

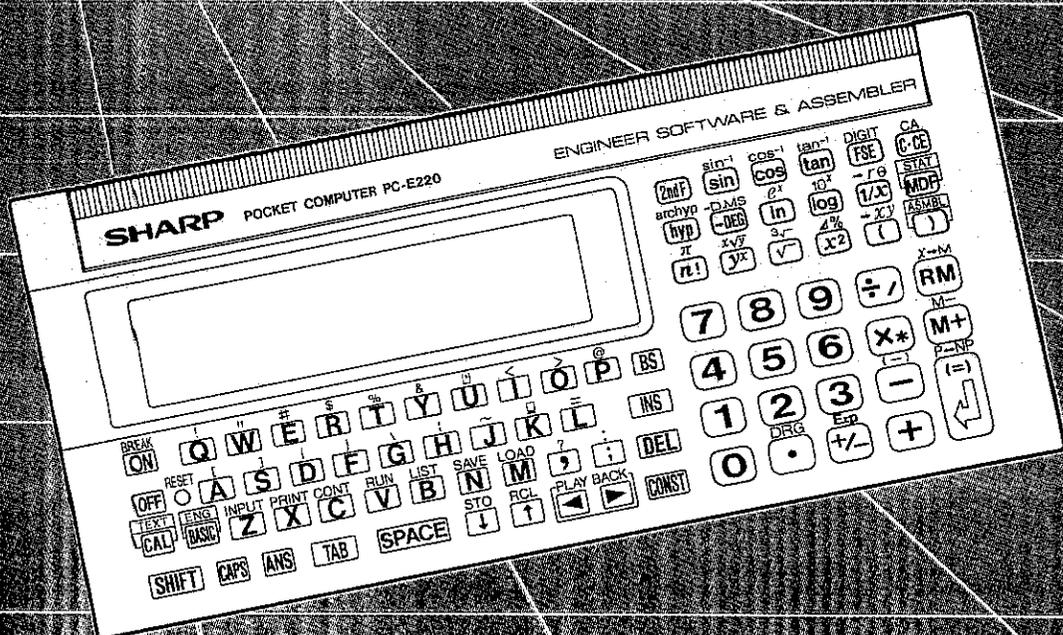
# SHARP®

## WASCHENCOMPUTER

MODELL

# PC-E220

## BEDIENUNGSANLEITUNG



**BESCHEINIGUNG DES HERSTELLERS/IMPORTEURS**

(nur für die Bundesrepublik Deutschland anwendbar)

Hiermit wird bescheinigt, daß der/die/das  
Taschenrechner, Modell PC-E220  
in Übereinstimmung mit den Bestimmungen der  
Vfg. 1046/1984

funk-entstört ist.

Der Deutschen Bundespost wurde das Inverkehrbringen dieses Gerätes angezeigt  
und die Berechtigung zur Überprüfung der Serie auf Einhaltung der Bestimmungen  
ingeräumt.

Sharp Electronics (Europe) GmbH

# EINFÜHRUNG

Dieser neue Taschencomputer PC-E220 von SHARP wurde nach dem neuesten Stand der Technologie entwickelt und verfügt über eine Vielzahl der fortschrittlichsten Funktionen:

- **Eingebauter Assembler:** Die Standardversion des PC-E220 ist mit einem Assembler ausgestattet, damit Sie sich einfach mit dem Programmieren in Maschinensprache vertraut machen können; zuerst schreiben Sie das Programm im TEXT-Modus, dann wird es in Maschinencode umgewandelt. Der PC-E220 enthält eine SMOS-CPU (Central Processing Unit), die dem bekannten Prozessor Z80 entspricht.
- **Wissenschaftliche Berechnungen:** Einfaches und leistungsfähiges Ausführen von wissenschaftlichen Berechnungen.
- **Statistische- und Regressionsberechnungen:** Ausführung von Berechnungen mit Einzel- oder Doppel-Variablen sowie lineare Regressionen.
- **Technikersoftware:** Mit der residenten Technikersoftware des Computers können Sie mathematische Gleichungen und physikalische Konstanten abrufen, metrische Umwandlungen ausführen und komplexe Zahlen verwenden.
- **Kompatibel mit bereits bestehender Software:** Software-Pakete, die für frühere Pocket- und tragbare Computer entwickelt wurden, sind mit dem PC-E220 kompatibel. Siehe Anhang H.
- **RAM-Diskette:** Ein Teil des internen Speichers kann wie eine RAM-Diskette verwendet werden. Damit können Sie Daten wie mit einer Diskette sichern und speichern.
- **Seriellere Eingabe-/Ausgabe-Interface:** Zur direkten Datenübertragung von Daten zwischen dem PC-E220 und einem Personal Computer.

Z80 ist ein eingetragenes Warenzeichen von Zilog, Inc.

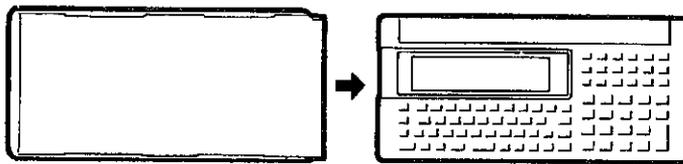
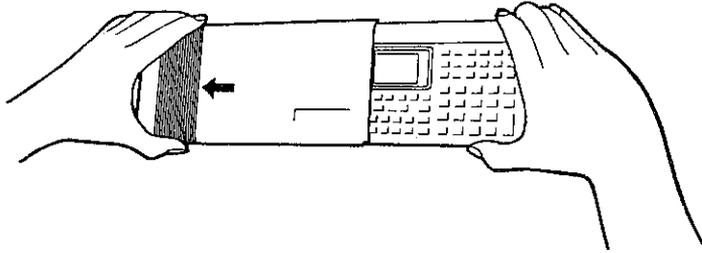
## HINWEISE

- SHARP empfiehlt sehr, daß separat permanente schriftliche Aufzeichnungen aller wichtigen Daten erstellt werden. Unter bestimmten Umständen können Daten in praktisch jedem Produkt mit elektronischem Speicher verlorengehen oder geändert werden. Daher übernimmt SHARP keine Haftung für Daten, die aufgrund von falscher Verwendung, Reparaturen, Defekten, Batteriewechsel, Verwendung nach Ablauf der angegebenen Batterielebensdauer oder aus irgendwelchen anderen Gründen verlorengehen oder andersweitig unbrauchbar werden.
- SHARP übernimmt keine Haftung, direkt oder indirekt, für finanzielle Verluste oder Schadensersatzansprüche Dritter, die aus der Verwendung dieses Produktes und seiner sämtlichen Funktionen entstehen, z.B. Verlust oder Veränderung von gespeicherten Daten usw.
- Änderungen der Informationen in dieser Bedienungsanleitung sind zum Zweck der Produktverbesserung ohne vorherige Bekanntgabe vorbehalten.

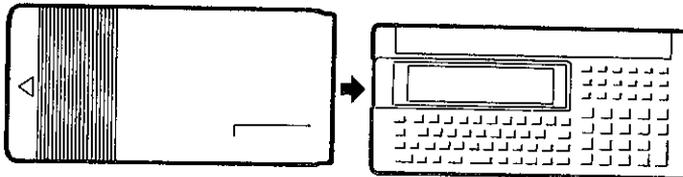
## Tastaturabdeckung

Zum Schutz der Tastatur und der Anzeige ist der Computer mit einer stabilen, doppelseitig aufschiebbaren Tastaturabdeckung ausgestattet.

Tastaturabdeckung bei Gebrauch des Rechners wie abgebildet vom Computer abnehmen.



Wenn der Computer nicht verwendet wird, die Tastaturabdeckung wie abgebildet über die Tastatur schieben.



# INHALTSVERZEICHNIS

EINFÜHRUNG .....	i
ERSTE VERWENDUNG DES PC-E220 .....	1
VERWENDUNG DIESES HANDBUCHS .....	4

## KAPITEL 1 HARDWARE

1. HARDWARE: ÜBERSICHT .....	6
2. HARDWARE: IM DETAIL .....	8
Einschalten .....	8
Automatische Abschaltung .....	8
Darstellung der Tasten in dieser Bedienungsanleitung .....	8
Wahl der Betriebsart .....	9
Das Display .....	10
Auswechseln der Batterien .....	12
3. PERIPHERE GERÄTE .....	16
CE-126P Thermodrucker/Kassetten-Interface .....	16
Kassetten-Interface .....	17
Verwendung des Kassetten-Interfaces .....	17
Verwendung des Drucker/Kassetten-Interfaces CE-126P .....	19

## KAPITEL 2 DIREKTEINGABE-BETRIEB

4. CAL-MODUS .....	22
Berechnungen .....	22
Grundlegende Bedienung .....	26
Wissenschaftliche Berechnungen .....	27
Verwendung der Klammern-Tasten .....	30
Nachkommastellen .....	31
Modifizierungs-Funktion .....	32
Vorrangordnung .....	32
5. TECHNIKERSOFTWARE-MODUS .....	34
Liste der Technikersoftware .....	35
Factorization (Zerlegung in Faktoren) .....	36
Trigonometric Function (Trigonometrische Funktionen) .....	38
Integration (Integralrechnung) .....	40
Greek (Griechisch) .....	42
Physical Constant (Physikalische Konstanten) .....	43
Metric Conversion (Metrische Umwandlung) .....	45
Complex Number (Komplexe Zahlen) .....	48
6. STAT-MODUS .....	50
Wählen und Abschalten des STAT-Modus .....	50
Statistische Berechnungen mit Einzel-Variablen .....	50
Statistische Berechnungen mit Doppel-Variablen .....	54

7. RUN-MODUS .....	58
Einige nützliche Hinweise .....	58
Einfache Berechnungen .....	59
Komplexere Berechnungen und Klammersetzung .....	59
Wiederabruf von Eingaben .....	60
Fehler .....	62
Verkettung von Berechnungen .....	63
Berechnungen mit Konstanten .....	63
Die Verwendung von Variablen in Berechnungen .....	64
Letztes-Ergebnis-Funktion .....	65
Maximale Berechnungslänge .....	66
Wissenschaftliche Berechnungen .....	66
Vorrangordnung bei manuellen Berechnungen .....	70
Ausdrucken von Rechenergebnissen .....	71
Rechenfehler .....	71

## KAPITEL 3 PROGRAMMIEREN

8. BEGRIFFE UND AUSDRÜCKE DES BASIC .....	74
Zeichenfolgen-Konstanten .....	74
Hexadezimalzahlen .....	74
Variablen .....	74
Programm-Dateien (RAM-Diskette) .....	80
Dateinamen .....	80
Erweiterung .....	80
Ausdrücke .....	80
Numerische Ausdrücke .....	81
Zeichenfolge-Ausdrücke .....	81
Verhältnis-Ausdrücke .....	81
Logische Ausdrücke .....	82
Klammerung und Vorrang der Operatoren .....	83
9. PROGRAMMIEREN .....	84
Programme .....	84
BASIC-Anweisungen .....	84
Zeilennummern .....	84
Programme mit Labels .....	85
BASIC-Kommandos (Befehle) .....	85
Direkt-Kommandos .....	86
Modi (Betriebsarten) .....	86
Der Anfang des Programmierens .....	86
Speichern von Programmen im Computer .....	92
10. FEHLERSUCHE .....	94
Vorgehen bei der Fehlersuche .....	95

## KAPITEL 4 ASSEMBLER

11. TEXT-MODUS (TEXT-EDITOR) .....	98
Funktionen des Text-Modus .....	98

Wahl der TEXT-Betriebsart .....	99
Editieren .....	99
Editieren von Programmen .....	100
Löschen eines TEXT-Programms (Del) .....	101
Ausdrucken einer TEXT-Programmauflistung (Print) .....	101
Sichern, Laden und Vergleichen eines TEXT-Programms mit einer Kassette (Cmt) .....	101
Serielle Eingabe/Ausgabe (Sio) .....	104
Programm-Dateien (File) .....	107
Der BASIC-Konverter (Basic) .....	109
<b>12. MASCHINENSPRACHEN-MONITOR .....</b>	<b>112</b>
Verwendung des Maschinensprachen-Monitors .....	112
Auflistung der Befehle für den Maschinensprachen-Monitor .....	113
Fehlermeldungen des Monitor-Modus .....	119
<b>13. ASSEMBLER .....</b>	<b>120</b>
Programmieren mit Assembler .....	121
Codieren und Editieren eines Quellprogramms .....	124
Assemblieren .....	128
Pseudobefehle für den Assembler .....	132
Assemblerfehler .....	136

## KAPITEL 5 COMPUTER-ANWEISUNGEN

<b>14. WISSENSCHAFTLICHE UND MATHEMATISCHE BERECHNUNGEN .....</b>	<b>140</b>
Rechenbereiche .....	149
<b>15. COMPUTER-ANWEISUNGEN (BASIC KOMMANDO LEXIKON) .....</b>	<b>151</b>

## KAPITEL 6 ANHÄNGE

<b>A DATENÜBERTRAGUNGSKABEL CE-T801 .....</b>	<b>214</b>
<b>B FEHLERMELDUNGEN .....</b>	<b>216</b>
<b>C TABELLE DER ZEICHENCODES .....</b>	<b>218</b>
<b>D TASTENFUNKTIONEN IN BASIC .....</b>	<b>220</b>
<b>E FEHLERSUCHE .....</b>	<b>223</b>
<b>F AUFTEILUNG DES SPEICHERBEREICHES .....</b>	<b>225</b>
<b>G TECHNISCHE DATEN .....</b>	<b>226</b>
<b>H VERWENDUNG VON PROGRAMMEN VON ANDEREN   SHARP-COMPUTERN .....</b>	<b>228</b>
<b>I PFLEGE DES PC-E220 .....</b>	<b>230</b>

REGISTER DER KOMMANDOS  
REGISTER

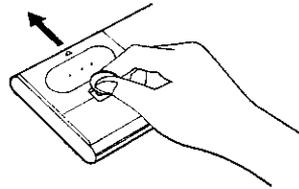


# ERSTE VERWENDUNG DES PC-E220

## Einsetzen der Batterien

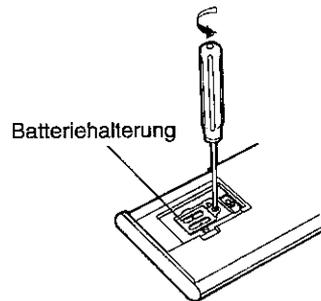
Für den PC-E220 werden zwei Arten von Batterien verwendet: vier Batterien vom Typ AA für die Stromversorgung des Computers selbst und eine Lithiumbatterie für die Speichersicherung. Die mitgelieferten vier Betriebsbatterien vom Typ AA und die Speichersicherungs-Batterie (Lithium) werden folgendermaßen eingesetzt:

1. Den Computer mit der Oberseite nach unten auf die Abdeckung legen und die Schraube zur Sicherung des Batteriefachs mit einer Münze oder einem Flachkopf-Schraubenzieher lösen.

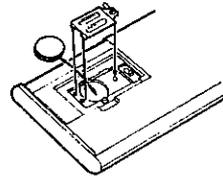


2. Die Abdeckung des Batteriefachs entfernen, indem sie in Pfeilrichtung auf die Seite geschoben wird.

3. Die Batteriehalterung für die Speichersicherungs-Batterie mit einem kleinen Schraubenzieher aufschrauben und die Halterung entfernen.

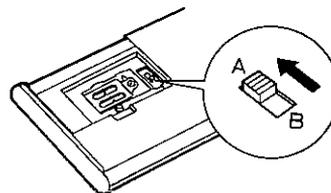


4. Die mitgelieferte Lithiumbatterie mit einem trockenen Tuch abreiben und mit der positiven Seite nach unten in das Fach einlegen.



5. Die Batteriehalterung wieder anbringen und mit der Schraube befestigen.

6. Den Speicherschutz-Schalter auf die Position A schieben.





3. Eine beliebige Taste drücken. Folgende Anzeige erscheint:

```
RUN MODE
>
```

## Überprüfen der normalen Computerfunktion

Zur Sicherstellung der normalen Computerfunktion folgenden Tasten drücken:



```
RUN MODE
FRE
30435.
```

Wenn die oben dargestellte Anzeige erscheint, funktioniert der Computer normal und ist bereit für Eingaben.

Die Zahl "30435." bedeutet die Speicherkapazität für Programme oder Daten.

### Hinweis:

Wenn der PC-E220 nach den oben beschriebenen Schritten nicht diese Anzeige darstellt, sollte die Anweisung für den entsprechenden Schritt noch einmal gelesen werden. Versuchen Sie noch einmal die richtige Eingabe für diesen Schritt.

Falls die Anzeige ~~ERR~~ erscheint, siehe Seite 12.

# VERWENDUNG DIESES HANDBUCHS

Dieses Handbuch soll Sie mit den Fähigkeiten und Charakteristiken Ihres Computers vertraut machen und Ihnen als wertvolles Nachschlagewerk dienen. Es wurde in sechs Abschnitte eingeteilt, jeder mit einer Einleitung versehen, die einen speziellen Aspekt des PC-E220 behandeln. So ist das Handbuch letztlich ein Nachschlagewerk und wir empfehlen Ihnen, die einleitenden Abschnitte im Kapitel 1 (Hardware) und Kapitel 2 (Bedienung) sorgfältig durchzulesen. Der PC-E220 ist ein äußerst wirkungsvolles Werkzeug mit vielen wertvollen und zeitsparenden Funktionen, deren Entdeckung sogar alte Computerhasen erfreuen dürfte!

## **Kapitel 1: Hardware**

Der erste Abschnitt von Kapitel 1 ist eine Einführung in die speziellen Eigenschaften des Computers. Abschnitt 2 beschreibt die Grundausrüstung des Computers, die Bedienung der Tasten und die Bedeutungen der verschiedenen Displaysymbole. Die Lektüre von Abschnitt 1 und 2 ist deshalb äußerst wichtig. Die folgenden Abschnitte behandeln die Verwendung der peripheren Geräte (wie Drucker und Kassettenrekorder).

## **Kapitel 2: Direkteingabe-Betrieb**

Kapitel 2 ist dem Gebrauch des Computers als Rechner (für wissenschaftliche, technische oder statistische Zwecke) oder "Direkteingabe"-Computer gewidmet. Direkte Eingabe bezieht sich auf die unabhängige Verwendung von BASIC-Befehlen (d.h. Befehle nicht innerhalb eines BASIC-Programms).

## **Kapitel 3: Programmier-Betrieb**

Kapitel 3 führt Sie in die BASIC-Programmiersprache zur Anwendung des PC-E220 ein. Auch wenn Sie bereits früher BASIC-Programme durchgeführt haben, so empfehlen wir Ihnen doch, diesen Teil durchzulesen, da unterschiedliche Versionen der BASIC-Sprache existieren. Dieses Kapitel enthält auch zeitsparende Information in der Form von Programmier-Vereinfachungen und Techniken zur Fehlerbeseitigung.

## **Kapitel 4: Assembler**

Kapitel 4 beschreibt die Monitor- und Assembler-Funktion des PC-E220, die Ihnen beim Erlernen des Programmierens in Maschinensprache helfen soll. Der Assembler, der für die CPU dieses PC-E220 verwendet wurde, entspricht dem Zilog Z80. Er soll Sie mit der Arbeit in der Zilog-Maschinensprache vertraut machen.

## **Kapitel 5: BASIC-Referenz**

Kapitel 5 enthält eine alphabetische Auflistung der numerischen Funktionen und BASIC-Begriffe, die zur Programmierung des Computers gebraucht werden. Viele dieser Begriffe können im direkten Eingabemodus des Computers benutzt werden.

## **Kapitel 6: Anhang**

Kapitel 6 enthält in der Hauptsache Nachschlagematerialien, wie Code-Tabellen, Fehlermeldungen und Zusammenstellungen von Kenndaten. Außerdem finden Sie hier Hinweise, wie Sie Ihren Computer funktionstüchtig halten.

Dieses Handbuch ist nicht als Lehrbuch zum Selbsterlernen von BASIC konzipiert, eine vollständige Beschreibung ist nicht die Absicht dieses Handbuches. Wenn Sie bisher noch nicht in BASIC programmiert haben, empfehlen wir Ihnen, daß Sie sich hierfür ein separates Lehrbuch zulegen oder an einem BASIC-Kurs teilnehmen, bevor Sie dieses Handbuch durcharbeiten.

## HARDWARE

Die ersten Abschnitte von Kapitel 1 sind eine Einführung in die speziellen Eigenschaften des PC-E220\*, eine Beschreibung der Grundausrüstung des Computers, der Bedienung der Tasten und der Bedeutungen der verschiedenen Displaysymbole.

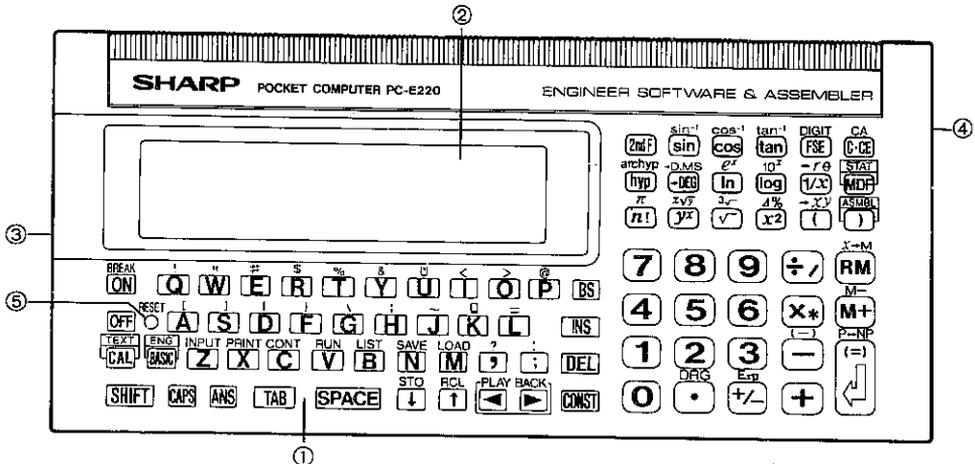
Die folgenden Abschnitte behandeln die Verwendung der peripheren Geräte (z.B. Drucker und Kassettenrecorder).

\* Der PC-E220 wird im folgenden als "der Computer" bezeichnet.

# 1. HARDWARE: ÜBERSICHT

Die Ausstattung des SHARP Computers besteht aus einer QWERTY-Tastatur, die der von herkömmlichen Schreibmaschinen ähnelt, einem Flüssigkristall-Display (LCD) mit einstellbarem Kontrast und einem 11-Stift-Interface-Stecker.

Auf den folgenden Seiten werden die einzelnen Teile des Computers beschrieben, um Sie mit ihren Positionen und Funktionen bekannt zu machen.



Die Tasten **BREAK ON** und **OFF** wurden in die Tastatur eingelassen, damit sie nicht aus Versehen gedrückt werden.

## 1. Tastatur

Die Anordnung der Tasten auf der Tastatur entspricht der einer normalen Schreibmaschine; weiterhin ist eine Zahlen-Tastatur vorhanden. Insgesamt hat das Gerät 82 Tasten.

## 2. LCD-Bildschirm

Der Bildschirm besteht aus vier Zeilen mit je 24 Stellen. Der Kontrast kann den jeweiligen Sichtverhältnissen optimal angepaßt werden.

## 3. 11-Stift-Stecker

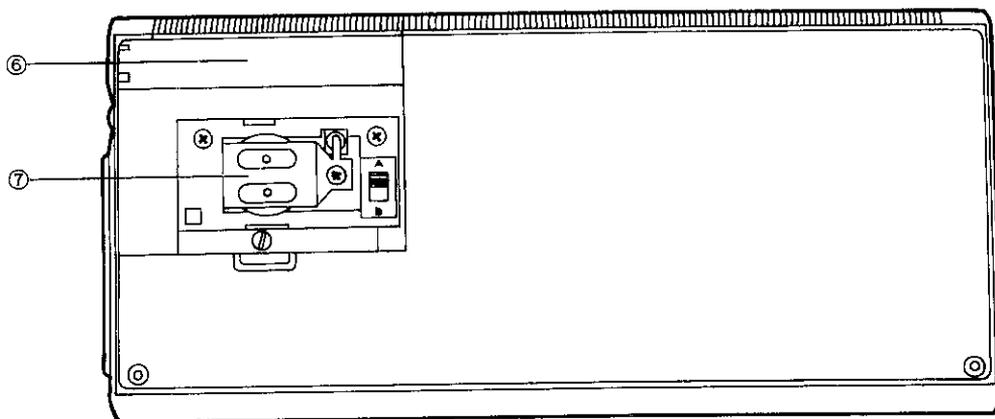
Zum direkten Anschließen des Computers an ein zusätzliches peripheres Gerät, z.B. Drucker (CE-126P) und Kassettenrecorder. Der PC-E220 kann über das getrennt erhältliche Datenübertragungskabel CE-T801 mit einem Personal Computer verbunden werden.

## 4. LCD-Kontrastregler

Hier kann der Bildschirm aufgehellt oder abgedunkelt und somit auf die aktuellen Lichtverhältnisse eingestellt werden.

## 5. Rücksetzschalter (RESET)

Falls der Computer aus irgend einem Grund während des Betriebs "bockt", arbeitet die Tastatur nicht. Auch kann es unmöglich sein, den Ein/Aus-Schalter oder die Taste **BREAK** zum erneuten Start des Computers zu betätigen. Drücken Sie den RESET-Schalter zum Löschen des Speichers, wodurch der Computer wieder in den Zustand wie beim Stromeinschalten (Power ON) kommt. Benützen Sie diesen Schalter mit Vorsicht, da Sie den Verlust von Daten und Programmen riskieren. Zur weiteren Information über den Gebrauch dieses Schalters lesen Sie Anhang E.



## 6. Betriebsbatteriefach

In diesem Fach befinden sich die vier Trockenbatterien als Stromquelle für den Computer. Im Abschnitt 2 finden Sie Hinweise, wie Sie die Batterien ersetzen.

## 7. Speicherbatteriefach

In diesem Fach befindet sich eine Lithiumbatterie für die Speichersicherung des internen Speichers.

## 2. HARDWARE: IM DETAIL

### Einschalten

Die **ON** -Taste auf der linken Seite der Computertastatur drücken. Der Computer befindet sich nach dem Einschalten in der RUN-Betriebsart.

Wenn der Computer ein- oder ausgeschaltet wird, kann es vorkommen, daß kurzfristig keine Anzeige auf dem Bildschirm dargestellt werden kann, oder es werden unregelmäßige Punkte, Linien oder Symbole angezeigt. Dies deutet nicht auf eine Fehlfunktion hin.

Falls die Anzeige **BATT** erscheint, siehe Seite 12.

### Automatische Abschaltung

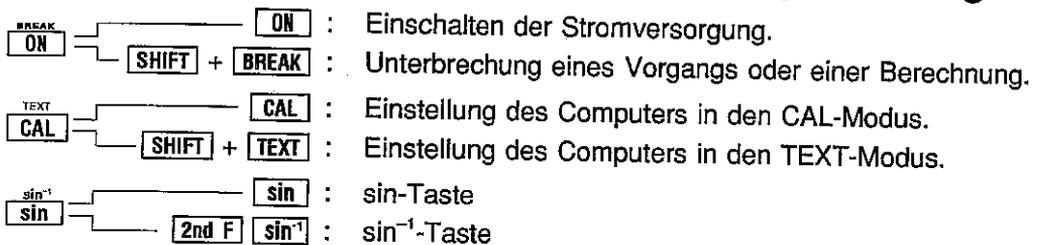
Zum Schutz der Batterien schaltet sich der Computer automatisch selbst aus, wenn etwa 11 Minuten lang keine Tasten betätigt werden.

Die **ON** -Taste zum erneuten Einschalten drücken, wenn sich der Computer selbst abgeschaltet hat.

Während der Computer einen INKEY\$- Befehl ausführt, ist die automatische Abschaltfunktion wirkungslos. Sie ist aber wirksam, während der Computer einen INPUT-Befehl ausführt.

Wenn der Computer längere Zeit nicht benutzt wird, während die automatische Abschaltung nicht wirksam ist, wird Batteriestrom verbraucht. Es kann dabei zum Verlust von gespeicherten Programmen bzw. Daten kommen.

### Darstellung der Tasten in dieser Bedienungsanleitung



Wenn in dieser Anleitung von Zahlen, Buchstaben oder den auf oder über den Tasten gedruckten Symbolen die Rede ist, werden nur die im Zusammenhang notwendigen Zeichen angegeben, während die Erwähnung der **SHIFT** -Taste unterbleibt, wie im folgenden Beispiel dargestellt:

**S** **H** **A** **R** **P** → SHARP  
**SHIFT** + **R** **4** **5** → \$45

Bei den Erklärungen der Rechner-Betriebsart wird die Taste  als  dargestellt, während sie bei Erklärungen der anderen Betriebsarten als  angegeben wird.

### Unterschied zwischen der **SHIFT** - und der **2nd F** -Taste

Die **SHIFT** und die **2nd F** -Taste sind funktionell identisch. Der einzige Unterschied besteht darin, daß die **SHIFT** -Taste gedrückt gehalten werden muß, während eine andere Taste gedrückt wird; die **2nd F** muß nur einmal vor dem Drücken einer andern Tasten gedrückt werden. Bei der Beschreibung von Berechnungen in der CAL-Betriebsart und bei Berechnungen von Funktionen wird im Rahmen dieser Anleitung immer die **2nd F** -Taste verwendet, während die **SHIFT** -Taste für alle anderen Beschreibungen verwendet wird.

Wenn eine Leerstelle mit der **SPACE** -Taste eingegeben werden muß, wird dies durch das Symbol **\_** angezeigt, z.B.

"SHARP\_**\_**EL-865\_**\_**WN-104"

Bei Bedarf werden in dieser Anleitung die Tasten als Kästen mit Symbolen dargestellt, z.B. **sin**.

Zur Unterscheidung der Zahl Null und dem Großbuchstaben "O" erscheint die Null auf dem Display des PC-E220 als "0". Bei den Beschreibungen dieser Anleitung erscheint die Zahl Null ebenfalls als "0", wenn es zu Verwechslungen kommen könnte.

## Wahl der Betriebsart

Bevor Sie den Computer in Gang setzen, müssen Sie eine Betriebsart bestimmen. Der Computer hat sechs Betriebsarten.

- BASIC-Modus:** Der BASIC-Modus ist in RUN und PRO unterteilt.  
**PRO-Modus:** Zum Schreiben oder Korrigieren von BASIC-Programmen.  
**RUN-Modus:** Zum Ausführen von BASIC-Programmen oder BASIC-Kommandos.
- CAL-Modus:** Zur Verwendung des Computers wie einen normalen Rechner.  
**ENG-Modus:** Zur Verwendung der Technikersoftware.  
**STAT-Modus:** Zur Ausführung von statistischen und Regressions-Berechnungen.
- TEXT-Modus:** Für Eingabe, Bearbeitung, Löschen, Speichern oder Laden eines TEXT-Programms im ASCII-Format oder zur Umwandlung nach BASIC und umgekehrt.
- ASMBL-Modus:** Zum Assemblieren eines Quellprogramms (TEXT) in ein Maschinencode-Programm (Objekt).

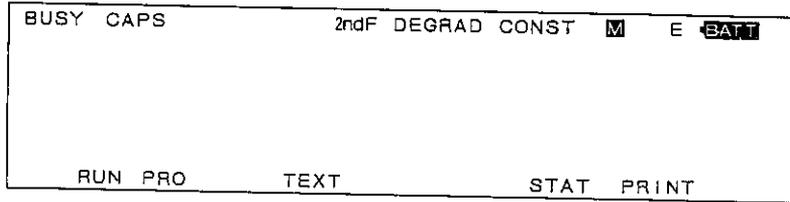
### Verwendete Tasten

Die folgende Tabelle zeigt die Tasten, die zum Aufrufen der Betriebsarten verwendet werden.

Zu wählende Betriebsart	Tastenbedienung
RUN-Modus	<b>BASIC</b>
PRO-Modus	<b>BASIC</b> oder <b>BASIC</b> <b>BASIC</b>
CAL-Modus	<b>CAL</b>
ENG-Modus	<b>SHIFT</b> + <b>ENG</b>
STAT-Modus	<b>SHIFT</b> + <b>STAT</b>
TEXT-Modus	<b>SHIFT</b> + <b>TEXT</b>
ASMBL-Modus	<b>SHIFT</b> + <b>ASMBL</b>

← Zum Umschalten nach PRO von einer anderen als der RUN-Betriebsart wird **BASIC** zweimal gedrückt.

# Das Display



Der Computer hat ein 4-Zeilen-Flüssigkristall-Display mit 24 Zeichen pro Zeile und einer Statuszeile am oberen und am unteren Rand. Jedes Zeichen besetzt eine 5 x 7 Punktmatrix. Das Display zeigt Tastenbezeichnungen und Berechnungsvorgänge. Die Display-Beispiele dieser Anleitung geben nur die für die jeweilige Erklärung der Funktion notwendigen Symbole wieder.

In der BASIC-Betriebsart zeigt das Display folgendes:

- > Bereitschaftssymbol. Dieses Symbol erscheint, wenn der Computer in der BASIC-Betriebsart bereit ist, eine Eingabe anzunehmen. Beim Tippen verschwindet das Bereitschaftssymbol und wird durch den Cursor ersetzt.
- ┌, █ Der Cursor. Dieses Symbol markiert die Stelle des nächsten einzugebenden Zeichens. Wenn man mit dem Tippen beginnt, ersetzt der Cursor das Bereitschaftssymbol. Als Markierungssymbol wird der Cursor auch im Zusammenhang mit den **INS**ert- und **DE**Lete-Funktionen benutzt. Der Unterstrich-Cursor ändert sich zum Block-Cursor, wenn er sich nicht auf einem Zeichen befindet.

Die Statuszeilen geben folgendes wieder:

- BUSY** Dieses Wort erscheint auf dem Display, wenn der Computer ein Programm oder ein Kommando ausführt.
- CAPS** Zeigt an, daß sich der Computer in der Betriebsart für Großbuchstaben (CAPS) befindet. Wenn diese Anzeige nicht auf dem Display erscheint, werden alle Buchstaben des Alphabets als Kleinbuchstaben eingegeben. Mit der **[CAPS]**-Taste kann die CAPS-Betriebsart ein- und wieder ausgeschaltet werden.
- 2ndF** Die Anzeige erscheint, wenn die **[2nd F]**-Taste gedrückt wurde und erlischt mit dem folgenden Tastenkommando. Denken Sie daran, daß die **[2nd F]**-Taste vor dem Druck auf eine andere Taste freigegeben werden muß, wenn die zweite Funktion dieser Taste benutzt werden soll.
- DEG** Zeigt für Winkelfunktionen den Altgrad-Modus an.
- RAD** Zeigt für Winkelfunktionen den Bogenmaß-Modus an.
- GRAD** Zeigt für Winkelfunktionen den Neugrad-Modus an.

- CONST** Zeigt an, daß im Computer eine Konstante für Berechnungen mit Konstanten gespeichert ist (siehe Seite 63). Wenn dieses Symbol angezeigt wird, führt der Computer jedesmal beim Drücken der Taste  eine Berechnung mit dieser Konstante aus. Wenn die Konstante nicht mehr benötigt wird, kann sie mit **SHIFT** + **CA** gelöscht werden.
- M** Zeigt an, daß eine andere Zahl außer Null für manuelle Berechnungen gespeichert ist.
- E** Hiermit wird angezeigt, daß ein Fehler (ERROR) aufgetreten ist.
- BATT** Dieses Symbol weist Sie darauf hin, daß die Batterien schwach sind und ausgewechselt werden müssen.
- RUN** Dieses Symbol zeigt den RUN-Modus für den Computer an.
- PRO** Dieses Symbol zeigt den PROgrammierungs-Modus für den Computer an.
- TEXT** Zeigt an, daß sich der Computer in der TEXT-Betriebsart befindet. Zur Wahl der TEXT-Betriebsart die Tasten **SHIFT** + **TEXT** verwenden.
- STAT** Zeigt an, daß sich der Computer in der STATistik-Betriebsart befindet. Diese Betriebsart wird jedesmal beim Drücken von **2nd F** **STAT** (bzw. **SHIFT** + **STAT**) aktiviert bzw. deaktiviert.
- PRINT** Dieses Symbol zeigt an, daß der Computer bereit ist, im RUN-Modus Daten zum Drucker zu senden. Drücken Sie **SHIFT** + **P-HP** zum Ein- und Ausschalten (ON und OFF). (Nur möglich mit einem angeschlossenen zusätzlichen Drucker.)

In der Rechner-Betriebsart (CAL) werden auf der ersten Zeile "<CAL>", "HYP", "FIX" oder andere Symbole angezeigt. "<CAL>" bedeutet, daß sich der Computer in der Rechner-Betriebsart befindet. Für "FIX" siehe die Erklärung auf Seite 24.

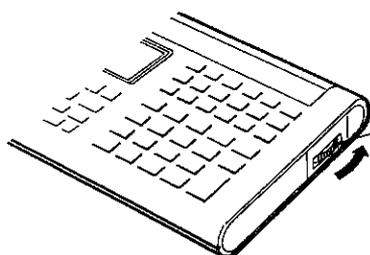
#### Hinweis:

Die japanischen Symbole "カナ" und "小" oder "CASL" können oben rechts auf dem Display angezeigt werden. Diese Symbole beeinträchtigen nicht die Funktionen oder die Bedienung des Computers.

#### Zur Anzeige von mehr als 4 Zeilen

Das Display des Computers besteht aus 4 Zeilen (24 Zeichen pro Zeile). Tastatureingaben bzw. Rechenergebnisse werden von der ersten Zeile auf dem Display angezeigt. Werden die 4 Zeilen durch die eingegebenen Zeilen überschritten, so werden die angezeigten Inhalte eine Zeile nach oben bewegt (die erste angezeigte Zeile verschwindet hinter der Oberkante des Bildschirms).

## Kontrastregulierung



Kontrastregler

Durch Drehen des Knopfes in Pfeilrichtung wird der Kontrast der Anzeige verstärkt und in entgegengesetzter Richtung verringert. Bitte stellen Sie ihn so ein, daß die Anzeige gut ablesbar ist.

## Auswechseln der Batterien

Der Computer verwendet zwei Arten von Batterien, nämlich vier Trockenbatterien vom Typ AA für den Betrieb des Computers selbst und eine Lithiumbatterie für die Speichersicherung.

Wenn die Trockenbatterien des Computers zu schwach sind, während der Drucker CE-126P gleichzeitig mit dem Computer verwendet wird, kann er auch über den CE-126P mit Strom versorgt werden. Dadurch wird die Belastung der internen Batterien vermindert.

### Zeitpunkt des Wechsels der Betriebsbatterien

Wenn in der oberen rechten Ecke des Displays die Warnanzeige **BATT** erscheint, bedeutet dies, daß die Batterien zu schwach sind. Sie sollten sofort durch neue ersetzt werden. Wenn der Computer weiterhin verwendet wird, obwohl **BATT** angezeigt wird, schaltet sich der Computer nach einiger Zeit aus. Danach kann er auch durch Drücken der **ON**-Taste nicht mehr eingeschaltet werden.

Wenn ein zusätzliches Peripheriegerät angeschlossen wurde, kann der Computer über dieses Gerät mit Strom versorgt werden. In diesem Fall erscheint die Warnanzeige **BATT** nicht, obwohl die Betriebsbatterien des Computers zu schwach sind. Vor dem Gebrauch sollte das Peripheriegerät kurzfristig abgetrennt werden um zu prüfen, ob die Warnanzeige **BATT** auf dem Display erscheint oder nicht.

### Hinweis:

Wenn die Trockenbatterien ersetzt werden, bleibt der Speicherinhalt durch die Lithiumbatterien gesichert. Falls zusätzliche periphere Geräte vorhanden sind, empfiehlt es sich, die Programme vom internen Speicher des Computers auf dieses Speichermedium zu übertragen.

### Ersetzen der Betriebsbatterien

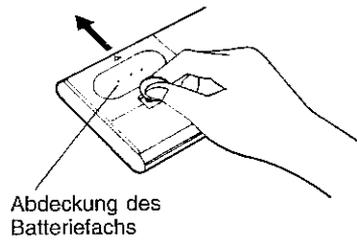
Zum Ersetzen der Trockenbatterien folgendermaßen vorgehen. Wenn die Anweisungen nicht richtig befolgt werden, kann der Computer unter Umständen nicht benutzt oder die Lebensdauer der Speicherschutz-Batterien kann verkürzt werden.

Folgende Batterie-Typen verwenden:

Betriebsbatterie: vier Trockenbatterien vom Typ AA (R06)

Batterie für Speichersicherung: eine Lithiumbatterie (CR2032)

1. **ON** und dann **OFF** drücken, um den Computer abzuschalten.
2. Mit einer Münze oder einem Flachkopf-Schraubenzieher die Schraube zur Sicherung der Abdeckung des Batteriefachs auf der Rückseite des Computers lösen und die Abdeckung abnehmen.



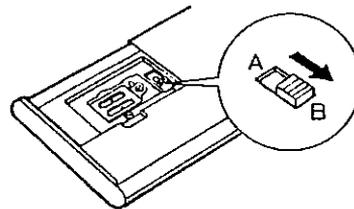
**Hinweis:**

Beim Abnehmen der Abdeckung darauf achten, daß nicht aus Versehen die Taste **ON** auf der Vorderseite des Gerätes gedrückt wird. Falls die Taste doch gedrückt wird, muß der Computer durch Drücken von **OFF** wieder ausgeschaltet werden. Falls zum folgenden Schritt 3 übergegangen wird, während das Gerät eingeschaltet ist, wird der gesamte Speicherinhalt zerstört oder vollständig gelöscht.

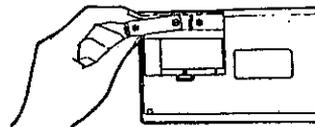
3. Den Speicherschutz-Schalter auf die Position B schieben.

**Hinweis:**

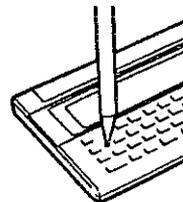
Falls die Batterien ersetzt werden, während sich der Schalter in der Position A befindet, wird der gesamte Speicherinhalt zerstört oder vollständig gelöscht.



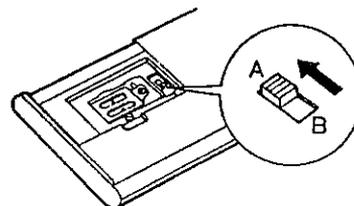
4. Die vier Betriebsbatterien ersetzen. (Achten Sie darauf, immer alle vier Batterien gleichzeitig zu ersetzen.) Die Batterien mit der negativen Seite zuerst einlegen.



5. Die Abdeckung kurzfristig über das Batteriefach schieben. Den Computer herumdrehen und den RESET-Schalter auf der Vorderseite mit einem Kugelschreiber oder einem ähnlichen Gerät eindrücken. Stellen Sie sicher, daß nichts auf dem Display angezeigt wird. Falls eine Anzeige erfolgt, muß der RESET-Schalter noch einmal eingedrückt werden.



6. Die Abdeckung des Batteriefachs wieder abnehmen und den Speicherschutz-Schalter in die Position A schieben.



7. Die Abdeckung des Batteriefachs wieder anbringen und mit der Schraube sichern.

**Hinweis:**

Wenn das Gerät eingeschaltet wird, während sich der Speicherschutz-Schalter in der Position B befindet, kann der Computer nicht bedient werden. In diesem Fall muß die Abdeckung des Batteriefachs abgenommen und der Speicherschutz-Schalter in die Position A geschoben werden.

**Zeitpunkt des Batteriewechsels für die Speichersicherung**

Die Batterie für die Speichersicherung hat bei Raumtemperatur (20°C) eine Lebensdauer von etwa fünf Jahren. Während dieser Zeit bleiben die Speicherinhalte erhalten, auch wenn die Betriebsbatterien schwach sind oder ersetzt werden. Wenn die Betriebsbatterien schwach sind oder entfernt wurden, wird allerdings die Lebensdauer der Batterie für die Speichersicherung erheblich verkürzt.

**Hinweis:**

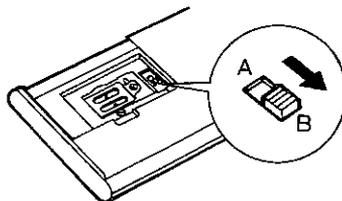
Unter extremen Umweltbedingungen, z.B. bei hohen Temperaturen, verkürzt sich die Lebensdauer der Batterie.

**Ersetzen der Batterie für die Speichersicherung**

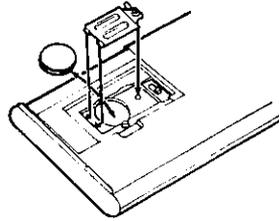
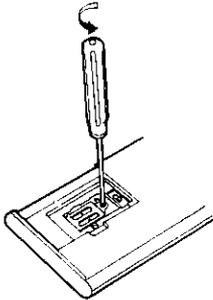
Vor dem Ersetzen der Batterie für die Speichersicherung muß sichergestellt werden, daß die vier Betriebsbatterien nicht zu schwach sind (die Warnanzeige **BATT** darf nicht erscheinen). Wenn sie schwach sind, müssen zuerst diese vier Trockenbatterien ersetzt werden. Wenn die Batterie für die Speichersicherung entnommen werden, während die Betriebsbatterien zu schwach sind, kann Speicherinhalt verloren gehen.

Zum Ersetzen der Batterie folgendermaßen vorgehen:

1. **ON** und dann **OFF** drücken, um den Computer abzuschalten.
2. Mit einer Münze oder einem Flachkopf-Schraubenzieher die Schraube für die Sicherung der Abdeckung des Batteriefachs auf der Rückseite des Gerätes lösen und die Abdeckung entfernen.  
(Beim Abnehmen der Abdeckung darauf achten, daß nicht aus Versehen die Taste **ON** auf der Vorderseite des Gerätes gedrückt wird. Falls die Taste doch gedrückt wird, muß der Computer durch Drücken von **OFF** wieder ausgeschaltet werden. Falls zum folgenden Schritt 3 übergegangen wird, während das Gerät eingeschaltet ist, wird der gesamte Speicherinhalt zerstört oder vollständig gelöscht.)
3. Den Speicherschutz-Schalter auf die Position B schieben.



4. Die Batteriehalterung für die Speichersicherungs-Batterie mit einem kleinen Schraubenzieher aufschrauben und die Halterung entfernen.
5. Die Ersatz-Lithiumbatterie mit einem trockenen Tuch abreiben und mit der positiven Seite nach unten in das Fach einlegen.



Hinweis:  
Darauf achten, daß während diesem Vorgang die Betriebsbatterien vom Typ AA nicht aus dem Batteriefach springen.

6. Die Batteriehalterung wieder im Batteriefach anbringen und mit der Schraube befestigen.
7. Den Speicherschutz-Schalter auf die Position A schieben.
8. Die Abdeckung des Batteriefachs wieder anbringen und mit der Schraube sichern.

#### **Vorsichtsmaßnahmen beim Auswechseln der Batterie**

Falsche Handhabung der Batterien kann zum Auslaufen der Elektrolytlösung oder zur Explosion führen. Die folgenden Maßnahmen unbedingt beachten:

1. Immer alle vier Trockenbatterien gleichzeitig auswechseln.
2. Gebrauchte und neue Batterien nicht gleichzeitig verwenden.
3. Sicherstellen, daß die neuen Batterien von der gleichen Sorte wie die entnommenen sind. Einige Batterien sind aufladbar, während andere nicht aufgeladen werden können. Die Beschreibung auf den Batterien lesen und keine aufladbaren Batterien verwenden.
4. Die Batterien entsprechend der im Batteriefach angegebenen Polarität korrekt einsetzen.

#### **Warnung:**

- **Die Batterien außerhalb der Reichweite von Kindern aufbewahren.**
- Entladene Batterien immer entnehmen. Sie könnten auslaufen und das Gerät beschädigen.
- Batterien nicht in offenes Feuer werfen, da sie explodieren können.

# 3. PERIPHERE GERÄTE

Der Computer kann zusammen mit den folgenden zusätzlichen SHARP Geräten verwendet werden:

MODELL	BESCHREIBUNG	ANSCHLUSS
CE-126P	Drucker/Kassetten-Interface	11-Stift Stecker
CE-T801	Datenübertragungskabel	11-Stift Stecker

Eine kurze Beschreibung eines jeden Gerätes folgt weiter unten. Zur genaueren Information lesen Sie bitte die entsprechenden Bedienungsanleitungen. Das Datenübertragungskabel CE-T801 kann zusätzlich verwendet werden. Es ermöglicht dem PC-E220 die Datenübertragung von und zu einem Personal Computer. (Siehe Anhang A.)

Die folgende Tabelle gibt eine Übersicht über die E/A-Befehle, die für die verschiedenen Peripheriegeräte verwendet werden.

	RAM-Diskette	Kassette	SIO	
SAVE LOAD	○	×	×	}
CSAVE CLOAD	×	○	×	
PRINT # INPUT #	×	○	○	}

○: Kann verwendet werden.  
×: Kann nicht verwendet werden.

In der TEXT-Betriebsart können TEXT-Programme im ASCII-Format auf RAM-Diskette, Kassette oder über den seriellen Port (SIO) übertragen werden.

## CE-126P Thermodrucker/Kassetten-Interface

Der Thermodrucker CE-126P mit integriertem Kassetteninterface gestattet den Anschluß eines Kassettenrekorders.

Merkmale des CE-126P:

- 24-Zeichen Thermodrucker.
- Bequemer Papiervorschub und Abreißschiene.
- Auf Wunsch Protokolldruck von Berechnungen.
- Leichte Kontrolle des durch BASIC-Programme Ausgedruckten.
- Eingebaute Kassetten-Interface-Fernbedienungsfunktion.
- Manuelle und Programmkontrolle des Rekorders zur Eingabe von Programmen und Daten.
- Zum leichten Transportieren batteriebetrieben.

Zum Anschließen des Computers an den CE-126P lesen Sie die Bedienungsanleitung, die dem CE-126P beiliegt.

## Kassetten-Interface

Dieses Gerät ist erforderlich, wenn der Computer an einen Kassettenrekorder angeschlossen werden soll, um auf der Kassette Programme zu sichern oder zu laden. Mit dem 11-Stift Stecker wird es an den Computer und an die Mikrophon (MIC) und Kopfhörer (EAR) Eingangsbuchsen des Rekorders angeschlossen. Benutzen Sie einen Kassettenrekorder mit den folgenden technischen Daten:

Gegenstand	Anforderungen
1. Rekordertyp	Beliebiger Standardkassetten- oder Mikro-Kassettenrekorder
2. Eingangsbuchse	"MIC" Mini-Eingangsbuchse (nie die "AUX"-Buchse)
3. Eingangsimpedanz	Niedrige Eingangsimpedanz (200 – 1000 Ohm)
4. Minimaler Eingangspegel	Unter 3 mV oder -50 dB
5. Ausgangsbuchse	EXTErn, MONITOR oder EARphone Mini-Ausgangsbuchse oder entsprechendes
6. Ausgangsimpedanz	Unter 10 Ohm
7. Ausgangspegel	Über 1 V (tatsächlicher Maximalausgang über 100 mW)
8. Verzerrung	Innerhalb 15% (Bereich: 2 kHz bis 4 kHz)
9. Gleichlaufschwankungen	0,3% maximal (WRMS)
10. Anderes	Rekordermotor mit stabiler Geschwindigkeit

## Verwendung des Kassetten-Interfaces

### Aufnahme (Sichern) auf Magnetband

Siehe Kassetten-Hinweise auf Seite 19.

Zum Sichern eines Programms vom Computer auf die Kassette wird folgendermaßen vorgegangen.

1. Stellen Sie den REMOTE-Schalter des CE-126P auf OFF.
2. Geben Sie ein Programm oder Daten in den Computer ein.
3. Legen Sie eine Kassette in den Rekorder. Fahren Sie das Band auf die Position, wo das Programm aufgenommen werden soll, und vermeiden Sie dabei sorgfältig das Vorlaufband (durchsichtiges Band ohne magnetische Beschichtung) und bereits vorher aufgenommene Programme.
4. Schließen Sie den roten Stecker des Interfaces an die MIC-Buchse und den schwarzen Stecker an die REM-Buchse bei jedem Gerät an.
5. Stellen Sie den REMOTE-Schalter auf ON.
6. Drücken Sie gleichzeitig die RECORD- und PLAY-Tasten des Kassettenrekorders.
7. Geben Sie die Aufnahmebefehle ein (CSAVE-Kommando) und drücken Sie zum Start der Ausführung die  -Taste.  
Bringen Sie den Computer in den RUN- oder PRO-Modus. Danach geben Sie folgendes ein:

CSAVE "Dateiname" 

Zum Beispiel: CSAVE "AA" 

Das Band beginnt sich nach dem Druck auf die  -Taste zu drehen und läßt eine Leerstelle von einigen Sekunden. Der Dateiname und der Inhalt der Datei werden dann aufgenommen.

Zur Ausgabe von Daten werden Anweisungen in die Programme eingefügt, siehe folgendes Beispiel:

```
100 OPEN "CAS:DATA" FOR OUTPUT
110 PRINT #1, AB, C$, D(5)
120 CLOSE #1
```

8. Ist die Aufnahme beendet, erscheint im Display das Bereitschaftssymbol (>) und der Kassettenrekorder geht automatisch in Stoppstellung. Jetzt haben Sie das Programm auf Kassette (es verbleibt auch im Computer).

Mit dem Bandzählwerk des Rekorders das Programm auf dem Band aufsuchen.

### **Überprüfen eines gesicherten Programms**

Nach dem Laden oder Übertragen eines Programms auf oder von einem Band können Sie überprüfen, ob das Programm auf Band und das Programm im Computer identisch sind (und so sichergehen, daß alles in Ordnung ist, bevor Sie mit dem Programmieren fortfahren).

1. Stellen Sie den REMOTE-Schalter auf OFF.
2. Bringen Sie das Band in die Position unmittelbar vor der Datei, die überprüft werden soll.
3. Schließen Sie bei jedem Gerät den grauen Stecker an die Buchse EAR und den schwarzen Stecker an die Buchse REM bei jedem Gerät an.
4. Stellen Sie den REMOTE-Schalter auf ON.
5. Drücken Sie die Taste PLAY.
6. Geben Sie ein CLOAD?-Kommando ein und starten Sie mit der  -Taste die Ausführung. Machen Sie dies folgendermaßen: Bringen Sie den Computer in den RUN- oder PRO-Modus. Tippen Sie folgendes ein:

Der Dateiname, den Sie vorher eingegeben hatten.

CLOAD? "AA" 

Der Computer sucht jetzt automatisch nach dem bezeichneten Dateinamen und vergleicht die Inhalte auf Band mit den Inhalten im Speicher.

Ist der bezeichnete Dateiname auf Band gefunden, so wird an die auf dem Bildschirm getippte Zeile automatisch ein Stern (\*) angehängt. Hierdurch wird angezeigt, daß der Überprüfungsvorgang begonnen hat.

Das Bereitschaftssymbol (>) erscheint auf dem Display, wenn die Programme identisch sind.

Unterscheiden sich die Programme, so wird die Ausführung unterbrochen und auf dem Display wird eine Fehlermeldung angezeigt. Kommt das vor, versuchen Sie es erneut.

### **Laden von einem Magnetband**

Siehe die Kassetten-Hinweise auf Seite 19.

Zum Laden, Überführen oder Lesen von Programmen und Daten von einem Magnetband gehen Sie folgendermaßen vor:

1. Stellen Sie den REMOTE-Schalter auf OFF.
2. Legen Sie eine Kassette in das Tonbandgerät ein. Bringen Sie das Band unmittelbar vor die Stelle, die herausgelesen werden soll.

3. Schließen Sie bei jedem Gerät den grauen Stecker an die EAR-Buchse und den schwarzen Stecker an die REM-Buchse bei jedem Gerät an (hat das Gerät keine REM-Buchse, so kontrollieren Sie den Bandlauf manuell mit der PAUSE-Taste).
4. Stellen Sie den REMOTE-Schalter auf ON.
5. Drücken Sie die PLAY-Taste des Tonbandgerätes (Wiedergabestellung).  
Stellen Sie die LAUTSTÄRKE zwischen Mitte und Maximum ein.  
Stellen Sie den TON auf höchste Höhen ein.
6. Geben Sie die Übertragungsanweisungen ein (CLOAD-Kommando) und drücken Sie die  -Taste.  
Bringen Sie den Computer in den RUN- oder PRO-Modus. Dann geben Sie folgendes ein:

CLOAD "Dateiname" 

Zum Beispiel: CLOAD "AA" 

Der bestimmte Dateiname wird jetzt automatisch gesucht und seine Inhalte werden in den Computer übertragen.

Ist die bestimmte Datei auf dem Band gefunden, beginnt das Laden. Dies wird durch einen Stern (\*) angezeigt, der automatisch an die auf dem Bildschirm getippte Zeile angehängt wird.

7. Wurde das Programm überführt, stoppt der Computer automatisch das Band und das Bereitschaftssymbol (>) erscheint im Display.  
Wenn ein Fehler auftritt, so versuchen Sie erneut die Ladung von Anfang an durchzuführen. Tritt wieder ein Fehler auf, wiederholen Sie den Vorgang, nachdem Sie die Lautstärke ein wenig höher oder niedriger eingestellt haben. Erfolgt keine Fehlermeldung, das Band stoppt aber nicht, ist etwas falsch gelaufen. Drücken Sie die **BREAK** -Taste zum Stoppen des Bandes und versuchen Sie es erneut.

### Hinweise für die Kassetten

1. Verwenden Sie zum Überprüfen und Laden immer das gleiche Bandgerät, mit dem Sie das Programm gesichert haben. Die Verwendung eines anderen Gerätes kann Fehler verursachen.
2. Verwenden Sie gute Fe-Kassetten; CrO<sub>2</sub>-Kassetten sind weniger geeignet.
3. Bei Verwendung eines Netzadapters für das Interface CE-126P kann gelegentlich ein Brummen die Aufnahme begleiten. In diesem Fall schalten Sie auf Batteriebetrieb um.
4. Wird zur Aufnahme von Programmen ein bereits benutztes Band wiederverwendet, dann löschen Sie vor der Aufnahme alle alten Programme auf dem Band.

## Verwendung des Drucker/Kassetten-Interfaces CE-126P

### Benutzung des Druckers

Benutzen Sie den Computer zur direkten Rechnungseingabe (nur im RUN-Modus), so können Sie den CE-126P zum gleichzeitigen Ausdrucken Ihrer Berechnungen verwenden. Drücken Sie während des RUN-Modus **SHIFT** und dann die  -Taste (P↔NP).

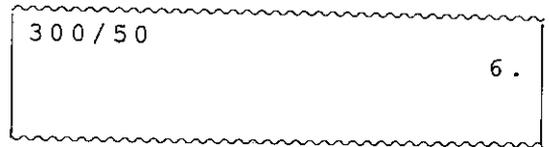
Am Ende einer Rechnung drücken Sie die  -Taste. Hierdurch werden die Inhalte einer Zeile des Displays ausgedruckt und die Ergebnisse auf der nächsten Zeile.

Zum Beispiel:

Eingabe

300/50 

Papier



3 0 0 / 5 0  
6 .

Mit der LPRINT-Anweisung kann der Drucker aus BASIC-Programmen heraus ausdrucken (zur weiteren Information lesen Sie das BASIC KOMMANDO LEXIKON). Benutzen Sie LPRINT in der gleichen Weise wie die PRINT-Anweisung.

Mit dem LLIST-Kommando (zur weiteren Information lesen Sie das BASIC KOMMANDO LEXIKON) können Sie Ihre Programme mit dem Drucker auflisten. Ohne Angabe einer Zeilennummer werden durch den LLIST-Befehl alle im Speicher befindlichen Programmzeilen in Reihenfolge Zeile für Zeile aufgelistet. Es kann mit dem LLIST-Kommando auch ein Zeilenbereich angegeben werden, um die Anzahl der auszudruckenden Zeilen zu beschränken. Wenn Programmzeilen mehr als 24 Zeichen enthalten, werden sie mehrzeilig gedruckt. Die zweite und die folgenden Zeilen werden eingerückt, so daß die Zeilennummer jeder separaten Programmzeile klar identifiziert ist.

#### **Hinweise:**

- Ist der Drucker von außen starken elektrischen Störungen ausgesetzt, so kann es vorkommen, daß er bedeutungslose Ziffern ausdruckt. Wenn dies passiert, drücken Sie zum Stoppen des Ausdrucks die  -Taste. Schalten Sie den CE-126P aus (OFF) und wieder ein (ON) und drücken dann die  -Taste. Das Drücken dieser Taste bringt den Drucker auf Normalbetrieb zurück.
- Wird der CE-126P nicht benutzt, so schalten Sie ihn aus (OFF), um die Lebensdauer der Batterien zu verlängern.

# DIREKTEINGABE- BETRIEB

Der PC-E220\* erlaubt Ihnen im CAL- (Rechner), ENG (Technikersoftware), STAT (Statistik) und RUN-Modus Direkteingabe-Rechnungen (im Gegensatz zu Rechnungen als Teil eines Programmes im PRO-Modus). KAPITEL 2 dieses Handbuches beschreibt den Gebrauch dieser Betriebsarten.

Da zwischen den Betriebsarten einige Funktionsüberschneidungen vorkommen, beginnt KAPITEL 2 mit einer Übersicht über die Ansteuerung von Bedienung und Modus des PC-E220. Die Betriebsart PRO wird detailliert im KAPITEL 3 behandelt.

\* Der PC-E220 wird im folgenden als "der Computer" bezeichnet.

# 4. CAL-MODUS

Der Computer kann wie ein Taschenrechner mit 10 Stellen verwendet werden. Dazu muß der Computer auf die CAL-Betriebsart eingestellt werden. Die Taste **CAL** drücken.

Die Anzeige "<CAL>" erscheint in der oberen linken Ecke des Displays.

**Hinweis:**

In der CAL-Betriebsart können die Ergebnisse nicht auf den Drucker ausgegeben werden.

## Berechnungen

Wir wollen jetzt einige einfache Berechnungen durchführen. Drücken Sie die folgenden Tasten und beachten Sie die Anzeige:

<u>Eingabe</u>	<u>Anzeige</u>
123	123.
<b>+</b>	123.
654	654.
<b>=</b>	777.

Haben Sie das richtige Ergebnis erhalten? Falls nicht, drücken Sie die **C•CE** -Taste und versuchen die gleiche Berechnung erneut.

Als nächstes soll der Wert von pi ( $\pi$ ) abgerufen werden.

