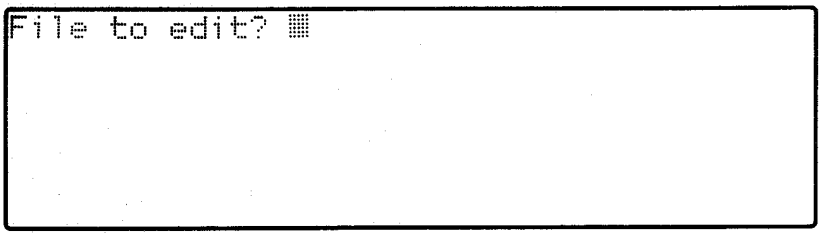


Abbildung 5-1 Der Bildschirm für das TEXT-Programm

DATEINAMEN IM TEXT-PROGRAMM

Als Antwort erscheint auf dem Bildschirm die Frage:

File to edit?

(Welche TEXT-Datei möchten Sie ändern oder neuerfassen?)

Wollen Sie den Inhalt einer bereits erstellten Datei ändern, so geben Sie den Dateinamen ein und drücken ⟨**ENTER**⟩ . Der vorhandene Text wird geladen, und Sie können ihn bearbeiten.

Möchten Sie eine Datei neu erstellen, dann geben Sie ebenfalls den Dateinamen ein und drücken ⟨**ENTER**⟩ . Der Bildschirm wird gelöscht; der Cursor erscheint im oberen linken Bildschirmeck, und Sie können mit der Eingabe Ihres Textes beginnen.

Beachten Sie, daß MIO-Dateinamenmaximal aus 6 Zeichen bestehen dürfen, wobei folgende Einschränkungen zu beachten sind:

- Ein Dateiname darf nicht mit einer Nummer beginnen
- Die folgenden Symbole werden nicht als erste Zeichen in einem Dateinamen akzeptiert:

  ! " # $ % & ' *

  ( ) + * : , . /

- Der Punkt (.) und der Doppelpunkt (:) dürfen in einem Dateinamen normalerweise gar nicht verwendet werden. Die einzige Ausnahme ist, wenn der Punkt rechts ans Ende des Dateinamens gesetzt wird. In diesem Falle ist er Bestandteil des Typkennzeichens (siehe Beispiel unten).
- Der MIO wandelt alle eingegebenen Kleinbuchstaben in Großbuchstaben um.

Wenn Sie den Dateinamen eingegeben haben, drücken Sie ⟨**ENTER**⟩ . Wenn bei Vergabe des Dateinamens irgendeine der angegebenen Einschränkungen nicht beachtet wurde,

gibt der M10 ein Warnzeichen "Biep" von sich, und es erscheint die Anzeige

File to edit

in der nächsten Zeile noch einmal. Wenn der Dateiname akzeptiert wurde, wird der Bildschirm gelöscht, und der Cursor erscheint in der linken oberen Ecke. Er weist einen nach links zeigenden Pfeil auf, der den Textanfang angibt.

Dateien, die unter TEXT erstellt werden, werden automatisch mit dem Typkennzeichen .DO am Ende versehen. Es ist nicht nötig, daß der Benutzer das selbst macht. Wenn Sie z. B. den Dateinamen RED angeben, wird die Datei automatisch als RED.DO im Hauptmenü angezeigt werden. Das folgende Beispiel gibt Ihnen einige Muster zulässiger und unzulässiger Dateinamen.

Eingegebener Filename	Ausgewiesener Filename
FILE	FILE.DO
file	FILE.DO
fILe	FILE.DO
file.	FILE.DO
F I L E	F I L E.DO
#FILE	Nicht akzeptiert
FILE#	FILE#.DO
FI.LE	Nicht akzeptiert
"FILE"	Nicht akzeptiert
FILE.DO	FILE.DO

Bis zu 19 Dateien können im RAM gleichzeitig gespeichert werden. Sie erscheinen auf dem Bildschirm zusätzlich zu den fünf Anwenderprogrammen. Wenn Sie mehr Dateien brauchen, müssen Sie einige auf einem externen Speicher abspeichern, z. B. einem Kassettenrekorder (siehe Kapitel 9).

Um eine Datei aus dem RAM zu löschen, wählen Sie im Hauptmenü BASIC an und geben nach der Meldung Ok den folgenden Befehl ein:

KILL "Dateiname" ⟨**ENTER**⟩

Es ist sehr wichtig, daß Sie nicht vergessen, die Anführungszeichen hinter und vor dem Dateinamen anzugeben. Genauso wichtig ist es, den vollen Dateinamen einschließlich des Typkennzeichens am Ende anzugeben.

DIE EINGABE VON TEXT IN EINE DATEI

Wenn Sie einem Text eine gültige Dateibezeichnung gegeben haben, können Sie nun den Text über die Tastatur eingeben. Er erscheint auf dem Bildschirm genauso, wie er eingegeben wurde. Wie schon erwähnt, nimmt der Bildschirm bis zu 40 Zeichen pro Zeile auf. Es ist nicht nötig, am Ende der Zeile ⟨**ENTER**⟩ für den Zeilenwechsel einzugeben. Der Cursor läuft automatisch von Zeile zu Zeile und schaltet in die neue Zeile, wo es nötig ist. Um Tippfehler zu verbessern, fahren Sie den Cursor rechts neben das Zeichen, das gelöscht werden soll, und drücken die ⟨**DEL/BS**⟩ - Taste. Wenn Sie die Tasten ⟨**SHIFT** + **DEL/BS**⟩ drücken, wird das Zeichen, auf dem der Cursor steht, gelöscht. Um ein Zeichen oder einen Textblock einzufügen, bringen Sie einfach den Cursor in die gewünschte Position und geben das Einzufügende (z. B. ein Zeichen oder einen ganzen Satz) ein. Der folgende Text wird um eine Position für jedes eingegebene Zeichen nach rechts verschoben. Beachten Sie: Der M10 befindet sich immer im INSERT-Modus. Das heißt, daß alle Zeichen an der Cursor-Position eingefügt werden und nicht die dort stehenden Zeichen durch die eingegebenen ersetzt werden. Um Text an einen bereits bestehenden Text anzuhängen, bewegen Sie den Cursor an das Ende des laufenden Textes und geben den zusätzlichen Text ein.

Der Cursor wird gesteuert durch die vier Cursor-Steuerungstasten. Dies sind die vier mit Pfeilen bezeichneten Tasten an der rechten Seite der Tastatur, unterhalb des Bildschirms. Die Steuerungstasten sind schon in Kapitel 1 beschrieben worden, aber wir wiederholen die Beschreibung hier noch einmal. In Abbildung 5-2 finden Sie eine ausführliche Übersicht über die Tasten.

Textprogramm 5-4

Wie fast alle Tasten des M10 sind auch die Cursor-Steuerungstasten mit einer Wiederholungsfunktion ausgerüstet. Wenn Sie die Taste mit dem Pfeil nach rechts drücken und festhalten, läuft der Cursor von links nach rechts und springt am Ende der Zeile automatisch an den Anfang der neuen Zeile. Wenn der Cursor am rechten Rand angelangt ist, schiebt der M10 automatisch die nächste Zeile in den Bildschirm. Die Linkspfeiltaste funktioniert entsprechend in Gegenrichtung. Wenn Sie die Taste mit dem Pfeil nach oben gedrückt halten, läuft der Cursor an den oberen Rand des Bildschirms und holt die vorhergehenden Zeilen auf den Bildschirm, bis er den Anfang der Datei erreicht. Die Taste mit dem Pfeil nach unten hat die gleiche Funktion, nur in umgekehrter Richtung. Man kann sagen, daß diese beiden Tasten eine Rollfunktion ausführen.

TASTE	CURSOR BEWEGUNG
	→ Ein Zeichen nach rechts
⟨SHIFT⟩ +	→ Bis zum Beginn des nächsten Wortes
⟨CTRL⟩ +	→ Bis zum Ende der aktuellen Zeile
	← Ein Zeichen nach links
⟨SHIFT⟩ +	← Bis zum Beginn des vorhergehenden Wortes (oder des aktuellen Wortes)
⟨CTRL⟩ +	← Bis zum Beginn der aktuellen Zeile
	↑ Eine Zeile nach oben
⟨SHIFT⟩ +	↑ Bis zur ersten Zeile des Bildschirms
⟨CTRL⟩ +	↑ Bis zum Beginn der Datei
	↓ Eine Zeile nach unten

Textprogramm 5-5

⟨SHIFT⟩ +	↓ Bis zum unteren Bildschirmrand
⟨CTRL⟩ +	↓ Bis zum Ende der Datei

Abbildung 5-2 Cursor-Steuerungstasten

FUNKTIONS- UND KOMMANDOTASTEN IM TEXTPROGRAMM

Wie Sie in der Abbildung 5-3 sehen, haben die Tasten **F1** bis **F8** spezielle Funktionen im TEXT-Modus.

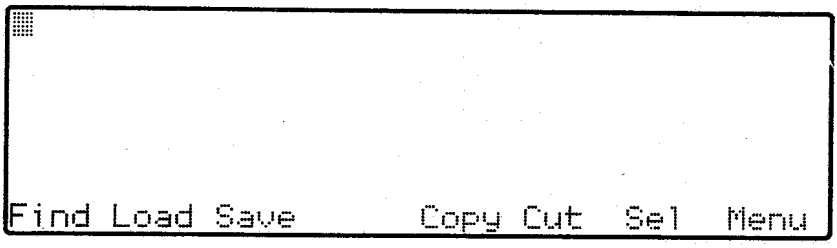


Abbildung 5-3 Bildschirm mit den Zuordnungen der Funktionstasten

Sie müssen ⟨**LABEL**⟩ drücken, um diese Funktionen auf dem Bildschirm anzeigen zu lassen. Der große Bereich von Textverarbeitungsmöglichkeiten, den diese Funktionen bieten, wird in den folgenden Abschnitten dargestellt. Eine Zusammenfassung der Funktionen und der Befehlstasten im TEXT-Modus wird in Abbildung 5-6 gegeben.

SUCHEN EINER ZEICHENFOLGE IM TEXT

Die Such-Funktion wird durch die Funktionstaste **F1** aufgerufen. Wenn Sie die Funktionstaste ⟨**F1**⟩ drücken, erscheint auf der Fußzeile die Meldung:

String:

wobei der Cursor unmittelbar rechts daneben positioniert ist. Geben Sie die Zeichenfolge ein, nach der Sie suchen, und drücken Sie ⟨**ENTER**⟩ . Normalerweise ist das ein Wort oder ein bestimmter Ausdruck. Der Cursor wird sich dann innerhalb Ihres Textes dort befinden, wo diese Zeichenfolge, von der momentanen Cursorposition abwärt, als nächstes gefunden wurde. Danach verschwindet die Meldung String:, und Ihr ursprünglicher Text wird wieder in der Fußzeile abgebildet. Drücken Sie danach wieder die Funktionstaste ⟨**F1**⟩, so erscheint die Meldung String mit Ihrer vorab eingegebenen zu suchenden Zeichenfolge. Wenn Sie jetzt ⟨**ENTER**⟩ drükken, veranlassen Sie den M10, nach weiteren Stellen zu suchen, an denen sich diese Zeichenfolge befindet. Wenn diese Folge in dem ganzen Text nicht mehr aufzufinden ist, zeigt der Computer an:

No match
(Nichts gefunden)

Einige wichtige Punkte sind bei Aufruf dieser Funktion zu beachten.

1. Die zu suchende Zeichenfolge darf nicht mehr als 24 Zeichen, einschließlich aller Leerzeichen, betragen. Wenn Sie mehr Zeichen als diese Maximalgröße eingeben, meldet der Computer ein warnendes "Biep", und der Cursor bewegt sich nicht mehr nach rechts.

2. Es ist sehr wichtig, daß Sie die Leerräume beachten. Wenn Sie die Leerräume falsch angeben, kann die Zeichenfolge nicht gefunden werden.

3. Die Suche beginnt an der augenblicklichen Position des Cursors. Wenn Sie den ganzen Text nach einer speziellen Zeichenkette durchsuchen lassen wollen, sorgen Sie zuerst dafür, daß der Cursor am Beginn der Datei ist, bevor Sie ⟨**F1**⟩ drücken. Rückwärts kann die Datei nicht durchsucht werden.

4. Bei dieser Anwendungsart, ebenso wie bei anderen, macht der M10 keinen Unterschied zwischen Groß- und Kleinbuchstaben und ignoriert demzufolge auch solche Unterschiede, wenn er mit der zu suchenden Zeichenfolge den Text durchgeht.

Wollen Sie eine neue zu suchende Zeichenfolge eingeben, dann drücken Sie **F1**. Die Meldung String: und die aktuelle Suchzeichenfolge erscheinen in der Fußzeile. Wenn Sie nun irgendeine Taste außer **ENTER** drücken, wird die zu suchende Zeichenfolge gelöscht, und Sie können eine neue eingeben.

DIE LOAD-FUNKTION IM TEXT-MODUS

Wenn Sie die Taste ⟨**F2**⟩ drücken, können Sie zusätzlichen Text von einem Kassettenrekorder in eine TEXT-Datei laden. Zu diesem Zweck gehen Sie folgendermaßen vor:

1. Überzeugen Sie sich, daß der M10 mit einem passenden Kassettenrekorder verbunden ist (wegen weiterer Einzelheiten siehe Kapitel 9).

2. Starten Sie das Textverarbeitungsprogramm, indem Sie im Menü eine TEXT-Datei anwählen (wenn nötig, erstellen Sie eine neue Datei dafür).

3. Drücken Sie ⟨**F2**⟩. Dieses ruft die folgende Anzeige hervor

   Load from:

4. Als Antwort auf diese Anzeige geben Sie den Datei-Namen ein und drücken ⟨**ENTER**⟩. Während der Computer nach dieser angegebenen Datei sucht, gibt er einen hohen Ton von sich. Wenn er sie gefunden hat, erscheint die folgende Anzeige auf dem Bildschirm:

   FOUND: filename

   "Filename" steht für den Namen, der in dem vorhergehenden Schritt eingegeben worden ist. Wenn es mehr als eine Datei auf der Kassette gibt, zeigt der M10 jedesmal, wenn er bei der Suche auf eine Datei stößt, auf dem Bildschirm die folgende Anzeige:

   SKIP: filename

Wenn Sie angeben, daß die Information von Kassette in eine Datei geladen werden soll, die schon Text enthält, wird der Inhalt der Datei auf der Kassette an die TEXT-Datei, die Sie im Menü angesprochen haben, angehängt.

Weitere Informationen über die Benutzung eines Kassettenrekorders in Verbindung mit dem M10 finden Sie in Kapitel 9.

DIE SAVE-FUNKTION IM TEXT-MODUS

Wenn Sie eine Text-Datei erstellt oder bearbeitet haben und diese auf Kassette speichern wollen, müssen Sie die Save-Funktion benutzen (**F3**). Hierzu gehen Sie folgendermaßen vor:

1. Überzeugen Sie sich, daß ein passender Kassettenrekorder mit dem M10 verbunden ist (siehe Kapitel 9).

2. Drücken Sie die Funktionstaste ⟨**F3**⟩. Dieses ruft in der Fußzeile des Bildschirms die Anzeige hervor:

   Save to:

3. Wählen Sie einen Dateinamen von nicht mehr als 6 Zeichen für die Kassetten-Datei, in die der Inhalt der TEXT-Datei gespeichert werden soll. Beachten Sie die Richtlinien für Dateinamen.

4. Geben Sie den gewählten Dateinamen ein (Anführungszeichen nicht vergessen!), und drücken Sie ⟨**ENTER**⟩.

Wenn die Anzeige vom Bildschirm verschwindet, ist die Information unter dem erwählten Dateinamen abgespeichert.

Für detailliertere Informationen über das Speichern von Dateien auf Kassette schlagen Sie in Kapitel 9 nach.

DIE SELECT-FUNKTION

Die Select-Funktion (**F7**) wird benutzt, um einen Textblock zu definieren, der kopiert ("Copy" - **F5**), gelöscht ("Cut" - **F6**) oder verschoben (Cut and PASTE) werden soll. Solch ein Block kann in der Länge variieren, von einem einzelnen Zeichen bis zu der Gesamtheit der Zeichen in der TEXT-Datei.

Grenzen werden Ihnen jedoch auferlegt von dem Ihnen zur Verfügung stehenden Speicherraum. Wenn der gewählte Block mehr Speicherplatz belegt, als in dem Hauptmenü unter Bytes free angegeben ist, erhalten Sie über den Bildschirm die Mitteilung:

Memory full

Wenn Sie beim Drücken der Taste **PASTE** ebenfalls diese Botschaft erhalten, bedeutet das, daß Sie die RAM-Kapazität überschritten haben. Deshalb sollten Sie bei langen Textblocks Vorsicht walten lassen.

Um einen Textblock zu definieren, positionieren Sie den Cursor an den Beginn des Textes, der verändert werden soll, und drücken ⟨**F7**⟩ . Mit den Cursor-Steuerungstasten können Sie dann den Block definieren, entweder Zeichen für Zeichen mit der Taste mit dem Pfeil nach rechts, oder Zeile für Zeile, indem Sie die Taste mit dem Pfeil nach unten benutzen. Zeichen in dem angesprochenen Textblock erscheinen "negativ" auf schwarzem Hintergrund, wie Sie es in der Abbildung 5-4 unten sehen.

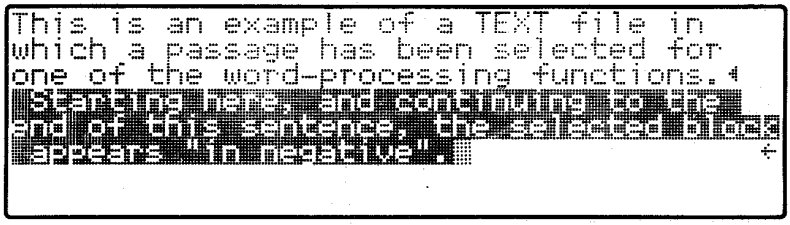


Abbildung 5-4 Ein ausgewählter Textblock

Der Text kann also definiert werden von der Cursorposition an nach vorne oder von derselben Position an nach hinten durch die gesamte Datei. Dieses wird zusammengefaßt in Abbildung 5-5.

Zu definierender Textblock	Tastenfolge
Nächstes Zeichen nach rechts	⟨F7⟩ dann →
Nächstes Zeichen nach links	⟨F7⟩ dann ←
Das auf den Cursor folgende Wort	⟨F7⟩ dann ⟨SHIFT⟩ + →
Das vor dem Cursor stehende Wort	⟨F7⟩ dann ⟨SHIFT⟩ + ←
Vom Cursor an bis ans Ende der aktuellen Zeile	⟨F7⟩ dann ⟨CTRL⟩ + →
Vom Cursor an bis zum Beginn der aktuellen Zeile	⟨F7⟩ dann ⟨CTRL⟩ + ←
Vom Cursor bis zur letzten Zeile des Bildschirms	⟨F7⟩ dann ⟨SHIFT⟩ + ↓
Vom Cursor bis zum oberen Rand des Bildschirms	⟨F7⟩ dann ⟨SHIFT⟩ + ↑
Den ganzen Text vom Cursor bis zum Ende der Datei	⟨F7⟩ dann ⟨CTRL⟩ + ↓
Den ganzen Text vom Cursor bis zum Beginn der Datei	⟨F7⟩ dann ⟨CTRL⟩ + ↑

Abbildung 5-5 Definieren eines Textblocks

Eine andere Methode, einen Textblock auszuwählen, schließt die "Find"(Such)-Funktion ein. Hiermit ist es möglich, Text von der Stellung des Cursors an bis zu einem bestimmten Wort oder einer Wortfolge anzusprechen. Hierzu gehen Sie folgendermaßen vor:

1. Drücken Sie ⟨**F7**⟩, wenn Sie den Cursor in die Startposition gebracht haben.

2. Nun drücken Sie ⟨**F1**⟩ und geben die Zeichenfolge oder das Wort an, mit dem der zu definierende Textblock endet.

3. Drücken Sie ⟨**ENTER**⟩. Der gesamte Text von der augenblicklichen Cursorposition bis zum ersten Zeichen der Zeichenfolge (exclusive) wird nun definiert und erscheint "negativ" auf dem Bildschirm.

Wenn mehr Text als nötig definiert worden ist, bewegen Sie den Cursor zurück bis zu dem erwählten Endpunkt, und der Extra-Text wird aus dem Block wieder entfernt. Um alle Definitionen rückgängig zu machen, drücken Sie ⟨**BREAK**⟩, d. h. ⟨**SHIFT + PAUSE**⟩. Alle Definitionen sind jetzt gelöscht, der definierte Textblock verschwindet vom Bildschirm, und Sie können mit einer neuen Definition beginnen.

DIE COPY-FUNKTION

Die Funktionstaste **F5** bietet Ihnen die Möglichkeit, einen Textblock (d. h. ein wiederkehrendes Wort oder eine Wortfolge) zu duplizieren und in einen anderen Teil der Datei zu übertragen. Die zwei Funktionen "Copy" und "Cut" schließen nicht nur die Funktionstasten (**F5** und **F6**) ein, sondern auch die "Select"-Funktion und den PASTE-Befehl. Um einen Kopiervorgang durchzuführen, gehen Sie folgendermaßen vor:

Drücken Sie ⟨**F7**⟩ und definieren Sie den Textblock, wie im vorhergehenden Abschnitt erklärt. der ausgewählte Block erscheint dann "negativ" auf schwarzem Hintergrund.

1. Drücken Sie ⟨**F5**⟩. Der dunkle Hintergrund verschwindet und zeigt dadurch an, daß der ausgewählte Textblock in dem PASTE-Puffer gespeichert ist.

2. Bewegen Sie den Cursor an die Stelle, an der die Kopie eingefügt werden soll, und drücken Sie ⟨**PASTE**⟩.

Beachten Sie bitte, daß die Einfügung so erfolgt, daß das letzte Zeichen des kopierten Textes den Raum unmittelbar vor der Cursorposition einnimmt, außer, wenn das letzte Zeichen in dem ausgewählten Block ein Zeichen für eine neue Zeile gewesen ist (wenn ⟨**ENTER**⟩ gedrückt wurde). In diesem Falle steht es in der vorhergehenden Zeile. Wenn ein und derselbe Text mehrere Male in mehr als eine Position in der Datei gebracht werden soll, braucht nur der zweite Schritt wiederholt zu werden, weil der ausgewählte Text in dem PASTE-Puffer verbleibt, bis er durch einen anderen Select-Vorgang ersetzt wird.

DIE "CUT AND PASTE"-FUNKTION

Diese Funktion dient dazu, einen Textblock zu löschen oder ihn innerhalb der Datei von einer Position auf eine andere zu schieben. Zur einfachen Löschung benutzen Sie die Funktionstaste ⟨**F6**⟩ in Verbindung mit der Select-Taste ⟨**F7**⟩. Bei einem "Cut and Paste"-Vorgang (Zerlegen und Montieren) benutzen Sie die Funktionstaste **F6** zusammen mit "Select" und ⟨**PASTE**⟩. Der Vorgang sieht folgendermaßen aus:

1. Drücken Sie ⟨**F7**⟩, und definieren Sie den Text, der gelöscht werden soll, wie vorher angegeben. Der ausgewählte Textblock erscheint auf dunklem Hintergrund.

2. Drücken Sie ⟨**F6**⟩. Der ausgewählte Text wird aus der Datei gelöscht und in dem **PASTE**-Puffer gespeichert. Wenn es sich nur um ein einfaches Löschen handelt, sind Sie nun fertig. Der gelöschte Text kann von dem **PASTE**-Puffer abgerufen werden, bis er durch eine andere "Select"-Operation ersetzt wird.

3. Um einen Text in eine andere Position in der Datei zu bringen, bewegen Sie den Cursor in die neue Position und drücken ⟨**PASTE**⟩. Der in der vorhergehenden Position gelöschte Text wird nun an der neuen Stelle eingefügt.

ANMERKUNG: Sie können sowohl die "Copy"- wie auch die "Cut and Paste"-Funktion benutzen, um Daten von einer Datei in eine andere zu übertragen. Die Prozedur bis zum letzten Schritt ist identisch mit der oben be-

schriebenen. Anstatt jedoch jetzt den Cursor für Einfügung an eine Stelle in der Arbeitsdatei zu bringen, geben Sie jetzt die Datei an, in die der Textblock übertragen werden soll. Positionieren Sie den Cursor an die entsprechende Stelle in dieser Datei, und drücken Sie **PASTE** .

DAS DRUCKEN EINER TEXT-DATEI

Wenn Sie die dafür vorgesehene Befehlstaste **PRINT** drükken, können Sie eine TEXT-Datei ausdrucken lassen. Zu diesem Zweck muß der M10 natürlich mit Hilfe eines Olivetti-Druck-Kabels mit einem Drucker verbunden sein. Die Verbindung erfolgt über die parallele PRINTER-Schnittstelle an der Rückseite.

- Wenn Sie ⟨**PRINT**⟩ drücken, wird der aktuelle Bildschirminhalt ausgedruckt.

- Wenn Sie ⟨**SHIFT + PRINT**⟩ drücken, wird die gesamte Textdatei ausgedruckt. Dieser Befehl ruft die Bildschirmanzeige

  Width?

  hervor und gibt auch die angegebene Zeilenlänge an. Wenn Sie diese ändern wollen, geben Sie einen Wert von 10 bis 132 an, und drücken Sie ⟨**ENTER**⟩. Hierdurch wird die Anzahl der Zeichen in einer Zeile des gedruckten Textes festgelegt.

  Wenn Sie ⟨**ENTER**⟩ drücken, ohne vorher einen Wert für die Zeilenlänge zu geben, richtet sich der M10 nach der zuletzt angegebenen Zeilenlänge. Die Standardzeilenlänge ist 80 Zeichen.

DIE MENÜ-FUNKTION

Die Funktionstaste **F8** ist gekennzeichnet als "Menü" und bewirkt zweierlei: Wenn Sie diese Taste drücken, wird das Arbeiten mit der augenblicklichen Datei abgeschlossen, und Sie werden zum Hauptmenü zurückgeführt. Die Datei wird dann im RAM gespeichert, und zwar unter ihrem Dateinamen, und das Menü wird auf dem Bildschirm angezeigt.

ZUSAMMENFASSUNG VON FUNKTIONS- UND BEFEHLSTASTEN

Abbildung 5-6 faßt die Bedeutung der verschiedenen Funktions- und Befehlstasten zusammen, wenn Sie im TEXT-Modus vorgehen.

TASTE	FUNKTION
F1 Find	Sucht eine vorgegebene Zeichenfolge im Text auf.
F2 Load	Lädt eine Datei von einem Kassettenrekorder in den M10.
F3 Save	Speichert eine Datei auf einem Kassettenrekorder.
F4	Im TEXT-Modus nicht benutzt.
F5 Copy	Wird in Verbindung mit **PASTE** benutzt, um einen Textblock innerhalb der gleichen Datei an eine andere Stelle zu kopieren oder ihn in eine andere Datei zu kopieren.
F6 Cut	Löscht einen Textblock, oder bewirkt in Verbindung mit **PASTE** einen Zerlegen- und Montieren-Vorgang.
F7 Select	Definiert einen Textblock, um ihn anschließend zu kopieren, zu verschieben oder zu löschen.
F8 Menu	Schließt das Arbeiten mit der Datei ab und bringt den Benutzer ins Hauptmenü zurück.
PASTE	Fügt den Inhalt des PASTE-Puffers in die Datei an der Position ein, die durch den Cursor bestimmt ist. Wird bei der Copy-Funktion und beim Zerlegen und Montieren benutzt.
LABEL	Gibt die Belegung der Funktionstasten auf dem Bildschirm an.

PRINT	Druckt den Inhalt des Bildschirmes oder der Datei auf einem externen Drucker aus.
PAUSE/BREAK	Nur BREAK wird im Text-Modus benutzt, um einen Vorgang wie "Select", "Find", "Load" oder "Save" zu stornieren.

Abbildung 5-6 Funktions- und Befehlstasten im TEXT-MODUS

KONTROLLZEICHEN

Alle Funktions- und Cursor-Steuerungstasten erzeugen innerhalb des M10 jeweils eine bestimmte Kontrollzeichenfolge. Es ist normalerweise nicht nötig, diese Zeichenfolge zu wissen, aber es könnte doch einmal nützlich sein. Daher wird in Abbildung 5-7 ein Überblick darüber gegeben.

Kontroll-zeichen-folge **CTRL** +	entspricht	Funktion
A	⟨**SHIFT**⟩ + ⟨←⟩	Cursor bis an den Anfang des vorhergehenden Wortes
B	⟨**SHIFT**⟩ + ⟨↓⟩	Cursor bis an den unteren Rand des Bildschirms
C	⟨**BREAK**⟩	Storniert "Select", **PRINT**, "Save", "Load" und "Find"
D	⟨→⟩	Cursor ein Zeichen nach rechts
E	⟨↑⟩	Cursor eine Zeile nach oben
F	⟨**SHIFT**⟩ + ⟨→⟩	Cursor bis an den Anfang des nächsten Wortes
G	⟨**F3**⟩	"Save"-Funktion
H	⟨**DEL/BS**⟩	Löscht das vorhergehende Zeichen
I	⟨→|⟩	Tabulator
L	⟨**F7**⟩	"Select"-Funktion

Abbildung 5-7 Kontrollzeichenfolgen für die Funktionstasten (siehe auch nächste Tabelle)

Kontroll-zeichen-folge CTRL +	Entspricht	Funktion
M	⟨ENTER⟩	Eingabe eines Befehls; Abschluß einer Eingabe
N	⟨F1⟩	"Find"-Funktion
Q	⟨CTRL⟩ + ⟨←⟩	Bringt den Cursor an den Beginn der aktuellen Zeile
R	⟨CTRL⟩ + ⟨→⟩	Bringt den Cursor ans Ende der aktuellen Zeile
S	⟨←⟩	Bringt den Cursor ein Zeichen nach links
T	⟨SHIFT⟩ + ⟨↑⟩	Bringt den Cursor bis an den Beginn des Bildschirms
U	⟨F6⟩	"Cut"- (Zerlegen-) Funktion
V	⟨F2⟩	"Load"-Funktion
W	⟨CTRL⟩ + ⟨↑⟩	Bringt den Cursor an den Beginn der Datei
X	⟨↓⟩	Bringt den Cursor eine Zeile nach unten
Y	⟨PRINT⟩	Ausgabe auf Drucker
Z	⟨CTRL⟩ + ⟨↓⟩	Bringt den Cursor ans Ende der Datei

Abbildung 5-7 Kontrollzeichenfolgen der Funktionstasten

6. DAS ADDRSS-ANWENDUNGSPROGRAMM

Dieses Programm bietet dem Anwender die Möglichkeiten, auf alle Informationen eines persönlichen Adreß- und Telefonbuches unmittelbar und in einer Vielzahl von Formen zugreifen zu können. Der Anwender kann die unterschiedlichen Möglichkeiten des Adreßprogramms nutzen, um es auf seine eigenen Bedürfnisse anzupassen und die Informationen in dem persönlichen Adreßbuch mit Name, Nummer, Kategorie, Beruf oder geographischen Bereich aufzulisten.

Erforderlich ist es jedoch, zuerst dem Computer die nötigen Informationen zu geben, in diesem Fall eine Liste der Namen, Adressen, Telefonnummern, die erfaßt werden sollen.

