

SHARP

TASCHENCOMPUTER

MODELL **PC-1500**

BEDIENUNGSANLEITUNG



**WWW.
PC-1500
.INFO**

Do not sale this PDF !!!



INHALTSVERZEICHNIS

	Seite
I. Einleitung	5
II. Betriebshinweise	6
III. Stromversorgung	7
1. Einschalten des Computers	8
2. Ausschalten des Computers	8
3. Die Bedienungselemente	9
4. Die Tastenfunktionen	10
5. Die Anzeige	12
6. Wahl der Betriebsart	12
7. Speicherorganisation	13
7.1 Standardvariablen-Speicher	14
7.2 Hauptspeicher	14
7.3 BASIC-Stack	14
7.4 RESERVE-Speicher	14
8. Rechnen ohne Programmunterstützung	15
8.1 Rechengenauigkeit	16
8.2 Kontrollieren und Verändern der Eingabe (Editieren)	16
8.3 Speichern von Zwischenergebnissen	17
9. Mathematische Funktionen	18
9.1 Klammerregeln	18
9.2 Hexadezimalzahlen	19
10. Textausdrücke	20
11. Vergleichsausdrücke	21
12. BASIC-Befehle	21
13. Numerische Funktionen	22
13.1 Trigonometrische-/Umkehrfunktion	22
DMS, DEG	23
13.2 Mathematische Funktionen	24
LN, LOG	24
EXP	24
ABS	24
INT	25
SGN	25
SQR oder $\sqrt{\quad}$	25
π/PI	25
13.3 Zufallszahlen	26
RND	26
RANDOM	26
13.4 NOT-Funktion	27
AND-Operator	28
OR-Operator	29
13.5 TIME	30
13.6 STATUS	31
14. Programmieren in BASIC	32
14.1 BASIC-Übersicht	32

	Seite
14.2 Programmaufbau	34
14.3 Überprüfen des Programms und Auflisten	35
14.4 Korrektur einer Zeile	35
14.5 Löschen einer Zeile	35
14.6 Programmausführung	36
14.7 Fehlermeldung/Fehlersuche	36
15. PC-1500 BASIC	37
16. Sprachelemente	38
16.1 Numerische Konstante	38
16.2 Numerische Variable	38
16.3 Textkonstante	38
16.4 Textvariablen	39
16.5 Numerische oder Textvariablen	39
16.6 Numerische Funktion/Textfunktion	39
16.7 Numerischer Ausdruck	40
16.8 Textausdruck	40
16.9 Logischer Ausdruck	41
16.10 Programmzeile	42
BASIC-Instruktionen	42
LET	42
DATA	43
READ	44
RESTORE	46
REM	47
STOP	48
END	49
GOTO	49
GOSUB	50
RETURN	52
ON . . . GOTO	52
ON ERROR GOTO	54
ON . . . GOSUB	55
IF . . . THEN	56
FOR . . . TO . . . STEP/NEXT	58
DIM	60
RUN, GOTO	63
Definable Keys	64
AREAD	64
ARUN	65
CONT	65
TRON	66
TROFF	66
CLEAR	66
NEW/NEW Ø	67
LIST	67
Ein- / Ausgabeanweisungen	68
USING	68
PRINT	73
PAUSE	76
WAIT	77
INPUT	77
CURSOR	80
GCURSOR	81

	Seite
GPRINT	82
POINT	86
CLS	88
BEEP	88
Textfunktionen	89
ASC	89
CHRS	90
LEN	91
STR\$	91
VAL	92
LEFT\$	93
RIGHT\$	93
MIDS	95
INKEY\$	95
17. Der RESERVE-Mode	97
17.1 Beschriftung der reservierbaren Tasten	98
17.2 Fest eingebaute Abkürzungen	98
18. CE-150 (Option)	99
18.1 Einleitung	99
18.2 Betriebshinweise	99
18.3 Anschluß des CE-150 an den PC-1500	99
18.4 Betriebsvorbereitung	101
18.4.1 Einschalten des CE-150	101
18.4.2 Auswechseln der Papierrolle	101
18.4.3 Auswechseln der Farbpatronen	102
18.4.4 Druckwerktest	104
18.4.5 Farbpatronen und ihre Handhabung	104
18.4.6 Stromversorgung	104
18.4.7 Anschluß des Netz-Ladegeräts EA-150	104
18.5 Druckerausgaben	105
TEXT/GRAPH	106
LF	107
CSIZE	108
TEST	109
LLIST	109
COLOR	110
TAB	110
LPRINT	111
ROTATE	114
GLCURSOR	116
SORGN	117
LINE	118
RLINE	121
19. Datenspeicherung auf Magnetband	124
19.1 Einleitung	124
19.2 Auswahlkriterien für einen Kassettenrekorder	124
19.3 Betriebshinweise	124
19.4 Anschluß der Rekorder an den CE-150	125
19.5 Datenspeicherung auf dem Magnetband	126
19.6 Benutzung des Magnetbandes	128
19.7 Überprüfen der Abspeicherung und Zurückgewinnung von Information	128
19.8 Vorbereitung für die CLOAD? — Anweisung	129

	Seite
CSAVE	130
CLOAD	132
CLOAD?	133
MERGE	134
CHAIN	136
PRINT#	139
INPUT#	141
RMT	141
20. CE-151/CE-155 (Option)	142
ANHANG	143
A. ASCII-Code Tabelle	144
B. Rechenbereich	146
C. Fehlermeldungen	147
D. Liste der Funktionen und Instruktionen	153
E. Referenzliste BASIC	160
F. Technische Daten PC-1500	161
G. Technische Daten CE-150 (Option)	162



I. Einleitung

Mit dem BASIC-programmierbaren Taschencomputer PC-1500 hat SHARP jetzt die Lücke zwischen seinem PC-1211 und dem Personal Computer Sektor geschlossen. Erweitertes BASIC, 16 kByte ROM Betriebssystem und 2,6 kByte RAM Anwenderbereich in der Grundversion und Optionen wie z.B. ein vierfarbiger Plotter/Drucker mit integriertem Kassetteninterface und RAM Steckmodule von 4 kB oder 8 kB lassen diesen Rechner auch gehobenen Ansprüchen aus fast allen Bereichen gerecht werden.