Das Symbol " $\pi$ " befindet sich in braun oberhalb der Taste **n!**. Die Funktionen, die durch braune Schrift gekennzeichnet sind, können eingesetzt werden, indem zuerst die Taste **2nd F** und dann die entsprechende Funktionstaste gedrückt wird.

Drücken Sie **2nd F** **n!**.

<u>Eingabe</u>	<u>Anzeige</u>
<b>2nd F</b> <b><math>\pi</math></b>	3.141592654

Auf der Anzeige erscheint jetzt der Wert von  $\pi$ .

Nun soll  $10^4$  berechnet werden. Für diese Berechnung wird die Funktion  $10^x$  verwendet, die auch in braun geschrieben ist. Daher muß die **2nd F** -Taste gedrückt werden.

<u>Eingabe</u>	<u>Anzeige</u>
4 <b>2nd F</b> <b><math>10^x</math></b>	10000. ( $10^4 = 10000$ )

Folgend werden die Funktionen der wichtigsten Tasten umrissen:

\* **CE** (Löschen/Löschen der Eingaben) (rote Taste)

Wird diese Taste unmittelbar nach Eingabe von numerischen Daten oder Abrufen des Speicherinhaltes gedrückt, werden diese Daten gelöscht. In jedem anderen Fall bewirkt die Betätigung der Taste **CE** Löschung des Operators und/oder der numerischer Daten, die eingegeben wurden. Der Speicherinhalt wird durch Betätigung der Taste **CE** nicht gelöscht.

<u>Eingabe</u>	<u>Anzeige</u>
123 <b>+</b> 456	456.
<b>CE</b>	0.
789 <b>=</b>	912. (123 + 789 = 912)
6 <b>X*</b> 2 <b>+</b>	12.
<b>CE</b>	0.
6 <b>÷/</b> 2 <b>+</b>	3.
5 <b>=</b>	8.

Die Taste **CE** kann außerdem zur Löschung eines Fehlers verwendet werden.

<u>Eingabe</u>	<u>Anzeige</u>
5 <b>÷/</b> 0 <b>=</b>	Fehlersymbol E
	<div style="display: flex; justify-content: space-between;"> <span>&lt;CAL&gt;</span> <span>0 .</span> </div>
<b>CE</b>	<div style="display: flex; justify-content: space-between;"> <span>&lt;CAL&gt;</span> <span>0 .</span> </div>

\* **FSE** (Anzeigemodus-Schalter)

Diese Taste dient zum Umschalten des Anzeigemodus für das Ergebnis von Berechnungen vom Gleitpunktsystem (normaler Modus) zu Festpunktsystem (FIX), wissenschaftliche Notation (SCI), technische Notation (ENG) oder umgekehrt.

Eingabe

23 **X\*** 1000 **=**

Anzeige

<CAL>  
23000.

(Normal)

**FSE**

<CAL>                   FIX  
23000.000

(FIX)

**FSE**

<CAL>                   SCI  
2.300E 04

(SCI)

**FSE**

<CAL>                   ENG  
23.000E 03

(ENG)

\* **DIGIT** (spezifiziert die Anzahl der Dezimalstellen)

Diese Taste dient, in Verbindung mit einer Zahlentaste, zur Spezifizierung der Anzahl der Dezimalstellen (Nachkommastellen). Schalten Sie den Computer aus und dann wieder ein. Drücken Sie die Taste **FSE**, dann erscheint "0.000" (FIX-Modus) auf der Anzeige.

**Beispiel:**

1. Spezifizieren von 2 Dezimalstellen.

Eingabe

**2nd F** **DIGIT** **2**

Anzeige

<CAL>                   FIX  
0.00

5 **÷/** 8 **=**

<CAL>                   FIX  
0.63

2. Spezifizieren von 5 Dezimalstellen.

**2nd F** **DIGIT** **5**

<CAL>                   FIX  
0.62500

\* **DRG** (spezifiziert die Winkeleinheit)

Diese Taste dient zur Spezifizierung der Winkeleinheit für numerische Daten für trigonometrische Funktionen, inverse trigonometrische Funktionen und Koordinatenumwandlung.

Eingabe

Anzeige

**2nd F** **DRG**

DEG

(Altgrad)

**2nd F** **DRG**

RAD

(Bogenmaß)

**2nd F** **DRG**

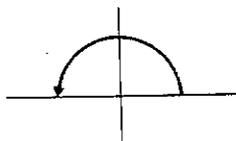
GRAD

(Neugrad)

DEG

(Altgrad)

$$180^\circ = \pi \text{ (rad)} = 200^g$$



DEG: Altgrad [°]  
 RAD: Bogenmaß [rad]  
 GRAD: Neugrad [g]

\* **0** bis **9**, **.**, **Exp** und **+/-**

**Exp**: Dient zur Eingabe von Zahlen in Exponentialform (auf der Anzeige erscheint "E" nach der eingegebenen Zahl).

Eingabe

Anzeige

**C•CE** 4 **2nd F** **Exp** 3

4.E 03  
 ( $4 \times 10^3$ )

**=**

4000.

**+/-**

-4000.

**+/-**: Dient zur Eingabe von negativen Zahlen (oder zur Umkehrung des Vorzeichens von negativ zu positiv).

<u>Eingabe</u>	<u>Anzeige</u>
1.23 <b>+/-</b>	-1.23
<b>2nd F</b> <b>Exp</b> 5 <b>+/-</b>	-1.23E-05 (-1.23 × 10 <sup>-5</sup> )
<b>=</b>	-0.0000123
<b>+/-</b>	0.0000123

## Grundlegende Bedienung

### 1. Addition und Subtraktion

Eingabe: 12 **+** 45.6 **-** 32.1 **+** 789 **-** 741 **+** 213 **=**  
 Ergebnis: 286.5

### 2. Multiplikation und Division

Eingabe: 841 **X\*** 586 **÷/** 12 **=**  
 Ergebnis: 41068.83333

Eingabe: 427 **+** 54 **X\*** 32 **÷/** 7 **-** 39 **X\*** 2 **=**  
 Ergebnis: 595.8571429

#### Hinweis:

Multiplikation und Division haben Vorrang vor Addition und Subtraktion. Der Computer führt intern zuerst die Multiplikation und Division durch.

Multiplikationen mit einer Konstanten:

Der zuerst eingegebene Wert ist eine Konstante.

Eingabe: 3 **X\*** 5 **=** Ergebnis: 15

Eingabe: 10 **=** Ergebnis: 30

Divisionen mit einer Konstanten:

Der nach dem Divisionszeichen eingegebene Wert ist eine Konstante.

Eingabe: 15 **÷/** 3 **=** Ergebnis: 5

Eingabe: 30 **=** Ergebnis: 10

#### Hinweis:

Je nach der Vorrangordnung stellt der Computer einige Berechnungen in den schwebenden Zustand. Bei Kettenrechnungen bleiben die entsprechend der Vorrangordnung die letzte Recheninstruktion und der jeweilige numerische Wert erhalten und können damit für weitere Berechnungen bzw. als Konstanten verwendet werden.

$a + b \times c =$	$+ bc$	(Konstanten-Addition)
$a + \underline{b \div c} =$	$\div c$	(Konstanten-Division)
$a \div \underline{b \times c} =$	$\frac{a}{b} \times$	(Konstanten-Multiplikation)
$a \times \underline{b - c} =$	$- c$	(Konstanten-Subtraktion)

### 3. Speicherberechnungen

Der unabhängige Speicher kann mit den Tasten  $\boxed{X\rightarrow M}$ ,  $\boxed{RM}$ ,  $\boxed{M+}$  oder  $\boxed{M-}$  angewählt werden. Vor dem Beginn einer Berechnung wird der Speicher durch Drücken von  $\boxed{C\rightarrow CE}$  und  $\boxed{X\rightarrow M}$  gelöscht.

Wenn ein anderer Wert als Null gespeichert ist, erscheint "M" auf dem Display.

Eingabe: 12  $\boxed{+}$  5  $\boxed{M+}$

Ergebnis: 17

Zur Subtraktion wird eingegeben: 2  $\boxed{+}$  5  $\boxed{2nd F}$   $\boxed{M-}$

Ergebnis: 7

Eingabe von  $\boxed{RM}$  zum Abruf des Speichers: 10 wird angezeigt.

Eingabe: 12  $\boxed{X*}$  2  $\boxed{=}$   $\boxed{X\rightarrow M}$

Ergebnis: 24 (ersetzt 10 im Speicher)

Eingabe: 8  $\boxed{\div/}$  2  $\boxed{M+}$

Ergebnis: 4  $\boxed{RM}$ : 28

Im CAL-Modus stehen 26 Speicher zur Verfügung, die mit  $\boxed{STO}$   $\boxed{A}$  bis  $\boxed{STO}$   $\boxed{Z}$  spezifiziert werden, außerdem der Speicher, der mit der Taste  $\boxed{X\rightarrow M}$  spezifiziert wird.

Durch Drücken der Tasten  $\boxed{STO}$   $\boxed{A}$  werden der numerischen BASIC-Variablen A Daten zugewiesen.

Zum Lesen dieser Daten müssen die Tasten  $\boxed{RCL}$   $\boxed{A}$  gedrückt werden. Das Drücken der Tasten  $\boxed{A}$   $\boxed{\leftarrow}$  im RUN-Modus ist identisch zum Drücken der der Tasten  $\boxed{RCL}$   $\boxed{A}$  im CAL-Modus.

## Wissenschaftliche Berechnungen

Zur Berechnung von trigonometrischen oder invers trigonometrischen Gleichungen und zur Koordinatenumwandlung müssen die Winkeleinheiten für die Berechnung zugewiesen werden. Die Zuweisung der Winkeleinheiten DEG, RAD oder GRAD erfolgt durch Drücken der Tasten  $\boxed{2nd F}$   $\boxed{DRG}$ .

### Hinweis:

Der Abschnitt über Fehler im Anhang B befaßt sich mit der Rechnerkapazität dieses Computers.

#### 1. Trigonometrische Funktionen

Winkeleinheit auf DEG

Aufgabe:  $\sin 30^\circ + \cos 40^\circ =$

Eingabe: 30  $\boxed{\sin}$  + 40  $\boxed{\cos}$   $\boxed{=}$

Ergebnis: 1.266044443

Aufgabe:  $\cos 0,25\pi$

Winkeleinheit auf RAD

Eingabe: 0.25  $\boxed{X*}$   $\boxed{2nd F}$   $\boxed{\pi}$   $\boxed{=}$   $\boxed{\cos}$

Ergebnis: 0.707106781

## 2. Inverse trigonometrische Funktionen (Arcusfunktionen)

Aufgabe:  $\sin^{-1} 0,5$

Winkeleinheit auf DEG

Eingabe: 0.5  $\boxed{2nd F}$   $\boxed{\sin^{-1}}$

Ergebnis: 30

Aufgabe:  $\cos^{-1} -1$

Winkeleinheit auf RAD

Eingabe: 1  $\boxed{+/-}$   $\boxed{2nd F}$   $\boxed{\cos^{-1}}$

Zur Eingabe einer negativen Zahl die  $\boxed{+/-}$  -Taste nach Eingabe der Zahl drücken.

Ergebnis: 3.141592654 (Wert  $\pi$ )

Die Ergebnisse von inversen trigonometrischen Funktionen können nur in folgenden Bereichen liegen:

$\theta = \sin^{-1} x, \theta = \tan^{-1} x$

DEG:  $-90 \leq \theta \leq 90$  [°]

RAD:  $-\pi/2 \leq \theta \leq \pi/2$  [rad]

GRAD:  $-100 \leq \theta \leq 100$  [g]

$\theta = \cos^{-1} x$

DEG:  $0 \leq \theta \leq 180$  [°]

RAD:  $0 \leq \theta \leq \pi$  [rad]

GRAD:  $0 \leq \theta \leq 200$  [g]

## 3. Hyperbel- und inverse Hyperbelfunktionen (Areafunktion)

Aufgabe:  $\sinh 4$

Eingabe: 4  $\boxed{hyp}$   $\boxed{\sin}$

Ergebnis: 27.2899172

Aufgabe:  $\sinh^{-1} 9$

Eingabe: 9  $\boxed{2nd F}$   $\boxed{archyp}$   $\boxed{\sin}$

Ergebnis: 2.893443986

## 4. Potenzfunktionen

Aufgabe:  $20^2$

Eingabe: 20  $\boxed{x^2}$

Ergebnis: 400

Aufgabe:  $3^3$  und  $3^4$

Eingabe: 3  $\boxed{y^x}$  3  $\boxed{=}$

Ergebnis: 27

Eingabe: 3  $\boxed{y^x}$  4  $\boxed{=}$

Ergebnis: 81

## 5. Wurzelfunktionen

Aufgabe:  $\sqrt{25}$

Eingabe: 25  $\boxed{\sqrt{}}$

Ergebnis: 5

Aufgabe: Dritte Wurzel von 27

Eingabe: 27  $\boxed{2nd F}$   $\boxed{\sqrt[3]{}}$

Ergebnis: 3

Aufgabe: Vierte Wurzel von 81

Eingabe: 81  $\boxed{2nd F}$   $\boxed{x\sqrt{y}}$  4  $\boxed{=}$

Ergebnis: 3

## 6. Logarithmische Funktionen

Aufgabe:  $\ln 21, \log 173$

Natürlicher Logarithmus

Eingabe: 21  $\boxed{\ln}$

Ergebnis: 3.044522438

Dekadischer Logarithmus

Eingabe: 173  $\boxed{\log}$

Ergebnis: 2.238046103

## 7. Exponentialfunktionen

Aufgabe:  $e^{3,0445}$

Eingabe: 3.0445  $\boxed{2nd F}$   $\boxed{e^x}$

Ergebnis: 20.99952881 (siehe  $\ln 21$ )

Aufgabe:  $10^{2,238}$

Eingabe: 2.238  $\boxed{2nd F}$   $\boxed{10^x}$

Ergebnis: 172.9816359 (siehe  $\log 173$ )

## 8. Reziprok-Rechnung

Aufgabe:  $1/6 + 1/7$

Eingabe: 6  $\boxed{1/x}$   $\boxed{+}$  7  $\boxed{1/x}$   $\boxed{=}$  Ergebnis: 0.309523809

## 9. Fakultät

Aufgabe: 69!

Eingabe: 69  $\boxed{n!}$

Ergebnis: 1.711224524E 98 ( $= 1,711224524 \times 10^{98}$ )

Aufgabe:  ${}_8P_3 = \frac{8!}{(8-3)!}$

Eingabe: 8  $\boxed{n!}$   $\boxed{\div}$  (  $\boxed{8}$   $\boxed{-}$  3  $\boxed{)}$   $\boxed{n!}$   $\boxed{=}$

Ergebnis: 336

## 10. Prozentrechnung

Aufgabe: 45% von 2.780 ( $2.780 \times \frac{45}{100}$ )

Eingabe: 2780  $\boxed{\times}$  45  $\boxed{2nd F}$   $\boxed{\Delta\%}$   $\boxed{=}$

Ergebnis: 1251

Aufgabe:  $200 - 200 \times \frac{30}{100}$

Eingabe: 200  $\boxed{-}$  30  $\boxed{2nd F}$   $\boxed{\Delta\%}$   $\boxed{=}$

Ergebnis: 140

Aufgabe:  $200 + 200 \times \frac{30}{100}$

Eingabe: 200  $\boxed{+}$  30  $\boxed{2nd F}$   $\boxed{\Delta\%}$   $\boxed{=}$

Ergebnis: 260

## 11. Umwandlung des Winkels und der Zeit

Zur Umwandlung von Winkel- oder Zeitangaben ( $^\circ$ ,  $'$ ,  $''$ , bzw. Std., Min., Sek.) in den entsprechenden Dezimalwert muß der Gradwert als ganze Zahl und die Werte in Minuten und Sekunden als Nachkommastellen eingegeben werden.

Aufgabe:

Umwandlung von  $12^\circ 47' 52''$  in den entsprechenden Dezimalwert

Eingabe: 12.4752  $\boxed{\rightarrow DEG}$

Ergebnis: 12.79777778

Bei der Umwandlung des dezimalen Winkel-/Zeitwerts in den sexagesimalen Wert ( $^\circ$ ,  $'$ ,  $''$  bzw. Std., Min., Sek.) ist das angezeigte Ergebnis wie folgt zuzuordnen:

Ganze Zahl = Grad; erste und zweite Dezimalstelle = Minuten; dritte und vierte Dezimalstelle = Sekunden; fünfte und folgende Dezimalstellen = Teile von Sekunden.

Aufgabe:

Umwandlung des dezimalen Winkelwerts 24,7256 in den entsprechenden Sexagesimalwert (Grad, Minute, Sekunden)

Eingabe: 24.7256  $\boxed{2nd F}$   $\boxed{\rightarrow DMS}$

Ergebnis: 24.433216 oder  $24^\circ 43' 32''$

Ein Pferd läuft Rundenzeiten von 2 Min. 25 Sek., 2 Min. 38 Sek. und 2 Min. 22 Sek. Wie groß ist die durchschnittliche Rundenzeit?

Eingabe:

0.0225  $\rightarrow$ DEG  $\square$  +  $\square$  0.0238  $\rightarrow$ DEG  $\square$  +  $\square$  0.0222  $\rightarrow$ DEG  $\square$  =  $\square$

Ergebnis 1: 0.123611111

Eingabe:  $\square$   $\div$   $\square$  3  $\square$  =  $\square$

Ergebnis 2: 0.041203703

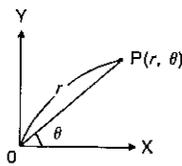
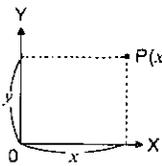
Eingabe:  $\square$  2nd F  $\square$   $\rightarrow$ DMS  $\square$

Ergebnis 3: 0.022833333

Die durchschnittliche Rundenzeit ist 2 Minuten 28 Sekunden.

## 12. Koordinatenumwandlung

Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten ( $x, y \rightarrow r, \theta$ )



$$r = \sqrt{x^2 + y^2} \quad \text{DEG: } 0 \leq |\theta| \leq 180$$

$$\text{RAD: } 0 \leq |\theta| \leq \pi$$

$$\theta = \tan^{-1} \frac{y}{x} \quad \text{GRAD: } 0 \leq |\theta| \leq 200$$

Umwandlung der rechtwinkligen Koordinaten  $x = 6$  und  $y = 4$  in Polarkoordinaten.

Winkeleinheit: DEG

Eingabe: 6  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ ) 4  $\square$  2nd F  $\square$   $\rightarrow$ r $\theta$   $\square$  Ergebnis: 7.211102551 (r)

Eingabe:  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ )  $\square$  Ergebnis: 33.69006753 ( $\theta$ )

Berechnung der Höhe und Richtung (Phase) des Vektors  $i = 12 + j9$

Eingabe: 12  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ ) 9  $\square$  2nd F  $\square$   $\rightarrow$ r $\theta$   $\square$  Ergebnis: 15 (r)

Eingabe:  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ )  $\square$  Ergebnis: 36.86989765 ( $\theta$ )

Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten ( $r, \theta \rightarrow x, y$ )

Berechnung von  $P(14, \pi/3)$ ,  $r = 14$ ,  $\theta = \pi/3$

Winkeleinheit: RAD

Eingabe:  $\square$  2nd F  $\square$   $\pi$   $\square$   $\div$   $\square$  3  $\square$  =  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ ) 14  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ )  $\square$  2nd F  $\square$   $\rightarrow$ x $y$

Ergebnis: 7.000000002 (x)

Eingabe:  $\square$   $\rightarrow$  (oder  $\square$   $\leftarrow$ )  $\square$

Ergebnis: 12.12435565 (y)

Es wird zuerst  $\theta = \pi/3$  errechnet.  
Nach der Eingabe von r wird es durch  $r = 14$  ersetzt, wenn die Taste  $\square$  (oder  $\square$ ) gedrückt wird.

## Verwendung der Klammern-Tasten

Die Verwendung der Klammern-Tasten  $\square$  (  $\square$  und  $\square$  ) ist dann zwingend notwendig, wenn eine Serie von Einzelberechnungen zu einem Rechenvorgang zusammengefaßt werden soll und dabei die Vorrangordnung der Operatoren außer Kraft gesetzt werden muß. Die Berechnungen in der Klammer haben Priorität vor allen anderen Berechnungen.

Sobald die  $\square$  (  $\square$  -Taste gedrückt wird, erscheint "( )" im Display.

Die Klammerfunktion kann im CAL-Modus bis zu 15 Mal verwendet werden. Als erstes werden die Berechnungen der innersten Klammer ausgeführt.

Aufgabe:  $12 + 42 \div (8 - 6)$

Eingabe: 12  $\square$  +  $\square$  42  $\square$   $\div$   $\square$  (  $\square$  8  $\square$  -  $\square$  6  $\square$  )  $\square$  =  $\square$

Ergebnis: 33

Aufgabe:  $126 + \{(3 + 4) \times (3 - 1)\}$

Eingabe:

126  $\div$   $\left( \left( 3 + 4 \right) \times \left( 3 - 1 \right) \right) =$

Ergebnis: 9

### Hinweis:

Das Schließen der beiden Klammern unmittelbar vor der [=] -Taste (oder [M+] -Taste) kann unterbleiben.

## Nachkommastellen

Die Anzahl der Nachkommastellen kann festgelegt werden; dazu dienen die [2nd F] - und [DIGIT] -Taste in Verbindung mit den Tasten [0] bis [9]. In diesem Fall muß die Anzeige auf Festkomma (FIX), wissenschaftliche Notationen (SCI) oder technische Notationen (ENG) eingestellt werden.

- [2nd F] [DIGIT] [0] → Keine Nachkommastellen.  
(Die Zahl wird auf die nächste ganze Zahl gerundet.)
- [2nd F] [DIGIT] [1] → Eine Nachkommastelle.  
(Die Zahl wird auf die 1. Nachkommastelle gerundet.)
- [2nd F] [DIGIT] [9] → Neun Nachkommastellen.  
(Die Zahl wird auf die 9. Nachkommastelle gerundet.)
- [2nd F] [DIGIT] [.] → Löschen der Zuweisung für Dezimalstellen.

Die Zuweisung für Dezimalstellen wird auch gelöscht, wenn der Computer aus- und wieder eingeschaltet wird oder wenn eine andere Betriebsart gewählt wird. Dann erscheint wieder die normale Anzeige.

### Beispiel:

[2nd F] [DIGIT] [9]  
0.5  $\div$  9 =

→ 0.055555556 (FIX-Betriebsart)  
(Die Zahl wird auf die 9. Nachkommastelle gerundet.)

[FSE]

→ 5.555555556E-02 (SCI-Betriebsart)  
(Die Mantisse wird auf die 9. Dezimalstelle gerundet.)

[2nd F] [DIGIT] [3]

→ 5.556E-02 (SCI-Betriebsart)  
(Die Mantisse wird auf die 4. Dezimalstelle gerundet.)

[FSE]

→ 55.556E-03 (ENG-Betriebsart)

[FSE]

→ 0.055555555

Der Wert ist im Rechner in der Form von  $5.5555555555 \times 10^{-2}$  gespeichert.

Die Rundung in der 11. Nachkommastelle der Mantisse ergibt  $5.555555556 \times 10^{-2}$ .

Da das Display 10 Stellen anzeigt, ist die Rundung in der Ergebnisanzeige nicht enthalten.

Diese Funktion kann für statistische oder Regressions-Berechnungen nicht verwendet werden.

## Modifizierungs-Funktion

Mit der Funktion für Dezimalstellen zeigt der Computer nur die festgelegte Anzahl der Dezimalstellen an, intern speichert er aber insgesamt immer 12 Stellen. Daher können sich die angezeigten Daten von den internen Daten unterscheiden. Um die internen mit den angezeigten Daten in Übereinstimmung zu bringen, wird die Modifizierungs-Funktion verwendet.

### Beispiel:

2nd F DIGIT 2

C•CE 5 ÷/ 9 =

Anzeige: 0.56 (FIX)

X\* 9 =

Anzeige: 5.00 (FIX)

C•CE 5 ÷/ 9 = MDF

Anzeige: 0.56 (FIX)

X\* 9 =

Anzeige: 5.04 (FIX)

## Vorrangordnung

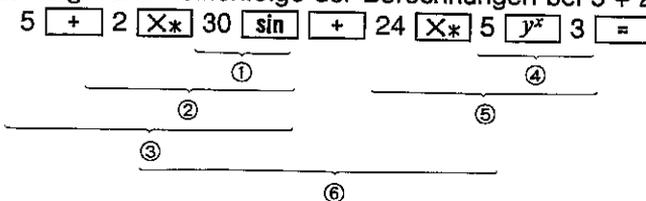
Der Computer ist mit einer Funktion ausgestattet, welche die Vorrangordnung von einzelnen Berechnungen beurteilt; dadurch können die Eingaben wie bei einer gegebenen mathematischen Formel erfolgen. Es folgt eine Darstellung der Vorrangordnungen von einzelnen Berechnungen.

### Vorrangordnung

1. Funktionen wie  $\sin$ ,  $x^2$
2.  $y^x$ ,  $\sqrt[y]{x}$
3.  $\times$ ,  $\div$  (Berechnungen, die gleiche Vorrangordnung besitzen, werden in der Reihenfolge der Eingabe ausgeführt.)
4.  $+$ ,  $-$
5.  $=$ ,  $M+$ ,  $\Delta\%$

### Beispiel:

Tastensequenz und Reihenfolge der Berechnungen bei  $5 + 2 \times \sin 30 + 24 \times 5^3 =$



Die Nummern ① – ⑥ geben die Reihenfolge der Berechnungen an. Bei der Ausführung von Berechnungen mit einer höheren Vorrangordnung müssen die Berechnungen mit niedriger Vorrangordnung gespeichert werden.

### Hinweis:

Der Computer ist dazu mit einem zusätzlichen Speicherbereich für 3 schwebende Berechnungen ohne Klammern ausgestattet.

Da der Speicherbereich ebenfalls für Berechnungen, einschließlich Klammern, verwendet werden kann, können Berechnungen nach einer gegebenen mathematischen Formel ausgeführt werden, wenn die Klammern bzw. die Vorrangordnung der schwebenden Berechnungen nicht 8 überschreiten.

Funktionen mit einer Variablen ( $x^2$ ,  $1/x$ ,  $n!$ ,  $\rightarrow$ DEG,  $\rightarrow$ D.MS, usw.) werden sofort nach dem Drücken der Tasten berechnet und benötigen somit keinen Speicher.

### Berechnungen ohne Klammern

#### Beispiel:

1  $\boxed{+}$  2  $\boxed{=}$  1 Ebene

1  $\boxed{+}$  2  $\boxed{\times}$  3  $\boxed{=}$  2 Ebenen

1  $\boxed{+}$  2  $\boxed{\times}$  3  $\boxed{y^x}$  4  $\boxed{=}$  3 Ebenen

1  $\boxed{+}$  2  $\boxed{\times}$  3  $\boxed{y^x}$  4  $\boxed{\div}$  5  
  
 Mit Drücken der  $\boxed{y^x}$ -Taste sind 3 Ebenen erreicht. Nach dem Drücken der  $\boxed{\div}$ -Taste werden die Berechnungen "y<sup>x</sup>" mit höchster Priorität und "x" mit gleicher Priorität ausgeführt. Es bleiben somit nach Drücken der  $\boxed{\div}$ -Taste noch 2 Ebenen erhalten.

### Berechnungen mit Klammern

#### Beispiel:

i) 1  $\boxed{+}$  2  $\boxed{\times}$  3  $\boxed{y^x}$   $\boxed{(}$  4 Zahlen und Befehle bleiben schwebend.

4  $\boxed{\div}$  5

ii) 1  $\boxed{+}$  2  $\boxed{\times}$   $\boxed{(}$  3  $\boxed{-}$  Nach Drücken der  $\boxed{(}$ -Taste wird zuerst die Berechnung  $3 - 4 \div 5$  in Klammern ausgeführt; 2 Berechnungen bleiben schwebend.

4  $\boxed{\div}$  5  $\boxed{)}$

#### Hinweis:

Klammern können verwendet werden, wenn die Anzahl der schwebenden Berechnungen nicht acht überschreitet. Bei Folgeberechnungen können bis zu 15 Klammern eingegeben werden.

#### Beispiel:

$a \times (((b - c \times ((d + e) \times f) + g) \dots$

Es können bis zu 15 Klammern nacheinander verwendet werden.

# 5. TECHNIKERSOFTWARE-MODUS

Mit der Technikersoftware können Sie mathematische Formeln und physikalische Konstanten abrufen bzw. metrische Umwandlungen und Berechnungen mit komplexen Zahlen durchführen.

## Wahl eines Programms

Die Technikersoftware hat ein zweiseitiges Programm-Menü. Mit den Tasten  bzw.  können Sie eine dieser Seiten wählen. Zur Wahl und Ausführung des Programms wird die entsprechende Nummer des Menüs eingegeben.

Ein Technikersoftware-Programm muß im Speicherbereich für Programme geladen sein, damit es ausgeführt werden kann. Bei der erstmaligen Wahl eines Technikersoftware-Programms fragt der Computer zur Bestätigung, ob andere BASIC-Programme des Speicherbereiches überschrieben werden sollen. (Die Frage zur Bestätigung unterbleibt, wenn im Speicher kein Programm vorhanden ist.)  drücken, wenn nicht überschrieben werden soll; damit wird die Technikersoftware-Betriebsart beendet. Nun kann das BASIC-Programm auf die RAM-Diskette oder eine Kassette gesichert werden. Wenn das bestehende Programm überschrieben werden kann, wird bei der Frage  gedrückt. Nun führt der Computer das gewählte Programm aus. **Nach einer kurzen Verzögerung, während der "GOTO 100" angezeigt wird, erscheint die erste Anzeige des gewählten Programms.**

## Hinweis:

Andere Informationen im Zusammenhang mit einem BASIC-Programm gehen beim Überschreiben ebenfalls verloren.

Wenn Sie z.B. versuchen, das Programm FACTORIZATION zu benutzen und Sie gerade das Programm INTEGRATION verwenden, wird die Taste  einmal gedrückt. Danach  +  drücken; der Computer gibt nun das Menü der Technikersoftware-Programme aus. Danach die entsprechende Nummer für das Programm FACTORIZATION eingeben.

In der Programm-Betriebsart können Sie alle Technikersoftware-Programme ansehen oder modifizieren und die modifizierten Programme auf die RAM-Diskette oder ein anderes Speichermedium sichern. Da die Original-Inhalte des Programms im ROM gespeichert sind, können sie jederzeit wieder geladen werden.

Bei den Erklärungen in diesem Kapitel wird davon ausgegangen, daß Sie sich bereits in der Technikersoftware-Betriebsart befinden, die durch Drücken von  +  aktiviert wird. In dieser Betriebsart können Formeln mit den Tasten  bzw.  abgerufen werden. Durch längeres Drücken einer dieser Tasten können die Formeln der Reihe nach abgerufen werden.

### Hinweise:

- Wenn bei Eingabe der numerischen Daten ein Fehler auftritt, wird der Fehler mit der **CE**-Taste gelöscht und dann die richtigen Daten eingegeben.
- Wenn sich der Computer in der Betriebsart für Technikersoftware befindet, kann die **OFF**-Taste nicht zum Ausschalten der Stromversorgung verwendet werden. In diesem Fall erst die **BREAK**-Taste drücken und dann die **OFF**-Taste.

Bei der Ausführung eines Technikersoftware-Programms werden alle BASIC-Programme und alle Werte für Variable, die in BASIC zugewiesen wurden, gelöscht.

Bei Brüchen oder Quadratwurzeln ( $\sqrt{\quad}$ ) innerhalb von Gleichungen zeigt das Display die folgenden Notationen an:

**Beispiel:** •  $\tan 2\theta = \frac{2 \tan \theta}{1 - \tan^2 \theta} \rightarrow \tan 2\theta = 2 \tan \theta / (1 - \tan^2 \theta)$

•  $\sin \frac{\theta}{2} = \pm \sqrt{\frac{1 - \cos \theta}{2}} \rightarrow \sin(\theta/2) = \pm \sqrt{(1 - \cos \theta)/2}$

- Es kann zu Änderungen der physikalischen Konstanten kommen, die in diesem Handbuch beschrieben werden.
- Mit der Technikersoftware wird möglicherweise nicht der erwartete Exaktheitsgrad erreicht, den der Benutzer erwartet. Überprüfen Sie sorgfältig die Exaktheit der Berechnungsergebnisse und prüfen Sie die Exaktheit der Daten, die zur Berechnung verwendet werden.

## Liste der Technikersoftware

Die Größensannäherungen beziehen sich auf die ungefähre Anzahl der Byte, die von dem Programm benötigt werden. (Ausgenommen Variable)

Programmbezeichnung	Ungefähre Größe (Byte)
FACTORIZATION (Zerlegung in Faktoren)	5.800
TRIGONOMETRIC FUNCTION (Trigonometrische Funktionen)	10.100
INTEGRATION (Integralrechnung)	7.900
GREEK (Griechisch)	2.000
PHYSICAL CONSTANT (Physikalische Konstanten)	8.200
METRIC CONVERSION (Metrische Umwandlung)	6.700
COMPLEX NUMBER (Komplexe Zahlen)	3.200

## FACTORIZATION (Zerlegung in Faktoren)

Dieses Programm enthält die folgenden 26 Gleichungen zur Zerlegung in Faktoren:

- (1)  $a^2 - b^2 = (a + b)(a - b)$
- (2)  $a^3 \pm b^3 = (a \pm b)(a^2 \mp ab + b^2)$
- (3)  $a^4 - b^4 = (a + b)(a - b)(a^2 + b^2)$
- (4)  $a^4 + b^4 = (a^2 + \sqrt{2}ab + b^2)(a^2 - \sqrt{2}ab + b^2)$
- (5)  $a^n - b^n = (a - b)(a^{n-1} + a^{n-2}b + a^{n-3}b^2 + \dots + b^{n-1})$
- (6)  $a^n + b^n = (a + b)(a^{n-1} - a^{n-2}b + a^{n-3}b^2 - \dots + b^{n-1})$  ( $n$  : eine Primzahl)
- (7)  $a^2 \pm 2ab + b^2 = (a \pm b)^2$
- (8)  $a^3 \pm 3a^2b + 3ab^2 \pm b^3 = (a \pm b)^3$
- (9)  $(a \pm b)^2 \mp 4ab = (a \mp b)^2$
- (10)  $a^2 + b^2 + c^2 + 2bc + 2ca + 2ab = (a + b + c)^2$
- (11)  $a^4 + a^2b^2 + b^4 = (a^2 + ab + b^2)(a^2 - ab + b^2)$
- (12)  $a^3 + b^3 + c^3 - 3abc = (a + b + c)(a^2 + b^2 + c^2 - bc - ca - ab)$
- (13)  $(ac - bd)^2 + (ad + bc)^2 = (a^2 + b^2)(c^2 + d^2)$
- (14)  $(ac + bd)^2 + (ad - bc)^2 = (a^2 + b^2)(c^2 + d^2)$
- (15)  $(ac + bd)^2 - (ad + bc)^2 = (a^2 - b^2)(c^2 - d^2)$
- (16)  $(ac - bd)^2 - (ad - bc)^2 = (a^2 - b^2)(c^2 - d^2)$
- (17)  $a^2(b - c) + b^2(c - a) + c^2(a - b) = -(b - c)(c - a)(a - b)$
- (18)  $(b - c)^3 + (c - a)^3 + (a - b)^3 = 3(b - c)(c - a)(a - b)$
- (19)  $a^4 + b^4 + c^4 - 2b^2c^2 - 2c^2a^2 - 2a^2b^2 = (a + b + c)(b - c - a)(c - a - b)(a - b - c)$
- (20)  $x^2 + (a + b)x + ab = (x + a)(x + b)$
- (21)  $acx^2 + (ad + bc)x + bd = (ax + b)(cx + d)$
- (22)  $x^3 + (a + b + c)x^2 + (bc + ca + ab)x + abc = (x + a)(x + b)(x + c)$
- (23)  $a^2 - b^2 - c^2 - 2bc = (a + b + c)(a - b - c)$
- (24)  $(a + b + c)(bc + ca + ab) - abc = (b + c)(c + a)(a + b)$
- (25)  $(a + b + c)^3 - (a^3 + b^3 + c^3) = 3(b + c)(c + a)(a + b)$
- (26)  $a^3(b - c) + b^3(c - a) + c^3(a - b) = -(b - c)(c - a)(a - b)(a + b + c)$

Eingabe:

1

$$\begin{array}{l} * \text{ FACTORIZATION } * \\ 1 : a^2 - b^2 = (a+b)(a-b) \end{array}$$

↓ ↓ ↓

$$\begin{array}{l} * \text{ FACTORIZATION } * \\ 4 : a^4 + b^4 \\ = (a^2 + \sqrt{2} \cdot ab + b^2)(a^2 - \sqrt{2} \cdot \\ ab + b^2) \end{array}$$

Mit den folgenden Tasten wird eine spezifische Gleichung abgerufen:



: Abruf der ersten Gleichung.



: Abruf der 26. (letzten) Gleichung.



: Abruf der nächsten Gleichung.



: Abruf der vorhergehenden Gleichung.

# TRIGONOMETRIC FUNCTION (Trigonometrische Funktionen)

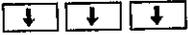
Dieses Programm enthält die folgenden 49 trigonometrischen Funktionsgleichungen.

- (1)  $\sin^2\theta + \cos^2\theta = 1$
- (2)  $1 + \tan^2\theta = \sec^2\theta$
- (3)  $1 + \cot^2\theta = \operatorname{cosec}^2\theta$
- (4)  $\sin(\alpha \pm \beta) = \sin \alpha \cdot \cos \beta \pm \cos \alpha \cdot \sin \beta$
- (5)  $\cos(\alpha \pm \beta) = \cos \alpha \cdot \cos \beta \mp \sin \alpha \cdot \sin \beta$
- (6)  $\tan(\alpha \pm \beta) = \frac{\tan \alpha \pm \tan \beta}{1 \mp \tan \alpha \cdot \tan \beta}$
- (7)  $\cot(\alpha \pm \beta) = \frac{\cot \alpha \cdot \cot \beta \pm 1}{\cot \beta \pm \cot \alpha}$
- (8)  $\sin(\alpha + \beta) \cdot \sin(\alpha - \beta) = \sin^2\alpha - \sin^2\beta$
- (9)  $\sin(\alpha + \beta) \cdot \sin(\alpha - \beta) = \cos^2\beta - \cos^2\alpha$
- (10)  $\cos(\alpha + \beta) \cdot \cos(\alpha - \beta) = \cos^2\alpha - \sin^2\beta$
- (11)  $\cos(\alpha + \beta) \cdot \cos(\alpha - \beta) = \cos^2\beta - \sin^2\alpha$
- (12)  $\sin(\alpha \pm \beta) \cdot \cos(\alpha \mp \beta) = \sin \alpha \cdot \cos \alpha \pm \sin \beta \cdot \cos \beta$
- (13)  $\frac{\sin(\alpha + \beta)}{\sin(\alpha - \beta)} = \frac{\tan \alpha + \tan \beta}{\tan \alpha - \tan \beta}$
- (14)  $\frac{\cos(\alpha + \beta)}{\cos(\alpha - \beta)} = \frac{1 - \tan \alpha \cdot \tan \beta}{1 + \tan \alpha \cdot \tan \beta}$
- (15)  $\sin 2\theta = 2 \sin \theta \cdot \cos \theta$
- (16)  $\cos 2\theta = \cos^2\theta - \sin^2\theta$
- (17)  $\cos 2\theta = 1 - 2 \sin^2\theta$
- (18)  $\cos 2\theta = 2 \cos^2\theta - 1$
- (19)  $\tan 2\theta = \frac{2 \tan \theta}{1 - \tan^2\theta}$
- (20)  $\sin \frac{\theta}{2} = \pm \sqrt{\frac{1 - \cos \theta}{2}}$
- (21)  $\cos \frac{\theta}{2} = \pm \sqrt{\frac{1 + \cos \theta}{2}}$
- (22)  $\tan \frac{\theta}{2} = \pm \sqrt{\frac{1 - \cos \theta}{1 + \cos \theta}}$
- (23)  $\tan \frac{\theta}{2} = \frac{1 - \cos \theta}{\sin \theta}$
- (24)  $\tan \frac{\theta}{2} = \frac{\sin \theta}{1 + \cos \theta}$
- (25)  $\tan \frac{\theta}{2} = \operatorname{cosec} \theta - \cot \theta$
- (26)  $\cot \frac{\theta}{2} = \pm \sqrt{\frac{1 + \cos \theta}{1 - \cos \theta}}$
- (27)  $\cot \frac{\theta}{2} = \frac{\sin \theta}{1 - \cos \theta}$
- (28)  $\cot \frac{\theta}{2} = \frac{1 + \cos \theta}{\sin \theta}$
- (29)  $\cot \frac{\theta}{2} = \operatorname{cosec} \theta + \cot \theta$
- (30)  $\sin 3\theta = -4 \sin^3\theta + 3 \sin \theta$
- (31)  $\cos 3\theta = 4 \cos^3\theta - 3 \cos \theta$
- (32)  $\tan 3\theta = \frac{3 \tan \theta - \tan^3\theta}{1 - 3 \tan^2\theta}$
- (33)  $\sin \alpha \cdot \cos \beta = \frac{\sin(\alpha + \beta) + \sin(\alpha - \beta)}{2}$
- (34)  $\cos \alpha \cdot \sin \beta = \frac{\sin(\alpha + \beta) - \sin(\alpha - \beta)}{2}$
- (35)  $\cos \alpha \cdot \cos \beta = \frac{\cos(\alpha + \beta) + \cos(\alpha - \beta)}{2}$
- (36)  $\sin \alpha \cdot \sin \beta = -\frac{\cos(\alpha + \beta) - \cos(\alpha - \beta)}{2}$
- (37)  $\sin \alpha + \sin \beta = 2 \sin\left(\frac{\alpha + \beta}{2}\right) \cdot \cos\left(\frac{\alpha - \beta}{2}\right)$
- (38)  $\sin \alpha - \sin \beta = 2 \cos\left(\frac{\alpha + \beta}{2}\right) \cdot \sin\left(\frac{\alpha - \beta}{2}\right)$
- (39)  $\cos \alpha + \cos \beta = 2 \cos\left(\frac{\alpha + \beta}{2}\right) \cdot \cos\left(\frac{\alpha - \beta}{2}\right)$
- (40)  $\cos \alpha - \cos \beta = -2 \sin\left(\frac{\alpha + \beta}{2}\right) \cdot \sin\left(\frac{\alpha - \beta}{2}\right)$
- (41)  $\tan \alpha \pm \tan \beta = \frac{\sin(\alpha \pm \beta)}{\cos \alpha \cdot \cos \beta}$
- (42)  $\cot \alpha \pm \cot \beta = \frac{\sin(\beta \pm \alpha)}{\sin \alpha \cdot \sin \beta}$
- (43)  $\cos \alpha \pm \sin \alpha = \sqrt{2} \sin(45^\circ \pm \alpha)$
- (44)  $\cos \alpha \pm \sin \alpha = \sqrt{2} \cos(45^\circ \mp \alpha)$
- (45)  $\tan\left(45^\circ \pm \frac{\theta}{2}\right) = \sec \theta \pm \tan \theta$
- (46)  $\tan\left(45^\circ \pm \frac{\theta}{2}\right) = \frac{1 \pm \sin \theta}{\cos \theta}$
- (47)  $\tan\left(45^\circ \pm \frac{\theta}{2}\right) = \cot\left(45^\circ \mp \frac{\theta}{2}\right)$
- (48)  $\tan(45^\circ + \theta) = \frac{1 + \tan \theta}{1 - \tan \theta}$
- (49)  $\cot(45^\circ - \theta) = \frac{1 + \cot \theta}{1 - \cot \theta}$

Eingabe:

2

$$\begin{array}{c} * \text{ TRIGONOMETRY } * \\ 1: \sin^2 \theta + \cos^2 \theta = 1 \end{array}$$



$$\begin{array}{c} * \text{ TRIGONOMETRY } * \\ 4: \sin(\alpha \pm \beta) \\ = \sin \alpha \cdot \cos \beta \pm \cos \alpha \cdot \sin \beta \end{array}$$

Mit den folgenden Tasten wird eine spezifische Gleichung abgerufen:



: Abruf der ersten Gleichung.



: Abruf der 49. (letzten) Gleichung.



: Abruf der nächsten Gleichung.



: Abruf der vorhergehenden Gleichung.

# INTEGRATION (Integralrechnung)

Dieses Programm enthält die folgenden 42 Gleichungen für die Integralrechnung:

- |  |  |
|--|--|
| <p>(1) <math>\int dx = x + C</math></p> <p>(2) <math>\int x^n dx = \frac{x^{n+1}}{n+1} + C \quad (n+1 \neq 0)</math></p> <p>(3) <math>\int \frac{1}{x} dx = \log x  + C</math></p> <p>(4) <math>\int \frac{1}{x \pm a} dx = \log x \pm a  + C</math></p> <p>(5) <math>\int e^x dx = e^x + C</math></p> <p>(6) <math>\int e^{ax} dx = \frac{1}{a} e^{ax} + C</math></p> <p>(7) <math>\int a^x dx = \frac{a^x}{\log a} + C \quad (a &gt; 0, a \neq 1)</math></p> <p>(8) <math>\int a^{nx} dx = \frac{a^{nx}}{n \cdot \log a} + C \quad (a &gt; 0, a \neq 1)</math></p> <p>(9) <math>\int \log x dx = x(\log x - 1) + C</math></p> <p>(10) <math>\int x^n \cdot \log x dx = \frac{x^{n+1}}{n+1} \left( \log x - \frac{1}{n+1} \right) + C \quad (n \neq -1)</math></p> <p>(11) <math>\int x e^{ax} dx = \frac{e^{ax}}{a^2} \cdot (ax - 1) + C</math></p> <p>(12) <math>\int \sin x dx = -\cos x + C</math></p> <p>(13) <math>\int \sin ax dx = -\frac{1}{a} \cdot \cos ax + C</math></p> <p>(14) <math>\int \cos x dx = \sin x + C</math></p> <p>(15) <math>\int \cos ax dx = \frac{1}{a} \cdot \sin ax + C</math></p> <p>(16) <math>\int \tan x dx = -\log \cos x  + C</math></p> <p>(17) <math>\int \tan ax dx = -\frac{1}{a} \log \cos ax + C</math></p> <p>(18) <math>\int \cot x dx = \log \sin x  + C</math></p> <p>(19) <math>\int \cot ax dx = \frac{1}{a} \log \sin ax + C</math></p> <p>(20) <math>\int \sec ax dx = \frac{1}{a} \log \tan \left( \frac{\pi}{4} + \frac{ax}{2} \right) + C</math></p> <p>(21) <math>\int \sec ax dx = \frac{1}{2a} \log \left( \frac{1 + \sin ax}{1 - \sin ax} \right) + C</math></p> | <p>(22) <math>\int \operatorname{cosec} ax dx = \frac{1}{a} \log \tan \frac{ax}{2} + C</math></p> <p>(23) <math>\int \operatorname{cosec} ax dx = -\frac{1}{2a} \log \left( \frac{1 + \cos ax}{1 - \cos ax} \right) + C</math></p> <p>(24) <math>\int \sin^2 x dx = \frac{1}{2} x - \frac{1}{4} \sin 2x + C</math></p> <p>(25) <math>\int \cos^2 x dx = \frac{1}{2} x + \frac{1}{4} \sin 2x + C</math></p> <p>(26) <math>\int \sec^2 ax dx = \frac{1}{a} \cdot \tan ax + C</math></p> <p>(27) <math>\int \operatorname{cosec}^2 ax dx = -\frac{1}{a} \cdot \cot ax + C</math></p> <p>(28) <math>\int \frac{1}{\sin x} dx = \log \tan \frac{x}{2} + C</math></p> <p>(29) <math>\int \frac{1}{\cos x} dx = \log \tan \left( \frac{\pi}{4} + \frac{x}{2} \right) + C</math></p> <p>(30) <math>\int e^{ax} \sin bx dx = \frac{1}{a^2 + b^2} e^{ax} (a \cdot \sin bx - b \cdot \cos bx) + C</math></p> <p>(31) <math>\int e^{ax} \cos bx dx = \frac{1}{a^2 + b^2} e^{ax} (a \cdot \cos bx + b \cdot \sin bx) + C</math></p> <p>(32) <math>\int \sin^{-1} x dx = x \sin^{-1} x + \sqrt{1-x^2} + C</math></p> <p>(33) <math>\int \cos^{-1} x dx = x \cos^{-1} x - \sqrt{1-x^2} + C</math></p> <p>(34) <math>\int \sinh x dx = \cosh x + C</math></p> <p>(35) <math>\int \cosh x dx = \sinh x + C</math></p> <p>(36) <math>\int \tanh x dx = \log \cosh x + C</math></p> <p>(37) <math>\int \frac{1}{\sqrt{a^2 - x^2}} dx = \sin^{-1} \frac{x}{a} + C \quad ( x  &lt; a)</math></p> <p>(38) <math>\int \frac{1}{a^2 - x^2} dx = \frac{1}{2a} \log \left  \frac{a+x}{a-x} \right  + C</math></p> <p>(39) <math>\int \frac{1}{a^2 + x^2} dx = \frac{1}{a} \tan^{-1} \frac{x}{a} + C</math></p> <p>(40) <math>\int \frac{1}{\sqrt{x^2 \pm a^2}} dx = \log(x + \sqrt{x^2 \pm a^2}) + C</math></p> <p>(41) <math>\int \sqrt{a^2 - x^2} dx = \frac{1}{2} \cdot \left( x \sqrt{a^2 - x^2} + a^2 \sin^{-1} \frac{x}{a} \right) + C</math></p> <p>(42) <math>\int \frac{1}{x^2 - a^2} dx = \frac{1}{2a} \log \left( \frac{x-a}{x+a} \right) + C \quad (x &gt; a)</math></p> |
|--|--|

## Hinweis:

- "log" bedeutet natürlicher Logarithmus ( $\log_e$ ).
- Alle trigonometrischen Funktionen werden als Radiane dargestellt.

Eingabe:

3

```
* INTEGRATION *  
1: ∫ dx = x + c
```

↓ ↓ ↓

```
* INTEGRATION *  
4: ∫ 1 / (x ± a) dx  
= log | x ± a | + C
```

Mit den folgenden Tasten wird eine spezifische Gleichung abgerufen:



: Abruf der ersten Gleichung.



: Abruf der 42. (letzten) Gleichung.



: Abruf der nächsten Gleichung.



: Abruf der vorhergehenden Gleichung.

## GREEK (Griechisch)

Dieses Programm enthält die folgenden 24 griechischen Buchstaben als Groß- und Kleinbuchstaben, ihre Aussprache und die entsprechenden lateinischen Buchstaben:

<i>A</i>	$\alpha$	Alpha	(a)	<i>N</i>	$\nu$	Nu	(n)
<i>B</i>	$\beta$	Beta	(b)	<i>E</i>	$\xi$	Xi	(x)
<i>Γ</i>	$\gamma$	Gamma	(g)	<i>O</i>	$o$	Omicron	(ö)
<i>Δ</i>	$\delta$	Delta	(d)	<i>Π</i>	$\pi$	Pi	(p)
<i>E</i>	$\epsilon$	Epsilon	(ë)	<i>P</i>	$\rho$	Rho	(r)
<i>Z</i>	$\zeta$	Zeta	(z)	<i>Σ</i>	$\sigma$	Sigma	(s)
<i>H</i>	$\eta$	Eta	(ē)	<i>T</i>	$\tau$	Tau	(t)
<i>Θ</i>	$\theta$	Theta	(th)	<i>Υ</i>	$\upsilon$	Upsilon	(u)
<i>I</i>	$\iota$	Iota	(i)	<i>Φ</i>	$\phi$	Phi	(ph)
<i>K</i>	$\kappa$	Kappa	(k)	<i>X</i>	$\chi$	Chi	(ch)
<i>Λ</i>	$\lambda$	Lambda	(l)	<i>Ψ</i>	$\psi$	Psi	(ps)
<i>M</i>	$\mu$	Mu	(m)	<i>Ω</i>	$\omega$	Omega	(ō)

**Eingabe:**

↓ 4

*	GREEK	*
A	$\alpha$	Alpha (a)
B	$\beta$	Beta (b)
Γ	$\gamma$	Gamma (g)

↓ ↓ ↓

I	$\iota$	Iota (i)
K	$\kappa$	Kappa (k)
Λ	$\lambda$	Lambda (l)
M	$\mu$	Mu (m)

Mit den folgenden Tasten werden die einzelnen Buchstaben abgerufen:



: Abrufen des ersten Bildschirms.



: Abrufen des letzten Bildschirms.



: Abrufen des nächsten Bildschirms.



: Abrufen des vorhergehenden Bildschirms.

**Hinweis:**

Bei jedem Drücken dieser Tasten wird das Display drei Zeilen auf- bzw. abgerollt.

## PHYSICAL CONSTANT (Physikalische Konstanten)

Dieses Programm stellt die folgenden 47 physikalischen Konstanten dar:

Name und Symbol	Wert	Einheit
(1) Vakuumlichtgeschwindigkeit $c$	$2.99792458 \times 10^8$	$\text{m} \cdot \text{s}^{-1}$
(2) Gravitationskonstante $G$	$6.67259 \times 10^{-11}$	$\text{N} \cdot \text{m}^2 \cdot \text{kg}^{-2}$
(3) Gravitationsbeschleunigung $g$	9.80665	$\text{m} \cdot \text{s}^{-2}$
(4) Ruhemasse von Elektronen $m_e$	$9.1093897 \times 10^{-31}$	kg
(5) Ruhemasse von Protonen $m_p$	$1.6726231 \times 10^{-27}$	kg
(6) Ruhemasse von Neutronen $m_n$	$1.6749286 \times 10^{-27}$	kg
(7) Ruhemasse von Muonen $m_\mu$	$1.8835327 \times 10^{-28}$	kg
(8) Atomare Masseinheit $u$	$1.6605402 \times 10^{-27}$	kg
(9) Elementarladung $e$	$1.60217733 \times 10^{-19}$	C
(10) Planck-Konstante $h$	$6.6260755 \times 10^{-34}$	$\text{J} \cdot \text{s}$
(11) Boltzmann-Konstante $k$	$1.380658 \times 10^{-23}$	$\text{J} \cdot \text{K}^{-1}$
(12) Magnetische Leitfähigkeit $\mu_0$	$12.566370614 \times 10^{-7}$	$\text{H} \cdot \text{m}^{-1}$
(13) Absolute Dielektrizitätskonstante $\epsilon_0$	$8.854187817 \times 10^{-12}$	$\text{F} \cdot \text{m}^{-1}$
(14) Verhältnis von Elektronenladung zu Masse $e/m_e$	$1.75881962 \times 10^{11}$	$\text{C} \cdot \text{kg}^{-1}$
(15) Klassischer Elektronenradius $r_e = e^2/4\pi\epsilon_0 m_e c^2$	$2.81794092 \times 10^{-15}$	m
(16) Feinstrukturkonstante $\alpha = e^2/4\pi\epsilon_0 \hbar c$	$7.29735308 \times 10^{-3}$	
(17) Quantum des Umlaufintegrals $h/2m_e$	$3.63694807 \times 10^{-4}$	$\text{J} \cdot \text{s} \cdot \text{kg}^{-1}$
(18) Bohr-Radius $a_0 = 4\pi\epsilon_0 \hbar^2 / m_e e^2$	$5.29177249 \times 10^{-11}$	m
(19) Rydberg-Konstante $R_\infty = e^2/16\pi^2 \epsilon_0 \alpha \hbar c$	$1.0973731534 \times 10^7$	$\text{m}^{-1}$
(20) Magnetisches Flußquantum $h/2e$	$2.06783461 \times 10^{-15}$	Wb
(21) Bohr-Magneton $\mu_B = e\hbar/2m_e$	$9.2740154 \times 10^{-24}$	$\text{J} \cdot \text{T}^{-1}$
(22) Magnetisches Moment eines Elektrons $\mu_e$	$9.2847701 \times 10^{-24}$	$\text{J} \cdot \text{T}^{-1}$
(23) g-Faktor des freien Elektrons $2\mu_e/\mu_B$	2.002319304386	
(24) Kern-Magneton $\mu_N = e\hbar/2m_p$	$5.0507866 \times 10^{-27}$	$\text{J} \cdot \text{T}^{-1}$
(25) Magnetisches Moment eines Protons $\mu_p$	$1.41060761 \times 10^{-26}$	$\text{J} \cdot \text{T}^{-1}$
(26) g-Faktor des Protons $2\mu_p/\mu_N$	5.585694772	
(27) Gyromagnetisches Verhältnis eines Protons $\gamma_p$	$2.67522128 \times 10^8$	$\text{s}^{-1} \cdot \text{T}^{-1}$
(28) Magnetisches Moment eines Neutrons $\mu_n$	$9.6623707 \times 10^{-27}$	$\text{J} \cdot \text{T}^{-1}$
(29) Magnetisches Moment eines Muons $\mu_\mu$	$4.4904514 \times 10^{-26}$	$\text{J} \cdot \text{T}^{-1}$
(30) Compton-Wellenlänge eines Elektrons $\lambda_c = h/m_e c$	$2.42631058 \times 10^{-12}$	m

Name und Symbol	Wert	Einheit
(31) Compton-Wellenlänge eines Protons $\lambda_{cp} = h/m_p c$	$1.32141002 \times 10^{-15}$	m
(32) Stefan-Boltzmann-Konstante $\sigma = \pi^2 k^4 / 60 \hbar^3 c^2$	$5.67051 \times 10^{-8}$	$W \cdot m^{-2} \cdot K^{-4}$
(33) Avogadro-Konstante $N_A$	$6.0221367 \times 10^{23}$	$mol^{-1}$
(34) Normvolumen idealer Gase $V_0$	$2.241410 \times 10^{-2}$	$m^3 \cdot mol^{-1}$
(35) Gaskonstante $R = N_A k$	8.314510	$J \cdot mol^{-1} \cdot K^{-1}$
(36) Faraday-Konstante $F = N_A e$	$9.6485309 \times 10^4$	$C \cdot mol^{-1}$
(37) Josephsons Verhältnis Frequenz-Spannung $2e/h$	$4.8359767 \times 10^{14}$	$Hz \cdot V^{-1}$
(38) Widerstand der Quantenübergänge $R_H$	25812.8056	$\Omega$
(39) Elektronenvolt $eV$	$1.60217733 \times 10^{-19}$	J
(40) Astronomische Einheit $AU$	$1.49597870 \times 10^{11}$	m
(41) Parsek $pc$	$3.0856776 \times 10^{16}$	m
(42) Seemeile sea mile	1852	m
(43) Angström $\text{Å}$	$1 \times 10^{-10}$	m
(44) Knoten knot	1852/3600	$m \cdot s^{-1}$
(45) Torr Torr	101325/760	Pa
(46) Standard des atmosphärischen Drucks $atm$	101325	Pa
(47) Kalorie cal	4.1868	J

Eingabe:

↓ 5

\* PHYSICAL CONSTANT \*  
1 : c = 2.99792458E+8  
[ m · s<sup>-1</sup> ]

↓

\* PHYSICAL CONSTANT \*  
2 : G = 6.67259E-11  
[ N · m<sup>2</sup> · kg<sup>-2</sup> ]

Mit den folgenden Tasten wird eine spezifische Konstante abgerufen:

◀

: Abrufen der ersten physikalischen Konstante.

▶

: Abrufen der letzten physikalischen Konstante.

↓

: Abrufen der nächsten physikalischen Konstante.

↑

: Abrufen der vorhergehenden physikalischen Konstante.

# METRIC CONVERSION (Metrische Umwandlung)

Dieses Programm wandelt Einheiten für Längen, Flächen, Volumen, Gewicht und Energien um.

Längeneinheiten I		m	in(inch)	ft(foot)	yd(yard)
	m	1	39.3701	3.28084	1.09361
	in	0.0254	1	0.0833333	0.0277778
	ft	0.3048	12	1	0.333333
	yd	0.9144	36	3	1
	mile	1609.344	63360	5280	1760
	cm	0.01	0.393701	0.0328084	0.0109361
	Å	$1 \times 10^{-10}$	$3.93701 \times 10^{-9}$	$3.28084 \times 10^{-10}$	$1.09361 \times 10^{-10}$
pc	$3.0856776 \times 10^{16}$	$1.214834357 \times 10^{18}$	$1.01236145 \times 10^{17}$	$3.37452788 \times 10^{16}$	

Längeneinheiten II		mile	cm	Å	pc(parsec)
	m	0.000621371	100	$1 \times 10^{10}$	$3.24077927 \times 10^{-17}$
	in	$1.57828 \times 10^{-5}$	2.54	254000000	$8.231579346 \times 10^{-19}$
	ft	0.000189394	30.48	304800000	$9.877895215 \times 10^{-18}$
	yd	0.000568182	91.44	914400000	$2.963368564 \times 10^{-17}$
	mile	1	160934.4	$1.609344 \times 10^{13}$	$5.215528674 \times 10^{-14}$
	cm	$6.21371 \times 10^{-6}$	1	100000000	$3.24077927 \times 10^{-19}$
	Å	$6.21371 \times 10^{-14}$	0.00000001	1	$3.24077927 \times 10^{-27}$
pc	$1.917350576 \times 10^{13}$	$3.0856776 \times 10^{18}$	$3.0856776 \times 10^{26}$	1	

### Ablesung der Tabelle:

Zur Umwandlung von "Meter" in "Inch" wird mit 39,3701 multipliziert.

Andere Umwandlungen werden auf die gleiche Weise wie beschrieben durchgeführt.