DAS ERSTELLEN DER "ADRS.DO"-DATEI

Greifen Sie auf das Textprogramm zu, indem Sie den Cursor auf das Wort **TEXT** im Hauptmenü bringen und dann **ENTER** drücken. Als Antwort auf die Bildschirmanzeige geben Sie den Dateinamen an. Für diesen besonderen Anwendungsbereich gibt es keine Auswahl an Dateinamen - der Name ADRS.DO ist verpflichtend.

Wenn Sie das ADDRSS-Programm aufrufen, bevor Sie die ADRS.DO-Datei erstellt haben, gibt der Computer folgende Meldung auf dem Bildschirm:

ADRS.DO not found
Press space bar for MENU

(ADRS.DO wurde nicht gefunden, Leertaste drücken, um zum Menü zurückzukehren)

Nachdem Sie die ADRS.DO-Datei erstellt haben, geben Sie die Namen, Adressen und Telefonnummern, die Sie in Ihrer Liste führen wollen, ein. Der Aufbau der Eingabe ist Ihnen überlassen, wir empfehlen folgende Reihenfolge:

Name :Telefonnummer: Adresse

Wir empfehlen Ihnen sehr, Ihre Aufzeichnungen in einem festen und geordneten Format einzugeben, damit die Suchfunktion leichter vor sich gehen kann, und damit

Sie eine übersichtliche Adreßliste vor sich haben, wenn Sie sie auf dem Bildschirm oder als Ausdruck betrachten.

Es ist unbedingt erforderlich, zu geeignetem Zeitpunkt die Taste **ENTER** zu drücken. Das ist die einzige Möglichkeit, durch die der Computer einer zusammengehörigen Gruppe von Informationen von einer anderen unterscheiden kann. Das Symbol für **ENTER**, eine dunkle, nach links weisende Pfeilspitze, sollte am Ende einer solchen Eingabegruppe erscheinen und nirgendwo sonst.

Beachten Sie, daß ADRS.DO eine TEXT-Datei ist, und daß deshalb alle die Möglichkeiten zur Erstellung von Texten, die in Kapitel 5 beschrieben worden sind, auch hier für die Eingabe und das Erstellen von Texten in die ADRS.DO-Datei gültig sind. Lassen Sie sich nicht verwirren durch die nahe beieinanderliegenden, aber doch deutlich unterschiedenen Funktionen ADRS.DO und ADDRSS. ADRS.DO ist diejenige Datei, die das Adressbuch enthält; ADDRSS ist dasjenige Programm, welches diese Informationen sortiert und organisiert, so wie Sie das wünschen.

Wenn die Liste von Namen, Adressen und Telefonnummern komplett ist, schließen Sie die Arbeit mit der Datei ab und kehren in das Hauptmenü zurück, indem Sie die Taste **F8** drücken. Die Datei erscheint in dem Menü als ADRS.DO.

DER GEBRAUCH DES ADDRSS-PROGRAMMS

Das ADDRSS-Programm kann nun aufgerufen werden, indem man den Cursor auf ADDRSS bringt und ⟨**ENTER**⟩ drückt.

Der Bildschirm sieht jetzt folgendermaßen aus:

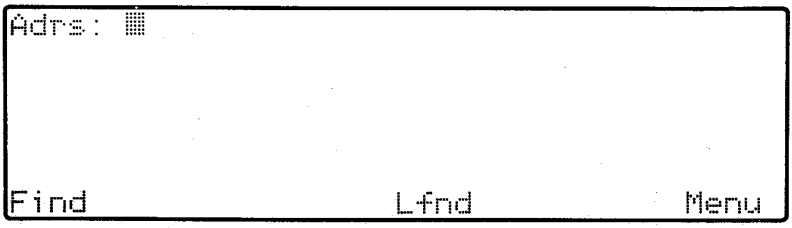


Abbildung 6-1 Bildschirm beim ADDRSS-Programm

Bei diesem Programm werden nur drei der Funktionstasten verwendet, nämlich **F1**, **F5** und **F8**.

- **F1** ruft die "Find"- (Such-) Funktion auf.

- **F5** ist gekennzeichnet mit "Lfnd" und hat dieselbe Funktion wie **F1** mit der Ausnahme, daß das Ergebnis auf dem Drucker ausgegeben wird, falls dieser angeschlossen ist.

- **F8** bringt den Anwender genau wie in den anderen Programmen in das Menü zurück.

Die Such-Funktion läuft genauso ab wie im TEXT-Programm. In der oberen Zeile des Bildschirms erscheint die Mitteilung:

Adrs:

Wenn Sie nun eine spezielle Eingabe in der Adress-Datei suchen wollen, brauchen Sie nur ⟨**F1**⟩ zu drücken. Dann erscheint die Anzeige **Find**, und Sie geben die gesuchte Folge von Zeichen ein. Solch eine Zeichenfolge kann aus einem einzigen Zeichen bestehen oder aus der gesamten Eintragung (max. 24 Zeichen). Alle Eintragungen, die die gesuchte Zeichenfolge enthalten, werden in voller Länge auf dem Bildschirm angezeigt.

Wenn Sie anstelle von⟨**F1**⟩ die Taste⟨**F5**⟩ drücken, wird das Ergebnis auf dem Bildschirm genauso wie auf dem Drucker ausgegeben. Wenn Sie ⟨**F5**⟩ drücken und kein Drucker angeschlossen ist, wird der Computer blockiert. Sie können ihn wieder starten, indem Sie ⟨**SHIFT** + **PAUSE/BREAK**⟩drücken.

Wenn Sie den Inhalt der ADRS.DO-Datei in der Reihenfolge sehen wollen, in der er eingegeben wurde, rufen Sie ADDRSS auf und drücken⟨**F1**⟩ und dann⟨**ENTER**⟩. Hierdurch werden die sechs ersten Zeilen der Datei auf dem Bilschirm angezeigt. Die Funktionen **F3** und **F4** sind gekennzeichnet mit "More" und "Quit". Wenn Sie ⟨**LABEL**⟩ drücken, können Sie diese Bezeichnungen entfernen. Um die nächsten sechs Zeilen zu sehen, drücken Sie⟨**F3**⟩usw. bis zum Ende der Datei. Wenn Sie⟨**F4**⟩ drücken, wird die Bildschirmanzeige beendet, und die Anzeige **ADDRSS** erscheint auf dem Bildschirm.

Das folgende Beispiel wird Ihnen verdeutlichen, was wir gerade erklärt haben.

Nehmen wir an, daß die ADRS.DO-Datei in der folgenden Art gestaltet worden ist:

1. G. V. Hubschmidt : 0231/74320 : Flugschnepfenweg 34, 4600 Dortmund

2. Heinrich Napiralla : 02302/78210 : Fahrendelle 3, 5810 Witten

3. Walter Tütenglanz : 08231/342 : Hodlerweg 3, 8231 Chiming

4. P. A. Bruchsitz : 0208/123245 : Walterstraße 5, 4000 Düsseldorf

5. Tilman Schnepf : 07212/12783 : Lockerweg 17, Mieming

Nun nehmen wir an, daß Sie das ADDRSS-Programm aufrufen und die "Find"- (Such-) Option ansteuern.

Wenn die eingegebene zu suchende Zeichenfolge "Hubschmidt" ist, dann wird die erste Eintragung ganz auf dem Bildschirm erscheinen.

Wenn die gesuchte Zeichenfolge "Walter" ist, dann werden die Eintragungen 3 und 4, in denen Walter in irgendeiner Form vorkommt, auf dem Bildschirm angezeigt.

Nun ändern Sie die gewünschte Zeichenfolge in "Schnepf". Dieses bringt natürlich die Eintragung 5 auf den Bildschirm, aber ebenfalls wird die Eintragung 1 angezeigt, in der das Wort "Schnepf" in Flugschnepfenweg vorkommt. (Denken Sie daran, daß zwischen Groß- und Kleinbuchstaben kein Unterschied gemacht wird.) Wenn in diesem Fall die zu suchende Zeichenfolge dargestellt wird als " Schnepf " (d. h. mit einer Leertaste vor und nach dem Buchstaben), wird nur Eintragung 5 erscheinen. Wenn Sie vermeiden wollen, daß mehrere Eintragungen erscheinen, muß die zu suchende Zeichenfolge lang genug sein, um einmalig zu sein (Leerzeichen berücksichtigen!)

Die Möglichkeit, daß bei einer suche mehrere Eintragungen angezeigt werden, kann aber auch zum Vorteil des Benutzers eingesetzt werden. Wenn Sie eine unwahrscheinliche Zeichenfolge an jede Eintragung anhängen, kann man Einteilungen (Klassifizierungen) vornehmen, und das ADDRSS-Programm wird Ihnen alle unter der entsprechenden Zeichenfolge abgespeicherten Eintragungen auflisten, wenn Sie diese spezielle Zeichenfolge angeben. Z. B. könnte man alle Männer mit dem Anhang XXX, alle Frauen mit YYY, alle Personalchefs mit dem Anhang PXP und alle Bankmanager mit dem Anhang BXB kennzeichnen usw.

Das Programm läßt sich also an die speziellen Bedürfnisse des Benutzers anpassen. Wenn Sie solche Selektionskriterien benutzen, können Sie die Informationen, die Sie gespeichert haben, mit Hilfe von ADRS.DO auf ganz unterschiedliche Arten ausgeben lassen, nämlich unterschieden nach den Selektionskriterien, die Sie wünschen, z. B. Beruf, geographische Lage usw.

7. DAS SCHEDL-ANWENDERPROGRAMM

Dieses Programm ermöglicht es dem Benutzer unter anderem, einen Terminkalender individuell auf seine Bedürfnisse zuzuschneiden und dabei die unterschiedlichsten Informationen zu erstellen und zu verwalten. Ähnlich wie das Programm ADDRSS ist SCHEDL ein Dateiverwaltungsprogramm, das die Termine und zugeordneten Informationen innerhalb der Datei NOTE.DO verwaltet. Sie können alle möglichen weiteren Informationen in NOTE.DO speichern, wie z. B. eine Aufzeichnung Ihrer Ausgaben, Ihr Tagebuch, allgemeine Notizen oder sonstige von Ihnen benötigte Informationen.

ERSTELLEN DER "NOTE.DO"-DATEI

Erstellen Sie sich Ihre TEXT-Datei, indem Sie den Cursor über TEXT im Hauptmenü positionieren und dann ⟨**ENTER**⟩ drücken. Als Antwort auf die Bildschirmmeldung

File to edit?

geben Sie NOTE ein und drücken ⟨**ENTER**⟩ (wie immer erscheint automatisch das Typkennzeichen .DO). Der Bildschirm wird gelöscht, und der Cursor steht am Anfang der Datei. Wenn Sie SCHEDL aufrufen, bevor Sie NOTE.DO erstellt haben, erscheint die folgende Bildschirmmeldung:

NOTE.DO not found
Press space bar for MENU

(NOTE.DO wurde nicht gefunden; drücken Sie die Leertaste, um ins Menü zurückzukehren)

In die Datei NOTE.DO geben Sie diejenigen Informationen ein, die normalerweise in einem Terminplan, einem Tagebuch usw. enthalten sind, wie z. B.: eine Liste ihrer Terminverpflichtungen, Ereignisse und Daten, an die sie erinnert werden wollen, geschäftliche Ausgaben usw. Unabhängig jedoch davon, was Sie in dieser Datei haben wollen, sind die folgenden Punkte als Richtlinie nützlich:

- Gestalten Sie die Informationen in einer logischen, geordneten Reihenfolge. Bedenken Sie dabei, wie sie auf dem Bildschirm erscheinen werden, wenn Sie sie wieder aufrufen.

- Jede separate Eintragung oder jede Gruppe von zusammengehörigen Informationen sollte abgeschlossen werden, indem Sie ⟨**ENTER**⟩ drücken. Weiterhin sollten sich Ihre Aufzeichnungen innerhalb eines vernünftigen Größenrahmens halten, so daß Sie die gesamte Information schnell zur Verfügung haben, wenn Sie sie wieder auf dem Bildschirm abrufen.

- Eine große Erleichterung für den Aufbau bzw. für die Organisation der Aufzeichnungen ist das Voranstellen von Symbolen oder Zeichen vor die Eintragung. Ausgaben könnten z. B. vorher ein $- oder ein DM-Zeichen erhalten, Verabredungen ein # (Zahlenzeichen), dringend zu erledigende Angelegenheiten ein ! (Ausrufungszeichen) usw.

Wenn Sie die Eingaben in NOTE.DO beendet haben, drücken Sie **F8**, um die Datei zu schließen, und kehren damit in das Menü zurück.

DER EINSATZ DES SCHEDL-PROGRAMMS

Sie können SCHEDL aufrufen, indem Sie den Cursor über SCHEDL im Hauptmenü bringen und ⟨**ENTER**⟩ drücken. Der Bildschirm sieht dann auch wie in Abbildung 7-1 unten.

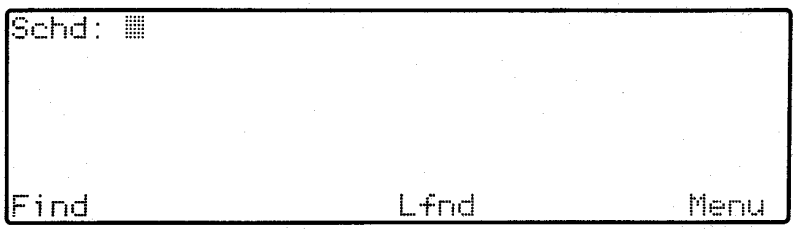


Abbildung 7-1 Der Bildschirm beim SCHEDL-Programm

In der ersten Zeile des Bildschirm erscheint die Anzeige:

Schd:

Bei diesem Programm werden nur drei spezielle Funktionstasten benutzt, nämlich F1 - Find, F5 - Lfnd und F8 - Menu, und diese werden in der Fußzeile des Bildschirms angezeigt. Um die Bezeichnungen vom Bildschirm zu entfernen, müssen Sie ⟨**LABEL**⟩ drücken.

- F1 (Find = Suchen) hat dieselbe Funktion wie im ADDRSS-Programm. Drücken Sie jetzt ⟨**F1**⟩, und wenn die Bildschirmanzeige **Find** erscheint, geben Sie diejenige Zeichenfolge ein, nach der Sie in NOTE.DO suchen wollen. Dann drücken Sie die Taste ⟨**ENTER**⟩. Jetzt wird jede Eintragung, in der die gesuchte Zeichenfolge vorkommt, in ganzer Länge auf dem Bildschirm angezeigt.

- F5 (Lfnd) hat im wesentlichen die gleiche Funktion wie "Find", mit der Ausnahme, daß das Ergebnis nicht auf dem Bildschirm, sondern als Ausdruck auf einem an den M10 angeschlossenen Drucker erscheint.

  Beachten Sie bitte, daß der Computer blockiert wird, wenn Sie diese Option wählen, ohne daß ein Drucker angeschlossen ist. Die Blockierung kann aufgehoben werden, wenn Sie die Taste ⟨**SHIFT**⟩ zusammen mit ⟨**PAUSE/BREAK**⟩ drücken.

- Wenn Sie ⟨**F8**⟩ drücken, kommen Sie genauso wie in den anderen Anwenderprogrammen zum Hauptmenü zurück.

Wenn Sie den Inhalt der NOTE.DO-Datei überprüfen wollen, rufen Sie SCHEDL auf, drücken ⟨**F1**⟩ und dann ⟨**ENTER**⟩. Die ersten sechs Zeilen von NOTE.DO werden dann auf dem Bildschirm angezeigt. Die Funktionstaste F3 trägt die Bezeichnung "More" und die Taste F4 die Bezeichnung "Quit" (Ende). Um die nächsten sechs Zeilen auf dem Bildschirm zu sehen, drücken Sie ⟨**F3**⟩ usw. bis zum Ende der Datei. Wenn Sie keine weiteren Eintragungen angezeigt haben möchten, drücken Sie ⟨**F4**⟩, und Sie kommen zur SCHEDL-Anzeige zurück.

Der Gebrauch von SCHEDL ist am besten an einem Beispiel zu verdeutlichen. Nehmen wir an, die folgenden Eintra-

gungen stellten einen Auszug aus der NOTE.DO-Datei dar.

```
11/07 $ Ausgaben in Genf
        Taxi vom Flughafen SF 42,70
        Lunch              SF 25,00
        Dinner             SF 67,90
        Taxi zum Flughafen SF 45,00

12/07 # 10.30  Verabredung mit P. M. Engel

12/07 # 14.30  Treffen der Abteilungsleiter

13/07 # 12.30  Mittagessen mit Dr. Wegerich

14/07 ! Telefonrechnung bezahlen

15/07 * 15.00 Abflug nach London LH 151
        Reservierung Savoy Hotel

16/07 ! Herrn Hesserich in Bergkamen anrufen
```

Diese unterschiedlichen Informationen für die Daten 11/07 bis 16/07 können mit dem SCHEDL-Programm selektiv abgerufen werden. In diesem besonderen Falle sind die Eingaben wie folgt codiert:

$ steht für Geschäftsausgaben

steht für Verabredungen

! steht für dringend zu Erledigendes

* steht für Reise

Es gibt nun eine Vielzahl an Möglichkeiten, die gespeicherten Informationen abzurufen:

Zuerst rufen Sie im Menü SCHEDL auf und drücken ⟨F1⟩.

- Wenn Sie die Zeichenfolge

  14/07 !

  eingeben, werden alle Punkte, die am 14/Ø7 dringend zu erledigen sind, angezeigt (in diesem Fall das Bezahlen der Telefonrechnung).

- Auf die gleiche Art und Weise können Sie sich eine Liste der Verabredungen, Ausgaben oder Reisetermine aufzeigen lassen, wenn Sie die einzelnen Symbole #, $ oder * eingeben.

- Wenn Sie ein spezielles Datum eingeben, erhalten Sie eine Liste aller Eintragungen für diesen speziellen Tag.

- Wenn Sie einen Namen (z. B. Wegerich) eingeben, erhalten Sie alle Informationen, die diese Person betreffen.

Diese Art der Codierung kann natürlich stark erweitert werden. Z. B. kann die Doppelcodierung *$ für Eintragungen benutzt werden, die sowohl Reisen wie auch Ausgaben betreffen usw.

Beachten Sie bitte, daß die Symbole, die in diesem Beispiel benutzt wurden, nur das Prinzip der Codierung darstellen sollen. Der Benutzer kann beliebige Codes, die auf seine Bedürfnisse angepaßt sind, benutzen. In diesem Zusammenhang muß darauf hingewiesen werden, daß die Tastatur des M10 spezielle Symbole zur Verfügung stellt, wie z. B.

ein Flugzeug	⟨SHIFT⟩ + ⟨GRPH⟩ + **c**	
ein Telefon	⟨SHIFT⟩ + ⟨GRPH⟩ +	auf den US und UK Tastaturen
	⟨SHIFT⟩ + ⟨GRPH⟩ +	auf den italienischen, französischen und deutschen Tastaturen
ein Auto	⟨SHIFT⟩ + ⟨GRPH⟩ + **v**	

Die NOTE.DO-Datei kann aktualisiert werden, indem man von Zeit zu Zeit nicht mehr benötigte Eintragungen löscht. Wenn Sie aber Informationen über einen längeren Zeitraum speichern und auswerten wollen, sagen wir von Ihren Reisekosten, gibt es eine ausgeklügeltere Methode. Fügen Sie einen Buchstaben an den Code an als Zeichen für eine ältere Aufzeichnung. Nehmen wir einmal an, der gewählte Buchstabe ist ein x. Wenn Sie jetzt SCHEDL anweisen, *x zu suchen, erhalten Sie eine Liste

aller als "alt" zu betrachtenden Reiseausgaben.

Die Anwendungsmöglichkeiten des M10 sind lediglich durch Ihren Einfallsreichtum und Ihre Kreativität begrenzt. Die NOTE.DO-Datei kann so einfach oder auch so kompliziert sein, wie Sie sie haben wollen.

8. DAS TELCOM-PROGRAMM

Das TELCOM-Programm gestattet es dem Benutzer des M10, mit anderen Computern zu kommunizieren, indem er über eine Telefonleitung oder über eine Direktverbindung entweder Daten sendet oder empfängt.

Die zwei prinzipiellen Anschlußmöglichkeiten sind in der folgenden Tabelle angegeben.

Datenübertragung von oder zu einem Host-Rechner über das öffentliche Telefonnetz	Akustikkoppler
Direkter Anschluß an einen anderen Computer	Verbindungskabel "Null-Modem" bzw. "Kreuz-Kabel" oder Direkt-Kabel (eins zu eins)

Abbildung 8-1 Anschlußmöglichkeiten mit dem TELCOM-Programm

Um das TELCOM-Programm aufzurufen, positionieren Sie den Cursor über die Anzeige TELCOM im Hauptmenü und drücken ⟨**ENTER**⟩. Der Bildschirm sieht nun aus wie in Abbildung 8-2 unten.

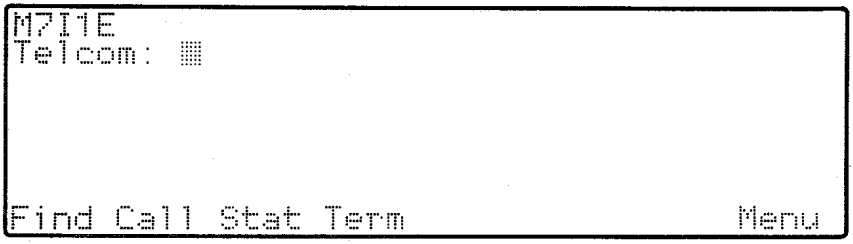


Abbildung 8-2 Bildschirm nach dem Aufruf von TELCOM

Der Code, der in der ersten Zeile der Anzeige erscheint, ist eine Liste der Übertragungsparameter, über die wir später in diesem Kapitel im einzelnen sprechen. In der zweiten Zeile sehen Sie die TELCOM-Anzeige. In

der Fußzeile finden Sie die Definitionen der Funktionstasten bei der TELCOM-Option.

ANSCHLUSS DES M10 AN DAS ÖFFENTLICHE TELEFONNETZ UND TELEFON-NEBENSTELLEN

Die Verbindung wird mit einem Akustik-Koppler realisiert, der über die RS 232- Schnittstelle des M10 angeschlossen wird. Der Akustik-Koppler kann von Olivetti bezogen werden.

In Anhang C finden Sie eine ausführliche Beschreibung dieser Einrichtung. Alle Länder haben ihre eigenen Vorschriften über den Gebrauch von Zusatzeinrichtungen, die an das Telefonnetz angeschlossen werden. Bevor Sie deshalb den M10 an das Telefonnetz anschließen, machen Sie sich mit den landestypischen Postvorschriften vertraut. Hierbei wird Ihnen Olivetti gern zur Seite stehen.

DIE FUNKTIONSTASTEN IM EINGABE-MODUS

Im Eingabe-Modus regeln die Funktionstasten die folgenden Operationen, wie Sie in Abbildung 8-2 sehen können.

F1 (Such)- Wenn Sie diese Taste drücken, kann das TELCOM-Programm einen Namen und eine Telefonnummer, die schon in der ADRS.DO- Datei aufgeführt ist, suchen und sie auf dem Bildschirm ausweisen. Diese Funktion ist nur beim Modell M10 MODEM verfügbar.

F2 (Wählen)- Wenn der M10 mit dem Telefonnetz verbunden ist, kann mit Hilfe dieser Taste automatisch eine angegebene Nummer gewählt werden. Diese Funktion existiert wiederum nur bei dem Modell M10 MODEM.

F3 (Stat)- Mit dieser Funktion werden die laufenden Kommunikationsparameter aufgelistet, und der Benutzer hat die Möglichkeit, diese zu modifizieren. Diese Funktion wird für den Datenaustausch mit einem anderen Computer benutzt und später in diesem Kapitel ausführlich besprochen.

F4 (Term)- Mit dieser Taste können Sie in den Terminal-Modus übergehen.

F5, **F6**, und **F7** werden nicht benutzt.

F8 (Menü)- Bringt den Benutzer in das Hauptmenü zurück.

TERMINAL-MODUS

Dieser Modus ermöglicht es dem M10, entweder über das Telefon oder direkt mit anderen Computern oder Informationsdiensten zu kommunizieren. In diesem Falle fungiert der M10 als Terminal eines Host-Computers. Es gibt zwei Möglichkeiten, den Terminal-Modus herzustellen - automatisch oder manuell.

Manuell wird der Terminal-Modus hergestellt, indem Sie **F4** drücken, nachdem das TELCOM-Programm vom Hauptmenü aufgerufen worden ist. Der Bildschirm sieht dann aus wie in Abbildung 8-4.

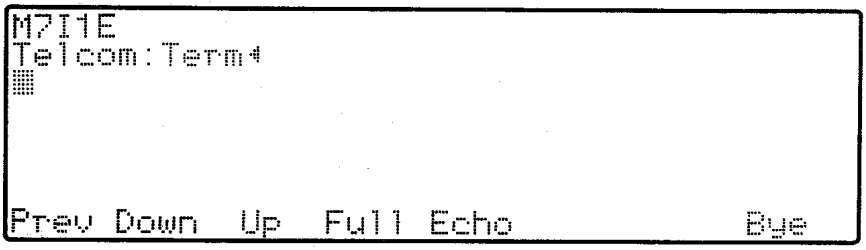


Abbildung 8-4 Bildschirm bei Eingabe des Terminal-Modus

DIE FUNKTIONSTASTEN IM TERMINAL-MODUS

Im Terminal- Modus werden durch die Funktionstasten **F1 - F8** folgende Aktivitäten ausgelöst:

F1 (Prev) - Wenn Sie im Terminal-Modus diese Taste drücken, können Sie die vorhergehenden acht Textzeilen auf dem Bildschirm sehen. Drücken Sie **F1**, wenn Sie auf dem Bildschirm in die Ausgangsposition zurückkehren wollen.

F2 (Down) - Der Sinn dieser Funktionstaste ist es, Daten vom Hostrechner zu übernehmen (Download) und sie anschließend auf einer Datei zu speichern. Wenn Sie **F2** drücken, erscheint die folgende Meldung auf dem Bildschirm:

File to Download?

Geben Sie jetzt den Namen der Datei an, in die die Daten gespeichert werden sollen, und drücken Sie **ENTER.** Drücken Sie wieder **F2**, wenn Sie das Abspeichern unterbrechen wollen.

F3 (Up) - Wenn Sie diese Funktionstaste benutzen, können Sie einen vorbereiteten Datensatz an einen Host-Computer senden. Wenn Sie die Taste **F3** drücken, erscheint die Anzeige:

File to upload?

Geben Sie jetzt den Namen des vorbereiteten Datensatzes ein, den Sie senden wollen, und drücken Sie **ENTER.** Dieses wiederum ruft die Anzeige hervor:

Width?

Als Antwort auf diese Anzeige geben Sie die Höchstzahl von Zeichen an, die der M10 senden wird, bevor ein

"Carriage Return" erfolgt. Sie können einen Wert zwischen 10 und 132 wählen.

F4 (Full/Half) - Mit dieser Funktionstaste können Sie zwischen Full Duplex und Half Duplex wählen. Im Full Duplex-Modus, einem Modus, den die meisten Host-Computer fordern, werden die Zeichen an den Host-Computer übertragen, bevor sie auf dem Bildschirm erscheinen. In diesem Falle wissen Sie, daß die Zeichen, die auf dem Bildschirm erscheinen, von dem Host-Computer akzeptiert wurden.

Im Half Duplex-Modus erscheinen die Zeichen auf dem Bildschirm und werden gleichzeitig gesendet, so daß Sie keine Garantie dafür haben, daß dasjenige, was Sie auf dem Bildschirm lesen, auch tatsächlich vom Gastcomputer empfangen wurde. Eine gestörte Telefonleitung kann z.B. eine Übertragung verfälschen.

F5 (Echo) - Wenn Sie einen Drucker an den M10 anschließen und **F5** drücken, erhalten Sie eine Hardcopy von dem Bildschirminhalt. Auf diese Art und Weise können Sie eine gedruckte Aufzeichnung des Dialogs mit dem Host-System erhalten. Die Funktionstaste **F5** ist ein Umschalter, d.h., wenn sie einmal gedrückt ist, ist diese Funktion in Betrieb genommen. Wenn Sie diese Taste noch einmal drücken, wird diese Funktion außer Betrieb gesetzt und die Anzeige "Echo" vom Bildschirm entfernt.

F6 und F7 werden nicht benutzt.

F8 (Bye) - Mit dieser Funktionstaste können Sie den Terminal-Modus verlassen. Wenn Sie F8 drücken, erscheint die Anzeige

Disconnect?

auf dem Bildschirm. Als Antwort darauf geben Sie ein **Y** (Yes) oder **N** (No) und drücken dann **ENTER**. Um den M10 vom Telefonnetz abzukoppeln, müssen Sie **"Y"** eingeben. Dieses bringt Sie in den Eingabe-Modus zurück.

DIE PARAMETER FÜR DATENKOMMUNIKATION

Wir haben schon früher auf die Übertragungs-Parameter hingewiesen, die in der ersten Zeile des Bildschirms erscheinen, wenn Sie TELCOM zum ersten Mal aufrufen. Diese Anzeige gibt die aktuellen Werte der Übertragungs-Parameter an. Um mit einem anderen Computer zu kommunizieren, müssen diese Parameter angepaßt werden, und Sie können die entsprechenden Werte dem M10 innerhalb der Grenzen, die in Abbildung 8-5 angegeben sind, eingeben.

PARAMETER	MÖGLICHE WERTE	BEDEUTUNG
Baud Rate	1	75 baud
	2	110 baud
	3	300 baud
	4	600 baud
	5	1200 baud
	6	2400 baud
	7	4800 baud
	8	9600 baud
	9	19200 baud
Word Length (Anzahl Bits pro Byte)	6	6 bits
	7	7 bits
	8	8 bits
Parity (Parität)	O	Odd (ungerade)
	E	Even (gerade)
	N	No Parity(kein Paritycheck)

	I	Ignore Parity (Paritycheck wird ignoriert)
Stop Bit (Anzahl Stopbits)	1	1 Stop Bit
	2	2 Stop Bits
Line Status	E	Enable
	D	Disable

Abbildung 8-5 Parameter für Datenkommunikation

In Abbildung 8-2 sind die Parameter gesetzt als 37I1E. Aus der folgenden Tabelle können Sie die Bedeutung der einzelnen Parameter entnehmen.

Baud Rate -	Die Übertragungsgeschwindigkeit beträgt 300 baud.
Word Length -	7 bits pro Byte
Parity -	I Paritycheck wird ignoriert
Stop Bit -	1 Stop Bit
Line Status -	E Enable

Es ist nicht unbedingt nötig, daß sich der Benutzer mit den Bedeutungen all dieser Parameter vertraut macht, um im Terminal-Modus mit einem Host-Computer umgehen zu können. Wir geben diese Bedeutungen hier nur an, damit Sie den M10 an die Spezifikation des Host-Computers anpassen können.

Um diese Übertragungs-Parameter zu ändern, gehen Sie in den Entry- (Eingabe-) Modus, und drücken Sie ⟨F3⟩ (Stat). Die Anzeige **Stat** erscheint auf dem Bildschirm. Geben Sie jetzt das Protokoll für die Kommunikation ein (d.h. die Folge von Parametern in dem gegebenen Format, wie z.B. 37I1E), und drücken Sie ⟨ENTER⟩; wiederum erscheint die TELCOM-Anzeige. Überzeugen Sie sich nun, daß die neuen Werte von dem M10 auch registriert wurden, und drücken Sie deshalb noch einmal ⟨F3⟩. Wenn jetzt die Anzeige **Stat** erscheint, drücken Sie einfach ⟨ENTER⟩, und die neuen Werte werden in der nächsten Zeile angezeigt. Beachten Sie, daß die Parameter für die Kommunikation

im Entry-Modus aufgerufen und modifiziert werden, obwohl ihre Verwendung einzig im Terminal-Modus erfolgt.