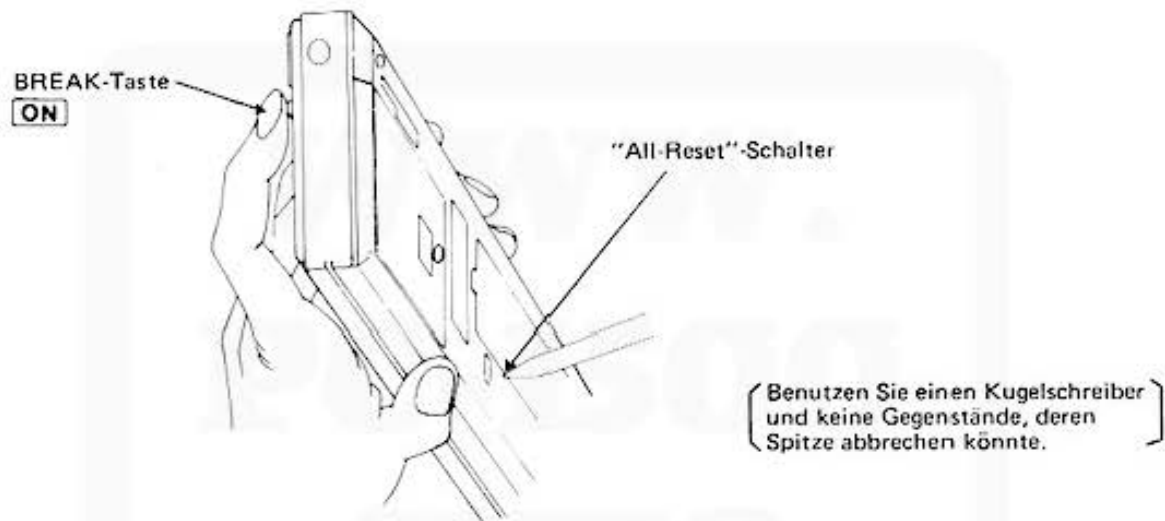
Bei der Gestaltung dieser Bedienungsanleitung haben wir uns bemüht, sowohl dem BASIC-Anfänger als auch dem fortgeschrittenen Programmierer gerecht zu werden. Dabei mußten naturgemäß Kompromisse eingegangen werden. Deshalb verweisen wir bereits an dieser Stelle auf eine umfangreiche BASIC-Literatur. Eine Referenzliste finden Sie im Anhang.

Wir hoffen, Ihnen mit diesen Materialien alle möglichen Hilfsmittel in die Hand zu geben, den Computer für Ihre spezifischen Belange optimal einzusetzen.



II. Betriebshinweise

- 1) Die Flüssigkristallanzeige des PC-1500 ist in Spezialglas eingebettet und kann daher bei Gewalteinwirkung zerbrechen. Behandeln Sie den Computer deshalb mit Sorgfalt. Beim Transport immer die Schutzhülle verwenden.
- 2) Schützen Sie den Computer vor Staub, Feuchtigkeit und allzu großen Temperaturschwankungen.
- 3) Zur Reinigung dient ein weiches trockenes Tuch. Keine Reinigungs- oder Lösungsmittel verwenden.
- 4) Durch elektrostatische Entladungen über den Computer oder durch Fehlbedienung (der Computer blieb beim Batteriewechsel oder Anschluß der Optionen CE-150, CE-151, CE-155 eingeschaltet etc.), kann der PC-1500 "abstürzen". Dadurch werden alle Tastenfunktionen blockiert, einschließlich der **ON**-Taste.
Sollte Ihnen dies passieren, betätigen Sie den "All-Reset"-Schalter für ca. 15 Sekunden. Gleichzeitig drücken Sie die **ON**-Taste.



Auf der Anzeige erscheint:



Durch Eingabe von **CL** NEW 0 **ENTER** wird der Computer initialisiert. Der Reservespeicher muß im Reserve-Mode durch NEW **ENTER** gelöscht werden.

Bleibt der PC-1500 nach vorgenannter Prozedur blockiert, müssen die Batterien eine Zeitlang ausgebaut werden. Nach dem Wiedereinsetzen der Batterien wird die oben genannte Prozedur nochmals wiederholt.

III. Stromversorgung

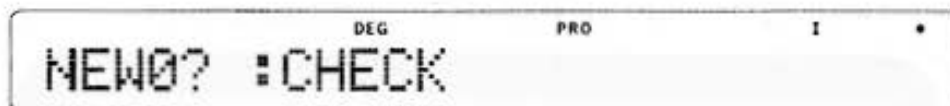
Der Taschencomputer PC-1500 wird mit Trockenbatterien bzw. mit dem Netz-Ladegerät des als Option erhältlichen Plotter/Drucker CE-150 betrieben.

Batteriewechsel

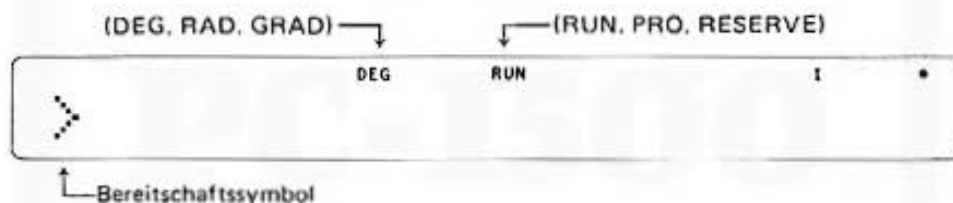
Vor Inbetriebnahme des PC-1500 bzw. bei erschöpften Batterien muß ein Batteriewechsel wie folgt vorgenommen werden:

- 1) Schalten Sie den Computer aus.
- 2) Lösen Sie die Schraube der Batterieabdeckung (Schraubendreher oder Münze).
- 3) Wechseln Sie die Batterien (Achten Sie auf richtige Polung (+), (-)).
- 4) Sie montieren die Batterieabdeckung und sichern sie mit der Schraube.
- 5) Schalten Sie den Computer mit der **[ON]** -Taste ein.

In der Anzeige steht entweder



oder



Bei der Anzeige 1) betätigen Sie die **[CL]** -Taste und geben ein NEW 0 **[ENTER]** , danach **[SHIFT] [MODE] NEW [ENTER] [MODE]** . Durch diese Maßnahme ist der Computer initialisiert.