Flächeneinheiten		m <sup>2</sup>	a(are)	acre	mile <sup>2</sup>
	m <sup>2</sup>	1	0.01	0.000247105	$3.86102 \times 10^{-7}$
	a	100	1	0.0247105	$3.86102 \times 10^{-5}$
	acre	4046.86	40.4686	1	0.0015625
	mile <sup>2</sup>	2589990	25899.9	640	1

Volumeneinheiten		cm <sup>3</sup>	m <sup>3</sup>	in <sup>3</sup> (inch)	ℓ (litre)	ft <sup>3</sup> (foot)
	cm <sup>3</sup>	1	0.000001	0.0610237	0.001	$3.53147 \times 10^{-5}$
	m <sup>3</sup>	1000000	1	61023.7	1000	35.3147
	in <sup>3</sup>	16.3871	$1.63871 \times 10^{-5}$	1	0.0163871	0.000578704
	ℓ	1000	0.001	61.0237	1	0.0353147
	ft <sup>3</sup>	28316.8	0.0283168	1728	28.3168	1

Gewichtseinheiten		g	kg	oz(ounce)	lb (pound)
	g	1	0.001	0.035274	0.00220462
	kg	1000	1	35.274	2.20462
	oz	28.3495	0.0283495	1	0.0625
	lb	453.59237	0.45359237	16	1

Energieeinheiten I		eV	erg	cm <sup>-1</sup>	Hz
	eV	1	1.60218 × 10 <sup>-12</sup>	8065.54	2.41799 × 10 <sup>14</sup>
	erg	6.24151 × 10 <sup>11</sup>	1	5.03411 × 10 <sup>15</sup>	1.50919 × 10 <sup>26</sup>
	cm <sup>-1</sup>	0.000123984	1.98645 × 10 <sup>-16</sup>	1	2.99792 × 10 <sup>10</sup>
	Hz	4.13567 × 10 <sup>-15</sup>	6.62608 × 10 <sup>-27</sup>	3.33564 × 10 <sup>-11</sup>	1
	K	8.61738 × 10 <sup>-5</sup>	1.38066 × 10 <sup>-16</sup>	0.695039	2.08367 × 10 <sup>10</sup>
	G	5.78838 × 10 <sup>-9</sup>	9.27402 × 10 <sup>-21</sup>	4.66864 × 10 <sup>-5</sup>	1399620
	J/mol	1.03643 × 10 <sup>-5</sup>	1.66054 × 10 <sup>-17</sup>	0.0835934	2506070000
	kcal/mol	0.0433854	6.9511 × 10 <sup>-14</sup>	349.926	1.04905 × 10 <sup>13</sup>

Energieeinheiten II		K	G	J/mol	kcal/mol
	eV	11604.5	172760000	96485.3	23.0492
	erg	7.24292 × 10 <sup>15</sup>	1.07828 × 10 <sup>20</sup>	6.02214 × 10 <sup>16</sup>	1.43862 × 10 <sup>13</sup>
	cm <sup>-1</sup>	1.43877	21419.5	11.9627	0.00285774
	Hz	4.79922 × 10 <sup>-11</sup>	7.14478 × 10 <sup>-7</sup>	3.99031 × 10 <sup>-10</sup>	9.53241 × 10 <sup>-14</sup>
	K	1	14887.4	8.31451	0.00198624
	G	0.000067171	1	0.000558494	1.33418 × 10 <sup>-7</sup>
	J/mol	0.120272	1790.53	1	0.000238889
	kcal/mol	503.463	7495250	4186.05	1

**Eingabe:**

↓ 6

* METRIC CONVERSION *	
1 - LENGTH	4 - WEIGHT
2 - AREA	5 - ENERGY
3 - VOLUME	

(Menü)

Mit den Zahlentasten  -  einen Gegenstand für die metrische Umwandlung wählen.

**Beispiel 1:**

Umwandlung von 12 Metern in Inch:

* METRIC CONVERSION *	
<L>	X = ?

Eingabe der umzuwandelnden Daten:

12 

```
* METRIC CONVERSION *
<L>   X= 12
from?
<   1:m, 2:in, 3:ft, 4:yd >
```

Eingabe der Origineleinheit der Daten:

 (m)

```
* METRIC CONVERSION *
<L>   X= 12
from m to?
<   1:m, 2:in, 3:ft, 4:yd >
```

Eingabe der neuen Einheit:

 (in)

```
* METRIC CONVERSION *
<L>   X= 12
m      12
in     472.4412
```

Das Ergebnis ist 472,4412 Inch.

### Beispiel 2:

Umwandlung von  $10^{-3}$  Inch in Angström (Å):

Beim Drücken der -Taste, während das Ergebnis einer metrischen Umwandlung für die Länge noch auf dem Display angezeigt wird, kann eine weitere Umwandlung für eine andere Längeneinheit vorgenommen werden.



```
* METRIC CONVERSION *
<L>   X= ?
```

1  +  -3   (in)

```
* METRIC CONVERSION *
<L>   X= 0.001
from in to?
<   1:m, 2:in, 3:ft, 4:yd >
```

Da sich Angström (Å) nicht auf dem Menü der Anzeige befindet, muß es erst mit der  (oder ) -Taste aufgerufen werden; danach mit der  -Taste Angström wählen.

  (in)

```
* METRIC CONVERSION *
<L>   X= 0.001
in     0.001
Å      254000
```

Das Ergebnis ist 254000 Å.

### Hinweise:

- Wenn das Ergebnis einer metrischen Umwandlung den Bereich von  $10^{-100} < X < 10^{100}$  übersteigt, erscheint die Meldung: "Answer not found". Die -Taste drücken und die entsprechenden Daten für eine Umwandlung eingeben.
- Wenn die -Taste gedrückt wird, während "X=?" auf dem Display angezeigt wird, geht der Computer auf die vorhergehende Anzeige zurück.

# COMPLEX NUMBER (Komplexe Zahlen)

Dieses Programm führt unter Verwendung der folgenden Tasteneingaben Berechnungen mit komplexen Zahlen aus.

Taste	Funktion
<input type="button" value="X"/>	Speichert den Wert nach X (in der Reihenfolge der reellen und imaginären Teile).
<input type="button" value="Y"/>	Speichert den Wert nach Y (in der Reihenfolge der reellen und imaginären Teile).
<input type="button" value="+"/>	$X + Y \rightarrow X$ : Führt eine Addition durch und speichert das Ergebnis nach X.
<input type="button" value="-"/>	$X - Y \rightarrow X$ : Führt eine Subtraktion durch und speichert das Ergebnis nach X.
<input type="button" value="X*"/>	$X*Y \rightarrow X$ : Führt eine Multiplikation durch und speichert das Ergebnis nach X.
<input type="button" value="÷"/>	$X/Y \rightarrow X$ : Führt eine Division durch und speichert das Ergebnis nach X.
<input type="button" value="M+"/>	$X + M \rightarrow M$ : Addiert den Wert von X zum Speicher.
<input type="button" value="RM"/>	$M \rightarrow X$ : Ruft den Wert aus dem Speicher auf und weist ihn X zu.
<input type="button" value="X→M"/>	$X \rightarrow M$ : Speichert den Wert von X im Speicher.
<input type="button" value="▶"/> oder <input type="button" value="◀"/>	$X \leftrightarrow Y$ : Tauscht den Wert für X und Y aus.
<input type="button" value="√"/>	$\sqrt{X} \rightarrow X$ : Berechnet die Quadratwurzel von X und speichert das Ergebnis nach X.
<input type="button" value="1/X"/>	$1/X \rightarrow X$ : Berechnet den Reziprokwert von X und speichert das Ergebnis nach X.
<input type="button" value="X²"/>	$X*X \rightarrow X$ : Berechnet das Quadrat von X und speichert das Ergebnis nach X.
<input type="button" value="arg"/>	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">                     Absoluter Wert von <math>X \rightarrow X</math>                      Argument von <math>X \rightarrow Y</math> </div> <div style="font-size: 2em; margin-right: 10px;">}</div> <div>                     Speichert die absoluten Werte von <math>X (\sqrt{X_R^2 + X_I^2})</math> nach X und das Argument von <math>X (\tan^{-1} \frac{X_I}{X_R})</math> nach Y; dabei ist <math>X_R</math> der reelle Teil von X und <math>X_I</math> der imaginäre Teil von X.                 </div> </div>

### Eingabe:

```

*   COMPLEX NUMBER   *
X =  0

<< X, Y, +-*/ , M+, RM, x→M >>
```

### Beispiel:

Zu  $X = 3 + 4i$  und  $Y = 6 + 9i$  soll X und Y addiert und dann das Ergebnis quadriert werden.

(X wird vom Menü gewählt)

```

*   COMPLEX NUMBER   *
X (real) = 0.  _

<< X, Y, +-*/ , M+, RM, x→M >>
```

3  4

```
* COMPLEX NUMBER *
X(real) =3
X(image)=0. 4
<< X,Y,+* / ,M+,RM,x→M >>
```

```
* COMPLEX NUMBER *
X= 3
+4i
<< X,Y,+* / ,M+,RM,x→M >>
```

(Y wird vom Menü gewählt)

```
* COMPLEX NUMBER *
Y(real) =0. _
<< X,Y,+* / ,M+,RM,x→M >>
```

6  9

```
* COMPLEX NUMBER *
Y= 6
+9i
<< X,Y,+* / ,M+,RM,x→M >>
```

```
* COMPLEX NUMBER *
Y= 6
+9i
<< X,Y,+* / ,M+,RM,x→M >>
```

Das Ergebnis ist  $(3 + 4i) + (6 + 9i) = 9 + 13i$ .

Zur Anzeige des Ergebnisses folgende Tasten drücken:

```
* COMPLEX NUMBER *
X= 9
+13i
<< X,Y,+* / ,M+,RM,x→M >>
```

Nun wird das Ergebnis quadriert:

```
* COMPLEX NUMBER *
X=-88
+234i
<< X,Y,+* / ,M+,RM,x→M >>
```

Das letzte Ergebnis ist:  $(9 + 13i)^2 = -88 + 234i$ .

### Hinweis:

Durch das Drücken der Taste  bzw.  wird der angezeigte Inhalt der vierten Zeile geändert.

# 6. STAT-MODUS

Mit dem Computer können statistische und Regressions-Berechnungen mit einer oder zwei Variablen durchgeführt werden. Bei statistischen Berechnungen werden Mittelwerte, Standardabweichungen und andere statistische Werte aus den Stichprobendaten erhalten. Regressions-Berechnungen gestatten die Bestimmung von Koeffizienten der linearen Regression oder von Schätzwerten aus Stichprobendaten.

## Wählen und Abschalten des STAT-Modus

Die Tasten **SHIFT** + **STAT** drücken, um das folgende STAT-Menü zur Anzeige zu bringen.

```
* Statistical analysis *
1: Single-variable stat
2: Two-variable stat
Select No.?
```

Durch Drücken der **1**-Taste werden Berechnungen mit Einzel-Variablen gewählt, mit der **2**-Taste Berechnungen mit Doppel-Variablen.

Zum Beenden der STAT-Betriebsart eine beliebige andere Betriebsart wählen.

## Statistische Berechnungen mit Einzel-Variablen

Durch statistische Berechnungen mit Einzel-Variablen können die folgenden Angaben gewonnen werden.

- $n$ : Umfang der Stichprobenwerte  $x$
- $\sum x$ : Summe der Stichprobenwerte  $x$
- $\sum x^2$ : Summe der Quadrate der Stichprobenwerte  $x$
- $\bar{x}$ : Stichproben-Mittelwert  $x$
- $s$ : Stichproben-Standardabweichung mit Grundgesamtheits-Parameter  $n-1$ .

$$s = \sqrt{\frac{\sum x^2 - n\bar{x}^2}{n-1}}$$

Diese Gleichung dient zur Schätzung der Standardabweichung einer Grundgesamtheit aus einem Teil der Stichprobendaten ( $x$ ) dieser Grundgesamtheit.

- $\sigma$ : Stichproben-Standardabweichung mit Grundgesamtheits-Parameter  $n$ .

$$\sigma = \sqrt{\frac{\sum x^2 - n\bar{x}^2}{n}}$$

Diese Gleichung nimmt die gesamte Grundgesamtheit als Stichprobendaten ( $x$ ) an oder bestimmt die Standardabweichung von Stichprobendaten, die als eine Grundgesamtheit angesehen werden.



```
* Delete a data *
x=_
```

Die Werte der falschen Daten oder der zu löschenden Daten genauso wie bei der Dateneingabe eingeben. Mehrere Datensätze können durch Verwendung von Kommas (,) gelöscht werden.

Nach dem Löschen werden die richtigen Daten am Bereitschaftszeichen eingegeben.

### Durchführung statistischer Berechnungen

Im Untermenü für Einzel-Variable die Taste  drücken, um das Untermenü für Analysen zur Anzeige zu bringen.

```
* Analysis * (x)
1:n      2:Σx    3:Σx2  4: x̄
5:s      6:σ
Select No.?
```

Durch Drücken der Tasten  bis  können jetzt folgende Statistiken erstellt werden:

<input type="text" value="1"/>	... n:	Umfang der Stichprobenwerte
<input type="text" value="2"/>	... $\sum x$ :	Summe der Stichprobenwerte
<input type="text" value="3"/>	... $\sum x^2$ :	Summe der Quadrate der Stichprobenwerte
<input type="text" value="4"/>	... $\bar{x}$ :	Stichproben-Mittelwerte
<input type="text" value="5"/>	... s:	Stichproben-Standardabweichung
<input type="text" value="6"/>	... $\sigma$ :	Grundgesamtheits-Standardabweichung

Durch Druck auf die **BREAK** -Taste wird das Untermenü für Einzel-Variable wieder angezeigt.

### Beginn einer neuen Berechnung (Löschen der vorherigen Daten)

Einen der folgenden Schritte ausführen:

1. Die STAT-Betriebsart zunächst beenden und dann erneut wählen. Die vorherigen Daten werden gelöscht.
2. Mit der Funktion für Löschen die Daten löschen. Im Untermenü für Einzel-Variable die Taste  drücken, um das Untermenü für Löschen zur Anzeige zu bringen.

```
* Delete *
1:Data
2:All clear
Select No.?
```

Das Untermenü für Löschen wird angezeigt.

```
* All clear *
1:YES
2:NO
Select No.?
```

Das Untermenü für "Alles Löschen" wird angezeigt.

Die  -Taste drücken, um die vorherigen Daten zu löschen. Beim Drücken von  bleiben die vorherigen Daten erhalten.

**Beispiel:**

Die Punktzahlen für 35 zufällig ausgewählte Studenten sind wie folgt. Bestimme den Mittelwert und die Standardabweichung dieser Punktzahlen.

Nr.	Punktzahl	Häufigkeit	Nr.	Punktzahl	Häufigkeit
1	30	1	5	70	8
2	40	1	6	80	9
3	50	4	7	90	5
4	60	5	8	100	2

Im STAT-Menü "1: Single variable stat" wählen.

1

```
* Functions *      (x)
  1: Input          2: Delete
  3: Analysis       4: Printer
Select No.?
```

"1: Input" wählen und die Daten eingeben.

1

30 ← 40 ← 50,4 ←  
60,5 ← 70,8 ← 80,9 ←  
90,5 ← 100,2 ←

```
20: x=80., 9.
29: x=90., 5.
34: x=100., 2.
36: x=_
```

Die Dateneingabe ist damit abgeschlossen.

Das Untermenü für Einzel-Variable zur Anzeige bringen.

BREAK

"3: Analysis" wählen.

3

```
* Analysis *      (x)
  1: n      2: Σx   3: Σx2 4: x̄
  5: s      6: σ
Select No.?
```

Ausgabe der Mittelwerte.

4

```
* Analysis *      (x)
  1: n      2: Σx   3: Σx2 4: x̄
  5: s      6: σ
x̄= 71.42857143
```

Ausgabe der Standardabweichungen der Grundgesamtheit.

6

```
* Analysis *      (x)
  1: n      2: Σx   3: Σx2 4: x̄
  5: s      6: σ
σ= 16.23802542
```

Zurückgehen auf das Untermenü für Einzel-Variable.

BREAK

- Zum Erhalt der Stichprobengröße, der Summe bzw. der Summe der Quadrate der Stichprobe oder der Standardabweichung wird , ,  oder  gedrückt.

Nach dem Erhalt von Zwischen-Statistiken, z.B. Mittelwerte und Standardabweichungen, können weitere Daten eingegeben werden, wenn im Untermenü für Einzel-Variable "1: Input" gewählt wird.

### Ausdrucken von Statistiken

Die Ergebnisse können auf dem zusätzlichen Drucker CE-126P ausgedruckt werden. Den Drucker an den Computer anschließen und die Stromversorgung einschalten. Die Daten eingeben und im Untermenü für Einzel-Variable "4: Printer" wählen, um die Statistiken auszudrucken.

```
* Functions *      (x)
  1:Input          2:Delete
  3:Analysis       4:Printer
Select No.?
```

Ausdruck-Beispiel:

```
n=                35.
Σx=               2500.
Σx²=              187800.
MEAN(x)= 71.42857143
s=                16.47508942
σ=                16.23802542
```

Nach dem Ausdruck geht die Anzeige auf das Untermenü für Einzel-Variable zurück.

## Statistische Berechnungen mit Doppel-Variablen

Das Vorgehen bei statistischen Berechnungen mit Doppel-Variablen entspricht dem für Einzel-Variable. Daher zuerst den Abschnitt über Berechnungen mit Einzel-Variablen durchlesen.

Durch Berechnungen mit Doppel-Variablen können die folgenden Statistiken erhalten werden.

$n$ ,  $\Sigma x$ ,  $\Sigma x^2$  und  $\bar{x}$ : Wie bei Berechnungen mit Einzel-Variablen.

$s_x$  und  $\sigma_x$ : Wie für  $s$  und  $\sigma$ .

$\Sigma y$ : Summe der Stichprobenwerte  $y$

$\Sigma y^2$ : Summe der Quadrate der Stichprobenwerte  $y$

$\Sigma xy$ : Summe der Produkte der  $x$ - und  $y$ -Stichprobenwerte

$\bar{y}$ : Stichproben-Mittelwert  $y$

$s_y$ : Stichproben-Standardabweichung mit Grundgesamtheits-Parameter  $n-1$ .

$$s_y = \sqrt{\frac{\Sigma y^2 - n\bar{y}^2}{n-1}}$$

$\sigma_y$ : Grundgesamtheits-Standardabweichung der Stichprobe ( $y$ ) mit Grundgesamtheits-Parameter  $n$ .

$$\sigma_y = \sqrt{\frac{\sum y^2 - n\bar{y}^2}{n}}$$

$a$ :  $a = \bar{y} - b\bar{x}$

Koeffizient für lineare Regression  $y = a + bx$

$b$ :  $b = \frac{S_{xy}}{S_{xx}}$

Koeffizient für lineare Regression  $y = a + bx$

$r$ :  $r = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}$

Korrelationskoeffizient

$x'$ :  $x' = \frac{y - a}{b}$

Schätzwert ( $x$ -Wert aus  $y$  geschätzt)

$y'$ :  $y' = a + bx$

Schätzwert ( $y$ -Wert aus  $x$  geschätzt)

**Hinweise:**

$$S_{xx} = \sum x^2 - \frac{(\sum x)^2}{n} \quad S_{yy} = \sum y^2 - \frac{(\sum y)^2}{n} \quad S_{xy} = \sum xy - \frac{\sum x \cdot \sum y}{n}$$

**Wahl von statistischen Berechnungen mit Doppel-Variablen**

Nach der Darstellung des STAT-Menüs wird  gedrückt, um statistische Berechnungen mit Doppel-Variablen zu wählen.

**Dateneingabe**

Im Untermenü für Doppel-Variable die  -Taste drücken, um das Bereitschaftszeichen für die Dateneingabe darzustellen. Die Daten für  $x$  und  $y$  wie gezeigt eingeben.

Zur Eingabe eines einzigen Datenpaares: Daten  $x$   Daten  $y$  .

Zur gleichzeitigen Eingaben von mehreren identischen Datenpaaren: Daten  $x$   Daten  $y$ , Datenhäufigkeit .

Zur Eingabe von negativen Werten die  -Taste vor Eingabe der Daten drücken.

Zum Abschluß der Dateneingabe  drücken und das Untermenü für Doppel-Variable zur Anzeige bringen.

**Durchführung statistischer Berechnungen**

Im Untermenü für Doppel-Variable die Taste  drücken, um das Untermenü für Analysen zur Anzeige zu bringen. Es gibt zwei Untermenüs, zwischen denen durch Drücken der Taste  oder  umgeschaltet werden kann. (Das Symbol  oder  erscheint.)

Beim Drücken der  -Taste im Untermenü für Doppel-Variable:  
(Erstes Untermenü für Analysen)

* Analysis *		(x, y) ↓	
1: n	2: $\sum x$	3: $\sum x^2$	4: $\bar{x}$
5: $s_x$	6: $\sigma_x$	7: $\sum y$	8: $\sum y^2$
Select No.?			



(Zweites Untermenü für Analysen)

```

* Analysis *      (x,y) ↑
1:Σxy 2:ȳ      3:sy 4:σy
5:a 6:b 7:r 8:x' 9:y'
Select No. ?

```

Durch Drücken von erscheint wieder das erste Untermenü für Analysen.

### Beispiel:

Die folgende Tabelle enthält das Datum (April), an dem Zugvögel ein bestimmtes Gebiet überfliegen, sowie die Temperaturen dieses Gebietes im März. Mit dieser Tabelle werden die Koeffizienten  $a$  und  $b$  der linearen Regression  $y = a+bx$  sowie der Korrelationskoeffizient  $r$  bestimmt. Das Datum des Durchflugs wird geschätzt, wenn die Durchschnittstemperatur im März  $9,1^{\circ}\text{C}$  beträgt. Weiterhin wird die Durchschnittstemperatur im März geschätzt, wenn das Durchflugdatum der Zugvögel der 10. April ist.

Jahr	1	2	3	4	5	6	7	8
Durchschnittstemperatur ( $x^{\circ}\text{C}$ )	6,2	7,0	6,8	8,7	7,9	6,5	6,1	8,2
Datum des Durchflugs (Tag $y$ )	13	9	11	5	7	12	15	7

Im STAT-Menü "2: Two-variable stat" wählen.

```

* Functions *      (x,y)
1:Input           2:Delete
3:Analysis        4:Printer
Select No. ?

```

"1: Input" wählen und die Daten eingeben.

```

6.2 [←] 13 [←] 7.0 [←]
9 [←] 6.8 [←] 11 [←]
8.7 [←] 5 [←] 7.9 [←]
7 [←] 6.5 [←] 12 [←]
6.1 [←] 15 [←] 8.2 [←]
7 [←]

```

```

y=15.
8:x=8.2
y=7.
9:x=_

```

Die Dateneingabe ist damit abgeschlossen.

Das Untermenü für Doppel-Variable zur Anzeige bringen.

**BREAK**

"3: Analysis" wählen und das zweite Untermenü für Analysen zur Anzeigen bringen.

Bestimmung des Koeffizienten  $a$ .

```

* Analysis *      (x,y) ↑
1:Σxy 2:ȳ      3:sy 4:σy
5:a 6:b 7:r 8:x' 9:y'
a=                      34.44951017

```

Bestimmung des Koeffizienten  $b$ .

6

```

* Analysis *      (x, y)  ↑
1: Σxy  2:  $\bar{y}$    3: sy   4: σy
5: a    6: b    7: r    8: x'  9: y'
b=                               -3.425018839

```

Schätzung des Durchflugsdatums.

9

```

* Analysis *      (x, y)
x=_

```

Eingabe der Durchschnittstemperatur.

9.1 ←

```

* Analysis *      (x, y)
x=9.1
y=                               3.281838734
x=_

```

(Geschätztes Datum: 3. April)

Anzeige des zweiten Untermenüs für Analysen.

BREAK

Schätzung der Durchschnittstemperatur.

8

```

* Analysis *      (x, y)
y=_

```

Eingabe des Durchflugsdatums.

10 ←

```

* Analysis *      (x, y)
y=10.
x=                               7.13850385
y=_

```

(Geschätzte Durchschnittstemperatur am 10. März: etwa 7,1°C)

**Hinweis:**

Die Ergebnisse von Statistiken oder Regressionen werden automatisch in den festen Variablen U bis Z gespeichert.

	U	V	W	X	Y	Z
Wert	$\Sigma y^2$	$\Sigma y$	$\Sigma xy$	$\Sigma x^2$	$\Sigma x$	n

- Der Inhalt von nicht berechneten Variablen wird Null.
- Diese Statistiken werden gelöscht, wenn die Betriebsart für Statistik erneut aktiviert wird.

# 7. RUN-MODUS

Die RUN-Betriebsart ist sehr vielseitig; damit können auch BASIC-Programme ausgeführt werden, die in der PRO-Betriebsart geschrieben wurden.

## Einstellen des RUN-Modus

Wählen der BASIC-Betriebsart durch Drücken der **BASIC**-Taste. Wenn PRO angezeigt wird, kann durch Drücken der **BASIC**-Taste die RUN-Betriebsart gewählt werden. Durch (>) wird angezeigt, daß der Computer nun bereit für Eingaben ist. Das Display sollte folgendermaßen aussehen:

```
      CAPS          DEG
RUN  MODE
>
      RUN
```

## Einige nützliche Hinweise

Wenn Sie bei der Eingabe einen Fehler machen und eine Fehlermeldung erscheint, kann der Fehler am einfachsten durch Drücken der **C•CE**-Taste gelöscht werden. Danach erfolgt die neue Eingabe.

Wenn der Computer aber blockiert (er reagiert überhaupt nicht mehr), betätigen Sie den RESET-Schalter, während die **ON**-Taste gedrückt wird (siehe Anhang E.)

Das Bereitschaftssymbol (>) weist Sie darauf hin, daß der Computer auf eine Eingabe wartet. Sobald Sie Daten eingeben, verschwindet das Bereitschaftssymbol, der Cursor ( \_ ) zeigt auf die nächste Eingabeposition.

Der Cursor kann mit den Tasten  (rechts),  (links),  (oben) und  (unten) bewegt werden.

Das Display des Computers besteht aus vier Zeilen (je 24 Zeichen pro Zeile). Alle Eingaben und Rechenergebnisse werden jeweils von der obersten Zeile aus angegeben. Wenn die Daten die Kapazität von 4 Zeilen überschreiten, wird der Anzeigehalt jeweils um eine Zeile nach oben gerollt (die erste angezeigte Zeile rollt nach oben aus dem Display).

Mit  informieren Sie den Computer, daß Sie die Dateneingabe beendet haben und daß der Computer die angegebenen Operationen durchführen kann. **Am Ende einer Eingabe muß die -Taste gedrückt werden, andernfalls bearbeitet der Computer die Berechnungen nicht.**

Bei der Ausführung numerischer Berechnungen erscheint die Eingabe links im Display und das Ergebnis rechts.

```
- 1 2 3 4 5 6 7 8 9 1 / 1 0
                        - 1 2 3 4 5 6 7 8 9 . 1
```

Benutzen Sie bei der Eingabe von Berechnungen in den Computer keine Dollarzeichen oder Kommata. Diese Zeichen haben in der BASIC-Programmiersprache eine ganz besondere Bedeutung.

Bei Verwendung der **SHIFT**-Taste zur Aktivierung der zweiten Funktion einer anderen Taste muß die **SHIFT**-Taste gehalten werden, während die andere Taste gedrückt wird. Die **2nd F**-Taste kann ebenfalls verwendet werden, z.B. in der CAL-Betriebsart.

Bei Verwendung der **SHIFT**-Taste wird das eigentliche Zeichen mit dem folgenden Tastendruck wiedergegeben. Wenn z.B. **SHIFT** + **Y** gedrückt wird, entsteht das Zeichen "&". Dieser Vorgang wird als **SHIFT** + **&** dargestellt.

Nach jeder Berechnung muß unbedingt **CE** eingegeben werden (außer bei Durchführung von Kettenrechnungen). Durch **CE** wird die Anzeige vom Display gelöscht oder eine Fehlerbedingung beseitigt. Dabei werden aber keine Speicherinhalte des Computers gelöscht.

### Hinweis:

Ausführliche Erläuterungen über die Dezimalstellen und die Modifizierungs-Funktion finden sich auf der Seite 31 im Kapitel über die Rechnungs-Betriebsart.

## Einfache Berechnungen

Der Computer rechnet auf 10 Stellen genau. Schalten Sie den Computer ein und bringen Sie ihn in den RUN-Modus. Nun versuchen Sie, die folgenden einfachen Rechenbeispiele nachzuvollziehen.

### Beispiel:

2 + 3 × 4 +  
2 **+** 3 **X\*** 4 **←**

2 + 3 * 4	14 .
-----------	------

### Beispiel:

5 × (-6) + 7 =  
5 **X\*** **-** 6 **+** 7 **←**

2 + 3 * 4	14 .
5 * - 6 + 7	-23 .

## Komplexere Berechnungen und Klammersetzung

Bei der Durchführung der Rechenbeispiele im CAL-Modus hätten Sie einige dieser Berechnungen in einem Schritt zusammenfassen können.

Sie können z.B. folgendes eingeben:

675 + 6750/45000

Solche komplexeren Rechnungen müssen jedoch sehr sorgfältig eingegeben werden, um Zweideutigkeiten zu vermeiden.

Bei der Durchführung komplexer mathematischer Operationen folgt der Computer spezifischen Regeln der impliziten Klammerung und der Vorrangordnung von mathematischen Operationen (siehe Seite 70). Verwenden Sie daher Klammern für eindeutige Bezeichnungen.

$(675 + 6750)/45000$  oder  $675 + (6750/45000)$

## Wiederabruf von Eingaben

Auch nachdem der Computer die Ergebnisse der Rechenoperation angezeigt hat, können Sie die vorherige Eingabe überprüfen und neu bearbeiten. Dazu benutzen Sie die Tasten  und . Die Rechts- und Links-Pfeile werden auch benutzt, um den Cursor innerhalb der Zeile zu bewegen. Die -Taste wird benutzt, um den Cursor an die Stelle hinter dem letzten Zeichen zu bringen. Die -Taste wird benutzt, um den Cursor auf das erste Zeichen zu bringen.

### Beispiel:

300  6 

3 0 0 / 6	5 0 .
-----------	-------

Ändern Sie die Berechnung auf 300/5.

Rufen Sie die letzte Eingabe mit Hilfe von  wieder auf.



3 0 0 / 6	5 0 .
3 0 0 / 6 _	

Da Sie daran gedacht haben,  zu benutzen, steht der Cursor nun am Ende der Anzeige. Benutzen Sie noch einmal , um den Cursor eine Stelle nach links zu bewegen.



3 0 0 / 6	5 0 .
3 0 0 / 	

Beachten Sie, daß nach dieser Bewegung der Cursor sich in ein blinkendes Viereck verwandelt. Immer, wenn Sie den Cursor auf ein schon bestehendes Zeichen setzen, wird er in dieser Form dargestellt.

Geben Sie nun anstelle der 6 eine 5 ein. Achtung beim Einsetzen von Zeichen: Wenn Sie ein neues Zeichen über ein bereits bestehendes eingeben, wird das ursprüngliche Zeichen endgültig gelöscht! Sie können keinen Ausdruck wieder abrufen, der gelöscht wurde.

5 

3 0 0 / 6	5 0 .
3 0 0 / 5	6 0 .

Es können auch Zeichen in die bestehende Eingabe eingefügt oder gelöscht werden. Ändern Sie die vorherige Berechnung auf 3000/5. Rufen Sie die Eingabe mit Hilfe von  noch einmal auf.



3 0 0 / 5	5 0 .
3 0 0 / 5	6 0 .

Weil Sie  benutzt haben, steht der blinkende Cursor jetzt auf dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, müssen Sie jetzt eine zusätzliche Null einfügen. Mit Hilfe von  bewegen Sie den Cursor, bis er über der Null steht. Wenn Sie nun mit der  -Taste eine Einfügung machen wollen, setzen Sie den blinkenden Cursor auf denjenigen Buchstaben, vor dem das neue Zeichen eingefügt werden soll.



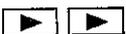
3 0 0 / 5	5 0 .
3 0 0 / 5	6 0 .

Durch Drücken der Taste  werden alle Zeichen um eine Stelle nach rechts bewegt und eine Leerstelle eingefügt. Der blinkende Cursor wird nun auf diese Leerstelle bewegt und zeigt damit die Stelle für eine neue Eingabe an. Geben Sie eine Null ein. Nach Korrektur der Eingabe kann das neue Ergebnis dargestellt werden.



3 0 0 / 5	6 0 .
3 0 0 0 / 5	6 0 0 .

Zum Löschen von Zeichen wird die  -Taste gedrückt. Ändern der vorherigen Berechnung auf 3/5. Rufen Sie die Eingabe mit Hilfe von  auf. Um die Eingabe zu korrigieren, müssen Sie die Nullen löschen. Mit Hilfe von  bewegen Sie den Cursor auf die erste Null. Wenn Sie ein Zeichen löschen wollen, setzen Sie den Cursor auf das zu löschende Zeichen.



3 0 0 0 / 5	6 0 .
3 0 0 0 / 5	6 0 0 .

Um die Nullen zu löschen, betätigen Sie nun die  -Taste.



3 0 0 0 / 5	6 0 .
3 / 5	6 0 0 .

Durch Betätigen von **DEL** rutschen alle Zeichen um eine Stelle nach links. Das Zeichen, auf dem der Cursor steht, wird gelöscht. Da Sie keine weiteren Veränderungen vorzunehmen haben, beenden Sie die Rechnung durch Darstellung des Ergebnisses.



3 0 0 0 / 5	
3 / 5	6 0 0 .
	0 . 6

### Hinweis:

Steht der Cursor auf einem Zeichen und Sie betätigen die **SPACE** -Taste, wird das betreffende Zeichen gelöscht und durch eine Leerstelle ersetzt. Mit **DEL** löschen Sie das Zeichen und die von ihm eingenommene Leerstelle.

Sie können auch die **BS** -Taste zum Löschen von Fehlern verwenden. Beachten Sie, daß die **BS** -Taste den Cursor um eine Position zurückversetzt und das dort befindliche Zeichen löscht, während die **DEL** -Taste das Zeichen löscht, auf dem der Cursor steht.

## Fehler

Der Abruf der vorherigen Eingabe ist notwendig, wenn Sie eine Fehlermeldung erhalten. Sie haben zum Beispiel aus Versehen folgendes in den Computer eingegeben:

**C•CE** 3 0 0  $\div$   $\div$  5

3 0 0 // 5
ERROR 10

“ERROR 10” ist die einfache Art des Computers, Ihnen mitzuteilen: “Ich weiß nicht, was ich hier tun soll.” Wenn Sie in diesem Fall die Taste oder betätigen, erscheint der blinkende Cursor an der Stelle, wo der Fehler liegt.

(oder )

3 0 0 // 5
ERROR 10
3 0 0 /  5

Um diesen Fehler zu korrigieren, benutzen Sie die **DEL** -Taste.

**DEL**

3 0 0 // 5
ERROR 10
3 0 0 / 5
6 0 .

Wenn Sie bei einem Fehler feststellen, daß bei der Eingabe ein Zeichen vergessen wurde, können Sie es mit Hilfe der **INS** -Taste einfügen.

Wenn Sie den Computer wie einen normalen Taschenrechner benutzen, werden Sie in der Hauptsache mit Syntaxfehlern konfrontiert. Eine komplette Aufstellung der Fehlermeldungen und ihrer Bedeutungen finden Sie im Anhang B.

## Verkettung von Berechnungen

Der Computer erlaubt die Weiterverwendung von dem Ergebnis einer Rechnung in der folgenden Berechnung.

### Beispiel:

Was ist 15% von  $300 * 150$ ?

**C•CE** 3 0 0 **X\*** 1 5 0 **←**

3 0 0 * 1 5 0	4 5 0 0 0 .
---------------	-------------

Bei verketteten Berechnungen ist es nicht notwendig, die vorherigen Ergebnisse nach einmal einzugeben, aber drücken Sie zwischen den Eingaben KEINESFALLS

**C•CE**.

**X\*** . 1 5 **←**

3 0 0 * 1 5 0	4 5 0 0 0 .
4 5 0 0 0 . * . 1 5	6 7 5 0 .

Beachten Sie, daß der Computer das Ergebnis der ersten Berechnung automatisch links im Display anzeigt und in die neue Rechnung mit einbezieht, wenn Sie die Kommandos für die zweite Berechnung ( $*.15$ ) eingeben. In verketteten Berechnungen muß die Eingabe mit einem Operator beginnen. Beenden Sie die Eingabe, wie immer, mit **←**.

### Hinweis:

Die Tasten **%** und **Δ%** können in der RUN-Betriebsart nicht für Berechnungen verwendet werden. Die Taste **%** kann nur als Zeichen eingesetzt werden, während die Taste **Δ%** überhaupt nicht benutzt werden kann.

Beispiel: 4 5 0 0 0 **X\*** 1 5 **SHIFT** + **%** **←** → ERROR 10

Zur Änderung des Vorzeichens des letzten Ergebnisses mit  $-1$  multiplizieren:

**X\*** - 1 **←**

4 5 0 0 0 . * . 1 5	6 7 5 0 .
6 7 5 0 . * - 1	- 6 7 5 0 .

Das Vorzeichen kann auch durch Drücken von **SHIFT** + **(-)** oder **+/-** geändert werden.

## Berechnungen mit Konstanten

Mit der Taste **CONST** können Konstanten wie im folgenden beschrieben für die Grundrechenarten verwendet werden.

### Verwendung von Konstanten

Addition: **+** a **CONST** oder a **+** **CONST**

Subtraktion: **--** a **CONST** oder a **-** **CONST**

Multiplikation: **X\*** a **CONST** oder a **X\*** **CONST**

Division: **÷/** a **CONST** oder a **÷/** **CONST**

Dabei bedeutet "a" die Konstante.

Nach dem Drücken von **CONST** erscheint "CONST" oben rechts auf dem Display.

### Hinweis:

Wenn die Konstanten-Funktion nicht verwendet wird, muß sichergestellt werden, daß die Anzeige "CONST" nicht auf dem Display erscheint.

### Überprüfung der Konstanten-Einstellung

Zur Überprüfung der zuletzt eingegebenen Konstante wird **CONST** gedrückt, während "CONST" angezeigt wird.

**2nd F** **CONST** (**SHIFT** + **CONST**)

### Löschen der letzten Konstante

Zum Löschen der zuletzt eingegebenen Konstante folgendermaßen vorgehen. Sie kann auch durch Abschalten des Gerätes gelöscht werden.

**2nd F** **CA** (**SHIFT** + **CA**)

### Beispiel:

Speichern von "+ (4,8 + 3,6)" als Konstante und Berechnung von "24 - 18,5 + (4,8 + 3,6)" und "8,2 x 6 + (4,8 + 3,6)"

Eingabe: **+** 4.8 **+** 3.6 **CONST**

Die Konstante muß nicht in Klammern geschrieben werden.

Eingabe: 24 **-** 18.5 **←** Ergebnis:13.9

Eingabe: 8.2 **X\*** 6 **←** Ergebnis: 57.6

## Die Verwendung von Variablen in Berechnungen

Der Computer kann bis zu 26 einfache numerische Variablen unter den alphabetischen Zeichen A - Z speichern. Wenn Ihnen der Begriff der Variablen nicht vertraut ist: Im Abschnitt 8 finden Sie ausführliche Erklärungen. Die Variablenzuweisung geschieht über eine Zuweisungsbefehl:

A = 5 **←**

B = -2 **←**

### Hinweis:

Zur Eingabe von "=" werden die Tasten **SHIFT** + **L** gedrückt.

Sie können auch den Wert einer Variablen (rechts) einer andern (links) zuweisen.

C = A + 3 **←**

D = C **←**

Eine Variable kann in jeder Rechenoperation anstelle einer Zahl benutzt werden. Wenn Sie nun **←** drücken, führt der Computer die Berechnung durch und zeigt den neuen Wert der Variable an. Sie können den Wert von jeder Variablen zur Anzeige bringen, indem Sie den Buchstaben eingeben, unter dem er gespeichert ist:

**C•CE** C **←**

C	8 .
---	-----

Variable behalten ihren zugewiesenen Wert auch wenn das Gerät ausgeschaltet wird oder sich mit AUTO OFF automatisch ausschaltet. Variable gehen nur in folgenden Situationen verloren.

- Zuweisung eines neuen Wertes
- Eingabe von CLEAR  $\leftarrow$  (nicht Drücken der  $\text{C}\cdot\text{CE}$  -Taste)
- Löschen der Eingaben durch Drücken des RESET-Schalters.

Es gibt gewisse Grenzen der Variablenzuweisung sowie gewisse Programmabläufe, die Variablen verändern können. Im Abschnitt 8 finden Sie ausführliche Informationen über die Zuweisung und Verwendung von Variablen beim Programmieren.

## Letztes-Ergebnis-Funktion

In einer einfachen Berechnung kann das Ergebnis der vorherigen Berechnung lediglich als erste Zahl der gegenwärtigen Berechnung verwendet werden.

**Beispiel:**

3  $\text{+}$  4  $\leftarrow$   
 $\text{X}\cdot$  5  $\leftarrow$

3 + 4	7 .
7 . * 5	35 .

Der Computer hat aber eine Funktion, die erlaubt, das Ergebnis der vorherigen Berechnung zurückzuholen und es irgendwo in der gegenwärtigen Berechnung zu benutzen. Dies ist die Funktion für das "Letzte Ergebnis." Der vorige Wert kann beliebig oft zurückgerufen werden, indem Sie die  $\text{ANS}$  -Taste drücken. Wenn Sie das letzte Beispiel eingegeben haben, so drücken Sie  $\text{C}\cdot\text{CE}$  und dann  $\text{ANS}$  , worauf die "35." angezeigt wird.

Bei dem folgenden Beispiel wird ein früheres Ergebnis zweimal in der gegenwärtigen Berechnung verwendet. Bitte beachten Sie, daß sich in diesem Beispiel jedesmal beim Drücken von  $\leftarrow$  das letzte Ergebnis verändert und auf den neuesten Stand der gegenwärtigen Berechnung gebracht wird.

**Beispiel:**

Verwende das Ergebnis (6.25) der Berechnung 50/8, um folgendes zu berechnen:

$$12 \times 5/6.25 + 24 \times 3/6.25 =$$

50  $\div$  8  $\leftarrow$

7 . * 5	35 .
50 / 8	6.25

Letztes Ergebnis

12  $\text{X}\cdot$  5  $\div$   $\text{ANS}$

50 / 8	35 .
12 * 5 / 6.25	6.25

Abruf des letzten Ergebnisses

**+** 2 4 **X\*** 3 **÷/** **ANS**

```
5 0 / 8                               3 5 .
1 2 * 5 / 6 . 2 5 + 2 4 * 3 / 6 . 2 5 6 . 2 5
_
```

**←**

```
5 0 / 8                               6 . 2 5
1 2 * 5 / 6 . 2 5 + 2 4 * 3 / 6 . 2 5
2 1 . 1 2
```

**C•CE** **ANS**

```
2 1 . 1 2 _
```

Durch Druck auf **←** wird das vorige "Letzte Ergebnis" durch das Ergebnis der vorangegangenen Rechnung ersetzt. Das letzte Ergebnis wird durch das Betätigen von **C•CE** oder **SHIFT** + **CA** jedoch nicht gelöscht, sondern nur, wenn die Stromversorgung abgeschaltet wird.

Das letzte Ergebnis kann nur abgerufen werden, wenn sich der Computer im RUN-Modus befindet. Bei Ausführung eines Programms wird das letzte Ergebnis ersetzt.

## Maximale Berechnungslänge

Die Länge eines zu berechnenden Ausdrucks ist auf max. 255 Tastenanschläge vor dem Drücken von **←** begrenzt. Wenn versucht wird, diese Grenze zu überschreiten, beginnt der Cursor zu blinken und zeigt damit an, daß weitere Eingaben nicht mehr möglich sind. In diesem Fall muß die Berechnung in zwei oder drei Abschnitte aufgeteilt werden.

## Wissenschaftliche Berechnungen

Der Computer hat zahlreiche numerische Funktionen für wissenschaftliche Berechnungen. Kapitel 5 enthält eine alphabetische Auflistung dieser Funktionen. Beachten Sie, daß die Notation der Funktionen in BASIC sich von den herkömmlichen mathematischen Notationen unterscheiden kann.

Alle wissenschaftlichen Funktionen können in der RUN-Betriebsart entweder durch Drücken der entsprechenden Funktionstaste oder durch Eingabe des BASIC-Befehls eingegeben werden.

Der Computer ermöglicht die Bestimmung der Winkleinheiten Altgrad, Bogenmaß und Neugrad mit den Anweisungen DEGREE, RADIAN bzw. GRAD.

Winkleinheit	Befehl	Erklärung
Altgrad	DEGREE	Rechter Winkel = 90[°].
Bogenmaß	RADIAN	Rechter Winkel = $\pi/2$ [rad].
Neugrad	GRAD	Rechter Winkel = 100[g].

Verwenden Sie diese Befehle in den folgenden Beispielen zur Spezifizierung der Winkleinheiten:

**Beispiel:**  $\sin 30^\circ =$

**Eingabe:**

DEGREE  (spezifiziert "Altgrad" als Winkleinheit)

SIN 30

oder

sin 30

DEGREE	
SIN 30	0.5

**Beispiel:**  $\tan \pi/4 =$

**Eingabe:**

RADIAN  (spezifiziert "Bogenmaß" als Winkleinheit)

TAN ( ( PI  / 4  )

RADIAN	0.5
TAN ( PI / 4 )	1.

**Beispiel:**  $\cos^{-1}(-0.5) =$

**Eingabe:**

DEGREE  (spezifiziert "Altgrad" als Winkleinheit)

ACS  - 0.5

DEGREE	1.
ACS -0.5	120.

**Beispiel:**  $\log 5 + \ln 5 =$

**Eingabe:**

LOG 5  + LN 5

ACS -0.5	120.
LOG 5+ LN 5	2.308407917

**Beispiel:**  $e^{2+3} =$

**Eingabe:**

EXP ( ( 2  + 3  )

LOG 5+ LN 5	2.308407917
EXP (2+3)	148.4131591

**Beispiel:**  $\sqrt{4^3 + 6^4} =$

**Eingabe:**

C·CE SQR ( ( 4  <sup>y<sup>x</sup></sup> 3  +

6  <sup>y<sup>x</sup></sup> 4  )

SQR (4^3+6^4)	36.87817783
---------------	-------------

**Beispiel:**

Umwandlung von 30 Grad 30 Minuten in DMS-Darstellung in Dezimaldarstellung.

**Eingabe:**

DEG 30.30

SQR (4^3+6^4)	36.87817783
DEG 30.30	30.5

(30.5 ALTGRAD)

**Beispiel:**

Umwandlung von 30.755 Grad in Dezimaldarstellung in DMS-Darstellung.

**Eingabe:**

DMS 30.755

DEG 30.30	30.5
DMS 30.755	30.4518

(30 ALTG. 45 MIN. 18 SEK.)

**Beispiel:**

Umwandlung von rechtwinkligen in Polarkoordinaten: Bestimmung der Polarkoordinate (r, θ) für Punkt (3, 8) einer rechtwinkligen Koordinate:

**Eingabe:**

DEGREE  (spezifiziert "Altgrad" als Winkleinheit)

POL (  3,  8 )

DEGREE	30.4518
POL (3, 8)	8.544003745

Z

POL (3, 8)	8.544003745
Z	69.44395478

Der Wert von θ wird in der Variablen Z und der Wert von r in der Variablen Y gespeichert.

**Beispiel:**

Umwandlung von Polar- in rechtwinklige Koordinaten: Bestimmung der rechtwinkligen Koordinaten (x, y) für Punkt (12, 4π/5) einer Polarkoordinate.

**Eingabe:**

RADIAN  (spezifiziert "Bogenmaß" als Winkleinheit)

REC (  12, (  4  5  ) )

PI ( ) )

RADIAN	
REC (12, (4/5*PI))	-9.708203933

Z

REC (12, (4/5*PI))	-9.708203933
Z	7.053423028

Die Werte von y und x werden in den Variablen Z bzw. Y gespeichert.

**Hinweis:**

Bei Koordinatenumwandlungen werden die Ergebnisse in den Variablen Z und Y gespeichert; dabei werden die vorigen Inhalten von Z und Y gelöscht.

**Beispiel:**

Umwandlung von CF8 in eine Dezimalzahl.

**Eingabe:**

**C•CE** **2nd F** **&** H C F 8 

& H C F 8	3 3 2 0 .
-----------	-----------

"&H" bezeichnet einen hexadezimalen Wert.

Ausdrücke, die Verhältniszeichen (=, >, <, >=, <=, <>) enthalten, können die in der folgenden Tabelle angeführten Werte annehmen: x und y repräsentieren numerische Werte.

=*	-1 wenn $x = y$ 0 wenn $x \neq y$	>=	-1 wenn $x \geq y$ 0 wenn $x < y$
>	-1 wenn $x > y$ 0 wenn $x \leq y$	<=	-1 wenn $x \leq y$ 0 wenn $x > y$
<	-1 wenn $x < y$ 0 wenn $x \geq y$	<>	-1 wenn $x \neq y$ 0 wenn $x = y$ <span style="font-size: 1.2em;">( "<math>&lt;&gt;</math>" bedeutet "<math>\neq</math>" )</span>

- \* Wenn beispielsweise "A = numerischer Wert" oder "B = Formel" in einer logischen Gleichung verwendet wird, behandelt der Computer dies nicht als logische Gleichung, sondern als Zuweisungs-Anweisung für Variable. Das Gleich-Zeichen (=) für eine logische Gleichung muß daher in der Form "numerischer Wert = A" oder "Formel = B" mit der Ausnahme von Bedingungsaustrücken in IF-Anweisungen verwendet werden.

**Hinweis:**

Die Symbole ">", "<" und "!" können nicht als Verhältnis-Operatoren verwendet werden.

**Direkt-Berechnungsfunktion**

Bei den bisher beschriebenen manuellen Berechnungen wurde immer die Taste  zur Beendigung eines Ausdrucks und Berechnung des Ergebnisses dieses Ausdrucks verwendet. Es ist jedoch auch möglich, die Funktionen des Computers direkt mit der entsprechenden Funktionstaste (ohne Betätigung der Taste ) zu verwenden, wenn die numerischen Daten für die Funktion auf der Anzeige vorhanden sind.

**Beispiel:**Bestimmung von  $\sin 30^\circ$  und  $8!$ .**Eingabe:**DEGREE  $\leftarrow$   
C•CE 3 0 sin

3 0	0 . 5
-----	-------

**Eingabe:**

C•CE 8 n!

8	4 0 3 2 0 .
---	-------------

**Beispiel:**Für  $\tan^{-1} \frac{5}{12}$ , soll zuerst das Ergebnis von  $\frac{5}{12}$  und dann  $\tan^{-1} \frac{5}{12}$  berechnet werden.**Eingabe:**DEGREE  $\leftarrow$   
5  $\div$  12  $\leftarrow$  2nd F tan<sup>-1</sup>

DEGREE	
5 / 1 2	0 . 4 1 6 6 6 6 6 6 6
	2 2 . 6 1 9 8 6 4 9 5

Es muß jedoch beachtet werden, daß diese Direkt-Berechnungsmethode nicht für Funktionen verfügbar ist, die mehr als einen numerischen Wert (binomiale Funktionen) benötigen, beispielsweise Potenzen, Potenzwurzeln oder Koordinatenumwandlungen. Die Direkt-Berechnungsfunktion ist nur für numerische Werte wirksam. Wenn daher die Hexadezimalzahlen A bis F für Hexadezimal-Dezimal-Umwandlung eingegeben wurden, ist die Direkt-Berechnungsfunktion unwirksam. Führen Sie in solchen Fällen normale manuelle Berechnungen mit der Taste  $\leftarrow$  durch.

Die Direkt-Berechnungsfunktion ist nicht wirksam für Formeln.

**Beispiel:**C•CE 5 X\* 4  $\rightarrow$  5 \* 4  
log  $\rightarrow$  5 \* 4LOG

Wenn keine Daten auf dem Display angezeigt werden, erscheint beim Drücken der Funktionstaste der entsprechende BASIC-Befehl.

## Vorrangordnung bei manuellen Berechnungen

Sie können Ausdrücke in derselben Reihenfolge eingeben, in der sie geschrieben sind, einschließlich der Klammerung und der Funktionen. Die interne Vorrangordnung bei der Berechnung und Behandlung von Zwischenergebnissen wird vom Computer selbst kontrolliert.

Die interne Vorrangordnung bei manuellen Berechnungen ist die folgende:

1. Abruf von Variablen oder PI
2. Funktionen (sin, cos, usw.)
3. Exponenten ( $\wedge$ ), Wurzeln (ROT)

4. Vorzeichen (+, -)
5. Multiplikation und Division (\*, /)
6. Addition und Subtraktion (+, -)
7. Größenvergleich (>, >=, <, <=, <>, =)
8. Logisches AND, OR

**Hinweise:**

- Werden in einem Ausdruck Klammern verwendet, hat die innerhalb der Klammer angegebene Berechnung höchste Priorität.
- Zusammengesetzte Funktionen werden von rechts nach links abgearbeitet (sin cos<sup>-1</sup> 0.6).
- Mehrstufige Exponenten (3<sup>4</sup> oder 3 ^ 4 ^ 2) werden von rechts nach links abgearbeitet.
- Bei den obigen Punkten 3. und 4. hat die letzte Eingabe die höhere Priorität.

**Beispiel:**

$$-2 \wedge 4 \rightarrow -(2^4)$$

$$3 \wedge -2 \rightarrow 3^{-2}$$

## Ausdrucken von Rechenergebnissen

Berechnungen und Ergebnisse können durch Betätigen von **SHIFT** + **P-NP** ausgedruckt werden, wenn der optionale Drucker angeschlossen und eingeschaltet ist. Beachten Sie, daß Berechnungen im CAL-Modus nicht ausgedruckt werden können.

Wenn kein Ausdruck gewünscht wird, schalten Sie entweder den Drucker aus oder drücken Sie **SHIFT** + **P-NP** erneut (NON PRINT-Modus).

## Rechenfehler

Folgende Fehlertypen treten bei gewöhnlichen Rechnern, Taschen- und Personal Computern auf:

### Fehler, die mit der Anzahl der Dezimalstellen zusammenhängen

Normalerweise ist die maximale Anzahl der Dezimalstellen, mit denen ein Computer rechnen kann, festgelegt. Zum Beispiel 4/3 ergibt 1.333333333... Bei einem Computer mit einem Maximum von 8 Stellen werden die ersten 8 Ziffern als signifikant bezeichnet, weitere Ziffern werden entweder ignoriert oder gerundet.

**Beispiel:**

Computer mit 10 signifikanten Stellen:

10 signifikante Ziffern

$$4 \left[ \div \right] 3 \left[ \leftarrow \right] \rightarrow \overbrace{1.3333333333}^{10 \text{ signifikante Ziffern}} \dots$$

ignoriert, gerundet

Aus diesem Grund weicht das errechnete Ergebnis von dem tatsächlichen Ergebnis durch den ignorierten oder gerundeten Betrag ab. (Dieser Unterschied macht dann die Abweichung aus.)

**Beispiel:**  $4/3 \times 3$

4  $\boxed{\div}$  3  $\boxed{\times}$  3  $\boxed{\leftarrow}$   $\rightarrow$  4

4  $\boxed{\div}$  3  $\boxed{\leftarrow}$   $\rightarrow$  1.333333333 }  
 $\boxed{\times}$  3  $\boxed{\leftarrow}$   $\rightarrow$  3.999999999 }

wird sukzessive verarbeitet.

wird unabhängig verarbeitet.

Wenn die Berechnung sukzessive ausgeführt wird, bleibt der Berechnungsfehler klein.

Wenn der Rechner unabhängig arbeitet, wird der angezeigte Wert (10 Kommastellen) für die Berechnung benutzt.

### **Fehler bei der Bestimmung von Algorithmen**

Der Computer benutzt eine Reihe von Algorithmen, um die Werte von Funktionen zu berechnen, wie z.B. Potenz- und trigonometrische Funktionen. Wenn solche Funktionen in Rechnungen eingesetzt werden, summiert sich die Fehlerquelle. Dieser Fehlerfaktor nimmt zu, je öfter diese Funktionen in die Rechnung eingesetzt werden. Der eigentliche Fehler für jede einzelne Funktion hängt von den eingesetzten Werten ab und ist am größten bei singulären Stellen und Wendepunkten (d.h. wenn ein Winkel sich 90 Grad annähert, nähert sich die Tangente der Unendlichkeit).

# PROGRAMMIEREN

Kapitel 3 ist der Anwendung der BASIC-Programmiersprache für den PC-E220\* gewidmet. Wir beginnen mit einer generellen Diskussion von Programmierkonzepten und kommen dann zu einer mehr speziellen Anwendung dieser Konzepte beim PC-E220. Kapitel 3 endet mit einigen Abkürzungsvorschlägen zum Programmieren und mit Hinweisen zum Auffinden von Fehlern in Ihren Programmen.

PRO(Programm)- und RUN-Modus (der detailliert im Kapitel 2 beschrieben wurde) werden in diesem Abschnitt behandelt.

\* Der PC-E220 wird im folgenden als "der Computer" bezeichnet.

# 8. BEGRIFFE UND AUSDRÜCKE DES BASIC

In diesem Kapitel wollen wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennenlernen.

## Zeichenfolgen-Konstanten

Der Computer ist in der Lage, außer Zahlen auch Buchstaben und spezielle Symbole in vielfacher Weise zu verarbeiten. Diese Buchstaben, Zahlen und speziellen Symbole werden Zeichen genannt.

Im BASIC wird eine Folge von Zeichen als Zeichenfolge bezeichnet. Damit der Computer den Unterschied zwischen einer Zeichenfolge und andern Programmteilen, wie z.B. Befehlen oder Variablen-Bezeichnungen, erkennen kann, muß man die Zeichen, die zu einer Zeichenfolge gehören, in Anführungszeichen (") einschließen. Zur Verwendung von Anführungszeichen als Zeichen wird "CHR\$&H22" eingegeben.

Es folgen einige Beispiele für Zeichenfolgen-Konstanten:

```
"HELLO"  
"Goodbye"  
"SHARP COMPUTER"
```

Die folgenden Beispiele werden nicht als Zeichenfolgen-Konstanten akzeptiert:

```
"COMPUTER           Anführungszeichen am Ende fehlen.  
"VALUE OF "A" IS"  Anführungszeichen dürfen nicht innerhalb einer  
Zeichenfolge benutzt werden.
```

## Hexadezimalzahlen

Das Dezimalsystem ist nur eines von verschiedenen Zahlensystemen. Ein anders, dessen Bedeutung im Zusammenhang mit Computern stark zugenommen hat, ist das Hexadezimalsystem. Das Hexadezimalsystem basiert auf der Zahl 16 statt auf der Zahl 10. Um hexadezimale Ziffern zu schreiben, benutzt man die Ziffern 0 bis 9, sowie sechs weitere "Ziffern" A, B, C, D, E und F. Diese entsprechen den Zahlen 10, 11, 12, 13, 14 und 15. Wenn der Computer eine Zahl als hexadezimal auffassen soll, setzen Sie ein UND-Zeichen (&) und "H" vor die Zahl:

```
&HA    = 10  
&H10   = 16  
&H100  = 256  
&HFFFF = 65535
```

## Variablen

Computer sind aus einer Vielzahl von kleinsten Speichereinheiten aufgebaut, genannt Bytes. Jedes Byte kann man sich als einzelnes Zeichen vorstellen. Das Wort "Byte" erfordert beispielsweise vier Speicherbytes, weil es vier Buchstaben enthält. Um herauszufinden, wieviele Bytes zum Arbeiten zur Verfügung stehen,

geben Sie einfach im RUN-Modus FRE ein und drücken . Die angezeigte Zahl gibt an, wieviel Bytes zum Programmieren frei sind.

Dieses Verfahren funktioniert gut für Worte, ist aber zum Speichern von Zahlen sehr unzulänglich. Aus diesem Grund werden Zahlen in codierter Form gespeichert. Aufgrund dieser Codierung ist der Computer in der Lage, auch lange Zahlen in nur 8 Bytes zu speichern. Die größtmögliche Zahl, die gespeichert werden kann, ist +9.99999999E + 99. Die kleinste Zahl ist 1.E -99. So erhalten Sie einen recht großen Zahlenbereich zum Arbeiten. Wenn jedoch das Ergebnis der Rechnung diesen Rahmen übersteigt, teilt der Computer dies mit, indem er eine Fehlermeldung auf dem Display ausgibt. (Siehe Anhang B.) Geben Sie nun zur Probe ein:

9 + 9 9 9

```
9E99*9
ERROR 20
```

Damit der Computer wieder normal weiterarbeiten kann, brauchen Sie lediglich die -Taste zu drücken. Aber wie speichert man nun soviel Information? Das ist ganz einfach. Der Computer zieht es vor, verschiedenen Datengruppen Namen zu geben. Wir wollen die Zahl 556 abspeichern. Sie können diese Zahl benennen, wie Sie wollen, aber für dieses Beispiel möchten wir ihr den Namen R geben. Die Anweisung LET kann benutzt werden, um den Computer zu veranlassen, einer Variablen einen Wert zuzuweisen, allerdings nur in einem Programmbefehl. Der LET-Befehl ist aber nicht unbedingt erforderlich, darum werden wir ihn nicht sehr oft benutzen.

Geben Sie nur R = 556 ein und drücken Sie .

Der Computer hat nun den Wert 556 mit dem Buchstaben R in Verbindung gebracht. Diese Buchstaben, die man benutzt, um Informationen zu speichern, nennt man Variable. Zum Abfragen des Inhalts der Variablen R drücken Sie die -Taste, und die -Taste. Der Computer antwortet, indem er den Wert 556 rechts im Display anzeigt. Diese Möglichkeit kann sehr nützlich sein, wenn man Programme oder Formeln eingibt.

Wir wollen als nächstes die Variable R in einer einfachen Formel verwenden. In dieser Gleichung steht R für den Radius eines Kreises, dessen Fläche wir berechnen wollen. Die Gleichung der Kreisfläche lautet:  $A = \pi R^2$ . Geben Sie ein:

R 2   
 +

```
R=556
                                     556.
R^2* PI
                                     971179.3866
```

Das Ergebnis ist 971179.3866.

Dieses Verfahren, Variablen zu verwenden, wird eingehender erklärt, wenn wir uns mit dem Programmieren beschäftigen.

Bislang haben wir uns nur mit numerischen Variablen befaßt. Wie speichert man nun alphabetische Zeichen? Grundsätzlich ist das Prinzip das gleiche, aber damit der Computer den Unterschied zwischen den beiden Variablentypen erkennen kann, muß nun ein \$ zum Namen der Variablen gesetzt werden. Wir wollen z.B. das Wort BYTE unter der Variablen B\$ speichern. Beachten Sie das \$-Zeichen hinter dem B.

Dies sagt dem Computer, daß der Inhalt der Variablen B\$ alphanumerisch bzw. eine Zeichenfolge ist. Damit dies klarer wird, geben Sie ein:

B  +  =  +    
 B Y T E  +

```
R^2 * PI
971179.3866
B$ = "BYTE"
BYTE
```

Die Zeichenfolge BYTE ist nun unter der Variablen B\$ gespeichert. Um sich zu vergewissern, betätigen Sie  und tippen dann B\$ .