Die aktuellen Werte der Übertragungs-Parameter können jederzeit geprüft werden, wenn Sie ⟨F3⟩ und ⟨ENTER⟩ drücken.

DATENAUSTAUSCH MIT EINEM HOST-COMPUTER

Der Terminal-Modus dient dazu, dem M10 den Datenaustausch mit anderen Computern entweder über eine direkte Verbindung oder über das Telefonnetz zu ermöglichen.

Beim Anschluß über das Telefonnetz muß ein Akustik-Koppler an den M10 angeschlossen werden.

Bevor Sie den M10 im Terminal-Modus benutzen, muß er an das Telefonsystem angeschlossen werden.

Für den Anschluß eines Rechners über Akustik-Koppler gehen Sie wie folgt vor:

1. Überzeugen Sie sich, daß der Akustik-Koppler MC 10 mit der RS232C- Schittstelle an der Rückseite des M10 verbunden ist.

2. Schalten Sie den Ein-/Ausschalter an der Oberseite des MC 10 ein. Die Anzeigelampe zeigt dann ein blinkendes grünes Licht.

3. Heben Sie den Telefonhörer ab, und wählen Sie die Nummer des Host-Systems auf die herkömmliche Art und Weise.

   Wenn das Host-System über eine Einrichtung zum automatischen Senden von Daten verfügt, hören Sie einen hohen Dauerton, wenn die Verbindung hergestellt wurde. Wenn das Host-System nicht über solch eine Einrichtung verfügt, sondern manuell vorgegangen werden muß, bitten Sie den Bediener, die Verbindung zur Datenübertragung herzustellen. Wenn Sie einen Dauerton hören, ist diese Verbindung hergestellt worden.

4. Wenn die Verbindung hergestellt worden ist, drücken Sie **F4**, um in den Terminal-Modus überzugehen.

5. Bringen Sie den Telefonhörer auf die MC 10-Aufnahme, wobei Sie zuerst die Hörmuschel verbinden und dann die Sprechmuschel mit der Aufnahme, die mit CORD bezeichnet ist.

   Nach einem Moment zeigt die Anzeigelampe ein kontinuierliches grünes Licht.

Wenn Sie in den Terminal-Modus übergegangen sind, gibt der M10 einen hohen Ton von sich, und die neue Belegung der Funktionstasten **F1 - F8** erscheint in der Fußzeile des Bildschirms, wie Sie in Abbildung 8-4 sehen können.

Die Kombination von M10/MC 10 ist nun zum Anschluß an einen Host-Computer bereit.

MANUELLER ANSCHLUSS AN EINEN HOST-COMPUTER

Wir gehen davon aus, daß Sie Teilnehmer des Informationsdienstes 'Hurtig und Co.' (HuC) sind, dessen Telefon-Nummer 04711/77095 ist. Nehmen wir wei- ter an, daß HuC Ihnen die Teilnehmer-Nummer 5138Ø (USER ID) zugeteilt hat und Sie das Password "Dalles" haben.

Für den Netzanschluß sind dann folgende Instruktionen vorzunehmen:

1. Geben Sie dem System durch das Senden von Kontroll-C Nachricht, daß Sie bereit zum Anschluß sind.

2. Warten Sie auf die Anzeige **USER ID:**, dann geben Sie Ihre Identitätsnummer ein, mit der wir Sie versehen haben.

3. Warten Sie auf die Anzeige **Password:**, dann geben Sie das Passwort ein.

4. Warten Sie mindestens drei Sekunden, dann senden Sie die Frage **OK?**.

5. Das System wird Ihnen die Nachricht **Log-on successfully completed** senden.

Die Prozedur ist die folgende:

1. Verbinden Sie den Akustik-Koppler MC 10 mit der RS-232C-Schnittstelle an der Rückseite des M10, schalten Sie den Netzschalter ein, wie schon vorher in diesem Kapitel gezeigt wurde.

   Wenn Sie einen anderen Akustik-Koppler benutzen, befolgen Sie die Bedienungsanweisungen, die von dem Hersteller geliefert wurden, um sicherzugehen, daß er bereit zum Aufruf des Host-Computers ist.

2. Wählen Sie TELCOM vom Hauptmenü.

   Überzeugen Sie sich, daß die Übertragungs-Parameter, die in der ersten Zeile des Bildschirms angezeigt sind, den IDS-Spezifikationen entsprechen. Wenn das nicht der Fall ist, drücken Sie **F3** und geben die entsprechenden Werte ein. Beachten Sie, daß bei dem MC 10 die höchstmögliche Übertragungs-

rate (baud rate) 300 bauds beträgt.

3. Heben Sie den Telefonhörer ab, und wählen Sie die IDS-Nummer - 492 70095. Sobald die Verbindung hergestellt ist, legen Sie den Hörer in die Aufnahme des Akustik-Kopplers, wie in dem Abschnitt über den manuellen Zugang in den Terminal-Modus beschrieben wurde.

4. Drücken Sie ⟨**F4**⟩ (Term), um in den Terminal-Modus überzugehen. Wenn dies geschehen ist, gibt der M10 einen hohen Ton von sich. Zur gleichen Zeit werden in der Fußzeile des Bildschirms die Tastenbelegungen von **F1 - F8** für den Terminal-Modus angezeigt.

5. Nun folgen Sie den IDS-Anweisungen für die Anschlußprozedur.

6. Nachdem Sie einen Moment gewartet haben, um sicherzugehen, daß eine gute Verbindung mit dem Host-System aufgebaut worden ist, drücken Sie ⟨**CTRL** + **c**⟩, um dem IDS-System anzuzeigen, daß Sie bereit zum Anschluß sind.

7. Warten Sie, bis die Anzeige **USER ID:** auf dem Bildschirm erscheint. Dann geben Sie 5138Q ein und drücken ⟨**ENTER**⟩.

   Auf dem Bildschirm erscheint dann die Anzeige **Password:**, und als Antwort darauf geben Sie "Redwing" ein und drücken ⟨**ENTER**⟩. Ihr Passwort wird nicht angezeigt, aber dem Host-System übermittelt.

9. Warten Sie mindestens drei Sekunden, bevor Sie die Frage OK? eingeben. Wenn alles normal abläuft, erhalten Sie die Nachricht **Log-on successfully completed**, und der M10 wird an das IDS-System angeschlossen.

DIE UPLOAD- UND DOWNLOAD-EINRICHTUNGEN

In der Einführung zur Bedienung im Terminal-Modus oben in diesem Kapitel haben wir schon auf diese beiden Möglichkeiten bei der Beschreibung der Funktionstasten hingewiesen. Um diese Möglichkeiten zu nutzen, braucht der Bediener aber noch weitere Einzelheiten.

Die Upload-Einrichtung

Diese Option ermöglicht es dem Benutzer, im Voraus einen Datensatz von Informationen vorzubereiten, der anschließend einem anderen Computer gesendet werden soll. Hierzu muß man folgendermaßen vorgehen:

1. Gehen Sie nach der in Kapitel 5 erklärten Methode zur Erstellung einer TEXT-Datei vor, die die Daten enthält, die einem anderen Computer übermittelt werden sollen. Speichern Sie diese Daten in RAM.

2. Stellen Sie die Kommunikation mit dem anderen Computer über das TELCOM-Programm her, und starten Sie die Befehlsfolge, die der Host-Computer braucht, um Daten zu empfangen. Dann drücken Sie ⟨**F3**⟩ (Up).

   Hierauf folgt die Anzeige:

   File to upload?

3. Geben Sie den Namen der Datei an, die Sie vorher erstellt haben, und drücken Sie ⟨**ENTER**⟩.

4. Nun erscheint die Anzeige **Width?**, und Sie geben einen Wert zwischen 10 und 132 ein und drücken ⟨**ENTER**⟩. Dies bestimmt die Zeilenbreite der zu ladenden Datei. Die physische Breite einer M10-Datei beträgt 40 Zeichen (die Breite des Bildschirms). Bei fortlaufendem Text macht jedoch die Einrichtung zum automatischen Zeilensprung eine Zeilenschaltung per Hand unnötig. Eine Textzeile kann daher das vielfache dieser Zeilenbreite betragen. Wenn der M10 nicht auf eine Zeilenschaltung stößt, die der Anzahl von Zeichen entspricht, die Sie als Antwort auf die Anzeige **Width** eingegeben haben, wird er selbst eine vornehmen. Alle in der Datei vorgefundenen Zeilenschaltungen werden gesendet. Wenn Sie als Antwort auf die Anzeige die Taste ⟨**ENTER**⟩ drücken, werden keine zusätzlichen Zeilenschaltungen vorgenommen.

Die Download-Einrichtung

Diese Möglichkeit erlaubt es dem Benutzer, hereinkommende Daten von einem Host-Computer zwischenzuspeichern (download), d. h. sie in einer Datei für zukünftige

Absichten zu speichern. Dieses ist speziell dann nützlich, wenn die Daten in einem dichten Zeichenfluß kommen und anschließende Interpretation benötigen. Um diese Option in Gang zu setzen, gehen Sie folgendermaßen vor:

1. Da Sie bereits mit einem Host-Computersystem kommunizieren, befindet sich der M10 im TELCOM-Programm und im Terminal-Modus.

   Geben Sie dem Host-Computer den Befehl, mit dem Senden des Datenpakets zu beginnen, aber **drücken Sie nicht die Taste** ⟨**ENTER**⟩.

   Drücken Sie ⟨**F2**⟩ (Down = nach unten).

   Dieses ruft die Mitteilung hervor:

   File to download?

2. Geben Sie den Namen des Files an, in die die Information gespeichert werden soll, und drücken Sie **ENTER**. Es ist nicht nötig, daß Sie schon vorher ein File erstellt haben. Wenn Sie unter Berücksichtigung der Regeln für die Nomenklatur von Text-Files einen Namen auswählen, der nicht mehr als 6 Zeichen beträgt, wird er der Name einer erstellten Text-Datei und in RAM gespeichert.

3. Drücken Sie ⟨**ENTER**⟩, um den Befehl an das Host-System zu komplettieren, und beginnen Sie den Download-Prozess.

4. Während dieser Prozeß im Gange ist, erscheint in der Fußzeile des Bildschirms die Bezeichnung **"Down"** in Reverse-Schreibung, womit der Computer anzeigt, daß die ankommenden Daten zwischengespeichert werden.

5. Wenn Sie die Daten, die Sie benötigen, gespeichert haben, beenden Sie den Download-Prozeß mit der Taste ⟨**F2**⟩.

Wenn Sie die so gespeicherten Daten lesen und bearbeiten wollen, müssen Sie ⟨**F8**⟩ (Bye) drücken, um so die Kommunikation mit dem Host-System zu beenden. Wenn die Bildschirmanzeige **Disconnect?** erscheint, drücken Sie

die Taste ⟨**Y**⟩, dann ⟨**ENTER**⟩, um den M10 von dem Telefonnetz abzukoppeln und in den Eingabe-Modus zurückzukehren. Jetzt drücken Sie ⟨**F8**⟩ und kehren ins Hauptmenü zurück. Dann wählen Sie das File, das die gespeicherten Daten enthält. Diese Daten können Sie aufrufen und bearbeiten, wie Sie das auch mit einem TEXT-File machen würden. Eine solche Bearbeitung ist durchaus erwünscht, wenn das Host-System bestimmte Kontrollzeichen in die übermittelte Information eingebaut hat.

Die Datei (File), die Sie mit der Download-Funktion gespeichert haben, ist ein BASIC-Programm-File. Deshalb müssen Sie **BASIC** eingeben und das File laden. Dieses wird in Binärform konvertiert und ermöglicht es Ihnen, es als .BA-File laufen zu lassen und abzuspeichern. Wenn Sie es als BASIC-File speichern, kann das .DO-File gelöscht werden.

9. DER GEBRAUCH EINES KASSETTENREKORDERS MIT DEM M10

Wie schon mehrmals erwähnt, können die auf dem M10 erstellten Dateien und Programme auf Kassettenrekordern abgespeichert werden. Diese Einrichtung ist sehr praktisch, wenn Sie Sicherungskopien Ihrer Dateien erstellen wollen und dadurch den Arbeitsspeicher zum Abspeichern neuer Dateien freimachen wollen.

Die M10 Anwender-Programme BASIC und TEXT beinhalten die notwendigen Befehle, um eine Datei auf Kassette abzuspeichern und um eine Datei von Kassette einzulesen. In dieser Hinsicht kann der Gebrauch eines Kassettenrekorders als eine Standardeinrichtung des M10 betrachtet werden. Das vorliegende Kapitel ist speziell dieser Möglichkeit gewidmet.

DIE VERBINDUNG DES M10 MIT EINEM KASSETTENREKORDER

Wenn Sie den M10 mit einem Kassettenrekorder verbinden, haben Sie die Möglichkeit, entweder Dateien von dem M10 auf Kassette zu übertragen (speichern), oder Dateien von der Kassette in den M10 einzulesen. Bevor Sie einen Rekorder anschließen, legen Sie eine Kassette ein und spulen Sie sie in die Position, von der an die Aufzeichnung beginnen soll. Halten Sie mit dem Bandzählwerk die Startposition fest. Beim Speichern von Dateien auf Kassette sollten Sie immer das Bandzählwerk beachten. Hierdurch wird das Auffinden von gespeicherten Informationen sehr vereinfacht. Sie können jeden Standardkassettenrekorder benutzen, vorausgesetzt, er ist außer mit den zu der normalen Einrichtung eines Rekorders gehörenden Einrichtungen wie Aufnahme, Wiedergabe, Stop, Kassettenauswurf, Vor- und Rückspulen auch mit den folgenden ausgestattet:

- Mikrofoneingang MIC
- Ausgang für Kopfhörer EAR
- Eingang für Fernsteuerung REM
- Bandzählwerk

Die Verbindung zwischen dem M10 und dem Kassettenrekorder sind in Abbildung 9-1 dargestellt. Das Verbindungskabel können Sie von Ihrem OLIVETTI-Händler erhalten.

Dieses Kabel hat an seinem einen Ende drei Stecker zum Anschluß an den Kassettenrekorder und am anderen Ende einen runden Stecker mit 8 Kontakten zum Anschluß an den M10. Die drei Stecker sind rot, weiß und schwarz. Der schwarze Stecker hat einen deutlicheren Kontakt als die anderen.

1. Stecken Sie den Stecker mit den 8 Kontakten am einen Ende in die Buchse mit der Beschriftung TAPE an der Rückseite des M10.

2. Stecken Sie den weißen Stecker in den Ausgang mit der Bezeichnung EAR.

3. Stecken Sie den schwarzen Stecker in die Fernsteuerungsbuchse REM.

4. Stecken Sie den roten Stecker in den Mikrofoneingang MIC.

5. Schalten Sie den Kassettenrekorder und den M10 ein.

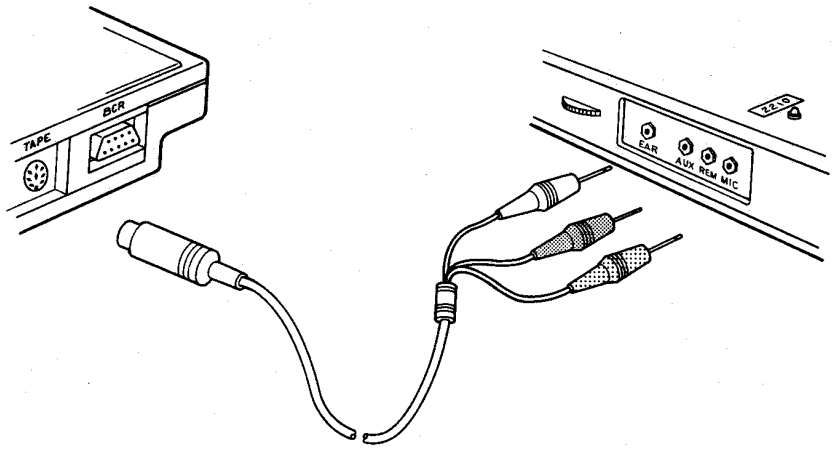


Abbildung 9-1 Verbindung von M10 und Kassettenrekorder

Wenn der Recorder auf diese Weise verbunden ist, können zwei Signale von dem M10 gesendet werden:

MOTOR ON - Wenn dieses Signal gesendet wird, erfolgt die Steuerung der Rekorderfunktionen durch die Rekorder-Tasten.

MOTOR OFF - Dieses Signal setzt die Rekorder-Tasten außer Funktion und stellt den Rekorder voll unter die Kontrolle des M10. Der Motor startet und stopt automatisch, wenn Dateien übertragen werden.

Diese beiden Befehle sind BASIC-Befehle (siehe auch im Band 2, BASIC-Handbuch). Der Befehl MOTOR OFF muß erteilt werden, bevor man lädt oder speichert.

SPEICHERN EINER DATEI ODER EINES BASIC-PROGRAMMS AUF KASSETTE

Das Speichern einer .DO-Datei (Text-Datei) erfolgt mit den TEXT-Funktionstasten. Eine .BA-Datei (BASIC-Programm) wird mit den BASIC-Funktionstasten gespeichert. In beiden Fällen spulen Sie zuerst das Band an die Position, von der an Sie Ihre Datei bzw. Ihr Programm speichern wollen. Verbinden Sie den Rekorder, wie es im vorhergehenden Kapitel beschrieben wurde, schalten Sie ihn an und drücken Sie die Tasten **RECORD** und **PLAY** gleichzeitig.

SPEICHERN EINER TEXTDATEI AUF KASSETTE

1. Bringen Sie den Cursor auf TEXT im Hauptmenü, und drücken Sie die Taste ⟨**ENTER**⟩.

2. Drücken Sie die Funktionstaste ⟨**F3**⟩ (Speichern); dies bringt die Bildschirmanzeige:

   Save to:

   Wählen Sie einen Dateinamen für die Datei, die gespeichert werden soll. Diese Bezeichnung muß den Namensvorschriften für M10-Dateien entsprechen (siehe Kapitel 5).

   Nun geben Sie den gewählten Dateinamen ein und drücken ⟨**ENTER**⟩.

4. Der Rekorder startet jetzt automatisch. Wenn die Datei gespeichert ist, stopt der Rekorder, und die Anzeige verschwindet vom Bildschirm.

Die Datei ist nun unter ihrem Dateinamen auf dem Kassettenrekorder gespeichert.

SPEICHERN EINES BASIC-PROGRAMMS AUF KASSETTE

Die Vorgehensweise bei BASIC-Programmen ist etwas anders.

1. Rufen Sie BASIC vom Hauptmenü auf.

2. Drücken Sie die Funktiontaste ⟨**F2**⟩ (Laden). Dann erscheint die Anzeige" **Load** ". Geben Sie jetzt den RAM- Programmnamen des Programms, das gespeichert werden soll, ein, und drücken Sie ⟨**ENTER**⟩.

3. Nun drücken Sie ⟨**F3**⟩ (Speichern). Jetzt erscheint die Bildschirmanzeige **Save** ", und Sie geben ein:

   CAS:Programmname ⟨**ENTER**⟩

   Der "Programmname" ist der Name für das Programm, das auf der Kassette gespeichert werden soll. Dieser Name muß entsprechend den Regeln für die Bildung von M10-Dateinamen gewählt werden.

Zwischen dem Doppelpunkt von CAS: und dem Kassetten-Programmnamen sollte kein Leerzeichen stehen.

Der Rekorder läuft nun, bis die "Save"- (Speicher-) Operation beendet ist, und stopt dann. Das Programm ist nun unter dem gewählten Kassetten-Programmnamen auf Band gespeichert.

DAS LADEN EINER TEXT-DATEI ODER EINES BASIC-PROGRAMMES VON KASSETTE

Diese Prozedur variiert wiederum etwas, je nachdem, ob es eine TEXT-Datei oder ein BASIC-Programm ist, das geladen werden soll.

DAS LADEN EINER TEXT-DATEI VON KASSETTE

1. Spulen Sie die Kassette bis zum Anfang der Datei zurück, die in den M10 geladen werden soll. Wenn Sie nicht wissen, wo der gesuchte Text anfängt, spulen Sie die Kassette ganz bis zum Anfang zurück; dann sucht der M10 die gesamte Kassette später nach Ihrer gewünschten Datei durch.

2. Verbinden Sie den Rekorder mit dem M10 wie vorher beschrieben. Schalten Sie den Rekorder ein, und drücken Sie die PLAY-Taste. Der Rekorder startet erst dann, wenn er das Signal dazu von dem M10 bekommen hat.

3. Sie können eine spezielle TEXT-Datei im RAM erstellen, in die die von dem Kassettenrekorder überspielte Datei empfangen wird, oder Sie können eine bereits bestehende Datei im RAM wählen, an die die Kassettendateien angehängt werden soll.

   Drücken Sie die Funktionstaste ⟨**F2**⟩. Es erscheint die Mitteilung auf dem Bildschirm:

   Load from:"

4. Geben Sie den Dateinamen ein, unter dem der Text auf Kassette früher gespeichert war, und drücken Sie dann ⟨**ENTER**⟩.

5. Der M10 beginnt nun, die ganze Kassette nach der gewünschten Datei zu durchsuchen. Während dieses Vorganges gibt er einen hohen Ton von sich. Jedesmal, wenn er auf eine Datei stößt, die nicht die gewünschte ist, zeigt er die Bildschirmnachricht an:

   SKIP: Dateiname

   Hierbei ist Dateiname der Name einer Datei oder eines Programms, das er auf der Kassette gefunden hat.

   Wenn der gesuchte Text aufgefunden ist, zeigt der M10 an:

   FOUND: Dateiname

Wenn diese Nachricht erscheint, ist die Kassettendatei in die TEXT-Datei im RAM geladen worden, von der aus Sie sie aufgerufen haben. Bei einem Ladevorgang ersetzt die Kassettendatei nicht den Inhalt dieser TEXT-Datei im RAM, sondern hängt sie an das Ende der TEXT-Datei an.

DAS LADEN EINES BASIC-PROGRAMMS VON DER KASSETTE

1. Spulen Sie die Kassette bis zu der Position zurück, an der das Programm etwa beginnt. Wenn Sie diese nicht wissen, spulen Sie zum Bandbeginn zurück. Verbinden Sie dann den Rekorder mit dem M10 und drücken Sie die Taste PLAY, wie im vorigen Kapitel in den Schritten 1 und 2 gezeigt wurde.

2. Rufen Sie vom Hauptmenü BASIC auf, und drücken Sie ⟨**F2**⟩. Jetzt sehen Sie die Bildschirmanzeige:

   LOAD "

3. Geben Sie ein:

   CAS:Programmname ⟨**ENTER**⟩

   Hierbei ist "Programmname" der Name, unter dem das BASIC-Programm früher auf Kassette gespeichert wurde.

4. Sie hören jetzt wieder einen hohen Ton, da der M10 nach dem angegebenen Programm auf der Kassette sucht. Die Mitteilung

   SKIP: Dateiname

   wird jedesmal angezeigt, wenn eine nicht gewünschte Datei bzw. ein nicht gewünschtes BASIC-Programm gefunden wird, und

   FOUND: Programmname

   wird angezeigt, wenn das gesuchte Programm aufgefunden wurde.

Achten Sie bitte darauf, daß es bei den Funktionen "Load" (Laden) und "Save" (Speichern) in BASIC nötig ist, als Antwort auf die Bildschirmmeldung nicht nur

den Dateinamen zu spezifizieren, sondern auch den Gerätenamen, in diesem Falle CAS: (Der Doppelpunkt muß mit angegeben werden). Der Grund dafür ist, daß der BASIC-Interpreter Zugang zu anderen externen Geräten hat und entweder dort oder vom RAM laden und entweder auf externem Gerät oder im RAM speichern kann. Der "Load"- und "Save"-Vorgang kann auch durch einen direkten Befehl bewirkt werden, wenn der BASIC-Interpreter geladen ist (also nach Anwahl von BASIC im Menü). Dann können Sie auch die BASIC-Befehle CLOAD und CSAVE benutzen (z.B in einem BASIC-Programm selbst). Eine genauere Beschreibung dieser Befehle finden Sie in Band 2 (BASIC-Handbuch).

GERÄTEWARTUNG

Wenn Sie einen an den M10 angeschlossenen Kassettenrecorder regelmäßig benutzen, müssen Sie unbedingt den Empfehlungen des Herstellers zur Wartung und Säuberung des Gerätes folgen. Überzeugen Sie sich besonders davon, daß die Tonköpfe des Rekorders sauber und entmagnetisiert sind, damit Ihre Computer-Dateien nicht beschädigt werden und keine "Störgeräusche" (Rauschen) auftauchen.

10. EINFÜHRUNG IN BASIC

Die nächsten Kapitel sind ganz BASIC gewidmet (Beginners All-purpose Symbolic Instruction Code), der Programmiersprache des M10. Es existieren unterschiedliche Versionen von BASIC, aber alle Beschreibungen und Verweise in diesem Handbuch beziehen sich auf die Version, die den M10 betrifft. In Kapitel 4 wurde eine kurze Beschreibung des Programms "BASIC-Interpreter" gegeben. Daher ist der Benutzer schon mit gewissen Aspekten des M10-BASIC vertraut, wie z. B. mit dem Zugriff auf BASIC, den drei verschiedenen Möglichkeiten des Umgangs mit BASIC, den Programmzeilen, der Rolle der Funktionstasten usw. Die folgenden Kapitel beschäftigen sich mit dem Schreiben und Abarbeiten von Programmen, mit Daten und arithmetischen, relationalen und logischen Operatoren. Am Ende wird dann ein Nachschlageverzeichnis der BASIC-Befehle aufgestellt.

SCHREIBWEISEN

Die Konventionen zur Schreibweise, die in Kapitel 1 aufgestellt wurden, gelten auch in diesem Teil des Handbuches. Aus Gründen der BASIC-Syntax kommt jedoch in diesem Teil noch eine Erweiterung hinzu.

Die folgenden Zeichen kommen in der BASIC-Syntax vor und sollten so eingegeben werden, wie sie da stehen.

Anführungszeichen	"
Punkt	.
Komma	,
Doppelpunkt	:
Semikolon	;
Plus-Zeichen	+
Minus-Zeichen	-
Multiplikations-Zeichen	*
Divisions-Zeichen	/

Exponential-Symbol	^
Schrägstrich links	\
Gleichheitszeichen	=
Größer als	>
Kleiner als	<
Fragezeichen	?
Prozentzeichen	%
Ausrufungszeichen	!
Nummernzeichen	#
Dollarzeichen	$
Klammern	( )

11. BASIC-PROGRAMME

ORGANISATION

BASIC-Programme für den M10 werden im Arbeitsspeicher gespeichert. Bis zu 19 Programmnamen können gleichzeitig im Arbeitsspeicher vorhanden sein. Programme können ebenfalls auf Kassette oder anderen externen Einheiten abgespeichert werden. Es hat sich bewährt, zwei getrennt abgespeicherte Kopien eines Programmes aufzubewahren; eine für den normalen Gebrauch und eine Sicherungskopie.

DOKUMENTATION EINES PROGRAMMES

Es ist außergewöhnlich wichtig, daß Programme auf eine logische Art und Weise erstellt werden und daß sie gut dokumentiert werden. Eine Prozedur, die zum Zeitpunkt der Erstellung übersichtlich und klar ist, kann zu einem späteren Zeitpunkt durchaus undurchsichtig erscheinen, wenn nicht entsprechende Anmerkungen hinzugefügt werden. Dies kann mit einer REM-Anweisung oder mit dem Kürzel ', dem Apostroph ('=Hochkomma), geschehen.

REM

Anmerkungen können eine eigene Programmzeile in Anspruch nehmen, oder sie können in derselben Zeile mit anderen Anweisungen stehen. Im letzteren Fall muß die Anmerkung am Ende einer Programmzeile stehen.

ERSTELLUNG EINES PROGRAMMS

Der Anwender erstellt ein BASIC-Programm auf dem M10, indem er die einzelnen Anweisungen in Form von Programmzeilen über die Tastatur eingibt. Die eingegebenen Programmzeilen können mit der LIST-Anweisung auf dem Display aufgelistet werden, um sie nochmals zu kontrollieren und gegebenenfalls zu korrigieren. Programme können jederzeit auf Kassette gespeichert und von Kassette geladen werden.

EINGABE EINES PROGRAMMS

Jede Programmzeile muß mit einer Zeilennummer beginnen, die über die Tastatur eingegeben wird.

Programmzeilen können in jeder beliebigen Reihenfolge eingegeben werden. Die LIST-Anweisung ordnet die Programmzeilen und zeigt sie in der korrekten Reichenfolge auf dem Display an. Solch eine Auflistung kann zu jedem Zeitpunkt vorgenommen werden. Das Programm kann ebenfalls zu jedem Zeitpunkt seiner Erstellung gespeichert werden (siehe Speichern eines Programms). Alle Befehle, Anweisungen und Funktionen, die auf dem M10 benutzt werden können, sind im Detail in Kapitel 15 beschrieben.

AUFLISTEN EINES PROGRAMMS

Um ein Programm auf dem Bildschirm aufzulisten, steht Ihnen der Befehl LIST zur Verfügung. Der Befehl LLIST listet das Programm auf dem Drucker.

LIST kann zur Darstellung einer oder mehrerer Programmzeilen auf dem Bildschirm wie folgt genutzt werden:

LIST n -m **ENTER**

Hierbei ist n die erste Zeile, die gelistet wird, und m ist die letzte Zeile, die angezeigt wird. Wenn n nicht bezeichnet ist, werden alle Zeilen bis m aufgelistet; und wenn m nicht angegeben ist, wird nur die Zeile n gelistet. Wird nur der Befehl LIST eingegeben, wird das ganze Programm aufgelistet.

LIST. listet die zuletzt bearbeitete Programmzeile.

Ein Programm, dessen Anweisungen mit LIST auf dem Display dargestellt werden, ist im Speicher abgelegt. Eine Fehlerkontrolle der einzelenen Anweisungen ist jedoch noch nicht erfolgt. Dies geschieht nach dem RUN-Befehl zum Starten eines Programms während der Abarbeitung der einzelnen Programmzeilen.

Der Befehl LLIST arbeitet auf genau die gleiche Art und Weise, aber die Programmzeilen werden auf dem Drucker aufgelistet.

Einzelheiten von LIST und LLIST werden in Kapitel 15 gegeben.

SPEICHERN EINES PROGRAMMS

Ein Programm, das im Arbeitsspeicher vorhanden ist, kann auf eine Kassette oder irgendeinem anderen externen Datenspeicher, der an den M10 angeschlossen ist, gespeichert werden. Dem Befehl SAVE stehen eine Anzahl von Optionen für unterschiedliche Peripheriegeräte zur Verfügung. Diese sind in Kapitel 15 im einzelnen beschrieben.

Um im Arbeitsprogramm (ein Programm an oder mit dem Sie momentan arbeiten) auf Kassette zu speichern, können Sie den Befehl CSAVE"tape filename" benutzen. "tape filename" ist gleichbedeutend mit dem Programmnamen Ihres Programmes im Arbeitsspeicher, lediglich kennzeichnet es Ihr Programm auf der Kassette. Benutzen Sie, wenn möglich, den gleichen Programmnamen zur Speicherung im RAM und auf Kassette.

Haben Sie Ihr Programm mit CSAVE "tape filename" auf Kassette gespeichert, so können Sie mit dem Befehl CLOAD? "tape filename" die Datenübertragung auf Kassette kontrollieren. CLOAD? vergleicht Ihr Programm im Arbeitsspeicher mit der Kassettenaufzeichnung. Erscheint während der Überprüfung die Mitteilung **verify failed** auf dem Bildschirm, müssen Sie die Datenabspeicherung mit CSAVE "tape filename" auf Kassette wiederholen. Ihr Programm wurde fehlerhaft übertragen.