Bei der Anzeige 2) ist der Computer im allgemeinen sofort betriebsbereit (Nur bei abnormalem Betriebsverhalten ist auch hier die Eingabe von **[CL] NEW 0 [ENTER]** notwendig).

Stehen andere oder keine Informationen in der Anzeige, wiederholen Sie die Schritte 1) bis 5).

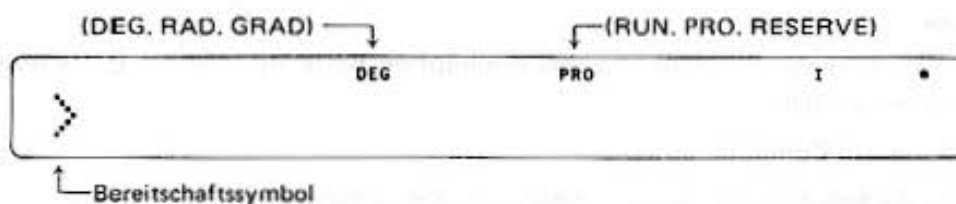
Wichtige Hinweise:

- 1) Wechseln Sie alle Batterien gleichzeitig. Geeignet sind z.B. Philips LR6A, Varta 4106, Mallory MN 1500, UCAR 410 etc.
- 2) Wird der Computer längere Zeit nicht benutzt, sollen die Batterien ausgebaut werden.
- 3) Leere Batterien müssen sofort aus dem Gerät entfernt werden.
- 4) Wiederaufladbare Batterien (Ni-Cd-Zellen) dürfen im PC-1500 nicht verwendet werden.
- 5) Wird der PC-1500 ohne Plotter/Drucker CE-150 betrieben, kann das Netz-Lade-Gerät EA-150 als externe Stromversorgung verwendet werden. Diese Anwendung ist wichtig, um beim Batteriewechsel Programm/Datenverluste zu vermeiden. Das Netz-Lade-Gerät wird an die mit "Adaptor" bezeichnete Buchse am PC-1500 angeschlossen.

Achtung: Wenn der PC-1500 mit dem Plotter/Drucker CE-150 verbunden ist, darf das Netz-Lade-Gerät nur an den Plotter/Drucker angeschlossen werden.

1. Einschalten des Computers

Die Einschalttaste **ON** rechts oben im Tastenfeld wird kurz gedrückt. In der Anzeige erscheinen die folgenden Symbole:



- 1) > Das Symbol zeigt an, daß der PC-1500 betriebsbereit ist.
- 2) DEG, RAD oder GRAD gibt die Einheit an, in der Winkel vom Computer verarbeitet werden.
- 3) RUN, PRO oder RESERVE zeigen die gewählte Betriebsart an.
- 4) I, II, III zeigen die Ebenen der RESERVE-Tasten an.
- 5) Die Batterieüberwachung signalisiert ausreichende Stromversorgung.
- 6) Nach Batteriewechsel, Betätigen des All Reset und nach Anschluß von Optionen kann auf der Anzeige.

NEW 0 : CHECK

oder

NEW 0 : CHECK 6

erscheinen.

Durch Eingabe von

CL NEW 0 **ENTER** muß der Computer initialisiert werden.

2. Ausschalten des Computers

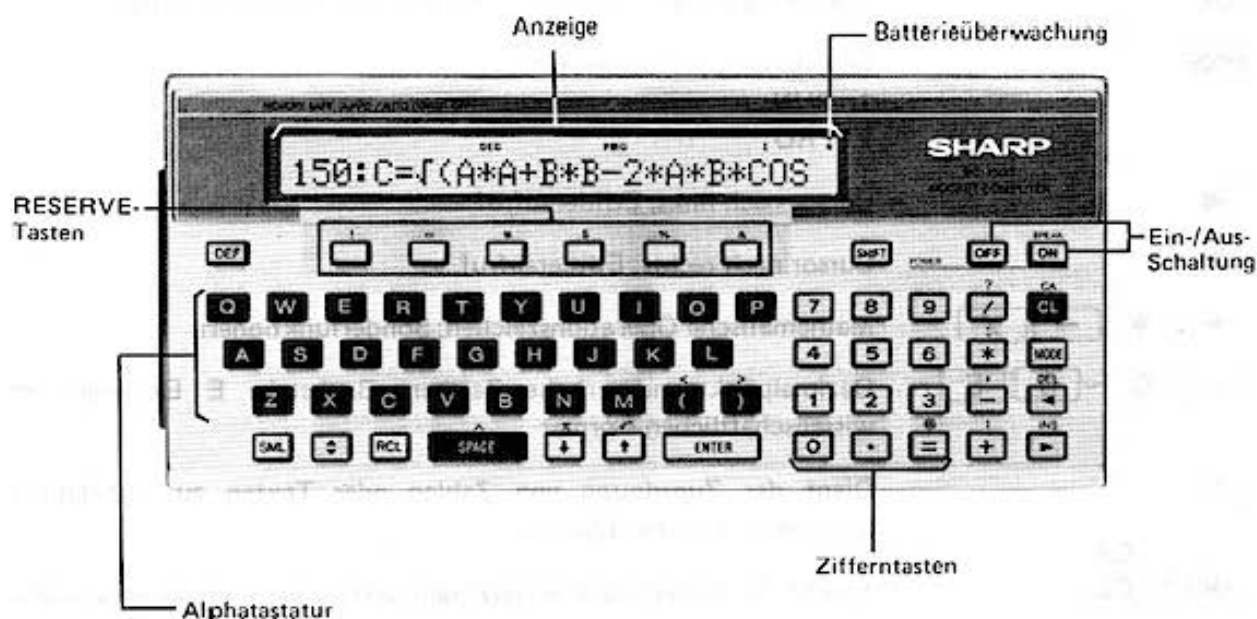
1) Automatisch

Wenn der Computer kein Programm abarbeitet, d.h. kein "BUSY" in der Anzeige erscheint, schaltet er sich ca. 7 Minuten nach der letzten Tastenbedienung automatisch aus. Dabei bleiben alle Daten- und Programminformationen gespeichert. Das Zeilendisplay erscheint beim Einschalten unverändert auf der Anzeige.

2) Manuell

Soll der Computer sofort abgeschaltet werden, muß man die Taste **OFF** betätigen. Die Daten- und Programminformationen bleiben erhalten, jedoch wird das Zeilendisplay gelöscht. Der Computer läßt sich nicht ausschalten, wenn er "BUSY" ist.