B  +

```
B$
BYTE
```

### Arten von Variablen

Die Variablen, mit denen der Computer arbeitet, sind folgendermaßen aufgegliedert:

Numerische Variablen:

- Feste numerische Variablen (A bis Z)
- Einfache numerische Variablen (AB, C1, usw.)

Numerische Feldvariablen

Zeichenfolge-Variablen:

- Feste Zeichenfolge-Variablen (A\$ bis Z\$)
- Einfache Zeichenfolge-Variable (BB\$, C2\$, etc.)
- Zeichenfolge-Feldvariablen

### Feste Variable

Bei der ersten Art, den festen Variablen, handelt es sich um Variable mit bereits zugewiesenem Speicherbereich. Mit andern Worten, egal, wieviel Speicherplatz das Programm in Anspruch nimmt, Ihnen stehen immer mindestens 26 Variable zur Speicherung von Daten offen. Es handelt sich dabei um zwei Arten von Variablen: numerische oder Zeichenfolge-Variable (alphanumerische Zeichen). Die zugewiesenen Speicherbereiche haben eine Kapazität von acht Bytes und können jeweils nur für eine Art von Variablen verwendet werden.

Beispiel:

A = 123    
 A\$

Die folgende Meldung wird ausgegeben:

ERROR 91

Dies bedeute, daß numerische Daten in einen als "A" bezeichneten Speicherbereich zugewiesen wurden und danach der Computer die Anweisung erhielt, diese Informationen wieder als Zeichenfolge auszugeben. Der Computer ist dadurch verwirrt und gibt eine Fehlermeldung aus.  drücken, um die Fehlermeldung zu löschen. Nun wird folgendes eingegeben:

A\$ = "ABC"    
 A

Wieder ist der Computer verwirrt und gibt die Fehlermeldung 91 aus. Die Variablenbezeichnung A entspricht im Speicher dem gleichen Bereich wie die Variablenbezeichnung A\$, weiterhin entspricht B dem Bereich B\$ und so weiter für alle Buchstaben des Alphabetes.

## Einfache Variablen

Einfache Variablenbezeichnungen werden durch alphanumerische Zeichen charakterisiert, z.B. AB oder C8\$. Anders als feste Variablen haben die einfachen Variablen keinen fest im Speicher reservierten Bereich. Der Speicherbereich für einfache Variablen wird automatisch bereitgestellt (im Programm oder im Datenbereich), sobald eine einfache Variable erstmalig benutzt wird.

Da für einfache numerische und einfache Zeichenfolge-Variablen verschiedene Speicherbereiche vorgesehen sind, können Variablen mit dem gleichen Namen, z.B. AB und AB\$ gleichzeitig benutzt werden.

Für die Bezeichnungen von einfachen Variablen können alphanumerische Zeichen verwendet werden; das erste Zeichen muß allerdings immer ein Großbuchstabe sein. Zur Bestimmung eines Variablennamens können zwei oder mehr Zeichen verwendet werden, der Computer liest allerdings nur die ersten beiden.

### Hinweise:

- Die im Computer residenten Bezeichnungen für Funktionen und BASIC-Befehle, z.B. PI, IF, TO, ON, SIN u.a., können nicht für Variablenbezeichnungen verwendet werden.
- Jede einfache Zeichenvariable kann bis zu 16 Zeichen und Symbole enthalten. Jede feste Zeichenvariable kann bis zu 7 Zeichen und Symbole enthalten.

## Feldvariablen

In einigen Fällen ist es sinnvoll, Zahlen in organisierten Gruppen zu verarbeiten, z.B. eine Tabelle der Fußballergebnisse oder eine Steuertabelle. Im BASIC werden solche Gruppen Felder genannt. Ein Feld kann eindimensional sein, z.B. eine Liste, es kann aber auch zweidimensional sein, z.B. eine Tabelle.

Um ein Feld zu definieren, benutzt man den DIM-Befehl (Kürzel für Dimension). Felder müssen vor Gebrauch immer definiert werden. (Dies war nicht der Fall bei den Einwert-Variablen, die wir bislang benutzt haben.) Die Form für die DIMensionierung numerischer Felder ist:

**DIM** Name der Feldvariable (Größe)

Dabei bedeutet:

**Name der Feldvariable** ist eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für numerische oder Feld-Variablen.

**Größe** bedeutet die Anzahl der Speicherplätze und sollte eine Zahl im Bereich von 0 bis 255 sein. Bei der Zuweisung einer Zahl für die Größe wird ein Speicherplatz mehr als zugewiesen bereitgestellt.

Beispiel für zugelassene Befehle für numerische oder Zeichenfolge-Dimensionierung:

```
DIM X(5)           → X(0), X(1), X(2), X(3), X(4), X(5)
DIM AA(24)
DIM Q5(0)
```

Der erste Befehl schafft ein Feld X mit 6 Speicherplätzen. Der zweite Befehl baut ein Feld AA mit 25 Speicherplätzen auf, der dritte ein Feld mit einem Speicherplatz, was unsinnig (zumindest für Zahlen) ist, da man ebensogut eine einwertige numerische Variable definieren könnte.

Es ist wichtig zu wissen, daß eine Feldvariable X und eine Variable X vom Computer unterschieden werden. Das erste X bezeichnet eine Serie von numerischen Speicherplätzen, das zweite einen einzelnen und getrennten Speicherplatz.

Nachdem Sie nun wissen, wie man Felder aufbaut, mögen Sie sich fragen, wie man die einzelnen Speicherplätze anspricht. Da die gesamte Gruppe unter einem einzigen Namen abgelegt ist, sprechen wir einen einzelnen Speicherplatz ("Element" genannt) an, indem wir an den Namen der Gruppe eine Zahl in Klammern anschließen. Diese Zahl wird "Index" genannt. So müßte man z.B., um die Zahl 8 an fünfter Stelle in unserem (vorher definierten) Feld X unterzubringen, schreiben:

$$X(4) = 8$$

Wenn Sie die Zahl 4 verwirrt, bedenken Sie, daß die Numerierung der Elemente in einem Feld mit Null beginnt und dann bis zu der in der DIM-Anweisung definierten Anzahl von Elementen fortläuft.

Der besondere Vorteil von Feldern liegt in der Möglichkeit, einen längeren Ausdruck oder eine Variable als Index zu benutzen.

Zur Bestimmung einer Zeichenfolge-Feldvariable wird eine etwas andere Form der DIM-Anweisung verwendet:

**DIM** Name der Zeichenfolge-Variable (Größe) \*Länge

Dabei bedeutet:

**Name der Zeichenfolge-Variable** ist eine Bezeichnung für die Variable, die den bereits besprochenen Regeln für Bezeichnungen entspricht.

**Größe** bedeutet die Anzahl der Speicherplätze und sollte eine Zahl im Bereich von 0 bis 255 sein. Bei der Zuweisung einer Zahl für die Größe wird ein Speicherplatz mehr als zugewiesen bereitgestellt.

**\*Länge** ist optional. Falls diese Option verwendet wird, wird damit die Länge für jede Zeichenfolge dieser Feldvariablen bestimmt. Die Länge muß durch eine Zahl von 1 bis 255 angegeben werden. Falls hier keine Eingabe erfolgt, wird die Grundeinstellung von 16 Zeichen für die Zeichenfolge angenommen.

Beispiele für zulässige Bestimmungen von Zeichenfolge-Feldvariablen:

```
DIM X$(4)
DIM NM$(10)*10
DIM IN$(1)*255
DIM R$(0)*26
```

Beim ersten Beispiel wird ein Feld von fünf Zeichenfolgen geschaffen, in denen jeweils 16 Zeichen gespeichert werden können. Beim zweiten Beispiel wird durch die DIM-Anweisung ein Feld NM bestimmt, das 11 Zeichenfolgen mit jeweils 10 Zeichen enthält. Die eindeutige Zuweisung von Zeichenfolgen, die kürzer als die Grundeinstellung von 16 Zeichen sind, hilft bei der Einsparung von Speicherplatz. Beim dritten Beispiel wird ein Zwei-Elemente-Feld mit 255 Zeichenfolgen bestimmt und beim letzten Beispiel eine einzelne Zeichenfolge mit 26 Zeichen.

Neben den einfachen Feldvariablen, die bisher besprochen wurden, kann der Computer auch "zweidimensionale" Felder bearbeiten. Bei einem eindimensionalen Feld wird eine Folge von Daten in einer einzigen Spalte aufgelistet. Ein zweidimensionales Feld entspricht einer Tabelle mit Zeilen und Spalten. Zweidimensionale Felder werden durch die folgende Anweisung bestimmt:

**DIM** Bezeichnung des numerischen Feldes (Zeilen, Spalten)

oder

**DIM** Bezeichnung der Zeichenfolge-Feldvariable (Zeilen, Spalten) \*Länge

Dabei bedeutet:

**Zeilen** bedeutet die Anzahl der Zeilen in einem Feld. Es muß eine Zahl im Bereich von 0 bis 255 sein. Bei der Zuweisung einer Zahl für die Zeilen wird eine Zeile mehr als zugewiesen bereitgestellt.

**Spalten** bedeutet die Anzahl der Spalten in einem Feld. Es muß eine Zahl im Bereich von 0 bis 255 sein. Bei der Zuweisung einer Zahl für die Spalten wird eine Spalte mehr als zugewiesen bereitgestellt.

Die folgende Tabelle illustriert die Speicherplätze, die sich aus der Anweisung DIM T (2, 3) und den Indizes, die zu den jeweiligen Speicherplätzen gehören, ergeben (in diesem Beispiel zwei Zahlen):

	Spalte 0	Spalte 1	Spalte 2	Spalte 3
Zeile 0	T (0, 0)	T (0, 1)	T (0, 2)	T (0, 3)
Zeile 1	T (1, 0)	T (1, 1)	T (1, 2)	T (1, 3)
Zeile 2	T (2, 0)	T (2, 1)	T (2, 2)	T (2, 3)

**Hinweis:**

Zweidimensionale Felder nehmen viel Speicherplatz in Anspruch. Z.B. benötigt ein Feld mit 25 Zeilen und 35 Spalten 875 Speicherplätze!

Die folgende Tabelle zeigt die Anzahl der Bytes, die zur Definierung jeder einzelnen Variable benötigt werden, sowie die Anzahl der für jeden einzelnen Programm-Befehl erforderlichen Bytes.

Variablen-Typ	Anzahl der verwendeten Byte	
	Variablen-Name	Daten
Numerische Variable Numerische Feldvariable	7 Byte	8 Byte
Zeichenfolge-Variablen	7 Byte	16 Byte
Zeichenfolge-Feldvariable	7 Byte	Zugewiesene Anzahl

\* Wenn zum Beispiel DIM Z\$(2,3)\*10 ist, werden 12 Variable mit einem Speicherplatz für jeweils 10 Zeichen bereitgestellt. Dafür werden 127 Byte benötigt: 7 Byte (Variablenname) + 10 Byte (Anzahl der Zeichen) x 12.

Element	Zeilennummer	Befehl & Funktion	 andere
Anzahl der verwendeten Byte	3 Byte	2 Byte	1 Byte

## Programm-Dateien (RAM-Diskette)

Programm-Dateien stellen den wichtigsten Aspekt beim Gebrauch des Computers dar. Ein Teil des internen Speichers des Computers kann als RAM-Diskette verwendet werden. Auf die RAM-Diskette können nur Programme gespeichert werden, keine Daten. Auf RAM-Diskette gespeicherte Programme müssen vor der Ausführung in den Programmdatei-Bereich (Benutzerbereich) geladen werden. (Siehe BASIC KOMMANDO LEXIKON für Hinweise zu den Befehlen SAVE, LOAD, KILL und FILES.)

## Dateinamen

Vor dem Speichern auf ein anderes Speichermedium, z.B. eine RAM-Diskette, muß die Datei einen Namen bekommen. Dieser Name wird benutzt zum Laden der Programm-Datei in den Computerspeicher. Der Dateiname ist beliebig und kann aus bis zu 8 der folgenden Zeichen bestehen:

A - Z, a - z, 0 - 9, #, \$, %, &, ' , ( , ) , { , } , -, @

## Erweiterung

Die Erweiterung ist eine zusätzliche Art, Datei-Typen (wie z.B. BASIC-Programm-Dateien oder Text-Dateien) zu identifizieren. Die Erweiterung besteht aus drei Zeichen, die am Ende des Dateinamens durch einen Punkt getrennt angehängt werden. Die Erweiterung wird spezifiziert, wenn eine Datei gesichert wird.

BASIC-Programme erhalten automatisch die Erweiterung .BAS, wenn sie mit dem SAVE-Kommando gesichert werden. Beim Laden in den Speicher mit dem LOAD-Kommando muß die .BAS-Erweiterung nicht spezifiziert werden.

Wird mit den Anweisungen FILES oder LFILES ein Verzeichnis der Dateien in der RAM-Diskette aufgelistet, erscheinen die BASIC-Programme mit der .BAS-Erweiterung, unabhängig davon, ob irgendeine andere Erweiterung bei der Sicherung der Datei bestimmt wurde.

## Ausdrücke

Ein Ausdruck ist eine Kombination von Variablen, Konstanten und Operatoren, die auf einen einzigen Wert zusammengefaßt werden. Die Rechenbeispiele, die Sie vorher eingegeben haben, waren Beispiele für solche Ausdrücke. Ausdrücke sind ein wesentlicher Bestandteil von BASIC-Programmen. Zum Beispiel kann ein Ausdruck eine Formel sein, mit der das Ergebnis einer Gleichung errechnet wird, oder ein Test zur Bestimmung des Verhältnisses zwischen zwei Größen oder ein Mittel, um eine Reihe von Zeichenfolgen zu formatieren.

## Numerische Ausdrücke

Ein numerischer Ausdruck wird in der gleichen Weise konstruiert, wie Sie komplexe Rechenbefehle eingegeben haben. Numerische Ausdrücke können jede aussagefähige Kombination von numerischen Konstanten, numerischen Variablen und numerischen Operatoren beinhalten. Es gibt folgende numerische Operatoren:

- + Addition
- Subtraktion
- \* Multiplikation
- / Division
- ^ Exponentiation

Dies sind die arithmetischen Rechenzeichen, die Sie verwendet haben, als Sie die Benutzung des Computers als Taschenrechner im Abschnitt 4 kennenlernten. Beispiele für zulässige numerische Ausdrücke:

$$(A * B) \wedge 2$$
$$A(2,3) + A(3,4) + 5.0 - C$$
$$(A/B) * (C + D)$$

## Zeichenfolge-Ausdrücke

Zeichenfolge-Ausdrücke sind den numerischen Ausdrücken ähnlich, allerdings gibt es nur einen einzigen Zeichenfolgen-Operator: die Verkettung (+). Das benutzte Symbol ist dasselbe wie das Pluszeichen. Wird es mit einem Zeichenfolge-Paar benutzt, knüpft das + die zweite Zeichenfolge an das Ende der ersten an und schafft dadurch eine längere Zeichenfolge. Wenn Sie eine komplexere Zeichenfolgen-Verkettung und andere Zeichenfolgen-Operationen vernehmen, bedenken Sie bitte, daß der Computer nur 255 Zeichen annimmt (Siehe Seite 173).

### Hinweis:

Zeichenfolge-Einheiten und numerische Einheiten können nicht in demselben Ausdruck definiert werden, es sei denn, man benutzt eine der Funktionen, die Zeichenfolge-Werte in numerische Werte umwandeln oder umgekehrt:

"15" + 10 ist unzulässig.  
"15" + "10" ist "1510", nicht "25".

## Verhältnis-Ausdrücke

Ein Verhältnis-Ausdruck vergleicht zwei Ausdrücke und gibt an, ob das festgestellte Verhältnis wahr oder unwahr ist. Die Verhältnis-Operatoren sind:

- > größer als
- > = größer oder gleich
- = gleich
- < > ungleich
- < = kleiner oder gleich
- < kleiner als

Die folgenden Ausdrücke werden als Verhältnis-Ausdrücke akzeptiert:

A < B  
C(1,2) > = 5  
D(3) < > 8

Wenn A gleich 10 wäre, B gleich 12, C(1,2) gleich 6 und D(3) gleich 9, wären alle diese Ausdrücke wahr.

Zeichenfolgen können ebenfalls mit Hilfe von Verhältnis-Ausdrücken verglichen werden. Die beiden Zeichenfolgen werden Zeichen für Zeichen gemäß dem Wert ihres ASCII-Codes verglichen (siehe Anhang C). Ist eine Zeichenfolge kürzer als eine andere, wird 0 oder NULL in die freibleibenden Positionen eingesetzt. Alle folgenden Beispiele sind wahr:

"ABCDEF" = "ABCDEF"  
"ABCDEF" < > "ABCDE"  
"ABCDEF" > "ABCDE"

Verhältnis-Ausdrücke beurteilen nach wahr oder unwahr. Beim Computer wird wahr durch -1 angegeben, unwahr durch eine 0.

## Logische Ausdrücke

Logische Operationen benutzen zum Bau von Verbindungen zwischen Verhältnis-Ausdrücken die Funktionen der Booleschen Algebra AND (und), OR (oder), und NOT (nicht). Die logischen Operationen in einem einzigen Ausdruck werden nach arithmetischen und Verhältnis-Operationen bewertet.

Auf diese Weise können logische Operatoren benutzt werden, die ihre Programmentscheidungen auf Grund mehrerer Bedingungen unter Verwendung der Anweisungen IF...THEN treffen.

### Beispiel:

```
IF A < = 32 AND B > = 90 THEN 150
```

Diese Anweisung bewirkt, daß die Ausführung auf Zeile 150 springt, wenn der Wert der numerischen Variable A kleiner oder gleich 32 ist und wenn gleichzeitig der Wert der numerischen Variablen B größer oder gleich 90 ist.

```
IF X < > 13 OR Y = 0 THEN 50
```

Diese Anweisung bewirkt, daß die Ausführung auf Zeile 50 springt, es sei denn die Variable X hat den Wert 13, oder die Variable Y ist ungleich 0.

In einer logischen Operation, in der zwei Ziffern im Bereich -32768 bis +32767 involviert sind, werden die zwei Ziffern in 16-stellige Binärzahl umgewandelt (in die Zweierkomplementform) und die logische Verbindung wird dann für jedes entsprechende Bit-Paar in den zwei Ziffern bewertet.

Die Ergebnisse, angezeigt durch die logischen Operatoren für diese Bit-Bewertungen, werden im folgenden aufgelistet:

AND			OR			NOT	
X	Y	X AND Y	X	Y	X OR Y	X	NOT X
1	1	1	1	1	1	1	0
1	0	0	1	0	1	0	1
0	1	0	0	1	1		
0	0	0	0	0	0		

Nachdem nach jedem Bit-Paar das entsprechende Ergebnis (eine 1 oder eine 0) entsprechend obiger Tabelle zugewiesen wurde, wird die sich ergebende 16-stellige Binärzahl zurück in einen Dezimalwert verwandelt. Diese Zahl ist das Ergebnis der logischen Operation.

**Beispiel:**

41 AND 27 →                      41 = 101001  
gleich                              27 = 011011    AND  
9                                      ← 001001

41 OR 27 →                        41 = 101001  
gleich                              27 = 011011    OR  
59                                    ← 111011

NOT 3 →                            3 = 0000000000000011    NOT  
gleich                              ← 1111111111111100  
-4 (Zweierkomplementform)

NOT X kann generell berechnet werden durch die Gleichung NOT X = -(X+1).

## Klammerung und Vorrang der Operatoren

Bei der Bearbeitung komplexer Ausdrücke folgt der Computer einer Reihe definierter Prioritäten, die bestimmen, in welcher Reihenfolge die Operatoren bearbeitet werden.

5 + 2 \* 3 kann sein

5 + 2 = 7    oder    2 \* 3 = 6  
7 \* 3 = 21        6 + 5 = 11

Die genauen Regeln des "Operatoren-Vorrangs" werden auf Seite 70 beschrieben. Damit Sie sich nicht alle diese Regeln merken müssen und damit Sie Ihre Programme eindeutiger gestalten, benutzen Sie immer Klammern, um die Reihenfolge der Bearbeitung von Operatoren vorzugeben. Das obige Beispiel wird eindeutig, wenn Sie schreiben:

(5 + 2) \* 3 oder 5 + (2 \* 3)

# 9. PROGRAMMIEREN

Im vergangenen Abschnitt haben wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennengelernt. In diesem Abschnitt wollen wir nun diese Elemente benutzen, um Programme zu schreiben. Wir möchten noch einmal darauf hinweisen, daß dieses Handbuch nicht als Einführung in die BASIC-Programmierung verstanden werden soll. Dieser Abschnitt soll Sie lediglich in den besonderen Gebrauch des BASIC auf dem Computer einführen.

## Programme

Ein Programm besteht aus einer Reihe von an den Computer gerichteten Befehlen. Denken Sie daran, daß der Computer nur eine Maschine ist, die genau die Operationen durchführt, die Sie angeben. Sie als Programmierer sind dafür verantwortlich, daß korrekte Befehle eingegeben werden.

## BASIC-Anweisungen

Der Computer setzt Programme entsprechend einem bestimmten Format um. Dieses Format wird Anweisung genannt. Sie geben die BASIC-Anweisungen immer nach einem bestimmten Muster ein. Anweisungen müssen mit einer Zeilennummer beginnen:

### Beispiel:

```
10: INPUT A
20: PRINT A*A
30: END
```

## Zeilennummern

Jede Programmzeile muß eine eigene Nummer haben, und zwar muß diese eine ganze Zahl zwischen 1 und 65279 sein. Zeilennummern sind die Bezugspunkte des Computers. Sie geben dem Computer an, in welcher Reihenfolge er ein Programm abarbeiten muß. Es ist nicht erforderlich, daß Sie die Programmzeilen folgerichtig eingegeben haben (obwohl das sicher weniger verwirrend sein dürfte, besonders wenn Sie noch Anfänger sind). Der Computer beginnt beim Durcharbeiten eines Programms immer mit der niedrigsten Zeilennummer und arbeitet die folgenden in ansteigender Reihenfolge ab.

Mit dem AUTO-Kommando können Sie automatisch Zeilennummern einfügen. Jedesmal, wenn Sie die -Taste drücken, wird eine neue Zeilennummer, in der richtigen aufsteigenden Reihenfolge, automatisch eingefügt. (Eine vollständige Beschreibung dieser nützlichen Funktion finden Sie im BASIC KOMMANDO LEXIKON.)

Beim Programmieren ist es sinnvoll, genug Raum für spätere Einschübe zwischen den einzelnen Zeilen zu lassen (10, 20, 30, ... 10, 30, 50, usw.). Dies ermöglicht den Einschub von zusätzlichen Zeilen, falls notwendig.

Wenn Sie gleiche Zeilennummer mehr als einmal benutzen, wird die ältere Zeile gelöscht, wenn Sie eine neue mit derselben Nummer eingeben.

## Programme mit Labels

Es wird häufiger vorkommen, daß Sie mehrere verschiedene Programme zur gleichen Zeit im Speicher abspeichern wollen. (Vergessen Sie nicht, verschiedene Zeilennummern zu vergeben.) Um ein Programm mit einem RUN- oder einem GOTO-Befehl zu starten, müssen Sie sich normalerweise an die erste Zeilennummer eines jeden Programms erinnern. Aber es gibt eine einfachere Möglichkeit! Sie können jedes Programm alphanumerisch benennen und starten.

Die erste Zeile jedes Programms benennen, das als Referenz verwendet werden soll. Das Label besteht aus einem Buchstaben und alphanumerischen Zeichen; davor muß entweder ein \* stehen oder der Ausdruck muß in Anführungszeichen eingeschlossen sein, gefolgt von einem Doppelpunkt.

### Beispiel:

```
10: *A: PRINT "FIRST"  
20: END  
80: "B": PRINT "SECOND"  
90: END
```

Die Formate \*Label oder "Label" können beide verwendet werden. Das Format \*Label wird allerdings empfohlen, da es schneller zur Ausführung kommt und in einer Programmauflistung besser erkennbar ist.

## BASIC-Kommandos (Befehle)

Alle BASIC-Anweisungen müssen Befehle enthalten. Diese Befehle sagen dem Computer, welche Operation er durchführen soll. Ein Befehl ist immer ein Programmbestandteil, insofern erfolgt die Operation nicht direkt. Einige Anweisungen erfordern oder erlauben den Gebrauch eines Operanden.

### Beispiel:

```
10: DATA "HELLO"  
20: READ B$  
30: PRINT B$  
40: END
```

Operanden informieren den Computer darüber, auf welche Daten sich die vom Befehl angeordnete Operation bezieht. Einige Befehle erfordern Operanden, bei andern Befehlen sind sie fakultativ. Einige Befehle erlauben keine Operanden. (Im BASIC KOMMANDO LEXIKON finden Sie die BASIC-Befehle und ihre Anwendung auf dem Computer.)

### Hinweis:

Kommandos, Funktionen und Variable müssen in Großbuchstaben eingegeben werden.

## Direkt-Kommandos

Direkt-Kommandos sind Anweisungen an den Computer, die außerhalb eines Programms eingegeben werden. Sie fordern den Computer auf, bestimmte Vorgänge durchzuführen oder einen bestimmten Modus zu setzen, der dann wieder die Art der Programmbearbeitung determiniert.

Direkt-Kommandos haben unmittelbare Wirkung — sobald Sie die Eingabe des Direkt-Kommandos beendet haben (durch Betätigen der -Taste), wird die Anweisung ausgeführt. Direkt-Kommandos gehen keine Zeilennummern voraus:

RUN  
NEW  
RADIAN

## Modi (Betriebsarten)

Sie erinnern sich sicherlich, daß Sie, als Sie den Computer als Rechner benutzt haben, im CAL- oder im RUN-Modus gearbeitet haben. Der RUN-Modus wird ebenfalls gebraucht, um die von Ihnen geschriebenen Programme abzuarbeiten. Der PRO-Modus wird gewählt, wenn Sie Programme eingeben oder editieren wollen.

## Der Anfang des Programmierens

Nun können Sie mit dem Programmieren beginnen!

Nachdem Sie in vielen Übungen den Computer als Taschenrechner benutzt haben, sind Sie sicherlich schon recht vertraut mit der Tastatur. Von nun an werden wir, wenn wir eine Eingabe zeigen, nicht mehr jede Tastenbedienung angeben. Denken Sie immer daran, **SHIFT** zu betätigen, um Zugang zu den Zeichen oberhalb der Tasten zu bekommen, und vergessen Sie nicht am Ende jeder Zeile die Taste  zu betätigen.

Um Programmieranweisungen in den Computer einzugeben, schalten Sie den Computer erst in den PRO-Modus, indem Sie die **BASIC**-Taste benutzen. Auf dem Display erscheint dann folgende Anzeige:

```
PROGRAM MODE
>
PRO
```

Geben Sie das NEW-Kommando ein.

NEW 

```
PROGRAM MODE
NEW
>
```

Das NEW-Kommando löscht alle vorhandenen Programme und Daten aus dem Arbeitsspeicher. Das Bereitschaftssymbol erscheint, wenn Sie die -Taste gedrückt haben. Es zeigt an, daß der Computer nun Eingaben erwartet.

## Eingabe und Ausführung eines Programms

Vergewissern Sie sich, daß der Computer sich im PRO-Modus befindet, und geben Sie das folgende Programm ein:

10 PRINT

**SHIFT** + **"** HELLO

**SHIFT** + **"**

```
PROGRAM MODE
NEW
10PRINT"HELLO" _
```

Beachten Sie, daß der Computer bei Druck auf **←** automatisch zwischen die Zeilennummer und das Kommando einen Doppelpunkt setzt.

Überprüfen Sie, daß die Anweisung das richtige Format hat, und ändern Sie dann durch Druck auf die **BASIC** -Taste den Modus auf den RUN-Modus.

**C•CE** RUN **←**

```
RUN
HELLO
>
```

Da dies die einzige Programmzeile ist, beendet der Computer das Programm und geht auf das Bereitschaftszeichen von BASIC zurück, ">".

## Editieren eines Programms

Nehmen wir an, Sie wollen den Inhalt dessen, was Ihr Programm anzeigt, verändern, d.h. Sie wollen Ihr Programm editieren. Bei einem einzeiligen Programm können Sie die Eingabe schlicht wiederholen, aber wenn Sie komplexere Programme schreiben, wird das Editieren zu einem wichtigen Bestandteil des Programmierens. Wir wollen das Programm, das Sie gerade geschrieben haben, editieren.

Sind Sie noch im RUN-Modus? Wenn ja, so schalten Sie zurück in den PRO-Modus.

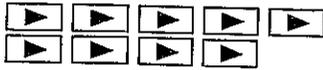
Um das Programm editieren zu können, müssen Sie es nun wieder aufrufen. Benutzen Sie dazu die **↑** -Taste. Sofern Ihr Programm vollständig durchgelaufen war, ruft die **↑** -Taste lediglich die letzte Programmzeile auf. Falls ein Fehler im Programm vorlag oder Sie die Ausführung mit **BREAK** unterbrochen hatten, ruft die **↑** -Taste die Zeile auf, in der der Fehler liegt oder in der der Abbruch erfolgte. Um Veränderungen im Programm vorzunehmen, kann das Programm mit der **↑** -Taste aufgerollt (Aufrufen der vorherigen Zeile) und mit der **↓** -Taste abgerollt werden (Anzeige der folgenden Zeile). Beim kontinuierlichen Drücken von **↑** oder **↓** wird das Programm auf dem Display auf- oder abgerollt.

Um den Cursor innerhalb einer Zeile des Programms zu bewegen, werden die Taste **▶** (rechter Pfeil) und **◀** (linker Pfeil) verwendet.

Mit Hilfe von **▶** setzen Sie nun den Cursor auf das ersten Zeichen, das Sie ändern möchten.

**↑**

```
10:PRINT "HELLO"
```



```
10 PRINT "HELLO"
```

Beachten Sie, daß der Cursor die Form eines blinkenden Rechtecks angenommen hat, um so anzuzeigen, daß er auf einem schon bestehenden Zeichen steht. Geben Sie folgendes ein:

GOODBYE **SHIFT** + **"**  
**SHIFT** + **!**

```
10 PRINT "GOODBYE"!_
```

Vergessen Sie nicht, am Zeilenende **←** zu drücken. Schalten Sie um in den RUN-Modus.

RUN **←**

```
RUN MODE  
RUN  
ERROR 10 IN 10
```

Die Fehlermeldung zeigt die Art des Fehlers an und die Nummer der Zeile, in welcher der Fehler aufgetreten ist.

Drücken Sie **C•CE**, um die Fehlermeldung zu beseitigen.

Schalten Sie zurück in den PRO-Modus. Sie müssen unbedingt im PRO-Modus sein, um Änderungen im Programm vornehmen zu können. Mit Hilfe von **↑** (oder **↓**) rufen Sie die Zeile, in der sich der Fehler befindet, ab.

**↑** (oder **↓**)

```
10 PRINT "GOODBYE"!_
```

Der blinkende Cursor steht auf der kritischen Stelle. Sie haben gelernt, daß Sie, wenn Sie Zeichenfolgen-Konstanten in BASIC eingeben, alle dazugehörigen Zeichen in Anführungszeichen einschließen müssen. Benutzen Sie die **DEL**-Taste, um das "!" zu löschen.

**DEL**

```
10 PRINT "GOODBYE" _
```

Wir wollen nun das Ausrufezeichen "!" an die richtige Stelle setzen. Wenn man Programme aufbereitet, werden die DEL- und die INS-Funktion genauso benutzt wie beim Korrigieren einfacher Rechenoperationen. Mit Hilfe von **←** bringen Sie den Cursor auf das der Einfügung folgende Zeichen. Drücken Sie die **INS**-Taste. Eine Leerstelle markiert die Stelle, an der ein neues Zeichen einzugeben ist:

◀ INS

```
10 PRINT "GOODBYE"
```

Geben Sie das "!" ein. Die Anzeige sieht nun folgendermaßen aus:

SHIFT + !

```
10 PRINT "GOODBYE!"
```

Denken Sie daran  zu drücken, damit die Korrektur in das Programm aufgenommen wird.

### Hinweise:

- Wenn Sie eine ganze Zeile aus dem Programm löschen möchten, geben Sie nur die Zeilennummer ein; dann wird die ursprüngliche Zeile gelöscht. Mit dem DELETE-Kommando können Sie mehr als eine Zeile auf einmal löschen.
- Werden im PRO-Modus Daten eingegeben, ohne daß der Cursor angezeigt wird, werden die entsprechenden Zeichen gewöhnlich in der linken Spalte der obersten Displayzeile angezeigt: Benutzen Sie dagegen die  - oder  - Taste, wenn der Cursor angezeigt wird, werden die sukzessiven Eingaben von der Cursorposition aus angezeigt.

### Gebrauch von Variablen beim Programmieren

Wenn Ihnen der Gebrauch numerischer und Zeichenfolgen-Variablen nicht vertraut ist, lesen Sie noch einmal die entsprechenden Beschreibungen in Abschnitt 8.

Der Gebrauch von Variablen beim Programmieren erlaubt eine sehr differenzierte Ausnutzung des Computers.

Erinnern Sie sich, daß Sie feste numerische Variablen mit Buchstaben von A bis Z belegen:

A = 5

Für die Zuweisung von Zeichenfolge-Variablen können auch Buchstaben verwendet werden, gefolgt von einem \$-Zeichen.

**Zur Bestimmung einer festen numerischen und einer festen Zeichenfolge-Variablen darf nicht der gleiche Buchstabe verwendet werden.** Im gleichen Programm dürfen nicht A und A\$ bestimmt werden.

Die Länge für jede feste Zeichenfolge-Variable darf sieben Zeichen nicht überschreiten.

A\$ = "TOTAL"

Die einer Variablen zugewiesenen Werte können sich im Laufe eines Programms ändern, sie können jeweils den im Programm eingegebenen oder errechneten Wert annehmen. Eine Möglichkeit der Variablenzuweisung ist, den Befehl INPUT zu benutzen. Im folgenden Programmbeispiel ändern sich der Wert der Variablen A\$ aufgrund der Daten, die eingegeben werden, um auf die Frage "WORD?" zu antworten.

Geben Sie dieses Programm ein:

```
10: INPUT "WORD?";A$
20: B=LEN(A$)
30: PRINT "THE WORD (";A$;) HAS"
40: PRINT B;" LETTERS"
50: END
```

### Hinweis:

Zur Eingabe von "=" werden die Tasten **SHIFT** + **L** gedrückt.

Das zweite neue Element in diesem Programm ist der Gebrauch der END-Anweisung, um die Beendigung des Programms anzuzeigen. END läßt den Computer wissen, daß das Programm beendet ist. Es ist ein Vorteil beim Programmieren, immer eine END-Anweisung zu benutzen.

Wenn die Programme komplexer werden, möchten Sie sie möglicherweise noch einmal durchsehen, bevor Sie sie ausführen. Zu diesem Zweck benutzen Sie das LIST-Kommando. LIST, das nur im PRO-Modus benutzt werden kann, bringt das gesamte Programm, angefangen bei der niedrigsten Zeilennummer, zur Anzeige. Versuchen Sie das obige Programm aufzulisten:

LIST

```
10:INPUT "WORD?";A$
20:B=LEN (A$)
30:PRINT "THE WORD (";A$
;") HAS"
```

Benutzen Sie dazu die Tasten  und , um sich im Programm zu bewegen, bis Sie das gesamte Programm durchgesehen haben. Nachdem Sie das Programm kontrolliert haben, wählen Sie die RUN-Betriebsart.

**C•CE** RUN

```
RUN
WORD? _
```

HELP

```
WORD?HELP
THE WORD (HELP) HAS
4. LETTERS
>
```

Dies ist das Ende des Programms. Natürlich können Sie es noch einmal laufen lassen, indem Sie RUN eingeben. Auf jeden Fall wäre das Programm interessanter, wenn es mehr als eine Eingabemöglichkeit enthielte. Wir wollen das Programm nun so modifizieren, daß es weiterläuft, ohne daß man nach jeder Antwort RUN eingeben muß.

Gehen Sie zurück in den PRO-Modus und benutzen Sie den Auf- oder Abwärts-Pfeil (oder LIST), um in Zeile 50 zu kommen oder geben Sie folgendes ein:

LIST 50

```
50:END
```

Sie können nun 50 eingeben, um so die gesamte Zeile zu löschen. Sie können aber auch den Cursor mit  auf das E in END setzen. Ändern Sie die Zeile 50 folgendermaßen:

```
50: GOTO 10
```

Nun lassen Sie das modifizierte Programm laufen (RUN).

Die GOTO-Anweisung schafft eine Programm-Schleife (d.h. eine bestimmte Operation wird immer wieder durchgeführt). Da Sie der Schleife kein Ende gesetzt haben, wird sie nicht abgebrochen (eine "endlose" Schleife). Um dieses Programm zu unterbrechen, drücken Sie die **BREAK**-Taste.

Wenn Sie ein Programm mit der **BREAK**-Taste unterbrochen haben, können Sie es mit dem CONT-Kommando neu starten. CONT steht für CONTInue (fortsetzen). Bei Eingabe des CONT-Kommandos läuft das Programm von da aus weiter, wo es mit **BREAK** unterbrochen wurde.

### Komplexere Programmierung

Das folgende Programm berechnet die Fakultät von N (N!). Das Programm beginnt mit 1 und berechnet N! bis zu der Grenze, die Sie eingeben. Geben Sie das Programm ein:

```
100: F=1: WAIT 128
110: INPUT "LIMIT?" ;L
120: FOR N=1 TO L
130: F=F*N
140: PRINT N,F
150: NEXT N
160: END
```

In diesem Programm sind einige neue Erscheinungen enthalten. Der Befehl WAIT in Zeile 100 kontrolliert den Zeitraum, über den eine Zeile im Display bleibt, ehe das Programm fortgesetzt wird. Die Zahlen und ihre Fakultäten werden angezeigt, wenn sie berechnet werden. Durch die WAIT-Anweisung bleiben sie etwa 2 Sekunden auf der Anzeige.

Ebenfalls in Zeile 100 tauchen zwei Anweisungen auf, die durch einen Doppelpunkt (:) getrennt werden. Sie können so viele Anweisungen in eine Zeile schreiben, wie Sie möchten, solange Sie die einzelnen Anweisungen durch Doppelpunkte voneinander trennen und die maximale Anzahl von 255 Zeichen pro Zeile nicht überschreiten. Solche Mehrfach-Anweisungen machen allerdings das Lesen und Schreiben eines Programms schwieriger. Es ist deshalb besser, sie nur zu benutzen, wenn die Einzel-Anweisungen sehr kurz und einfach sind oder wenn es einen besonderen Grund dafür gibt, Mehrfach-Anweisungen zu verwenden.

In diesem Programm wurde in Zeile 120 der Befehl FOR und in Zeile 150 der Befehl NEXT benutzt, um eine Schleife aufzubauen. In einem vorherigen Programm hatten Sie eine "endlose" Schleife geschaffen, die sich kontinuierlich bis zum Drücken der Taste **BREAK** wiederholte. Mit dieser FOR...NEXT-Schleife addiert der Computer jedesmal 1 zu N hinzu, wenn der Programmablauf den NEXT-Befehl erreicht. Dann wird geprüft, ob N die anfangs gesetzte Grenze L überschreitet. Ist N kleiner oder gleich L, beginnt der Programmdurchlauf wieder am Anfang der Schleife. Ist N größer als L, geht der Programmablauf in Zeile 160 weiter und wird dort beendet.

In einer FOR...NEXT-Schleife können Sie jede feste numerische Variable oder einfache Variable mit einfacher Genauigkeit benutzen. Sie müssen auch nicht mit Schritt 1 anfangen und Sie können bei jedem Programmschritt jeden beliebigen Betrag dazu addieren. (Näheres dazu finden Sie im BASIC KOMMANDO LEXIKON.)

Wir haben dieses Programm mit Zeilennummern von 100 aufwärts benannt. Eine solche Benennung mit unterschiedlichen Zeilennummern macht es möglich, daß man mehrere Programme zur gleichen Zeit im Arbeitsspeicher hat. Um dieses Programm anstelle des Programms der Zeile 10 durchlaufen zu lassen, schalten Sie in den RUN-Modus und geben folgendes ein:

```
C•CE  
R U N 1 0 0
```

Das Programm kann auch einen Namen mit einem Label erhalten; dann wird es mit "RUN \*Label" gestartet.

Hinweise zum PRINT-Befehl:

Wenn mehr als vier Zeilen angezeigt werden sollen, wird die erste Zeile nach oben abgerollt und kann nicht mehr zurückgerufen werden. Mit dem WAIT-Befehl innerhalb des Programms kann die Anzeige auf dem Display verlangsamt werden. Außerdem kann auch ein Drucker verwendet werden. (Siehe das BASIC KOMMANDO LEXIKON für die Befehle für WAIT oder LPRINT.)

Der WAIT-Befehl trifft auf jeden PRINT-Befehl zu. Lange PRINT-Befehle können damit in kürzere Befehle umgesetzt werden, wenn die Anzeige zu schnell abrollt.

**Beispiel:**

```
100 PRINT A, B, ..., P  
    ↓  
100 PRINT A, B, ... H: PRINT I, J, ..., P
```

Da die meisten Computer den WAIT-Befehl nicht unterstützen, kann auch eine Warteschleife verwendet werden, um die Zeit für die Darstellung zu verlangsamen, z.B. FOR J=1 TO 500:NEXT J.

## Speichern von Programmen im Computer

Sie erinnern sich sicherlich, daß Einstellungen und Funktionen im Computer erhalten bleiben, auch wenn Sie ihn abschalten. Programme bleiben ebenfalls im Speicher erhalten, wenn Sie den Computer abschalten oder er das automatisch tut (mit AUTO OFF). Selbst wenn Sie die Tasten **BREAK**, **C•CE** oder **SHIFT + CA** betätigen, bleiben die Programme erhalten.

Programme gehen nur dann aus dem Speicher verloren, wenn Sie einen der folgenden Vorgänge ausführen:

- NEW eingeben, ehe Sie in der PRO-Betriebsart mit dem Programmieren beginnen.
- Den Computer mit dem RESET-Schalter initialisieren.
- Ein neues Programm mit den gleichen Zeilennummern schreiben, die bereits für sich im Speicher befindliche Programme verwendet wurden.

## Ausführung von Programmen

Sofern die Kapazität des Arbeitsspeichers es zuläßt, können mehrere Programme geladen werden. Zur Ausführung des zweiten oder eines weiteren Programms können Sie einen der folgenden Schritte wählen:

Der RUN-Befehl: RUN Zeilennummer 

Der GOTO-Befehl: GOTO Zeilennummer 

Die Ausführung beginnt von der bestimmten Zeilennummer aus. Falls das Programm ein Label enthält, z.B. \*AB, wird das Programm durch die Eingabe von "RUN\*AB 

ausgeführt. Die folgende Liste zeigt die Unterschiede der Variablen und des Status bei der Ausführung eines Programms mit den Befehlen GOTO und RUN.

Ausführung mit RUN	Ausführung mit GOTO
<ul style="list-style-type: none"><li>• Löscht die WAIT-Einstellung.</li><li>• Löscht das USING-Format.</li><li>• Löscht Felder und einfache Variable.</li><li>• Initialisierung der DATA-Anweisung für eine READ-Anweisung.</li><li>• Löscht die PRINT=LPRINT-Einstellung.</li><li>• Der parallele Port wird geschlossen.</li></ul>	<ul style="list-style-type: none"><li>• Erhält die WAIT-Einstellung.</li><li>• Erhält das USING-Format.</li><li>• Erhält Felder und einfache Variable.</li><li>• Keine Initialisierung der DATA-Anweisung für eine READ-Anweisung.</li><li>• Erhält die PRINT=LPRINT-Einstellung.</li><li>• Der parallele Port bleibt offen.</li></ul>

### Hinweis:

Bei der Ausführung eines Programms mit dem RUN-Befehl werden alle Variablen für Daten gelöscht. (Feste Variable bleiben erhalten.) Sollen die Variablen erhalten bleiben, muß der Befehl GOTO benutzt werden.

# 10. FEHLERSUCHE

Nachdem Sie ein neues BASIC-Programm eingeben, wird es in der Regel beim ersten Startversuch nicht laufen. Selbst wenn Sie ein Programm nur abtippen, von dem Sie wissen, daß es korrekt ist, wie z.B. die in diesem Handbuch vorgestellten, dürfte Ihnen normalerweise mindestens ein Tippfehler unterlaufen. Handelt es sich um ein längeres Programm, wird es oft auch mindestens einen logischen Fehler enthalten.

Es folgen einige grundsätzliche Hinweise, wie Sie solche Fehler finden und korrigieren.

Sie lassen das Programm laufen und erhalten eine Fehlermeldung:

1. Schalten Sie zurück in den PRO-Modus und benutzen Sie  oder , um die fehlerhafte Zeile ins Display zu rufen. Der Cursor befindet sich an der Stelle, an welcher der Fehler auftrat.
2. Wenn Sie aus der Art, wie die Programmzeile geschrieben ist, keinen offensichtlichen Syntaxfehler entnehmen können, kann das Problem auch an den von Ihnen verwendeten Werten liegen. So erzeugt beispielsweise CHR\$(A) eine Leerstelle, wenn A den Wert 1 hat. Überprüfen Sie die Werte der von Ihnen verwendeten Variablen, indem Sie entweder im RUN- oder im PRO-Modus die einzelnen Variablennamen gefolgt von  eingeben.

Sie lassen das Programm mit RUN laufen und erhalten keine Fehlermeldung, aber das Programm tut nicht, was Sie von ihm erwarten:

1. Überprüfen Sie das Programm Zeile für Zeile unter Verwendung von LIST und den  und  -Tasten, um herauszufinden, ob Sie das Programm korrekt eingegeben haben. Es ist erstaunlich, wieviele Fehler beim bloßen erneuten Durchsehen eines Programms gefunden werden können!
2. Versuchen Sie jede einzelne Zeile beim Durchlesen so zu interpretieren, als wären Sie ein Computer. Nehmen Sie einfache Werte und realisieren Sie die Operationen der einzelnen Zeilen, um herauszufinden, ob Sie die gewünschten Ergebnisse erhalten.
3. Fügen Sie eine oder mehrere zusätzliche PRINT-Anweisungen in das Programm ein, um wichtige Werte und wichtige Positionen zur Anzeige zu bringen. Benutzen Sie diese, um die korrekten Teile des Programms von den möglicherweise fehlerhaften zu isolieren. Diese Vorgehensweise ist auch nützlich um zu bestimmen, welche Teile eines Programms schon abgearbeitet wurden. Sie können den Programmablauf auch an kritischen Stellen vorübergehend mit STOP unterbrechen, um dann einzelne Variable zu überprüfen.
4. Verwenden Sie TRON (TRace ON) und TROFF (TRace OFF) als Direkt-Kommandos innerhalb des Programms, um den Programmablauf durch die einzelnen Zeilen hindurch verfolgen zu können. Halten Sie das Programm an kritischen Punkten an, um den Inhalt von wichtigen Variablen zu überprüfen. Dies ist zwar eine sehr langsame Methode, Fehler aufzuspüren, aber es ist manchmal die einzige.

## Trace-Modus

Unabhängig davon, wie vorsichtig Sie auch programmieren, wird es vorkommen, daß Sie ein Programm schreiben, das nicht das macht, was Sie von ihm erwarten. Dann geht es darum, den Fehler zu finden. BASIC enthält deshalb eine spezielle Programmablauf-Routine, den "Trace"-Modus.

TRON (TRace ON) startet den Trace-Modus. Die TRON-Anweisung kann als Direkt-Kommando (im RUN-Modus) oder auch innerhalb eines Programms als Befehl verwendet werden. Wird TRON als Direkt-Kommando verwendet, informiert es den Computer, daß eine Ablaufverfolgung während der Ausführung aller nachfolgenden Programme benötigt wird. Die zu verfolgenden Programme werden dann ganz normal gestartet, mit Hilfe des GOTO- oder RUN-Kommandos. Wird TRON innerhalb eines Programms verwendet, so wird der Trace-Modus erst beim Erreichen der entsprechenden Zeile eingeschaltet. Wenn aus irgendeinem Grund die Zeile nie erreicht wird, bleibt der Trace-Modus ineffektiv.

## Vorgehen bei der Fehlersuche

1. Die RUN-Betriebsart einstellen.
2. Eingabe von TRON  , um den Trace-Modus zu aktivieren.
3. Eingabe von RUN  , um das Programm ausführen zu lassen. Nach jeder Zeile unterbricht der Computer die Ausführung und zeigt die Zeilennummer der gerade ausgeführten Zeile an.
4. Die  -Taste drücken, um auf die zu prüfende Zeile zu gelangen. Durch Gedrückthalten der  -Taste wird das Programm Schritt für Schritt ausgeführt. Beim Loslassen der Taste hält die Ausführung an. Der Inhalt der zu prüfenden Zeile kann durch Gedrückthalten der Taste  angesehen werden. (Beim Loslassen von  erscheint die Befehlseingabezeile. Zur Fortsetzung der Ausführung im Trace-Modus die Taste  drücken.)
5. Zur Weiterführung CONT  eingeben. Wenn die Ausführung während der Dateneingabe mit dem INPUT-Befehl unterbrochen wurde, braucht zur Fortführung des Programms nur die  -Taste gedrückt zu werden.
6. Den Trace-Vorgang weiterführen und prüfen, ob das Programm ordnungsgemäß ausgeführt wird, indem nach jeder Zeile die Reihenfolge der Ausführung und der Inhalt der Variablen überprüft wird. Wenn das Programm nicht ordnungsgemäß abläuft, muß die Logik verbessert werden.
7. Nach der Fehlersuche durch Eingabe von TROFF  die Trace-Betriebsart wieder deaktivieren.

### Beispiel:

```
10 INPUT "A=";A,"B=";B
20 C=A*2
30 D=B*3
40 PRINT "C=";C;" D=";D
50 END
```

## Das Programm laufen lassen.

RUN-Modus

TRON

RUN

8  (Dateneingabe)

9  (Dateneingabe)

>

A=\_

B=\_

} Ausführen des INPUT-Befehls.

10:

20:

30:

C=16. D=27.

40:

Wenn die Ausführung durch die **BREAK** -Taste unterbrochen wird, können die Variablen manuell abgerufen und ihre Werte überprüft werden. Durch Drücken von  wird jeweils eine Zeile ausgeführt. Eingabe von CONT  führt zur fortlaufenden Ausführung der Zeilen.

### Hinweise:

- Wenn an der mit LOCATE spezifizierten Stelle ein Ergebnis oder eine andere Information ausgegeben wird, erscheint die folgende Zeilennummer auf der Zeile nach dieser Information. (Siehe BASIC KOMMANDO LEXIKON für Hinweise zum Befehl LOCATE.)
- Wenn eine Variable manuell abgerufen oder eine manuelle Berechnung durchgeführt wurde, nachdem eine Stelle mit dem LOCATE-Befehl zugewiesen wurde, wird diese Zuweisung gelöscht.
- Der Trace-Modus bleibt aktiv, bis TROFF  eingegeben wird, die **SHIFT** + **CA** -Tasten gedrückt werden oder die Stromversorgung unterbrochen wird.
- Bei der Ausführung einer Kommentarzeile im Trace-Modus wird für diese Zeile keine Zeilennummer angezeigt. In diesem Fall bleibt die Nummer der zuletzt ausgeführten Zeile auf der Anzeige.

Zur Fehlersuche durch Unterbrechen eines laufenden Programms gehen Sie folgendermaßen vor:

- Drücken der **BREAK** -Taste während der Ausführung des Programms.
- Eingabe des STOP-Befehls an der entsprechenden Stelle.

Die Meldung für den Abbruch erfolgt und die Ausführung wird unterbrochen. Danach:

1. Die Inhalte der Variablen werden manuell überprüft.
2. Drücken der  -Taste zur Ausführung der Anweisungen, Zeile für Zeile. CONT  eingeben, um auf den vorherigen Vorgang zurückzugehen.

Ein Programm, welches mit der **BREAK** -Taste oder dem STOP-Befehl unterbrochen wurde, kann mit der  -Taste Zeile für Zeile ausgeführt werden.

# ASSEMBLER

Der PC-E220\* hat einen eingebauten Assembler, mit dem sich der Benutzer mit der Maschinensprache vertraut machen kann. In Kapitel 4 wird beschrieben, wie ein Quellprogramm im ASCII-Format innerhalb der TEXT-Betriebsart geschrieben und wie TEXT- und BASIC-Programme umgewandelt werden.

In Abschnitt 12 wird beschrieben, wie ein assembliertes Programm (auch "Objekt" genannt) in der Monitor-Betriebsart ausgeführt wird.

- \* Der PC-E220 wird im folgenden als "der Computer" bezeichnet.

# 11. TEXT-MODUS (TEXT-EDITOR)

In der TEXT-Betriebsart (Text-Editor) können Sie Programme im ASCII-Format schreiben oder editieren und sie dann durch die serielle E/A-Schnittstelle ein- oder ausgeben.

Alle BASIC-Anweisungen für den Computer werden in einem 2-Byte-Format gespeichert, dem sog. "Zwischencode". Da dieser Code je nach verwendeter Hardware bzw. BASIC-Interpreter unterschiedlich ist, kann er nicht für die Kommunikation zwischen Personal Computern oder anderen Geräten verwendet werden. Der ASCII-Code wird allgemein für die Datenkommunikation zwischen Personal Computern verwendet, da die ASCII-Darstellung von alphanumerischen Zeichen und Grundsymbolen die gleiche ist, unabhängig von der verwendeten Hardware.

Mit der TEXT-Betriebsart dieses Computers können Sie Programme in ASCII schreiben, editieren und speichern. Die Programme können auch vom Zwischen-code (BASIC) in ASCII konvertiert werden und umgekehrt.

Dieser Abschnitt beschreibt die Funktionen der TEXT-Betriebsart.

## Funktionen des Text-Modus

In der TEXT-Betriebsart stehen die folgenden Funktionen zur Verfügung:

**SHIFT** + **TEXT** → **TEXT-Betriebsart** (TEXT-Editor)

- **Edit** (Programmieren und Editieren von Programmen)
- **Del** (Löschen von Programmen)
- **Print** (Ausgabe der Programmauflistung an den Drucker)
- **Cmt** (Ein- und Ausgabe von Programmen mit Kassette)
  - Save (Schreiben)
  - Load (Lesen)
  - Verify (Vergleichen)
- **Sio** (Serieller E/A-Port)
  - Save (Senden)
  - Load (Empfangen)
  - Format (Einstellung der Parameter)
- **File** (Ein- und Ausgabe auf RAM-Diskette)
  - Save (Registrieren eines Dateinamens)
  - Load (Abrufen)
  - Kill (Löschen von Dateien)
  - Files (Abrufen von Dateinamen)
- **Basic** (Programm-Konvertierung zwischen BASIC- und TEXT-Formaten)
  - Basic ← text (Konvertierung von TEXT nach BASIC)
  - Text ← basic (Konvertierung von BASIC nach TEXT)

## Wahl der TEXT-Betriebsart

Zur Wahl der TEXT-Betriebsart wird **SHIFT** + **TEXT** gedrückt. Der Computer zeigt nun den abgebildeten TEXT EDITOR-Bildschirm an. Dieser Bildschirm wird auch das "Hauptmenü" genannt.

```
*** TEXT EDITOR ***
Edit   Del   Print Cmt
Sio    File  Basic
```

Von diesem Menü wählen Sie die gewünschte Funktion durch Eingabe des ersten Buchstabens (der als Großbuchstabe angegeben ist) der Funktionsbezeichnung. Nach der Wahl einer Funktion gibt der Computer manchmal ein Untermenü für diese Funktion aus oder die gewählte Funktion wird direkt ausgeführt.

### Hinweise:

- Um eine Funktion doch nicht auszuführen oder um auf ein Funktions-Untermenü bzw. das Hauptmenü zurückzugehen, wird die Taste **BREAK** gedrückt. Mit der Taste **C•CE** kann eine Fehlermeldung gelöscht bzw. ein oder mehrere eingegebene Zeichen gelöscht werden (z.B. ein Dateiname).
- Die TEXT-Betriebsart wird durch Wahl einer anderen Betriebsart oder durch Aus- und wieder Einschalten des Computers verlassen.

## Editieren

Zur Wahl der Editierfunktion vom Hauptmenü wird **E** gedrückt.

**E**

```
TEXT EDITOR
<
```

In der Editierfunktion ist die Eingabeaufforderung in der Befehlseingabe-Zeile wie folgt: "<" (anstatt ">" wie in BASIC).

Wie bei einem BASIC-Programm beginnt jede Zeile eines TEXT-Programms mit einer Zeilennummer. Bei einem TEXT-Programm fügt der Computer allerdings nicht automatisch nach der Zeilennummer einen Doppelpunkt (:) ein, wie das bei BASIC-Programmen der Fall ist. Auch wird nicht automatisch eine Leerstelle zwischen den eingegebenen Befehlen eingefügt. Jede Zeile erscheint genau so, wie sie eingetippt wird.

### Hinweise:

- Die Nummern der Programmzeilen werden automatisch in aufsteigender Reihenfolge sortiert.
- Der Bereich der möglichen Zeilennummern für ein Programm ist von 1 bis 65279. Wenn dieser Bereich überschritten oder keine Zeilennummer eingegeben wird, erfolgt die Anzeige einer Fehlermeldung (LINE NO. ERROR). Zum Löschen der Fehlermeldung **C•CE** drücken.

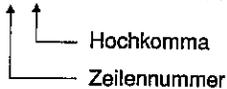
Zum Zurückgehen auf das Hauptmenü **BREAK** drücken.

### Hinweis:

Eine TEXT-Zeile kann nicht mit einer Zahl beginnen, die der Zeilennummer direkt folgt. Wenn die Zeile unbedingt mit einer Zahl beginnen soll, muß zwischen der Zeilennummer und der Zahl ein Hochkomma (') eingefügt werden.

### Beispiel:

50 ' 100 FORMAT (17X, A) ←



(Beispiel-Programm) Geben Sie das folgende Programm ein:

```
10INPUT A
20B=A*A
30PRINT A, B
40END
```

```
10INPUT SPACE A ←
20B=A*A ←
30PRINT SPACE A, B ←
40END ←
```

```
10 INPUT A
20 B=A*A
30 PRINT A, B
40 END
```

### Hinweis:

Für Details über das Assemblieren dieses Beispiel-Programms siehe Seite 122.

## Editieren von Programmen

Ein TEXT-Programm wird genau so wie ein BASIC-Programm editiert. (Siehe die Erklärungen zum Programmieren in BASIC in Kapitel 9.)

Die Befehle L (Anzeigen) und R (Neunumerieren) der TEXT-Betriebsart entsprechen den Befehlen LIST und RENUM in BASIC. (Für Details über die Befehle L und R siehe die Erklärungen für die Befehle LIST und RENUM im BASIC KOMMANDO LEXIKON.)

Wenn der R-Befehl in einem TEXT-Programm ausgeführt wird, das von einem BASIC-Programm konvertiert wurde, werden lediglich die Zeilennummern am Anfang einer Zeile neu numeriert, während die Zeilennummern innerhalb der Anweisungen GOTO, THEN, GOSUB bzw. RESTORE nicht neu numeriert werden. In diesem Fall läuft das Programm nicht, wenn es wieder in BASIC zurückkonvertiert wird.

### Befehlsformate für L:

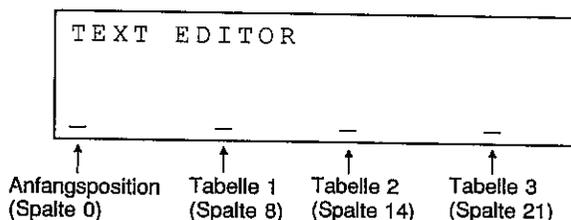
- (1) L ←
- (2) L Zeilennummer ←
- (3) L Label ←

### Befehlsformat für R:

R [neue Zeilennummer][, [erste Zeile]][, Inkrement] ←

### Die Taste TAB

In der Editierfunktion kann der Cursor mit der Taste TAB an die nächste Tabellenposition bewegt werden.



Beim ersten Drücken von **TAB** bewegt sich der Cursor auf die Spalte 8. Beim nächsten Drücken bewegt er sich auf die Spalte 14 (6 Stellen nach der ersten Tabellenposition). Bei jedem folgenden Drücken auf **TAB** bewegt sich der Cursor um sieben Stellen vorwärts.

## Löschen eines TEXT-Programms (Del)

Zur Wahl der Löschfunktion vom Hauptmenü wird **D** gedrückt.

**D**

```
*** TEXT EDITOR ***  
TEXT DELETE OK? (Y)
```

Beim Drücken von **Y** wird der gesamte TEXT-Speicherbereich vollständig gelöscht, einschließlich dem TEXT-Programm, und das Hauptmenü wird wieder angezeigt.

Wenn eine andere Taste als **Y** gedrückt wird, geht der Computer auf das Hauptmenü zurück, ohne etwas zu löschen.

### Hinweis:

Wenn im TEXT-Bereich kein Text gespeichert ist, reagiert der Computer bei angezeigtem Hauptmenü nicht auf das Drücken von **D**.

## Ausdrucken einer TEXT-Programmauflistung (Print)

Den zusätzlichen Drucker CE-126P mit dem Computer verbinden und den Computer sowie den Drucker einschalten. Das Hauptmenü zur Anzeige bringen und **P** drücken, um das gespeicherte TEXT-Programm auszudrucken.

**P**

```
*** TEXT EDITOR ***  
--- PRINTING ---
```

Nach dem Ausdrucken zeigt der Computer wieder das Hauptmenü an.

### Hinweis:

Zum Abbrechen des Ausdrucks **BREAK** drücken. Wenn der Drucker nicht eingeschaltet oder nicht mit dem Computer verbunden ist, reagiert der Computer bei angezeigtem Hauptmenü nicht auf das Drücken von **P**.

## Sichern, Laden und Vergleichen eines TEXT-Programms mit einer Kassette (Cmt)

Zur Anzeige des Kassetten-Menüs (CMT) vom Hauptmenü wird **C** gedrückt.

**C**

```
<< CMT >>  
Save Load Verify
```

Vom Kassetten-Menü wird die entsprechende Funktion gewählt — Sichern (Save), Laden (Load) oder Vergleichen (Verify) —, indem der erste Buchstabe der Funktion eingegeben wird (S, L oder V).

Vor der Wahl einer Funktion muß der zusätzliche Drucker CE-126P und ein Kassettenrekorder an den Computer angeschlossen und die entsprechenden Vorbereitungen zum Sichern (Aufnehmen) bzw. Laden (Wiedergeben) ausgeführt werden.