Erscheint auf dem Bildschirm die Anzeige **Ok** und das Kassettenlaufwerk hält an, so ist Ihr Programm korrekt auf Kassette übertragen.

Wenn Sie das Abspeichern eines Programmes unterbrechen wollen, müssen Sie die Taste **SHIFT + BREAK** drücken. Einzelheiten von CSAVE und CLOAD? werden in Kapitel 15 gegeben.

DAS ANWENDEN EINES PROGRAMMS

Wenn Sie das Programm benutzen wollen, muß es zuerst in den Arbeitsspeicher geladen werden, bevor es ausgeführt wird. Wenn es Probleme bei dem Ablauf der Programme gibt, die den Ablauf unterbrechen, z. B. Syntax-Fehler, werden sie durch das System angezeigt. Das Korrigieren von Fehlern in Ihrem Programm wird weiter unten beschrieben.

DAS LADEN EINES PROGRAMMS

Beim Befehl LOAD stehen verschiedene Optionen zur Verfügung, so daß Programme von verschiedenen Peripheriegeräten in den Arbeitsspeicher geladen werden können. Diese Varianten werden detailliert in Kapitel 15 beschrieben. Programme, die mit CSAVE auf Kassette abgespeichert wurden, können mit dem Befehl CLOAD "tape filename" in den Arbeitsspeicher geladen werden. Hierbei ist "tape filename" der Name Ihres Programms, unter dem es auf Kassette abgespeichert ist. Wenn Sie "tape filename" weglassen, wird das erste gefundene Programm auf der Kassette in den Arbeitsspeicher geladen. Um den Ladevorgang zu unterbrechen, drücken Sie **SHIFT + BREAK** und geben den Befehl MOTOR OFF ein, um das Kassettenlaufwerk zu stoppen.

Es besteht die Möglichkeit, mit einem einzigen Befehl ein Programm von einer Kassette zu laden und zu starten. Diese Einzelheit wird weiter unten beschrieben.

ABLAUF EINES PROGRAMMS

Um ein Programm ablaufen zu lassen, benutzen Sie den Befehl RUN. Geben Sie den Befehl ein

RUN **ENTER**

Hierdurch starten Sie das Programm im Arbeitsspeicher. Wenn das Programm sich nicht im Speicher befindet, aber auf einer Kassette abgespeichert ist, kann es mit dem folgenden Befehl geladen und gestartet werden:

RUN "CAS:tape filename" **ENTER**

Wenn es irgendwelche Probleme mit dem Programm gibt, zeigt das System einen Fehler (Error) an: Die Prozedur zur Korrektur von Fehlern wird unten beschrieben.

Wenn das Programm von dem Benutzer Informationen in Form von Daten benötigt, erscheint die Anzeige ?. Normalerweise geht diesem Fragezeichen eine Mitteilung voraus, die erklärt, welche Daten benötigt werden. Wenn Sie wissen wollen, wie man diese Erklärung in ein Programm schreibt, schlagen Sie in Kapitel 15 unter INPUT nach. Geben Sie jetzt die benötigten Daten ein, und drücken Sie die Taste **ENTER** .

DRUCK

Durch PRINT oder LPRINT können Programme entweder auf dem Bildschirm, dem Drucker oder auf einem Microplotter angezeigt werden.

Die Anweisung PRINT zeigt Daten auf dem Bildschirm an. LPRINT (L steht für Lineprinter (= Drucker) gibt die Ausgabewerte auf den Drucker.

Mit Hilfe der Anweisungen PRINT USING oder LPRINT USING (= Drucker) ist es möglich, Daten in einem spezifizierten Format auszugeben. Wird PRINT USING im Zusammenhang mit dem Schreiben von Daten in ein Datenfile verwendet, so ist es ebenfalls mögliche, diese Daten in einem spezifizierten Format in die Datei zu schreiben.

Um Daten auf einem seriellen Drucker (verbunden mit der RS-232-Schnittstelle) auszugeben, müssen diese in eine Datei geschrieben werden: Dieses wird in Kapitel 15 erklärt.

KORREKTUR VON FEHLERN

Fehler, die die Programmausführung unterbrechen, werden vom Interpreter mit einer Fehlermeldung und der Zeilennummer, in der sie auftraten, angezeigt, so daß sie leicht korrigiert werden können.

FEHLER (ERROR)

In Anhang B finden Sie eine vollständige Liste der Fehlermeldungen. Jede Fehlermeldung besteht aus einem Fehlercode (2 Buchstaben als Abkürzung) und einer Fehlernummer zwischen 1 und 255. Viele dieser Fehlernummern werden nicht genutzt. Sie stehen für eine Systemerweiterung in BASIC zur Verfügung, oder der Benutzer kann sie zur Definition eigener Fehler einsetzen. Zusätzlich zu den Fehlern, die bei der Erstellung eines Programmes gemacht werden können, gibt es eine Anzahl von Fehlern, die in einer anderen Situation auftauchen können, z. B. bedeutet IO Error, daß beim Lesen oder Schreiben von Daten ein Eingabe- oder Ausgabefehler gemacht wurde.

Durch die Anweisung ERROR n kann der Benutzer selbst einen Fehler simulieren. Hierbei ist n die Nummer des spezifizierten Fehlers. Einzelheiten zu dieser Anweisung werden in Kapitel 15 gegeben.

FEHLERBEHANDLUNGSROUTINE

In ein Programm können auch Unterroutinen (Unterprogramme) für das Vorgehen beim Auftauchen von Fehlern geschrieben werden. Die Anweisung ON ERROR GOTO n wird benutzt, um die erste Zeile, n, einer solchen Routine anzugeben. Wenn der Befehl ON ERROR GOTO einmal erteilt wurde, springt das Programm jedesmal in diese Unterroutine, wenn ein Fehler auftaucht. Eine aktive Fehlerbehandlungsroutine wird durch die Anweisung ON ERROR GOTO Ø aufgehoben.

PROGRAMM-MODIFIKATION

Ein Programm kann nur dann modifiziert werden, wenn es im Arbeitsspeicher vorhanden ist; wenn es sich im Moment nicht im Arbeitsspeicher befindet, muß es zuerst geladen werden, wie oben beschrieben wurde.

ÄNDERN EINES PROGRAMMES

Wenn Programmzeilen geändert werden sollen, können Sie den EDIT-Befehl benutzen oder die komplette Zeile neu eingeben und anschließend die Taste **ENTER** drücken. Die neue Zeile ersetzt dann die alte im Arbeitsspeicher. Wenn Sie nur eine bestimmte Zeile löschen wollen, müssen Sie die Zeilennummer eingeben und anschließend **ENTER** drücken.

ERSTELLUNG EINES PROGRAMMS IM EDIT-MODUS

Für Änderungen innerhalb eines Programmes ist es einfacher, den EDIT-Befehl zu benutzen. Der Befehl EDIT bringt das System in den TEXT-Modus. Einige Einzelheiten über diesen Befehl werden in Kapitel 15 erläutert. Ausführlich wird der Umgang mit dem TEXT-Modus in Band 1 erklärt.

Der TEXT-Modus liefert einen Bildschirm-Editor, der zur Erstellung von TEXT-Dateien und Programmen benutzt wird. In diesem Modus stehen Ihnen eine Anzahl von Befehlen zur Verfügung, die die Erstellung und Modifi-

zierung von Dateien und Programmen vereinfachen. Eine ausführliche Beschreibung dieses Modus wird in Band 1 gegeben.

VERKNÜPFEN VON ZWEI PROGRAMMEN

Ein Programm, das im ASCII-Format (CSAME "tape filename",A) auf Kassette abgespeichert wurde, kann mit einem programm im Arbeitsspeicher durch den Befehl

```
MERGE "CAS: tape filename"
```

verknüpft werden.

Mit diesem Befehl werden die Programmzeilen auf der Kassette mit denjenigen im Speicher verknüpft und in numerischer Reihenfolge geordnet. Wenn eine Programmzeile von Kassette dieselbe Zeilennummer hat wie eine Programmzeile im Arbeitsspeicher, wird die letztere ersetzt, und die Originalzeile geht verloren.

12. DATEN UND ARITHMETIK

DATEN IN BASIC-PROGRAMMEN

Daten können in Programmen als Konstanten oder Variablen auftreten. Eine Konstante hat den gleichen Wert innerhalb der Programmdurchführung, hingegen kann eine Variable verändert werden. Variablen erhalten deshalb beim Schreiben eines Programms einen Namen.

So ergibt z. B. die Formel Temperatur Kelvin = Temperatur Celsius + 273 Kelvin

```
C + 273
```

eine Temperatur in Kelvin, wobei C eine Variable darstellt, die die Temperatur in Celsius angibt. Ein Programm mit diesem Ausdruck kann beliebig oft mit jeweils unterschiedlichen Werten für C ablaufen.

Variablennamen können länger als zwei Zeichen sein, jedoch werden nur die ersten beiden Zeichen identifiziert und müssen deshalb unverwechselbar sein. Z. B.:

```
Ok
AB=2
Ok
ABC=5
Ok
PRINT AB
5
Ok
PRINT ABC
5
Ok
```

Auch eine Konstante kann einen Namen erhalten. Es ist z. B. einfacher, einen kurzen Namen einzugeben als eine lange Konstante. Z. B.:

```
Ok
1Ø PRINT 3.141593 + 3.141593 + 2.718282
RUN
9.ØØ1467
Ok
1Ø PI=3.141593:E=2.718282
2Ø PRINT 2*PI+E;"ist besser"
RUN
9.ØØ1467 ist besser
Ok
```

Variablennamen dürfen keine reservierten BASIC-Wörter enthalten, z. B.: IF=10 oder ON=3.14. Wenn man nur Variablennamen mit maximal 2 Zeichen verwendet, sind nur die Variablen IF, ON, OR und TO auszuschließen.

Sowohl Konstanten als auch Variablen können entweder reine Zahlen oder aber "Strings" sein. Ein String ist eine Folge beliebiger Zeichen mit Ausnahme des Anführungszeichens ("), da dieses benutzt wird, den String zu definieren oder zu begrenzen. Z. B.:

```
Ok
A$="Olivetti"
Ok
B$="Personal Computer M10"
Ok
PRINT B$
Personal Computer M10
Ok
```

Ein "Ausdruck" ist ein allgemeiner Name für ein Zeichen, eine Zeichenkombination oder Daten. Er kann Konstanten oder Variablen oder auch beide enthalten, jedoch kann er keine Mischung von numerischen und String-Daten enthalten. Die ganz oben genannte Formel ist ein numerischer Ausdruck, und die numerische Variable C muß einen Wert haben, bevor der Ausdruck berechnet werden kann.

Richtig:
```
A=B+C
A$=B$+C$
```

Falsch:
```
A=b$+C
```

STRING-DATEN

String-Daten haben eine variable Länge von Ø - 255 darstellbaren Zeichen, wobei BASIC unabhängig von der String-Länge den benötigten Speicherplatz zur Verfügung stellt. Enthält ein String keine Zeichen, also dargestellt durch zwei Anführungszeichen ohne ein Zeichen dazwischen, wird er "Null"- ("Leer"-) String genannt.

NUMERISCHE DATEN

BASIC behandelt numerische Daten auf drei verschiedene Arten, abhängig von der Genauigkeit, die für die Berechnung erforderlich ist, nämlich integer (ganzzahlig), single precision (einfache Genauigkeit) oder double precision (doppelte Genauigkeit). Alle numerischen Daten werden in Dezimalform angegeben (d. h. zur Basis 10).

Integers (ganzzahlige Zahlen)

Sie sind die speicherplatzsparensten, und in der Berechnungsgeschwindigkeit die schnellsten Variablen, die BASIC zur Verfügung stellt. Jedoch sind die Werte begrenzt auf ganze Zahlen zwischen -32768 und 32767.

Single Precision Numbers
(Zahlen mit einfacher Genauigkeit)

Sie werden für allgemeine Anwendungen in BASIC benutzt. Der Zahlenbereich liegt zwischen $\pm 10^{-64}$ und $\pm 10^{62}$. Für die Berechnung werden bis zu sieben signifikante Stellen benutzt. Es können bis zu sechs Ziffern angezeigt werden, wobei die letzte Stelle aufgerundet wird, wenn diese Zahl größer oder gleich 5 ist. Für Zahlen mit einfacher Genauigkeit wird mehr Speicherplatz benötigt, und die Berechnung dauert länger als für ganzzahlige.

Double Precision Numbers
(Zahlen mit doppelter Genauigkeit)

Dies erlaubt die genaueste Darstellung von Daten in BASIC. Der Wertebereich liegt ebenfalls zwischen $\pm 10^{-64}$ und $\pm 10^{62}$; aber bis zu 16 signifkante Stellen können für die Berechnung benutzt werden. bis zu 15 Zeichen können angezeigt werden, wobei das letzte Zeichen aufgerundet wird, wenn es größer oder gleich 5 ist.

Zahlen mit doppelter Genauigkeit benötigen den meisten Speicherplatz, und die Berechnung dauert am längsten.

EINTEILUNG DER DATEN

Normalerweise wird ein String daran erkannt, daß er im Programm in Anführungszeichen eingeschlossen angegeben wird. Während der Dateneingabe oder in einer DATA-Anweisung ist es normalerweise nicht nötig, die Zeichen in Anführungszeichen einzuschließen.

Numerische Konstanten werden durch die niedrigstmögliche Kategorie klassifiziert, z. B.:

- wenn eine Zahl mehr als sieben Zeichen hat, ist es eine Zahl mit doppelter Genauigkeit

- ist eine ganze Zahl im Bereich zwischen -32768 und 32767, so ist sie eine Ganzzahl

- gehört eine Zahl zu keiner dieser beiden Kategorien, ist es eine Zahl mit einfacher Genauigkeit.

Numerische Variablen werden als Zahlen mit doppelter Genauigkeit klassifiziert, wenn an den Variablennamen nicht zusätzlich ein besonderes Zeichen angehängt wird, das eine andere Klassifizierung vorschreibt.

ÄNDERUNG DER KLASSIFIZIERUNG

Sowohl Konstanten als auch Variablen können unterschiedlich klassifiziert werden. Der Typ einer Konstante kann nur verändert werden, wenn an das Ende der Daten ein Typkennzeichen angehängt wird. Die vier zur Verfügung stehenden Typkennzeichen werden in Abbildung 12-1 dargestellt.

Variablen können am Start eines Programms als bestimmter Typ definiert werden, indem eine der folgenden Anweisungen benutzt wird:

- DEFINT Anfangsbuchstabe(n) legt Ganzzahl-Variablen fest

- DEFSNG Anfangsbuchstabe(n) legt Variablen mit einfacher Genauigkeit fest

- DEFDBL Anfangsbuchstabe(n) legt Variablen mit doppelter Genauigkeit fest

- DEFSTR Anfangsbuchstabe(n) legt String-Variablen fest

wobei in jedem Falle "Anfangsbuchstabe(n)" die Buchstaben angibt, die einen bestimmten Variablen-Typ identifizieren, wenn er als erster Buchstabe in einem Variablennamen erscheint und kein Typkennzeichen am Ende angegeben ist. Beispiel:

```
10 DEFINT A - C, M, X - Z
```

gibt an, daß alle Programm-Variablen, die mit den Buchstaben A, B, C, M, X, Y und Z beginnen und kein Typkennzeichen am Ende haben, Ganzzahlen sind. Das Format der DEF-Anweisung wird in Kapitel 15 detaillierter beschrieben.

Es ist möglich, sich über diese Definitionen hinwegzusetzen, indem die Typkennzeichen, die in Abbildung 12-1 dargestellt sind, ans Ende der Variablennamen angehängt werden.

ZEICHEN	BEDEUTUNG
%	Integer
!	Single Precision
#	Double Precision
$	String

Tabelle 12-1 Typkennzeichen

Sofern Variablen nicht als Ganzzahlen oder Zahlen mit einfacher Genauigkeit durch DEF-Anweisungen oder Typkennzeichen definiert wurden, werden alle Berechnungen mit doppelter Genauigkeit ausgeführt.

Beispiel:

NR# ist eine andere Variable als NR!, diese wieder eine andere als NR%.

Alle drei Variablen sind allerdings dazu bestimmt, einen numerischen Wert aufzunehmen. Die - wiederum andere - Variable NR$ kann nur einen String-Inhalt, z. B. "12" (String!) bekommen.

UMWANDLUNGEN

In machen Fällen ist es nötig, eine Konstante als Wert in eine Variable zu bringen oder einer Variablen einen anderen Wert zu geben. Dies kann durch die Anweisung a = b erreicht werden. Einige Beispiele sind unten gegeben.

Die Umwandlung von einfacher oder doppelter Genauigkeit in Ganzzahlen und die Umwandlung von doppelter Genauigkeit in einfache Genauigkeit führt zur Verstümmelung von Ausdrücken oder zu Rundungen, und Zahlen außerhalb des Bereichs der veränderten Variablen erzeugen einen Overflow-Error. Die Umwandlung von Ganzzahlen in Zahlen mit einfacher Genauigkeit erzeugen keinen Fehler (Error).

Beispiel 1

Umwandlung des konstanten Wertes 23.56 (einfache Genauigkeit) in die Integer-Variable A%. Beachten Sie die BASIC-Anweisungs-Folge:

```
A% = 23.56  ENTER
Ok
?A%  ENTER
 23
Ok
```

Beachten Sie, daß die Zahl abgeschnitten und nicht gerundet worden ist.

Beispiel 2

Umwandlung des konstanten Wertes 5.ØØ69999999 (doppelte Genauigkeit) in die Variable A! in einfacher Genauigkeit. Beachten Sie die BASIC-Anweisungs-Folge:

```
A! = 5.ØØ69999999  ENTER
Ok
?A!  ENTER
 5.ØØ7
Ok
```

In diesen Beispielen sind die Zeilen, die **ENTER** enthalten, Anweisungen, die vom Benutzer eingegeben werden müssen; die anderen Zeilen sind Ausgaben vom BASIC-Interpreter.

Beachten Sie, daß das Gleichheitszeichen in den genannten Anweisungen folgende Vorschrift gibt: Bringe das, was rechts von = steht, nach dem, was links steht. Links muß immer eine Variable stehen! Rechts kann eine andere Variable, eine Konstante oder ein Ausdruck stehen. Es dürfen links und rechts vom =-Zeichen keine unterschiedlichen Datenarten (also String und numerisch gemischt) auftreten!

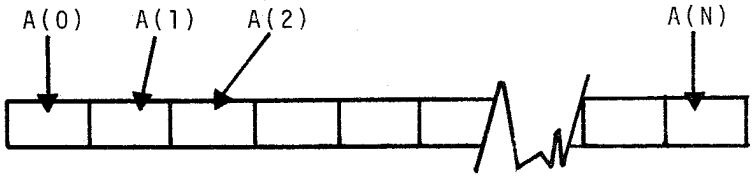
FELDER

Ein Feld (im englischen Array genannt) ist eine Zusammenfassung von Variablen desselben Typs mit einem gemeinsamen Namen. Ein Feld besteht aus mehreren Elementen, die mit dem Feldnamen und über einen oder mehrere Indices angesprochen werden. Z. B.:

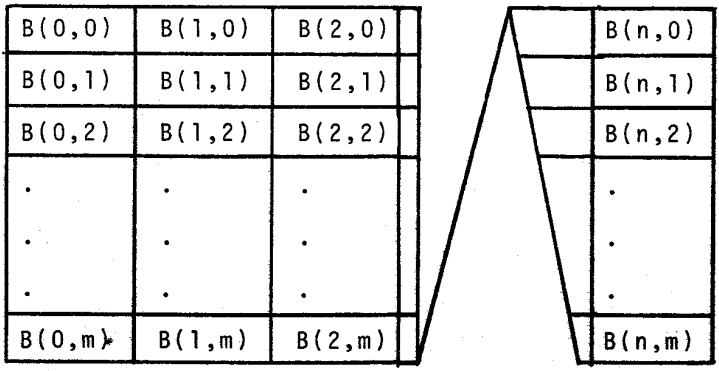
A(0), A(1), A(2)

sind die drei Elemente eines eindimensionalen Arrays namens A() -"eindimensional", weil nur ein index in Klammer verwendet wird, um das gewünschte Element zu bestimmten. Der höchstmögliche Index ist in diesem Beispiel 2.

Ein Array kann eine beliebige Anzahl von Dimensionen haben, vorausgesetzt, es ist ausreichend Speicherkapazität vorhanden. Ebenso kann eine Dimension eine beliebige Anzahl von Elementen enthalten, vorausgesetzt, der Speicherplatz ist vorhanden.



Der entsprechende Speicherraum für ein Array wird von BASIC festgelegt, wenn ein Array zum ersten Mal definiert wird. Die folgende Zeichnung stellt die einzelnen Elemente eines zweidimensionalen Arrays B() mit den maximalen Indexzahlen n und m dar.



Dieser zweidimensionale Array hat n + 1-Zeilen und m + 1-Spalten. Ein eindimensionaler Array ist geeignet für die Aufnahme eines Listeninhalts, und ein zweidimensionaler Array ist typisch für Tabellenverarbeitung.

Das Definieren von Arrays

Ein Array muß definiert werden, indem die DIM-Anweisung unter Angabe des Namens und der maximalen Indices des Arrays gegeben wird. Wenn der Array Strings aufnehmen soll, muß dies durch Benutzung des $-Zeichens am Namen angegeben werden, z. B.:

```
10 DIM A$(5,7), B(12)  ENTER
```

erstellt einen zweidimensionalen String-Array mit Indizes von Ø,Ø bis 5,7 und einen eindimensionalen numerischen Array mit Indizes zwischen Ø und 12.

Wird in einem Programm ein Array-Element angesprochen, ohne daß vorher eine DIM-Anweisung ausgeführt wurde, so wird der Array automatisch angelegt. Jeder Index des Arrays wird dann auf maximal 1Ø festgesetzt.

Die DIM-Anweisung setzt alle Element des jeweils angegebenen numerischen Arrays auf Null und alle Elemente des jeweils angegebenen String-Arrays auf den Leer-String. Es gibt keinen speziellen Befehl für die Löschung eines Arrays.

BASIC-ARITHMETIK

BASIC-Operationen werden in drei Kategorien eingeteilt, Vergleichsoperationen, numerische Operationen und logische Operationen. Für jede Kategorie gibt es eine Anzahl von Funktionen, und jede Funktion wird durch ein bestimmtes Symbol identifiziert.

VERGLEICHSOPERATOREN

Die einfachste Operation ist der Vergleich zwischen zwei Ausdrücken. Es können nur Daten gleicher Art miteinander verglichen werden. Z. B. ist es nicht möglich, einen Vergleich zu erstellen zwischen Strings und Zahlen. Die folgenden Operatoren sind sowohl für Strings als auch Zahlen benutzbar. Strings müssen die gleiche Länge haben, oder der längere String wird gekürzt, um zum kürzeren String zu passen. Der "Wert" eines String wird durch den ASCII-Wert jedes Zeichens festgelegt, wobei das erste Zeichen das signifikanteste ist.

Die Tabelle in Abbildung 12-2 stellt die Vergleichsoperatoren und ihre Bedeutung dar.

OPERATOR	BEDEUTUNG
=	gleich
>	größer als
<	kleiner als
=> oder >=	größer gleich
=< oder <=	kleiner gleich
<> oder ><	ungleich

Abbildung 12-2 Vergleichsoperatoren

Das Ergebnis eines Vergleichsausdrucks wird mit -1 angegeben, wenn der Vergleich wahr ist, und als Ø, wenn er falsch ist.

ARITHMETISCHE OPERATOREN

Arithmetische Operatoren sind nur für Zahlen möglich. Das + Symbol bedeutet eine Addition. Es kann auch benutzt werden, um Strings zusammenzufügen. Alle anderen arithmetischen Operatoren können jedoch nicht bei Strings verwendet werden. Die Tabelle in Abbildung 12-3 beschreibt die verfügbaren Operatoren und ihre Bedeutung.

OPERATOR	BEDEUTUNG
+	Addition
-	Subtraktion oder Negativ
*	Multiplikation
/	Division
\	Integer Division
^	Potenzierung

Abbildung 12-3 Arithmetische Operatoren

Anmerkung zu Abbildung 12-3:

\ (Backslash) wird im deutschen Zeichensatz durch den Großbuchstaben Ö dargestellt.

Die Zahlen, die bei einem Operator benutzt werden, werden Operanden genannt.

Bei der Integer-Division werden die Zahlen auf die nächste Ganzzahl gerundet, und das Ergebnis der Division wird auf den Ganzzahl-Teil abgeschnitten. Der Rest einer solchen Division kann durch Benutzung der MOD-Funktion abgefragt werden.

Die Reihenfolge der arithmetischen Operatoren ist:

1. Potenzierung

2. Negation

3. Multiplikation und Division

4. Integer-Division

5. Addition und Subtraktion

Bei Operationen mit der gleichen Priorität werden die Operationen in der Reihenfolge der Eingabe von links nach rechts durchgeführt. Prioritäten können durch Klammern, die die Operation, die zuerst ausgeführt werden soll, einschließen, geändert werden. Dieses Vorgehen kann durch Klammern innerhalb von Klammern erweitert werden, wie in den folgenden Beispielen dargestellt.

1. $\frac{x + y + z}{2}$ ist eingegeben als `(X + Y + Z)/2`

2. $2x + 5$ ist eingegeben als `2 * X + 5`

3. $2(x + 5)$ ist eingegeben als `2 * (X + 5)`

4. $(x + y)^2$ ist eingegeben als `(X + Y) ^ 2`

5. $\frac{x^2(x + y)^4}{z}$ ist eingegeben als `X ^ 2 * ((X + Y) / Z) ^ 4`

Klammern können auch benutzt werden, wenn sie nicht unbedingt erforderlich sind. Dies kann helfen, einen Ausdruck optisch zu verdeutlichen.

Enthält ein Ausdruck eine Variable, die bis dahin noch keinen Wert erhalten hat, wird die Variable auf Null gesetzt, wenn Sie numerisch ist, und auf Leerstring, wenn sie String ist.

Ergebnisse von Berechnungen

Der Typ des Ergebnisses eines Ausdrucks ist abhängig von den benutzten Operandentypen. Ein Ausdruck, der mehrere Operanden enthält, kann in eine Reihe von Operationen, die zwei Operanden benutzen, aufgeteilt werden. Z. B. im Ausdruck

```
2 * (X + 5)      X + 5 wird zuerst berechnet und
                 liefert Y
                 danach wird 2 * Y berechnet.
```

Das Ergebnis jeder einzelnen Operation benutzt die höhere Genauigkeitsstufe, wenn die zwei Operanden verschieden sind. Wenn z. B. ein Operand doppelte Genauigkeit hat und der andere eine Ganzzahl ist, erscheint das Ergebnis in doppelter Genauigkeit. Numerische Konstanten können grundsätzlich auch in "wissenschaftlicher Schreibweise" dargestellt werden, z. B.:

5.26E4 bedeutet $5.26 * 10^4$
-3-257E-3 bedeutet $-3.257 * 10^{-3}$; dies ist gleich $-3.257*(1/10^3)$; dies ist gleich -Ø.ØØ3257.

Ist der Wert des Ergebnisses größer bzw. kleiner als das mögliche Maximum bzw. Minimum für diesen Datentyp, wird eine overflow-Mitteilung angezeigt. Der Versuch einer Division durch Ø liefert das gleiche Ergebnis mit der Fehlermeldung Division durch Null (/Ø). Diese Fehlermeldung erscheint auch für eine Potenzierung mit einem negativen Exponenten. Ist der Exponent Null, ist das Ergebnis 1.

Ist der Wert des Ergebnisses näher an Null, als durch den Datentyp noch ausgedrückt werden kann (sehr negativer Exponent!), wird Null angenommen.

LOGISCHE OPERATOREN

Logische Operationen können in BASIC ebenfalls verwendet werden. Die folgenden Boolschen-Funktion sind verfügbar. Sie sind nach ihrer Priorität aufgeführt:

- NOT (Negation)
- AND (UND)

- OR (inclusives ODER)

- XOR (exclusives ODER)

- IMP (Impliziert Y)

- EQV (Äquivalenz)

Die Wahrheitstafeln für diese Funktionen werden in Abbildung 12-4 dargestellt.

	A 0 B 0	A 0 B 1	A 1 B 0	A 1 B 1
NOT A	1	1	0	0
A AND B	0	0	0	1
A OR B	0	1	1	1
A XOR B	0	1	1	0
A EQV B	1	0	0	1
A IMP B	1	1	0	1

Abbildung 12-4 Logische Operatoren

In diesen Wahrheitstafeln steht 1 für "wahr" oder "erfüllt" und Ø für "falsch" oder "nicht erfüllt"

Logische Operatoren können benutzt werden, um Vergleichsbedingungen zu verbinden. Sie können aber auch mit zwei rein numerischen Operanden verwendet werden. in diesem Fall werden die duale Darstellungen der Operanden Bit für Bit mit der verwendeten Booleschen Operation verbunden. Befindet sich dabei ein Operand außerhalb des Bereichs von -32768 und +32767, wird eine Fehlermeldung angezeigt. Die Operation wird Bit für Bit ausgeführt, und das Ergebnis wird als Dezimalwert im gleichen Bereich ausgegeben. Negative binäre Zahlen werden in der Form des Zweierkomplements dargestellt. Die folgenden Beispiele verdeutlichen, wie logische Operatoren benutzt werden können. Es ist nur möglich, zwei aufeinanderfolgende logische Operatoren zu benutzen, wenn der zweite Operator NOT ist.

Beispiel 1

Geben Sie in BASIC ein:

```
? 4 OR 2  ENTER
6
Ok
```

Die erste Zeile ist die Anweisung, die zweite das Ergebnis, das vom M10 angezeigt wird, und die dritte die Meldung des BASIC-Interpreters.

4 ist 100 in dualer Darstellung
2 ist 010 in dualer Darstellung
6 ist 110 in dualer Darstellung

Dies ist das Ergebnis der ODER-Operation für jedes Bit-Paar.

Beispiel 2

Geben Sie ein:

```
? 7 AND 1Ø  ENTER
2
Ok
```

Die erste Zeile ist wieder die Anweisung, die zweite Zeile das vom Computer angezeigte Ergebnis und die dritte Zeile die Meldung des BASIC-Interpreters.

7 ist 0111 in dualer Darstellung
1Ø ist 1010 in dualer Darstellung
2 ist 0010 in dualer Darstellung

Dies ist das Ergebnis der UND-Operation Bit für Bit.

Logische Ausdrücke können unter anderem benutzt werden, um eine Bedingung in IF... GOTO.. ELSE- oder IF... THEN... ELSE-Anweisungen zu testen. Sie können aber auch als numerische Ausdrücke direkt in Anweisungen verwendet werden.

WERTZUWEISUNGEN

Daten können Werte erhalten, indem die Anweisungen LET, DATA, und READ benutzt werden. Es ist auch möglich, Daten einzugeben, während das Programm ausgeführt wird. Dies wird erreicht, indem die Anweisungen INPUT oder LINEINPUT für über Tastatur einzugebende Daten oder aus einer Datei zu lesende Daten benutzt wird. Das Befehlsformat ist in beiden Fällen unterschiedlich (siehe Anweisungen INPUT und LINEINPUT in Kapitel 15).

13. PROGRAMMIERSTRUKTUREN

VERZWEIGUNGEN

Innerhalb eines BASIC-Programms unterbricht die Anweisung GOTO n das Abarbeiten eines Programms in der durch die Zeilennummern vorgegebene Reihenfolge und bewirkt einen Sprung auf die Programmzeile mit der Nummer n.