3. Die Bedienungselemente



Das Bedienungsfeld des PC-1500 besteht aus 65 Tasten, die zu 3 Blöcken zusammengefaßt sind:

Schreibmaschinen-Alphatastatur, die Zifferntasten mit den Sondertasten CL/CA, MODE, \leftarrow /DEL, \rightarrow /INS und mathematischen Operatoren und in der oberen und unteren Reihe die Tasten für Sonderzeichen, bzw. RESERVE-Tasten mit den zugeordneten Abruftasten, DEF, SHIFT, OFF und ON/BREAK, SPACE, Zeilenkontrolle, ENTER und SML.

Die Alphatastatur ist mit 3 Ebenen belegt.

Bei einfachem Tastendruck gelten die auf den Tasten stehenden Bezeichnungen; will man die über den Tasten stehenden Funktionen ansprechen, muß man vorher die Doppelfunktionstaste **[SHIFT]** drücken. Zur Kontrolle erscheint der Schriftzug SHIFT in der Anzeige. Zweimaliges Bedienen der Taste **[SHIFT]** löscht den Doppelfunktionsbefehl. Die SHIFT-Funktion ist nur gültig für das unmittelbar folgende Zeichen.

Die Tasten Q, W, E, R, T, Y, U, I, O, P sind mit häufig benutzten BASIC-Befehlen in der Doppelfunktion belegt. (siehe Kapitel). Diese werden über die **[DEF]**-Taste abgerufen, die übrigen dienen als Definable Keys (siehe Kapitel).

Über die **[SHIFT]**-Taste in Verbindung mit der Alphatastatur läßt sich die Kleinschreibung abrufen; ist man im SML-Mode, d.h. hat man generelle Kleinschreibung gewählt, so werden die Großbuchstaben über **[SHIFT]** abgerufen.

MODE, SHIFT, DEF sind Wechselschalter.

4. Die Tastenfunktionen

CL	Löscht die Anzeige und die Blockade durch Fehlermeldungen
MODE	Bestimmung der Betriebsart 1. RUN 2. PRO
	Cursor nach links; Editieraufruf
	Cursor nach rechts; Editieraufruf
+ * - /	Mathematische Operationszeichen; Sonderfunktionen
. 0 ~ 9 E	Dezimalpunkt; numerische Zeichen; Buchstabe E Exponent im wissenschaftlichen Format
=	Dient der Zuordnung von Zahlen oder Texten zu bestimmten Speichern; logischer Operator
SHIFT CA CL	Löscht die Anzeige und versetzt den Computer in den Grundzustand
SHIFT MODE DEL	Wahl der Betriebsart RESERVE
SHIFT	Löschen einzelner Zeichen
SHIFT INS	Platzhalter für nachträgliches Einfügen von Zeichen
SHIFT / * - +	Sonderzeichen
SHIFT @ =	Zeichen @ als Sonderfunktion im RESERVE-Mode und indirekte Adressierung der Standardvariable
A ~ Z	Alphatastatur, Variablenname
()	Klammern
SHIFT A ~ SHIFT Z	Kleinbuchstaben
SHIFT < SHIFT >	Vergleichsoperatoren
DEF Q ~ DEF P	Abruf häufig benutzter BASIC-Befehle
DEF A ~ DEF M	Start benannter Programme
DEF	Diese Taste in Verbindung mit den Tasten A, S, D, F, G, H, J, K, L, Z, X, C, V, B, N, M, Space, = ermöglicht das Benennen von Programmen und Starten des Programmlaufs über den Programmnamen. In Verbindung mit den Tasten Q, W, E, R, T, Y, U, I, O, P werden die in der Doppe funktion vorhandenen BASIC-Befehle abgerufen.
! ~ &	RESERVE-Tasten; Sonderzeichen (!, % haben reinen Kommentarbe- deutung); sind die RESERVE-Tasten mit weiteren Ausdrücken programmiert, erfolgt der Abruf der Sonderzeichen über SHIFT

SHIFT

Doppelfunktionstaste

OFF

Abschalttaste

BREAK

ON

ON; Einschalten des Rechners

BREAK; Unterbrechung des Programmablaufs. In der Anzeige erscheint

BREAK IN < Zeilennummer >; unterbricht auch den Drucker- und Magnetbandbetrieb

SML

Kleinschreibung der Alphazeichen; durch nochmaliges Drücken der SML-Taste wird diese Betriebsart wieder aufgehoben; Großbuchstaben können hierbei über **SHIFT** abgerufen werden.

↑

Wahl der RESERVE-Ebene I, II, oder III

RCL

Abruf der RESERVE-Tastenbelegung

SPACE

Leerraum

↓ **↑**

Editiertasten; Sonderfunktion

ENTER

Beschließt die Eingabe von Programmzeilen, RESERVE-Ausdrücken etc.; beim manuellen Rechnen ersetzt es das "=".

SHIFT **SPACE**

Zeichen ^; Sonderfunktion; Exponentiation

SHIFT **↓**



Konstante Pi

SHIFT **√**

Quadratwurzel

SHIFT **↑**

5. Die Anzeige

Der PC-1500 verfügt über eine 26-stellige alphanumerische Flüssigkristallanzeige. Sie besteht aus einer durchgehenden 7 x 156 Punktmatrix; die einzelnen Zeichen werden in einer 5 x 7 Punktmatrix dargestellt. Das Eingaberegister umfaßt 80 Zeichen; alles was über die 26 möglichen Zeichen der Anzeige hinausgeht, wird im Rollschreiberverfahren durchgezogen. Der Cursor zeigt die Stelle, auf welche die nächste Information geschrieben wird. Er steht normalerweise auf der ersten freien Stelle und wird durch einen kurzen, waagerechten Strich dargestellt. Rückt man den Cursor mit Hilfe der  oder -Taste auf ein Zeichen, so erkennt man seine Position am Blinken dieser Stelle. Anhaltender Druck auf den Cursor läßt den gesamten Inhalt der Anzeige schnell in die gewünschte Richtung durchrollen; ein kurzer Druck verschiebt die Anzeige nur um einen Schritt. Versucht man mehr als 80 Zeichen einzulesen, werden diese vom Rechner nicht mehr angenommen. Jedes zusätzliche Symbol überschreibt das vorhergehende. Der Cursor verändert hier sein Aussehen; er blinkt auf dem letzten Zeichen als Warnsignal in der vollen Punktmatrix. Mit Hilfe des Cursors können Eingaben auch nachträglich noch korrigiert werden.