### Sichern eines Programms auf Kassette (Save)

Beim Drücken von  bei angezeigtem Kassetten-Menü fragt der Computer nach dem Namen der zu sichernden Datei.

```
<< CMT >>
→Save  Load  Verify
FILE NAME=?
```

Den Dateinamen eingeben, dann  drücken. Das Programm wird nun gesichert.

### Beispiel:

Sichern eines Programms mit dem Namen "TEXT"

TEXT

```
<< CMT >>
→Save  Load  Verify
FILE NAME=TEXT_
```

```
<< CMT >>
--- SAVING ---
```

Nach Beendigung des Sicherungsvorgangs zeigt der Computer wieder das Kassetten-Menü an. Um zu überprüfen, ob die Datei richtig gesichert wurde, wird die Funktion "Verify" vom Kassetten-Menü gewählt.

### Hinweise:

- Beim Drücken von  ohne Eingabe eines Dateinamens bei entsprechender Aufforderung wird das Programm ohne einen Namen auf die Kassette gesichert.
- Ein Dateiname kann aus bis zu acht Zeichen bestehen.
- Wenn keine TEXT-Datei gespeichert ist, geht der Computer auf das Kassetten-Menü zurück, wenn bei Aufforderung zur Eingabe des Dateinamens  gedrückt wird.

### Sichern/Laden des gesamten Inhalts auf die RAM-Diskette

Mit einem Dateigruppenzeichen (\*.\*) können alle Dateien auf die RAM-Diskette gespeichert werden. Bei der Aufforderung zur Eingabe des Dateinamens \*.\* eingeben. (Dieses Vorgehen eignet sich besonders, um kurzfristig den gesamten Inhalt auf die RAM-Diskette zu speichern, wenn der Maschinencode-Bereich neu geordnet werden muß.)

Um alle Dateien von der Kassette zu laden, kann ebenfalls das Dateigruppenzeichen (\*.\*) bei der Aufforderung zur Eingabe des Dateinamens verwendet werden, nachdem die Ladefunktion vom Kassetten-Menü gewählt wurde. Die Dateien werden im gleichen Format geladen, in dem sie geschrieben wurden (z.B. ASCII oder BASIC-Zwischencode).

### Laden eines Programms von der Kassette

Beim Drücken von  bei angezeigtem Kassetten-Menü (CMT) fragt der Computer nach dem Namen der zu ladenden Datei.

```
<< CMT >>
Save →Load Verify
FILE NAME=?
```

Den Namen der zu ladenden Datei eingeben, dann  drücken. Das Programm wird nun geladen.

### Beispiel:

Laden eines Programms mit dem Namen "TEXT"

TEXT

```
<< CMT >>
--- LOADING ---
```

Der Computer sucht auf dem Band nach dem eingegebenen Dateinamen (in diesem Beispiel "TEXT") und lädt den ASCII-Inhalt der Datei in den TEXT-Bereich des Computers. Während das Programm geladen wird, erscheint ein Sternchen (\*) in der unteren rechten Ecke des Displays.

Nach Beendigung des Ladevorgangs zeigt der Computer wieder das Kassetten-Menü an. Um zu überprüfen, ob die Datei richtig geladen wurde, wird die Funktion "Verify" vom Kassetten-Menü gewählt.

### Hinweise:

- Beim Drücken von  ohne Eingabe eines Dateinamens bei entsprechender Aufforderung wird die erste ASCII-Datei der Kassette geladen, die nach Einschalten des Gerätes gefunden wird.
- Wenn der eingegebene Dateiname nicht gefunden wird, setzt der Computer die Suche fort, auch nachdem das Band sich nicht mehr bewegt. In diesem Fall  drücken, um die Suche abzubrechen.
- Wenn beim Laden ein Fehler auftritt oder der Ladevorgang mit  unterbrochen wurde, wird nur der Teil bis zu diesem Punkt in den TEXT-Bereich gesichert.

### Vergleichen des Inhalts einer gesicherten bzw. geladenen Datei

Mit der Funktion zum Vergleichen kann überprüft werden, ob ein Programm richtig auf Kassette gesichert wurde bzw. von der Kassette wieder auf den Computer geladen wurde.

Das Kassetten-Menü zur Anzeige bringen und  drücken, um die Funktion zum Vergleichen zu wählen.

```
<< CMT >>
Save Load →Verify
FILE NAME=?
```

Den Namen der zu vergleichenden Datei eingeben, dann  drücken. Die Programme werden nun verglichen.

**Beispiel:**

Vergleichen des Inhalts eines Programms mit dem Namen "TEXT"

TEXT

```

      << CMT >>
    --- VERIFYING ---
  
```

Der Computer sucht auf dem Band nach dem eingegebenen Dateinamen (in diesem Beispiel "TEXT") und vergleicht den ASCII-Inhalt der Datei mit dem Inhalt, der in den TEXT-Bereich des Computers geladen wurde. Während dem Vergleichen erscheint ein Sternchen (\*) in der unteren rechten Ecke des Displays.

Nach Beendigung des Vergleichens zeigt der Computer wieder das Kassetten-Menü an.

Beim Auffinden eines nicht übereinstimmenden Teils wird eine Fehlermeldung (VERIFY ERROR) angezeigt.

**Hinweise:**

- Beim Drücken von  ohne Eingabe eines Dateinamens bei entsprechender Aufforderung wird die erste ASCII-Datei der Kassette verglichen, die nach Einschalten des Gerätes gefunden wird.
- Wenn der eingegebene Dateinamen nicht gefunden wird, setzt der Computer die Suche fort, auch nachdem das Band sich nicht mehr bewegt. In diesem Fall  drücken, um die Suche abubrechen.

**Serielle Eingabe/Ausgabe (Sio)**

Beim Drücken von  bei angezeigtem Hauptmenü gibt der Computer das Menü für die serielle Eingabe/Ausgabe (SIO-Menü) aus.

```

      << SIO >>
    Save  Load  Format
  
```

Vom SIO-Menü wird die entsprechende Funktion gewählt — Senden (Save), Empfangen (Load) oder Formatieren (Format) —, indem der erste Buchstabe der Funktion eingegeben wird (S, L oder F).

**Einstellung der E/A-Parameter (Format)**

Mit dieser Funktion können die Parameter der seriellen Kommunikation eingestellt werden. Die Kommunikations-Parameter müssen mit demjenigen Gerät übereinstimmen, mit dem dieser Computer kommunizieren soll.

Zur Anzeige eines Hilfsmenus vom SIO-Menü wird  gedrückt. Danach eine beliebige Taste drücken oder etwas warten, bis die Einstellung der Kommunikations-Parameter auf dem Display angezeigt werden.

```

      << SIO >>
    Select ←, →, ↑, ↓ key
    Set    ┘ key
    --- push any key ---
  
```

Eine beliebige Taste drücken oder etwas warten, bis die folgende Anzeige erscheint.

```
→baud rate =1200
data bit   =7
stop bit   =1
parity     =none
```

→ zeigt den gewählten Parameter an. Mit den Taste  bzw.  kann → auf einen zu ändernden Parameter bewegt werden. Es können insgesamt sieben Parameter eingestellt werden. Mit der Taste  können die Parameter auf dem Display durchlaufen werden.

⋮

```
parity     =none
end of line =CR
end of file =1A
→line number =yes
```

Mit  und  werden die Einstellungen geändert. Die Einstellung für den Parameter "end of file" (Ende der Datei) muß allerdings manuell eingegeben werden. Nach der Eingabe der Änderungen wird  gedrückt, um die Änderungen zu speichern. Wenn die neuen Einstellungen nicht gespeichert werden, verwendet der Computer die vorher eingestellten Parameter.

### Erläuterung der Kommunikations-Parameter

- Baudrate : 300, 600, 1200, 2400, 4800  
Die Baudrate entspricht der Geschwindigkeit der Datenübertragung. Je größer die Baudrate, desto schneller die Geschwindigkeit der Datenübertragung. Die Baudrate kann zwischen 300, 600, 1200, 2400 und 4800 bps (Baud pro Sekunde) gewählt werden.
- Data bit : 7 oder 8  
Die Datenlänge (Datenbit) ist die Anzahl der Bit, die benötigt werden, um ein Zeichen darzustellen. Es können entweder 7 oder 8 Bit eingestellt werden.
- Stop bit : 1 oder 2  
Das Stopbit bestimmt der Länge des Stopbits am Ende jedes Zeichens.
- Parity : none, even oder odd  
Die Parität bestimmt die Art der Datenüberprüfung (Paritätsprüfung).  
keine (none) ... Zu den übertragenen Daten wird kein Paritätsbit hinzugefügt (keine Paritätsprüfung).  
gerade (even)... Bestimmt eine gerade Parität.  
ungerade (odd) ... Bestimmt eine ungerade Parität.
- End of line : CR, LF oder CR + LF  
Mit "Ende der Zeile" wird der Begrenzungscode für das Ende jeder Programmzeile bestimmt.  
CR ... Bestimmt den Code für einen Wagenrücklauf.  
LF ... Bestimmt den Code für einen Zeilenvorschub.  
CR + LF ... Bestimmt den Code für CR und LF.
- Ende of file : 00 bis FF (zweistellige Hexadezimalzahl)  
Mit "Ende der Datei" wird ein Code für das Ende des Textes bestimmt, der das Ende eines Programms oder einer anderen Datei anzeigt.

- Zeilennummer : yes oder no

Mit der "Zeilennummer" wird bestimmt, ob ein TEXT-Programm mit oder ohne Zeilennummern gesendet wird.

ja (yes) ... Das Programm wird mit Zeilennummern gesendet.

nein (no) ... Das Programm wird ohne Zeilennummern gesendet.

Mit "Zeilennummer" wird weiterhin bestimmt, ob beim Empfang automatisch eine Zeilennummer (in Schritten von 10) zu jeder Programmzeile hinzugefügt werden soll.

ja (yes) ... Es werden keine Zeilennummern hinzugefügt. "ja" wird gewählt, wenn das Programm bereits Zeilennummern enthält.

nein (no) ... Es werden automatisch Zeilennummern hinzugefügt.

Wenn die empfangene Datei keine Zeilennummern enthält, obwohl "ja" eingestellt war, wird eine Fehlermeldung (LINE NO. ERROR) angezeigt.

Bei diesem Computer sind die Kommunikations-Parameter ab Werk folgendermaßen eingestellt:

Parameter	Bedingung
baud rate	1200
data bit	7
stop bit	1
parity	none
end of line	CR
end of file	1 A
line number	yes

Die Werte dieser Parameter können wie bei "Einstellung der I/O-Parameter (Format)" beschrieben eingestellt werden. Nachdem sie geändert und gespeichert wurden, gelten diese neuen Parameter, bis der RESET-Schalter gedrückt wird, um den Speicher zu löschen, oder bis die Batterie ausgewechselt wird oder bis die Einstellungen erneut geändert werden.

### Senden von Programmen (Save)

Beim Drücken von  bei angezeigtem SIO-Menü beginnt der Computer, ein TEXT-Programm über den seriellen I/O-Port zu senden.

```

      << SIO >>
    --- SENDING ---
  
```

Nach dem Senden geht der Computer auf das SIO-Menü zurück.

### Hinweise:

- Zum Abbrechen des Sendens  drücken. Der Computer geht auf das SIO-Menü zurück.
- Wenn im TEXT-Bereich kein Programm gespeichert ist, reagiert der Computer nicht auf das Drücken von .

## Empfangen von Programmen (Load)

Beim Drücken von  bei angezeigtem SIO-Menü beginnt der Computer, die Daten über den seriellen I/O-Port zu empfangen.

```
<< SIO >>
--- RECEIVING ---
```

Nach dem Empfangen geht der Computer auf das SIO-Menü zurück.

### Hinweise:

- Zum Abbrechen des Empfangs  drücken. Der Computer geht auf das SIO-Menü zurück.
- Wenn das Programm nicht richtig empfangen wurde oder wenn ein Paritätsfehler auftrat, erscheint eine Fehlermeldung (I/O DEVICE ERROR). Zum Löschen der Fehlermeldung  drücken.

## Programm-Dateien (File)

Beim Drücken von  bei angezeigtem Hauptmenü gibt der Computer das Programm-Datei-Menü aus.

```
<< PROGRAM FILE >>
Save Load Kill Files
```

Von diesem Menü wird die entsprechende Funktion gewählt — Registrieren (Save), Abrufen (Load), Löschen (Kill) oder Dateien (Files) —, indem der erste Buchstabe der Funktion eingegeben wird (S, L, K oder F).

### Registrieren eines TEXT-Programms (Save)

Registrieren bedeutet, dem TEXT-Programm einen Namen zuzuweisen. Der Computer verwendet dann diesen Namen, um mit dieser Datei weitere Funktionen auszuführen.

Beim Drücken von  bei angezeigtem Programm-Datei-Menü fordert der Computer zur Eingabe des Namens der zu registrierenden Datei auf.

```
<< PROGRAM FILE >>
→Save Load Kill Files
FILE NAME=?
```

Den Dateinamen eingeben und  drücken. Der Computer registriert nun diese Datei.

### Beispiel:

Registrieren einer Datei mit dem Dateinamen "TEST".

TEST

```
<< PROGRAM FILE >>
→Save Load Kill Files
FILE NAME=TEST_
```



```

<< PROGRAM FILE >>
Save   Load   Kill   Files

```

Der Computer registriert die Datei "TEST" und geht dann wieder auf das Programm-Datei-Menü zurück.

**Hinweise:**

- Nach der Wahl der Registrier-Funktion aus dem Programm-Datei-Menü muß unbedingt ein Dateiname eingegeben werden. Wenn die Taste ohne Eingabe eines Namens gedrückt wird, erfolgt eine Fehlermeldung (ILLEGAL FILE NAME). Zum Löschen der Fehlermeldung **C•CE** drücken.
- Ein Dateiname kann aus bis zu acht Zeichen bestehen und eine Erweiterung von bis zu drei Zeichen enthalten. Wenn keine Erweiterung eingegeben wird, weist der Computer automatisch die Erweiterung ".TXT" zu.
- Wenn kein TEXT-Programm im TEXT-Bereich gespeichert ist, kann auch keine Dateiregistrierung ausgeführt werden.

**Abrufen einer TEXT-Datei (Load)**

Beim Drücken von bei angezeigtem Programm-Datei-Menü gibt der Computer eine Liste der registrierten Dateien aus; dabei weist "LOAD =>" auf den ersten Dateinamen (wenn kein Programm registriert wurde, reagiert der Computer nicht auf das Drücken von ).



(Hier wird ein Beispiel für eine Liste mit registrierten Dateien angeführt.)

```

LOAD =>ABC      .TXT
        PRO      .TXT
        SAMPLE01 .BAS
        TEST     .TXT

```

Mit den Tasten und wird "LOAD =>" auf den Namen der abzurufenden Datei bewegt; danach drücken. Der Computer lädt den Inhalt der gewählten Datei in den TEXT-Bereich und geht dann auf das Programm-Datei-Menü zurück.

**Hinweis:**

In der TEXT-Betriebsart können nur Programme und Dateien abgerufen werden, die in der TEXT-Betriebsart registriert wurden. Beim Versuch des Abrufens eines BASIC-Programms, das mit einem SAVE-Befehl von BASIC gesichert wurde, wird eine Fehlermeldung (FILE MODE ERROR) angezeigt. Zum Löschen der Fehlermeldung **C•CE** drücken.

**Löschen einer Programm-Datei (Kill)**

Mit dieser Funktion wird eine bestimmte Datei gelöscht.

Beim Drücken von bei angezeigtem Programm-Datei Menü fragt der Computer nach dem Namen der zu löschenden Datei.



```

<< PROGRAM FILE >>
Save   Load =>Kill   Files
FILE NAME=?

```

Den Namen der zu löschenden Datei eingeben und drücken. Die Datei wird nun gelöscht.

### Beispiel:

Löschen einer Datei mit dem Dateinamen "TEST".

TEST

```
<< PROGRAM FILE >>
Save Load Kill Files
```

Der Computer löscht die Datei "TEST" und geht dann wieder auf das Programm-Datei-Menü zurück.

### Hinweise:

- Wenn der eingegebene Dateiname keine Erweiterung enthält, wird automatisch ".TXT" angenommen.
- Wenn die bestimmte Datei nicht gefunden wird, zeigt der Computer eine Fehlermeldung (FILE NOT FOUND) an. Zum Löschen der Fehlermeldung  drücken.

### Auflisten von Dateinamen (Files)

Beim Drücken von  bei angezeigtem Programm-Datei-Menü gibt der Computer eine Liste aller registrierten Dateien aus, dabei zeigt → auf den ersten Dateinamen der Liste. (Wenn keine Dateien registriert sind, reagiert der Computer nicht auf das Drücken von )

(Hier wird ein Beispiel für eine Liste mit registrierten Dateien angeführt.)

```
⇒ABC .TXT
PRO .TXT
SAMPLE01.BAS
```

Die noch nicht angezeigten Teile der Liste können durch Drücken von  oder  angesehen werden.

Um ein Programm zu laden, das durch → markiert ist, wird  +  gedrückt (oder  .

### Der BASIC-Konverter (Basic)

Mit dieser Funktion wird ein BASIC-Programm im Zwischencode-Format in eine TEXT-Datei im ASCII-Format umgewandelt oder umgekehrt. Diese Funktion ist nützlich, um BASIC-Programme mit einem Personal Computer zu bearbeiten, die auf diesem Computer PC-E220 geschrieben wurden.

Beim Drücken von  bei angezeigtem Hauptmenü gibt der Computer das BASIC-Konverter-Menü aus.

```
<< BASIC CONVERTER >>
Basic←text Text←basic
```

Von diesem Menü kann die Konvertierung von TEXT nach BASIC oder von BASIC nach TEXT gewählt werden. Den ersten Buchstaben des Formates eingeben, in das umgewandelt werden soll.

## Umwandlung von TEXT- und BASIC-Programmen (zwischen ASCII-Format und Zwischencode-Format)

Beim Drücken von  B bei angezeigtem BASIC-Konverter-Menü wandelt der Computer das TEXT-Programm im TEXT-Bereich in ein BASIC-Programm um und speichert es im Programmdaten-Bereich.

Beim Drücken von  T bei angezeigtem BASIC-Konverter-Menü wandelt der Computer das BASIC-Programm im Programmdaten-Bereich in ein TEXT-Programm um und speichert es im TEXT-Bereich.

### Beispiel:

Umwandeln eines TEXT-Programms in BASIC.

B

```
<< BASIC CONVERTER >>
--- CONVERTING ---
```

Nach der Konvertierung geht der Computer auf das Hauptmenü zurück. (Bei Umwandlung eines kurzen Programms beansprucht die Konvertierung nur sehr wenig Zeit.)

Wenn sich im Programm-Bereich bereits ein BASIC-Programm befindet, während ein TEXT-Programm in ein BASIC-Programm umgewandelt wird oder wenn andererseits im TEXT-Bereich bereits ein Programm vorhanden ist, während ein BASIC-Programm umgewandelt wird, fragt der Computer zur Sicherheit, ob das vorhandene Programm vor der Konvertierung gelöscht werden soll.

B

```
<< BASIC CONVERTER >>
→Basic←text  Text←basic
BASIC DELETE OK? (Y)
```

Beim Drücken von  Y löscht der Computer das vorhandene BASIC-Programm und beginnt mit der Konvertierung.

Beim Drücken einer anderen Taste als  Y wird die Konvertierung abgebrochen und der Computer geht auf das Hauptmenü zurück.

Im allgemeinen behält der Computer das Originalprogramm bei, nachdem es in ein anderes Format umgewandelt wurde. Wenn allerdings nach der Umwandlung eines Programms nicht genügend Speicherplatz zur Verfügung steht, fragt der Computer zur Sicherheit, ob das Originalprogramm gelöscht werden soll.

```
<< BASIC CONVERTER >>
--- CONVERTING ---
TEXT DELETE OK? (Y)
```

Beim Drücken von  Y löscht der Computer das Originalprogramm im Verlauf der Konvertierung. Am Ende der Konvertierung ist das Originalprogramm vollständig gelöscht.

Beim Drücken einer anderen Taste als  Y wird die Konvertierung abgebrochen und der Computer geht auf das Hauptmenü zurück.

**Hinweise:**

- Bei aktivierter Speicherschutzfunktion mit einem Paßwort kann die BASIC-Konverter-Funktion nicht vom Hauptmenü gewählt werden. Zuerst muß das Paßwort in der Betriebsart RUN oder PRO gelöscht werden.
- Der BASIC-Konverter löscht während der Konvertierung nicht die Inhalte der Datenbereiche (Variable). Falls nicht genügend Speicherplatz zur Verfügung steht, kann die Konvertierung manchmal nicht ausgeführt werden. Vor der Konvertierung muß sichergestellt werden, daß genügend freier Speicherplatz vorhanden ist. Bei Bedarf müssen Variable mit dem Befehl CLEAR aus dem Datenspeicherbereich gelöscht werden.
- Wenn während der Konvertierung und Löschen des Originalprogramms nicht genügend Speicherplatz zur Verfügung steht, wird eine Fehlermeldung (MEMORY OVER) angezeigt. Wenn dieser Fehler auftritt, wird das Programm in einen umgewandelten und einen noch nicht umgewandelten Teil getrennt. Es empfiehlt sich daher, das Programm auf Kassette oder ein anderes Speichermedium zu sichern, bevor eine Konvertierung ausgeführt wird. Dieser Fehler tritt häufiger auf, wenn ein Programm vom BASIC- in das TEXT-Format umgewandelt wird, als bei Konvertierung von TEXT- ins BASIC-Format.
- Bei der Konvertierung einer Datei von TEXT in BASIC ist es für den Computer unwichtig, welchen Inhalt die Original-TEXT-Datei enthält. Manchmal kann der vollständige Inhalt nicht mehr erhalten werden, wenn eine Original-TEXT-Datei zunächst in BASIC und dann wieder zurück in das TEXT-Format umgewandelt wird.
- Wenn die umgewandelte Datei mehr als 255 Zeichen (bzw. 255 Byte) enthält, wird der Überfluß nicht berücksichtigt.

**Beispiel:**

```
TEXT    10FORMULA
↓
BASIC   10:FOR_MULA
↓
TEXT    10FOR_MULA
```

# 12. MASCHINENSPRACHEN-MONITOR

Mit diesem Computer können Sie Programme sowohl in Maschinensprache als auch in BASIC schreiben. Um Ihnen bei der Programmierung im Maschinencode zu helfen, hat der Computer einen Maschinensprachen-Monitor (im folgenden als "der Monitor" bezeichnet). Dieser Monitor ist eines der besonderen Merkmale dieses Systems, mit dem Sie eine bestimmte Folge von Befehlen eingeben bzw. ausgeben oder ein in Maschinencode geschriebenes Programm ausführen können. Dieser Abschnitt beschreibt die Funktionen der Befehle dieses Maschinensprachen-Monitors für diesen Computer.

Die CPU des Computers ist ein Z80-Mikroprozessor (entsprechend CMOS Z80A), der häufig in den besonders leistungsfähigen 8-Bit-Computern verwendet wird. Im Handel sind zahlreiche Bücher über den Prozessor Z80 vorhanden, aus denen Sie Hinweise zur Maschinensprache des Z80 und andere wichtige Informationen entnehmen können.

Zur Ausführung der Befehle R und W wird ein Datenübertragungskabel benötigt, z.B. das Kabel CE-T801.

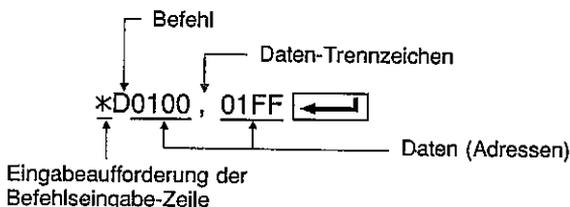
## Verwendung des Maschinensprachen-Monitors

Die Monitor-Betriebsart wird durch Eingabe von MON  in der BASIC-Betriebsart (RUN oder PRO) gewählt; die folgende Anzeige erscheint.

```
MACHINE LANGUAGE MONITOR
*
```

Das Sternchen (\*) auf der Anzeige bedeutet in der Monitor-Betriebsart die Eingabeaufforderung der Befehlseingabe-Zeile. Hier werden alle Befehle eingegeben. Alle notwendigen Adressen oder weitere Daten können nach dem Befehl hier eingegeben werden. Am Ende jeder Zeile  drücken, um den Befehl auszuführen.

### Beispiel:



### Hinweise:

- Bei aktivierter Speicherschutzfunktion mit Paßwort kann der Computer nicht in die Monitor-Betriebsart eingestellt werden.
  - Alle Adressen und Daten müssen im hexadezimalen Code sein.
  - Um mehr als eine Adresse oder um Datenteile voneinander zu trennen, wird ein Komma (,) verwendet.
  - Wenn kein hexadezimaler Code verwendet wird oder ein anderes Symbol außer dem Komma zur Datentrennung eingegeben wird, erfolgt ein Fehler (SYNTAX ERROR).
  - Die Monitor-Betriebsart kann durch Wahl einer anderen Betriebsart oder durch Aus- und wieder Einschalten des Computers verlassen werden.
- ① Da die Maschinensprache sehr kompliziert ist, kommt es häufig zu Programmfehlern. Bei der Ausführung eines Programms in Maschinensprache kann es vorkommen, daß BASIC-Programme, Daten oder andere Teile des Computerspeichers zerstört werden. Aus diesem Grund wird empfohlen, alle BASIC-Programme und andere Informationen auf Kassette oder ein anderes Speichermedium zu sichern, bevor ein Programm in Maschinensprache ausgeführt wird.
- ② Bei der Verwendung des Monitors kann der Zugang zu einem anderen als dem Maschinensprachen-Bereich (der mit dem USER-Befehl zugewiesen wurde) zur Zerstörung von BASIC- oder TEXT-Programmen führen oder andere Daten in diesem Bereich zerstören oder zu Fehlfunktionen führen. Stellen Sie sicher, daß Sie nur den für Maschinencode vorgesehenen Bereich verwenden.

## Auflistung der Befehle für den Maschinensprachen-Monitor

### USER — Benutzerbereich

Wirkung: Zuweisung eines Bereiches für Maschinensprache und Anzeige der Adressen dieses Bereiches.

Format: (1) USER01FF   
(2) USER   
(3) USER00FF 

Anmerkungen: • Mit Format (1) wird der Speicher-Adreßbereich von 0100H (der ersten Adresse) bis 01FF (der letzten Adresse) als Bereich für den Maschinencode zugewiesen. Die erste Adresse ist automatisch auf 0100H eingestellt.

Zugewiesener Bereich →

```
* USER01FF
FREE : 0100 - 01FF
*
```

- Mit Format (2) wird der Bereich der zugewiesenen Adressen für den Maschinencode-Bereich angezeigt.

```
* USER
FREE : 0100 - 01FF
*
```

Wenn kein Bereich für den Maschinencode zugewiesen wurde, wird "FREE: NOT RESERVED" angezeigt.

- Mit Format (3) wird ein bereits vorhandener Maschinencode-Bereich aus dem Speicher gelöscht und die Meldung "FREE: NOT RESERVED" angezeigt.
- Bei Eingabe eines unzulässigen Adreßbereiches für den Maschinencode-Bereich wird eine Fehlermeldung (MEMORY ERROR) angezeigt.

## S — Aktualisierung des Speichers

Wirkung: Aktualisieren des Speicherbereiches.

Format: (1) S0100

(2) S

Anmerkungen: • Mit Format (1) wird der Inhalt von Adresse 0100H (der ersten Adresse) angezeigt und zur Eingabe einer neuen Einstellung aufgefordert.

```
* S0100  
0100:10-
```

Vorhandener Speicherinhalt

- Um die Daten zu ändern, wird ein Byte anderer Daten eingegeben (zweistellig hexadezimal) und dann  gedrückt. Der Computer gibt nun den Inhalt der folgenden Adresse an und fordert wieder zur Eingabe von Daten auf.  
Wenn vorhandene Daten nicht geändert werden müssen, wird  gedrückt, ohne neue Daten einzugeben. Der Computer zeigt dann den Inhalt der folgenden Adresse an und fordert wieder zur Eingabe von Daten auf.
- Es können maximal zwei Stellen hexadezimaler Daten eingegeben werden. Um eine Eingabe vor dem Drücken von  zu löschen, wird  oder  gedrückt.
- Zum Abrufen des Inhalts der vorherigen Adresse  und zum Abrufen des Inhalts der folgenden Adresse  drücken.
- Mit Format (2) wird der Inhalt derjenigen Adresse angezeigt, die direkt neben der letzten mit dem S-Befehl dargestellten Adresse liegt.
- Zum Zurückgehen auf die Befehlseingabe-Zeile wird  gedrückt.

## D — Speicherauszug ausgeben

Wirkung: Ausgabe des Speicherinhaltes.

Format: (1) D0100 

(2) D 

(3) D0100, 01FF 

Anmerkungen: • Mit Format (1) werden 16 Byte aus dem Adreßbereich von 0100H (erste Adresse) bis 010FH ausgegeben. (Die ausgegebenen Daten werden in der Drucker-Betriebsart ausgedruckt.)

### Beispiel:

Erste Adresse eines 16-Byte-Segments →	0 1 0 0	:	3 E	0 1	1 8	0 4	> . . .
Kontrollsumme →	( 1 D )	:	3 A	0 F	0 1	3 C	: . . . <
			3 2	0 F	0 1	C 9	2 . . . <sup>8</sup>
			3 1	0 0	0 0	0 0	1 . . .

ASCII-Codedarstellungen der ausgegebenen Daten werden hier angezeigt. 00H-1FH wird allerdings als Punkt (.) wiedergegeben.

### Hinweise:

- Die Größe des ausgebenen Bereiches ist auf XXX0H-XXXFH festgelegt. Der Inhalt des gleichen Segments wird immer dann ausgegeben, wenn eine Adresse innerhalb eines 16-Byte-Segments angegeben wird. Wenn Sie zum Beispiel die Adresse 0104H angeben, wird der Inhalt der festgelegten 16 Segmente dieser Adresse ausgegeben, in diesem Fall 0100H-010FH.
- Zur Ausgabe des vorhergehenden 16-Byte-Segments  und zur Ausgabe des folgenden 16-Byte-Segments  drücken.
- Mit Format (2) wird der Inhalt desjenigen Segments ausgegeben, das direkt neben dem letzten mit dem D-Befehl ausgegebenen Segment liegt.
- Wenn das Format (3) in der Drucker-Betriebsart ausgeführt wird, gibt der Computer den Inhalt der bestimmten Bereiche, 0100H (erste Adresse)-01FFH (letzte Adresse), in 16-Byte-Inkrementen auf dem Drucker aus. Nach Beendigung der Ausgabe wird wieder die Befehlseingabe-Zeile dargestellt.  
Wenn das Format (3) ausgeführt wird, ohne daß sich der Computer in der Drucker-Betriebsart befindet, gibt der Computer den Inhalt der 16-Byte-Segmente auf dem Bildschirm aus, beginnend mit Adresse 0100H (der ersten Adresse). Während der Ausgabe auf dem Bildschirm in dieser Betriebsart berücksichtigt der Computer nicht die letzte bestimmte Adresse.
- Die Drucker-Betriebsart wird mit dem P-Befehl (siehe später) oder mit den Tasten  +  beendet oder verlassen.
- Zum Zurückgehen auf die Befehlseingabe-Zeile  drücken.

## Kontrollsumme

Kontrollsumme bezieht sich auf die Summe der Werte eines spezifischen Datensatzes. Diese Summe wird berechnet und einem Datensatz zugewiesen, wenn dieser Datensatz geschrieben oder ausgegeben wird. Der Computer berechnet die Summe der Inhalte eines 16-Byte-Segments, das mit dem D-Befehl ausgegeben wurde, und zeigt das wertniedrigste Einzelbyte der Summe als Ergebnis der Kontrollsumme an.

Wenn Sie zum Beispiel manuell ein Maschinencode-Programm eingeben, das von einem gedruckten Programm kopiert wird, dann können Sie in jedem ausgegebenen 16-Byte-Segment nach Fehlern suchen, indem die Ergebnisse der Kontrollsumme mit den Werten des Originalprogramms verglichen werden. Wenn das Programm allerdings mehr als einen Fehler enthält, kann die Kontrollsumme fälschlicherweise mit der des Originalprogramms übereinstimmen.

## P — Druckereinstellung

Wirkung: Aktivieren oder Desaktivieren der Drucker-Betriebsart

Format: P

Anmerkungen:

- Die Drucker-Betriebsart wird jedesmal beim Drücken von P  aktiviert bzw. deaktiviert. (Bei aktivierter Drucker-Betriebsart erscheint in der unteren rechten Ecke des Displays die Anzeige "PRINT".) Die Drucker-Betriebsart kann auch durch Drücken von  +  aktiviert bzw. deaktiviert werden.

### Hinweis:

- Der P-Befehl wird nicht ausgeführt, wenn kein Drucker angeschlossen oder ein angeschlossener Drucker nicht eingeschaltet ist.

## G — GOSUB

Wirkung: Ausführen von Maschinencode-Programmen von einer bestimmten Adresse aus.

Format: G0100

Anmerkungen:

- Dieser G-Befehl entspricht dem GOSUB-Befehl im BASIC. Damit wird ein Maschinencode-Programm von einer bestimmten Adresse aus ausgeführt, bis zum Auftreten eines RET-Befehls (Rückkehranweisung). Nach der Ausführung der Rückkehranweisung zeigt der Computer wieder die Befehlseingabe-Zeile an.

### Hinweis:

- Am Ende des Programms muß unbedingt eine Rückkehranweisung (RET-Befehl) eingefügt werden, andernfalls kann das Programm nicht richtig ausgeführt werden.

### Außer Kontrolle geratene Programme

Ein "Runaway"-Programm kann nicht richtig ausgeführt werden, weil es außer Kontrolle geraten ist. Die Zurückstellung des Systems ist die einzige Möglichkeit, ein derartiges Programm zu unterbrechen. In den meisten Fällen zerstört ein aus der Kontrolle geratenes Programm die Speicherinhalte, einschließlich der Maschinencode-Programme, BASIC-Programme und anderer Daten.

Ein Maschinencode-Programm kann bereits außer Kontrolle geraten, wenn es auch nur einen einzigen kleinen Fehler enthält. Aus diesem Grund wird empfohlen, alle BASIC-Programme und andere Informationen auf Kassette oder ein anderes Speichermedium zu sichern, bevor ein Programm in Maschinensprache ausgeführt wird.

#### Hinweis:

- Zum Abbrechen der Programmausführung den RESET-Schalter drücken.

### BP — Unterbrechungspunkt

- Wirkung: Einfügung eines Unterbrechungspunktes an einer bestimmten Adresse.
- Format:
- (1) BP011E 
  - (2) BP 
  - (3) BP0 
- Anmerkungen:
- Mit Format (1) wird an der Adresse 011EH ein Unterbrechungspunkt eingefügt. In einem Programm können mit dem Format (1) bis zu zwei Unterbrechungspunkte an verschiedenen Adressen eingefügt werden.
  - Beim Versuch der Eingabe eines dritten Unterbrechungspunktes wird der erste Unterbrechungspunkt gelöscht. Es können daher niemals mehr als zwei Unterbrechungspunkte in einem Programm enthalten sein.
  - Ein Unterbrechungspunkt sollte unbedingt bei einer Befehlsadresse (OP-Code) eingefügt werden. Wenn der Unterbrechungspunkt an einer Operandenadresse eingefügt wird, kann das Programm den Unterbrechungspunkt nicht lesen und er wird daher auch nicht richtig ausgeführt.
  - Mit Format (2) wird die Adresse des Unterbrechungspunktes angezeigt. Wenn kein Unterbrechungspunkt eingefügt wurde, erscheint nur die Eingabeaufforderung der Befehlseingabe-Zeile (\*) auf der folgenden Zeile.
  - Mit Format (3) werden vorhandene Unterbrechungspunkte gelöscht.
  - Ein Unterbrechungspunkt wird nach seiner Ausführung unwirksam. Wenn sich daher ein Unterbrechungspunkt innerhalb einer Programmschleife befindet, wird er nur bei der ersten Ausführung der Schleife berücksichtigt. Er kann allerdings mit dem G-Befehl wieder wirksam gemacht werden.

- Der Computer behält Unterbrechungspunkte bei, die bei der letzten Verwendung des Monitors bestimmt wurden. Wenn der Computer aus einer anderen Betriebsart wieder in die Monitor-Betriebsart zurückgestellt wird, können diese Unterbrechungspunkte mit dem G-Befehl wieder wirksam gemacht werden.

**Hinweis:**

- Der Inhalt einer Adresse, die eine Unterbrechungspunkt-Anweisung enthält, wird zeitweise durch "F7H" ersetzt, während das Programm ausgeführt wird. Wenn der RESET-Schalter vor der Ausführung dieses Unterbrechungspunktes gedrückt wird, bleibt der Inhalt "F7H". In diesem Fall muß "F7H" durch den ursprünglichen Inhalt ersetzt werden.

**R — Empfangen über SIO**

- Wirkung:** Empfangen von Daten über den seriellen I/O-Port (SIO). Dieser Befehl dient dem Empfangen von Maschinencodes von einem Personal Computer oder einem anderen Gerät.
- Format:** (1) R   
 (2) R0100 
- Anmerkungen:** Mit dem R-Befehl werden Daten im Intel-Hex-Format über SIO übertragen.
- Mit Format (1) werden empfangene Daten in bestimmte Adressen geladen, die von den empfangenen Daten spezifiziert werden.
  - Mit Format (2) werden nachfolgende Daten geladen, die nach einer Adresse mit dem Beginn bei 0100H empfangen wurden.
  - Nach Übertragung aller Daten wird der Adreßbereich angezeigt, in den die Daten geladen wurden.
  - Um die Übertragung von Daten abubrechen, wird die Taste **BREAK** gedrückt gehalten, bis die Befehlseingabe-Zeile angezeigt wird.

**W — Senden über SIO**

- Wirkung:** Ausgabe von Daten über den seriellen I/O-Port (SIO). Dieser Befehl dient dem Senden von Maschinencodes zu einem Personal Computer oder einem anderen Gerät.
- Format:** W0100, 01FF 
- Anmerkungen:**
- Bei der Ausführung des oben angegebenen Beispiels sendet der Computer Daten im Intel-Hex-Format aus dem Speicherbereich von 0100H (der ersten Adresse) bis 01FFH (der letzten Adresse) über den seriellen I/O-Port.
  - Um das Senden von Daten abubrechen, wird die Taste **BREAK** gedrückt gehalten, bis die Befehlseingabe-Zeile angezeigt wird.

**Hinweis:**

- Wenn ein Drucker an den peripheren Interface-Stiftstecker (11 Stifte) angeschlossen ist und der W-Befehl ausgeführt wird, kann es zu Fehlfunktionen sowohl des Computers als auch des Druckers kommen. In diesem Fall den Drucker ausschalten und dann die Taste **BREAK** gedrückt halten, bis die Befehlseingabe-Zeile angezeigt wird.

## Fehlermeldungen des Monitor-Modus

Es folgt eine Liste mit Fehlermeldungen, die während der Monitor-Betriebsart ausgegeben werden. Zum Löschen der Fehlermeldung **C•CE** drücken.

<b>Fehlermeldung</b>	<b>Beschreibung</b>
SYNTAX ERROR	Unzulässige Befehlssyntax.
MEMORY ERROR	Es wurde versucht, einen Maschinencode-Bereich außerhalb des zulässigen Bereiches zuzuweisen.
I/O DEVICE ERROR	Fehler bei der Datenübertragung oder Fehler der Kontrollsumme während eines I/O-Vorgangs.
OTHER ERROR	Andere Fehler.

# 13. ASSEMBLER

## **Referenz**

Es folgt eine Liste der Fachvokabeln, die häufig beim Umgang mit Maschinensprachen-Programmen verwendet werden.

## **Assemblieren, übersetzen:**

Ein Quellprogramm der Assemblersprache in eine Maschinsprache übersetzen. Ein übersetztes Maschinencode-Programm wird "Objektprogramm" oder kurz "Objekt" genannt.

## **Assembler:**

Übersetzungsprogramm zur Übersetzung eines Quellprogramms in ein Objektprogramm.

## **Generieren:**

Ein Objekt aus einem mnemonischen Code erzeugen.

## **Assemblieren von Hand:**

Manuelle Übersetzung eines Quellprogramms ohne einen Assembler.

## **Maschinsprache:**

Eine Computersprache, die direkt von einer Maschine interpretiert und deren Befehle ausgeführt werden. Als Hexadezimalcode dargestellt (intern als Binärcode verarbeitet.)

## **Mnemonische Code:**

Symbole, die dem Programmierer helfen sollen, sich die Befehle für Maschinencode-Anweisungen zu behalten. Z.B. die Abkürzung "ADD" für einen zusätzlichen Befehl (additional command). Eine Sprache, deren mnemonische Anweisungen eine spezifische Übereinstimmung mit dem Maschinencode haben, wird "Assemblersprache" genannt.

## **Objektprogramm:**

Ein vollständig assembliertes Programm, das bereit zum Laden in einen Computer ist. Die Bezeichnung bezieht sich im allgemeinen auf ein Maschinencode-Programm, das von einem Quellprogramm übersetzt wurde. Manchmal einfach als "Objekt" bezeichnet. ("Objekt" kann sich entweder auf einen individuellen Maschinencode beziehen, der nach einer Übersetzung entstand, oder es kann sich auf ein ganzes Maschinensprachen-Programm beziehen.)

## **Pseudobefehl:**

Eine Folge von Assembler-Steuerungsbefehlen, die nicht in ein Maschinencode-Programm übersetzt werden. Eine derartige Folge wird verwendet, um eine Adresse zu bestimmen, ein Maschinencode-Programm zu speichern oder Daten zu generieren.

## **Quellprogramm:**

Ein Programm, das in einem mnemonischen Code (Assemblersprache) geschrieben wurde. Ein Maschinencode-Programm ist eine Übersetzung eines Quellprogramms.

# Programmieren mit Assembler

Vor der Beschreibung des Assemblers soll zunächst ein Beispiel-Programm assembliert und das daraus entstehende Objekt (das Maschinencode-Programm) ausgeführt werden.

Zunächst muß allerdings darauf hingewiesen werden, daß verschiedene störende Situationen bei der Ausführung von Programmen auftreten können. Wenn das Maschinencode-Programm einen oder mehrere Fehler enthält, können die folgenden Fehlersituationen auftreten:

- ① Der Computer führt das Programm mit einer Endlosschleife aus und reagiert nicht mehr auf das Drücken von Tasten.  
Zum Unterbrechen dieser Endlosschleife muß der RESET-Schalter gedrückt werden.
- ② Das Programm führt zur Anzeige von zufälligen oder unsinnigen Zeichen oder zeigt andere Auffälligkeiten.  
In einigen Fällen kann das Programm mit der Taste **BREAK** gestoppt werden, aber in anderen Fällen muß der RESET-Schalter gedrückt werden.
- ③ Teile oder das ganze Programm werden zerstört oder gehen verloren.  
Hier liegt ein Speicherfehler vor. Es kann auch zur Zerstörung von Quellprogrammen (TEXT), BASIC-Programmen oder aller Daten des Computers kommen, einschließlich dem Maschinencode-Programm.

Diese Probleme können einzeln oder gleichzeitig alle zusammen auftreten. Falls eines dieser Probleme auftritt und Sie nicht feststellen können, was gerade vorgeht, drücken Sie den RESET-Schalter, um den gesamten Speicherinhalt zu löschen.

- Die obigen Probleme ① und ② werden "Runaway-Programme" genannt.

Bei der Ausführung eines Maschinencode-Programms können eines oder alle diese Probleme auftreten, wenn das Programm auch nur einen einzigen Fehler enthält. Aus diesem Grund wird empfohlen, alle Programme und andere Informationen auf Kassette oder einem anderen Speichermedium zu sichern, bevor ein Programm in Maschinensprache ausgeführt wird. Wenn ein Programm oder andere Daten im Speicher durch einen Absturz eines Maschinencode-Programms verloren gegangen sind, können sie nicht wiederhergestellt werden.

## Eingabe eines Quellprogramms

Ein Quellprogramm wird dem Computer wie ein TEXT-Programm eingegeben (siehe Seite 99).

**SHIFT** + **TEXT** drücken, um die TEXT-Betriebsart zu wählen und das Hauptmenü zur Anzeige zu bringen.

```
*** TEXT EDITOR ***
Edit   Del   Print Cmt
Sio    File  Basic
```

Beim Drücken von **E** bei angezeigtem Hauptmenü wird die Editierfunktion gewählt.

```
TEXT EDITOR
<
```

Wenn im TEXT-Bereich bereits ein TEXT-Programm vorhanden ist, wird bei angezeigtem Hauptmenü **D** (Delete) und dann **Y** gedrückt, um das vorhandene Programm zu löschen.

## Beispiel-Programm

Mit dem folgenden Programm werden die Hexadezimalzahlen von 20H bis 9FH in den Speicherbereich von 0400H bis 047FH geladen (das H am Ende zeigt an, daß es sich um eine hexadezimale Notation handelt):

Tastenbedienung		Anzeige	
10	<code>TAB</code> ORG <code>TAB</code> 0100H <code>←</code>	10	ORG 0100H
20	START: LD <code>TAB</code> A, 20H <code>←</code>	20	START:LD A, 20H
30	<code>TAB</code> LD <code>TAB</code> HL, 0400H <code>←</code>	30	LD HL, 0400H
40	LBL: <code>TAB</code> LD <code>TAB</code> (HL), A <code>←</code>	40	LBL: LD (HL), A
50	<code>TAB</code> INC <code>TAB</code> A <code>←</code>	50	INC A
60	<code>TAB</code> INC <code>TAB</code> HL <code>←</code>	60	INC HL
70	<code>TAB</code> CP <code>TAB</code> 0A0H <code>←</code>	70	CP 0A0H
80	<code>TAB</code> JP <code>TAB</code> NZ, LBL <code>←</code>	80	JP NZ, LBL
90	<code>TAB</code> RET <code>←</code>	90	RET
100	<code>TAB</code> END <code>←</code>	100	END

Mit der Taste `SPACE` oder mit `TAB` können eine oder mehrere Leerstellen eingefügt werden.

Beschreibung des Beispiel-Programms:

- 10: (Laden des Objekts in den Bereich, der mit der Adresse 0100H beginnt.)
- 20: 20H in das Register A laden.
- 30: 0400H in das Registerpaar HL laden.
- 40: Den Inhalt von Register A in eine Adresse laden, die durch das Registerpaar HL bestimmt wird.
- 50: Den Wert von Register A um eins erhöhen und das Ergebnis in Register A laden.
- 60: Den Wert des Registerpaars HL um eins erhöhen und das Ergebnis in HL laden.
- 70: Den Inhalt von Register A mit dem Wert A0H vergleichen (A0H vom Inhalt von A subtrahieren).
- 80: Wenn das Ergebnis des letzten Vorgangs nicht Null ist (Inhalt von A ist nicht gleich A0H), erfolgt ein Sprung auf das Label LBL. Das Label wird in eine Adresse (0105H) übersetzt.
- 90: Rückkehr (Rückkehr von der Subroutine).
- 100: (Ende des Quellprogramms)

Die Zeilen 10 und 100 dieses Quellprogramms werden Pseudobefehle genannt. Sie dienen der Steuerung des Assemblers und werden nicht in Maschinencodes (Objekte) umgewandelt (siehe Seite 132).

Nach der Eingabe aller Zeilen dieses Beispiel-Programms muß eine Fehlersuche ausgeführt werden. Vor der Assemblierung des Quellprogramms muß ein Bereich zum Speichern für den Maschinencode zugewiesen werden, andernfalls ist es nicht möglich, das Quellprogramm zu assemblieren.

### Zuweisen eines Maschinencode-Bereiches

Zur Zuweisung eines Maschinencode-Bereiches wird in der Monitor-Betriebsart der USER-Befehl verwendet (siehe Seite 113).

Zunächst wird die Monitor-Betriebsart gewählt.

`BASIC` MON `←`

```
MACHINE LANGUAGE MONITOR
*
```

Nun wird mit dem USER-Befehl der Maschinencode-Bereich zugewiesen, in diesem Beispiel von 0100H bis 04FFH.

USER 04FF 

```
MACHINE LANGUAGE MONITOR
*USER04FF
FREE:0100-04FF
*
```

Der Computer zeigt den zugewiesenen Maschinencode-Bereich (Benutzerbereich) an.

### Assemblieren des Quellprogramms

Nun kann das Quellprogramm dieses Beispiels in Maschinencode umgewandelt werden.

Die Assembler-Funktion wählen.

 + 

(Die Größe des Arbeitsbereiches kann anders sein als die in diesem Beispiel dargestellte. Für Details siehe Seite 129.)

```
***** ASSEMBLER *****
user area=0100H-04FFH
work area=29221bytes
< Asm Display Print >
```

Die Taste  drücken, um mit der Assemblierung zu beginnen.

```
***** ASSEMBLER *****

--- assembling ---
```

Nach Beendigung der Assemblierung erscheint eine Anzeige wie die rechts abgebildete (Erläuterungen zu diesen Informationen finden sich auf Seite 129).

```
object:0100H-010DH
size :000EH( 14)bytes
label : 2
error : 0 complete !
```

Wenn während der Assemblierung ein Fehler auftritt, zeigt der Computer die entsprechende Fehlermeldung und die Zeilennummer an, bei der der Fehler auftrat (siehe Seite 130).

```
***** ASSEMBLER *****
* FORMAT ERROR (1)
0105 ***** 40
LBL: LD HL),A
```

In diesem Fall zum Editor zurückgehen und das Quellprogramm korrigieren.

### Überprüfung des generierten Objektprogramms

Das generierte Objektprogramm wird mit dem Monitor überprüft. Das Programm ist in einem Bereich von 0100H bis 010DH gespeichert.

Zur Einstellung der Monitor-Betriebsart  drücken (oder  MON  eingeben).

```
MACHINE LANGUAGE MONITOR
*
```

Das Objektprogramm mit dem D-Befehl ausgeben.

D0100 

Der Computer zeigt den abgebildeten Speicherauszug des Objektprogramms an.

0100	:	3E	20	21	00	>	!	.
(88)		04	77	3C	23	.	w	<#
		FE	A0	C2	05	.	1	.
		01	C9	00	00	.	8	.

(Teile des letzten Speicherinhaltes können in einem Bereich beginnend mit 010DH (C9) angezeigt werden.)

(88) ist die Kontrollsumme (siehe Seite 116).

### Ausführen des Objektprogramms (Maschinencode-Programms)

Nun soll das generierte Objektprogramm ausgeführt werden. Dafür wird der Monitorbefehl G (GOSUB) verwendet.

Die Eingabeaufforderung der Befehlseingabe-Zeile der Monitor-Betriebsart zur Anzeige bringen.

**BREAK**

```
*

```

Den folgenden G-Befehl verwenden, um das Objektprogramm laufen zu lassen.

G0100 

Nach der Ausführung wird wieder die Befehlseingabe-Zeile des Monitors angezeigt.

```
* G0100
*
```

Nun wird das Ergebnis der Programmausführung überprüft.

D0400 

0400	:	20	21	22	23	!	"	#
(78)		24	25	26	27	\$	%	'
		28	29	2A	2B	(	)	*+
		2C	2D	2E	2F	,	-	/



Die hexadezimalen Zahlen von 20H bis 9FH wurden in den Adreßbereich von 0400H bis 047FH geschrieben.

0410	:	30	31	32	33	0	1	2	3
(78)		34	35	36	37	4	5	6	7
		38	39	3A	3B	8	9	:	;
		3C	3D	3E	3F	<	=	>	?

## Codieren und Editieren eines Quellprogramms

Der Assembler des Computers übersetzt (assembliert) das im TEXT-Bereich gespeicherte Quellprogramm in ein Maschinencode-Programm. Das assemblierte Maschinencode-Programm wird sequentiell in einen Speicherbereich geladen, der an der spezifizierten Adresse beginnt.

In diesem Abschnitt werden die Konventionen und Regeln (Eingabeformate u.a.) beschrieben, die bei der Erstellung eines Quellprogramms verwendet werden.

## Konfiguration des Quellprogramms

Jede Zeile eines Quellprogramms enthält normalerweise eine einzige Anweisung. Ein Programm besteht im allgemeinen aus einigen Zeilen. Ein Quellprogramm in der Assemblersprache beginnt mit einer ORG-Anweisung und endet mit einer END-Anweisung (die ORG- und die END-Anweisungen können weggelassen werden).

### Beispiel:

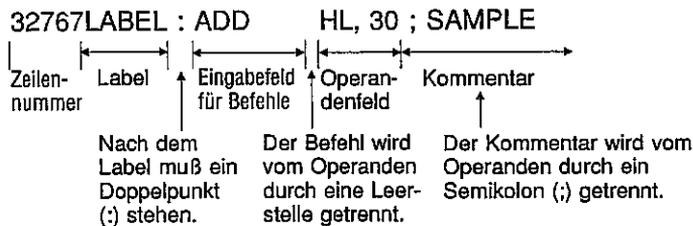
```
10   ORG   0100H
    {
100  END
```

- Die ORG-Anweisung dient der Spezifizierung der ersten Adresse des Speicherbereiches, in dem das generierte Maschinencode-Programm gespeichert werden soll. Dies bedeutet, daß die Zeilen des Maschinencode-Programms der Reihe nach gespeichert werden, beginnend von der Adresse, die mit der ORG-Anweisung bestimmt wurde. Wenn keine Adresse bestimmt wird, nimmt der Computer die Adresse 0100H als erste Adresse an.
- Die END-Anweisung zeigt das Ende des Quellprogramms an. Der Computer beendet die Assemblierung, wenn er diese Anweisung im Quellprogramm erreicht.

Diese Anweisungen dienen der Steuerung des Assemblers; sie werden nicht in einen Maschinencode umgewandelt (siehe "Pseudobefehle" auf Seite 132).

## Zeilenkonfiguration (Anweisungen)

Jede Zeile des Quellprogramms besteht aus einer Zeilennummer, einem Label, einem Befehl, einem Operanden, einem Kommentar bzw. einem Pseudobefehl.



- Eine Zeile kann aus bis zu 254 Zeichen bestehen, einschließlich dem Kommentar.
- Klein- und Großbuchstaben werden wie Großbuchstaben verarbeitet, außer wenn sie bei Operanden oder Kommentaren benutzt werden.

### ① Zeilennummern

Wenn eine Zeilennummer außerhalb des zulässigen Bereiches von 1 bis 65279 eingegeben wird, erfolgt die Fehlermeldung "LINE NO. ERROR".

### ② Labels

- Ein Label kann direkt nach der Zeilennummer eingefügt werden (zwischen der Zeilennummer und dem Label darf keine Leerstelle sein, andernfalls erfolgt ein Fehler).
- Das Label kann aus bis zu sechs Zeichen bestehen. Bei mehr als sechs Zeichen erfolgt ein Fehler (siehe Seite 136).
- Die folgenden Zeichen können für Label verwendet werden:
  - Buchstaben: A bis Z (a bis z werden wie A bis Z gelesen).
  - Zahlen: 0 bis 9
  - Symbole: [, ], @, ? und \_

- Das erste Zeichen eines Labels muß allerdings unbedingt ein Buchstabe oder ein Symbol sein (eine Zahl kann nicht von der Zeilennummer unterschieden werden).
- Als Label können genau die gleichen wie die im folgenden aufgelisteten einzelnen Zeichen oder Zeichenpaare für Register oder Bedingungscode nicht verwendet werden:
  - Einzelregister: A, B, C, D, E, H, L, I, R
  - Registerpaare: AF, BC, DE, HL, IX, IY, SP
  - Bedingungscode: NZ, Z, NC, C, PO, PE, P, M
- Ein Label muß von einem Doppelpunkt (:) gefolgt werden, andernfalls entsteht ein Fehler. Eine Ausnahme bildet die Definition eines Wertes für ein Label mit dem Pseudobefehl EQU; in diesem Fall muß nach dem Label kein Doppelpunkt folgen (siehe Seite 135).
- Wenn kein Label benötigt wird, muß eine oder mehrere Leerstellen zwischen der Zeilennummer und dem folgenden Befehlsword eingefügt werden. Zum Einfügen von Leerstellen kann die Taste `SPACE` oder `TAB` verwendet werden.

### ③ Eingabefeld für Befehle (OP-Code)

- In dem Eingabefeld für Befehle kann ein Z80-Befehl als mnemonisches Symbol eingegeben werden. Auch einer der Pseudobefehle, die ab Seite 132 aufgelistet sind, kann hier eingefügt werden. Ein Befehl ist Teil einer Anweisung, die Anweisungscode oder OP-Code genannt wird.
- Der eingegebene Befehl muß durch eine oder mehrere Leerstellen von dem folgenden Operanden getrennt werden. Zum Einfügen von Leerstellen kann die Taste `SPACE` oder `TAB` verwendet werden.

### ④ Operandenfeld

Als Operanden werden Register, Adressen oder Konstanten bezeichnet, die bei der Befehlsgangausführung (OP-Code) verwendet werden.

- Zur Trennung von Operanden werden Kommas (,) verwendet.
- Jeder Operand kann aus bis zu 32 Zeichen bestehen.
- Die folgenden Arten von Konstanten können ebenfalls als Operanden benutzt werden:

#### [Numerische Konstanten]

Binäre, dezimale oder hexadezimale Zahlen:

Binäre Zahlen: Dargestellt als Folge von 1 und 0, mit einem "B" am Ende.

Beispiele: 10111100B, 100000B

Dezimalzahlen: Dargestellt als Basis 0 bis 9

Beispiele: 188, 32

Hexadezimale Zahlen: Dargestellt durch Dezimalzahlen 0 bis 9 sowie den Großbuchstaben A bis F; mit einem "H" am Ende.

Wenn eine hexadezimale Zahl mit einem Buchstaben beginnt, muß am Anfang eine "0" stehen, um sie von einem Befehl zu unterscheiden.

Beispiele: 0BCH, 20H

### [Zeichenfolge-Konstanten]

Zeichenfolgen bei Operanden müssen in einfachen Hochkommas (') eingeschlossen werden. ASCII-Darstellungen von Zeichen werden bei Operanden als Konstanten verwendet.

Beispiel:

(Spezifizierung)	(Zeichenfolge)	(Konstanten)
'A'	A	41H
'AB'	AB	41H, 42H
'B"C'	B"C	42H, 27H, 43H
""D'	'D	27H, 44H
'E'''	E'	45H, 27H
""	'	27H
''	(NULL)	00H

### [Label-Konstanten]

Wenn eine Konstante für ein Label mit dem Befehl EQU definiert wird, kann dieses Label als Konstante in einem Operanden benutzt werden (siehe Seite 135).

- Ausdrücke (einschließlich arithmetischer Operatoren) können als Operanden verwendet werden. Die folgenden Zeichen und arithmetischen Operatoren können in Operanden benutzt werden; dabei hat jedoch kein Operator Vorrang gegenüber einem anderen.

Vorzeichen: positiv (+), negativ (-)

Operatoren: \*, /, +, -

- Der Computer führt interne Vorgänge mit 16-Bit-Daten aus. Eine Kapazitätsüberschreitung wird nicht beachtet (es tritt kein Fehler auf). Das Objekt wird mit dem 8-Bit- oder 16-Bit-Ergebnis generiert.
- Bei Anweisungen mit Ausdrücken überprüft der Computer nicht die Richtigkeit des eingegebenen Ausdrucks.

Beispiele:

LD A, 4142H → Gelesen als LD A, 42H

DB 1234H → Gelesen als DB 34H

### ⑤ Kommentare

Jeder Zeile eines Quellprogramms kann ein Kommentar folgen, getrennt durch ein Semikolon (;). Der Anteil der Zeile von einem Semikolon bis zum Ende der Zeile wird als Kommentar angesehen und nicht in Maschinencode (Objekt) übersetzt.

### Löschen eines Quellprogramms

Das Hauptmenü in der TEXT-Betriebsart zur Anzeige bringen und  drücken, um die Lösch-Funktion zu wählen. Der Computer fragt zur Sicherheit, ob der Inhalt des TEXT-Bereiches gelöscht werden soll. (Wenn kein Programm im TEXT-Bereich gespeichert ist, reagiert der Computer nicht auf das Drücken von )

TEXT DELETE OK? (Y)

Zum Löschen aller Informationen aus dem TEXT-Bereich die Taste  drücken. Der Computer geht wieder auf das Hauptmenü der TEXT-Betriebsart zurück. Beim Drücken einer anderen Taste als  geht der Computer ohne etwas zu löschen auf das Hauptmenü der TEXT-Betriebsart zurück.

### Eingabe eines Quellprogramms

Das Hauptmenü in der TEXT-Betriebsart zur Anzeige bringen und  drücken, um die Editierfunktion zu wählen.

```
TEXT EDITOR
<
```

Durch Drücken der Taste  bzw.  kann der Inhalt des TEXT-Bereiches dargestellt werden, z.B. ein Quellprogramm. Wenn nichts gespeichert ist, ändert sich die Anzeige nicht.

Ein neues Programm kann erst dann in den TEXT-Bereich geladen werden, wenn der vorhandene Inhalt vollkommen gelöscht wurde.  drücken, um auf das Hauptmenü zurückzugehen, die Lösch-Funktion wählen und die Inhalte des TEXT-Bereiches löschen. Folgen Sie dabei den Schritten, die im obigen Abschnitt über das Löschen von Quellprogrammen beschrieben wurden.

Eingabe eines Quellprogramms:

- ① Die Zeilennummer eingeben.
- ② Wenn kein Label benötigt wird, kann durch Drücken von  eine oder mehrere Leerstellen eingefügt werden. Der Cursor bewegt sich wieder auf das Eingabefeld für Befehle. (Die Taste  kann ebenfalls anstelle von  verwendet werden.)  
Ein Label wird direkt nach der Zeilennummer eingegeben, ohne eine Leerstelle. Am Ende des Labels muß ein Doppelpunkt (:) stehen. Nach dem Doppelpunkt können nach Belieben eine oder mehrere Leerstellen eingefügt werden.
- ③ Einen Befehl eingeben. Wenn nach dem Befehl ein Operand folgt, muß er durch eine oder mehrere Leerstellen von dem Befehl getrennt werden; dazu  oder  drücken.
- ④ Die Operanden eingeben.  
Die Operanden werden durch Kommas (,) getrennt.
- ⑤ Wenn diese Zeile mit einem Kommentar versehen werden soll, muß vor dem Kommentar ein Semikolon (;) eingegeben werden.
- ⑥ Nach Eingabe der gesamten Zeile wird  gedrückt, um die Zeile zu speichern. Der Cursor verschwindet beim Drücken von .