Für eine bedingte Verzweigung stehen folgende Anweisungen zur Verfügung:

```
IF... THEN... ELSE
IF... GOTO... ELSE
ON... GOTO
```

Diese Anweisungen veranlassen eine Veränderung des Befehlsfolgeablaufs aufgrund von Wahrheitswerten einzelner Variablen. Einzelheiten des Formats dieser Anweisungen werden in Kapitel 15 gegeben.

SCHLEIFEN

Programmschleifen können im BASIC mit der FOR... NEXT-Anweisung programmiert werden. Schleifen können mehrere Anweisungen beinhalten, die wiederholt ausgeführt werden sollen. Eine NEXT-Anweisung kann mehr als eine Schleife schließen, wenn die innere Schleife am gleichen Punkt endet wie die äußere Schleife. Wird eine FOR-Anweisung ohne eine dazu gehörende NEXT-Anweisung erteilt, wird kein Fehler angezeigt. Wird jedoch eine NEXT-Anweisung ohne eine vorherige FOR-Anweisung erteilt, wird die Fehlermeldung NF angezeigt.

UNTERPROGRAMME

Unterprogramme benutzt man, wenn gleiche BASIC-Anweisungen mehrmals vom Programm ausgeführt werden müssen, gegebenenfalls mit verschiedenen Variablenwerten. Sie ersparen dem Programmierer Tiparbeit, verkleinern die Programmgröße und dienen der besseren Übersicht des Programms. Die Anweisung GOSUB n veranlaßt das Programm, in Zeile n zu springen und mit der Abarbeitung der Unterroutine, die in Zeile n beginnt, fortzufahren.

Das Ende eines Unterprogramms wird durch die Anweisung RETURN angegeben. Diese fordert den Rücksprung auf die nächste Anweisung nach dem Unterprogrammaufruf mit GOSUB an.

UNTERPROGRAMME IN MASCHINENSPRACHE

Einige Unterprogramme in Maschinensprache können auch von einem BASIC-Programm aufgerufen werden. Dazu wird die CALL-Anweisung eingesetzt. Einige Unterprogramme können nicht durch die CALL-Anweisung aufgerufen werden (siehe CALL im Verzeichnis der BASIC-Befehle, Kapitel 15).

14. DATEIEN

Dateien können entweder Programme oder Daten enthalten. BASIC-Programm-Dateien werden automatisch mit dem Anhang .BA versehen, TEXT-Dateien und BASIC-Daten-Dateien haben den Anhang .DO. Programmdateien in Maschinensprache haben den Anhang .CO. Alle Daten-Dateien sind sequentiell organisiert. Normalerweise werden sie im Speicher erstellt; sie können aber auf einen Kassettenrekorder oder ein anderes peripheres Gerät übertragen werden. Es ist zwar möglich, eine Datei direkt auf Kassette zu schreiben, aber es ist nicht möglich, Daten an eine Kassetten-Datei anzuhängen. Alle Informationen, die auf eine Kassetten-Datei geschrieben werden, überschreiben das Original. Um von einer Daten-Datei zu lesen oder auf eine Daten-Datei zu schreiben, muß diese zunächst durch die OPEN-Anweisung geöffnet werden.

ÖFFNEN EINER DATEI

Die OPEN-Anweisung wird benutzt, um die Richtung der Datenübertragung zu bestimmten. Um aus einer Datei zu lesen, muß diese im Input-Modus geöffnet werden, um auf eine Datei zu schreiben, muß diese im Output-Modus geöffnet werden. Eine Datei kann zu einem bestimmten Zeitpunkt nur in einem einzigen Modus geöffnet sein, d. h., es kann entweder nur gelesen oder nur geschrieben werden.

Daten-Dateien im Speicher können im Append-Modus geöffnet werden. Danach wir alles, was in die Datei geschrieben wird, an die bestehende Datei angehängt, ohne daß der bisherige Inhalt - wie im Output-Modus - verlorengeht. Eine Datei, die sich im Speicher befindet, kann mit dem SAVE-Befehl auf den Kassettenrekorder übertragen werden, und eine auf Kassette befindliche Datei kann mit dem Befehl LOAD in den Speicher geladen werden.

Als Teil der OPEN-Anweisung wird der Daten-Datei eine Nummer zugewiesen. Die zugewiesene Dateinummer kann nicht an eine andere Datei vergeben werden, bevor die unter der Dateinummer geöffnete Datei wieder geschlossen wurde.

LESEN EINER DATEN-DATEI

Daten aus einer Datei können in einem Programm benutzt werden, indem zunächst die Datei mit der OPEN-Anweisung im Input-Mode geöffnet wird und anschließend die INPUT-Anweisung benutzt wird, um Daten einzulesen und sie Programm-Variablen zuzuweisen.

Ist die Anzahl der Einträge in eine Datei unbekannt, kann die EOF-Funktion (End Of File = Ende der Datendatei) dazu benutzt werden, das Ende der Datei zu ermitteln. (Erläuterndes Beispiel siehe EOF-Funktion) Versucht man, mehr Daten aus einer Datei zu lesen als vorhanden, so wird die EF-Fehlermeldung angezeigt.

SCHREIBEN EINER DATEN-DATEI

Daten können in eine Datei geschrieben werden, indem zunächst die Datei im Output-Mode eröffnet wird (womit der bisherige Inhalt verlorengeht!) und anschließend die Anweisungen PRINT oder PRINT USING dazu benutzt werden, Daten in die Datei zu schreiben. Eventuell können die Daten auch an den bisherigen Inhalt angehängt werden, indem die Datei im Append- statt im Output-Modus geöffnet wird (geht nur für Dateien im RAM).

SCHLIESSEN EINER DATEI

Einige Anweisungen und Befehle schließen automatisch alle offenen Dateien, z. B.: END. Um eine Datei unabhängig von diesen Anweisungen zu schließen, steht die CLOSE-Anweisung zur Verfügung.

15. VERZEICHNIS DER BASIC-BEFEHLE, -FUNKTIONEN UND -ANWEISUNGEN

ABS

Funktion

ABS liefert den Absolutwert (Betrag) eines numerischen Werts.

Format: ABS(x)

wobei: x ein numerischer Ausdruck ist.

Beispiel:

```
Ok
PRINT ABS(6*(-3))
 18
Ok
```

ASC

Funktion

ASC liefert den ASCII-Code dezimal des ersten Zeichens eines Strings.

Format: ASC(x$)

wobei: x$ ein String ist.

Ist x$ der Leerstring, wird ein FC-Fehler (illegal function call = verbotener Funktionsaufruf) angezeigt. Siehe auch die Funktion CHR$, um aus einem ASCII-Code den zugehörigen String zu entwickeln.

Beispiel:

```
Ok
10 X$ = "M10"
20 PRINT ASC(X$)
RUN
 77
Ok
```

ATN

Funktion

ATN liefert den Arcustangens einer Zahl.

Format: ATN(x)

wobei: x eine Zahl oder ein numerischer Ausdruck irgendeines numerischen Typs ist.

Das Ergebnis wird als Bogenmaß ausgegeben. Es liegt zwischen $-\pi/2$ und $+\pi/2$.

BEEP

Anweisung

Der Lautsprecher erzeugt für etwa eine halbe Sekunde einen Ton.

Format: BEEP

Beispiel:

```
IF X = 0 THEN BEEP
```

Diese Anweisung kann auch innerhalb einer FOR/NEXT-Schleife benutzt werden, um den Ton zu verlängern oder um das Ende eines langen Programms in einer GOTO-(Endlos-) Schleife anzuzeigen.

Beispiel:

```
 990  BEEP
1000  GOTO 990
```

Siehe auch SOUND-Anweisung.

CALL

Anweisung

CALL ruft ein Unterprogramm in Maschinensprache auf.

Format: CALL Adresse A ,HL

wobei: Adresse ist die Startadresse des Unterprogramms in Maschinensprache im ROM.

A ist eine Ganzzahl, die in das Register A (=ACCU) übergeben wird.

HL ist eine Ganzzahl, die in das Register HL übergeben wird.

Die Startadresse muß im Bereich zwischen -32758 und 65535 liegen. Die Ganzzahlen für die Register müssen im Bereich zwischen 0 und 255 liegen.

CDBL

Funktion

CDBL wandelt eine Zahl in eine Zahl mit doppelter Genauigkeit um.

Format: CDBL(x)

wobei: x eine Ganzzahl ist oder eine Zahl mit einfacher Genauigkeit.

Kapitel 12 liefert Einzelheiten, wie Umwandlungen durchgeführt werden. Siehe auch CINT und CSNG für die Umwandlung in Ganzzahlen oder Zahlen mit einfacher Genauigkeit.

CHR$

Funktion

CHR$ liefert das ASCII-Zeichen, das der genannten Ganzzahl im Bereich zwischen 0 und 255 (dezimal) in der ASCII-Code-Tabelle zugeordnet ist.

Format: CHR$(i)

wobei: i eine Ganzzahl im Bereich zwischen 0 und 255 ist.

Beispiel:

```
Ok
PRINT CHR$(9Ø)
Z
Ok
```

CINT

Funktion

CINT wandelt eine Zahl in eine Ganzzahl um, indem alle Nachkommastellen entfernt werden.

Format: CINT(x)

wobei: x eine Zahl oder ein Ausdruck mit einfacher oder doppelter Genauigkeit ist.

Die Zahl wird abgeschnitten, und Nachkommastellen gehen verloren. Befindet sich das Ergebnis nicht im Integer-Bereich zwischen -32768 und 32767, wird eine OV-Fehlermeldung (overflow) angezeigt. Kapitel 12 liefert Einzelheiten, wie Umwandlungen durchgeführt werden. Siehe auch CDBL und CSNG für die Umwandlung in Zahlen mit einfacher und doppelter Genauigkeit.

CLEAR

Befehl

CLEAR löscht den Speicherbereich, der für Daten benutzt worden ist.

Format: CLEAR Stringbereich ,Speichermaximum

wobei: "Stringbereich" die Anzahl der Bytes angibt, die für String-Variablen zur Verfügung stehen.

"Speichermaximum" die größte Speicherausdehnung ist, die für BASIC zur Verfügung steht. Wenn MAXRAM angegeben ist, steht der gesamte Speicher zur Verfügung.

Der Standardwert für "Stringbereich" beträgt 256 Bytes. CLEAR setzt alle numerischen Variablen auf Ø und alle String-Variablen auf Leerstring. Ebenso werden alle offenen Dateien geschlossen.

BEACHTEN SIE, daß "Stringbereich" vom System nur benutzt wird, wenn kein anderer Raum für Strings vorhanden ist. In einem Programm wird für String-Variablen automatisch Platz verwendet, wenn ausreichend vorhanden ist; und der Speicherraum, der von CLEAR erstellt wurde, wird dann nicht benötigt.

Mit CLEAR kann der Wert für die Funktion HIMEM definiert werden ("Speichermaximum").

CLOAD, CLOAD?, CLOADM

Befehle

CLOAD und CLOADM werden unter LOAD beschrieben. CLOAD? wird benutzt, um ein vorher auf Kassette gespeichertes Programm mit dem BASIC-Programm im Speicher zu vergleichen.

Format: CLOAD? "Filename"

wobei: "Filename" der Name ist, unter dem das Programm auf Kassette gespeichert war.

Dieser Befehl kontrolliert, ob ein Programm richtig auf Kassette gespeichert wurde. CLOAD? kann nur für Programme benutzt werden, die im Binärformat gespeichert wurden.

Das Programm auf dem Band wird Byte für Byte mit dem Programm im RAM verglichen. Bestehen Unterschiede, war die Datenübertragung auf das Band nicht zufriedenstellend, und die Mitteilung **Verify failed** wird angezeigt. In diesem Fall muß das Programm noch einmal auf Kassette abgespeichert werden.

CLOSE

Anweisung

CLOSE schließt den Zugang zu einem peripheren Gerät oder einer Datei ab.

Format: CLOSE File-Nr. 1, File-Nr. 2 ...

wobei: "File-Nr. k" (Dateinummer) die Nummer ist, unter der die Datei geöffnet wurde (siehe OPEN-Anweisung). Eine beliebige Anzahl von Dateien kann zur gleichen Zeit geschlossen werden.

Alle Verbindungen zu peripheren Geräten des Systems werden durch die CLOSE-Anweisung geschlossen. Alle Dateien werden automatisch geschlossen, wenn folgende Befehle oder Anweisungen gegeben wurden:

- wenn eine END-Anweisung oder ein NEW-Befehl ausgeführt wurde.
- wenn eine RUN-Anweisung oder ein LOAD-Befehl ausgeführt wurde.
- wenn das Programm gewechselt wird.
- wenn CLOSE ohne einen Operanden angegeben wird.

CLS

Befehl

CLS löscht das Display.

Format: CLS

Das komplette LCD-Display wird gelöscht, mit Ausnahme der Bezeichnungen der Funktions-Tasten, falls diese in der untersten Zeile angezeigt sind. In diesem Fall werden nur die oberen sieben Zeilen gelöscht. Zum Löschen der achten Zeile vgl. SCREEN-Anweisung.

COM ON, COM OFF, COM STOP

Anweisungen

COM ON/OFF/STOP aktiviert und inaktiviert den Datenaustausch über die RS-232C (V24)-Schnittstelle.

Format:
```
COM ON
COM OFF
COM STOP
```

COM ON ermöglicht Zugang zu der Schnittstelle. Wenn die Anweisung ON COM GOSUB benutzt wird, um auf ein Unterprogramm zu verweisen, kontrolliert BASIC bei jeder Programmzeile den RS-232C-Puffer. Wurden über die Verbindungsleitung irgendwelche neuen Zeichen empfangen, verzweigt das Programm in das Unterprogramm.

COM OFF inaktiviert ein vorher durch COM ON aktiviertes Unterprogramm.

COM STOP unterbricht die Verzweigung in ein durch COM ON aktiviertes Unterprogramm so lange, bis das Programm

auf die Anweisung COM ON trifft. Werden Zeichen empfangen, während COM STOP aktiviert ist, wird das Unterprogramm sofort aufgerufen, wenn ein späteres COM ON abgearbeitet wird.

Siehe auch ON ... GOSUB für weitere Informationen.

CONT

Befehl

CONT wird benutzt, um das Programm fortzusetzen, nachdem es durch die **BREAK** -Taste oder die STOP-Anweisung unterbrochen wurde.

Format: CONT

Dieser Befehl wird hauptsächlich bei der Fehlersuche verwendet. Wird das Programm während einer Pause in der Ausführung modifiziert, d. h. eine Programmzeile editiert oder neu eingegeben, führt der Versuch fortzufahren dazu, daß ein CN- (cannot continue-) Fehler angezeigt wird.

COS

Funktion

COS liefert den Cosinus eines Winkels.

Format: COS(x)

wobei: x der Winkel im Bogenmaß ist.

Wenn x in Grad bekannt ist, multiplizieren Sie x mit π/180 (Ø.Ø1745329), um zum Bogenmaß zu gelangen.

Beispiel:

```
PRINT COS(6Ø*.Ø1745329)
 .5ØØØØØ13Ø94ØØ4
Ok
```

CSAVE, CSAVEM

Befehle

Siehe SAVE-Befehl.

CSNG

Funktion

CSNG wandelt eine Zahl in eine Zahl mit einfacher Genauigkeit um.

Format: CSNG(x)

wobei: x eine Ganzzahl oder eine Zahl mit doppelter Genauigkeit ist.

Kapitel 12 liefert Einzelheiten über Konvertierungen. Siehe auch CDBL und CINT für die Konvertierung in doppelte Genauigkeit oder in Ganzzahlen.

CSRLIN

Funktion

CSRLIN liefert die vertikale (Zeilen-) Position des Cursors.

Format: CSRLIN

Diese Funktion kann sowohl für LCD (Display) oder VDU (Bildschirm) aufgerufen werden, wenn LCD oder VDU verfügbar sind. Bei LCD liegt das Ergebnis im Bereich zwischen Ø und 7, bei VDU liegt das Ergebnis im Bereich zwischen Ø und 24. Ø steht immer für die oberste Zeile.

DATA

Anweisung

DATA speichert die numerischen und String-Konstanten, auf die später durch die READ-Anweisung zugegriffen werden soll.

Format: DATA Konstante-1, Konstante-2 ...

wobei: Konstante k eine numerische oder String-Konstante darstellt.

Eine beliebige Zahl von Konstanten, bis zur maximalen Länge der Programmzeile kann angegeben werden. String-Konstanten brauchen nicht in Anführungszeichen einge-

schlossen werden, es sei denn, sie enthalten Kommata, Doppelpunkte oder Leerzeichen.

Die Konstanten in den DATA-Anweisungen werden vor dem eigentlichen Programmbeginn am Ende des Programms hintereinander gespeichert. Die Reihenfolge der Speicherung ist einzig abhängig von den Zeilennummern der DATA-Anweisungen (aufsteigende Reihenfolge) und nicht von der Reihenfolge, in der das Programm später, z. B. aufgrund von Sprunganweisungen, abgearbeitet wird. Bei der Abarbeitung werden die DATA-Anweisungen übergangen, da sie ja bereits vor Programmbeginn verwertet wurden.

Siehe auch die Anweisungen READ und RESTORE.

Beispiel:

```
Ok
10 READ A, B$, C, D$
20 PRINT A, B$, C, D$
30 DATA 56, AB, -3.55E + 20,6abc7
RUN 10
 56                    AB
-3.55E+20              6abc7
Ok
```

DATE$

Besondere Variable

In der Variablen DATE$ wird das laufende Datum verwaltet.

Format: DATE$ = "dd/mm/yy" (Wertzuweisung)
PRINT DATA$ (Ausgabe)

wobei: dd dem Tag entspricht und im Bereich zwischen 01 und 31 liegen muß;

mm dem Monat entspricht und zwischen 01 und 12 liegen muß;

yy dem Jahr entspricht und zwischen 01 und 99 liegen muß.

Beim Kaltstart des Systems wird DATE$ im Menü auf "01/01/00 gesetzt.

DAY$

Besondere Variable

In der Variablen DAY$ wird der Tag verwaltet.

Format:
```
DAY$ = "xxx"    (Wertzuweisung)
PRINT DAY$      (Ausgabe)
```

wobei: "xxx" folgende Werte haben darf: "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun".

Bei einem Kaltstart des Systems setzt das System "Sun" (Sonntag) ins Menü.

DIM

Anweisung

DIM wird benutzt, um die Anzahl der Dimensionen und den maximalen Index in jeder Dimension eines Arrays anzugeben.

Format:
```
DIM x(dim1, dim2 ..., dimk), y(dim1, dim2 ..., dim1), z(dim1, dim2 ..., dimn)
```

wobei: x, y und z numerische oder String-Variablen-Namen sind.

dim1, dim2 usw. sind Ganzzahlen, die den maximalen Index in der jeweiligen Dimension angeben.

Beachten Sie, daß der Elementzähler bei 0 startet, so daß ein Array A(1) ein eindimensionaler Array ist mit zwei Elementen usw. Arrays können eine beliebige Anzahl von Dimensionen haben, und jede Dimension kann eine beliebige Anzahl von Elementen haben, vorausgesetzt, es ist ausreichend Speicherplatz verfügbar. Wird ein Verweis auf einen Array gemacht, ohne eine DIM-Anweisung zu benutzen, wird der maximale Index in jeder Dimension mit 10 angenommen.

Die DIM-Anweisung setzt alle Elemente eines angegebenen Arrays auf einen Anfangswert von Null bzw. Leerstring.

Beispiel:
```
10 DIM A(5), B(2,3), C$(6,7)
```

Hier wird ein eindimensionaler Array mit 6 numerischen Elementen, ein zweidimensionaler mit 3 x 4 numerischen Elementen und ein zweidimensionaler String-Array mit 7 x 8 Elementen erstellt.

EDIT

Befehl

EDIT bringt den M10 aus dem BASIC-Modus in den TEXT-Modus, so daß das laufende Programm geändert werden kann. Um zum BASIC-Modus zurückzukehren, drücken Sie **F8** .

Format: EDIT n -m

wobei: n die erste Zeilennummer, die geändert werden kann, darstellt

m für die letzte Zeilennummer, die geändert werden kann, steht.

Wenn keine Zeilennummern angegeben sind, steht das gesamte Programm für eine Änderung zur Verfügung. Bei EDIT stehen alle Möglichkeiten des TEXT-Programms zur Verfügung mit der Ausnahme der Zeilenschaltung. Beachten Sie, daß hier das Zeichen ▶ (Zeilenabschluß) nicht zweimal hintereinander auftreten darf. Ist dies dennoch der Fall, löschen Sie es mit der **DEL** -Taste. TEXT wird in Kapitel 5 in diesem Handbuch beschrieben.

END

Anweisung

END beendet die Programmausführung, schließt alle offenen Dateien und bringt den M10 in den Direkt-Modus zurück.

Format: END

Durch diese Anweisung wird keine BREAK-Mitteilung angezeigt. Diese Anweisung kann überall in ein Programm eingebaut werden und beendet den Programmablauf in einer beliebigen Programmzeile. Es ist freigestellt, eine END-Anweisung als letzte Anweisung in ein Programm einzubauen.

Beispiel:

```
10 INPUT A, B
20 GOSUB 100
.
.
90 END
100 C = SQR (A*A + B*B)
110 IF A  B GOTO 120 ELSE GOTO 140
120 D = SQR (A*A - B*B)
130 RETURN
140 D = SQR (B*B - A*A)
150 RETURN
```

EOF

Funktion

EOF kontrolliert das Ende einer Datei.

Format: EOF(n)

wobei: die Dateinummer ist, die in der OPEN-Anweisung vergeben wurde.

Die EOF-Funktion liefert -1, wenn das Ende der angegebenen Datei erreicht ist, und Ø, falls das nicht der Fall ist. Sie sollte benutzt werden, um am Dateiende zu vermeiden, daß das System EF anzeigt, den end of file-(Ende der Datei-) Fehler. Die Funktion kann für eine beliebige sequentielle Datei oder für Daten, die über die RS-232C-Schnittstelle gesendet werden, verwendet werden.

Beispiel:

```
10 OPEN"RAM:DATA" FOR INPUT AS 1
20 IF EOF(1) GOTO 100
30 INPUT  1 A$,B
 .
 .
90 GOTO 20
100 CLOSE 1
```

Funktionen

ERL und ERR liefern die Zeilennummer und die Fehlernummer eines Fehlers innerhalb einer Fehlerbehandlungsroutine, die mit ON ERROR GOTO ... aktiviert wurde.

Format: IF ERL = Zeilennummer THEN ...
IF ERR = Fehlernummer THEN ...

wobei: "Zeilennummer" die Zeile angibt, in der der letzte Fehler auftauchte.

"Fehlernummer" die Nummer des letzten Fehlers liefert; siehe Anhang B, dort befindet sich eine komplette Liste der Fehler mit ihren Codes und Nummern.

Diese Funktionen werden normalerweise in IF ... THEN-Anweisungen benutzt, um den Programmablauf innerhalb einer Fehlerbehandlungsroutine festzulegen. Bei einem Fehler in einer Anweisungszeile ohne Zeilennummer enthält ERL die Zahl 65535:. Um dies zu testen, benutzen Sie

```
IF 65535 = ERL THEN ...
```

Nach Abarbeitung einer RESUME-Anweisung und sofort nach Aktivierung des BASIC-Interpreters hat ERR den Wert Ø. Der Wert von ERL ändert sich erst beim Auftreten eines neuen Fehlers.

Weitere Informationen über Fehlerbehandlung finden Sie unter ERROR und RESUME.

Anweisung

Die ERROR-Anweisung wird benutzt, um eine Fehlerbedingung zu simulieren oder einen Fehler zu erzeugen.

Format: ERROR n

wobei: n eine Ganzzahl zwischen Ø und 255 ist.

Wenn n die Zahl eines definierten Fehlers ist, wird eine Fehlermeldung angezeigt.

Beispiel:

```
ERROR 5
?FC ERROR
Ok
```

Fehlerbehandlungsroutinen wurden mit der Anweisung ON ERROR GOTO n aktiviert, wobei n die erste Zeile einer solchen Routine ist. Um die Anweisung ON ERROR GOTO n aufzuheben, benutzen Sie ON ERROR GOTO Ø. Alle auftretenden Fehler werden jetzt wieder angezeigt, und der Programmablauf wird gestopt.

Um eigene neue Fehlermeldungen zu definieren, wählen Sie Fehlernummern, die nicht vom BASIC-Interpreter benutzt werden (siehe Anhang B). Wurde keine Fehlerroutine mit ON ERROR GOTO ... angegeben, so meldet der BASIC-Interpreter "unprintable ERROR" (UE), also einen "nicht beschreibbaren Fehler". Zum Weiterarbeiten aus einer Fehlerbehandlungsroutine heraus vergleichen Sie die RESUME-Anweisung.

Beispiel:

```
4Ø ON ERROR GOTO 1ØØØ
 .
 .
9Ø INPUT A
1ØØ IF A < 0 THEN ERROR 25Ø
 .
 .
1ØØØ IF ERR = 25Ø THEN PRINT "Muß
    positiv sein": RESUME 9Ø
 .
```

EXP

Funktion

EXP liefert eine Potenz der Euleschen Zahl e (e = 2.71882817).

Format: EXP(x)

wobei: x eine Zahl im Bereich von -78.3365 bis 87.3365 sein muß. Wird diese Bedingung nicht erfüllt, wird eine OV- (overflow-) Meldung gegeben.

Beispiel:

```
Ok
?EXP(1)
 2.7182817
Ok
```

FILES

Befehl

FILES zeigt die Programm- und Daten-Dateien an, die sich im Augenblick im RAM befinden.

Format: FILES

Jeder Dateiname besitzt ein Typkennzeichen, das die Datei wie folgt klassifiziert:

- .BA ist eine BASIC-Programm-Datei
- .CO ist eine Maschinensprachen-Programm-Datei
- .DO ist eine Daten-Datei, die unter Benutzung des TEXT-Programms oder des BASIC-Interpreters erstellt und im ASCII-Format gespeichert wurde.

Die Dateien im RAM werden auch angezeigt, wenn man auf die Funktionstaste **F1** drückt.

FIX

Funktion

FIX schneidet von einer Kommazahl die Nachkommastellen ab.

Format: FIX(x)

wobei: x die Zahl darstellt, die ganzzahlig gemacht werden soll.

Beispiel:

```
Ok
?FIX(3.9)
 3
Ok
?FIX(-3.9)
-3
Ok
```

Beachten Sie, daß INT(-3.9) oder INT(-3.2) statt -3 -4 liefert.

FOR...NEXT

Anweisungen

FOR- und NEXT-Anweisungen ermöglichen, eine Anzahl von Anweisungen innerhalb einer Schleife so oft wie vorgegeben durchzuführen.

Format:

```
For a = n TO p  STEP s
  .
  .
NEXT  z,y  ...,  a
```

wobei: a, y, z Variablen sind

n, p, s numerische Ausdrücke sind.

Die Variable a ist ein Zähler, der als Anfangswert n erhält und der für jeden Schleifendurchgang um s erhöht wird. Die Programmzeilen zwischen der FOR- und der NEXT-Anweisung werden so oft ausgeführt, bis a = p ist. Das Programm fährt danach mit der Zeile fort, die der NEXT-Anweisung folgt. Das bedeutet, daß die Schleife (p-u)/s-mal durchlaufen wird. Ist s nicht definiert, wird die Schrittweite 1 angenommen. Negative Schritte können ebenfalls benutzt werden, vorausgesetzt der Wert von n ist größer als der Wert von p. Wird eine sich widersprechende Anzahl von Werten eingegeben, z. B. wenn s größer als die Differenz zwischen n und p ist, wird die Schleife einmal durchlaufen, d. h., daß die Veränderung von a erst beim NEXT erfolgt ("post-checked loop"). Wird s auf Ø gesetzt, wird die Schleife fortlaufend durchlaufen, bis sie unterbrochen wird.

FOR...NEXT-Schleifen können ineinander verschachtelt sein. Die einzige Begrenzung für die Anzahl der ineinander verschachtelten Schleifen ist die verfügbare Speicherkapazität. Jede Schleife muß mit einem NEXT abgeschlossen werden, und die innerste Schleife muß zuerst geschlossen werden. Enden die Schleifen am gleichen Punkt, ist es möglich, die verschiedenen Variablen in einer NEXT-Anweisung zusammenzufassen, bei der die Variablen der innersten Schleife zuerst geschrieben werden. Wird eine NEXT-Anweisung gefunden, die keine Variablen besitzt, wird vermutet, daß diese zur letztgenannte FOR-Anweisung gehört. Wird eine NEXT-Anweisung vor einer FOR-Anweisung angesprochen, wird ein NF-(next without for-) Fehler angezeigt. Das bedeutet, daß eine Verzweigung in eine FOR NEXT-Schleife einen Fehler erzeugt, aber eine Verzweigung aus einer Schleife heraus nicht.

Beispiel:

```
1Ø INPUT J
2Ø FOR I = 1 TO J
3Ø PRINT I;
4Ø FOR K = 1 TO 3
5Ø PRINT "a";
6Ø NEXT K, I
8Ø GOTO 1Ø
RUN
? 4
1 aaa 2 aaa 3 aaa 4 aaa?
```

FRE

Funktion

FRE liefert die Anzahl der für den Benutzer verfügbaren Bytes im Speicher.

Format: FRE(x)
FRE(x$)

wobei: x ein numerisches Argument ist; normalerweise Ø.

x$ ein String-Argument ist; normalerweise der Leerstring, "".

Wenn FRE mit einem numerischen Argument benutzt wird, liefert BASIC die Anzahl der Bytes, die für ein BASIC-Programm, eine TEXT-Datei, ein Maschinenprogramm usw. zur Verfügung steht. Wenn FRE mit einem String-Argument benutzt wird, liefert es die Speicherkapazität, die für Strings momentan zur Verfügung steht. Der String-Speicherplatz wird durch die vorherige CLEAR-Anweisung festgesetzt. Der reservierte Speicherplatz wird nicht immer benötigt, da der BASIC-Interpreter meist Speicherplatz für String-Variablen von selbst reserviert. (Z. B.: DIM)

Beispiel:

```
Ok
PRINT FRE("")
200
Ok
```

In diesem Beispiel beträgt der verfügbare String-Speicherplatz 200 Bytes.

GOSUB, RETURN

Anweisungen

GOSUB- und RETURN-Anweisungen werden benutzt, um die Programmkontrolle an ein Unterprogramm zu übergeben bzw. bei dessen Beendigung die Programmkontrolle an das Haupt-Programm zurückzugeben.

Format: GOSUB n

.

RETURN

wobei: n der Zeilennummer entspricht, in der das aufgerufene Unterprogramm beginnt.

GOSUB verzweigt in das angegebene Unterprogramm, und RETURN führt die Programmkontrolle zur nächsten Anweisung hinter die aufrufende GOSUB-Anweisung zurück. Falls erforderlich, kann ein Unterprogramm mehr als ein RETURN enthalten. Dies sollte aber der Überschaubarkeit wegen nicht unbedingt in Anspruch genommen werden.

Dem Unterprogramm sollte eine STOP-, END- oder GOTO-Anweisung vorausgehen, um einen unprogrammierten Eingang in das Unterprogramm zu verhindern. Wird eine

RETURN-Anweisung ohne eine vorherige GOSUB-Anweisung aufgerufen, wird ein RG- (RETURN-without-GOSUB-) Fehler angezeigt. Eine Verzweigung mittels GOTO in ein Unterprogramm verursacht keinen Fehler, jedoch entsteht der Fehler beim Versuch des Rücksprungs aus dem Unterprogramm (RG-Fehler).