6. Wahl der Betriebsart

Der PC-1500 unterscheidet 3 verschiedene Betriebsarten (Modes):

- 1) Im RUN-Mode kann man den Computer manuell bedienen oder gespeicherte Programme ablaufen lassen.
- 2) Im PROGRAMM-Mode können BASIC-Programme eingegeben oder verändert werden.
- 3) Im RESERVE-Mode können sechs programmierbare Tasten mit Funktionen belegt werden.

Durch Drücken der **MODE**-Taste wird der PROGRAMM-Mode und der RUN-Mode eingestellt. Durch **SHIFT MODE** wird in die Reserve-Betriebsart geschaltet.

Das LOCK-UNLOCK Kommando

Durch das LOCK-Kommando wird das Umschalten von einer Betriebsart in eine andere verhindert, d.h. die Mode-Taste wird blockiert.

Das Kommando UNLOCK hebt die Sperre wieder auf.

Gibt man das Kommando LOCK im RUN-Mode, so sind gespeicherte Programme und die programmierbaren Tastenfunktionen gegen unbeabsichtigte Änderungen geschützt, da diese nur im PRO- bzw. RESERVE-Mode möglich sind.

Beispiel:

Ein Navigationsprogramm wird von der Kassette geladen. Im RUN-Mode wird das Kommando LOCK gegeben. Damit ist es z.B. auf einem schwankenden Schiff nicht möglich, versehentlich in den PROGRAMM-Mode umzuschalten und das Programm durch Drücken falscher Tasten zu zerstören.

7. Speicherorganisation

Der PC-1500 enthält mehrere Speicherbereiche:

BASIC – STACK		196 BYTE
BASIC – PROGRAMME ↓ ODER ↑ NUMERISCHE + TEXTVARIABLEN	STANDARD	1,850 KBYTES
	CE-151	+ 4,096 KBYTES
	CE-155	+ 8,192 KBYTES
EINGABE – PUFFER		80 BYTES
STANDARDVARIABLENSPEICHER		
A ~ Z bzw. @(1) ~ @(26)		208 BYTES
RESERVEPEICHER		188 BYTES
STANDARDVARIABLENSPEICHER		
A\$ – Z\$ bzw. @\$ (1) ~ @\$ (26)		416 BYTES

7.1 Standardvariablen-Speicher

Der Bereich enthält

- die 26 numerischen Variablen A – Z. Diese sind identisch mit dem numerischen Feld @ (1) – @ (26).
- die 26 Textvariablen AS – ZS, die mit dem Textfeld @\$ (1) – @\$ (26) identisch sind. Die maximale Zeichenkapazität pro Variable beträgt 16 Zeichen.

Diese numerischen und Textvariablen können nicht gelöscht werden. Durch das Kommando NEW oder CLEAR wird ihr Wert auf Null bzw. das ASCII-Nullzeichen gesetzt.

7.2 Hauptspeicher

Der Hauptspeicher umfaßt den im PC-1500 enthaltenen Speicherplatz von 1,85 KByte und die möglichen Erweiterungen CE-151 (4 KByte) oder CE-155 (8 KByte).

Im Hauptspeicher werden gespeichert:

- Die BASIC-Programme – Jede Zeilennummer verbraucht 3 Bytes, jedes BASIC-Schlüsselwort 2 Bytes und jedes sonstige Zeichen einschließlich ENTER 1 Byte.
 - Alle numerischen und Textvariablen mit zwei Zeichen im Namen. Der Speicherplatz für diese Variablen wird bei der ersten Benutzung reserviert.
 - Alle mit der DIM-Anweisung erzeugten numerischen und Textfelder. Jede Variable verbraucht 7 Bytes für den Namen. Numerische Variable und Felder benötigen zusätzlich 8 Bytes für den Zahlenwert bzw. für jedes Element. Textvariable verbrauchen zusätzlich 16 Bytes, Textfelder benötigen so viele Bytes, wie ihre Buchstabenkapazität ausmacht.
- Ist der Hauptspeicher vollständig belegt, so wird bei dem Versuch, weitere Programmzeilen oder Variable zu definieren, ein Fehler gemeldet (ERROR 10).

Hinweis: 1 Byte ist der Speicherbereich, der ein Zeichen speichern kann. 1 Kbyte sind 1024 Bytes.

Die Funktion STATUS gibt Auskunft über die Speicherbelegung.

7.3 BASIC-Stack

Der BASIC-Stack ist ein spezieller Speicherbereich, der bei der Berechnung von numerischen Ausdrücken, zum Bearbeiten von FOR-NEXT-Schleifen und GOSUB-RETURN-Sprüngen benutzt wird. Er hat eine Länge von 196 Bytes.

Von diesen 196 Bytes verbraucht jede aktive FOR-NEXT Schleife 12 Bytes. Jede aktive Unterprogrammebene verbraucht 6 Bytes. Bei der Berechnung eines numerischen Ausdrucks verbraucht jedes Zwischenergebnis 8 Bytes und jeder Operator oder Funktion 2 Bytes.

Daraus folgt, daß auf dem PC-1500 maximal 16 FOR-NEXT Schleifen ineinandergeschachtelt werden dürfen. Das ist allerdings nur dann möglich, wenn gleichzeitig keine Unterprogramme aktiv sind und keine komplexen numerischen Ausdrücke berechnet werden. Reicht der Stack-Speicherplatz nicht aus, so wird ein Fehler gemeldet. (ERROR 14 oder ERROR 15).

7.4 RESERVE-Speicher

In diesem Speicher stehen die Bedeutungen der programmierbaren Funktionstasten (s. RESERVE-Mode).

8. Rechnen ohne Programmunterstützung

Der PC-1500 bietet dem Benutzer eine Vielzahl von Möglichkeiten. Er kann wie ein normaler Taschenrechner verwendet werden. Dabei stehen alle gebräuchlichen mathematischen Funktionen zur Verfügung. Bei angeschlossenem Drucker werden alle Eingaben und Ergebnisse protokolliert, wenn der Schalter PRINT auf P steht.