Zur Eingabe der folgenden Zeile werden die obigen Schritte wiederholt. (Siehe das Eingabebeispiel auf Seite 122.)

## Assemblieren

In diesem Abschnitt wird ausführlich beschrieben, wie ein Quellprogramm assembliert wird, das in der TEXT-Betriebsart eingegeben wurde. Eine Voraussetzung dabei ist, daß das Beispiel-Programm von Seite 122 bereits im Computer geladen ist. Andernfalls muß es vor dem Ausführen der folgenden Schritte geladen werden.

### Das Menü der Assembler-Funktion

Um ein Quellprogramm zu assemblieren, muß zunächst die Assembler-Funktion aktiviert werden.

 + 

Das rechts abgebildete Menü der Assembler-Funktion erscheint.

```
***** ASSEMBLER *****
user  area=0100H-04FFH
work  area=29221bytes
< Asm Display Print >
```

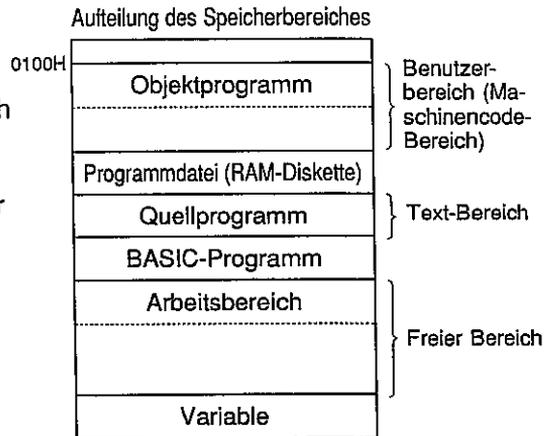
Benutzerbereich (Maschinencode-Bereich):  
Adressen 0100H bis 04FFH  
Größe des Arbeitsbereiches: 29221 Byte

-  zur Ausführung des Assemblierens.
-  zur Anzeige des Assemblerprotokolls.
-  zum Ausdruck des Assemblerprotokolls.

- Das Menü zeigt auf der zweiten Zeile den zugewiesenen Maschinencode-Bereich. Zum Zuweisen des Maschinencode-Bereiches wird in der Monitor-Betriebsart der USER-Befehl verwendet. Wenn kein Maschinencode-Bereich zugewiesen wurde oder der Bereich zu klein ist, um das Objekt zu speichern, wird während dem Assemblieren eine Fehlermeldung (NOT RESERVED oder USER AREA OVER) angezeigt. In diesem Fall **BASIC** MON **←** eingeben, um die Monitor-Betriebsart zu wählen und den Maschinencode-Bereich zuweisen bzw. vergrößern mit dem USER-Befehl.
- In der dritten Zeile des Assembler-Menüs wird die Größe des vorhandenen Arbeitsbereiches in Byte angegeben. Diese Information zeigt die Byteanzahl des freien Bereiches im Speicher; der Wert entspricht der Anzahl, die mit dem FRE-Befehl von BASIC erhalten wird.  
Der für den Umwandlungsvorgang im Computer benötigte Arbeitsbereich wird automatisch im freien Bereich zugewiesen. Wenn der Arbeitsbereich nicht zugewiesen werden kann, wird eine Fehlermeldung (WORK AREA OVER) angezeigt. In diesem Fall den freien Bereich durch Löschen von vorhandenen BASIC-Programmen oder anderen Daten vergrößern oder den Maschinencode-Bereich verkleinern.

**Hinweis:**

Ein Fehler tritt auf, wenn nicht wenigstens 307 Byte für den freien Bereich zugewiesen sind, während sich der Computer in der Assembler-Betriebsart befindet. Wenn ein Quellprogramm Label enthält, stellt der Assembler einen Label-Arbeitsbereich mit der notwendigen Größe bereit. Ein Fehler tritt auf, wenn der Assembler diesen notwendigen Bereich nicht zuweisen kann.



**Assemblieren**

**Erfolgreiches Assemblieren**

Zum Starten des Assemblierens wird **A** bei angezeigtem Assembler-Menü gedrückt.

```
***** ASSEMBLER *****
--- assembling ---
```

Während dem Assemblieren wird "--- assembling---" angezeigt. Nach der Beendigung des Vorgangs wird "complete !" sowie der Objektbereich und seine Größe, die Anzahl der Labels und die Anzahl der Fehler angezeigt.

```
object:0100H-010DH
size :000EH( 14)bytes
label : 2
error : 0 complete !
```

Objektbereich: 0100H bis 010DH  
Größe des Objektes: 0EH (14) Byte  
Anzahl der Labels: 2  
Anzahl der Fehler: 0



Bei der Anzeige der Meldung "complete!" wird **C•CE** gedrückt, um auf die Monitor-Betriebsart zurückzugehen. In der Monitor-Betriebsart können Sie das assemblierte Objektprogramm mit dem D-Befehl überprüfen oder es mit dem G-Befehl ausführen lassen.

### Nicht erfolgreiches Assemblieren

Wenn während dem Assemblieren im Quellprogramm ein Fehler gefunden wird, beendet der Assembler den Vorgang und zeigt eine entsprechende Fehlermeldung und die Zeilennummer an, bei der der Fehler gefunden wurde. Zum Fortsetzen des Assemblierens **↓** drücken.

#### Beispiel:

Es wird angenommen, daß das Beispiel-Programm auf Seite 122 in den Zeilen 50 und 80 je einen Fehler enthält, wie im folgenden abgebildet.

```
50      INB      A          ...."INC A" ist richtig.
80      JP       NZ, KBL   ...."JP NZ, LBL" ist richtig.
```

Die Taste **A** bei angezeigtem Assembler-Menü drücken, um das Programm mit den angegebenen Fehlern zu assemblieren.

Beim Auffinden des ersten Fehlers wird die rechts abgebildete Fehlermeldung angezeigt.

```
***** ASSEMBLER *****
* OPCODE ERROR
0106 ***** 50
      INB      A
```

Adresse                      Befehl            Operand  
 Objekt (siehe Anmerkung)                      Zeilennummer

Fehlermeldung (zeigt einen OP-Code-Fehler an)

#### Anmerkung:

Wenn der Assembler das richtige Objektprogramm nicht generieren kann, weil im Quellprogramm ein Fehler vorliegt, wird nach der entsprechenden Adresse eine Folge von Sternchen (\*\*\*) angezeigt.

Die Taste **↓** drücken, um das Assemblieren fortzusetzen. Nun wird die Fehlermeldung für den zweiten Fehler in Zeile 80 angezeigt.

```
***** ASSEMBLER *****
* UNDEFINED SYMBOL
0109 ***** 80
      JP       NZ, KB
```

Fehlermeldung (ein undefiniertes Symbol wurde für ein Label verwendet)

Die Taste **↓** erneut drücken.

Die letzte Anzeige des Assemblers erscheint, diesmal allerdings ohne die Meldung "complete!".

```
object:0100H-010CH
size :000DH( 13)bytes
labe : 2
error : 2
```

Bei der letzten Anzeige **C•CE** drücken, um auf das Assembler-Menü zurückzugehen.

#### Hinweise:

- Der Assembler ignoriert die Anweisung von Zeile 50 und nimmt an, daß das Label von Zeile 80 die Adresse 0000H spezifiziert; an dieser Stelle wird das Programm dieses Beispiels assembliert.
- Wenn ein Fehler im Quellprogramm aufgefunden wird, weist auch das generierte Objekt Fehler auf. Beim Ausführen eines derartigen Objektprogramms kann das Programm außer Kontrolle geraten oder zu Zerstörungen des Speicherinhaltes führen. Das Quellprogramm muß unbedingt korrigiert und neu assembliert werden, damit das Objektprogramm fehlerfrei ausgeführt werden kann.

## Anzeige des Assemblerprotokolls

Mit der Anzeige-Möglichkeit des Assembler-Menüs kann ein Objektprogramm überprüft werden, bevor das Quellprogramm übersetzt wird. Das Assemblerprotokoll enthält das zu generierende Maschinencode-Programm, seine Adressen und weitere Objekt-Informationen.

Beim Drücken von **D** bei angezeigtem Assembler-Menü wird die erste Zeile des Assemblerprotokolls dargestellt. Durch Drücken von **↓** können die weiteren Zeilen eingesehen werden.

Laden Sie zur Übung das Beispiel-Programm von Seite 122, wenn es nicht bereits geladen ist, und überprüfen Sie sein Assemblerprotokoll.

Bei angezeigtem Assembler-Menü **D** drücken.

**** ASSEMBLE LIST ****			
0100			10
	ORG	0100H	
0100	3E20		20
Adresse	Objekt		Zeilennummer
		Quellprogramm	

Wenn das Objektfeld leer ist, gibt es kein Objekt, das generiert werden kann. Wenn der Maschinencode mehr als acht Stellen enthält, werden die übrigen Stellen auf der folgenden Zeile dargestellt.

Die Taste **↓** mehrfach drücken, um die folgenden Zeilen des Assemblerprotokolls einzusehen.

**** ASSEMBLE LIST ****			
0100			10
	ORG	0100H	
0100	3E20		20
	START: LD	A, 20H	
0102	210004		30
	LD	HL, 04	
	00H		
0105	77		40
	LBL: LD	(HL),	
	A		
0106	3C		50
	INC	A	
0107	23		60
	INC	HL	
0108	FEA0		70
	CP	0A0H	
010A	C20501		80
	JP	NZ, LB	
	L		
010D	C9		90
	RET		
010E			100
	END		
**** SYMBOL TABLE ****			
START	: 0100	LBL	: 0105
object	: 0100H-010DH		
size	: 000EH ( 14) bytes		
label	: 2		
error	: 0	complete	!

Liste der Symbole (\*) →  
 Objekt-Adressen →  
 Größe des Objektprogramms →  
 Anzahl der Labels →  
 Anzahl der Fehler →

\* Die Liste der Symbole zeigt die Werte, die den Labels in hexadezimal zugewiesen wurden.

### Hinweise:

- **[C•CE]** drücken, um vom Assemblerprotokoll wieder auf das Assembler-Menü zurückzugehen.
- Mit der Anzeige-Möglichkeit können Sie das Assemblerprotokoll überprüfen; der Maschinencode des Protokolls kann damit aber nicht in den Maschinencode-Bereich geladen werden. Zum Laden muß das Quellprogramm ordnungsgemäß assembliert werden.

### Ausdrucken des Assemblerprotokolls

Das Assemblerprotokoll kann mit der Drucker-Möglichkeit des Assembler-Menüs ausgedruckt werden. Den zusätzlichen Drucker CE-126P mit dem Computer verbinden, den Drucker einschalten und bei angezeigtem Assembler-Menü **[P]** drücken.

### Hinweise:

- Bei Wahl der Drucker-Option, ohne daß der Drucker CE-126P angeschlossen bzw. eingeschaltet ist, erscheint eine Fehlermeldung (\*PRINTER ERROR). In diesem Fall die Meldung mit **[C•CE]** löschen und den Drucker überprüfen.
- Ein Assemblerprotokoll kann ausgedruckt werden, unabhängig davon, ob auf der unteren rechten Seite des Displays "PRINT" angezeigt wird.
- Nach dem Ausdruck zeigt der Assembler den letzten Assembler-Bildschirm. **[C•CE]** drücken, um wie der auf das Assembler-Menü zurückzugehen.
- Das Protokoll wird in der gleichen Weise ausgedruckt, in der es auf dem Display angezeigt wird.
- Zum Abbrechen des Ausdrucks die Taste **[BREAK]** gedrückt halten, bis der Drucker stoppt. Auf dem Display wird "--- break ---" angezeigt. **[C•CE]** drücken, um auf das Assembler-Menü zurückzugehen.

## Pseudobefehle für den Assembler

Pseudobefehle dienen der Steuerung des Assemblers selbst und werden nicht in Maschinencode umgewandelt. Dieser Computer kennt die folgenden Pseudobefehle:

- **ORG:** Bestimmt die erste Adresse des Maschinencode-Bereiches.
- **DEFB/DB/DEFM/DM, DEFS/DS und DEFW/DW:** Definieren von Daten innerhalb der Operanden.
- **EQU:** Definieren von Labelwerten.
- **END:** Zeigt das Ende des Assemblierens an.

Im folgenden werden einige Konventionen und Regeln beschrieben, die bei den Erklärungen für Pseudobefehle verwendet werden.

Ausdruck:	Ausdrücke können Zahlen, Formeln, Labels oder "Zeichenfolgen" sein.
Formel:	Formeln können Zahlen, Labels oder irgendwelche arithmetischen Ausdrücke sein, bei denen Zahlen oder Labels verwendet werden.
{ }:	Wenn mehrere Elemente durch eine geschweifte Klammer zusammengefaßt werden, muß eines dieser Elemente gewählt werden.
[ ]:	Ein Element innerhalb eckiger Klammern bezeichnet eine optionale Anweisung.
[ ]...:	Punkte für Folgeinformationen nach eckigen Klammern bedeuten, daß das Element in den Klammern optional ist oder wiederholt werden kann.

## ORG — Beginn

- Format:** ORG Ausdruck  
**Wirkung:** Bestimmt die erste Adresse des Maschinencode-Bereiches.  
**Anmerkungen:**
- Der Ausdruck bestimmt die erste Adresse eines Bereiches, in dem der generierte Maschinencode gespeichert wird. Das Maschinencode-Programm wird sequentiell in einen Speicherbereich geladen, der an derjenigen Adresse beginnt, die mit diesem Ausdruck bestimmt wurde.
  - Wenn das Quellprogramm keine ORG-Anweisung enthält, nimmt der Assembler die Anweisung "ORG 100H" an; damit wird 100H zur ersten Adresse, von der aus der Maschinencode gespeichert wird.

### **Beispiel:**

ORG 0400H

Mit dieser Anweisung wird der Maschinencode in einen Bereich gespeichert, der bei der Adresse 0400H beginnt.

## DEFB/DB/DEFM/DM — Definiere Byte/Definiere Meldung

- Format:**
- |           |   |      |   |                         |
|-----------|---|------|---|-------------------------|
| [Label: ] | { | DEFB | } | Ausdruck [,Ausdruck]... |
|           |   | DB   | } |                         |
|           |   | DEFM | } |                         |
|           |   | DM   | } |                         |
- Wirkung:** Übersetzt eine Zeichenfolge oder das wertniedrigste Einzelbyte einer Zahl des Ausdrucks in den Maschinencode.  
**Anmerkungen:**
- Mit dieser Anweisung wird das wertniedrigste Einzelbyte einer gegebenen Zahl des Ausdrucks in den Maschinencode umgewandelt.

### **Beispiele:**

DEFB 1234H; Übersetzt 1234H in den Maschinencode "34H".

DB 1234; Übersetzt 1234 in den Maschinencode "D2H".

- Eine Zeichenfolge in einem Operanden muß in Hochkommas (') eingeschlossen werden. Sie kann aus bis zu 32 Zeichen bestehen. Einzelne Zeichen einer Operanden-Zeichenfolge werden als die entsprechenden ASCII-Codes übersetzt.

### **Beispiel:**

DEFM 'DATA'; Übersetzt die einzelnen Zeichen der Folge 'DATA' in die Maschinencodes 44H, 41H, 54H und 41H.

- Die Operanden werden durch Kommas (,) getrennt.

### **Beispiel:**

DB 32\*4+5, 'X2'; 85H, 58H und 32H werden im Maschinencode generiert.

### Beispiel-Programm

	Quellprogramm	Maschinencode-Programm
10	ORG 0100H	
20	LD HL, DATA	21 0C 01
30	LD DE, 300H	11 00 03
40	LD BC, 5	01 05 00
50	LDIR	ED B0
60	RET	C9
70	DATA: DB 'ABCDEFGH'	41 42 43 44
		45 46 47 48
80	END	

Die einzelnen Zeichen der Operanden-Zeichenfolge von Zeile 70 werden in die entsprechenden ASCII-Codes übersetzt.

Bei diesem Beispiel-Programm werden fünf Datenbyte kopiert, die sich in einem Bereich befinden, dessen erste Adresse durch das Label DATA spezifiziert ist; sie werden in einen Bereich kopiert, der mit der Adresse 300H beginnt. Dies bedeutet also, daß die Daten 41H, 42H, 43H, 44H und 45H an den Adreßbereich von 300H bis 304H kopiert werden.

### DEFW/DW — Definiere Wort

Format: [Label: ] {DEFW  
DW} Ausdruck [,Ausdruck]...

Wirkung: Übersetzt die wertniedrigsten zwei Byte einer Zahl oder einer Zeichenfolge des Ausdrucks (zwei Zeichen oder weniger) in den Maschinencode.

Anmerkungen: • Mit dieser Anweisung werden die wertniedrigsten zwei Byte einer Zahl übersetzt, die im Operanden spezifiziert wurde.

#### Beispiele:

DW 1234H; Übersetzt 1234H in die Maschinencodes 34H und 12H (in der Reihenfolge von wertniedrigstem und werthöchstem Byte).

DEFW 34H; Übersetzt 34H in die Maschinencodes 34H und 00H.

- Eine Zeichenfolge in einem Operanden muß von Hochkommas (') eingeschlossen sein. Es können bis zu zwei Zeichen für eine Zeichenfolge definiert werden.

#### Beispiele:

DEFW 'DA'; Übersetzt die Zeichenfolge 'DA' in den Maschinencode 41H und 44H.

DW 'Z'; Übersetzt die Zeichenfolge 'Z' in den Maschinencode 5AH und 00H.

- Die Operanden werden durch Kommas (,) getrennt.

#### Beispiel:

DW 'AB','CD', 5678H; Im Maschinencode werden 42H, 41H, 44H, 43H, 78H und 56H generiert.

## DEFS/DS — Definiere Speicher

Format: [Label:] {DEFS  
DS} Numerischer Ausdruck

Wirkung: Generierung von Null-Codes (00H) mit einer Anzahl, die im Operanden spezifiziert wird.

Anmerkungen: • Mit dieser Anweisung werden Null-Codes (00H) entsprechend der spezifizierten Byteanzahl generiert.

### Beispiel:

DS 12; Generiert 12 Byte mit 00H.

### Beispiel-Programm

	Quellprogramm	Maschinencode-Programm
10	ORG 0100H	
20	LD HL, DATA	21 10 01
30	LD DE, 300H	11 00 03
40	LD BC, 5	01 05 00
50	LDIR	ED B0
60	RET	C9
65	DS 4	00 00 00 00
70DATA:	DB 'ABCDEFGH'	41 42 43 44
		45 46 47 48
75NXT00:	DS 500H-NXT00	(00H wird an allen folgenden Adressen bis 04FFH eingefügt.)
80	END	

Dieses Beispiel-Programm ist genau so wie das auf Seite 134, es enthält aber die zusätzlichen Zeilen 65 und 75. Mit Zeile 65 wird ein Speicherbereich für späteren Gebrauch zugewiesen. Mit Zeile 75 werden Null-Codes (00H) eingefügt, um unnötigen Speicherinhalt zu löschen.

00H ist ein "Nulloperations-Code" (NOP), der den Computer anweist, nichts zu tun.

## EQU — EQU

Format: Label EQU Ausdruck

Wirkung: Dem Label wird ein Wert zugewiesen, der durch den Operanden spezifiziert wird.

Anmerkungen: • Mit dieser Anweisung wird dem Label ein Wert zugewiesen, der durch den Ausdruck gegeben ist.  
• Der Wert kann eine Zahl oder eine Zeichenfolge von ein oder zwei Byte sein.  
• Nach dem Label darf kein Doppelpunkt (:) stehen.

### Beispiel:

START EQU 1000H; Zuweisung des Wertes 1000H zum Label START; das Label kann dann als Konstante 1000H bearbeitet werden.

OK EQU 'Y'; Zuweisung des Wertes 59H zum Label OK.

## END — Ende

Format: END

Wirkung: Zeigt das Ende eines Quellprogramms an.

- Anmerkungen:
- Mit der END-Anweisung wird das Ende eines Quellprogramms bestimmt; der Assembler beendet an dieser Stelle den Umwandlungsvorgang. Informationen, die nach dieser Anweisung folgen, werden nicht mehr assembliert.
  - Wenn am Ende eines Quellprogramms keine END-Anweisung vorhanden ist, assembliert der Assembler bis zum letzten Teil des TEXT-Bereiches.

## Assemblerfehler

Dieser Abschnitt enthält eine Liste mit Fehlermeldungen, die während dem Assemblieren angezeigt werden können, sowie Erklärungen zu diesen Meldungen. Zum Löschen der Fehlermeldung **C•CE** drücken. Wenn das Assemblieren beim Auftreten eines Fehlers im Quellprogramm abgebrochen wird, kann zum Wiederaufnehmen die Taste **↓** gedrückt werden. Die Fehlermeldung wird ebenfalls gelöscht, wenn der Computer auf eine andere Betriebsart eingestellt wird.

Fehlerart	Beschreibung (Ursache)
OPECODE ERROR	Ungültiger OP-Code (Befehlscode).
FORMAT ERROR (1)	Ungültiges Trennzeichen für Operatoren
FORMAT ERROR (2)	Ungültiger Code (ASCII-Code 01H-1FH o.a.) oder Zeichen in einem Operanden (derartige Codes bzw. Zeichen können normalerweise allerdings nicht eingegeben werden).
FORMAT ERROR (3)	Ungültige Anzahl von Operanden.
FORMAT ERROR (4)	Ungültige Zeichen wurden in einem Label verwendet.
FORMAT ERROR (5)	Ein Label hat mehr als sechs Zeichen.
FORMAT ERROR (6)	Die Zeichenfolge im Operanden ist nicht in Hochkommas eingeschlossen.
FORMAT ERROR (7)	Die Anzahl der Zeichen in einer Anweisung oder einem einzelnen Operanden überschreitet 32 (z.B. Wert der Adresse o.a. in einem Operanden hat zu viele voranstehende Nullen.)
QUESTIONABLE OPERAND (1)	Ungültiger Operand.
QUESTIONABLE OPERAND (2)	Ungültige Bedingung (NZ, Z, NC o.a.)
QUESTIONABLE OPERAND (3)	Der Wert des Operanden überschreitet die zulässige Grenze.
QUESTIONABLE OPERAND (4)	Die Zeichenfolge im Operanden überschreitet die zulässige Länge von 32 Zeichen.
QUESTIONABLE OPERAND (5)	Versuch der Division durch Null.
QUESTIONABLE OPERAND (6)	Andere ungültige Werte oder Ausdrücke.
UNDEFINED SYMBOL	Ein nicht definiertes Symbol (Label) wurde verwendet.

Fehlerart	Beschreibung (Ursache)
MULTI DEFINE SYMBOL	Das gleiche Symbol (Label) wurde mehr als einmal definiert.
FILE NOT EXIST	Das zu assemblierende Programm befindet sich nicht im TEXT-Bereich.
USER AREA OVER	Das Objekt konnte nicht in den Maschinencode-Bereich geladen werden. (Die erste Adresse des Objektbereiches, die mit der ORG-Anweisung spezifiziert wurde, liegt außerhalb des Maschinencode-Bereiches oder das Objekt hat beim Laden die Kapazität des Maschinencode-Bereiches überschritten.)
WORK AREA OVER	Die Größe des freien Bereiches ist zu klein für den notwendigen Arbeitsbereich zum Assemblieren (wenn sich der Computer in der Assembler-Betriebsart befindet oder assembliert).
PRINTER ERROR	Der Drucker ist nicht startbereit oder funktioniert nicht. (Der Drucker ist nicht angeschlossen, ausgeschaltet oder wegen einer entladenen Batterie nicht funktionsfähig.)



# COMPUTER- ANWEISUNGEN

Kapitel 5 enthält eine alphabetische Liste aller BASIC-Anweisungen für den PC-E220\*. Dieser Abschnitt dient zum Nachschlagen.

Der erste Abschnitt enthält eine alphabetische Liste der numerischen Funktionen und Pseudovariablen.

Der zweite Abschnitt besteht aus einer alphabetischen Liste aller anderen BASIC-Anweisungen.

\* Der PC-E220 wird im folgenden als "der Computer" bezeichnet.

# 14. WISSENSCHAFTLICHE UND MATHEMATISCHE BERECHNUNGEN

Der Computer ist mit zahlreichen eingebauten Funktionen für wissenschaftliche, mathematische und statistische Berechnungen ausgestattet. Eine alphabetische Liste folgt weiter unten. All diese Funktionen können bei der Anwendung des Computers im RUN-Betrieb als Teil von Berechnungen benutzt werden bzw. auch als BASIC-Anweisungen in einem Programm.

Für trigonometrische Funktionen können nach Bedarf Altgrad-, Bogenmaß- oder Neugrad-Werte eingegeben werden:

**ALTGRAD:** Stellen Sie den Computer auf den Altgrad-Eingabe-Modus durch Tippen von DEGREE (auf der Statuszeile im Display erscheint DEG). Dies ist die Grundeinstellung.

**BOGENMASS:** Stellen Sie den Computer in den Bogenmaß-Eingabe-Modus durch Tippen von RADIAN (auf der Statuszeile im Display erscheint RAD).

**NEUGRAD:** Stellen Sie den Computer in den Neugrad-Eingabe-Modus durch Tippen von GRAD (auf der Statuszeile im Display erscheint GRAD).

Diese drei Modi (DEG, RAD und GRAD) können auch innerhalb eines Programms eingestellt werden. Ist ein Modus eingestellt, müssen alle trigonometrischen Funktionen der Einstellung des Computers angepaßt werden (Altgrad-, Bogenmaß- oder Neugrad-Werte), bis der Modus von Hand oder innerhalb eines Programmes geändert wird. Die folgenden Beispiele gelten für Direkt-Eingabe der in Altgrad eingegebenen Funktionen.

Die meisten Funktionen können auch durch Drücken der entsprechenden Funktionstaste eingegeben werden. Funktionen mit einem Stern (\*) haben keine entsprechende Taste und müssen mit der Tastatur eingegeben werden.

**\*ABS**

|x|

**Funktion:** Absoluter Wert**Anmerkungen:** Zeigt den absoluten Wert des numerischen Arguments an. Der absolute Wert ist der Wert einer Ziffer unabhängig von seinem Vorzeichen. ABS-10 ist 10.**Beispiel:** ABS -10 

10

**ACS** $\cos^{-1}x$ **Funktion:** Reziproker oder Arcuscosinus**Anmerkungen:** Zeigt den Arcuscosinus des numerischen Arguments an. Der Arcuscosinus ist der Winkel, dessen Cosinus gleich dem Ausdruck ist. Der Wert ist abhängig von der Einstellung in Altgrad, Bogenmaß oder Neugrad.**Beispiel:** DEGREE   
ACS -0.5 

120

**AHC** $\cosh^{-1}x$ **Funktion:** Reziproker hyperbolischer Cosinus (Areafunktion)**Anmerkungen:** Zeigt den Cosinus der Areafunktion des numerischen Arguments an.**Beispiel:** AHC 10 

2.993222846

**AHS** $\sinh^{-1}x$ **Funktion:** Reziproker hyperbolischer Sinus**Anmerkungen:** Zeigt den Sinus der Areafunktion des numerischen Arguments an.**Beispiel:** AHS 27.3 

4.000369154

**AHT** $\tanh^{-1}x$ **Funktion:** Reziproker hyperbolischer Tangens**Anmerkungen:** Zeigt den Tangens der Areafunktion des numerischen Arguments an.**Beispiel:** AHT 0.7 

0.867300527

## ASN

$\sin^{-1}x$

**Funktion:** Reziproker oder Arcussinus

**Anmerkungen:** Zeigt den Arcussinus des numerischen Arguments an. Der Arcussinus ist der Winkel, dessen Sinus gleich dem Ausdruck ist. Der angezeigte Wert ist abhängig vom eingestellten Modus (Altgrad, Bogenmaß oder Neugrad).

**Beispiel:** DEGREE   
ASN 0.5

30

## ATN

$\tan^{-1}x$

**Funktion:** Reziprok oder Arcustangens

**Anmerkungen:** Zeigt den Arcustangens des numerischen Arguments an. Der angezeigte Wert ist abhängig vom eingestellten Modus (Altgrad, Bogenmaß oder Neugrad).

**Beispiel:** DEGREE   
ATN 1

45

## COS

$\cos x$

**Funktion:** Cosinus

**Anmerkungen:** Zeigt den Cosinus des Winkelarguments an. Der angezeigte Wert ist abhängig vom eingestellten Modus (Altgrad, Bogenmaß oder Neugrad).

**Beispiel:** DEGREE   
COS 120

-0.5

## \*CUB

$x^3$

**Funktion:** Kubikwert

**Anmerkungen:** Zeigt den Kubikwert des Arguments an.

**Beispiel:** CUB 3

27

## CUR

$\sqrt[3]{x}$

**Funktion:** Kubikwurzel

**Anmerkungen:** Zeigt die Kubikwurzel des Arguments an.

**Beispiel:** CUR 125

5

## DEG

$dd^{\circ}mm'ss'' \rightarrow ddd.dddd^{\circ}$

**Funktion:** Umwandlung von Altgrad/Minuten/Sekunden in eine dezimale Form

**Anmerkungen:** Wandelt das Argument eines Winkels in der DMS-Form (Altgrad, Minuten, Sekunden) in die DEG-Form (Dezimal-Grad) um. In der DMS-Form repräsentiert die ganze Zahl die Gradzahl, die erste und zweite Stelle die Minuten und die dritte und vierte Stelle hinter dem Komma repräsentieren die Sekunden. Jede weitere Stelle gibt Dezimalsekunden an.

**Beispiel:** DEG 30.5230  (30°52'30'') 30.875

## DMS

$ddd.dddd^{\circ} \rightarrow dd^{\circ}mm'ss''$

**Funktion:** Umwandlung Dezimal-Grad von Grad/Minuten/Sekunden

**Anmerkungen:** Wandelt das Winkelargument von DEG auf DMS um. (Siehe DEG.)

**Beispiel:** DMS 124.8055  124.48198 (124°48'19"8)

## EXP

$e^x$

**Funktion:** Exponentialfunktion

**Anmerkung:** Zeigt den Wert von e (2.718281828... die Basis des natürlichen Logarithmus) erhoben zum Wert des numerischen Arguments an. Die entsprechende Funktionstaste ist .

**Beispiel:** EXP 1.2  3.320116923

## FACT

$n!$

**Funktion:** Fakultät n

**Anmerkungen:** Zeigt die Fakultät des eigenen Arguments an.

**Beispiel:** FACT 7  5040

## HCS

$\cosh x$

**Funktion:** Hyperbolischer Cosinus

**Anmerkungen:** Zeigt den hyperbolischen Cosinus des numerischen Arguments an.

**Beispiel:** HCS 3  10.067662

**HSN**

sinh x

**Funktion:** Hyperbolischer Sinus**Anmerkungen:** Zeigt den hyperbolischen Sinus des numerischen Arguments an.**Beispiel:** HSN 4  27.2899172**HTN**

tanh x

**Funktion:** Hyperbolischer Tangens**Anmerkungen:** Zeigt den hyperbolischen Tangens des numerischen Arguments an.**Beispiel:** HTN 0.9  0.71629787**\*INT****Funktion:** Ganze Zahl**Anmerkungen:** Zeigt die ganze Zahl des eigenen Arguments an. Die ganze Zahl von PI ist 3.**Beispiel:** INT -1.9  -2**LN** $\log_e x$ **Funktion:** Natürlicher Logarithmus**Anmerkungen:** Zeigt den Logarithmus der Basis e (2.718281828...) des eigenen numerischen Arguments an.**Beispiel:** LN 2  0.69314718**LOG** $\log_{10} x$ **Funktion:** Dekadischer Logarithmus mit der Basis 10**Anmerkungen:** Zeigt den Logarithmus der Basis 10 des eigenen numerischen Arguments an.**Beispiel:** LOG 1000  3

**\*NCR**

$${}_nC_r = n! / r!(n-r)!$$

**Funktion:** Kombination

**Anmerkungen:** Gibt die Werte als NCR (n,r) ein.

**Beispiel:** NCR (6,3)

20

**\*NPR**

$${}_nP_r = n! / (n-r)!$$

**Funktion:** Permutation

**Anmerkungen:** Gibt die Werte als NPR (n,r) ein.

**Beispiel:** NPR (6,3)

120

**PI**

$\pi$

**Funktion:** PI

**Anmerkungen:** Pi ist eine numerische Pseudovariablen mit dem Wert  $\pi$ . Die Anwendung von Pi ist identisch mit dem Gebrauch der  - Taste.

**Beispiel:** PI

3.141592654

**POL**

$$(x,y) \rightarrow (r,\theta)$$

**Funktion:** Umwandlung einer rechtwinkligen in eine Polarkoordinate

**Anmerkungen:** Wandelt die numerischen Argumente in einem rechtwinkligen Koordinaten-Format in ein Polarkoordinaten-Format um.

Das erste Argument bezeichnet die Entfernung von der y-Achse und das zweite die von der x-Achse. Die umgewandelten Werte entsprechen der Entfernung und dem Winkel in den Polarkoordinaten und werden den festen Variablen Y und Z zugeordnet. Der umgewandelte Winkel hängt ab von der Einstellung (Altgrad, Bogenmaß oder Neugrad).

**Beispiel:** DEGREE   
POL (8,6)   
Z

10 (r = 10)  
36.86989765  
( $\theta \approx 36.9^\circ$ )

<b>^</b>	$y^x$
----------	-------

**Funktion:** x-te Potenz

**Anmerkungen:** Zeigt die x-te Potenz eines numerischen Werts an. Eingabe als  $y^x$ .

**Beispiel:** 4 ^ 2.5  32

<b>RCP</b>	$1/x$
------------	-------

**Funktion:** Reziprokwert

**Anmerkungen:** Zeigt den Reziprokwert des numerischen Arguments an.

**Beispiel:** RCP 4  0.25

<b>REC</b>	$(r,\theta) \rightarrow (x,y)$
------------	--------------------------------

**Funktion:** Umwandlung polarer in rechtwinklige Koordinaten

**Anmerkungen:** Wandelt numerische Argumente im Polarkoordinaten-Format in ein rechtwinkliges Koordinaten-Format um.

Das erste Argument bezeichnet die Entfernung und das zweite den Winkel. Die Winkelangabe hängt ab von der Einstellung des Computers (DEG, RAD oder GRAD). Die umgewandelten Werte, die den Entfernungen von der x-Achse und der y-Achse entsprechen, werden den festen Variablen Y und Z zugewiesen.

**Beispiel:** DEGREE   
REC (12,30)  10.39230485 (x ≈ 10.4)  
Z  6 (y = 6)

<b>*RND</b>
-------------

**Funktion:** Zufallszahl

**Anmerkungen:** Siehe RND und RANDOMIZE im BASIC KOMMANDO LEXIKON.

<b>ROT</b>	$\sqrt[x]{y}$
------------	---------------

**Funktion:** x-te Wurzel

**Anmerkungen:** Zeigt die x-te Wurzel des Arguments y an. Eingabe als y ROT x.

**Beispiel:** 7776 ROT 5  6

## \*SGN

**Funktion:** Vorzeichen des Arguments

**Anmerkungen:** Zeigt den auf das Vorzeichen des Arguments basierenden Wert an.

Wenn  $x > 0$ , wird 1 angezeigt.

Wenn  $x < 0$ , wird -1 angezeigt.

Wenn  $x = 0$ , wird 0 angezeigt.

## SIN

sin x

**Funktion:** Sinus

**Anmerkungen:** Zeigt den Sinus des Winkelarguments an. Der angezeigte Wert hängt von der Einstellung ab (Altgrad, Bogenmaß oder Neugrad).

**Beispiel:** DEGREE    
 SIN 30  0.5

## SQR

$\sqrt{x}$

**Funktion:** Quadratwurzel

**Anmerkungen:** Zeigt die Quadratwurzel des Arguments an.

**Beispiel:** SQR 3  1.732050808

## SQU

$x^2$

**Funktion:** Quadrat

**Anmerkungen:** Zeigt das Quadrat des Arguments an.

**Beispiel:** SQU 4  16

## TAN

tan x

**Funktion:** Tangens

**Anmerkungen:** Zeigt den Tangens des Winkelarguments an. Der angezeigte Wert hängt von der Einstellung ab (Altgrad, Bogenmaß oder Neugrad).

**Beispiel:** DEGREE    
 TAN 45  1

## TEN

10<sup>x</sup>

**Funktion:** Antilogarithmus

**Anmerkungen:** Zeigt den Wert von 10 (die Basis des dekadischen Logarithmus) erhoben auf den Wert des eigenen numerischen Arguments.

**Beispiel:** TEN 3  1000

## &H

**Funktion:** Umformung von hexadezimal auf dezimal

**Anmerkungen:** Wandelt einen hexadezimalen Wert in einen dezimalen um.

**Beispiel:** &H F82  3970

# Rechenbereiche

## Numerische Berechnungen:

Für Rechenoperationen mit  $x$  muß  $x$  in einem der unten aufgeführten Bereiche liegen.

$$-1 \times 10^{100} < x \leq -1 \times 10^{-99} \text{ für negatives } x$$

$$10^{-99} \leq x < 10^{100} \text{ für positives } x$$

$$x = 0$$

Der angezeigte Wert für  $x$  wird von der Anzahl der möglichen Stellen auf dem Display begrenzt.

## Funktionen:

Funktion	Bereich von $x$
$\sin x$ $\cos x$ $\tan x$	DEG: $ x  < 1 \times 10^{10}$ RAD: $ x  < \frac{\pi}{180} \times 10^{10}$ GRAD: $ x  < \frac{10}{9} \times 10^{10}$ Ebenso nur für $\tan x$ : ( $n$ =ganzzahlig) DEG: $ x  \neq 90 (2n - 1)$ RAD: $ x  \neq \frac{\pi}{2} (2n - 1)$ GRAD: $ x  \neq 100 (2n - 1)$
$\sin^{-1} x$ $\cos^{-1} x$	$-1 \leq x \leq 1$
$\tan^{-1} x$	$ x  < 1 \times 10^{100}$
$\sinh x$ $\cosh x$ $\tanh x$	$-227.9559242 \leq x \leq 230.2585092$
$\sinh^{-1} x$	$ x  < 1 \times 10^{50}$
$\cosh^{-1} x$	$1 \leq x < 1 \times 10^{60}$
$\tanh^{-1} x$	$ x  < 1$
$\ln x$ $\log x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
$e^x$	$-1 \times 10^{100} < x \leq 230.2585092$
$10^x$	$-1 \times 10^{100} < x < 100$
$\sqrt[3]{x}$	$ x  < 1 \times 10^{100}$
$\frac{1}{x}$	$ x  < 1 \times 10^{100}, x \neq 0$
$x^2$	$ x  < 1 \times 10^{50}$
$\sqrt{x}$	$0 \leq x < 1 \times 10^{100}$
$n!$	$0 \leq n \leq 69$ ( $n$ =ganzzahlig)
DMS→DEG DEG→DMS	$ x  < 1 \times 10^{100}$

Funktion	Bereich von x
$y^x$ ( $y^x = 10^{x \cdot \log y}$ )	wenn $y > 0$ , $-1 \times 10^{100} < x \log y < 100$ wenn $y = 0$ , $x > 0$ wenn $y < 0$ , $\left\{ \begin{array}{l} x = \text{ganzzahlig oder wenn } \frac{1}{x} = \text{ungeradzahlig (} x \neq 0) \\ \text{und } -1 \times 10^{100} < x \log  y  < 100 \end{array} \right.$
$\sqrt[x]{y}$ ( $\sqrt[x]{y} = 10^{\frac{1}{x} \log y}$ )	wenn $y > 0$ , $-1 \times 10^{100} < \frac{1}{x} \log y < 100$ , $x \neq 0$ wenn $y = 0$ , $x > 0$ wenn $y < 0$ , $\left\{ \begin{array}{l} x \text{ oder } \frac{1}{x} \text{ mu\ss ganzzahlig und nicht Null sein,} \\ \text{und } -1 \times 10^{100} < \frac{1}{x} \log  y  < 100 \end{array} \right.$
&H	$0 \leq x \leq 2540BE3FF$ (x in hexadezimal) $FDABF41C01 \leq x \leq FFFFFFFF$
$x, y \rightarrow r, \theta$	$\left\{ \begin{array}{l} (x^2 + y^2) < 1 \times 10^{100} \\ \frac{y}{x} < 1 \times 10^{100} \end{array} \right. \quad \left\{ \begin{array}{l} r = \sqrt{x^2 + y^2} \\ \theta = \tan^{-1} \frac{y}{x} \end{array} \right.$
$r, \theta \rightarrow x, y$	$\left\{ \begin{array}{l} r < 1 \times 10^{100} \\  r \sin \theta  < 1 \times 10^{100} \\  r \cos \theta  < 1 \times 10^{100} \end{array} \right. \quad \left\{ \begin{array}{l} x = r \cos \theta \\ y = r \sin \theta \end{array} \right.$
nPr	$0 \leq r \leq n < 10^{100}$ n, r ganzzahlig
nCr	$0 \leq r \leq n < 10^{100}$ n, r ganzzahlig wenn $n - r < r$ , $n - r \leq 69$ wenn $n - r \geq r$ , $r \leq 69$

# 15. COMPUTER-ANWEISUNGEN (BASIC KOMMANDO LEXIKON)

Die folgenden Seiten enthalten eine Auflistung aller BASIC-Kommandos, die Sie auf dem Computer benutzen können.

Der Einfachheit halber wurden für diesen Nachschlageteil folgende Symbole hinzugefügt:

<b>Ausdruck</b>	Zeigt einen numerischen Wert, eine numerische Variable oder Formel, einschließlich der numerischen Werte und der numerischen Variablen, an.
<b>Variable</b>	Zeigt eine numerische Variable oder eine Zeichenfolge-Variablen an, einschließlich Feldvariablen.
<b>“Zeichenfolge”</b>	Zeigt eine Zeichenfolge innerhalb von Anführungszeichen an.
<b>Zeichenfolge-Variablen</b>	Zeigt die Zeichenfolge-Variablen oder Feldvariablen an.
<b>*Label</b>	Zeigt *Label an. (Obwohl bei diesem Computer sowohl *Label als auch “Label”-Formen verwendet werden können, wird *Label empfohlen. *AB und “AB” werden als die gleichen Labels erkannt.)
<b>[ ]</b>	Die eckigen Klammern sind Ihnen freigestellt. Sie sind selbst nicht Teil der Kommando-Eingabe.
<b>( )</b>	Sie werden benutzt, um geklammerte Werte in gewissen Kommandos einzuschließen. Sie sollten als Teil des Kommandos mit eingegeben werden.
<b>“ ”</b>	Sie werden zum Einschließen von Parametern in gewissen Kommandos benutzt.
<b>{ A } { B }</b>	A oder B kann gewählt werden.
<b>P</b>	Die Ausführung des Programms ist möglich.
<b>D</b>	Direkte Eingabe ist möglich.
<b>Abkürzungen</b>	Die meisten Kommandos können abgekürzt werden. Die kürzesten, noch zulässigen Abkürzungen werden in dieser Anleitung verwendet. <b>Beispiel:</b> Abkürzung: P. (für PRINT) Die folgenden Abkürzungen sind ebenfalls zulässig: PR. PRI. PRIN.

---

# ASC

---

P  
D

**FORMAT:** ASC { "Zeichenfolge"  
Zeichenfolge-Variable }

**Abkürzung:** AS.

**Siehe auch:** CHR\$

---

**WIRKUNG:**

Berechnet den Zeichencode für das erste Zeichen in der vorgegebenen Zeichenfolge.

**ANMERKUNGEN:**

Die Zeichenfolge kann als Inhalt einer Zeichenfolge-Variablen in der Form X\$ oder als wirkliche Zeichenfolge in Anführungszeichen, z.B. "XXXX", angegeben werden. Nur der Zeichencode des ersten Zeichens der Zeichenfolge wird dargestellt. Die Zeichencode-Tabelle finden Sie im Anhang C.

**BEISPIEL:**

```
10: INPUT "ENTER A CHARACTER ";A$
20: N = ASC A$
30: PRINT "THE CHARACTER CODE IS ";N
40: GOTO 10
```

[10] Der Benutzer drückt eine Taste, um irgendein Zeichen einzugeben.

[20] ASC findet die Codenummer für dieses Zeichen.

[30] Das Ergebnis wird angezeigt.

[40] Wiederholung, bis der Benutzer das Programm durch Betätigen der **BREAK**-Taste stoppt.

---

# BEEP

---

P  
D

**FORMAT:** BEEP Ausdruck

**Abkürzung:** B.

**Siehe auch:**

---

**WIRKUNG:**

Erzeugt einen akustischen Signalton.

**ANMERKUNGEN:**

Die BEEP-Anweisung veranlaßt den Computer, einen oder mehrere akustische Signale abzugeben.

Die Anzahl der Signaltöne wird durch den Ausdruck festgelegt, der eine Zahl (positive Zahl, kleiner als 65535) sein muß. Der Ausdruck wird berücksichtigt, aber nur der ganzzahlige Anteil der Zahl wird für die Anzahl der Signaltöne berücksichtigt. BEEP kann auch als Befehl mit numerischen Zeichen und definierten Variablen verwendet werden. In diesem Fall ertönt das akustische Signal direkt nach dem Drücken der Taste .

---

# CHR\$

---

P  
D

**FORMAT:** CHR\$ Ausdruck

**Abkürzung:** CH.

**Siehe auch:** ASC

---

**WIRKUNG:**

Springt auf das Zeichen zurück, das dem numerischen Zeichencode des Parameters entspricht.

**ANMERKUNGEN:**

Sie finden die Tabelle der Zeichencodes und ihres Verhältnisses zu Zeichen in Anhang C. Zum Beispiel ist CHR\$ 65 gleich "A".

Eine Hexadezimalzahl kann mit "&H" vor dem Zeichen bestimmt werden (z.B. A\$ = CHR\$ &H5A).

Ein Wert größer als 255 bewirkt einen Fehler.

**BEISPIEL:**

```
10: AA$=""  
20: INPUT "CODE=" ;A:CLS  
30: AA$=AA$+CHR$ A  
40: LOCATE 7,1:PRINT AA$  
50: GOTO 20
```

Zeigt die Zeichen an, die durch die Codeeingabe in Zeile 20 repräsentiert sind.

---

# CLEAR

---

P  
D

**FORMAT:** CLEAR

**Abkürzung:** CL.

**Siehe auch:** DIM

---

**WIRKUNG:**

Mit dem CLEAR-Befehl werden alle in einem Programm benutzten Variablen gelöscht und alle gesetzten Variablen auf Null bzw. leer gesetzt.

**ANMERKUNGEN:**

Der CLEAR-Befehl setzt Speicherraum frei, der zum Speichern von einfachen oder Feldvariablen mit der DIM-Anweisung gebraucht wurde. CLEAR kann am Beginn eines Programmes verwendet werden. Damit wird von vorher zur Ausführung eines Programms eingegebenen Variablen beanspruchter Platz freigemacht, wenn mehrere Programme gespeichert sind.

Verwenden Sie niemals das Kommando CLEAR in einer FOR...NEXT-Schleife.

**BEISPIEL:**

```
10: A = 5: DIM C(5)
```

```
20: CLEAR
```

[20] Setzt den C( ). zugewiesenen Bereich frei und setzt A gleich Null.

---

# CLOAD

D

---

**FORMAT:** 1. CLOAD "Dateiname"   
2. CLOAD 

**Abkürzung:** CLO.

**Siehe auch:** CLOAD?, CSAVE

---

**WIRKUNG:**

Lädt ein auf Band gesichertes Programm in das Gerät.

**ANMERKUNGEN:**

Nur als direkte Eingabe in der PRO- oder RUN-Betriebsart zulässig.

Format 1 des CLOAD-Kommandos löscht den Speicher, sucht das Band nach dem Programm ab, dessen Name mit "Dateiname" angegeben ist, und lädt das Programm.

Format 2 des CLOAD-Kommandos löscht die existierenden Programme aus dem Speicher und lädt das erste Programm auf dem Band, angefangen bei der aktuellen Position.

Während der Ausführung wird rechts unten nach dem Dateinamen ein "\*" angezeigt. Nach der Ausführung verschwindet das "\*" und das Zeichen (>) erscheint. Während der Suche nach einer Datei wird "\*" nicht angezeigt. Das gleiche gilt für den CLOAD?-Befehl.

Wenn ein bestimmter Dateiname nicht gefunden wird, sucht der Computer weiter nach der Datei, auch wenn das Ende des Bandes erreicht ist. Zum Unterbrechen die **BREAK**-Taste drücken.

Wenn bei der Ausführung des CLOAD-Befehls ein Fehler auftritt, ist das geladene Programm nicht mehr gültig.

Für Arbeiten mit Kassetten siehe den Abschnitt über Peripheriegeräte.

**BEISPIEL:**

CLOAD\*

CLOAD "PRO3"\*\*\*

\* Lädt das erste Programm vom Band.

\*\* Sucht auf dem Band die Datei "PRO3" und lädt das Programm.

---

# CLOAD?

---

D

**FORMAT:** 1. CLOAD? "Dateiname"   
2. CLOAD? 

**Abkürzung:** CLO.?

**Siehe auch:** CLOAD, CSAVE

---

## **WIRKUNG:**

Mit dem Kommando CLOAD? wird ein auf Band gespeichertes Programm mit einem im Speicher enthaltenen Programm verglichen.

## **ANMERKUNGEN:**

Nur als direkte Eingabe in der PRO- oder RUN-Betriebsart zulässig.

Um zu prüfen, ob ein Programm richtig gespeichert wurde, spulen Sie das Band an den Anfang zurück und geben Sie das CLOAD?-Kommando ein.

Format 1 des CLOAD?-Kommandos sucht nach dem mit "Dateinamen" bezeichneten Programm und vergleicht es dann mit dem Speicherinhalt. Format 2 des CLOAD?-Kommandos vergleicht das im Speicher enthaltene Programm mit dem ersten Programm auf dem Band, angefangen bei der aktuellen Position.

Wenn ein Bandprogramm nicht mit einem gespeicherten Programm übereinstimmt, erfolgt ein Fehler. Während der Ausführung wird rechts unten nach dem Dateinamen ein "\*" angezeigt. Nach der Ausführung verschwindet das "\*" und das Zeichen (>) erscheint.

## **Hinweis:**

Beim Laden eines Programms, das auf einem anderen Computer geschrieben und auf Band gespeichert wurde, wird dieses Programm in den PC-E220-Code umgewandelt. In diesem Fall kann der CLOAD?-Befehl nicht ausgeführt werden.

Für Arbeiten mit Kassetten siehe den Abschnitt über Peripheriegeräte.

## **BEISPIEL:**

CLOAD?\*  
CLOAD?"PRO3"\*\*\*

\* Vergleicht den Speicherinhalt mit dem ersten auf dem Band gefundenen Programm.

\*\* Sucht auf dem Band die Datei "PRO3" und vergleicht sie dann mit dem Speicherinhalt.

---

# CLOSE

---

P  
D

**FORMAT:** CLOSE [# 1]

**Abkürzung:** CLOS.

**Siehe auch:** OPEN

---

**WIRKUNG:**

Schließt eine Datei.

**ANMERKUNGEN:**

Mit diesem Kommando wird eine Datei auf dem gegenwärtig angeschlossenen Gerät, z.B. einem Kassettenrekorder, geschlossen. Wenn eine Datei mit OUTPUT geöffnet wurde, werden die Daten im Speicher (dem Ausgabe-Puffer) mit einem EOF-Code (end of file) auf Kassette ausgegeben und dann erst geschlossen.

Eine Datei wird in den folgenden Fällen geschlossen:

- Nach Durchführung eines END-, NEW- oder RUN-Kommandos.
- Der Strom wird abgeschaltet.
- Nach dem Editieren eines Programms (Programm-Eingaben, Korrekturen, Löschen oder Ausführung eines DELETE-Befehls.)
- Ein Programm wird auf die RAM-Diskette gesichert oder von der RAM-Diskette geladen oder gelöscht.
- Einer der Befehle CSAVE, CLOAD oder CLOAD? wurde ausgeführt.
- Die Monitor-Betriebsart wurde gewählt.

**Hinweis:**

Der CLOSE-Befehl muß **jedesmal** ausgeführt werden, wenn eine serielle Kommunikation durchgeführt wird. Wenn die Datei offen bleibt und das Datenübertragungskabel angeschlossen ist, wird ständig Strom von den Betriebsbatterien des Computers verbraucht.

---

# CLS

---

P

**FORMAT:** CLS

**Abkürzung:**

**Siehe auch:** LOCATE

---

**WIRKUNG:**

Das CLS-Kommando löscht das Display.

**ANMERKUNGEN:**

Das CLS-Kommando löscht die Anzeigen vom Display und versetzt es in die Startposition (0,0).

---

# CONT

---

D

**FORMAT:** CONT 

**Abkürzung:** C.

**Siehe auch:** STOP

---

**WIRKUNG:**

Mit dem CONT-Kommando setzen Sie die Ausführung eines zeitweise unterbrochenen Programms fort.

**ANMERKUNGEN:**

Nur als direkte Eingabe in der RUN-Betriebsart zulässig.

Wenn die Ausführung eines Programms mit dem Befehl STOP unterbrochen wurde, so kann sie mit der Anweisung CONT fortgesetzt werden.

Wurde ein Programm mit der  -Taste unterbrochen, so kann es nach Eingabe des CONT-Befehls am Aufforderungszeichen fortgesetzt werden.

**BEISPIEL:**

CONT

Setzt die Ausführung eines unterbrochenen Programms fort.

---

# CSAVE

---

P  
D

**FORMAT:** 1. CSAVE "Dateiname"  
2. CSAVE  
3. CSAVE "Dateiname", "Kennwort"  
4. CSAVE , "Kennwort"

**Abkürzung:** CS.

**Siehe auch:** CLOAD, CLOAD?, PASS

---

## **WIRKUNG:**

Mit dem Kommando CSAVE wird ein Programm auf Band gesichert.

## **ANMERKUNGEN:**

Format 1 des Kommandos CSAVE speichert alle Programmzeilen im Speicher des Rechners auf Band und weist den entsprechenden Dateinamen zu.

Format 2 des Kommandos CSAVE speichert alle Programmzeilen des Speichers ohne spezielle Dateinamen auf Band.

Format 3 des Kommandos CSAVE speichert alle Programmzeilen im Speicher des Gerätes auf Band und weist sowohl den angegebenen Dateinamen als auch das angegebene Kennwort zu.

Format 4 des Kommandos CSAVE schreibt alle Programmzeilen im Speicher des Gerätes ohne speziellen Dateinamen auf Band und weist das angegebene Kennwort zu.

Programme, die mit einem Kennwort geschützt sind, können von jedermann geladen, aber nur von jemandem aufgelistet oder verändert werden, der das korrekte Kennwort kennt. (Siehe PASS-Kommando.)

Wenn ein Programm im Speicher schreibgeschützt ist, wird der CSAVE-Befehl nicht beachtet.

Es sollte vermieden werden, verschiedene Programme mit dem gleichen Dateinamen auf die gleiche Seite (A oder B) eines Bandes zu schreiben. Dadurch kann ein falsches Programm gelesen werden. Es wird empfohlen, die Nummer des Bandzählwerks zu registrieren, wenn ein Programm auf Band geschrieben wird.

Für Arbeiten mit Kassetten siehe den Abschnitt über Peripheriegeräte.

## **BEISPIEL:**

CSAVE "PRO3", "SECRET"

Speichert die jetzt gesicherten Programme unter dem Namen "PRO3" mit dem Kennwort "SECRET" auf Band ab.

---

# DATA

---

P

**FORMAT:** DATA Liste der Werte

**Abkürzung:** DA.

**Siehe auch:** READ, RESTORE

---

**WIRKUNG:**

DATA wird benutzt, um Daten für READ bereitzustellen.

**ANMERKUNGEN:**

Soll einem Feld mit Anfangswerten zugewiesen werden, ist es empfehlenswert, die Werte in einer DATA-Anweisung aufzulisten und sie dann mit einer READ-Anweisung in einer FOR...NEXT-Schleife in das Feld einzulesen. Wird die erste READ-Anweisung ausgeführt, wird der erste Wert in der DATA-Anweisung abgerufen. Mit jedem der darauf folgenden READ-Befehle werden die nachfolgenden Werte in der Reihenfolge gelesen, in der sie im Programm erscheinen, ohne Rücksicht auf die Anzahl der in jeder DATA-Anweisung definierten Werte oder auf die Anzahl der benutzten DATA-Anweisungen.

Eine DATA-Anweisung kann aus beliebigen numerischen oder Zeichenfolge-Werten, getrennt durch Kommata, zusammengesetzt sein. Die Zeichenfolge-Werte müssen in Anführungszeichen gesetzt werden. Leerstellen am Anfang und am Ende einer Zeichenfolge müssen in die Anführungszeichen mit eingeschlossen werden.

DATA-Anweisungen haben keinen Einfluß, wenn sie im Laufe der normalen Ausführung des Programmes angetroffen werden, so daß sie an jeder Stelle des Programms eingefügt werden können. Viele Programmierer setzen sie gerne hinter den dazugehörigen READ-Befehl. Wenn nötig, können die Werte der DATA-Anweisung mit der RESTORE-Anweisung ein zweites Mal gelesen werden.

**BEISPIEL:**

```
10: DIM B(10)
20: WAIT 128
30: FOR I = 1 TO 10
40: REAR B(I)
50: PRINT B(I)
60: NEXT I
70: DATA 10,20,30,40,50,60
80: DATA 70,80,90,100
90: END
```

[10] Initialisierung eines Feldes.

[40] Lädt die Werte aus der DATA-Anweisung, so daß B(1) 10 wird, B(2) 20 wird, B(3) 30 wird, usw.

---

# DEGREE

---

P  
D

**FORMAT:** DEGREE

**Abkürzung:** DE.

**Siehe auch:** GRAD, RADIAN

---

**WIRKUNG:**

DEGREE wird benutzt, um Winkelwerte auf Altgrad umzuschalten.

**ANMERKUNGEN:**

Der Computer kann drei Formen der Winkeldarstellung verarbeiten: Altgrad (dezimal), Bogenmaß und Neugrad. Diese Formen werden für die Darstellung der Werte und Argumente von den SINus-, COSinus- und TANGens-Funktionen und der Ergebnisse der Umkehrfunktionen Arcussinus (ASN), Arcuscosinus (ACS) und Arcustangens (ATN) benutzt.

Die DEGREE-Funktion schaltet die Form für alle Winkelwerte auf Altgrad um, bis Sie GRAD und RADIAN einsetzen. Die DMS- und DEG-Funktionen werden benutzt, um Winkelwerte von Altgrad in Grade, Minuten, Sekunden und umgekehrt umzuformen.

**BEISPIEL:**

```
10: DEGREE
20: X = ASN 1
30: PRINT X
```

[20] X hat jetzt den Wert 90, d.h. 90 Grad, dem Arcussinus von 1.

---

# DELETE

D

**FORMAT:** 1. DELETE Zeilennummer   
2. DELETE Zeilennummer -   
3. DELETE Zeilennummer - Zeilennummer   
4. DELETE - Zeilennummer 

**Abkürzung:** DEL.

**Siehe auch:** NEW, PASS

---

## **WIRKUNG:**

Löscht die angegebenen Programmzeilen im Speicher.

## **ANMERKUNGEN:**

Nur als direkte Eingabe in der PRO-Betriebsart zulässig.

Format 1 löscht nur die angegebene Programmzeile.

Format 2 löscht die Programmzeilen von der angegebenen Zeilennummer bis zur höchsten Programmzeile im Speicher.

Format 3 löscht alle Programmzeilen zwischen der ersten angegebenen Zeilennummer (niedrigster Wert) und der zweiten angegebenen Zeilennummer (höchster Wert).

Format 4 löscht die Programmzeilen von der niedrigsten Zeilennummer im Speicher bis zur angegebenen Zeilennummer.

Wenn Sie DELETE im RUN-Modus benutzen, erfolgt ein Fehler. Falls Sie ein Kennwort eingesetzt haben, wird der Befehl nicht ausgeführt und das Aufforderungszeichen erscheint. Als Zeilennummern sind nur die Ziffern 0 bis 9 gültig. Wenn Sie eine Zeile angeben, die nicht existiert, erfolgt ein Fehler. Wenn Sie eine Startzeilennummer angeben, die größer ist als die Endzeilennummer, erfolgt ebenfalls ein Fehler.

Wenn die Zeilennummern der ersten und zweiten Zeile ausgelassen werden, erfolgt ein Fehler.

Um das ganze Programm zu löschen, benutzen Sie das NEW-Kommando.

## **BEISPIEL:**

```
DELETE 150*  
DELETE 200-**  
DELETE 50-150***  
DELETE -35****
```

\* Löscht nur die Zeile 150.

\*\* Löscht alles von Zeile 200 bis zur höchsten Zeilennummer.

\*\*\* Löscht alle Zeilen zwischen der Zeile 50 und der Zeile 150.

\*\*\*\* Löscht alle von Zeilen der niedrigsten Zeilennummer bis zur Zeile 35.

---

# DIM

---

P  
D

- FORMAT:**
1. DIM Name der numerischen Variablen 1 (Größe)  
[, Name der numerischen Variable 2 (Größe)]
  2. DIM Name der Zeichenfolge-Variable 1 (Größe) \*Länge  
[, Name der Zeichenfolge-Variable 2 (Größe) \*Länge]
  3. DIM Name des numerischen Feldes 1 (Zeilen, Spalten)  
[, Name des numerischen Feldes 2 (Zeilen, Spalten)]
  4. DIM Name des numerischen Feldes 1 (Zeilen, Spalten) \*Länge  
[, Name des numerischen Feldes 2 (Zeilen, Spalten) \*Länge]

**Abkürzung:** D.

**Siehe auch:** CLEAR, RUN

---

## **WIRKUNG:**

Zuweisung von Speicherplatz für numerische und Zeichenfolge-Feldvariable.

## **ANMERKUNGEN:**

DIM muß verwendet werden, um Speicherplatz für die Feldvariablen zuzuweisen. Die Größe eines Feldes entspricht der Anzahl der Elemente in diesem Feld.

Die maximale Anzahl von Dimensionen für ein Feld ist zwei, die maximale Größe einer Dimension kann 255 sein. Zusätzlich zu der Anzahl der mit der Dimensions-Anweisung zugewiesenen Elemente wird noch ein weiteres Null-Element reserviert. Mit der Anweisung DIM B(3) werden die Elemente B(0), B(1), B(2) und B(3) bereitgestellt. Bei zweidimensionalen Feldern wird eine gesonderte Null-Reihe und eine Null-Spalte bereitgestellt.