Unterprogramme können ineinander verschachtelt werden. Die Verschachtelung ist nur durch die Speicherkapazität begrenzt.

Beispiel:

```
100 GOSUB 200
110 PRINT "Zurück aus äußerem Unterpro-
    gramm"
.
190 END
200 PRINT "Lauf des äußeren Unterpro-
    gramms"
.
220 GOSUB 300
230 PRINT "Zurück aus innerem Unterpro-
    gramm"
.
250 RETURN
.
300 PRINT "Lauf des inneren Unterpro-
    gramms"
.
320 RETURN
RUN
Lauf des äußeren Unterprogramms
Lauf des inneren Unterprogramms
Zurück aus innerem Unterprogramm
Zurück aus äußerem Unterprogramm
Ok
```

Siehe auch ON ... GOSUB

GOTO

Anweisung

GOTO wird benutzt, um eine unbedingte Verzweigung (unbedingten Sprung) in eine andere Zeile außerhalb der normalen Programmreihenfolge zu erzeugen.

Format: GOTO n

wobei: n die Zeilennummer ist, zu der gesprungen wird.

Diese Anweisung kann z. B. ohne Zeilennummer benutzt werden, um das aktuelle BASIC-Programm im RAM in einer bestimmten Zeile zu starten. Normalerweise wird sie in Programmen benutzt, um die Programmkontrolle an eine andere Zeile zu übergeben.

Beispiel:

```
350 GOTO 10
```

veranlaßt das Programm, von Zeile 350 nach Zeile 10 zu springen. Als Verweis für bedingte Verzweigung siehe IF ... THEN ... ELSE und ON ... GOTO-Anweisungen.

HIMEM

Funktion

HIMEM enthält die größte Speicheradresse, die für den Benutzer verfügbar ist.

Format: ?HIMEM

Der Wert für HIMEM wird von der letzten CLEAR-Anweisung festgesetzt. Wurde dort nichts festgesetzt, wird 62960 (MAXRAM) angenommen.

Beispiel:

```
Ok
PRINT HIMEM
40960
Ok
```

IF ... GOTO/THEN ... ELSE

Anweisungen

IF ... GOTO ... ELSE- oder IF ... THEN ... ELSE-Anweisungen werden benutzt, um eine bedingte Verzweigung (bedingten Sprung) im Programm zu veranlassen.

Format:
```
IF x GOTO n ELSE yyy
IF x THEN yyy ELSE zzz
```

wobei: x ein logischer Ausdruck ist,

n eine Zeilennummer ist,

yyy und zzz Teile sind, die Anweisung(en) oder nur eine Zeilennummer enthalten können.

Wenn der Ausdruck x wahr ist, verzweigt die Programm-Kontrolle in den GOTO- oder THEN-Teil der Anweisung, arbeitet diesen ab und verzweigt anschließend zur nächsten Zeilennummer, falls während der Abarbeitung kein anderer Sprung erfolgte. Ist x nicht wahr, verzweigt die Programm-Kontrolle in den ELSE-Teil der Anweisung, arbeitet diesen ab und verzweigt anschließend zur nächsten Zeilennummer, falls während der Abarbeitung kein anderer Sprung erfolgte.

Beispiel:

```
.
11Ø IF A > B GOTO 12Ø ELSE GOTO 13Ø
12Ø D = SQR(A*A - B*B)
13Ø D = SQR(B*B - A*A)
.
```

IF ...-Anweisungen können verknüpft werden. Die Begrenzung ist die Zeilenlänge, da IF ... GOTO/THEN ... ELSE eine einzige Anweisung ist, die innerhalb einer Programmzeile vervollständigt werden muß.

Beispiel:

```
5Ø IF A > Ø THEN PRINT "POS" ELSE IF A=Ø
THEN PRINT "GLEICH" ELSE IF A < Ø THEN
PRINT "NEG"
```

Wenn nicht die gleiche Anzahl von THEN- und ELSE-Teilen in einer Anweisung vorhanden sind, wird jedes ELSE mit dem nächsten unverbundenen THEN verbunden.

Beispiel:

```
7Ø IF A = B THEN IF B = C THEN PRINT
"A = C" ELSE PRINT "?"
```

Druckt kein "?" aus, wenn A≠B, aber druckt ein "?" aus, wenn die Bedingungen A=B und B≠C erfüllt sind. "A=C" wird nur ausgedruckt, wenn A=B und B=C sind.

INKEY$

Funktion

Die Funktion INKEY$ liefert einen String der Länge 1 (1 Zeichen), wenn eine Taste gedrückt ist, oder einen Leer-String, wenn keine Taste gedrückt ist.

Format: INKEY$

Die Funktion INKEY$ wird meistens in einer Schleife verwendet, die solange durchlaufen wird, bis eine Taste gedrückt ist. Sie akzeptiert jede Tastenkombination mit der Ausnahme **CTRL** + **c** , die den Programmablauf unterbricht.

Beispiel:

```
.
110 PRINT "Drücken Sie eine beliebige
    Taste."
120 A$ = INKEY$
130 IF A$ = "" GOTO 120 ELSE GOTO 140
140 REM Weiterverarbeitung
 .
 .
 .
 .
 .
```

INP

Funktion

INP liefert den ASCII-Code des Bytes an einem bestimmten Ausgang für externe Geräte.

Format: INP(i)

wobei: i eine Ganzzahl im Bereich zwischen Ø und 255 ist, die die Nummer des Ausgangs ist.

INP ist eine Umkehrfunktion zu OUT.

Anweisung

INPUT wird benutzt, um während eines Programmlaufs Daten über die Tastatur einzugeben, die das Programm benötigt, um weiterzuarbeiten.

Format: INPUT "Erklärung"; a, b ..., n

wobei: die in " eingeschlossene Erklärung angezeigt wird, um dem Benutzer den erforderlichen Eingabewert und Datentyp zu erklären.

a, b, n sind numerische oder String-Variablen.

Die eingegebenen Daten müssen mit dem einzugebenden Variablentyp übereinstimmen, sonst wird die Mitteilung **Redo from start** angezeigt. Dies geschieht auch, wenn Kommata eingegeben werden, die nicht zur Trenung mehrerer Werte dienen können. In diesen Fällen muß die gesamte Eingabe korrekt nochmals eingetippt werden.

Wenn eine INPUT-Anweisung mehr als einen Wert verlangt, muß jeder einzugebende Wert vom nächsten durch ein Komma getrennt werden. Wenn weiniger Variablen eingegeben wurden als vom INPUT-Befehl verlangt, zeigt die M10 "??" an, um die restlichen Eingaben entgegenzunehmen.

Beispiel:

```
Ok
10 INPUT "Geben Sie den Tag und das Da-
   tum ein"; A$,B
20 PRINT "Heute ist "A$;B
RUN
Geben Sie den Tag und das Datum ein? (der
Benutzer tippt Sat,1   ENTER  )
Heute ist Sat 1
Ok
```

INPUT#

Anweisung

INPUT# wird benutzt, um Daten von einer Datei in ein Programm einzulesen.

Format: INPUT# File-Nummer, a, b ..., n

wobei: "File-Nummer" die Nummer ist, unter der die Datei geöffnet wurde. Siehe OPEN.

a, b, n sind numerische oder String-Variablen

Die einzelnen Werte in der Datei müssen mit der Liste der Variablen der INPUT-Anweisung übereinstimmen. Daten können aus dem RAM, von Kassette oder über die Kommunikations-Schnittstelle eingelesen werden.

Bei numerischen oder String-Werten werden führende Leerzeichen, Zeilenumschaltungen (ASCII 13) und Zeilenvorschub (ASCII 10) ignoriert. Das erste Zeichen, das kein Leerzeichen, keine Zeilenumschaltung oder Zeilenvorschub ist, wird als Anfang einer Zahl oder eines Strings betrachtet. Eine Zahl wird durch eine Leertaste, eine Zeilenschaltung oder einen Zeilenvorschub oder Komma abgeschlossen.

IPL

Befehl

IPL wird benutzt, um ein Programm direkt nach dem Einschalten zu starten (nur Warmstart).

Format: IPL "File-Name"

wobei: "File-Name" der gesamte Name der Programm-Datei ist.

Um ein vorher gesetztes IPL aufzuheben, benutzen Sie den gleichen Befehl ohne Argument.

Beispiel:

```
IPL "BASIC"
```

Dies veranlaßt den M10, sofort nach dem Einschalten BASIC aufzurufen, ohne das Menü anzuzeigen.

Anweisung

KEY wird benutzt, um einer bestimmten Funktionstaste (F1 - F8) einen String zuzuweisen. Dieser kann eine Zeichenkette sein, die bei der Erstellung eines Programms immer wieder benötigt wird, oder auch ein im Direkt-Modus sofort auszuführender Befehl sein. KEY LIST zeigt die aktuelle Funktionstastenbelegungen. KEY ON/OFF/STOP kontrolliert den Übergang in ein Unterprogramm falls eine bestimmte Funktionstaste gedrückt wird.

Format:
```
KEY n, xxx
KEY LIST
KEY (n) ON
        OFF
        STOP
```

wobei: n die Nummer einer Funktionstaste ist, die zwischen 1 und 8 liegen muß.

xxx ein String-Ausdruck mit bis zu 15 Zeichen ist.

Die Funktionstasten F1 bis F8 haben die Anfangswerte, die im Beispiel unten angegeben sind. Der Benutzer kann andere Funktionen zuweisen, indem er die Funktion KEY n, xxx benutzt.

Beispiel:
```
KEY LIST
Files               Load "
Save "              Run
List
                    Menu

Ok
```

Die Auflistung erscheint in folgender Form:

F1	F2
F3	F4
F5	F6
F7	F8

KEY(n) ON aktiviert den Zugang zu einem Unterprogramm, wenn die Taste n gedrückt wird. Das Unterprogramm muß in einer ON KEY GOSUB-Anweisung definiert sein (siehe ON ... GOSUB).

KEY(n) OFF inaktiviert das Unterprogramm, das mit ON KEY GOSUB aktiviert wurde.

KEY(n) STOP sperrt den Zugang zu dem Unterprogramm, das mit ON KEY GOSUB aktiviert wurde, bis ein KEY(n) ON abgearbeitet wird. Wurde die Funktionstaste n gedrückt, während KEY(n) STOP aktiv war, wird das Unterprogramm, das vorher mit ON KEY GOSUB aktiviert war, aufgerufen, sobald KEY(n) ON ausgeführt worden ist.

KILL

Befehl

KILL wird benutzt, um eine Datei aus dem RAM zu löschen.

Format: KILL "xxxxx.yy"

wobei: xxxxx der Dateiname ist

yy den Typ der Datei bedeutet, z. B.:

- .BA für eine BASIC-Programmdatei
- .CO für eine Programmdatei in Maschinensprache
- .DO für eine TEXT-Datei

Es muß die ganze Dateispezifikation angegeben werden. In BASIC kann das aktuelle Programm im Arbeitsspeicher nicht gelöscht werden.

LCOPY

Anweisung

LCOPY druckt den augenblicklichen Textinhalt des Displays dem Drucker aus.

Format: LCOPY

Die **PRINT** -Funktionstaste druckt ebenfalls den augenblicklichen Textinhalt des Displays aus.

LEFT$

Funktion

LEFT$ liefert einen Teilstring, der mit dem linken Zeichen eines vorgegebenen Strings beginnt.

Format: LEFT$(x$,i)

wobei: x$ der String ist, aus dem ein Teilstring von links gebildet werden soll

i eine Ganzzahl zwischen Ø und 255 ist.

Die ersten i Zeichen des String x$ werden ermittelt. Ist i größer als x$ an Zeichen enthält, wird der ganze String das Ergebnis der Funktion, ist i gleich Null, wird der Leerstring Ergebnis der Funktion.

LEN

Funktion

LEN liefert die Länge eines Strings.

Format: LEN(x$)

wobei: x$ ein String ist.

Das Ergebnis der Funktion ist die Anzahl der Zeichen in x$, einschließlich Leerzeichen und nicht-druckbarer Zeichen.

LET ... =

Anweisung

LET ... = weist einer Variablen einen Wert zu.

Format: LET x = a

wobei: x eine Variable ist

a ein Ausdruck von demselben Typ wie x ist.

Ein String-Ausdruck kann nicht einer numerischen Variablen zugewiesen werden und umgekehrt. Die Benutzung des Sprachelements LET ist freigestellt: dasselbe Ergebnis wird mit x = a erreicht.

LINE

Anweisung

LINE zeichnet eine Linie auf das Display oder kann zur Darstellung eines Rechtecks benutzt werden.

Format: LINE (x1,y1) - (x2,y2) ,c

LINE (x1,y1) - (x2,y2) ,c,B F

wobei: x1,y1 die Koordinaten des Startpunktes und x2,y2 die des Endpunktes der Linie sind. x muß im Bereich zwischen Ø (linker Rand des Displays) und 239 (rechter Rand des Displays) liegen. y muß im Bereich zwischen Ø (oberste Punktzeile) und 63 (unterste Punktzeile) liegen.

c ist der Farb-Parameter; wenn c = 1 ist, wird eine Linie oder ein Rechteck gezeichnet, wenn c = Ø ist, wird eine Linie oder ein Rechteck gelöscht.

B zeichnet ein Rechteck, dessen Seiten parallel zur obersten Punktzeile bzw. den Rändern des Displays sind, mit der angegebenen Linie als Diagonale

F füllt das Rechteck in Abhängigkeit vom Farb-Parameter c aus.

Koordinaten, die nicht innerhalb des Bereichs liegen, liefern einen FC- (illegal function call-) Fehler.

Werden die Koordinaten des ersten Punktes nicht angegeben, wird der zuletzt durch LINE, PRESET oder PSET angesprochene Punkt als Anfangskoordinaten (x1,y1) angenommen. Wurden vorher keine Bildpunkte angesprochen, so werden die Anfangskoordinaten x1=Ø und y1=Ø angenommen.

Beachten Sie, daß c für die Rechteck-Option zwingend ist, aber für eine Linie ausgelassen werden kann; in diesem Fall wird 1 angenommen.

LINE INPUT

Anweisung

LINE INPUT wird benutzt, um über Tastatur bis zu 254 beliebige Zeichen in eine String-Variable einzugeben, ohne Anführungszeichen als Begrenzung zu benutzen.

Format: LINE INPUT "Erklärung"; a$

wobei: die "Erklärung" angezeigt wird, bevor der String eingegeben wird.

Kein ? angezeigt wird, wenn keine "Erklärung" angegeben wurde.

a$ eine String-Variable ist, die auch Kommata, führende Leerzeichen, Leerzeichen am Ende oder Kommata enthalten darf.

Alle über Tastatur eingegebenen Zeichen, die als Antwort auf "Erklärung" gelten, werden der String-Variablen, bis **ENTER** gedrückt ist, zugewiesen. Wenn ein Zeilenvorschub mit Zeilenschaltung benutzt wird, ist dieser ein Teil der String-Variablen, wobei die Zeilenschaltung ignoriert und die Zeicheneingabe fortgesetzt wird.

Um einen LINE INPUT abzubrechen, drücken Sie **CTRL** + c oder **SHIFT** + **BREAK** . BASIC kehrt dann zum Direct-Modus zurück, und **Ok** wird angezeigt. Geben Sie danach CONT ein, und das Programm wird ab der LINE INPUT-Anweisung fortgesetzt.

LINE INPUT#

Anweisung

LINE INPUT# wird benutzt werden, um eine ganze Zeile von 254 Zeichen aus einer sequentiellen Datei in eine String-Variable zu lesen.

Format: LINE INPUT# File-Nummer, a$

wobei: "File-Nummer" (Dateinummer) die Nummer ist, unter der die Datei mit der OPEN-Anweisung geöffnet wurde.

a$ eine String-Variable ist.

Diese Anweisung liest alle Zeichen der Datei bis zum Zeilenwechsel. Die Folge Zeilenwechsel/Zeilenvorschub wird ignoriert, und die nächste LINE INPUT#-Anweisung liest die Zeichen bis zum nächsten Zeilenwechsel.

Befinden sich ein Zeilenvorschub und eine Zeilenschaltung in der Zeichenfolge, werden sie als Teil des Zeichen-Strings interpretiert.

Diese Anweisung kann für Dateien im RAM oder auf der Kassette benutzt werden, ebenso für sequentielle Daten aus einem Puffer, die über die Verbindungs-Schnittstelle empfangen worden sind.

LIST

Befehl

LIST zeigt Teile des aktuellen Programms oder das gesamte aktuelle Programm an.

Format: LIST n - m

wobei: n die erste Zeilennummer ist, die aufgelistet werden soll

m die letzte Zeilennumer ist, die aufgelistet werden soll.

Werden keine Zeilennummern angegeben, wird das ganze Programm aufgelistet. Fehlt der Wert für n, wird das ganze Programm bis zur Zeile m aufgelistet. Fehlt der Wert m, wird das gesamte Programm ab Zeile n bis zum Ende aufgelistet. Werden sowohl n als auch m angegeben, wird das Programm von Zeile n bis Zeile m aufgelistet.

Um die Auflistung zu beenden, drücken Sie **CTRL** + c oder **SHIFT** + **BREAK** .

Um die Auflistung zu unterbrechen, drücken Sie **CTRL** + s oder **PAUSE** ; um die Auflistung fortzusetzen, drücken Sie auf **CTRL** + q oder **PAUSE** .

Befehl

LLIST gibt das gesamte aktuelle Programm oder Teile daraus auf dem Drucker aus. Der Befehl hat das gleiche Format wie LIST.

LOAD, LOADM, CLOAD , CLOADM

Befehle

LOAD, LOADM, CLOAD und CLOADM werden benutzt, um ein Programm als aktuelles Programm in den Arbeitsspeicher zu laden. Ein solches Programm ist jedoch nicht immer im Verzeichnis (FILES-Befehl) registriert. Um ein Programm im Verzeichnis zu registrieren, muß es mit dem entsprechenden SAVE-Befehl gespeichert werden.

Format:

```
CLOAD  "File-Name"  ,R
LOAD "CAS:File-Name "  ,R
LOAD " RAM:File-Name"  ,R
LOAD "COM:cbpsx"  ,R
CLOADM  "File-Name"
LOADM "CAS:File-Name"
LOADM " RAM:File-Name"
```

wobei:

CLOAD und CLOADM Programme von Kassette laden.

"File-Name" (Dateiname) der Name ist, unter dem das Programm gespeichert wurde.

R das Programm automatisch startet, nachdem es geladen worden ist

CAS die Programm-Datei von Kassette lädt

RAM die Programm-Datei aus dem RAM lädt

COM das Programm von der Kommunikations-Schnittstelle lädt. Die Parameter sind in Band 1 beschrieben.

CLOAD und LOAD laden BASIC-Programme in den Arbeitsspeicher. Das bisherige aktuelle Programm ist nicht mehr aktuell. CLOADM und LOADM laden Programme in Maschinensprache, die sich an der bei der Speicherung angegebenen Stelle befinden.

Normalerweise schließt ein LOAD-Befehl alle Dateien. Die R-Option startet das Programm, nachdem es geladen worden ist, und hält Daten-Dateien offen. Wird LOAD zusammen mit der R-Option benutzt, ist es möglich, ein Programm an das andere anzuhängen. Siehe auch RUN-Befehl.

Die Befehle CLOAD und LOAD "CAS:..." haben die gleiche Funktion. Sie laden beide die angegebenen Programm-Dateien von der Kassette. Fehlt der "File-Name", wird die nächste BASIC-Programm-Datei auf der Kassette geladen. Um das Laden von der Kassette zu stoppen, drücken Sie die Tasten **SHIFT** + **BREAK** . Die Befehle CLOAD und CLOADM "CAS:..." haben die gleiche Funktion. Normalerweise ist es möglich, die Programmdaten von der Kassette zu hören; wenn jedoch der SOUND OFF-Befehl vor dem Befehl, ein Programm zu laden, erteilt wurde, wird dies unterdrückt.

LOAD "RAM:..." ist ähnlich dem Befehl LOAD "CAS:...", jedoch wird aus dem RAM geladen. Dabei kann RAM: entfallen.

Programme und Daten können auch über eine installierte Leitung für Datenfernverarbeitung geladen werden. Für den Empfang dieser Daten ist es jedoch erforderlich, daß die entsprechenden Parameter für die Datenübertragung richtig gesetzt werden. Einzelheiten dieser Übertragungsparameter werden in Band 1, Kapitel 8 Abbildung 8-5 gegeben.

LOG

Funktion

LOG liefert den natürlichen Logarithmus einer eingegebenen Zahl.

Format: LOG(x)

wobei: x ein numerischer Ausdruck ist, der größer als Ø sein muß.

Beispiel:

```
PRINT LOG(1Ø)
 2.3Ø2585Ø92994
Ok
```

Funktion

LPOS liefert die Position des nächsten noch nicht ausgegebenen Zeichens innerhalb des Druck-Puffers.

Format: LPOS(x)

wobei: x ein numerisches Argument ist, das normalerweise Ø ist.

Diese Funktion liefert nicht unbedingt die aktuelle Position des Druckkopfes.

LPRINT, LPRINT USING

Anweisungen

LPRINT und LPRINT USING geben Daten auf dem Drucker aus. Es besteht das gleiche Format wie für die PRINT- und die PRINT USING-Anweisungen.

Wird anstelle eines Druckers ein Microplotter eingesetzt, muß zur Veränderung des Plotter-Modus mit Hilfe der LPRINT-Anweisung eine bestimmte Zeichenfolge (definiert durch CHR$) an den Microplotter ausgegeben werden. Einzelheiten dieser und anderer Anweisungen finden Sie unter den Befehlen für den MICROPLOTTER.

Verschiedene Druckerfunktionen müssen auch bei Text-Modus durch Senden von bestimmten Zeichenfolgen (mit Hilfe der Funktion CHR$) über LPRINT veranlaßt werden.

Beispiel:

```
LPRINT CHR$(7); erzeugt ein akustisches
Signal am Drucker (nur PR2300)

LPRINT CHR$(27) + CHR$(81) + "Ø72" +
CHR$(27) + CHRS(9Ø); legt fest, daß ein
Formularblatt 72 Zeilen haben soll.
```

Vergleichen Sie zu den jeweiligen Möglichkeiten und Steuerzeichen die entsprechenden Druckerhandbücher.

MAXFILES

Besondere Variable

MAXFILES enthält die maximale Anzahl der Dateien, die geöffnet werden können.

Format: MAXFILES = n
?MAXFILES

wobei: n eine Ganzzahl zwischen Ø und 15 ist.

Wenn MAXFILES = n nicht benutzt wird, um die maximale Anzahl der Dateien festzulegen, nimmt das System einen Wert von Ø an.

?MAXFILES liefert die Zahl, die durch MAXFILES = n festgelegt wurde.

Beispiel:

```
MAXFILES = 10
Ok
PRINT MAXFILES
 10
Ok
```

MAXRAM

Funktion

MAXRAM liefert die höchste für BASIC verfügbare Speicheradresse.

Format: MAXRAM

Diese Funktion wird normalerweise benutzt, um damit den möglichen Höchstwert des zweiten Arguments innerhalb einer CLEAR-Anweisung anzugeben.

MENU

Befehl

MENU verläßt BASIC und kehrt zum Menü zurück.

Format: MENU

Wenn sich das System in BASIC befindet, ruft die Funktionstaste F8 diesen Befehl auf.

MERGE

Befehl

MERGE konstruiert die Zeilen eines Programms, das in ASCII-Format gespeichert ist (oder von der Kommunikations-Schnittstelle kommt) mit dem aktuellen Programm im Speicher.

Format:
```
MERGE "CAS:File-Name "
MERGE " RAM:File-Name"
MERGE "COM:cbpsx"
```

wobei: "File-Name" (Dateiname) der Name ist, unter dem die Datei gespeichert ist, und zwar im ASCII-Format (SAVE mit abschließender Angabe von ,A; s. SAVE-Befehl).

CAS konstruiert das Programm von der Kassette ein.

RAM konstruiert ein Programm aus dem RAM ein.

COM konstruiert ein Programm über die Schnittstelle für Datenübertragung ein.

Zeilen aus dem externen oder aus dem RAM-Programm werden mit den Programmzeilen, die sich als aktuelles Programm im Arbeitsspeicher befinden, gemischt. Treten Zeilennummern doppelt auf, werden die Zeilennummern, die sich im aktuellen Programm befinden, durch die entsprechenden Zeilen aus dem einzubindenden Programm ersetzt.

Im Arbeitsspeicher befindet sich als aktuelles Programm das bisherige aktuelle Programm, kombiniert mit dem einzubindenden Programm. Der MERGE-Befehl dient vor allem dazu, bereits vorhandene Programm-Modi (meist in Form von in sich abgeschlossenen Unterprogrammen) in ein vorhandenes Programmgerüst einzubinden. MERGE kann während des Laufens eines Programms nicht eingesetzt werden, da der BASIC-Interpreter nach MERGE sofort in den Direkt-Modus übergeht, und das gerade abzuarbeitende Programm somit abbricht.

Die BASIC-Anweisung für das Drucken auf dem Microplotter ist LPRINT. Es gibt verschiedene ASCII-Codes, die als Anweisungen zur Verfügung gestellt werden. Diese sind unten im einzelnen beschrieben. Um sie zu senden, muß die BASIC-Funktion CHR$ benutzt werden

Beispiel: LPRINT CHR(Ø8)

führt dazu, daß der ASCII-Code für einen Zeichenrückschritt (ein Zeichen nach links = Ø8) an den Microplotter gesendet wird. Wenn die gleiche Form für einen ASCII-Code, der ein druckbares Zeichen darstellt, benutzt wird, wird dieses Zeichen gedruckt.

Beispiel: LPRINT CHR(9Ø)

In diesem Falle wird der Buchstabe Z (Code 90) gedruckt.

Der Microplotter hat zwei Modi, den Text-Modus (der automatisch beim Einschalten eingestellt ist) und den Grafik-Modus. Durch das Senden von Code 18 kann vom Text- in den Grafik-Modus gewechselt werden. Ein Wechsel vom Grafik- zum Text-Modus erfolgt durch Senden des Codes 17. Im Grafik-Modus sind einige spezielle Befehle verfügbar, die unten detailliert beschrieben sind. Diese Befehle müssen als in Anführungszeichen eingeschlossener String gesendet werden.

Beispiel: LPRINT "A"

Dies bringt den Microplotter zum Text-Modus zurück.

TEXT-MODUS Befehle

CHR$(Ø8) Ein Zeichen zurück: führt den "Schreiber" ein Zeichen nach links. Dies wird vor allem für das Unterstreichen benötigt.

CHR$(11) Eine Zeile nach oben: Dies bewegt das Papier eine Zeile zurück. Mit dieser Funktion kann man überschreiben.

CHR$(18) Wählt den Grafik-Modus.

CHR$(29) Drehen des "Schreibers": Dies bewegt den "Schreiber" um eine Position auf die nächste Farbe.

GRAFIK-MODUS-BEFEHLE

Sind in den folgenden Befehlen x- bzw. y-Koordinaten angegeben, bezeichnet die x-Achse die Punktzeile und die y-Achse die Punktspalte. Beachten Sie, daß die Zeichenfläche auf dem Papier in 480 Einheiten in jeweils beide Richtungen aufgeteilt ist. Der Microplotter akzeptiert alle Werte für x und y, jedoch werden Werte größer als 480 für x als 480 behandelt.

CHR$(17) Wählt den Text-Modus.

A Rückkehr zum Text-Modus: Dieser Befehl verursacht eine Rückkehr zum Text-Modus und bewegt den "Schreiber" zum linken Rand.

Cn (color n) Farbe wechseln: Die Farbe kann entsprechend n gewechselt werden, wobei Ø (schwarz), 1 (rot), 2 (grün) oder 3 (blau) sein kann. Die angegebenen Farben arbeiten nur, wenn die "Schreiber" sich in der angegebenen Position befinden.

Dx1,y1,x2,y2,x3,y3 ...
Draw (Zeichnen): Dieser Befehl zeichnet eine Linie ab der augenblicklichen "Schreiber"- Position zu den Koordinaten, die durch x1,y1 angegeben worden sind, mit Bezug auf den Ausgangspunkt (sofern nichts anderes durch I angegeben, befindet sich der Ausgangpunkt am linken Papierrand). Danach wird eine Linie von x1,y1 nach x2,y2 gezeichnet usw. Die Werte von x und y können zwischen Ø und 999 liegen. Das folgende Beispiel zeichnet ein kleines Quadrat, vorausgesetzt, daß sich der "Schreiber" zu Beginn in Position Ø,Ø befindet.

DØ,25,25,25,25,Ø,Ø,Ø

H: Home: Dies bewegt den "Schreiber" zum Ausgangspunkt zurück, ohne eine Linie zu zeichnen.

I: Initialize (Initialisieren): Dieser Befehl macht die augenblickliche "Schreiber"-Position zur Ausgangsposition.

Jx1,y1,x2,y2,x3,y3, ...
Draw relative (relatives Zeichnen): Dieser Befehl arbeitet genauso wie D, aber die Koordinaten verweisen in jedem Fall auf die augenblickliche "Schreiber"-Position, z. B.:

JØ,25,25,Ø,Ø,-25,-25,Ø zeichnet ein kleines Quadrat. Die Werte von x und y müssen zwischen -999 und 999 liegen.

Mx,y Move (Bewegen): Bewegt den "Schreiber", ohne eine Linie zu zeichnen, zu der mit den Koordinaten x,y angegebenen Position, mit Bezug auf den Ausgangspunkt.

PZeichenfolge Print (Drucken): Dies druckt die alphanumerischen Zeichen, die in "Zeichenfolge" angegeben sind.

SGröße Size (Größe): Dies verändert die Zeichengröße, die mit P gedruckt wird. Der Wert der "Größe" kann zwischen Ø (80 Zeichen pro Zeile) und 63 (1 Zeichen pro Zeile) liegen. Der Standardwert von Größe liegt bei Ø.

Qn Print direction (Druckrichtung): Dies gibt die Richtung des Drucks, abhängig von n, an. Der Wert von n kann Ø (von links nach rechts), 1 (von oben nach unten), 2 (von rechts nach links und von unten nach oben), 3 (von unten nach oben) sein. Der Standardwert für n ist Ø, und beim Einschalten startet der Microplotter mit QØ.

Rx,y Move relative (relatives Bewegen): Dies bewegt den "Schreiber", ohne eine Linie zu zeichnen, auf die Position x,y mit Bezug

auf die augenblickliche Position. Die Werte von x und y müssen im Bereich zwischen -999 und 999 liegen.

XAchse,Schrittweite,Intervall

Draw axis (Achse zeichnen): Dies zeichnet die x- oder y-Achse für einen Graphen. Die Achse wird mit "Achse" angegeben, wobei 1 = x und Ø = y ist. Der Abstand zwischen zwei Markierungen auf der Achse ist ein "Schritt". Abstände können zwischen -999 und 999 liegen. "Intervall" gibt die Anzahl der zu wiederholenden Schritte an. Diese Zahl kann zwischen 1 und 255 liegen.

MID$

Funktion

MID$ liefert einen Teilstring, der sich innerhalb eines anderen Strings befindet.

Format: MID$(x$,i ,j)

wobei: x$ ein String-Ausdruck ist.

i und j sind Ganzzahlen zwischen 1 und 255.