Der Rechner verfügt je nach Speicherausrüstung über bis zu 700 Zahlenspeicher (Variable), in denen Zwischenergebnisse gespeichert werden können. Außerdem können bis zu 26 Klammerebenen benutzt werden.

Der PC-1500 kann in der Programmiersprache BASIC programmiert werden. Fast alle Befehle dieser Sprache können auch direkt eingegeben und ausgeführt werden.

Für einfache Rechnungen, die nur einmal ausgeführt werden, kann der PC-1500 wie ein Taschenrechner verwendet werden. Dies geschieht im RUN-Mode.

Die allgemeine Form einer solchen Rechnung ist:

CL < numerischer Ausdruck > **ENTER**

Als numerischen Ausdruck bezeichnen wird jede mathematische Formel, die einen Zahlenwert als Ergebnis hat.

Beispiele:

CL	2 + 4	ENTER	(Addition)
CL	2 - 4	ENTER	(Subtraktion)
CL	5.5 / 2.7	ENTER	(Division)
CL	2 * 3.1415	ENTER	(Multiplikation)
CL	27 * (15 + 3) + 15^3	ENTER	(Benutzung von Klammern, Exponentiation)

Die Taste **CL** dient dabei zum Löschen der Anzeige.

Hinweis:

Erst durch Drücken der Taste **ENTER** wird die Rechnung oder das Kommando ausgeführt und das Ergebnis angezeigt. Dieses Ergebnis kann in der nächsten Rechnung weiter verwendet werden. Man löscht diese Zahl dann nicht mit der **CL**-Taste, sondern gibt als nächste Zeichen der neuen Formel ein Operationszeichen +, -, *, oder ^ ein.

Beispiel:

Anzeige	
CL 2 * 3.141527 ENTER	6.283054
/ 180	6.283054/180_
ENTER	3.490585556E-02

Zahlen können in den PC-1500 auch im wissenschaftlichen Format eingegeben werden. Das E im letzten Zahlenbeispiel steht für "10 hoch". Durch dieses Zahlenformat ist es möglich, in den PC-1500 Zahlen bis zu einer Größe von $\pm 9.999999999 \text{ E } + 99$ einzugeben und zu verarbeiten.

Wird dieser Bereich bei einer Rechnung überschritten, so wird ein Fehler gemeldet (ERROR 16). Ist der Betrag einer Zahl kleiner als $1.E-99$, so wird die Zahl auf Null gesetzt.

Hinweis: Wird eine Zahl mit mehr als 2stelligem Exponenten eingegeben, so wird die überzählige Ziffer ignoriert. Es werden dabei die beiden zuletzt eingegebenen Zahlen bewertet.

Bemerkung:

- 1) Wie im englischen Sprachraum üblich, wird nicht das Komma, sondern der Punkt zum Trennen von ganzem und gebrochenem Anteil von Zahlen benutzt.
- 2) Für den Operator "geteilt durch" wird der Schrägstrich / und nicht der Doppelpunkt verwendet.
- 3) Der Stern * steht für "multipliziert mit"
- 4) Das Dach steht für "hoch", d.h. beispielsweise $15 \wedge 3$ bedeutet, daß 15 mit 3 exponenziert wird.
Ist die Basis negativ, muß sie in Klammern gesetzt werden und es sind nur ganzzahlige Exponenten zugelassen. Ansonsten wird ein Fehler gemeldet (ERROR 39).

Der PC-1500 zeigt Ergebnisse im wissenschaftlichen Format an, wenn mehr als 10 Ziffern für die Darstellung erforderlich sind. Man kann das Anzeigeformat durch die BASIC-Anweisung USING verändern.

8.1 Rechengenauigkeit



Zahlen werden vom PC-1500 mit einer Genauigkeit von 10 Ziffern verarbeitet. Dies kann bei längeren Rechnungen zu Rundungsfehlern führen.







Diese Genauigkeitsbeschränkung tritt auch beim Exponenzieren auf oder wenn Zahlen sehr unterschiedlicher Größe addiert werden.



Beispiel:

		Ergebnis
$41 \wedge 3$	<input type="button" value="ENTER"/>	68920.99999
$41 \wedge 3 - 41 * 41 * 41$	<input type="button" value="ENTER"/>	-0.0000055
$1.E 12 + 1 - 1.E 12$	<input type="button" value="ENTER"/>	0

8.2 Kontrollieren und Verändern der Eingabe (Editieren)

Der PC-1500 bietet die Möglichkeit, Tippfehler während der Eingabe zu korrigieren und schon eingegebene Formeln zu überprüfen und zu verändern. Dazu dienen die Cursortasten  und . Mit diesen Tasten kann der Cursor nach rechts oder links bewegt werden, ohne daß eingegebene Formeln verloren gehen. Man kann jedes Zeichen verändern, indem man den Cursor darunter stellt und ein neues Zeichen eingibt.

Mit   (DELe.) wird das Zeichen über dem Cursor gelöscht. Mit   (INSert) wird ein Platzhalterzeichen  in die Formel eingefügt, das dann durch ein neues Zeichen überschrieben werden kann. Mit dem  werden alle überflüssigen Platzhalter aus der Formel gelöscht.

Ist eine Formel schon ausgerechnet oder wurde beim Ausrechnen eine Fehlermeldung erzeugt, so kann die Formel durch die Taste  oder  in die Anzeige zurückgeholt und wie oben beschrieben geändert werden.

Hinweis:

BASIC-Schlüsselwörter wie SIN, COS, LET usw. werden mit dem Ausrechnen nach Betätigung der ENTER-Taste intern abgekürzt. Obwohl das Schlüsselwort unverändert auf der Anzeige erscheint, wird es beim Editieren durch ein einziges neues Zeichen vollständig überschrieben.

8.3 Speichern von Zwischenergebnissen

Der PC-1500 verfügt über bis zu 700 Speicherplätze (Variablen) für einzelne Zahlenwerte. Zusätzlich kann mit der BASIC-Anweisung DIM weiterer Speicherplatz reserviert werden.

Auf Variable wird über den Variablennamen zugegriffen. Ein solcher Name besteht aus einem einzelnen Buchstaben, zwei Buchstaben oder einem Buchstaben und einer Ziffer. An den Namen können noch weitere Buchstaben oder Ziffern angehängt werden, solange dadurch kein BASIC-Schlüsselwort entsteht. Diese zusätzlichen Zeichen werden vom Rechner ignoriert. (Es sind nur Großbuchstaben für Variablennamen zulässig).