Bei Zeichenfolge-Feldern wird die Größe jedes Zeichenfolge-Elementes und die Anzahl der Elemente spezifiziert. Mit der Anweisung DIM B\$(3)\*12 wird Speicherplatz für 4 Zeichenfolgen mit jeweils einem Maximum von 12 Zeichen bereitgestellt. Wenn die Länge nicht angegeben wird, nimmt der Computer die Maximallänge von 16 Zeichen an.

Wenn ein numerisches Feld dimensioniert ist, werden anfänglich alle Werte auf Null gesetzt; bei einem Zeichenfolge-Feld werden alle Werte auf Null gesetzt.

Für die Zuweisung von Speicherplatz können die Zahlen von 0 bis 255 verwendet werden (für Reihen und Spalten); ein Fehler kann allerdings auftreten, wenn die zugewiesene Größe nicht reserviert werden kann, weil nicht genügend freier Speicherplatz zur Verfügung steht.

Wenn die Zuweisung der Größe einen Dezimalpunkt enthält, wird nur der ganzzahlige Teil der Zuweisung berücksichtigt (der Dezimalanteil wird ignoriert.)

### *Beispiel:*

X(2.3) wird als X(2) erkannt.

Y(0.25) wird als Y(0) erkannt.

Der Speicherbereich und die Größe des Feldes können durch numerische Variable oder durch Ausdrücke bestimmt werden:

```
10: INPUT A, B
20: DIM X(A), YY(B-1, A*B)
```

Mit einer DIM-Anweisung kann mehr als ein Feld bestimmt werden:

*Beispiel:*

```
DIM V(5), K$(4,3), XB$(5)
```

Ein bereits bestimmtes Feld kann nicht noch einmal bestimmt werden. Beispielsweise können die Anweisungen DIM X(5) und DIM X(3,4) nicht gleichzeitig bestimmt werden, da die Variablennamen gleich sind.

Numerische Feldvariable und Zeichenfolge-Feldvariable werden als verschiedene Felder erkannt; daher können die Felder Z( ) und Z\$( ) gleichzeitig bestimmt werden.

Feldvariable können mit dem CLEAR-Befehl gelöscht werden (oder als nicht-definiert bestimmt werden). Wenn ein Programm mit dem RUN-Befehl gestartet wird, werden alle Feldvariable automatisch gelöscht.

Ein Feld kann nur einmal bestimmt werden und jeder Versuch, es noch einmal zu bestimmen, ohne vorher mit CLEAR zu löschen, führt zu einem Fehler. Besondere Vorsicht ist bei einem Programm mit der Anweisung GOTO geboten, damit die gleiche DIM-Anweisung nicht zweimal ausgeführt wird, ohne vorher mit CLEAR zu löschen.

**BEISPIEL:**

```
10: DIM B(10)
20: DIM C$(4,4)*10
30: DIM F$(12)
40: DIM H$(4,6)
```

[10] Bereitstellung von Speicherplatz für ein numerisches Feld mit 11 Elementen.

[20] Bereitstellung von Speicherplatz für ein zweidimensionales Zeichenfolge-Feld mit 5 Zeilen und 5 Spalten; jede Zeichenfolge kann maximal bis zu 10 Zeichen enthalten.

[30] Bereitstellung von Speicherplatz für die Zeichenfolge-Variable F\$ mit 13 Elementen.

[40] Bereitstellung von Speicherplatz für ein zweidimensionales Feld mit 5 Zeilen und 7 Spalten, also mit 35 Elementen.

---

# END

---

P

**FORMAT:** END

**Abkürzung:** E.

**Siehe auch:**

---

**WIRKUNG:**

Signalisiert das Ende eines Programms.

**ANMERKUNGEN:**

Durch die Ausführung der END-Anweisung wird das Programm beendet. Anweisungen nach der END-Anweisung auf der gleichen Zeile werden nicht mehr ausgeführt. Eine offene Datei wird geschlossen.

**BEISPIEL:**

```
10: PRINT "HELLO"  
20: END  
30: PRINT "GOODBYE"  
40: END
```

Mit diesen Programmen im Speicher druckt das Kommando "RUN 10" das Wort "HELLO", aber nicht das Wort "GOODBYE". Dieses wird auf das Kommando "RUN 30" ausgegeben.

---

# FILES

---

D

**FORMAT:** FILES 

**Abkürzung:** FI.

**Siehe auch:** LFILES, SAVE, LOAD

---

**WIRKUNG:**

Zeigt die Namen der Dateien auf der RAM-Diskette an.

**ANMERKUNGEN:**

Es können maximal vier Dateinamen gleichzeitig dargestellt werden; die Markierung ➔ erscheint rechts von diesen Namen. Durch Drücken der Tasten  und  können die Namen bzw. die Markierung ➔ auf- oder abgerollt werden.

Wenn die Taste  oder  gedrückt wird, wird die Aufforderung ">" dargestellt und der Computer wartet auf den nächsten Befehl.

Zur Anzeige einer Auflistung der Programme wird die Markierung ➔ auf den Dateinamen des gewünschten Programms bewegt und  +  gedrückt. Auflistungen von Programmen, die in der TEXT-Betriebsart registriert wurden, können allerdings nicht durchgeführt werden.

---

# FOR...NEXT

---

P

**FORMAT:** FOR numerische Variable = Ausdruck 1 TO Ausdruck 2  
[STEP Ausdruck 3]  
{  
NEXT [numerische Variable]

**Abkürzung:** F. N. STE.  
**Siehe auch:**

---

## **WIRKUNG:**

FOR wird in Verbindung mit NEXT benutzt, um eine Reihe von Operationen mehrmals zu wiederholen.

## **ANMERKUNGEN:**

FOR und NEXT schließen eine Gruppe von Anweisungen ein, die wiederholt werden sollen. Wenn die Variable nach NEXT ausgelassen wird, wird die Variable angenommen, die auf FOR folgt. Wird diese Gruppe das erste Mal ausgeführt, so hat die Schleifenvariable (die Variable, die unmittelbar nach FOR genannt wird) ihren Anfangswert (Ausdruck 1).

Erreicht die Ausführung nun die NEXT-Anweisung, so wird die Schleifenvariable durch den Ergänzungswert (Ausdruck 3) erhöht und dann mit dem Schlußwert (Ausdruck 2) verglichen. Ist der Wert der Schleifenvariablen weniger oder gleich dem Ausdruck 2, so wird die eingeschlossene Gruppe ein weiteres Mal ausgeführt, beginnend mit dem Befehl hinter FOR. Der vorgegebene Ergänzungswert ist 1; in der zweiten Form wird die Erhöhung durch den Ausdruck 3 vorgegeben. Ist der Wert der Schleifenvariablen größer als Ausdruck 2, so wird die Ausführung unmittelbar hinter NEXT fortgesetzt. Da der Vergleich am Ende durchgeführt wird, werden die Anweisungen innerhalb einer FOR...NEXT-Schleife mindestens einmal ausgeführt.

Wenn der Ergänzungswert Null ist, wird FOR...NEXT als eine unendliche Schleife fortgesetzt.

Die Schleifenvariable kann in der umschlossenen Gruppe von Anweisungen benutzt werden, z.B. als ein Feld-Index. Man sollte aber vorsichtig sein, wenn der Wert der Schleifenvariablen verändert wird. Programme sollen so geschrieben werden, daß die Ausführung nicht aus der FOR...NEXT-Schleife herausspringt, bevor der Zähler den Schlußwert erreicht. Um aus einer Schleife herauszukommen, bevor sie die bestimmte Anzahl von Malen wiederholt wurde, setzen Sie die Schleifenvariable höher als den Schlußwert.

Die Gruppe von Anweisungen, die von einer FOR...NEXT-Schleife eingefasst werden, kann ein weiteres Paar von FOR...NEXT-Anweisungen enthalten, das eine andere Schleifenvariable haben muß, solange das eingeschlossene Paar vollständig eingeschlossen ist. Wenn also eine FOR-Anweisung in der Gruppe auftaucht, so muß auch der dazugehörige NEXT-Befehl eingeschlossen sein. So können bis zu sechs FOR...NEXT-Schleifen ineinander "geschachtelt" sein. Ein illegales Herausspringen aus einer inneren Schleife führt zu einem Schachtelungsfehler.

Innerhalb einer FOR...NEXT-Schleife dürfen keinesfalls die Kommandos CLEAR oder DIM verwendet werden.

---

## FRE

P  
D

---

**FORMAT:** FRE

**Abkürzung:** FR.

**Siehe auch:**

---

**WIRKUNG:**

Stellt die freien Plätze in den Bereichen für Programme und Daten in Byte dar.

**ANMERKUNGEN:**

FRE zeigt die Byteanzahl des freien Platzes im Speicherbereich für die Programme und Daten (in einem Bereich, der nicht von BASIC-Programmen, Feldvariablen, einfachen Variablen bzw. Maschinencode belegt wird; er enthält auch die RAM-Diskette und den TEXT-Bereich).

**Hinweis:**

Bei Verwendung dieses Befehls in der PRO-Betriebsart wird PRINT FRE  eingegeben.

---

# GOSUB...RETURN

---

P

**FORMAT:** GOSUB { Zeilennummer }  
                  { \*Label }  
                  {  
                  RETURN

**Abkürzung:** GOS. RE.

**Siehe auch:** GOTO, ON...GOSUB

---

## **WIRKUNG:**

Verweist die Programmausführung auf eine BASIC-Subroutine.

## **ANMERKUNGEN:**

Wenn die gleiche Gruppe von Anweisungen innerhalb eines Programms mehrfach ausgeführt werden soll, ist es nützlich, die Möglichkeiten innerhalb BASIC für Subroutinen mit GOSUB und RETURN zu verwenden.

Die Gruppe von Anweisungen wird an einer Stelle des Programms eingegeben, an der sie bei der normalen Sequenz einer Ausführung nicht erreicht wird. Die normale Lokalisation ist nach der END-Anweisung, die das Ende des Hauptprogramms bestimmt.

An jeder Stelle des Hauptprogramms, an der die Subroutine ausgeführt werden soll, wird die GOSUB-Anweisung mit der Zeilennummer oder mit \*Label angegeben, welche die Nummer der Anfangszeile der Subroutine angibt. Die letzte Zeile einer Subroutine muß unbedingt RETURN sein.

Zur Ausführung von GOSUB geht der Computer auf die angegebene Zeilenzahl oder das \*Label und verarbeitet die Anweisungen bis zum Erreichen von RETURN. Danach geht die Kontrolle wieder zurück zur Anweisung, die GOSUB folgt.

Subroutinen können bis zu 10-fach "verschachtelt" sein.

Es existiert auch eine ON...GOSUB-Struktur zur Wahl von verschiedenen Subroutinen an bestimmten Lokalisationen innerhalb des Programms; der Ausdruck in einer GOSUB-Anweisung besteht nur in der gewünschten Zeilennummer oder \*Label.

## **BEISPIEL:**

```
10: GOSUB 100
20: END
100: PRINT "HELLO"
110: RETURN
```

Bei der Ausführung druckt dieses Programm einmal HELLO aus.

---

# GOTO

---

P  
D

**FORMAT:** GOTO { Zeilennummer }  
                  \*Label }

**Abkürzung:** G.

**Siehe auch:** GOSUB, ON...GOTO, RUN

---

## **WIRKUNG:**

Versetzt die Programmsteuerung zu einer spezifizierten Zeilennummer oder zu \*Label.

## **ANMERKUNGEN:**

GOTO veranlaßt den Rechner, von einer bestimmten Stelle in einem BASIC-Programm an eine andere Stelle zu springen. Anders als bei GOSUB "erinnert" sich GOTO nicht, von wo dieser Sprung ausgeführt wurde.

Im Allgemeinen wird ein Programm sequentiell von der kleinsten Zeilennummer aus ausgeführt. Die Ausführung kann aber auf eine bestimmte Zeilennummer oder ein \*Label übertragen werden. Die Programmausführung ab einer bestimmten Zeilennummer kann mit der GOTO-Anweisung als direkte Eingabe in der RUN-Betriebsart vorgenommen werden. Der Übergangspunkt wird durch Eingabe der Zeilennummer oder \*Label nach dem GOTO-Befehl spezifiziert.

## *Beispiel:*

GOTO 40     Springt nach Zeile 40.  
GOTO \*AB   Springt zur Zeile mit dem Label \*AB.

Wenn eine spezifizierte Zeilenzahl oder ein \*Label nicht existiert, erfolgt ein Fehler. Wenn zwei oder mehr identische \*Labels in einem Programm vorkommen, wird die Ausführung auf die Zeile mit der kleinsten Zeilennummer übertragen.

## **BEISPIEL:**

```
10: INPUT A$  
20: IF A$ = "Y" GOTO 50  
30: PRINT "NO"  
40: GOTO 60  
50: PRINT "YES"  
60: END
```

Dieses Programm druckt das Wort "YES" aus, wenn Sie "Y" eingeben, und druckt das Wort "NO", wenn Sie etwas anderes eingeben.

---

# GRAD

---

P  
D

**FORMAT:** GRAD

**Abkürzung:** GR.

**Siehe auch:** DEGREE, RADIAN

---

**WIRKUNG:**

GRAD wird benutzt, um die Form von Winkelwerten auf Neugrad umzuschalten.

**ANMERKUNGEN:**

Der Computer kann 3 Formen der Winkeldarstellung verarbeiten: Altgrad (dezimal), Bogenmaß und Neugrad. Diese Formen werden für die Darstellung der Werte der Argumente von den SINus-, COSinus- und TANgens-Funktionen und der Ergebnisse der Umkehrfunktionen ArcusSINus, ArcusCoSinus und ArcusTANgens benutzt.

Die GRAD-Funktion wird benutzt, um Winkelwerte in Gradienten umzuformen, bis DEGREE oder RADIAN eingestellt wird. Die Neugrad-Form stellt die Winkelmessung in Form von Prozentgradient dar, d.h. ein Winkel von  $45^\circ$  ist ein Gradient von 50 Prozent.

**BEISPIEL:**

10: GRAD

20: X = ASN 1

30: PRINT X

X hat jetzt den Wert 100, d.h. 100 Neugrad, dem Arcussinus von 1.

---

# IF...THEN

---

P

**FORMAT:** IF Bedingung THEN 

Zeilennummer
*Label
Anweisung

**Abkürzung:** IF T.

**Siehe auch:** AND, OR, NOT

---

## **WIRKUNG:**

Führt die Anweisung entsprechend der Bedingung aus, wenn das Programm ausgeführt wird.

## **ANMERKUNGEN:**

Wenn die Bedingung der IF-Anweisung zutrifft, wird die Anweisung nach THEN ausgeführt. Wenn sie nicht zutrifft, wird die Anweisung nach THEN übersprungen und die folgende Zeile ausgeführt.

Wenn THEN von einer GOTO-Anweisung gefolgt werden, kann entweder THEN oder GOTO ausgelassen werden.

## *Beispiel:*

1) IF A<5 THEN C=A\*B:GOTO 50

Wenn A kleiner als 5 ist wird das Produkt, A\*B auf C zugewiesen und nach Zeile 50 übergegangen.

2) IF B=C+1 GOTO 60

oder

IF B=C+1 THEN 60

Wenn B gleich C+1 ist, wird auf Zeile 60 übergegangen, andernfalls auf die folgende Zeile.

Die Bedingung in einer IF-Anweisung (z.B. A<5) kann ein beliebiger Verhältnis-Ausdruck sein, wie unten aufgeführt:

Verhältnis-Ausdruck	Beschreibung
○○ = ××	Gleich
○○ > ××	Größer als
○○ >= ××	Nicht weniger als
○○ < ××	Kleiner als
○○ <= ××	Nicht mehr als
○○ <> ××	Nicht gleich wie

Hinweis: ○○ und ×× stellen Ausdrücke dar (5\*4, A, 8, usw.).

Mehr als ein Verhältnis-Ausdruck können durch die logischen Operatoren "\*" oder "+" verbunden werden. Zum Beispiel:

```
IF (A>5)*(B>1) THEN ....
```

Wenn A größer als 5 und B größer als 1 ist, wird die Anweisung nach THEN ausgeführt. Der logische Operator "AND" kann anstelle von "\*" verwendet werden. Zum Beispiel:

```
IF (A>5)+(B>1) THEN ....
```

Wenn A größer als 5 oder B größer als 1 ist, wird die Anweisung nach THEN ausgeführt. Der logische Operator "OR" kann anstelle von "+" verwendet werden.

### Verwendung von Buchstaben-Zeichenfolgen in Verhältnis-Ausdrücken

Die Größenordnung von Buchstaben-Zeichenfolgen kann verglichen werden, wenn ein Verhältnis-Ausdruck in einer IF...THEN-Anweisung benutzt wird. Die Größenordnung der Zeichencodes wird verglichen. Die Buchstaben A, B und C haben die entsprechenden Codes 65, 66 und 67. Daher ist A kleiner als B und B kleiner als C.

### Länge der Zeichenfolgen in Ausdrücken

Die Gesamtanzahl der Zeichen, die in einem Ausdruck verwendet werden dürfen, mit dem zwei oder mehr Zeichenfolgen verbunden oder verglichen werden, darf 255 nicht überschreiten.

#### **BEISPIEL:**

```
10: INPUT "CONTINUE?";A$
20: IF A$="YES" THEN 10
30: IF A$="NO" GOTO 60
40: PRINT "YES OR NO, PLEASE"
50: GOTO 10
60:
```

#### **Hinweis:**

Wenn ein Variablenname von einer Anweisung gefolgt wird, muß unbedingt eine Leerstelle eingefügt werden, z.B.:

```
100 IF A=B THEN 200
```

↑ Eine Leerstelle ist notwendig.

Auf dieses Format muß besonders geachtet werden, wenn die Befehle IF, FOR, ON...GOTO oder ON...GOSUB verwendet werden.

---

# INKEY\$

---

P

**FORMAT:** INKEY\$

**Abkürzung:** INK.

**Siehe auch:**

---

**WIRKUNG:**

Weist der bestimmten Variablen den Wert der betätigten Taste zu, während die INKEY\$-Funktion ausgeführt wird.

**ANMERKUNGEN:**

INKEY\$ wird benutzt, um auf die Betätigung einzelner Tasten zu antworten, ohne auf das Ende der Eingabe durch die -Taste zu warten.

Auf den nächsten Seite finden Sie eine Liste anwendbarer Tasten und Zeichen, die ausgegeben werden.

Wenn beim Ausführen des INKEY\$-Befehls keine Taste gedrückt wird, wird der Variable ein Nullwert zugewiesen.

Der INKEY\$-Befehl kann keine Tasteneingabe erkennen, die bei gedrückter Taste  oder direkt nach dem Drücken von  vorgenommen werden. Auch das Drücken von  und  wird nicht erkannt. Kleingeschriebene Buchstaben können nicht gelesen werden.

**Hinweis:**

Wenn der INKEY\$-Befehl sich innerhalb der ersten Zeile eines Programms befindet, kann er bereits für auf die Taste wirksam sein, die zum Starten des Programms gedrückt wird.

**BEISPIEL:**

```
10: CLS: WAIT 60
20: IF INKEY$ < > " " THEN 20
30: A$=INKEY$
40: IF A$=" " THEN 30
50: PRINT "---"; ASC A$; "---"
60: GOTO 10
```

## INKEY\$ Zeichencodetabelle

Hoch \ Tief	0	1	2	3	4	5	...	8	9	...	F
0		2ndF	SPACE	0		P					
1				1	A	Q			ln		
2	C·CE			2	B	R			log		
3	CONST			3	C	S					
4	↑			4	D	T					
5	↓	CAPS		5	E	U			sin		
6				6	F	V			cos		
7	ANS	BS		7	G	W		1/x	tan		
8	BASIC	RM	(	8	H	X		x <sup>2</sup>			
9	CAL	M+	)	9	I	Y					
A	TAB		×*		J	Z					
B	INS		+	;	K				→DEG		π
C	DEL		,		L				FSE		√
D	←		-	+/-	M				hyp		
E	▶		.		N	y <sup>x</sup>			MDF		
F	◀		+ /		O						

- Die  -Taste funktioniert als Abbruchtaste (BREAK-Taste).

## INPUT

P

- FORMAT:**
1. INPUT Variable [,Variable]
  2. INPUT "Aufforderungs-Zeichenfolge", Variable  
[ [, "Aufforderungs-Zeichenfolge"], Variable]
  3. INPUT "Aufforderungs-Zeichenfolge"; Variable  
[ [, "Aufforderungs-Zeichenfolge"]; Variable]

**Abkürzung:** 1.

**Siehe auch:** INPUT#, INKEY\$, READ, LOCATE

### WIRKUNG:

INPUT wird benutzt, um einen oder mehrere Werte über die Tastatur einzugeben.

### ANMERKUNGEN:

Wollen Sie verschiedene Werte für jeden Programmablauf eingeben, benutzen Sie INPUT, um diese Werte über die Tastatur einzugeben.

Bei Format 1 wird das Symbol "?" zur Eingabe von Daten angezeigt. Wenn bei dieser Aufforderung Daten eingegeben und die  -Taste gedrückt wird, weist der Computer diese Daten der Variable zu und fährt mit der Ausführung des Programms fort.

Wenn mehr als eine Variable spezifiziert wird, wird die Aufforderung zur Dateneingabe entsprechend wiederholt.

Bei der Aufforderung zur Eingabe wird bei Format 2 eine Buchstaben-Zeichenfolge in zwei Anführungszeichen (" ") als Hilfe angezeigt. Diese Hilfe verschwindet, wenn Daten eingegeben werden.

Bei Format 3 wird bei der Aufforderung zur Eingabe ebenfalls diese Hilfe dargestellt. Diese Hilfe verschwindet allerdings nicht und die eingegebenen Daten erscheinen nach der Hilfe.

Die Formate 1, 2 und 3 können innerhalb einer INPUT-Anweisung mehrfach verwendet werden: INPUT "A=";A,B,"C=?",C

Die Arten der Variablen, welche in einer INPUT-Anweisung gegeben werden, müssen mit der Art der eingegebenen Daten übereinstimmen. Zeichenfolge-Daten werden Zeichenfolge-Variablen zugewiesen und numerische Daten numerischen Variablen. Wenn als Antwort auf die Aufforderung für numerische Daten "ABC" eingegeben wird, werden die Werte zur Variablen AB zugewiesen. Dadurch können Werte wie SIN30 eingegeben werden.

Wurde die Startposition des Bildschirms mit Hilfe der LOCATE-Anweisung vor der Ausführung der INPUT-Anweisung angegeben, so erscheint die Aufforderungs-Zeichenfolge oder ein "?" an dieser Stelle.

**BEISPIEL:**

```
10: INPUT A
20: INPUT "A=";A
30: INPUT "A=",A
40: INPUT "X=?";X,"Y=?";Y
```

[10] Setzt ein Fragezeichen an den linken Rand.

[20] Gibt "A=" aus und wartet auf Eingabe.

[30] Gibt "A=" aus. Bei der Eingabe verschwindet "A=" und der eingegebene Wert wird von linken Rand an angezeigt.

[40] Gibt "X=?" aus und wartet auf die erste Eingabe. Nach  wird die Anzeige gelöscht und "Y=?" wird am linken Rand angezeigt.

---

# INPUT#

---

P

**FORMAT:** INPUT# 1, Variable, Variable, ..., Variable

**Abkürzung:** I.#

**Siehe auch:** OPEN, PRINT#

---

**WIRKUNG:**

Liest Teile von sequentiellen Dateien von seriellen Ein-/Ausgabegeräten (SIO) oder von Kassette.

**ANMERKUNGEN:**

Mit INPUT# werden Datenwerte von einem SIO-Gerät oder von Kassette zu den bestimmten Variablen zugewiesen.

Diese Anweisung kann nur ausgeführt werden, wenn ein SIO-Gerät mit der Dateibeschriftung "COM:" geöffnet wurde oder wenn eine Datei auf Kassette mit der Dateibeschriftung "CAS:" geöffnet und die Zugriffsform "FOR INPUT" bestimmt wurde.

Variablen werden folgendermaßen spezifiziert:

- Feste Variable (A, X, B\$, usw.)
- Einfache Variable (CD, EF\$, usw.)
- Feldelemente (B(10), C\$(5,5), usw.)
- Alle Feldvariablen (B(\*), C\$(\*) usw.)

Wenn eine Datei weniger Daten als die Anzahl der bestimmten Variablen enthält, erfolgt ein Fehler. Wenn die Datei mehr Daten enthält, wird der Rest der Daten ignoriert.

Als Begrenzung beim Einlesen von Daten zu numerischen Variablen werden Komma (,), Leerstelle (&H20), CR (&H0D), LF (&H0A) oder CR + LF verwendet. Leerstellen vor den Daten werden ignoriert. Wenn Daten, die nicht als Zahlen übersetzt werden können, einer numerischen Variable zugewiesen werden, wird ein Wert von Null angenommen.

Als Begrenzung beim Einlesen von Daten zu Buchstaben-Variablen werden Komma (,), CR, LF oder CR + LF verwendet. Leerstellen vor den Daten werden ignoriert. Mit einem Begrenzer wird automatisch das 256. ausgegebene Zeichen gelöscht. Wenn am Anfang der Daten ein doppeltes Anführungszeichen erscheint, werden die Daten bis zum folgenden doppelten Anführungszeichen gelesen. Ein Komma innerhalb einer Zeichenfolge, das von doppelten Anführungszeichen eingeschlossen ist, wird nicht als Begrenzer angesehen.

Verarbeitung des EOF-Codes (end of file: &H1A):

- Ein EOF-Code vor den Daten führt zu einem Fehler.
- Ein EOF-Code in der Mitte von Daten wird als Begrenzer erkannt.

Reihenfolge der Dateneingabe von Feldvariablen:

*Beispiele:*

Für eindimensionale Feldvariable: B (0) → B (1) → B (2) → .....

Für zweidimensionale Feldvariable: C (0, 0) → C (0, 1) → C (0, 2) → .....

C (1, 0) → C (1, 1) → C (1, 2) → .....

---

# KILL

D

---

**FORMAT:** KILL "Dateiname" 

**Abkürzung:** K.

**Siehe auch:** SAVE

---

**WIRKUNG:**

Löscht eine Datei von der RAM-Diskette.

**ANMERKUNGEN:**

Mit KILL wird eine durch den Dateinamen spezifizierte Datei von der RAM-Diskette gelöscht.

Wenn die spezifizierte Datei nicht existiert, tritt ein Fehler auf.

---

# LEFT\$

P

D

---

**FORMAT:** LEFT\$("Zeichenfolge", Ausdruck)

**Abkürzung:** LEF.

**Siehe auch:** MID\$, RIGHT\$

---

**WIRKUNG:**

Gibt eine spezifizierte Anzahl von Zeichen von der linken Seite einer Zeichenfolge aus an.

**ANMERKUNGEN:**

LEFT\$ gibt die Anzahl von Zeichen, die durch den Ausdruck spezifiziert wurden, von der linken Seite der Zeichenfolge aus an.

Der Befehl A\$="ABCD", LEFT\$(A\$,3) führt z.B. zur Anzeige der 3 ersten (linken) Buchstaben "ABC".

**BEISPIEL:**

```
10: X$="SHARP": WAIT 60
20: FOR N=1 TO 5
30: LET S$=LEFT$(X$,N)
40: PRINT S$
50: NEXT N
```

```
RUN
S
SH
SHA
SHAR
SHARP
```

---

# LEN

---

P  
D

**FORMAT:** LEN "Zeichenfolge"

**Abkürzung:**

**Siehe auch:**

---

**WIRKUNG:**

Errechnet die Anzahl der Zeichen in der Zeichenfolge.

**ANMERKUNGEN:**

Die Anzahl der Zeichen in der Zeichenfolge schließt alle Leerzeichen oder Druckzeichen wie Kontrollcodes oder Wagenrückläufe ein.

**BEISPIEL:**

```
10: INPUT "ENTER A WORD ";A$
20: N=LEN A$
30: PRINT "THE WORD LENGTH IS ";N
40: END
```

```
RUN
ENTER A WORD CHERRY
THE WORD LENGTH IS 6.
```

[10] Fordert zur Eingabe eines Wortes auf. In diesem Beispiel gibt der Anwender "CHERRY" ein.

[20] Ermittelt die Länge des Worts.

[30] Druckt das Ergebnis aus.

---

# LET

P

---

**FORMAT:** 1. LET numerische Variable = Ausdruck  
2. LET Zeichenfolge-Variable = Zeichenfolge

**Abkürzung:** LE.  
**Siehe auch:**

---

**WIRKUNG:**

LET wird benutzt, um einer Variablen einen Wert zuzuweisen.

**ANMERKUNGEN:**

Mit LET weisen Sie der angegebenen Variablen den Wert des Ausdrucks zu. Die Form des Ausdrucks muß mit der Variablen übereinstimmen, d.h. nur numerische Ausdrücke können numerischen Variablen und nur Zeichenfolge-Ausdrücke können Zeichenfolge-Variablen zugewiesen werden.

Der LET-Befehl kann in allen LET-Anweisungen weggelassen werden.

**BEISPIEL:**

10: I=10  
20: A=5\*I  
30: X\$=STR\$ A  
40: IF I >= 10 THEN LET Y\$=X\$+".00"

[10] Weist I den Wert 10 zu.

[20] Weist A den Wert 50 zu.

[30] Weist X\$ den Wert 50 zu.

[40] Weist Y\$ den Wert 50.00 zu.

---

# LFILES

D

---

**FORMAT:** LFILES 

**Abkürzung:** LF.  
**Siehe auch:** FILES

---

**WIRKUNG:**

Ausdrucken der Dateinamen, die auf der RAM-Diskette gespeichert sind.

**ANMERKUNGEN:**

Alle auf der RAM-Diskette gespeicherten Dateinamen werden ausgedruckt.

---

# LIST

D

- FORMAT:**
1. LIST 
  2. LIST Zeilennummer 
  3. LIST \*Label 

**Abkürzung:** L.

**Siehe auch:** LLIST, PASS

---

## **WIRKUNG:**

Mit dem LIST-Kommando wird ein Programm zur Anzeige gebracht.

## **ANMERKUNGEN:**

Nur gültig als direkte Eingabe in der PRO-Betriebsart.

Mit Format 1 wird das Programm von der ersten Zeile angezeigt, bis das Display voll ist.

Mit Format 2 wird das Programm von der angegebenen Zeilennummer angezeigt, bis das Display voll ist. Mit der -Taste kommen Sie in die nächste Zeile der Liste. Gibt es die angegebene Zeilennummer nicht, so wird das Programm ab der nächstgrößeren Zeile angezeigt.

Mit Format 3 wird das Programm ab der mit diesem Label versehenen Zeile angezeigt, bis das Display voll ist.

Falls ein Passwort angegeben wurde, wird das LIST-Kommando ignoriert.

Wenn ein \*Label spezifiziert wird, das in diesem Programm nicht existiert oder wenn eine Zeilennummer angegeben ist, die größer ist als die letzte Zeilennummer des Programms, erfolgt ein Fehler.

## **BEISPIEL:**

LIST 100

Zeigt das Programm ab Zeilennummer 100 an.

---

# LLIST

D

- FORMAT:**
1. LLIST 
  2. LLIST { Zeilennummer } \*Label 
  3. LLIST [Zeilennummer 1] - [Zeilennummer 2] 

**Abkürzung:** LL.

**Siehe auch:** LIST, PASS

---

**WIRKUNG:**

Druckt auf dem optionalen Drucker CE-126P ein Programm aus.

**ANMERKUNGEN:**

Nur gültig als direkte Eingabe in der PRO- oder RUN-Betriebsart.

Format 1 druckt oder sendet alle im Speicher befindlichen Programme.

Format 2 druckt oder sendet nur die Programmzeile, deren Nummern oder Label bestimmt wurden.

Format 3 druckt die Anweisungen der ersten Zeilennummer bis zur zweiten. Es müssen zwischen den Nummern mindestens zwei Zeilen liegen.

Sowohl Zeilennummer 1 oder 2 kann ausgelassen werden. Wenn Zeilennummer 1 ausgelassen wird, erfolgt die Auflistung des Programms von der ersten Zeile bis zur Zeilennummer 2. Wenn Zeilennummer 2 ausgelassen wird, erfolgt die Auflistung von Zeilennummer 1 bis zum Ende des Programms.

Wenn eine mit Format 2 gegebene Zeilennummer nicht existiert oder die Zeilen mit Zeilennummern 1 und 2 in Format 3 nicht existieren, wird die nächste größere Nummer angenommen.

Falls ein Kennwort eingegeben wurde, wird das LLIST-Kommando ignoriert.

**BEISPIEL:**

LLIST 100 - 200 

Stellt die Programmauflistung zwischen Zeilennummern 100 und 200 dar.

LLIST - 200 

Stellt die Programmauflistung von der ersten bis zur Zeile 200 dar.

LLIST 100 - 

Stellt die Programmauflistung von Zeile 100 bis zur letzten Zeile dar.

---

# LOAD

D

---

**FORMAT:** LOAD "Dateiname"

**Abkürzung:** LO.

**Siehe auch:** SAVE, FILES

---

**WIRKUNG:**

Laden eines BASIC-Programms auf die RAM-Diskette.

**ANMERKUNGEN:**

Wenn durch das Laden des Programms der Programm-Speicherbereich überschritten wird, erfolgt ein Fehler. In diesem Fall müssen unnötige Variable aus dem Datenbereich gelöscht werden.

Wenn der Dateiname eines Programms mit dem FILES-Befehl dargestellt wurde, kann das gewünschte Programm einfach durch Anwählen mit der Taste  oder  und Eingabe von  +  geladen werden.

Die Datei-Erweiterung kann nur ausgelassen werden, wenn sie ".BAS" ist.

Während ein Programm geladen wird, werden alle offenen Dateien geschlossen.

Beim Versuch, ein TEXT-Programm zu laden, erfolgt ein Fehler.



---

# LPRINT

---

P  
D

- FORMAT:**
1. LPRINT {Ausdruck  
Zeichenfolge} [', {Ausdruck  
Zeichenfolge}]
  2. LPRINT {Ausdruck  
Zeichenfolge} ['; {Ausdruck  
Zeichenfolge}] [:]
  3. LPRINT USING "Format"; {Ausdruck  
Zeichenfolge} [{', } {Ausdruck  
Zeichenfolge}] [:]
  4. LPRINT

**Abkürzung:** LP.

**Siehe auch:** PRINT, USING

---

## **WIRKUNG:**

Ausgabe von Daten an den Drucker.

## **ANMERKUNGEN:**

Wenn am Ende einer Anweisung ein Semikolon (;) gegeben ist, kommt es beim folgenden LPRINT-Befehl zum Ausdruck der Daten direkt nach denjenigen Daten, die mit dem ersten LPRINT-Befehl ausgedruckt wurden. Am Ende einer Anweisung darf kein Komma (,) stehen.

Wenn mit Format 1 für den Drucker CE-126P nur ein einziger Gegenstand spezifiziert wurde, werden numerische Werte rechtsbündig und Zeichenfolgen linksbündig auf dem Papier ausgegeben.

Zur Trennung von Ausdrücken oder Zeichenfolge werden Kommas (,) verwendet, um jeweils 24 Spalten für jede Druckzeile in einen Bereich von 12 Spalten einzuteilen. Numerische Werte und Zeichenfolgen werden in diesen Bereichen ausgedruckt; numerische Berechnungen werden rechtsbündig in jedem 12-Spalten-Bereich ausgegeben, Zeichenfolgen dagegen linksbündig. Bei numerischen Werten mit mehr als 12 Stellen wird beim Ausdruck der zusätzliche Teil der Mantisse abgeschnitten. Bei Zeichenfolgen mit mehr als 12 Zeichen werden nur die ersten 12 Zeichen ausgedruckt, alle anderen Zeichen werden ignoriert.

Mit Format 2 werden die Daten der Reihe nach linksbündig ausgedruckt.

Wenn am Ende einer Anweisung ein Semikolon (;) steht, werden mit dem folgenden LPRINT-Befehl des Programms die Daten in der Reihenfolge nach den Daten des ersten LPRINT-Befehls ausgedruckt.

Mit Format 3 werden die Daten genau in dem Format ausgedruckt, das in der Anweisung spezifiziert wurde. Zur Trennung können ein Komma (,) oder ein Semikolon (;) verwendet werden.

Für das Format mit dem USING-Befehl siehe dort.

Mit Format 4 werden nur Begrenzer-Codes ausgedruckt. Wenn die vorhergehende LPRINT-Anweisung mit einem Semikolon (;) beendet wurde und noch nicht gedruckte Daten im Speicher verblieben sind, werden diese Daten mit dem Format 4 ausgedruckt.

## **BEISPIEL:**

LPRINT A,B

LPRINT A;B;Z\$

---

# MID\$

---

P  
D

**FORMAT:** MID\$ (Zeichenfolge,N,M)

**Abkürzung:** MI.

**Siehe auch:** LEFT\$, RIGHT\$

---

**WIRKUNG:**

Holt eine Zeichenfolge von M Zeichen aus einer Zeichenfolge, beginnend beim N-ten Zeichen in der Zeichenfolge.

**ANMERKUNGEN:**

Ist N größer als die Anzahl von Zeichen in der Zeichenfolge, wird eine Null-Zeichenfolge erstellt. Wenn N kleiner als 1 ist, tritt ein Fehler auf. M muß zwischen 0 und 255 liegen und N muß zwischen 1 und 255 liegen. Bruchteile werden nicht berücksichtigt.

**BEISPIEL:**

10: Z\$="ABCDEFGH"  
20: Y\$= MID\$(Z\$,3,4)  
30: PRINT Y\$

---

# MON

---

D

**FORMAT:** MON 

**Abkürzung:** MO.

**Siehe auch:**

---

**WIRKUNG:**

Wahl der Betriebsart für den Maschinensprachen-Monitor.

**ANMERKUNGEN:**

Kann nur als direkte Eingabe in der PRO- oder RUN-Betriebsart verwendet werden. Mit MON wird der Computer auf die Betriebsart für den Maschinensprachen-Monitor eingestellt. (Siehe die Erklärungen über den Maschinensprachen-Monitor.)

---

# NEW

D

---

**FORMAT:** NEW

**Abkürzung:**

**Siehe auch:** CLEAR, PASS

---

**WIRKUNG:**

NEW wird benutzt, um bestehende Programme oder Daten zu löschen.

**ANMERKUNGEN:**

Das NEW-Kommando löscht alle Programme und Daten, die sich momentan im Speicher befinden. (Die mit Kennwort geschützten Programme können nicht gelöscht werden.)

Eine offene Datei wird geschlossen.

---

**BEISPIEL:**

NEW

---

# ON...GOSUB

---

P

**FORMAT:** ON Ausdruck GOSUB { Zeilennummer 1 } { Zeilennummer 2 }  
{ \*Label 1 } { \*Label 2 }, ...

**Abkürzung:** O. GOS.

**Siehe auch:** GOSUB, GOTO, ON...GOTO

---

**WIRKUNG:**

ON...GOSUB wird benutzt, um eine der gegebenen Subroutinen abhängig vom Wert eines Kontrollausdrucks auszuführen.

**ANMERKUNGEN:**

Bei der Ausführung von ON...GOSUB wird der Ausdruck zwischen ON und GOSUB ausgewertet und auf einen ganzzahligen Wert reduziert. Ist der ganzzahlige Wert 1, so springt die Kontrolle auf die Zeilennummer 1 oder \*Label 1 der Liste, wie in einem normalen GOSUB. Ist der Wert gleich 2, springt die Kontrolle zur Zeilennummer 2 oder \*Label 2 in der Liste.

**Hinweis:**

Direkt vor dem GOSUB-Befehl muß unbedingt eine Leerstelle eingefügt werden. Andernfalls wird es als eine Variable angesehen.

Wenn der Wert des Ausdrucks Null, negativ oder größer als die Anzahl der in der Liste gegebenen Subroutinen ist, wird keine Subroutine ausgeführt und die Programmausführung geht weiter mit der nächsten Zeile.

Die Zeilennummern oder \*Labels in der Liste durch Kommas (,) trennen.

**BEISPIEL:**

```
10: INPUT A
20: ON A GOSUB 100,200,300
30: END
100: PRINT "FIRST"
110: RETURN
200: PRINT "SECOND"
210: RETURN
300: PRINT "THIRD"
310: RETURN
```

Die Eingabe einer 1 zeigt "FIRST" an, eine 2 zeigt "SECOND" an, eine 3 zeigt "THIRD" an. Auf jede andere Eingabe wird nichts ausgegeben.



---

# OPEN

---

P  
D

**FORMAT:** 1. OPEN "COM:"  
2. OPEN "CAS:[Dateiname]" FOR OUTPUT  
3. OPEN "CAS:[Dateiname]" FOR INPUT

**Abkürzung:** OP.  
**Siehe auch:** CLOSE

---

## **WIRKUNG:**

Öffnet ein SIO-Gerät (serielle Ein-/Ausgabe) oder eine Datei auf Kassette für die Eingabe bzw. Ausgabe von Daten.

## **ANMERKUNGEN:**

Mit Format 1 wird ein SIO-Gerät für die Eingabe oder Ausgabe von Daten geöffnet (der SIO-Schaltkreis wird geöffnet). Der Dateiname und die Art des Zugriffs kann nicht spezifiziert werden. Die Parameter der seriellen Kommunikation werden in der TEXT-Betriebsart eingegeben. (Siehe Seite 105.)

Mit Format 2 wird eine Datei auf Kassette für die Ausgabe von Daten geöffnet. Vor der Ausführung dieses Befehls muß sichergestellt werden, daß der Kassettenrekorder auf Aufnahme gestellt ist.

Während der Ausführung wird mit diesem Befehl ein Informationsblock (Kopfzeile) auf das Band geschrieben. Wenn ein Dateiname spezifiziert wurde, wird er ebenfalls in diesem Informationsblock aufgezeichnet.

Mit Format 3 wird eine Datei auf Kassette für die Eingabe geöffnet. Vor der Ausführung dieses Befehls muß sichergestellt werden, daß der Kassettenrekorder auf Wiedergabe gestellt ist.

Mit diesem Befehl wird nach dem Informationsblock gesucht, der den Dateinamen enthält und dann die entsprechende Datei vom Band geladen. Wenn kein Dateiname spezifiziert wurde, lädt der Computer die erste Datei, deren Dateiname als erstes in dem ersten Informationsblock nach dem Starten des Bandes gefunden wird.

## **Hinweise:**

- Wenn der eingegebene Dateiname nicht gefunden wird, setzt der Computer die Suche fort, auch nachdem das Bandende erreicht ist. In diesem Fall **BREAK** drücken, um die Suche abzubrechen.
- Es kann nur jeweils eine Datei geöffnet werden. Wenn der OPEN-Befehl ausgeführt wird, während bereits eine Datei geöffnet ist, erfolgt ein Fehler.

---

# PASS

---

D

**FORMAT:** PASS "Zeichenfolge" 

**Abkürzung:** PA.

**Siehe auch:** CSAVE, SAVE, CLOAD, LOAD, DELETE, LIST, NEW, RENUM

---

## **WIRKUNG:**

Mit dem PASS-Kommando können Kennwörter gesetzt und gelöscht werden.

## **ANMERKUNGEN:**

Kennwörter werden benutzt, um Programme von Einsichtnahme oder Modifikation durch andere zu schützen. Die 8 Zeichen müssen alphanumerische Zeichen oder Sonderzeichen sein. Die Zeichenfolge darf nicht nur Nullen enthalten.

Wurde das PASS-Kommando gegeben, so sind die Programme im Speicher geschützt. Ein Kennwort-geschütztes Programm kann im Speicher nicht überprüft oder modifiziert werden. Es kann nicht auf Band oder auf der RAM-Diskette abgespeichert oder mit LIST oder LLIST ausgegeben werden; weiterhin können weder Programmzeilen hinzugefügt noch gelöscht werden. Die einzige Möglichkeit, diesen Schutz auszuschalten, ist ein weiteres PASS-Kommando mit dem gleichen Kennwort einzugeben.

Wenn bei dem zu ladenden Programm ein Kennwort eingegeben wurde, gilt dieses Kennwort auch für den Computer. Andernfalls gilt keine Kennwort für den Computer. Wenn PASS ausgeführt wird, während kein Programm im Computer ist, erfolgt ein Fehler und kein Kennwort wird eingegeben.

Ein Kennwort-geschütztes Programm ist auch vor den Befehlen NEW oder DELETE geschützt.

## **BEISPIEL:**

PASS "SECRET" 

Erstellen des Kennwortes "SECRET" für das gespeicherte Programm.

---

# PRINT

---

P  
D

- FORMAT:**
1. PRINT  $\left\{ \begin{array}{l} \text{Ausdruck} \\ \text{Zeichenfolge} \end{array} \right\} \left[ , \left\{ \begin{array}{l} \text{Ausdruck} \\ \text{Zeichenfolge} \end{array} \right\} \right]$
  2. PRINT  $\left\{ \begin{array}{l} \text{Ausdruck} \\ \text{Zeichenfolge} \end{array} \right\} \left[ ; \left\{ \begin{array}{l} \text{Ausdruck} \\ \text{Zeichenfolge} \end{array} \right\} \right] [;]$
  3. PRINT USING "Format";  $\left\{ \begin{array}{l} \text{Ausdruck} \\ \text{Zeichenfolge} \end{array} \right\} \left[ \left[ , \right] \left\{ \begin{array}{l} \text{Ausdruck} \\ \text{Zeichenfolge} \end{array} \right\} \right] [;]$
  4. PRINT

**Abkürzung:** P.

**Siehe auch:** LPRINT, USING, WAIT, LOCATE

---

## **WIRKUNG:**

Anzeige von Informationen.

## **ANMERKUNGEN:**

PRINT dient der Anzeige von Aufforderungen, Informationen, Ergebnissen von Berechnungen u.a.

Wenn der Anfangspunkt durch den Befehl LOCATE bestimmt wird, werden die Daten von diesem Punkt aus dargestellt.

Wenn am Ende einer Anweisung ein Semikolon (;) gegeben ist, wird der Inhalt kontinuierlich dargestellt. Am Ende einer Anweisung darf kein Komma (,) stehen.

Format 1 zeigt folgendes an:

1) Für einen einzigen dargestellten Teil:

Wenn der Ausdruck numerisch ist, wird der Wert rechtsbündig auf dem Display dargestellt. Eine Zeichenfolge wird linksbündig dargestellt.

## **BEISPIEL:**

10: PRINT 1234

20: PRINT "ABCD"

ABCD	1234.
------	-------

2) Für 2 oder mehr dargestellte Teile (durch Kommas spezifiziert):

Die 24 Spalten jeder Zeile werden in zwei Bereiche mit je 12 Zeichen eingeteilt.

Numerische Werte werden in jedem Bereich mit 12 Spalten rechtsbündig angezeigt und Zeichenfolgen linksbündig.

**BEISPIEL:**

```
10: A = 1234: B = 5/9: C$ = "ABCDE":WAIT 200
20: CLS: PRINT "A=",A
30: CLS: PRINT A,C$,B
```

RUN 

```
A = 1 2 3 4 .
```

```
1 2 3 4 . A B C D E
5 . 5 5 5 5 5 E - 0 1
```

Wenn ein numerischer Wert mehr als 12 Stellen hat (wenn der Dezimalanteil eines Exponenten mehr als 7 Stellen hat), werden die kleinsten Stellen abgeschnitten. Wenn eine Zeichenfolge mehr als 12 Spalten einnimmt, werden nur die ersten 12 Zeichen (von links) dargestellt.

Format 2 zeigt die Daten kontinuierlich von der linken Seite des Displays aus an.

**BEISPIEL:**

```
10: A = 1234:B = 5/9:C$ = "ABCDE"
20: PRINT "A=";A
30: PRINT "EFGHI";B;C$;A
RUN 
```

```
A=1 2 3 4 .
E F G H I 5 . 5 5 5 5 5 5 5 5 6 E - 0 1 A B C D
E 1 2 3 4 .
>
```

Format 3 zeigt die Daten an, die durch das folgende Format spezifiziert werden. Für das Format mit dem USING-Befehl siehe dort. Kommas (,) und Semikolon (;) werden wie normal behandelt. Die USING-Anweisung gilt auch für die nächste PRINT-Anweisung, die im Programm gefunden wird.

*Beispiel:* PRINT USING "&&&&&&&";"ANSWER=";:PRINT USING "####.##";5/9

Format 4 zeigt den vorher dargestellten Wert so wie er ist an. (Normalerweise wird es zusammen mit dem WAIT-Befehl verwendet, um eine dargestellte Anzeige zu erhalten.)

**BEISPIEL:**

```
10: CLS
20: FOR A=0 TO 159
30: PRINT CHR$(A+32);
40: NEXT A
50: WAIT: PRINT
```

Die Zeichen, die zwischen Zeile 20 und 40 dargestellt werden, bleiben auch bei Darstellung von Zeile 50 erhalten. (Ein unbegrenztes Intervall wurde eingegeben.)

### **PRINT → LPRINT-Einstellung**

Der Computer kann alle PRINT-Befehle so umstellen, daß sie wie LPRINT-Befehle funktionieren. Vor Ausführung der folgenden Anweisung muß der Drucker angeschlossen werden:

Einstellung: PRINT=LPRINT

Rückstellung: PRINT=PRINT

Die Rückstellung kann auch folgendermaßen vorgenommen werden:

- 1) Ausführung eines RUN-Befehls.
- 2) Drücken der Tasten **SHIFT** + **CA** .
- 3) Ausschalten und erneutes Einschalten des Gerätes.

Da mit dem RUN-Befehl alle Einstellungen zurückgestellt werden, kann das Programm mit einem GOTO-Befehl ausgeführt werden.

# PRINT#

**FORMAT:** PRINT# 1, {Ausdruck  
Zeichenfolge} [ {,} ] {Ausdruck  
Zeichenfolge} [ {,} ] [ {,} ]

**Abkürzung:** P.#

**Siehe auch:** OPEN, INPUT#

## WIRKUNG:

Schreiben von Werten der spezifizierten Variablen auf ein SIO-Gerät (serielle Ein-/Ausgabe) oder auf Kassette.

## ANMERKUNGEN:

PRINT# kann nur ausgeführt werden, wenn ein SIO-Gerät mit der Dateibeschreibung "COM:" geöffnet wurde oder wenn eine Datei auf Kassette mit der Dateibeschreibung "CAS:" geöffnet und die Zugriffsform "FOR OUTPUT" bestimmt wurde.

Wenn eine Feldvariable (eine oder zwei Dimensionen) in der Form "Feldvariable(\*)" spezifiziert wurde, wird das gesamte Feld in die Datei geschrieben. Die Elemente werden dabei in Reihenfolge geschrieben, z.B. C\$(0,0), C\$(0,1), C\$(0,2) ... C\$(1,0)... C\$(5,5).

Zum Schreiben von Daten für Feldvariable wird die Anweisung FOR...NEXT empfohlen.

Wenn die entsprechenden Elemente eines Feldes spezifiziert werden, müssen sie in der Form "B(7)", "C\$(5,6)" usw. eingegeben werden.

Bei Verwendung von Buchstaben oder Zeichenfolge-Elementen dürfen sie nicht mit einem Semikolon (;) spezifiziert werden:

PRINT#1,"ABC"

PRINT#1,A\$

Bei Ausführung von PRINT#1,"ABC";A\$ wird kein Daten-Begrenzer geschrieben und es kann nicht zwischen "ABC" und A\$ unterschieden werden.

Ein numerischer Wert wird in der Reihenfolge Vorzeichen (Leerstelle wenn positiv), numerische Zeichenfolge, Leerstelle geschrieben.

Das Eingabeformat wird im folgenden gezeigt:

(1) Wenn kein Komma oder Semikolon den Daten folgt, wird CR(&H0D) und LF(&H0A) angenommen.

*Beispiel:*

PRINT#1,-1.2.

-	1	.	2		CR	LF
---	---	---	---	--	----	----

PRINT#1,"ABC"

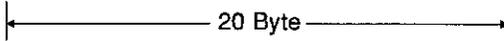
A	B	C	CR	LF
---	---	---	----	----

Wenn die Daten mit einem Komma (,) oder einem Semikolon (;) beendet werden, wird der Code CR+LF nicht ausgegeben.

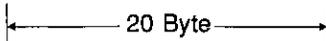
(2) Wenn den Daten ein Komma folgt, werden 20 Byte besetzt. Ein numerischer Wert wird rechtsbündig und eine Zeichenfolge linksbündig geschrieben.

*Beispiel:*

PRINT#1,-1.2,3



PRINT#1,"ABC","DEF"



Wenn die Zeichenfolge 20 Byte übersteigt, wird der Überlauf in den nächsten Bereich für 20 Byte geschrieben.

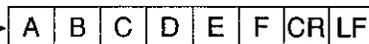
(3) Wenn den Daten ein Semikolon folgt, erfolgt die Speicherung ohne Zwischenräume.

*Beispiel:*

PRINT#1,-1.2;3



PRINT#1,"ABC";"DEF"



→ In diesem Fall werden die Zeichenfolgen "ABC" und "DEF" nicht getrennt gelesen.

Wenn alle Variable eines Feldes spezifiziert werden, wird jede Ausgabe einer Variablen von dem Code CR+LF gefolgt. Die Anweisung PRINT#1, B(\*) entspricht der Anweisung PRINT#1, B(0), die Anweisung PRINT#1, B(1) entspricht PRINT#1, B(2) und so weiter.

Exponentielle Werte werden bei Ausgabe der Mantisse von dem Symbol E, dem Vorzeichen und einem zweistelligen Exponenten gefolgt (einem einstelligen Exponenten muß eine Null vorgestellt werden).

**BEISPIEL:**

```
10: OPEN "CAS:DATA" FOR OUTPUT
20: FOR J = 0 TO N
30: FOR K = 0 TO M
40: PRINT#1,C$(J,K)
50: NEXT K:NEXT J:CLOSE
```

---

# RADIAN

---

P  
D

**FORMAT:** RADIAN

**Abkürzung:** RAD.

**Siehe auch:** DEGREE, GRAD

---

**WIRKUNG:**

RADIAN wird benutzt, um die Form von Winkelwerten auf Bogenmaß umzuschalten.

**ANMERKUNGEN:**

Der Computer kann 3 Formen der Winkeldarstellung verarbeiten: Altgrad, Bogenmaß und Neugrad. Diese Formen werden für die Darstellung der Werte der Argumente von den SINus-, COSinus- und TANgens-Funktionen und der Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens benutzt.

Die RADIAN-Funktion wird benutzt, um Winkelwerte auf Bogenmaß umzuformen, bis DEGREE oder GRAD eingestellt wird. Die Bogenmaß-Form gibt die Winkel als Bogenmaß in Abhängigkeit vom Radius wieder, z.B.  $360^\circ$  sind  $2\pi$ , da der Einheitskreis der Umfang  $2\pi$  mal Radius hat.

**BEISPIEL:**

10: RADIAN

20: X = ASN 1

30: PRINT X

X hat den Wert 1,570796327 oder  $\pi/2$ , der Arcussinus von 1.

---

# RANDOMIZE

---

P  
D

**FORMAT:** RANDOMIZE

**Abkürzung:** RA.

**Siehe auch:** RND

---

**WIRKUNG:**

RANDOMIZE wird benutzt, um eine neue Startzahl für den Zufallsgenerator zu setzen.

**ANMERKUNGEN:**

Wenn mit der RND-Funktion Zufallszahlen erzeugt werden, so startet der Computer bei einer vorgegebenen Startzahl. Der RANDOMIZE-Befehl setzt diese Startzahl auf einen neuen, zufällig gewählten Wert.

Die Startzahl ist jedesmal die gleiche, wenn der Computer eingeschaltet wird, so daß sich die Sequenz der Zufallszahlen mit RND jedesmal wiederholt, es sei denn, die Startzahl wird gewechselt. Diese Eigenschaft ist bei der Entwicklung eines Programms sehr wichtig, denn dies bedeutet, daß das Verhalten des Programms immer gleich sein muß, jedesmal, wenn es abläuft, obwohl es eine RND-Funktion enthält. Wenn Sie wirklich zufällige Zahlen haben wollen, kann mit der RANDOMIZE-Anweisung die Startzahl selbst zufällig ermittelt werden.

**BEISPIEL:**

10: RANDOMIZE

20: X = RND 10

Bei Start in Zeile 20 beruht der Wert von X auf einer Standard-Startzahl. Bei Start in Zeile 10 wird eine neue Basis ermittelt.

---

# READ

---

P

**FORMAT:** READ Variable, Variable, ..., Variable

**Abkürzung:** REA.

**Siehe auch:** DATA, RESTORE

---

**WIRKUNG:**

READ wird gebraucht, um Werte aus der DATA-Anweisung herauszulesen und sie Variablen zuzuweisen.

**ANMERKUNGEN:**

Wenn einem Feld Basiswerte zugewiesen werden sollen, so ist es sinnvoll, diese Werte in einer DATA-Anweisung unterzubringen und sie von dort mit einer READ-Anweisung in einer FOR...NEXT-Schleife in das Feld zu übertragen. Die folgenden READ-Befehle benutzen die folgenden Werte in der Reihenfolge ihrer Erscheinung im Programm, unabhängig davon, wieviele Werte in jeder DATA-Anweisung eingesetzt werden und wieviele DATA-Anweisungen verwendet werden.

Wenn nötig, können dieselben Werte in einer DATA-Anweisung ein zweites mal mit Hilfe der RESTORE-Anweisung gelesen werden.

**Hinweis:**

Die Art der Daten muß der Art der zugewiesenen Variablen entsprechen (numerisch oder Zeichenfolge).

**BEISPIEL:**

```
10: DIM B(10)
20: WAIT 60
30: FOR I = 1 TO 10
40: READ B (I)
50: PRINT B(I)*2;
60: NEXT I
70: DATA 10, 20, 30, 40, 50, 60
80: DATA 70, 80, 90, 100
90: READ C, D, E$, F$
100: PRINT C,D,E$,F$
110: DATA 3,5,G=,H=
120: END
```

[10] Dimensionierung eines Feldes.

[40] Lädt die Werte aus der DATA-Anweisung in B( ) ein; B(1) ist 10, B(2) ist 20, B(3) ist 30, usw.

[90] Laden der Werte der letzten DATA-Anweisung. C ist 3, D ist 5, E\$ ist G= und F\$ ist H=.

---

# REM(')

---

P

**FORMAT:** REM Kommentar oder ' Kommentar

**Abkürzung:**

**Siehe auch:**

---

**WIRKUNG:**

REM wird benutzt, um Kommentare in ein Programm einzufügen.

**ANMERKUNGEN:**

Manchmal ist es sinnvoll, in ein Programm erläuternde Kommentare einzufügen. Es kann sich hierbei um Titel, Autorennamen, Daten der letzten Änderungen, Anwendungshinweise, Erinnerungshinweise für Algorithmen, usw. handeln. Solche Kommentare können mit Hilfe der REM-Anweisung oder mit Apostroph (') eingefügt werden.

Die REM-Anweisung hat keinen Einfluß auf den Programmablauf und kann überall im Programm eingefügt werden. Alles, was in dieser Zeile nach dem REM-Befehl kommt, wird als Kommentar behandelt.

**BEISPIEL:**

```
10: ' THIS LINE HAS NO EFFECT
100: REM THIS LINE HAS NO EFFECT EITHER.
```

---

# RENUM

D

---

**FORMAT:** RENUM [neue Zeilennummer] [, [alte Zeilennummer] [,Inkrement]] 

**Abkürzung:** REN.

**Siehe auch:** DELETE, LIST

---

**WIRKUNG:**

Führt die Neunummerierung der Zeilen eines Programms aus.

**ANMERKUNGEN:**

Nur als direkte Eingabe in der PRO-Betriebsart zulässig.

Die Zeilennummern werden mit dem vorgegebenen Inkrement von der alten Zeilennummer auf die neue Zeilennummer geändert. Wird die neue Zeilennummer nicht angegeben, erfolgt die Neunummerierung von Zeile 10 an. Wenn das Inkrement nicht angegeben wird, erfolgt die Neunummerierung mit einem Inkrement von 10. RENUM aktualisiert die angegebenen Zeilennummern in den Anweisungen GOTO, ON...GOTO, GOSUB, ON...GOSUB, RESTORE und (IF)...THEN.

Wenn eine Zeilennummer als Variable (Beispiel: GOTO A) oder als numerischer Ausdruck (Beispiel: GOTO 2  50) angegeben ist, wird diese Zeile nicht ordnungsgemäß neu numeriert.

Wenn eine Zeilennummer durch eine Variable oder einen Ausdruck gegeben ist, muß sie kurzfristig in eine Bemerkung (REM) umgesetzt und nach Ausführung des RENUM-Befehls wieder korrigiert werden. Es wird empfohlen, derartige Befehle mit ON...GOTO u.a. zu ersetzen.

Ist eine Zeilennummer größer als 65279, entsteht ein Fehler. Existiert eine angegebene Zeilennummer nicht, erhalten Sie eine Fehlermeldung. Wenn Sie die Reihenfolge der Ausführung ändern, erfolgt ein Fehler. (Beispiel: RENUM 15,30, wenn die Zeilennummern 10, 20 und 30 bereits bestehen.) Wenn Sie ein Kennwort benutzt haben, erfolgt ein Fehler.

Wenn auf der Anzeige das Zeichen "\*" erscheint, drücken Sie die  -Taste, um die Neunummerierung zu unterbrechen. Wenn "x\*" erscheint, bedeutet dies, daß die Unterbrechung nicht möglich ist. Die Fehlererzeugung oder der Gebrauch der  -Taste läßt das Programm unverändert.

**BEISPIEL:**

```
10: INPUT "CONTINUE";A$
20: IF A$ = "YES" THEN 10
30: IF A$ = "NO" THEN 60
40: PRINT "ENTER YES OR NO PLEASE!"
50: GOTO 10
60: END
```

RENUM 100, 10, 5 

```
100: INPUT "CONTINUE";A$
105: IF A$ = "YES" THEN 100
110: IF A$ = "NO" THEN 125
115: PRINT "ENTER YES OR NO PLEASE!"
120: GOTO 100
125: END
```

---

# RESTORE

---

P

**FORMAT:** 1. RESTORE { Zeilennummer }  
                  \*Label }  
          2. RESTORE

**Abkürzung:** RES.

**Siehe auch:** DATA, READ

---

**WIRKUNG:**

RESTORE wird benutzt, um Werte in einer DATA-Anweisung mehrmals zu lesen oder die Reihenfolge, in der diese Werte gelesen werden sollen, zu ändern.