Ein String mit j Zeichen wird aus dem String x$ herausgenommen. Gestartet wird dabei bei dem i-ten Zeichen von x$. Wenn i weggelassen wird oder größer ist als die Anzahl der im String rechts verbleibenden Zeichen, werden alle Zeichen rechts von i als Ergebnis ermittelt. Wenn i größer ist als die Zeichenanzahl in x$, wird als Ergebnis der Leerstring geliefert.

Beispiel:

```
1Ø A$ = "Olivetti M1Ø"
2Ø B$ = MID$(A$,7,4)
3Ø PRINT B$
RUN
ti M
Ok
```

MOD

Operation

DIE MOD-Operation liefert den Rest einer Ganzzahl-Division.

Format: n MOD m

wobei: n der Dividend und m der Divisor ist (beides Ganzzahlen).

Beispiel:

```
?10 MOD 3
1.ØØØØØØØØØØØØØØ.
Ok
```

MOTOR ON, MOTOR OFF

Anweisungen

MOTOR ON und MOTOR OFF kontrollieren den Motor des Kassettenrekorders aus einem Programm heraus, oder im Direkt-Modus.

Format: MOTOR ON
MOTOR OFF

Diese Anweisungen ermöglichen oder sperren den Eingriff der Kassettenrekorder-Schalter. Sie sollten benutzt werden, um sicherzugehen, daß der Motor des Kassettenrekorders nur läuft, wenn Daten übertragen werden.

NAME

Befehl

NAME wird benutzt, um eine Datei umzubenennen.

Format: NAME "aaa.xx" AS "bbb.xx"

wobei: aaa der alte Datei-Name ist

bbb der neue Datei-Name ist

xx das Datei-Kennzeichen, das für beide Namen gleich sein muß, ist.

Wenn der alte Datei-Name nicht existiert oder der neue Datei-Name bereits benutzt wird, wird ein FC- (illegal function call-) Fehler angezeigt. Wird das Datei-Kennzeichen nicht angegeben, wird ein FF- (file not found-) Fehler angezeigt.

NEW

Befehl

NEW löscht das laufende Programm aus dem Arbeitsspeicher und löscht alle Variablen.

Format: NEW

Alle numerischen Variablen werden auf Ø gesetzt, und alle String-Variablen werden auf Leerstring gesetzt. Der String-Raum, der durch ein vorheriges CLEAR zugewiesen worden ist, wird nicht beeinflußt.

ON ... GOSUB

Anweisung

ON ... GOSUB verursacht eine Verzweigung (Sprung) in ein Unterprogramm.

Format:
```
ON COM GOSUB n
ON TIME$ = "hh:mm:ss" GOSUB n
ON KEY GOSUB a ,b,c ...
ON x GOSUB a ,b,c ...
```

wobei: a,b,c,n Programm-Zeilennummern sind

COM einen Sprung (eine Verzweigung) in Zeile n verursacht, falls sich Daten im Puffer für die Datenübertragung befinden.

TIME$ = "hh:mm:ss" gibt den Zeitpunkt an, zu dem BASIC in Zeile n verzweigen (springen) soll.

KEY verursacht eine Verzweigung (einen Sprung) in eine bestimmte Zeile, abhängig davon, welche der acht Funktionstasten gedrückt worden ist.

x ist ein numerischer Wert, dessen Ergebnis einen Sprung (eine Verzweigung) in eine bestimmte angegebene Zeile verursacht.

Die Anweisungen ON COM/GOSUB werden in Verbindung mit COM/ON/OFF/STOP benutzt. Vorausgesetzt, daß eine COM ON-Anweisung ausgeführt worden ist, verzweigt BASIC in die angegebene Zeile, falls Daten per Datenübertragung empfangen worden sind. Wenn eine COM OFF-Anweisung ausgeführt worden ist, ist der Zugang zum Unterprogramm gesperrt. COM STOP verhindert eine Verzweigung in das Unterprogramm, bis die nächste COM ON-Anweisung aufgerufen wird. Weitere Informationen finden Sie unter COM ON/OFF/STOP.

Mit TIME$="hh:mm:ss" wird der Zeitpunkt festgesetzt, wann BASIC in die angegebene Zeile springen (verzweigen) soll. Wird die Verzweigung (der Sprung) durchgeführt, erfolgt ein automatischer TIME$ STOP. Die Rückkehr (RETURN) aus der Unterroutine erzeugt ein automatisches TIME$ ON, sofern TIME$ OFF innerhalb der Unterroutine angegeben ist. Die Anweisung ON TIME$ ... GOSUB ist nur wirksam, wenn ein Programm ausgeführt wird.

Bei der Anweisung ON KEY GOSUB können maximal 8 Sprungziele hintereinander, durch Komma getrennt, angegeben werden. Bei Druck von **F1** wird dann zum ersten dieser Sprungziele, bei Druck von **F2** zum zweiten gesprungen, etc. Wird die Verzweigung (der Sprung) durchgeführt, findet ein automatischer KEY(n) STOP statt, und die Rückkehr (RETURN) aus der Unterroutine erzeugt einen automatischen KEY(n) ON, sofern KEY(n) OFF Teil der Unterroutine ist.

Die Anweisung ON x GOSUB wird benutzt, um abhängig vom x-Wert in verschiedene Zeilen zu springen (zu verzweigen). Ist x = 1, verzweigt (springt) BASIC zur ersten angegebenen Zeilennummer; ist x = 2, verzweigt (springt) BASIC zur zweiten Zeilennummer usw. Ist x keine Ganzzahl (Integer), werden die Nachkommastellen nicht beachtet. Ist x = Ø oder größer als die Anzahl der Positionen in der Liste, fährt das Programm mit der nächsten Anweisung fort (vorausgesetzt x < 255). Ist x kleiner Ø oder größer 255, wird ein FC- (illegal

function call-) Fehler angezeigt. Ist x größer als die angegebene Anzahl Sprungziele, wird ebenfalls dieser Fehler angezeigt.

Wenn ein Fehler auftritt, während eine Fehlerbehandlungsroutine durch ON ERROR GOTO ... aktiv ist, erfolgt kein Sprung zum Unterprogramm, und der Fehler wird angezeigt.

Beispiel:

```
10 INPUT "Wähle 1, 2 oder 3";A
20 ON A GOSUB 100,200,300
30 GOTO 10
100 PRINT "Unterprogramm 1": RETURN
200 PRINT "Unterprogramm 2": RETURN
300 PRINT "Unterprogramm 3": RETURN
RUN
Wähle 1, 2 oder 3? (der Benutzer tippt 2
     ENTER  )
Unterprogramm 2
Wähle 1, 2 oder 3? (der Benutzer drückt
     CTRL  +  c)
Break in 10
Ok
```

ON ... GOTO

Anweisung

ON ... GOTO verursacht eine Verzweigung (einen Sprung) in eine andere Programmzeile, und zwar bei ON ERROR GOTO ... in eine Fehlerbehandlungsroutine und bei ON x GOTO ... in diejenige Zeile, die durch den Wert von x festgelegt wird.

Format:

```
ON ERROR GOTO n
ON x GOTO a ,b,c ...
```

wobei: a,b,c,n Programm-Zeilennummern sind.

ERROR aktiviert eine Fehlerbehandlungsroutine und gibt deren erste Zeile an.

x ist ein numerischer Wert, dessen Ergebnis den Sprung (die Verzweigung) in die mit a, b, c ... bezeichnete Zeilennummer verursacht.

Nach einer ON ERROR GOTO-Anweisung verursachen alle auftretenden Fehler einschließlich der Fehler im Direct-Modus, z. B. SN (syntax error) automatisch einen Sprung in die angegebene Zeile (erste Zeile der Fehlerbehandlungsroutine). Normalerweise führt ein Fehler zu einem Abbruch des Programms, jedoch kann eine Fehlerbehandlungs-Routine benutzt werden, um die Programmkondition zu ändern, die zu der Fehlermeldung führte, und anschließend an einer bestimmten Stelle im Hauptprogramm fortzufahren. Die Fehlerbehandlungsroutine kann inaktiviert werden, indem die Anweisung ON ERROR GOTO Ø durchgeführt wird. Mit der RESUME-Anweisung wird eventuell nach Behebung der Fehlerursache in der Fehlerbehandlungsroutine die normale Programmausführung an der richtigen Stelle fortgesetzt.

ON x GOTO wird benutzt, um in Abhängigkeit von einem Wert x in verschiedene Zeilen zu verzweigen (zu springen); analog zu ON x GOSUB. Ist x gleich 1, verzweigt (springt) BASIC zur ersten angegebenen Zeilennummer; ist x gleich 2, springt BASIC zur zweiten Zeilennummer usw. Ist x keine Ganzzahl, werden die Nachkommastellen nicht beachtet. Ist x=Ø oder größer als die Zahl der in der Liste angegebenen Positionen, fährt das Programm mit der nächsten Anweisung fort (vorausgestzt x < 255). Ist x kleiner Ø oder größer als 255 oder größer als die angegebene Anzahl Sprungziele, wird ein FC- (illegal function call-) Fehler angezeigt.

OPEN

Anweisung

Die OPEN-Anweisung wird benutzt, um Dateien im RAM und auf Kassette zu öffnen oder einem peripheren Gerät einen Puffer, der als Datei benutzt wird, zuzuweisen.

Format:

```
OPEN "CAS:File-Name "  FOR Modus AS  File-
    Nummer
OPEN " RAM:File-Name"  FOR Modus AS  File-
    Nummer
OPEN "COM:cpbsx" FOR Modus AS File-Nummer
OPEN "LCD:" FOR OUTPUT AS   File-Nummer
OPEN "LPT:" FOR OUTPUT AS   File-Nummer
OPEN "WAND:" FOR INPUT AS   File-Nummer
```

wobei: CAS den Kassettenrekorder anspricht

RAM das RAM anspricht

COM die Datenübertragung an die Kommunikations-Schnittstelle eröffnet

LCD das LCD-Display auf dem M10 anspricht

LPT den "Line-printer" (Zeilendrucker) anspricht

WAND den Bar-Code-Leser anspricht

"File-Name" (Datei-Name) der Name ist, unter dem die Datei gespeichert wurde

"File-Nummer" (Datei-Nummer) die Nummer ist, die der Datei zugewiesen wird

Modus INPUT, OUTPUT oder APPEND ist

Wenn Eingabe-/Ausgabe-Anweisungen, die Anweisungen oder Funktionen mit einer Dateinummer benötigen (z. B. PRINT#, PRINT# USING, INPUT#, LINE INPUT#), durchgeführt werden sollen, muß vorher eine OPEN-Anweisung ausgeführt worden sein.

Die Datei-Nummer wird in anderen Eingabe-/Ausgabe-Anweisungen benutzt, um auf eine geöffnete Datei zu verweisen. Sie muß eine Ganzzahl zwischen 1 und der größten Zahl, die in der MAXFILES-Variablen gerade gültig ist, sein.

Die drei Modi INPUT, OUTPUT und APPEND sind nur für Dateien im RAM verfügbar. Für Kassettendateien oder Dateien der Datenübertragung sind nur INPUT (Eingabe) und OUTPUT (Ausgabe) verfügbar. APPEND heißt Ausgabe an das bisherige Ende der Datei. OUTPUT heißt Ausgabe ab Beginn der Datei (der alte Inhalt geht somit verloren).

Eine Datei kann nicht für einen bestimmten Modus eröffnet werden, wenn sie schon in einem anderen Modus offen ist, d. h. vor allem, daß wechselweises Schreiben und Lesen auf eine Datei nicht möglich ist. Es sind dazu zwei Dateien nötig, eine zum Lesen, die anderen zum (evtl. geändert) beschreiben.

Wenn die Datenübertragung als Datei eröffnet worden ist, müssen die Parameter für die Datenübertragung angegeben werden (siehe Band 1, Kapitel 8 Abbildung 8-5).

OUT

Anweisung

OUT wird benutzt, um ein Daten-Byte an einen Ausgang für externe Geräte zu senden.

Format: OUT n,i

wobei: n die Nummer des Ausgangs ist,
i der ASCII-Code des zu sendenden Daten-Bytes darstellt.

Sowohl n als auch i müssen Ganzzahlen im Bereich zwischen Ø und 255 sein. OUT ist das Gegenteil der INP-Funktion.

PEEK

Funktion

PEEK liefert den ASCII-Code des an einer bestimmten RAM-Adresse gespeicherten Bytes.

Format: PEEK(i)

wobei: i den Speicherplatz angibt, der sich im Bereich zwischen -32768 und 65535 befinden muß.

Der ausgegebene Wert ist eine Dezimalzahl im Bereich zwischen Ø und 255. Negativwerte werden von der Zahl 65536 abgezogen, z. B. -1 hat den gleichen Wert wie 65535.

PEEK wird meistens in Verbindung mit Maschinenprogrammen benutzt, um Ergebnisse eines Maschinenprogramms in BASIC weiterzuverarbeiten.

Anweisung

POKE schreibt ein Daten-Byte in eine angegebene Speicherposition.

Format: POKE k,i

wobei: k der Speicherposition entspricht und im Bereich zwischen -32768 und 65535 liegen muß.

i dem ASCII-Code des Daten-Bytes entspricht und eine Ganzzahl im Bereich zwischen Ø und 255 (dezimal) sein muß.

Wird eine negative Speicherposition angegeben, wird diese von 65536 abgezogen, z. B. - 1 ist gleich dem Wert 65535. Benutzen Sie die POKE-Anweisung mit Vorsicht, da das System nicht kontrolliert, ob die angegebene Speicherposition nicht vom BASIC-Programm belegt ist.

Mit POKE werden meistens Ausgangsparameter an ein Maschinenprogramm übergeben, oder das Maschinenprogramm selbst wird mit POKE-Anweisungen erstellt bzw. aktiviert.

Funktion

POS liefert die augenblickliche Cursorposition auf dem Display.

Format: POS(i)

wobei: i ein numerisches Argument ist, das normalerweise Ø ist.

Der linke Rand des Bildschirm ist Ø.

Beispiel:
```
PRINT POS(Ø);POS(Ø);POS(Ø)
 Ø  3  6
Ok
```

POWER i, CONT/OFF

Anweisungen

POWER steuert die Zeit vor dem automatischen Abschalten.

Format: POWER i
POWER CONT
POWER OFF ,RESUME

wobei: i ein Ganzzahlwert zwischen 1Ø (1 Minute) bis 255 (25,5 Minuten) ist

Der Strom wird automatisch ausgeschaltet, falls innerhalb der in der POWER i-Anweisung angegebenen Zeit keine Eingaben erfolgt sind. Diese Möglichkeit wird automatisch inaktiviert, wenn eine POWER CONT-Anweisung ausgeführt wird.

POWER OFF schaltet den Strom ab. Wenn das System durch Ein- und Ausschalten neu gestartet wird, wird das Menü angezeigt. Wird die Anweisung POWER OFF RESUME benutzt, meldet sich das System in demselben Zustand, der vor dem Ausschalten vorhanden war. Schaltet sich das System automatisch aus, so meldet es sich ebenfalls nach dem Einschalten im gleichen Zustand, in dem es beim automatischen Ausschalten war.

PRESET

Anweisung

PRESET wird benutzt, um einen Punkt, der auf dem Display in schwarz oder weiß dargestellt werden soll, auszugeben.

Format: PRESET(x,y ,c)

wobei: x und y die Koordinaten des Punktes sind. x muß im Bereich zwischen Ø und 239 liegen, y muß im Bereich zwischen Ø und 63 liegen.

c ist der Farb-Parameter: ist c = 1 oder fehlt c, wird das Element in weiß dargestellt, ist c = Ø, wird das Element in schwarz dargestellt.

Werden Koordinaten, die außerhalb des angegebenen Bereiches liegen, angegeben, wird ein FC- (illegal function call-) Fehler angezeigt.

Diese Anweisung ist ähnlich der PSET-Anweisung, jedoch steht hier c = 1 für ein schwarzes Element.

PRINT

Anweisung

PRINT zeigt Daten auf dem Display an.

Format: PRINT Liste der Daten
? Liste der Daten
PRINT§n, Liste der Daten

wobei: "Liste der Daten" eine Liste von Ausdrükken sein kann, die durch Kommata oder Semikola getrennt werden.

? eine Abkürzung für PRINT ist.

§n die Start-Position für den Ausdruck definiert; Einzelheiten werden unten erläutert.

Die Ausdrücke können sowohl numerisch als auch Strings sein, jedoch muß eine String-Konstante in Anführungszeichen eingeschlossen sein. Variablennamen können ebenfalls benutzt werden. Es wird deren zugewiesener Inhalt ausgedruckt.

Das Semikolon bewirkt, daß eine Information direkt hinter der anderen folgt, mit der Ausnahme, daß hinter Zahlen jeweils ein Leerzeichen steht. Vor positiven Zahlen steht ein Leerzeichen, vor negativen Zahlen steht ein Minus-Zeichen. Werden Kommata benutzt, um die Informationen zu trennen, werden diese in einer tabellierten Form dargestellt. Die Tabulations-Stops liegen 14 Zeichen auseinander; auf dem Bildschirm werden zwei Spalten geformt, wie im Beispiel unten dargestellt.

Beispiel:

```
Ok
PRINT 12;-345;"abc";" def"
 12 -345 abc def
Ok
a = 2*4 : b$ = "Jan" : c = 1983
PRINT a;b$;c
 8 Jan 1983
Ok
PRINT a,10,12,2*7
 8             10
 12            14
Ok
```

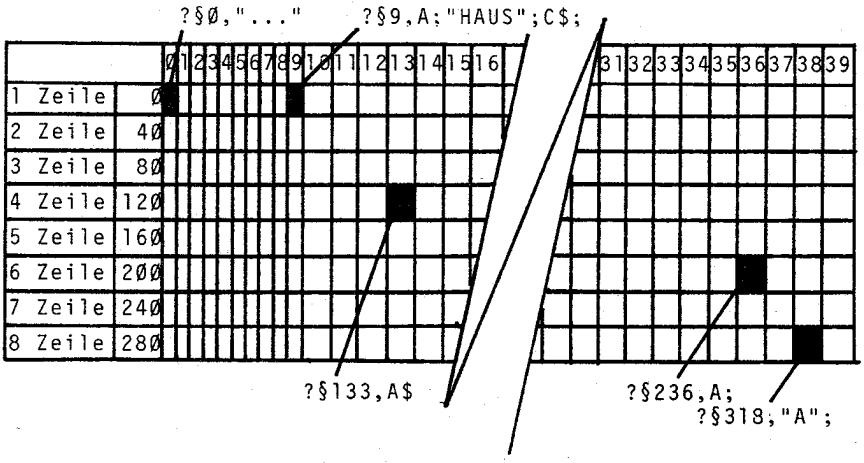
Wird ein Komma oder Semikolon an das Ende einer Liste oder einer Programmzeile gesetzt, beginnt die nächste PRINT-Anweisung den Druck an der durch das Komma bzw. Semikolon festgehaltenen Position.

Beispiel:

```
10 PRINT 1;2;3;4;
20 PRINT 1;2,3,4
RUN
 1  2  3  4  1  2
 3              4
Ok
```

Die PRINT§-Anweisung wird benutzt, um den Druck an einer bestimmten Position zu starten. Der Wert von n muß zwischen Ø und 319 liegen: Die Zahl entspricht der Position eines Zeichens. Die erste Position, dargestellt durch Ø, ist die obere linke Ecke des Bildschirms. Die Position für das zweite und alle weiteren Zeichen werden von links nach rechts auf der oberen Zeile des Bildschirms berechnet, anschließend von links nach rechts auf der zweiten Zeile usw. Der Maximalwert von n entspricht der Position der unteren rechten Ecke des Bildschirms. Für das LCD-Display ist der maximale Wert von n 319. Für einen Monitor von 40 Zeichen mal 25 Zeilen ist die maximale Größe von n 999, für einen Monitor von 80 Zeichen mal 25 Zeilen ist der maximale Wert von n 1999.

Beispiel:



PRINT#, PRINT# USING

Anweisungen

PRINT# und PRINT# USING schreiben Daten in eine sequentielle Datei.

Format: PRINT# File-Nummer, Liste der Daten
PRINT# File-Nummer, USING "x"; Liste der Daten

wobei: "File-Nummer" (Datei-Nummer) die Nummer ist, unter der die Datei geöffnet worden ist

"Liste der Daten" eine Liste der zu schreibenden Daten, die durch Komma oder Semikolon getrennt sind. Es können auch numerische oder String-Variable verwendet werden.

x ein String-Ausdruck ist, der das Ausgabe-Format (Maske) definiert.

Diese Anweisungen haben das gleiche Format wie PRINT und PRINT USING, aber die Daten werden in die durch "File-Nummer" bezeichnete Datei übertragen. Es kann auf

jede periphere Datei geschrieben werden, indem die Anweisungen PRINT# und PRINT# USING benutzt werden, jedoch muß die Datei zunächst im Output- bzw. APPEND-Modus eröffnet werden. Siehe OPEN.

PRINT USING

Anweisung

PRINT USING zeigt Daten unter Benutzung eines angegebenen Formats auf dem Bildschirm an (z. B. zum links- oder rechtsbündigen Ausgeben von numerischen Werten).

Format: PRINT USING "x"; Liste der Daten

wobei: x ein String-Ausdruck ist, der das Output- (Ausgabe-) Format definiert und in Anführungszeichen eingeschlossen sein muß.

"Liste der Daten" eine Liste der Ausdrücke ist, die durch Semikolon oder Komma getrennt ist.

Zwischen den ausgegebenen Informationen ist keine Leerstelle, es sei denn, sie ist im Formatier-String enthalten.

Folgende Format-Elemente sind verfügbar:

FORMAT-ELEMENT	BESCHREIBUNG
! für Strings	Gibt an, daß nur das erste Zeichen des jeweiligen Strings ausgegeben wird.
\ \ für Strings	Mit den Schrägstrichen nach links (backslash) wird die Anzahl Zeichen definiert, die für einen String vorgesehen sind. Befinden sich z. B. zwei Leerstellen zwischen den Schrägstrichen nach links, werden vier Zeichen gedruckt. Die Schrägstriche zählen also mit.
# für Zahlen	Das Nummernzeichen entspricht jeder einzelnen Position, die innerhalb

des Formates gefüllt wird. Besteht die Zahl aus weniger Positionen als zum Druck benötigt werden, werden auf dem Display vor der Zahl Leerzeichen angezeigt. Ein Dezimal-Punkt kann an jeder Position benutzt werden; wird nur eine Position vor dem Dezimal-Punkt angegeben, wird diese immer gedruckt, wenn nötig als Ø. Haben die Zahlen mehr Positionen hinter dem Dezimal-Punkt als für den Druck vorgesehen, werden sie gerundet. Wenn die eingegebenen Daten größer sind als die im angegebenen Format mögliche maximale Größe, wird vor dem Teil des Wertes, der ausgegeben werden kann, ein % ausgegeben.

`+` für Zahlen

Das Plus-Zeichen kann vor oder hinter die anderen Formatierungszeichen gesetzt werden. Es bewirkt, daß das Vorzeichen der Zahl vor oder hinter der Zahl gedruckt wird.

`-` für Zahlen

Ein Minus-Zeichen hinter den anderen Formatierungszeichen bewirkt den Ausdruck negativer Zahlen mit einem anschließenden Minus-Zeichen.

`**` für Zahlen

Zwei Sterne direkt vor numerischen Formatierungszeichen bewirken, daß führende Leerzeichen mit Sternen aufgefüllt werden. Die Sterne definieren ebenfalls einzelne Positionen.

`$$` für Zahlen

Zwei Dollar-Zeichen direkt vor numerischen Formatierungszeichen bewirken, daß ein Dollar-Zeichen direkt vor der Zahl ausgedruckt wird. Bei diesem Befehl ist es nicht möglich, die Exponentialschreibweise zu benutzen, und negative Zahlen können nur mit einem nachfolgenden Minus dargestellt werden.

**$
für Zahlen

Dies kombiniert die Auswirkung der zwei Sterne und der zwei Dollar-Zeichen: Führende Leerzeichen werden mit Sternen ausgefüllt, und ein Dollarzeichen wird direkt vor die Zahl gedruckt. Diese Kombination definiert zwei weitere Positionen für die Zahl.

,
für Zahlen

Ein Komma links vom Dezimal-Punkt innerhalb eines Formatier-Strings bewirkt, daß ein Komma für jeweils drei Positionen, die sich links vom Dezimal-Punkt befinden, gedruckt wird (Tausender-Separator). Das Komma gibt eine weitere einzelne Position an und hat keine Auswirkung auf Zahlen in Exponentialschreibweise. Ein Komma am Ende des Formatier-Strings wird als ein Zeichen ausgedruckt.

^^^^
für Zahlen

Vier Exponent-Symbole geben an, daß die Zahl in Exponentialschreibweise gedruckt werden soll. Die signifikanten Zeichen werden linksbündig dargestellt, und ein Leerzeichen oder ein Minus wird in der ersten Position angezeigt, vorausgesetzt, + oder - sind in den Formatier-Zeichen enthalten.

Beispiele:

```
A$ = "Olivetti" : B$ = "M1Ø"
Ok
PRINT USING "!";A$;" ";B$
O M
Ok

PRINT USING "\   \";A$;B$      (2 Leer-
      zeichen zwischen den\ )
OlivM1Ø
Ok

A=123.45:8=6.789:C=-123.45:D=-6.789
Ok
```

```
PRINT USING "##.##";A;B;C;D
%123.45  6.79 %-123.45 -6.79
Ok

PRINT USING "#.###^^^^ - ";A;B;C;D
1.235E+Ø2  6.789E+ØØ  1.235E+Ø2- 6.789E+ØØ
Ok

PRINT USING "+**$ .     ";A;B;C;D
+$123.45Ø **+$6.789 -$123.45Ø **-$6.789
Ok
```

PSET

Anweisung

PSET wird benutzt, um einen Punkt auf dem Display, der in schwarz oder weiß dargestellt werden soll, auszugeben.

Format: PSET(x,y ,c)

wobei: x,y die Koordinaten des Elements sind und x im Bereich zwischen Ø und 239 liegen muß und y im Bereich zwischen Ø und 63.

c die Farb-Parameter darstellen: Bei c = Ø wird das Element weiß dargestellt, bei c = 1 (oder c fehlt) wird das Element in schwarz dargestellt.

Wenn Koordinaten, die außerhalb des vorgegebenen Bereiches liegen, angegeben werden, wird ein FC- (illegal function call-) Fehler angezeigt.

Diese Anweisung ist der PRESET-Anweisung ähnlich, jedoch wird bei c = Ø ein schwarzes Element dargestellt.

READ

Anweisung

READ übernimmt Werte aus DATA-Anweisung(en) und weist sie Variablen zu.

Format: READ Variable-1 ,Variable-2 ... , Variable-n

wobei: Variable-k eine numerische oder String-Variable ist.

Die READ-Anweisung kann Zugang zu einer oder mehreren DATA-Anweisungen haben, die hintereinander gelesen werden. Wird versucht, per READ eine Variable mit Wert zu versehen, obwohl keine weiteren Werte im durch DATA definierten internen Datenfile vorhanden sind, wird ein OD- (out of data-) Fehler angezeigt. Die Variablen müssen mit der jeweils zu lesenden Konstanten in der DATA-Anweisung übereinstimmten (gleicher Variablentyp), sonst wird ein SN-Fehler (syntax error) angezeigt.

Sind weniger Variablen als Daten vorhanden, beginnen folgende READ-Anweisungen beim ersten ungelesenen Element. Sind mehr Datenelemente als Variablen am Ende der READ-Anweisung vorhanden, werden die restlichen Daten ignoriert.

RESTORE

Anweisung

RESTORE wird benutzt, um zum ersten Element in der ersten DATA-Anweisung zurückzukehren.

Beispiel:

```
Ok
1Ø READ A,B$,C,D$
2Ø PRINT A,B$,C,D$
3Ø DATA 56,AB,-3.55E+2Ø,6abc7
RUN
56             AB
-3.55E+2Ø    6abc7
Ok
```

REM

Anweisung

REM ermöglicht es, Anmerkungen in ein Programm einzufügen, die keine Auswirkungen auf die Programmausführung haben, aber im Programm beim Listen erscheinen.

Format: REM
'

Das Apostroph kann als Abkürzung benutzt werden. Der gesamte Text hinter REM wird als Anmerkung betrachtet. Daher muß in einer Anweisungszeile, die aus mehreren Einzelanweisungen mit : zusammengesetzt ist, die REM-Anweisung die letzte Anweisung sein.

Es ist möglich, mit Hilfe von GOTO oder GOSUB in eine REM-Zeile zu verzweigen (zu springen). Das Programm wird mit der nächsten ausführbaren Anweisung fortgesetzt.

REM darf nicht in einer DATA-Anweisung erscheinen, da die Bemerkungen sonst als Daten behandelt würden.

RESTORE

Anweisung

RESTORE wird benutzt, um die per DATA erstellten Elemente im internen Datenfile von Beginn an zu lesen oder ab der Position im internen Datenfile zu lesen, die durch eine DATA-Anweisung mit einer bestimmten Zeilennummer festgelegt wurde.

Format: RESTORE n

wobei: n eine Zeilennummer ist.

Nach einer RESTORE-Anweisung hat die nächste READ-Anweisung Zugriff auf das erste Element innerhalb einer DATA-Anweisung in einer angegebenen DATA-Zeile. Fehlt n, benutzt die READ-Anweisung die erste DATA-Anweisung im Programm. Ist die Zeilennummer nicht vorhanden, wird ein UL-Fehler (undefined line error) angezeigt.

RESUME

Anweisung

RESUME setzt die Programmdurchführung nach einer Fehlerbehandlungsroutine fest.

Format: RESUME
RESUME Ø
RESUME NEXT
RESUME n

wobei: n eine Zeilennummer ist.

RESUME und RESUME Ø bewirken die Fortsetzung der Ausführung mit der Anweisung, in der der Fehler im eigentlichen Programm vorkam. Dies setzt natürlich voraus, daß die Fehlerursache innerhalb der Fehlerbehandlungsroutine beseitigt wurde. RESUME NEXT bewirkt die Fortsetzung der Ausführung mit der Anweisung, die dem Fehler folgt. RESUME n bewirkt die Fortsetzung der Ausführung in Zeile n.

Wenn ein RESUME angetroffen wid, ohne daß vorher aufgrund eines Fehlers in eine aktivierte Fehlerbehandlungsroutine verzweigt wurde, wird ein RW- (resume without error-) Fehler angezeigt.

RIGHT$

Funktion

RIGHT$ liefert einen Teil-String aus einem vorgegebenen String, der eine bestimmte Anzahl von Zeichen am Ende des Strings enthält.

Format: RIGHT$(x$,i)

wobei: x$ ein String ist

i eine Ganzzahl zwischen Ø und 255 ist.

Die letzten i-Zeichen des Strings x$ sind das Ergebnis der Funktion. Ist i größer als die Anzahl Zeichen von x$, wird der gesamte String zum Ergebnis der Funktion, ist i gleich Null, wird der Leer-String das Ergebnis der Funktion.

Beispiel:

```
B$ = "Olivetti M1Ø"
B$ = RIGHT$(A$,5)
PRINT B$
i M1Ø
Ok
```

Funktion

RND liefert eine Pseudo-Zufallszahl zwischen Ø und 1.

Format: RND(x)

wobei: x eine Zahl ist.

Bei x > 0 wird bei jedem Programmlauf die gleiche Serie von Pseudo-Zufallszahlen erzeugt. Bei x <=0 wird die vorher erzeugte Zahl in den Serien wiederholt. Dies wird bei der Fehlersuche in Programmen eingesetzt. Die Reihenfolge der erstellten Zahlen kann mit Hilfe der TIME$-Funktion in einem Programm "verstreut" untergebracht sein. Zum Beispiel:

```
1Ø J=VAL(RIGHT$(TIME$,2))
2Ø FOR I=1 TO J
3Ø RN=RND(1):NEXT
```

liefert 60 verschiedene Werte für RN. Dies ist nur eine Möglichkeit, wie die Folge "verstreut" dargestellt werden kann.