Ausnahme: Die Namen LF, IF, LN, PI, TO sind verboten, um Verwechslungen mit den BASIC-Schlüsselwörtern zu vermeiden.

Eine Variable wird beschrieben durch eine Anweisung der Form:

< Variablenname > = < numerischer Ausdruck > **ENTER**

Beispiel:

Anzeige	
A = 55 ENTER	55
IX = 99 / 5 + 2 ENTER	21.8
V0 = IX * 9.81 ENTER	213.858

In einem numerischen Ausdruck kann jede Zahl durch einen Variablennamen ersetzt werden.

Beispiel:

Anzeige	
A = 5 ENTER	5
B = 10 ENTER	10
C = A + B ENTER	15

Der Wert der Variablen bleibt auch nach dem Aus- und Einschalten des Rechners erhalten. Wird ein BASIC-Programm mit dem Kommando RUN gestartet, so werden alle Variablen gelöscht, deren Name zwei oder mehr Zeichen hat. Variable, deren Name nur aus einem Buchstaben bestehen, bleiben erhalten.

Mit dem Kommando CLEAR werden alle Variablen gelöscht, d.h. der Wert der Variablen wird auf null gesetzt.

Beispiel:

Anzeige	
CLEAR ENTER	>
X ENTER	0

9. Mathematische Funktionen

Der PC-1500 verfügt über alle gebräuchlichen mathematischen Funktionen. Diese werden jedoch nicht durch spezielle Tasten aufgerufen, sondern über die Buchstabentasten in der Form eingegeben, wie sie in mathematischen Formeln gebräuchlich ist.

Beispiel:

CL	SIN	(2 * PI)	ENTER
CL	SIN	15 * SIN 15 + COS 15 * COS 15	ENTER
CL	EXP	(2 * LN 5)	ENTER

Hinweis: Rechnet man häufig mit bestimmten Funktionen, so kann man zur Vereinfachung der Eingabe die "programmierbaren Tasten" verwenden. (s. RESERVE MODE)

Eine Liste aller Funktionen ist in der Beschreibung der BASIC-Sprache auf Seite 32 enthalten.

9.1 Klammerregeln

Die Reihenfolge der einzelnen Schritte beim Ausrechnen eines komplexen numerischen Ausdrucks wird durch Klammern geregelt. Wie in der Mathematik gibt es bei verschiedenen Rechenoperationen eine implizite Klammerung, d.h. bestimmte Operationen haben Vorrang vor anderen.

Beispiel:

$$3 * 5 + 7 * 4 \quad \text{ist gleichbedeutend mit} \\ (3 * 5) + (7 * 4)$$

Die Rangfolge der mathematischen Operationen sind:

höchster Rang:

niedrigster Rang

1. Ausdrücke in Klammern werden immer zuerst berechnet.
2. Funktionsoperationen in Bezug auf das folgende Argument
Beispiel: $\text{SIN } 3 + 4 = (\text{SIN } 3) + 4$
3. Exponentiation
Beispiel: $2 * A ^ 3 = 2 * (A ^ 3)$
4. Vorzeichen +, -
5. Multiplikation, Division *, /
6. Addition, Subtraktion +, -
7. Vergleichsoperatoren
8. Logische Operatoren AND, OR, NOT.
Werden in einem numerischen Ausdruck mehrere Operationen gleichen Ranges nebeneinander gestellt, so werden sie von links nach rechts ausgeführt.

Beispiel:

$$24/3/2 \\ \text{entspricht} \\ (24/3)/2$$

Hinweis: Die Vergleichsoperatoren und logischen Operatoren werden später im Zusammenhang mit den BASIC-Instruktionen erklärt.

9.2 Hexadezimalzahlen

Natürliche Zahlen im Bereich zwischen 0 und 65535 können in den PC-1500 auch als Hexadezimalzahlen (Zahlen im 16er System) eingegeben werden. Der Zahl wird ein & vorangestellt. Es werden die Ziffern 0 – 9 und die Buchstaben A – F benutzt.

Beispiel:

Ausgabe:

& A	ENTER	10
& 10	ENTER	16
& FFFF	ENTER	65535

Die Ausgabe von Zahlen in Hexadezimalform ist auf dem PC-1500 nicht vorgesehen, kann aber durch folgendes Programm gelöst werden:

```

10 INPUT "Hexzahl?"; N
20 IN = INT N : NS$ = " "
30 HS$ = "0123456789ABCDEF"
40 M = N : N = INT (N/16)
50 M = M - N * 16 + 1
60 NS$ = MID$ (HS$, M, 1) + NS$
70 IF N > 0 THEN 40
80 PRINT "Hexzahl = "; NS$
90 END

```


10. Textausdrücke

Textausdrücke sind Bestandteil der BASIC-Sprache. Sie können auch in der Taschenrechnerfunktion eingegeben werden. Wie bei den numerischen Ausdrücken gibt es Textkonstanten, Textvariablen, Textfunktionen und zusammengesetzte Texte. Eine Textkonstante ist eine beliebige Zeichenfolge, die durch Anführungsstriche (") begrenzt ist. (Die Anführungsstriche sind nicht Bestandteil des Textes).

Beispiel:

Anzeige	
CL "JUHU" ENTER	JUHU

Textvariable sind Speicherplätze für Texte. Sie können bis zu 16 Zeichen enthalten. Mit der DIM-Instruktion lassen sich Textvariable mit größerer Länge erzeugen. Der Variablenname hat die gleiche Form wie bei einer numerischen Variablen mit einem zusätzlichen \$-Zeichen am Ende.

Beispiel:

Anzeige	
A\$ = "JUHU" ENTER	JUHU
A\$ ENTER	JUHU

Texte können mit dem + Zeichen aneinander gesetzt werden:

Beispiel:

Anzeige	
A\$ = "Pocket" ENTER	Pocket
A\$ + "Computer" ENTER	PocketComputer

Mit den Textfunktionen (MID \$, LEFT \$ etc.) können weitere Operationen mit Texten ausgeführt werden. Eine genaue Beschreibung der Textfunktionen ist in der BASIC-Beschreibung (Seite 89) enthalten.

Hinweis:

Auch das Leerzeichen (Space) ist innerhalb eines Textes ein gültiges Zeichen.