**ANMERKUNGEN:**

Beim normalen Gebrauch von READ beginnt der Computer das Lesen beim ersten Wert in einer DATA-Anweisung und geht sequentiell durch alle übrigen Werte. Format 1 setzt den Zeiger auf den ersten Wert der ersten DATA-Anweisung zurück, deren Zeilennummer oder \*Label gleich dem Wert des Ausdrucks ist. Format 2 setzt den Zeiger auf den ersten Wert der DATA-Anweisung zurück, damit er noch einmal gelesen werden kann.

**BEISPIEL:**

```
10: DIM B(10)
20: WAIT 32
30: FOR I = 1 TO 10
40: RESTORE
50: READ B(I)
60: PRINT B(I)*I;
70: NEXT I
80: DATA 20
90: END
```

[10] Dimensionierung eines Feldes

[50] Weist jedem Element von B( ) den Wert von 20 zu.

---

# RIGHT\$

---

P  
D

**FORMAT:** RIGHT\$(Zeichenfolge,N)

**Abkürzung:** RI.

**Siehe auch:** LEFT\$, MID\$

---

**WIRKUNG:**

Zeigt eine Zeichenfolge von N Zeichen vom rechten Ende der Zeichenfolge.

**ANMERKUNGEN:**

Bruchteile werden nicht berücksichtigt. Ist N kleiner als 1, wird eine Null-Zeichenfolge angezeigt. Ist N größer als die Anzahl der Zeichen in der Zeichenfolge, wird die ganze Zeichenfolge angezeigt.

**BEISPIEL:**

5: WAIT 60

10: XX\$ = "SHARP COMPUTER"

20: FOR N = 1 TO 14

30: SS\$ = RIGHT\$(XX\$,N)

40: PRINT SS\$

50: NEXT N

---

# RND

---

P  
D

**FORMAT:** RND numerischer Ausdruck

**Abkürzung:** RN.

**Siehe auch:** RANDOMIZE

---

**WIRKUNG:**

RND erzeugt eine Zufallszahl.

**ANMERKUNGEN:**

Ist der Wert des Ausdrucks kleiner als 2, aber größer oder gleich Null, so ist die Zufallszahl kleiner als 1 und größer oder gleich Null. Ist der Ausdruck eine ganze Zahl, die größer oder gleich 2 ist, ist das Ergebnis eine Zufallszahl, die größer oder gleich 1 oder kleiner oder gleich dem Ausdruck ist. Ist der Ausdruck größer oder gleich 2 und keine ganze Zahl, ist das Ergebnis eine Zufallszahl, die größer oder gleich 1 und kleiner oder gleich der kleinsten ganze Zahl ist, die wiederum größer als der Ausdruck ist (in diesem Fall ändert sich die Erzeugung der Zufallszahl, je nach dem Wert des Dezimalteils des Ausdrucks). Wenn der Ausdruck negativ ist, wird der vorher eingegebene numerische Wert verwendet, um eine Zufallszahl zu generieren.

<u>Argument</u>	<u>----- Ergebnis -----</u>	
	<u>Untere Grenze</u>	<u>Obere Grenze</u>
.5	0<	<1
2	1	2
2.5	1	3

Normalerweise wird die gleiche Sequenz von Zufallszahlen erzeugt, da jedesmal die gleiche Startzahl benutzt wird, wenn der Computer eingeschaltet wird. Um diese Startzahl zufällig zu ändern, verwenden Sie das RANDOMIZE-Kommando.

---

# RUN

D

**FORMAT:** 1. RUN

2. RUN { Zeilennummer }   
          \*Label

**Abkürzung:** R.  
**Siehe auch:** GOTO

---

**WIRKUNG:**

Mit dem RUN-Kommando wird das Programm im Speicher gestartet.

**ANMERKUNGEN:**

Format 1 startet ein Programm mit der Zeile, die die niedrigste Zeilennummer im Speicher hat.

Format 2 startet ein Programm mit der angegebenen Zeilennummer.

Wenn die spezifizierte Zeilennummer oder das \*Label nicht gefunden wird, erfolgt ein Fehler.

Wenn zwei oder mehr identische Label in einem Programm enthalten sind, wird dasjenige mit der kleinsten Zeilennummer ausgeführt.

**BEISPIEL:**

RUN 100

Beginnt die Ausführung des Programms ab Zeile 100.

---

# SAVE

---

D

**FORMAT:** SAVE "Dateiname"

**Abkürzung:** SA.

**Siehe auch:** LOAD, FILES

---

**WIRKUNG:**

Sichert ein BASIC-Programm auf die RAM-Diskette.

**ANMERKUNGEN:**

Mit der SAVE-Anweisung wird ein BASIC-Programm aus dem Speicher benannt und dann auf die RAM-Diskette geschrieben.

Ein Dateiname ist ein Name, der einem Programm gegeben wird. Die gewünschte Datei kann vom Computer immer schnell wiedergefunden werden, wenn dieser Dateiname benutzt wird.

Ein Dateiname kann aus bis zu 8 alphanumerischen Zeichen oder Symbolen bestehen.

Wenn keine Erweiterung angegeben ist, wird .BAS angenommen. Die Erweiterung kann aus bis zu 3 Zeichen bestehen.

Eine bestehende Datei wird gelöscht, wenn der gleiche Dateiname spezifiziert wird, aber ein Fehler tritt auf, wenn die bestehende Datei eine TEXT-Datei ist.

Der SAVE-Befehl wird ignoriert, wenn kein Programm geladen oder das Programm durch ein Kennwort geschützt ist.

---

# STOP

---

P

**FORMAT:** STOP

**Abkürzung:** S.

**Siehe auch:** CONT

---

**WIRKUNG:**

STOP wird benutzt, um die Ausführung eines Programms für Diagnosezwecke zu stoppen.

**ANMERKUNGEN:**

Wenn STOP während des Programmablaufs angetroffen wird, stoppt die Ausführung und es erscheint eine Meldung wie "BREAK IN 200"; dabei ist 200 die Nummer der Zeile, die STOP enthält. STOP wird während der Entwicklung eines Programms benutzt, um den Ablauf des Programms zu kontrollieren oder den Zustand der Variablen zu überprüfen. Die Ausführung kann mit dem Befehl CONT wieder aufgenommen werden. Beim Drücken der Taste  wird das Programm Zeile für Zeile ausgeführt.

**BEISPIEL:**

10: STOP

Läßt die Meldung "Break in 10" auf der Anzeige erscheinen.

---

# STR\$

---

P  
D

**FORMAT:** STR\$ Ausdruck

**Abkürzung:** STR.

**Siehe auch:** VAL

---

**WIRKUNG:**

Wandelt numerische Daten in Zeichenfolgedaten um.

**ANMERKUNGEN:**

Die STR\$-Funktion wandelt numerische Daten in eine Zeichenfolge um. Die Zeichenfolge besteht aus den gleichen Ziffern wie die ursprüngliche Zahl. Die STR\$-Funktion hat den entgegengesetzten Effekt der VAL-Funktion.

Sind die numerischen Daten negativ, so steht vor der Zeichenfolge ein Minuszeichen (-).

**BEISPIEL:**

```
:  
:  
110: N=N*3  
120: A$=STR$ N  
130: B$=LEFT$ (A$,3)  
140: M=VAL B$  
:
```

[110] Das Programm führt einige Berechnungen mit der numerischen Variable N aus.

[120] Die numerische Variable N wird in die Zeichenfolge-Variablen A\$ umgewandelt.

Zeichenfolge-Variablen können viel leichter bearbeitet werden als numerische Zeichen. In diesem Beispiel nehmen wir an, daß die ersten drei Ziffern der Zahl benötigt werden. Da wir die Zahl in eine Zeichenfolge umgewandelt haben, können wir irgendeines der Zeichenfolgen-Bearbeitungskommandos wie LEFT\$, RIGHT\$ oder MID\$ benutzen.

[130] Speichert nur die ersten drei Ziffern der Zahl in der Zeichenfolge-Variablen B\$.

[140] Die ersten 3 Stellen werden wieder in eine numerische Variable umgewandelt, so daß sie vom Programm als eine Zahl behandelt werden können.

---

# TROFF

---

P  
D

**FORMAT:** TROFF

**Abkürzung:** TROF.

**Siehe auch:** TRON

---

**WIRKUNG:**

TROFF wird benutzt, um den Trace-Modus (TRON) zu verlassen.

**ANMERKUNGEN:**

Die Ausführung von TROFF hat zur Folge, daß das Programm wieder auf normale Weise ausgeführt wird.

**BEISPIEL:**

Siehe TRON.

---

---

# TRON

---

P  
D

**FORMAT:** TRON

**Abkürzung:** TR.

**Siehe auch:** TROFF

---

**WIRKUNG:**

TRON wird benutzt, um den Trace-Modus einzuschalten.

**ANMERKUNGEN:**

Der Trace-Modus unterstützt die Fehlersuche in Programmen. Ist der Trace-Modus eingeschaltet, so wird die Zeilennummer der gerade ausgeführten Zeile ausgegeben. Zum Stoppen der Ausführung von TRACE wird die **BREAK**-Taste gedrückt oder der STOP-Befehl ausgeführt. Nach dem Abbruch des Trace-Modus wartet der Computer auf das Drücken der Taste mit dem Pfeil nach unten, bevor er auf die nächste Anweisung übergeht. Der Trace-Modus wird so lange ausgeführt, bis die Tasten **SHIFT** + **CA** gedrückt werden.

**BEISPIEL:**

```
10: TRON
20: FOR I = 1 TO 3
30: NEXT I
40: TROFF
```

Dieses Programm zeigt bei seiner Ausführung die Zeilennummern 10, 20, 30, 30, und 30 an.

---

# USING

---

P  
D

**FORMAT:** 1. USING Format-Zeichenfolge  
2. USING

**Abkürzung:** U.

**Siehe auch:** LPRINT, PRINT

---

## **WIRKUNG:**

USING wird gebraucht, um die angezeigte oder gedruckte Ausgabe zu formatieren.

## **ANMERKUNGEN:**

USING kann einzeln oder als Erweiterung einer PRINT- oder LPRINT-Anweisung eingesetzt werden. USING erstellt eine Ausgabeformatierung, die für alle folgenden Ausgaben Gültigkeit hat, bis die Formatierung durch ein neues USING geändert wird.

- #: Rechtsbündiges Zeichen eines numerischem Feldes
    - Länge des Feldes für ganze Zahlen: 2 bis 11 (einschließlich Vorzeichen)
    - Wenn der Wert kürzer ist als das spezifizierte numerische Feld, wird der zusätzliche Teil des Feldes mit Leerstellen ausgefüllt. Ein Fehler erfolgt, wenn der Wert größer als das Feld ist.
    - Wenn ein numerisches Feld mit einer Länge von 12 oder mehr Stellen spezifiziert wird, wird es als ein Feld mit 11 Stellen betrachtet.
    - Länge eines dezimalen Feldes: 0 bis 12 (0 bis 9 für den Exponenten)
    - Wenn der Wert kürzer ist als das spezifizierte Feld, erscheinen Nullen in den zusätzlichen Teilen des Feldes. Wenn der Wert länger ist, wird er abgeschnitten.
  - .: Dezimalpunkt (Begrenzer zwischen der ganzen Zahl und dem Dezimalanteil)
  - ^: Zur Anzeige dieser Zahl in wissenschaftlicher Notation.
    - Bei dieser Notation ist die Länge des Mantissenfeldes immer 2 (eine Ziffer und das Vorzeichen), unabhängig von der Länge des Feldes für die ganze Zahl.
    - Wenn die gegebene Länge des dezimalen Feldes 9 oder mehr Stellen hat, beträgt die Länge des dezimalen Feldes der Mantisse ebenfalls 9 Stellen.
  - &: Linksbündiges alphanumerisches Feld
    - Wenn eine Zeichenfolge kürzer als das spezifizierte Feld ist, erscheinen in den zusätzlichen Teilen des Feldes Leerstellen. Wenn sie länger ist, werden die zusätzlichen Zeichen abgeschnitten.
- (1) USING"###"  
Darstellen des Vorzeichens und 2 Stellen der ganzen Zahl.
  - (2) USING"###."  
Darstellen des Vorzeichens, 2 Stellen der ganzen Zahl und des Dezimalpunktes.
  - (3) USING"###.##"  
Darstellen des Vorzeichens, 2 Stellen der ganzen Zahl, des Dezimalpunktes und 2. Dezimalstellen.

- (4) USING"##.##^"  
Darstellung der numerischen Daten in Exponentenform mit bis zu 2 Dezimalstellen.  
Für die Mantisse wird automatisch Raum für das Vorzeichen und eine Stelle der ganzen Zahl reserviert; für den Exponenten werden 2 Stellen der ganzen Zahl, die Buchstaben E, und das Vorzeichen reserviert.
- (5) USING"&&&&&&"  
Darstellung einer Zeichenfolge mit 6 Buchstaben.
- (6) USING"###&&&"  
Darstellung einer Zeichenfolge neben dem numerischen Wert.
- (7) USING  
Das Format 2 löscht vorherige Formateingaben.

Formateingaben werden auch durch Ausführung des RUN-Befehls, Drücken von **SHIFT** + **CA** oder Aus- und wieder Einschalten des Gerätes gelöscht.

**BEISPIEL:**

10: B = -10:C = 10.7703  
20: PRINT USING "&&&###" ; "B=" ; B ; "\_C=" ;; PRINT USING "###.###" ; C

# VAL

P  
D

**FORMAT:** VAL Zeichenfolge

**Abkürzung:** V.  
**Siehe auch:** STR\$

**WIRKUNG:**

Wandelt eine Zeichenfolge aus numerischen Zeichen in einen Dezimalwert um.

**ANMERKUNGEN:**

Die VAL-Funktion wandelt Zeichenfolgen, die Bezeichner für Hexadezimalzahlen (&H), Ziffern (0–9), ein Vorzeichen (+,-) und exponentiale Symbole (E) in numerische Werte um.

Ist die Zeichenfolge dezimal, muß sie aus den Zeichen 0 bis 9 bestehen und kann einen Dezimalpunkt und ein Vorzeichen aufweisen. In dieser Form ist VAL das Gegenteil der STR\$-Funktion.

Werden unzulässige Zeichen eingegeben, wird die Unwandlung bis zum ersten unzulässigen Zeichen durchgeführt.

**BEISPIEL:**

A=VAL"-120" Weist der Variablen A den Wert -120 zu.  
B=VAL"3.2x4=" Weist der Variablen B den Wert 3.2 zu.  
C=VAL"&H64" Weist der Variablen C den Wert 100 zu.

---

# WAIT

---

P  
D

**FORMAT:** 1. WAIT Ausdruck  
2. WAIT

**Abkürzung:** W.

**Siehe auch:** PRINT

---

**WIRKUNG:**

Kontrolliert die Zeitdauer der Anzeige von Informationen vor der weiteren Ausführung eines Programms.

**ANMERKUNGEN:**

Format 1 spezifiziert die Zeitdauer der Unterbrechung des PRINT-Befehls. Das Programm wird entsprechend der angegebenen Zeitdauer kurzzeitig unterbrochen und dann automatisch wieder gestartet.

Der Wert des Ausdrucks kann in einem Bereich von 0 bis 65535 angegeben werden. Ein Wert von 1 entspricht in etwa einem Intervall von 1/64 Sekunde. Beim Einschalten des Computers oder beim Ausführen eines RUN-Befehls wird WAIT0 (eine Wartezeit von Null) angenommen. Der WAIT-Befehl gilt für alle PRINT-Befehle, die innerhalb des Programms verwendet werden. Um ein unbegrenzt langes Intervall einzustellen wird Format 2 verwendet. Bei Verwendung von Format 2 muß die Taste  gedrückt werden, um die Ausführung des Programms wieder aufzunehmen.

**Hinweis:**

Der WAIT-Befehl kann im allgemeinen auf normalen Personal Computern nicht verwendet werden. Bei PCs wird die Anweisung FOR...NEXT verwendet, um die Zeitdauer zu kontrollieren:

```
50: FOR J=1 TO 500:NEXT J
```

**BEISPIEL:**

```
10: WAIT 64
```

Der Rechner wartet nach PRINT etwa 1 Sekunde.

# ANHÄNGE

Datenübertragungskabel CE-T801	A
Fehlermeldungen	B
Tabelle der Zeichencodes	C
Tastenfunktionen in BASIC	D
Fehlersuche	E
Aufteilung des Speicherbereiches	F
Technische Daten	G
Verwendung von Programmen von anderen SHARP-Computern	H
Pflege des PC-E220	I

# ANHANG A

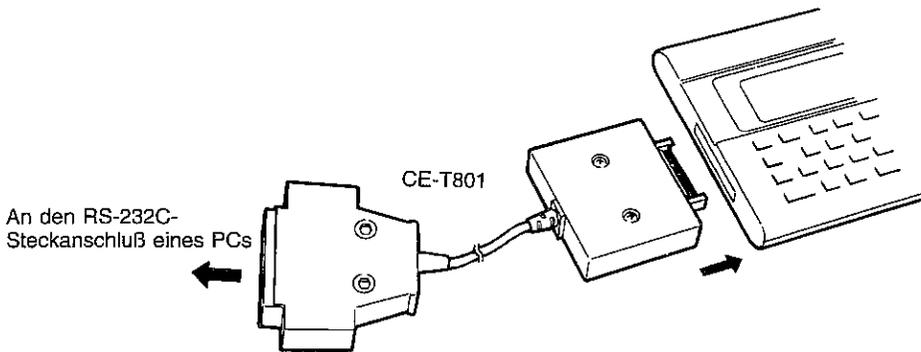
## DATENÜBERTRAGUNGSKABEL CE-T801

Das Datenübertragungskabel CE-T801 ist ein serielles Kommunikationskabel, mit dem der PC-E220 mit einem Personal Computer oder einem anderen seriellen Gerät kommunizieren kann. Mit diesem Kabel können TEXT- oder anderen Maschinensprachen-Programme unter Verwendung der SIO-Funktion der TEXT-Betriebsart oder den SIO-Befehlen des Monitors zu und von einem PC übertragen werden.

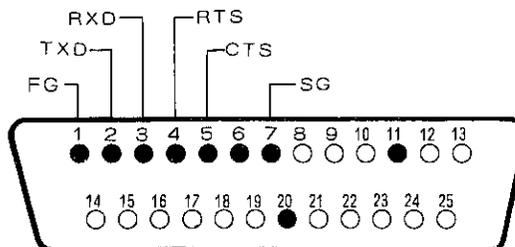
Das Datenübertragungskabel CE-T801 ist mit einem Pegelkonverter ausgestattet, der den RS-232C-Signalpegel auf einen Signalpegel für den PC-E220 konvertiert. Für die serielle Kommunikation mit dem PC-E220 wird bei einem PC normalerweise der serielle Kommunikationsport RS-232C verwendet. Da die serielle Schnittstelle des PC-E220 nicht mit dem RS-232C-Standard übereinstimmt, wird sie beschädigt, wenn die RS-232C-Signale direkt angelegt werden. Das Datenübertragungskabel CE-T801 ist mit einem pegelanpassenden Schaltkreis zum Angleichen der zwei Signalpegel ausgestattet und kann daher für die serielle Kommunikation verwendet werden.

### Vorsicht:

Niemals die Steckstifte des Datenübertragungskabel CE-T801 mit bloßen Händen berühren. Statische Entladungen des Körpers können die interne Schaltkreise beschädigen.



### Stiftbelegung des CE-T801-Steckers (DB-25(W))



Stift 6 und 20 sind verbunden.

## Signalbeschreibung

RS-232C-Signal			Signalrichtung vom PC-E220	Beschreibung
Stift-Nr.	Signalbezeichnung	Symbol		
1	Erdung	FG	—	
2	Daten senden	TXD (SD)	Eingang	Zum PC-E220 übertragene Datensignale
3	Daten empfangen	RXD (RD)	Ausgang (Hinweis 2)	Vom PC-E220 übertragene Datensignale
4	Aufforderung zum Senden	RTS (RS)	Eingang	Der PC überträgt Daten, wenn diese Leitung auf Hoch (High) eingestellt ist und stoppt die Datenübertragung, wenn sie auf Tief (Low) eingestellt ist.
5	Bereit zum Senden	CTS (CS)	Ausgang (Hinweis 2)	Auf Hoch (High) eingestellt, wenn der PC-E220 zum Empfang von Daten bereit ist; auf Tief (Low) eingestellt, wenn er nicht bereit ist.
7	Signalerdung	SG	—	Wird zur Einstellung des Referenz-Potentialpegels zwischen dem Eingangs- und Ausgangsgerät verwendet.
11	—	NC	—	Nicht belegt.

### Hinweise:

- Der CE-T801 ist so gebaut, daß er Ausgangssignale (Stifte 2 und 4) vom PC zu den Eingangsstiften des PC-E220 und Ausgangssignale vom PC-E220 zu den Eingangsstiften (Stifte 3 und 5) des PCs sendet. Das Kabel kann daher direkt an den RS-232C-Stecker eines PCs angeschlossen werden.
- Der Status der Ausgangssignale des PC-E220 ist außer in den folgenden Fällen nicht festgelegt:
  - SIO wurde in der BASIC-Betriebsart geöffnet.
  - R- oder W-Befehle werden in der Monitor-Betriebsart ausgeführt.
  - Daten werden in der TEXT-Betriebsart über SIO übertragen.

### Für Datenübertragung von einem Personal

Computer muß ein Arbeitsbereich von etwa 300 Byte zur Verfügung stehen.

## ANHANG B

### FEHLERMELDUNGEN

Wenn ein Fehler auftritt, wird einer der unten aufgeführten Codes auf der Anzeige dargestellt. Bei Fehlern, die während der Ausführung von Programmen auftreten, wird ebenfalls die Zeilennummer angezeigt, in welcher der Fehler aufgetreten ist.

Fehler-code	Bedeutung
10	Verwendung eines nicht zulässigen Ausdrucks oder einer nicht zulässigen Anweisung.
12	Es wurde versucht, einen Befehl auszuführen, der bei direkten Eingabevorgängen oder bei der Programmausführung nicht zulässig ist. Die Betriebsart für PRO oder RUN wurde falsch gewählt.
13	Die CONT-Anweisung wurde unzulässigerweise ausgeführt.
14	Es wurde versucht, ein Kennwort einem Programm zuzuweisen, welches nicht existiert.
20	Das Berechnungsergebnis übersteigt die Berechnungskapazität.
21	Es wurde versucht, durch Null zu dividieren.
22	Ein nicht zulässiger Vorgang wurde angestrebt.
30	Es wurde versucht, den Namen einer Feldvariablen zu vergeben, der bereits vergeben war.
31	Der Name einer Feldvariablen wurde ohne die DIM-Anweisung bestimmt.
32	Das Feld wurde unzulässigerweise adressiert (der Feldindex überschreitet die Größe, die dem Feld mit der DIM-Anweisung zugewiesen worden ist).
33	Der spezifizierte Wert überschreitet den zulässigen Bereich.
40	Die spezifizierte Zeilennummer oder die Benennung existieren nicht.
41	Die Zeilennummer wurde unzulässigerweise spezifiziert.
43	Unzulässige RENUM-Anweisung. (Eine nicht bestehende Zeilennummer wurde spezifiziert oder die Reihenfolge der Zeilenausführung wurde geändert.)
44	In einer Anweisung, z.B. LLIST oder DELETE, ist die Nummer der Endzeile geringer als die Nummer der Anfangszeile.
50	Das Niveau der Verschachtelung der Anweisungen GOSUB oder FOR überschreitet den zulässigen Bereich.
51	Es wurde versucht, eine RETURN-Anweisung ohne den Aufruf einer Subroutine auszuführen.
52	Die FOR-Anweisung fehlt für eine NEXT-Anweisung.
53	Die DATA-Anweisung fehlt für eine READ-Anweisung.
54	Die zulässige Anzahl für Datenpuffer (8) oder Funktionspuffer (16) wurde überschritten.

Fehler- code	Bedeutung
55	Die Länge der eingegebenen Zeichenfolge überschreitet 255 Byte. Die Zeile überschreitet 255 Byte.
60	Die Größe des Programms oder der Variablen überschreitet die Speicherkapazität.
70	Die Zeichen können nicht in dem Format ausgegeben werden, das mit der USING-Anweisung bestimmt wurde.
71	Das Format, das mit der USING-Anweisung bestimmt wurde, ist nicht zulässig.
80	Kontrollsummenfehler auf der Kassette oder Einlesefehler im SIO.
81	Abschaltfehler in der Kassettenschnittstelle oder bei SIO (die spezifizierte Wartezeit wurde während Programm- oder Daten-Ein-/Ausgabe überschritten).
82	Auftritt eines Datenfehlers während dem Vergleichen mit der Kassette oder anderen Datenträgern.
83	Die Art der spezifizierten Variablen im INPUT#-Befehl stimmt nicht mit der Art der gelesenen Daten überein.
84	Fehlfunktion des Druckers.
85	Ein Gerät wurde nicht geöffnet, wenn eine Datenübertragung zu oder vom Gerät mit dem PRINT#- oder INPUT#-Befehl versucht wurde.
86	Ein Gerät wurde zu öffnen versucht, während ein anderes Gerät bereits in Verwendung war.
87	Ein Versuch zum Lesen von weiteren Daten wurde vorgenommen, nachdem alle Daten in der Datei gelesen waren.
90	Ein Versuch der Zuweisung von Zeichen zu einer numerischen Variablen oder von Zahlen zu einer Zeichenfolge-Variable wurde vorgenommen oder eine Zeichenfolge-Variable wurde in einer Funktion spezifiziert, die nur Zahlen als Variable haben kann, z.B. SIN A\$.
91	Eine feste Variable, der Zahlen zugewiesen wurden, wurde als eine Zeichenfolge-Variable verwendet oder die Variable, der Buchstaben zugewiesen wurden, wurde als numerische Variable verwendet.
92	Kennwort stimmt nicht überein.
93	Ein Versuch zur manuellen Ausführung des MON-Befehls wurde bei Vorhandensein eines Kennwortes vorgenommen.
94	Die spezifizierte Datei wurde nicht gefunden.
95	Unzulässiger Dateiname.
96	Ein Versuch zum Lesen einer TEXT-Datei in der BASIC-Betriebsart oder einer BASIC-Datei in der TEXT-Betriebsart wurde vorgenommen.
97	Die Anzahl der Dateien überschreitet 255.

## ANHANG C

### TABELLE DER ZEICHENCODES

Diese Tabelle der Zeichencodes zeigt die Zeichen und ihre Zeichencodes, die mit den Befehlen CHR\$ und ASC verwendet werden. Jeder Zeichencode besteht aus 2 hexadezimalen Zeichen (oder 8 binären Bit). Das höherwertige Hexadezimalzeichen (4 Bit) wird oben auf der Tabelle angegeben, während das niederwertige Hexadezimalzeichen (4 Bit) auf der linken Seite der Tabelle aufgelistet ist. Wenn ein Zeichen nicht abgebildet ist, kann es von diesem Computer nicht erkannt und daher nicht eingegeben werden.

Das Zeichen "A" ist z.B. hexadezimal "41", dezimal "65" und binär "01000001". Das Zeichen "P" ist dezimal "80", hexadezimal "50" und binär "01010000".

Die Zeichencodes werden folgendermaßen dargestellt:

#### Beispiele:

Code für \*

Hexadezimal &H2A

Dezimal 42 (32 + 10)

Code für P

Hexadezimal &H50

Dezimal 80

#### Hinweis:

Beim Drucken mit dem CE-126P werden die Zeichen mit den Codes &H00-&H1F und &H7F-&HFF als Leerstellen gedruckt.

Höherwertige hexadezimale Ziffer

Niederwertige hexadezimale Ziffer

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NL		Leerstelle	0	@	P	'	p	Leerstelle	Π	Leerstelle	π	ë	∫	π	c
1			!	1	A	Q	a	q	A	P	α	ρ	°	0	ι	P
2			"	2	B	R	b	r	B	Σ	β	σ	¹	₁	χ	ē
3			#	3	C	S	c	s	Γ	T	γ	τ	²	₂	∣	ō
4			\$	4	D	T	d	t	Δ	Υ	δ	υ	³	₃	▶	◀
5			%	5	E	U	e	u	E	Φ	ε	φ	⁴	₄	≤	▶
6			&	6	F	V	f	v	Z	X	ζ	χ	⁵	₅	≥	°
7			'	7	G	W	g	w	H	Ψ	η	ψ	⁶	₆	⊕	■
8			(	8	H	X	h	x	Θ	Ω	θ	ω	⁷	₇	[	ḡ
9			)	9	I	Y	i	y	I	N	ι	ι	⁸	₈	]	■
A			*	:	J	Z	j	z	K	n	κ	κ	⁹	₉	•	^
B			+	;	K	[	k	{	Λ	H	λ	ȳ	-	Å	√	→
C			,	<	L	\	l		M	μ	μ	≠	+	ª	≠	↔
D			—	=	M	]	m	}	N	ñ	v	°C	∞	ᵇ	À	↑
E			.	>	N	^	n	~	Ξ	ð	ξ	1/2	±	°	B	↓
F			/	?	O	_	o		O	ö	o	X̄	℥	ᵐ	e	↙

Die Codes &H01-&H1F und &H7F sind Leerstellen.

## ANHANG D

### TASTENFUNKTIONEN IN BASIC

**BREAK**  
**ON**

- Zum Einschalten der Stromversorgung, wenn das Gerät durch die AUTO OFF-Funktion abgeschaltet wurde.
- Beim Drücken dieser Taste während der Ausführung von einem Programm entspricht sie der **BREAK**-Taste und führt zur Unterbrechung der Ausführung.
- Beim Drücken dieser Taste während einer direkten Eingabe wird die Ausführung von CLOAD unterbrochen.
- In der STAT-Betriebsart geht der Computer beim Drücken dieser Taste wieder auf das Untermenü zurück.
- In der TEXT- und der ASMBL-Betriebsart geht der Computer wieder auf das Hauptmenü oder das Menü zurück.

**SHIFT**  
**2nd F**

- Eine der gelben Tasten, die mit "SHIFT" bzw. "2ndF" markiert sind, muß gedrückt (und im Falle von "SHIFT" gehalten) werden, um die zweite Funktion einer Taste zu aktivieren (angezeigt direkt über der Taste).

Beispiel: **SHIFT** + **Y** → & wird eingegeben  
( **2nd F** **Y** )

**C•CE**

- Zum Löschen von Eingaben und Anzeigen auf dem Display.
- Zum Zurückstellen nach einem Fehler.

**SHIFT** + **CA**

- Zum Löschen der Anzeige auf dem Display und gleichfalls zum Zurückstellen des Computers auf den Anfangsstatus.
  - Rückstellung der WAIT-Zeiteinstellung.
  - Rückstellen des Anzeigeformats (USING-Format).
  - Rückstellen des TRON-Status (TROFF).
  - Rückstellen von PRINT=LPRINT.
  - Rückstellen der Fehlerbedingung.

**BASIC**

- Zum Umschalten des Untermodus von einer anderen Betriebsart nach RUN oder von PRO nach RUN.

**ANS**

- Zum Abrufen des letzten Ergebnisses.

**TAB**

- Zum Bewegen des Cursors auf die Tabulatorposition. In den Betriebsarten RUN und PRO wird der Cursor in Abständen von sieben Stellen bewegt. In der TEXT-EDITOR-Betriebsart wird der Cursor beim ersten Drücken um acht Stellen, beim zweiten Drücken um sechs Stellen und bei jedem weiteren Druck um sieben Stellen weiterbewegt.

**CONST**

- Zur Eingabe einer Konstanten und von Operatoren für Berechnungen mit Konstanten (die Anzeige "CONST" erscheint auf dem Display). Durch Drücken von **2nd F** **CONST** (**SHIFT** + **CONST**) wird die gegenwärtig gespeicherte Konstante angezeigt.

**SHIFT** + **<**  
**SHIFT** + **>**

- Zur Eingabe von logischen Operationen bei IF-Anweisungen.
- Zur Spezifizierung von Verhältnis-Ausdrücken.

**;**

- Für Mehrfach-Anzeige (zwei oder mehr Werte werden gleichzeitig auf dem Display angezeigt).
- Zur Trennung von Befehlen und Variablen.

**SHIFT** + **:**

- Zur Trennung von mehr als einer Anweisung in einer einzigen Zeile.

**,**

- Für Mehrfach-Anzeige (zwei oder mehr Werte werden gleichzeitig auf dem Display angezeigt).

**▶**

- Bewegt den Cursor nach rechts.
- Führt Wiedergabe-Anweisungen aus.
- Zur Darstellung des Cursors, wenn er nicht angezeigt wird, während ein Inhalt angezeigt wird.
- Löschen eines Fehlers bei der direkten Eingabe.

**◀**

- Bewegt den Cursor nach links.
- Ansonsten gleich der Taste **▶**.

**INS**

- Zum Einfügen einer Leerstelle an der gegenwärtigen Cursorposition.

**DEL**

- Löscht das Zeichen, auf dem sich der Cursor gerade befindet.

**BS**

- Löscht das Zeichen direkt links vom Cursor.

**←**

- Eingabe einer Programmzeile in den Computer.
- Beim Schreiben von Programmen.
- Fragt nach manueller Berechnung oder direkter Ausführung einer Befehlsanweisung.
- Zum Wiederaufnehmen eines Programms, welches mit dem INPUT-Befehl kurzfristig unterbrochen wurde.

**SHIFT** + **P-HP**

- Zur Einstellung der Betriebsart für Drucken oder kein Drucken, wenn ein zusätzlicher Drucker angeschlossen ist.

**CAPS**

- Zum Ein- und Ausschalten der Betriebsart für Großbuchstaben. Das Symbol CAPS erscheint auf der Anzeige.
- Wenn das CAPS-Symbol angezeigt wird, werden alle Buchstaben als Großbuchstaben eingegeben. Wenn **CAPS** gedrückt wird, verschwindet das CAPS-Symbol wieder und Kleinbuchstaben werden eingegeben.

**SHIFT** + **&**

- Zur Markierung eines hexadezimalen Wertes.

Die Tasten **↓** und **↑** haben die entsprechenden Funktionen, abhängig von der Betriebsart und dem Status des Computers, die in der folgenden Tabelle aufgelistet werden.

Betriebsart	Status	↓	↑
RUN	Programm wird ausgeführt.	Keine Funktion.	
	Unterbrochen durch den STOP-Befehl oder die <b>BREAK</b> - Taste.	Ausführung der folgenden Zeile und Unterbrechen.	Gedrückt halten, um die ausführende bzw. bereits ausgeführte Programmzeile zur Anzeige zu bringen.
	Fehlerbedingung während der Ausführung eines Programms.	Keine Funktion.	Gedrückt halten, um eine fehlerhafte Zeile anzuzeigen.
	Trace-Modus ON.	Ausführen eines Programms im Trace-Modus.	Gedrückt halten, um die auszuführende bzw. ausgeführte Programmzeile zur Anzeige zu bringen.
PRO	(Bei Änderung in die PRO-Betriebsart und fehlender Darstellung der Programmzeilen.)		
	Das Programm wird zeitweise unterbrochen.	Anzeige der unterbrochenen Zeile.	Wie links.
	Fehlerbedingung	Anzeige der fehlerhaften Zeile.	Wie links.
	Andere Bedingung.	Anzeige der ersten Zeile.	Anzeige der letzten Zeile.
	(Wenn Programmzeilen angezeigt werden.)		
	Anzeige der folgenden Programmzeile.	Anzeige der vorherigen Programmzeile.	

**Hinweis:**

Wenn während der Betriebsart für Tasteneingaben für etwa 11 Minuten keine Taste gedrückt wird, schaltet das Gerät automatisch die Stromversorgung ab (AUTO OFF-Funktion).

# ANHANG E

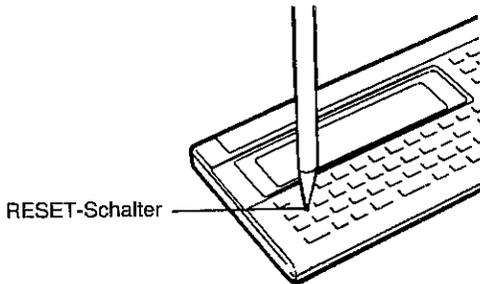
## FEHLERSUCHE

In diesem Anhang finden sich Hinweise bei Problemen, wenn der Computer die eingegebenen Befehle nicht ausführt. Sie sollten einen nach dem anderen die folgenden Vorschläge ausprobieren, bis Sie das Problem beseitigt haben.

1. Wenn das Display zu hell oder zu dunkel ist:
  - Einstellen des Kontrastreglers.
2. Wenn die Stromversorgung nicht funktioniert (es wird nichts angezeigt):
  - Die Batterien sind zu schwach. Die Batterien auswechseln.
  - Der Speichersicherungs-Schalter befindet sich in Position B. Sicherstellen, daß dieser Schalter sich in der Position A befindet.
3. Wenn das Gerät nicht abgeschaltet werden kann:
  - Der Computer führt gerade ein Programm aus, welches einige Zeit beansprucht, z.B. CSAVE oder CLOAD. Zur Unterbrechung des Programms die **BREAK** -Taste und dann die **OFF** -Taste drücken. Wenn sich das Gerät immer noch nicht abschalten läßt, siehe Punkt 4.
4. Wenn der Computer nicht vorschriftsmäßig funktioniert, weil:
  - bei eingeschaltetem Gerät möglicherweise ein Peripheriegerät angeschlossen oder abgetrennt wurde, oder wenn bei der Ausführung eines Programms möglicherweise ein Fehler auftrat oder der Computer vielleicht einem starken elektrischen Feld oder Schock ausgesetzt war.In derartigen Fällen folgendermaßen vorgehen:

### Zurückstellen

Den RESET-Schalter mit einem Kugelschreiber oder ähnlichem Instrument eindrücken.



Zum Eindrücken des RESET-Schalters einen Kugelschreiber oder ein ähnliches Instrument verwenden. Keinen Druckbleistift mit herausragender Miene oder ein besonders spitzes Instrument, z.B. eine Nähnadel verwenden.

Nach dem Loslassen des RESET-Schalters erscheint die folgende Meldung. Falls eine andere Meldung erscheint, muß der RESET-Schalter erneut eingedrückt werden. (Der Computer fragt zur Sicherheit, ob der gesamte Speicherinhalt gelöscht werden kann.)

MEMORY CLEAR O.K.? (Y/N)

- Wenn das Programm bzw. die gespeicherten Daten doch nicht gelöscht werden sollen, wird die Taste  N  gedrückt (oder eine beliebige Taste außer  Y ). Die Anzeige geht wieder zurück auf den ersten Bildschirm der RUN-Betriebsart.
- Wenn bei der Ausführung des Programms wieder ein Fehler auftritt, gehen Sie folgendermaßen vor:
- Der gesamte Speicherinhalt wird gelöscht.

Beim Erscheinen der obigen Meldung wird die Taste  Y  gedrückt. Der gesamte Speicherinhalt wird gelöscht und die folgende Meldung blinkt auf dem Display (dies zeigt an, daß der Computer initialisiert und der gesamte Speicherinhalt gelöscht wurde):

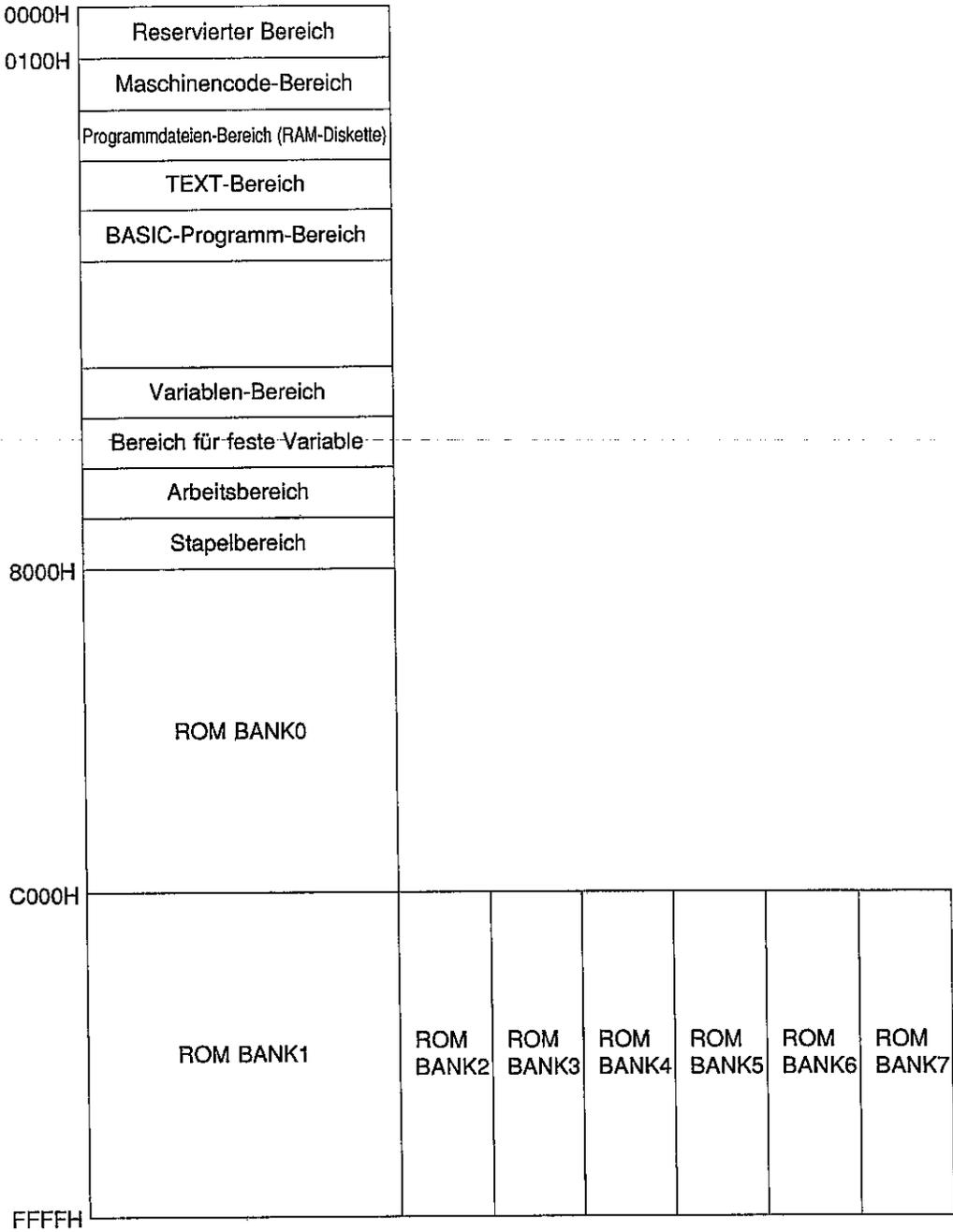


Eine beliebige Taste drücken, um auf den ersten Bildschirm der RUN-Betriebsart zurückzukehren.

# ANHANG F

## AUFTEILUNG DES SPEICHERBEREICHES

Speicherbereich



# ANHANG G

## TECHNISCHE DATEN

Prozessor:	8-Bit CMOS CPU (entspricht Z80A)	
Speicherkapazität:	Systemintern	etwa 2,1 Kilobyte
	Bereich der festen Variablen	208 Byte
	Bereich für Programme/Daten	30435 Byte
Stapelspeicher:	Subroutinenstapel:	10 Puffer
	Funktionsstapel:	16 Puffer
	FOR-NEXT-Stapel:	5 Puffer
	Datenstapel:	8 Puffer
Operatoren:	Addition, Subtraktion, Multiplikation, Division, trigonometrische und invers trigonometrische Funktionen, logarithmische und Exponentialfunktionen, Umrechnung von Winkeln, Quadrat und Quadratwurzel, Potenz, Vorzeichen, absolute Zahlen, ganze Zahlen, Koordinaten-Umwandlung, Pi u.a.	
Numerische Präzision:	10 Stellen (Mantisse) + 2 Stellen (Exponent)	
Editierfunktionen:	Cursor nach rechts und links, Zeile nach oben und unten, Einfügen von Zeichen, Löschen von Zeichen TEXT-Editor, Monitor für die Maschinensprache Z80	
Speichersicherung:	Batterie-Backup	
Interface-Möglichkeiten:	11 Stifte (für Kassetten-Interface, Diskettenlaufwerk, Drucker, SIO-Gerät)	
Display:	4 Zeilen, 24 Zeichen, Flüssigkristall-Anzeige mit 5 × 7 Punktmatrix	
Stromversorgung:	Für den Computerbetrieb: Vier 6,0 V Gleichstrom Typ AA-Trockenbatterien (R06)	
	Für die Speichersicherung: Eine 3,0 V Gleichstrom Lithiumbatterie (CR2032)	
Stromverbrauch:	0,37 W	
	Etwa 80 Stunden kontinuierlicher Betrieb unter normalen Bedingungen (bezogen auf 10 Minuten Betrieb oder Programmausführung und 50 Minuten Anzeige pro Stunde bei Temperaturen von 20°C). Hinweis: Wenn der Computer für die serielle Kommunikation über das Datenübertragungskabel CE-T801 verbunden ist, verringert sich die Zeitdauer, in der der Computer kontinuierlich betrieben werden kann, auf etwa 48 Stunden (bei 2 Minuten Kommunikation, 8 Minuten Berechnungen oder Programmausführungen und 50 Minuten Anzeige pro Stunde bei einer Umgebungstemperatur von 20°C).	

Die Betriebszeit kann unterschiedlich sein, je nach dem Gebrauch und der Art der verwendeten Batterie.

Betriebstemperatur:	0°C – 40°C
Abmessungen:	215 (B) × 100 (T) × 18 (H) mm
Gewicht:	280 g (mit Batterien)
Zubehör:	Tastaturabdeckung, vier Trockenbatterien, eine Lithiumbatterie und eine Bedienungsanleitung
Zusatzgeräte:	Drucker/Kassetten-Interface (CE-126P) Datenübertragungskabel (CE-T801)

## ANHANG H

### VERWENDUNG VON PROGRAMMEN VON ANDEREN SHARP-COMPUTERN

Auf dem Computer PC-E220 können mit geringfügigen Modifikationen die folgenden Programme der folgenden Computer der SHARP PC-Serie verwendet werden:

PC-1403(H), PC-1460, PC-1425

Der WAIT-Befehl des PC-E220 ist auf den Wert von 0 voreingestellt. Dies bedeutet, daß die Anzeige bei Ausführung eines PRINT-Befehls nicht kurzzeitig anhält. Wenn mehr als vier Zeilen gleichzeitig dargestellt werden sollen, muß ein WAIT-Befehl in ein übertragenes Programm eingefügt werden, um die Anzeige kurzfristig anzuhalten (siehe WAIT-Befehl). Da der PC-E220 24 Anzeige-Zeichen hat, müssen bei übertragenen Programmen möglicherweise Modifizierungen auf die Breite des PC-E220-Displays durchgeführt werden. Programme, welche Befehle oder Zeichen enthalten, die der PC-E220 nicht kennt, müssen ebenfalls modifiziert werden.

- **Speicherkapazität**

Programme für andere Computer benötigen eine andere Speicherkapazität als der PC-E220.

- **Vom Benutzer definierte Tasten**

Der PC-E220 besitzt keine vom Benutzer definierte Tasten für die Ausführung von Programmen. Anstelle der vom Benutzer definierten Tasten kann

GOTO\*Label [←] verwendet werden. Bei einer GOTO-Anweisung sollte \*Label und nicht "Label" verwendet werden.

- **Zeilennummern**

Variable oder Ausdrücke können nicht für die Zeilennummern verwendet werden, die mit den Anweisungen für GOTO, GOSUB, RESTORE oder THEN spezifiziert wurden.

- **Kassettengerät**

Zum Laden von Programmen anderer Computer in den PC-E220 kann der folgende Befehl verwendet werden:

CLOAD@"Dateiname" [←] oder CLOAD@ [←]

Wenn kein Dateiname eingegeben wird, wird das erste Programm nach Bandbeginn geladen. Der PC-E220 übersetzt automatisch geladene Programm-Codes in PC-E220-Programm-Codes, während er die einzelnen Zeilen liest. Wenn eine übersetzte Programmzeile mehr als 255 Byte beträgt, entsteht ein Fehler. Wenn ein Programm geladen wird, das den Programm-Speicherbereich des PC-E220 übersteigt, entsteht ebenfalls ein Fehler. In diesem Fall müssen die nicht notwendigen Variablen aus dem Programm-Speicherbereich gelöscht und das gleiche Programm noch einmal geladen werden. Bei der Übersetzung von Codes wird ein Arbeitsbereich von etwa 300 Byte zusätzlich zum Programm-Speicherbereich verwendet.

Bei der Eingabe/Ausgabe von Programmen von oder einem Kassettengerät muß der OPEN-Befehl ausgeführt werden.

**Beispiel:**

```
20 PRINT #F, G → 10 OPEN "CAS:" FOR OUTPUT
                   20 PRINT #1, F, G
                   30 CLOSE #1
```

- **Feldvariable**

Wenn in einem anderen Computerprogramm Feld A( ) ohne eine DIM-Anweisung bestimmt wurde, muß die Feldvariable am Beginn des Programms bestimmt werden. Der Bereich für feste Variable wird nicht mit der Feldvariablen A( ) geteilt.

- **Verschiedenes**

1. Bei logischen Operationen gibt der PC-E220 den Wert -1 für wahr und 0 für falsch.
2. In einer DATA-Anweisung dürfen keine Ausdrücke verwendet werden, z.B. DATA SIN 30+2, CHR\$66. Diese Anweisungen müssen z.B. folgendermaßen gegeben werden: DATA 2.5, B.

## **ANHANG I**

### **PFLEGE DES PC-E220**

Um einen störungsfreien Betrieb dieses Computer zu sichern, sollten die folgenden Hinweise beachtet werden:

- Den Computer immer vorsichtig behandeln, da die Flüssigkristall-Anzeige aus Glas besteht.
- Den Computer nicht extremen Temperaturschwankungen, Feuchtigkeit oder Staub aussetzen. Im Sommer steigt die Temperatur in Fahrzeugen bei direkter Sonneneinstrahlung extrem hoch an. Längeres Liegen bei hohen Temperaturen kann zu Beschädigungen dieses Computers führen.
- Zum Reinigen ein weiches, trockenes Tuch verwenden. Keine Lösungsmittel, Wasser oder feuchte Tücher benutzen.
- Um Auslaufen der Batterien zu vermeiden, sollten die Batterien herausgenommen werden, wenn der Computer für längere Zeit nicht gebraucht wird.
- Wenn der Computer starker statischer Elektrizität oder externen Störungen ausgesetzt wird, kann er plötzlich überhaupt nicht mehr funktionieren (keine Reaktion mehr auf Tastendruck). In diesem Fall den RESET-Schalter eindrücken. (Siehe FEHLERSUCHE.)
- Diese Bedienungsanleitung für spätere Referenz sorgfältig aufbewahren.

## REGISTER DER KOMMANDOS

ASC .....	152	REM ( ' ) .....	200
BEEP .....	152	RENUM .....	201
CHR\$ .....	153	RESTORE .....	202
CLEAR .....	154	RIGHT\$ .....	203
CLOAD .....	155	RND .....	204
CLOAD? .....	156	RUN .....	205
CLOSE .....	157	SAVE .....	206
CLS .....	158	STOP .....	207
CONT .....	158	STR\$ .....	208
CSAVE .....	159	TROFF .....	209
DATA .....	160	TRON .....	209
DEGREE .....	161	USING .....	210
DELETE .....	162	VAL .....	211
DIM .....	163	WAIT .....	212
END .....	165		
FILES .....	166		
FOR...NEXT .....	167		
FRE .....	168		
GOSUB...RETURN .....	169		
GOTO .....	170		
GRAD .....	171		
IF...THEN .....	172		
INKEY\$ .....	174		
INPUT .....	175		
INPUT# .....	177		
KILL .....	178		
LEFT\$ .....	178		
LEN .....	179		
LET .....	180		
LFILES .....	180		
LIST .....	181		
LLIST .....	182		
LOAD .....	183		
LOCATE .....	184		
LPRINT .....	185		
MID\$ .....	186		
MON .....	186		
NEW .....	187		
ON...GOSUB .....	188		
ON...GOTO .....	189		
OPEN .....	190		
PASS .....	191		
PRINT .....	192		
PRINT# .....	195		
RADIAN .....	197		
RANDOMIZE .....	198		
READ .....	199		

# REGISTER

## A

- Altgrad 25, 66
- Anweisungen 84, 125
- ASCII-Code 98, 109
- ASMBL-Modus (Assembler) 9, 123
- Assembler 120
  - Assemblieren 128
  - Bereich zuweisen 122
  - Fehler 136
  - Protokoll 131
  - Pseudobefehle 132
  - Quellprogramm, Codieren und Editieren 124
- Ausdrücke 80
  - Logische 82
  - Verhältnis 81
  - Zeichenfolge 81
- Automatische Abschaltung (AUTO OFF) 8

## B

- BASIC
  - Anweisungen 84
  - Befehle 85
  - Begriffe und Ausdrücke 74
  - Betriebsarten (Modus) 9
  - Computeranweisungen 139
  - KOMMANDO LEXIKON 151
  - Konverter 9
  - Programmieren 73
- BATT-Anzeige 11, 12
- Batterien
  - Auswechseln 12
  - Betriebsbatterie 7, 12
  - Handhabung 15
  - Speichersicherung 7, 14
- Befehle
  - BASIC KOMMANDO LEXIKON 151
  - Maschinensprache 112
- Berechnungen
  - Fehler 71
  - Länge 66
  - Rechenkapazität 149
  - Serielle 63
  - Statistische 50
  - Wissenschaftliche 27, 66, 140

- Betriebsarten, Modus 9
  - Wahl 9
- Bogenmaß 25, 66
- BUSY-Anzeige 10

## C

- CAL-Modus 9, 22
- CAPS-Anzeige 10
- CONST-Anzeige 11
- Cursor 10

## D

- DEG-Anzeige 10
- Dateigruppenzeichen 102
- Dateinamen 80
- Datenübertragungskabel 16, 214
- Dezimalstellen 24, 31
- Direkte Befehle 86, 151
- Direkte Berechnungen 69
- Direkte Eingabe 21
- Display, Anzeige 6, 10
- Display-Modus 24
- Drucken, Direkteingabe 71
- Drucker/Kassette-Interface 16, 19

## E

- E-Anzeige (Fehler) 11
- E/A, serielle 98, 104
- E/A; I/O
  - Seriell 16, 104
  - Tabelle der Befehle 16
- Eingabeaufforderung 10, 114
- Einschalten (ON) 8
- Einzelvariablen bei Statistiken 50
- Erweiterung von Dateinamen 80

## F

- Fakultät 34
- Formeln 36
- Fehler 62, 71, 136
- Fehlermeldungen 119, 216
- Fehlersuche 94, 223
- Feldvariable 77
- Feste Variable 76

Formeln und Gleichungen  
Fakultät 36  
Integralrechnung 40  
Trigonometrische 38  
Freier Speicherbereich (FRE) 3

**G**  
GRAD-Anzeige 10  
Grischiches Alphabet 42  
Grundlegende Bedienung 26

**H**  
Hardware 6, 8  
Hexadezimalzahlen 74

**I**  
Initialisierung des Computers 2  
Integralrechnung 34  
Formeln 40

**K**  
Kassetten  
Aufnahme 17  
Hinweise 19  
Kassettenerkorder  
Interface 16, 17  
Technische Daten 17  
Klammern 30, 33, 83  
Kommunikationsparameter 105  
Komplexe Zahlen 48  
Konstanten  
Berechnungen 63  
Physikalische 43  
Zeichenfolge 74  
Kontrast 12

**L**  
Labels 85, 126  
LCD-Flüssigkristall-Anzeige 6, 10  
Kontrastregler 6, 12  
Letztes Ergebnis, Abruf 65  
Logische Ausdrücke 82

**M**  
M-Anzeige (Speicher) 11  
Metrische Umwandlung 45  
Modifizierungsfunktion 32  
Monitor für Maschinensprache 112  
Befehle 113

Fehlermeldungen 119  
Monitor-Modus 112

**N**  
Namen, Dateinamen 80  
Neugrad 25, 66  
Numerische Operatoren 26

**O**  
Objektprogramm 120, 124  
Operatoren  
Logische 82  
Numerische 26, 81  
Verhältnis 69, 81  
Vorrangordnung 32, 70, 83

**P**  
PRINT-Anzeige 11  
PRO-Modus 9, 73, 86  
Anzeige 11  
Peripheriegeräte 16  
Pflege des Computers 230  
Physikalische Konstanten 43  
Programme  
Auflistung 87, 90  
Ausführung 93  
Dateien 80  
Eingabe 86  
Speicherung 92  
Von anderen Computern 228  
Programmdateien 80  
Programmieren 84, 86  
Pseudobefehle 132

**Q**  
Quellprogramm 120, 125

**R**  
RAD-Anzeige 10  
RAM-Diskette i, 80, 225  
Rückstellung  
RESET-Schalter 7  
Vorgehen 2, 223  
RUN-Modus 9, 11, 58  
Anzeige 11  
Berechnungen 58

## S

- SHIFT-Taste 9
- Speicher
  - Aufteilung des Speicherbereichs 225
  - Berechnungen mit 27
  - Verfügbarer 3
- Speichern, Programme 92
- STAT-Modus 11, 50
  - Anzeige 11
- Statistiken
  - Ausdruck 54
  - Berechnungen 50, 54
  - Doppel-Variable 54
  - Eingabe von Daten 51, 55
  - Einzel-Variable 50
  - Löschen von Daten 51
- Statuszeile 10
- Stecker, 11-Stift 6, 16

## T

- TAB-Taste 100
- Tastatur 6
- Tastaturabdeckung ii
- Tastenfunktionen 220
- Technikersoftware (ENG) 9, 34
  - Liste 35
- Technische Daten 226
- TEXT-Modus 98
  - Anzeige 11
  - Editor 99
  - Funktionen 98
  - Kommunikationsparameter 105
- TEXT/BASIC-Umwandlung 109
- Trace-Modus 94
- Trigonometrische Berechnungen 27, 67
  - Formeln 38

## U

- Umwandlungen
  - Metrische 45
  - Polar/rechtwinkliger Koordinaten 30, 68
  - TEXT/BASIC 109
  - Winkel/Zeit 29, 68

## V

- Variable 74
  - Arten 76
  - Einfache 77

Feste 76

Numerische 76

Verwendung bei Berechnungen 64

Verwendung bei Programmen 89

Zeichenfolge 76

Verhältnis-Operatoren 69, 81

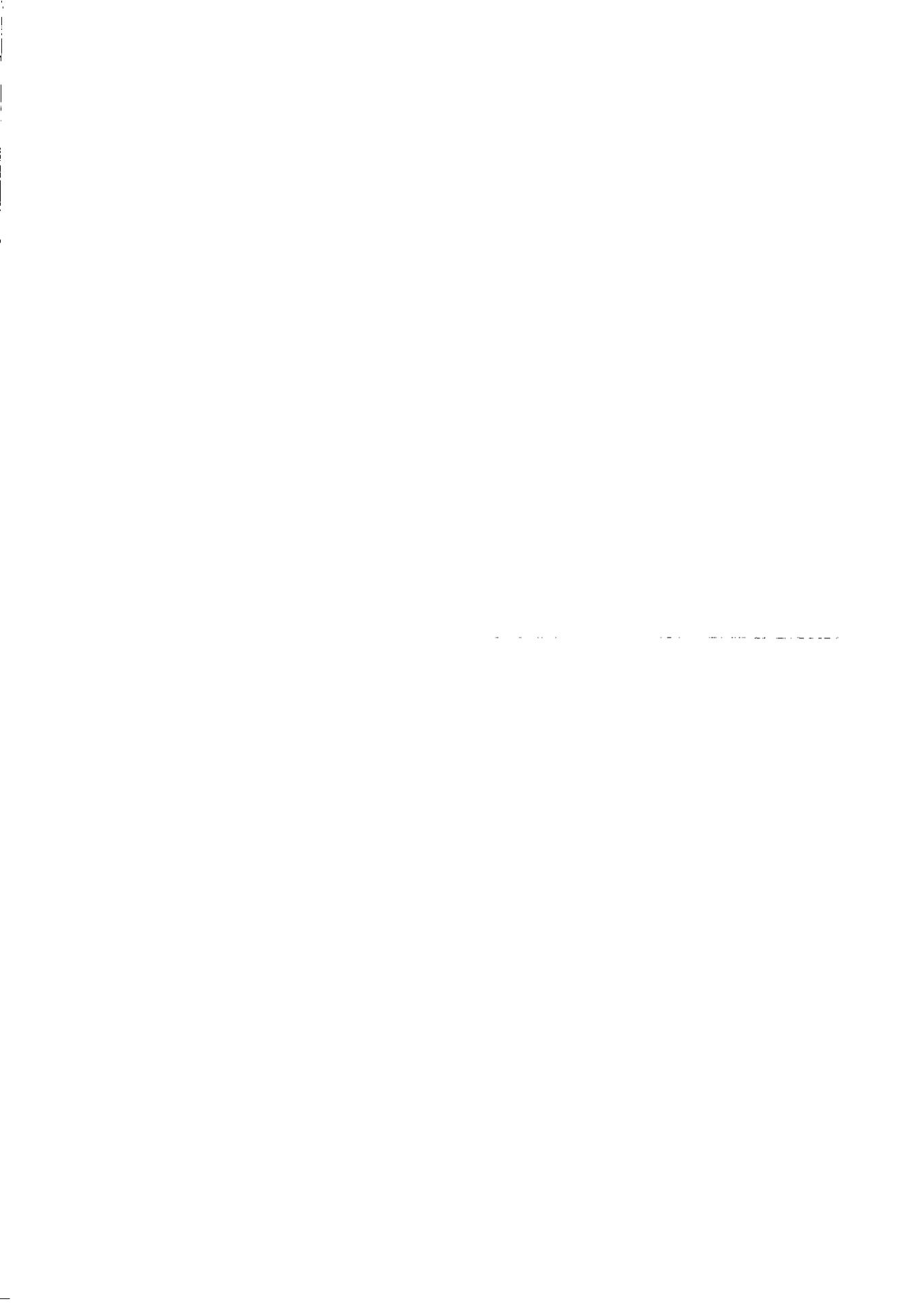
Vorrangordnung 32, 70, 83

## W

- Wissenschaftliche
  - Berechnungen 27, 66, 140
  - Notationen (SCI) 23

## Z

- Zeichencode 219
- Zeilennummern 84
- Zweite Funktion (2ndF)
  - Anzeige 10
  - Taste 9



# **SHARP CORPORATION**

OSAKA, JAPAN

© 1991 SHARP CORPORATION

PRINTED IN JAPAN/IMPRIMÉ AU JAPON

1A6KS(TINSG1208ECZZ)1