Befehle

RUN und RUNM werden benutzt, um Programme in den Arbeitsspeicher zu laden und zu starten oder das aktuelle Programm zu starten. Programme werden nicht in das Verzeichnis aufgenommen, bevor sie nicht durch den SAVE-Befehl gespeichert worden sind.

Format:

```
RUN n  "name.xx"    ,R
RUN "CAS:File-Name "   ,R
RUN " RAM:File-Name"   ,R
RUN "COM:cbpsx"   ,R
RUNM "CAS:File-Name "
RUNM  "RAM:File-Name"
```

wobei: n eine Zeilennummer ist

name.xx der Programm-Datei-Name und sein Typkennzeichen ist

"File-Name" (Datei-Name) der Name ist, unter dem das Programm gespeichert wurde

R alle momentan offenen Daten-Dateien offenhält

CAS ein Programm von Kassette lädt und startet

RAM ein Programm aus dem RAM lädt und startet

COM ein Programm aus der Kommunikations-Schnittstelle lädt und startet.

RUN lädt ein BASIC-Programm als aktuelles Programm und startet es, wobei das vorher dort befindliche Programm gelöscht wird. RUNM lädt und startet ein Maschinensprachen-Programm, das unter der angegebenen Position gespeichert wurden.

Ein Programm im RAM kann ab einer bestimmten Zeile, die durch n angegeben ist, gestartet werden. Wird bei Eingabe des Programm-Datei-Namens das Typkennzeichen nicht mit angegeben, werden zunächst alle .BA-Dateien und anschließend alle .DO-Dateien gesucht, falls der Name nicht schon unter den .BA-Dateien gefunden wurde.

Normalerweise schließt der RUN-Befehl alle offenen Dateien. Die R-Option hält jedoch die Daten-Dateien offen. RUN hat daher die gleiche Bedeutung wie LOAD ... ,R.

Wird RUN oder RUNn ohne sonstige Zusätze gegeben, wird das aktuelle Programm gestartet.

SAVE, SAVEM

SAVE und SAVEM werden benutzt, um Programme zu speichern.

Format:

```
CSAVE"File-Name"
SAVE "CAS:File-Name"  ,A
SAVE " RAM:File-Name"  ,A
SAVE "COM:cbpsx"
CSAVEM "File-Name",a,b ,c
SAVEM "CAS:File-Name",a,b ,c
SAVEM " RAM:File-Name",a,b ,c
```

wobei:

CSAVE und CSAVEM Programme im Binär-Format auf Kassette speichern

"File-Name" (Datei-Name) der Name des zu speichernden Programmes ist

A das Programm im ASCII-Format speichert

CAS das aktuelle Programm auf Kassette speichert

RAM das aktuelle Programm im RAM speichert

COM das aktuelle Programm in die Kommunikationsschnittstelle im ASCII-Format speichert.

M am Ende von SAVE oder CSAVE besagt, daß es sich um ein Maschinensprachen-Programm handelt

a,b,c Adressen sind.

CSAVE und SAVE "CAS..." (ohne die A-Option) sind die gleichen Befehle: Beide speichern BASIC-Programme in binärer Form auf Kassette. In ähnlicher Weise sind CSAVEM und SAVEM "CAS..." gleich, beide speichern Maschinensprachen-Programme auf Kassette.

SAVE "RAM:..." ist ähnlich dem Befehl SAVE "CAS:...", RAM kann dabei weggelassen werden.

SAVE kann auch benutzt werden, um das augenblickliche Programm auf dem Display oder einem Monitor anzuzeigen, und zwar mit dem Befehl SAVE "LCD:" (auf dem LCD-Display) oder SAVE "CRT:" für VDU.

Programme können per Datenübertragung auf andere Geräte gespeichert werden. Sende- und Empfangsbedingungen

müssen übereinstimmen. Das Empfangsgerät muß auf LOAD gesetzt werden, und es müssen die gleichen Parameter benutzt werden wie bei SAVE.

SCREEN

Anweisung

SCREEN bestimmt, über welches Sichtgerät die Ausgaben bei PRINT, LIST usw. erfolgen und ob die aktuelle Funktionstastenbelegung ständig angezeigt werden soll oder nicht.

Format: SCREEN d ,k

wobei: d das Ausgabemedium angibt: ist d = Ø, wird die Anzeige über LCD-Display ausgegeben, ist d = 1, wird die Anzeige über einen Monitor ausgegeben.

k kontrolliert die Anzeige der Funktionstastenbelegung: Bei k = Ø wird nicht angezeigt; bei k = 1 erfolgt die Anzeige.

Der Standardwert für sowohl d als auch k ist Ø. Wird d auf 1 gesetzt, und es ist kein Monitor angeschlossen, wird ein FC- (illegal function call-) Fehler angezeigt.

SGN

Funktion

SGN liefert eine Information über das Vorzeichen einer Zahl.

Format: SGN(x)

Diese Funktion liefert 1 für x > Ø, Ø für x = Ø und -1 für x < Ø.

SIN

Funktion

SIN liefert den Sinus eines Winkels.

Format: SIN(x)

wobei: x ein Winkel im Bogenmaß ist.

Beispiel:

```
Ok
PRINT SIN(Ø.5)
 .47942553860422
Ok
```

SOUND

Anweisung

SOUND spielt einen angegebenen Ton. SOUND ON/OFF aktiviert oder inaktiviert den Lautsprecher, wenn ein Programm von Kassette geladen wird oder die TELECOM-Einrichtung benutzt wird.

Format: SOUND Note, Dauer
SOUND ON
SOUND OFF

wobei: "Note" eine Ganzzahl zwischen Ø und 16383 ist; die unten aufgeführte Tabelle listet die Werte für die entsprechenden Noten auf.

"Dauer" eine Ganzzahl ist im Bereich zwischen Ø und 255; die Dauer des Tons beträgt immer (Dauer + 1) * 2Ø ms.

SOUND ON aktiviert den Lautsprecher und SOUND OFF inaktiviert ihn.

OKTAVE NOTE	1	2	3	4	5	6
C		9394	4697	2348	1171	587
C		8866	4433	2216	1103	554
D		8368	4184	2092	1043	523
D	15800	7900	3950	1975	987	493
E	14912	7456	3728	1864	932	466
F	14064	7032	3516	1758	879	439
F	13284	6642	3321	1660	830	415
G	12538	6269	3134	1567	783	
G	11836	5918	2954	1479	739	
A	11172	5586	2793	1396	693	
A	10544	5272	2636	1318	659	
B	9952	4968	2484	1244	622	

SPACE$

Funktion

SPACE$ liefert einen String aus einer bestimmten Anzahl Leerzeichen.

Format: SPACE$(x)

wobei: x eine Zahl zwischen Ø und 255 ist.

Wenn x keine Ganzzahl ist, werden die Nachkommastellen ignoriert.

Beispiel:

```
Ok
A$ = SPACE$(5)
Ok
PRINT "x";A$;"y";A$;"z"
x     y     z
Ok
```

SQR

Funktion

SQR liefert die Quadratwurzel einer Zahl.

Format: SQR(x)

wobei: x eine positive Zahl ist.

Wenn x negativ ist, wird ein FC- (illegal function call-) Fehler angezeigt.

Beispiel:

```
Ok
PRINT SQR(5Ø)
 7.Ø71Ø678118655
Ok
```

STOP

Anweisung

STOP beendet die Programmdurchführung und kehrt zum Direkt-Modus zurück.

Format: STOP

Diese Anweisung kann überall im Programm stehen, um die Programmausführung zu unterbrechen. Es wird die Meldung **Break in n** angezeigt.

wobei: n die Zeilennummer darstellt.

Alle Variablenwerte bleiben erhalten. Die Dateien bleiben offen, und das Programm kann fortgesetzt werden, indem der CONT-Befehl benutzt wird, vorausgesetzt, daß das Programm, während es gestopt wurde, nicht modifiziert (verändert) wurde. (Neuaufnahme oder Löschen von Programmzeilen, Editieren von Programmzeilen.)

STRING$

Funktion

STRING$ liefert einen String mit einer bestimmten Anzahl von einem bestimmten Zeichen.

Format: STRING$(a,b)
STRING$(a,x$)

wobei: a,b Ganzzahlen im Bereich zwischen Ø und 255 sind

x$ ein String ist.

Der ausgegebene String hat a Zeichen, die alle aus dem ASCII-Code b gebildet sind oder das erste Zeichen von x$ sind.

Beispiel:

```
Ok
A$ = STRING$(2Ø,47)
Ok
PRINT A$
////////////////////
Ok
```

STR$

Funktion

STR$ liefert die optisch gleiche String-Darstellung einer Zahl.

Format: STR$(x)

wobei: x eine Zahl ist

Die String-Darstellung einer Zahl wird z. B. bei der Datenübertragung an periphere Geräte eingesetzt.

Beispiel:

```
Ok
PRINT STR$(34)
 34
Ok
```

Die Funktion STR$ wandelt Zahlen in Strings um. Die Funktion VAL ist die Umkehrung von STR$. Die angezeigte Zahl, 34, wird Zeichen für Zeichen erzeugt, d. h. ein Byte für jedes Zeichen einer Zahl. Für positive Zahlen entsteht an erster Position ein Leerzeichen.

TAB

Funktion

TAB wird im Zusammenhang mit den Befehlen PRINT und LPRINT benutzt, um ab einer bestimmten Position die Ausgabe zu beginnen.

Format: TAB(i)

wobei: i eine Ganzzahl zwischen Ø und 255 ist.

Wird ein Wert für i angegeben, der sich vor der augenblicklichen Cursor-Position befindet, wird das nächste Zeichen direkt hinter den Cursor gedruckt.

Ø entspricht dem linken Rand des Bildschirms. Jeder Zahl entspricht eine bestimmte Position: 4Ø z. B. ist der linke Rand der zweiten Zeile und 255 ist das sechzehnte Zeichen in der siebten Zeile.

Beispiel:

```
Ok
PRINT "a" TAB(5) "b" TAB(3) "c" TAB(5Ø)
"d"
a     bc
          d
Ok
```

TAN

Funktion

TAN liefert den Tangens eines Winkels.

Format: TAN(x)

wobei: x ein Winkel im Bogenmaß ist.

Beispiel:

```
Ok
PRINT TAN(Ø.5)
 .54630248984381
Ok
```

TIME$, TIME$ ON/OFF/STOP

Besondere Variable und Anweisungen

In TIME$ wird die augenblickliche Zeit für die Uhr verwaltet. TIME$ ON/OFF/STOP-Anweisungen werden benutzt, um den Zugang zu einem Unterprogramm aus der ON TIME$ ... GOSUB-Anweisung zu ermöglichen oder zu inaktivieren.

Format: TIME$ = "hh:mm:ss"
TIME$

```
TIME$ ON
TIME$ OFF
TIME$ STOP
```

wobei: "hh:mm:ss" das Format für die augenblickliche Uhrzeit angibt.

Wenn Sie die Zeit festsetzen, benutzen Sie für hh (Stunde) Werte zwischen ØØ und 23. BEACHTEN SIE, DASS WERTE ZWISCHEN 24 UND 29 VOM SYSTEM ANGENOMMEN WERDEN, JEDOCH ALS ØØ REGISTRIERT WERDEN, WOHINGEGEN WERTE VON ÜBER 3Ø EINEN SN- (SYNTAX ERROR-) FEHLER ANZEIGEN.

Für mm (Minuten verwenden Sie Werte zwischen ØØ und 6Ø und entsprechend für ss (Sekunden). TIME$ ON aktiviert den Zugang zu einem Unterprogramm und TIME$ OFF inaktiviert diesen. TIME$ STOP hemmt den Zugang zu einem Unterprogramm, bis daß eine TIME$ ON-Anweisung abgearbeitet worden ist: Wird die Zeit-Festsetzung in einer ON TIME$ ... GOSUB-Anweisung übergeben, findet ein sofortiger Zugang zu einem Unterprogramm statt.

Wenn TIME$ als Variable benutzt wird, wird der augenblickliche Wert für die Zeit angegeben. Wird auf TIME$ ein neuer Wert zugewiesen, gilt dieser als neue Zeit, ab der vom System ständig weitergerechnet wird.

Beispiel:

```
Ok
PRINT TIME$
16:13:16
Ok
TIME$="19:Ø2:Ø7"
Ok
```

VAL

Funktion

VAL liefert den numerischen Wert eines Strings.

Format: VAL(x$)

wobei: x$ ein String ist.

Führende Leerzeichen, Tabulationsstops und Zeilenvorschübe werden ignoriert.

Beispiel:

```
PRINT VAL(" 34")
 34
Ok
```

Die Funktion STR$ ist die Umkehrung der Funktion VAL. Hat der String keinen numerischen Inhalt (z. B. "NAME"), liefert VAL den Wert Ø.

VARPTR

Funktion

VARPTR liefert die Adresse des ersten Bytes einer Variablen oder einer Datei im Arbeitsspeicher.

Format: VARPTR(x)
VARPTR(#n)

wobei: x eine Variable eines beliebigen Typs ist

n die Nummer ist, unter der die Datei geöffnet wurde.

Diese Funktion wird oft benutzt, um die Adresse einer Variablen zu erhalten, so daß diese an eine Maschinensprachen-Unterroutine gesendet werden kann.

Eine Variable, die durch VARPTR aufgerufen wird, muß vorher einen Wert erhalten haben, sonst wird ein FC-(illegal function call-) Fehler angezeigt.

Die angezeigte Zahl liegt im Bereich zwischen -32768 und 32767. Wird eine negative Zahl angezeigt, stellt sie eine Adresse zwischen 32768 (-32768) und 65535 (-1) dar.

Bei Arrays müssen alle einfachen Variablen vorher mit Werten versehen werden, und die Form VARPTR (Array(Ø)) sollte benutzt werden, um das niedrigste Adreß-Element auszugeben.

Für eine Datei liefert diese Funktion die Start-Adresse des Datenkontrollblocks.

WIDTH

Anweisung

WIDTH bestimmt die Breite des CRT-Monitors mit einer Breite von 40 oder 80 Zeichen.

Format: WIDTH n

wobei: n entweder 4Ø oder 8Ø ist und die Bildschirmbreite entsprechend festsetzt.

A. ZUSAMMENFASSUNG DER TECHNISCHEN SPEZIFIKATIONEN

Maße

30 x 22 x 6 cm

Gewicht

1900 Gramm

Stromversorgung

6V Netzteil oder 4 x 1,5 V Batterien

Mikropressor

80C85 (8 bits CPU=Zentraleinheit)

RAM

8k Byte Standard, erweiterbar auf 32k

ROM

32k Byte Standard, erweiterbar auf 64k.

B. BASIC-FEHLERMELDUNGEN

Fehler werden auf dem M10 mit den unten angegebenen Fehler-Codes angezeigt. Die Fehlernummern sind Ganzzahlen, die von BASIC erkannt werden und die angegeben werden müssen, wenn eine ERROR-Anweisung benutzt wird. Die Fehler werden in der zweiten Tabelle in numerischer Reihenfolge aufgelistet. Es stehen 255 Fehler-Nummern zur Verfügung, jedoch werden nicht alle vom BASIC-Interpreter genutzt. Nicht benutzte Fehler-Nummern sind nicht aufgeführt. Einige der Fehler-Nummern produzieren einen nicht druckbaren Fehler-Code (UE). Dies trifft auch auf alle Fehler-Nummern zu, die vom BASIC-Interpreter nicht genutzt werden.

FEHLER CODE	NR.	BESCHREIBUNG
AD	53	File already open: Es wurde versucht, bei sequentiellem Ausgabe-Modus (OPEN) eine bereits offene Datei erneut zu öffnen; oder es wurde versucht, eine offene Datei mit KILL zu löschen.
BS	9	Subscript out of range: Ein Array-Element wurde mit einer Index-Nummer, die außerhalb der Array-Größe liegt, angesprochen, oder aber es wurde die falsche Index-Nummer angegeben.
BN	51	Bad file number: Eine Anweisung oder ein Befehl bezieht sich auf eine Datei-Nummer, die nicht durch OPEN vergeben wurde oder die sich außerhalb der durch die MAXFILES festgelegten Anzahl liegt.
CF	58	File not open: Die Datei-Nummer, die in einer Anweisung wie PRINT#, INPUT# oder ähnlichen Anweisung gegeben wurde, wurde nicht durch OPEN vergeben.
CN	17	Cannot continue: Es wurde versucht, ein Programm fortzusetzen, was aus folgenden Gründen nicht geht:

1. Es wurde durch einen Fehler gestopt.

2. Es wurde während einer Unterbrechung eine Programmzeile verändert, eine neue Programmzeile aufgenommen oder eine vorhandene gelöscht.

3. Es besteht nicht.

DD 10 Redimensioned array: Es wurden zwei DIM-Anweisungen für ein Array gegeben, oder aber es wurde eine DIM-Anweisung gegeben für ein Array, das vorher automatisch auf den Standardwert dimensioniert wurde.

DS 56 Direct statement in file: In einer Datei im ASCII-Format wurde während des Ladens eine Anweisung ohne Zeilennummer gefunden; die LOAD-Anweisung wird gestopt.

EF 54 Input past end: Es wurde versucht, eine INPUT#-Anweisung durchzuführen, nachdem entweder alle Daten aus der Datei eingegeben worden sind oder aber bei einer leeren Datei. Benutzen Sie die EOF-Funktion, die das Ende der Datei anzeigt, um solch einen Fehler zu verhindern.

FC 5 Illegal function call: Es wurde versucht, einen Parameter, der außerhalb des zulässigen Bereiches liegt, an eine numerische oder String-Funktion zu übergeben. Diese Meldung tritt auch bei folgenden Fehlern auf:

1. Ein Index ist negativ oder größer als die maximale Größe.

2. Innerhalb einer LOG-Funktion befindet sich ein negativer Wert.

3. Innerhalb einer SQR-Funktion befindet sich ein negativer Wert.

4. Es wurde ein falsches Argument angegeben bei:

INP, INSTR, LEFT$, MID$, ON...GOTO, ON...GOSUB, OUT, PEEK, POKE, RIGHT$, SPACE$ oder STRING$.

FF	52	File not found: Eine LOAD-, KILL- oder OPEN-Anweisung bezieht sich auf eine Datei, die sich nicht im Speicher befindet.
FL	57	Too many files: Es wurde versucht, mehr als 19 Programme oder Datenfiles im RAM zu speichern.
ID	12	Illegal direct: Es wurde versucht, eine Anweisung, die im Direkt-Modus nicht zulässig ist, direkt (ohne Zeilennummer) einzugeben.
IE	50	Internal error: Ein interner Fehler ist aufgetreten. Benachrichtigen Sie Ihren Vertragshändler.
IO	23	Input/Output error: Ein Eingabe-/Ausgabefehler ist auf Kassette, Drucker oder auf dem Monitor aufgetreten. Dies ist ein schwerwiegender Fehler, da es z. B. passieren kann, daß BASIC nicht mehr aufgerufen werden kann.
LS	15	String too long: Es wurde versucht, einen String zu erstellen, der mehr als 255 Zeichen enthält.
MO	22	Missing operand: Ein Ausdruck enthält einen Operator, jedoch fehlt der dazugehörige Operand.
NF	1	NEXT without FOR: Eine Variable in einer NEXT-Anweisung bezieht sich auf keine vorher abgearbeitete Variable innerhalb einer FOR-Anweisung.
NM	55	Bad file name: Es wurde eine unzulässige Form für einen Datei-Namen benutzt, der mit LOAD, SAVE, KILL, NAME, OPEN usw. angegeben worden ist.

NR	19	No RESUME: Eine Fehlerbehandlungsroutine wurde erreicht. Es ist ein neuer Fehler aufgetreten, ohne daß vorher eine RESUME-Anweisung abgearbeitet wurde.
OD	4	Out of data: Es wurde versucht, eine READ-Anweisung abzuarbeiten, nachdem keine weiteren Elemente aus DATA-Anweisungen mehr vorhanden sind.
OM	7	Out of memory: Ein Programm ist zu lang; es enthält zu viele Dateien, zu viele FOR-Schleifen, zu viele GOSUBs, oder zu viele Variablen; oder es enthält zu komplexe Ausdrücke.
OS	14	Out of string space: Die String-Variablen verursachen, daß BASIC den freien Speicherraum überschreitet. BASIC verwaltet den String-Raum dynamisch, bis die Speichergrenze erreicht ist. (Reservieren Sie mit der CLEAR-Anweisung Speicherplatz für Strings.)
OV	6	Overflow: Das Ergebnis einer Berechnung ist zu groß, um im BASIC-Zahlenformat dargestellt zu werden.
RG	3	RETURN without GOSUB: Es wurde eine RETURN-Anweisung angesprochen, für die keine vorhergehende GOSUB-Anweisung erfolgte.
RW	20	RESUME without error: Es wurde eine RESUME-Anweisung aufgefunden, ohne daß die Fehlerbehandlungsroutine wegen eines Fehlers aktiviert worden ist.
SN	2	Syntax error: Es wurde eine Zeile aufgefunden, die eine falsche Zeichenfolge enthält, z. B. nicht geschlossene Klammern, falsch geschriebene Schlüsselworte oder falsche Zeichensetzung.

ST	16	String formula too complex: Ein String-Ausdruck ist zu lang oder zu komplex. Er sollte in kürzere Ausdrücke aufgeteilt werden.
TM	13	Type mismatch: Einem String-Variablen-Namen wurde ein numerischer Wert zugewiesen oder umgekehrt. Alternativ wurde einer Funktion, die ein numerisches Argument benötigt, ein String-Argument übergegeben oder umgekehrt.
UE	21	Unprintable error: Eine Fehlermeldung ist für die bestehende Fehler-Bedingung nicht vorhanden.
UE	24 - 49	Unprintable error: Diese Codes sind vorläufig nicht definiert und sollten für zukünftige Erweiterungen in BASIC reserviert bleiben.
UE	59 - 255	Unprintable error: Diese Codes sind nicht definiert und können vom Benutzer selbst verwendet werden (ERROR-Anweisung).
UL	8	Undefined line: Ein Zeilenverweis in einer GOTO-, GOSUB- oder IF...THEN... ELSE-Anweisung bezieht sich auf eine nicht existierende Zeile.
/0	11	Devision by zero: Es wurde versucht, eine Zahl durch Null zu teilen, oder aber es soll eine Null mit einem negativen Exponenten potenziert werden.

Die folgende Tabelle faßt die obengenannten Codes in numerischer Reihenfolge zusammen.

1	NF	NEXT without FOR.
2	SN	Syntax error.
3	RG	RETURN without GOSUB.
4	OD	Out of data.
5	FC	Illegal function call.
6	OV	Overflow.
7	OM	Out of memory.
8	UL	Undefined line.
9	BS	Subscript out of range.
10	DD	Redimensioned array.
11	/0	Division by zero.
12	ID	Illegal direct.
13	TM	Type mismatch.
14	OS	Out of string space.
15	LS	String too long.
16	ST	String formula too complex.
17	CN	Cannot continue.
19	NR	No RESUME.
20	RW	RESUME without error.
21	UE	Unprintable error.
22	MO	Missing operand.
23	IO	Input/Output error.
24 - 49	UE	Unprintable error.
50	IE	Internal error.
51	BN	Bad file number.
52	FF	File not found.
53	AO	File already open.
54	EF	Input past end.
55	NM	Bad file name.
56	DS	Direct statement in file.
57	FL	Too many files.
58	CF	File not open.
59 - 255	UE	Unprintable error.

BINAER - HEXADEZIMAL - DEZIMAL - Umrechnungstabelle

HEX	BIN	DEZ	DEZ*256	HEX	BIN	DEZ	DEZ*256	HEX	BIN	DEZ	DEZ*256	HEX	BIN	DEZ	DEZ*256
0	00000000	0	0	41	01000001	65	16640	82	10000010	130	33280	C3	11000011	195	49920
1	00000001	1	256	42	01000010	66	16896	83	10000011	131	33536	C4	11000100	196	50176
2	00000010	2	512	43	01000011	67	17152	84	10000100	132	33792	C5	11000101	197	50432
3	00000011	3	768	44	01000100	68	17408	85	10000101	133	34048	C6	11000110	198	50688
4	00000100	4	1024	45	01000101	69	17664	86	10000110	134	34304	C7	11000111	199	50944
5	00000101	5	1280	46	01000110	70	17920	87	10000111	135	34560	C8	11001000	200	51200
6	00000110	6	1536	47	01000111	71	18176	88	10001000	136	34816	C9	11001001	201	51456
7	00000111	7	1792	48	01001000	72	18432	89	10001001	137	35072	CA	11001010	202	51712
8	00001000	8	2048	49	01001001	73	18688	8A	10001010	138	35328	CB	11001011	203	51968
9	00001001	9	2304	4A	01001010	74	18944	8B	10001011	139	35584	CC	11001100	204	52224
A	00001010	10	2560	4B	01001011	75	19200	8C	10001100	140	35840	CD	11001101	205	52480
B	00001011	11	2816	4C	01001100	76	19456	8D	10001101	141	36096	CE	11001110	206	52736
C	00001100	12	3072	4D	01001101	77	19712	8E	10001110	142	36352	CF	11001111	207	52992
D	00001101	13	3328	4E	01001110	78	19968	8F	10001111	143	36608	D0	11010000	208	53248
E	00001110	14	3584	4F	01001111	79	20224	90	10010000	144	36864	D1	11010001	209	53504
F	00001111	15	3840	50	01010000	80	20480	91	10010001	145	37120	D2	11010010	210	53760
10	00010000	16	4096	51	01010001	81	20736	92	10010010	146	37376	D3	11010011	211	54016
11	00010001	17	4352	52	01010010	82	20992	93	10010011	147	37632	D4	11010100	212	54272
12	00010010	18	4608	53	01010011	83	21248	94	10010100	148	37888	D5	11010101	213	54528
13	00010011	19	4864	54	01010100	84	21504	95	10010101	149	38144	D6	11010110	214	54784
14	00010100	20	5120	55	01010101	85	21760	96	10010110	150	38400	D7	11010111	215	55040
15	00010101	21	5376	56	01010110	86	22016	97	10010111	151	38656	D8	11011000	216	55296
16	00010110	22	5632	57	01010111	87	22272	98	10011000	152	38912	D9	11011001	217	55552
17	00010111	23	5888	58	01011000	88	22528	99	10011001	153	39168	DA	11011010	218	55808
18	00011000	24	6144	59	01011001	89	22784	9A	10011010	154	39424	DB	11011011	219	56064
19	00011001	25	6400	5A	01011010	90	23040	9B	10011011	155	39680	DC	11011100	220	56320
1A	00011010	26	6656	5B	01011011	91	23296	9C	10011100	156	39936	DD	11011101	221	56576
1B	00011011	27	6912	5C	01011100	92	23552	9D	10011101	157	40192	DE	11011110	222	56832
1C	00011100	28	7168	5D	01011101	93	23808	9E	10011110	158	40448	DF	11011111	223	57088
1D	00011101	29	7424	5E	01011110	94	24064	9F	10011111	159	40704	E0	11100000	224	57344
1E	00011110	30	7680	5F	01011111	95	24320	A0	10100000	160	40960	E1	11100001	225	57600
1F	00011111	31	7936	60	01100000	96	24576	A1	10100001	161	41216	E2	11100010	226	57856
20	00100000	32	8192	61	01100001	97	24832	A2	10100010	162	41472	E3	11100011	227	58112
21	00100001	33	8448	62	01100010	98	25088	A3	10100011	163	41728	E4	11100100	228	58368
22	00100010	34	8704	63	01100011	99	25344	A4	10100100	164	41984	E5	11100101	229	58624
23	00100011	35	8960	64	01100100	100	25600	A5	10100101	165	42240	E6	11100110	230	58880
24	00100100	36	9216	65	01100101	101	25856	A6	10100110	166	42496	E7	11100111	231	59136
25	00100101	37	9472	66	01100110	102	26112	A7	10100111	167	42752	E8	11101000	232	59392
26	00100110	38	9728	67	01100111	103	26368	A8	10101000	168	43008	E9	11101001	233	59648
27	00100111	39	9984	68	01101000	104	26624	A9	10101001	169	43264	EA	11101010	234	59904
28	00101000	40	10240	69	01101001	105	26880	AA	10101010	170	43520	EB	11101011	235	60160
29	00101001	41	10496	6A	01101010	106	27136	AB	10101011	171	43776	EC	11101100	236	60416
2A	00101010	42	10752	6B	01101011	107	27392	AC	10101100	172	44032	ED	11101101	237	60672
2B	00101011	43	11008	6C	01101100	108	27648	AD	10101101	173	44288	EE	11101110	238	60928
2C	00101100	44	11264	6D	01101101	109	27904	AE	10101110	174	44544	EF	11101111	239	61184
2D	00101101	45	11520	6E	01101110	110	28160	AF	10101111	175	44800	F0	11110000	240	61440
2E	00101110	46	11776	6F	01101111	111	28416	B0	10110000	176	45056	F1	11110001	241	61696
2F	00101111	47	12032	70	01110000	112	28672	B1	10110001	177	45312	F2	11110010	242	61952
30	00110000	48	12288	71	01110001	113	28928	B2	10110010	178	45568	F3	11110011	243	62208
31	00110001	49	12544	72	01110010	114	29184	B3	10110011	179	45824	F4	11110100	244	62464
32	00110010	50	12800	73	01110011	115	29440	B4	10110100	180	46080	F5	11110101	245	62720
33	00110011	51	13056	74	01110100	116	29696	B5	10110101	181	46336	F6	11110110	246	62976
34	00110100	52	13312	75	01110101	117	29952	B6	10110110	182	46592	F7	11110111	247	63232
35	00110101	53	13568	76	01110110	118	30208	B7	10110111	183	46848	F8	11111000	248	63488
36	00110110	54	13824	77	01110111	119	30464	B8	10111000	184	47104	F9	11111001	249	63744
37	00110111	55	14080	78	01111000	120	30720	B9	10111001	185	47360	FA	11111010	250	64000
38	00111000	56	14336	79	01111001	121	30976	BA	10111010	186	47616	FB	11111011	251	64256
39	00111001	57	14592	7A	01111010	122	31232	BB	10111011	187	47872	FC	11111100	252	64512
3A	00111010	58	14848	7B	01111011	123	31488	BC	10111100	188	48128	FD	11111101	253	64768
3B	00111011	59	15104	7C	01111100	124	31744	BD	10111101	189	48384	FE	11111110	254	65024
3C	00111100	60	15360	7D	01111101	125	32000	BE	10111110	190	48640	FF	11111111	255	65280
3D	00111101	61	15616	7E	01111110	126	32256	BF	10111111	191	48896				
3E	00111110	62	15872	7F	01111111	127	32512	C0	11000000	192	49152				
3F	00111111	63	16128	80	10000000	128	32768	C1	11000001	193	49408				
40	01000000	64	16384	81	10000001	129	33024	C2	11000010	194	49664				

| HEX | BIN | DEZ | DEZ*256 | HEX | BIN | DEZ | DEZ*256 | HEX | BIN | DEZ | DEZ*256 | HEX | BIN | DEZ | DEZ*256 |

Das Handbuch dient der Information, sein Inhalt ist ohne ausdrückliche schriftliche Vereinbarung nicht Vertragsgegenstand. Technische Änderungen behalten wir uns vor. Die angegebenen Daten sind lediglich Nominalwerte.

olivetti

Druck Nr. 1828/1/1.84