11. Vergleichsausdrücke

Mit den Vergleichsoperatoren

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

können zwei numerische oder Textausdrücke miteinander verglichen werden. Ist die Vergleichsaussage richtig, liefert der Rechner das Ergebnis 1, sonst 0. Vergleichsausdrücke erhalten ihre Bedeutung in der BASIC-Anweisung IF (s. S. 56)

Beispiel:

		Anzeige
<input type="button" value="CL"/> 2 < 1 <input type="button" value="ENTER"/>		0
<input type="button" value="CL"/> 1 <= 2 <input type="button" value="ENTER"/>		1
<input type="button" value="CL"/> SIN 1 < .7 <input type="button" value="ENTER"/>		0 im RAD-Mode 1 im DEG-Mode 1 im GRAD-Mode
<input type="button" value="CL"/> "Willi" > "Anton" <input type="button" value="ENTER"/>		1 weil Willi im Alphabet später kommt

Da das Ergebnis eines logischen Ausdrucks eine Zahl ist, kann es in einer numerischen Variablen gespeichert werden.

12. BASIC-Befehle

Die meisten BASIC-Instruktionen können auch direkt ausgeführt werden.

Beispiel:

		Anzeige
GPRINT 127 <input type="button" value="ENTER"/>		<div style="border: 1px solid black; padding: 5px; display: inline-block;"> : : : : : </div>
A = 10 + 20 <input type="button" value="ENTER"/> LPRINT "Summe"; A <input type="button" value="ENTER"/>		Der Drucker druckt Summe 30
COLOR 0		Der 1. Farbstift wird angewählt

13. Numerische Funktionen

Der PC-1500 bietet Ihnen eine große Anzahl von mathematischen Standardfunktionen.

Winkelfunktionen

DEGREE	Winkeleinheit-Kommando
GRAD	"
RADIAN	"
SIN	Sinus
COS	Cosinus
TAN	Tangens
ACS	Arcuscosinus
ASN	Arcussinus
ATN	Arcustangens
DMS } DEG }	Dezimal ↔ Hexadezimalsystem Umrechnungsfunktion

13.1 Trigonometrische-/Umkehrfunktion

Die Parameter bzw. die Funktionsergebnisse der trigonometrischen Funktionen (ACS, ASN, ATN) können in verschiedenen Winkeleinheiten festgelegt werden. Die Kommandos zur Festlegung sind:

DEGREE	(0 – 90°)	DEG – Mode (Altgrad)
GRAD	(0 – 100°)	GRAD – Mode (Neugrad)
RADIAN	(0 – $\pi/2$)	RAD – Mode (Bogenmaß)

(In Klammern sind die Winkel für einen Viertelkreis angegeben)

Programmgesteuert können Sie das Maß der Winkeleinheiten mit den Kommandos DEGREE/GRAD/RADIAN verändern; die zur Zeit gültige Einheit ist auf der Anzeige sichtbar.

Beispiel:

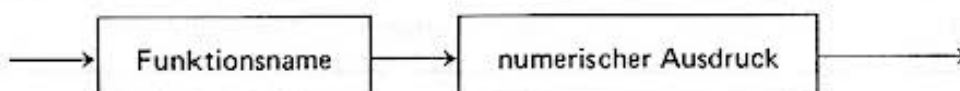
```

10  DEGREE : PRINT SIN 45
20  GRAD   : PRINT SIN 50
30  RADIAN : PRINT SIN ( $\pi/8$ )

```

Die Ergebnisse der Funktionen sind numerische Werte. Als Parameter ist im allgemeinen ein numerischer Ausdruck zugelassen (s. Kap. 16.7).

Syntax

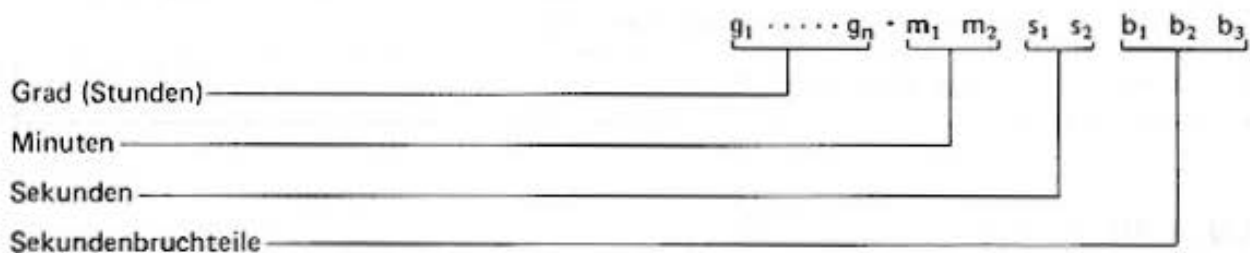


Funktionsnamen sind:

SIN, COS, etc.

DMS, DEG

Bei diesen Umrechnungen werden sexagesimal geteilte Winkel oder Zeiten in folgendem Format verarbeitet



DMS Umrechnung Dezimalzahl \rightarrow Sexagesimalzahl
 Funktionsergebnis ist eine Sexagesimalzahl

< Sexagesimalergebnis > \leftarrow DMS < Dezimalzahl >

DEG Umrechnung Sexagesimalzahl \rightarrow Dezimalzahl
 Funktionsergebnis ist eine Dezimalzahl

< Dezimalergebnis > \leftarrow DEG < Sexagesimalzahl >

Beispiele: Umrechnung von 15.4125° in Grad/Minuten/Sekunden
 Umrechnung von $15^\circ 24' 45''$ in Dezimalgrad

Eingabe	Anzeige	Anmerkung
CL D M S 15.4125 ENTER D E G 15.2445 ENTER	DMS15.4125 _ 15.2445 DEG15.2445 _ 15.4125	 $15^\circ 24' 45''$ 15.4125°

13.2 Mathematische Funktionen

LN	natürlicher Logarithmus (Basis e)
LOG	allgemeiner Logarithmus (Basis 10)
EXP	Exponentiation zur Basis e

LN, LOG

Die Funktionen berechnen den natürlichen Logarithmus LN bzw. Zehnerlogarithmus LOG.

Für die beiden Logarithmusfunktionen darf der Wert des numerischen Ausdrucks nicht negativ sein (ERROR 39).

Beispiel:

1)

```
10  A = LN 100
20  B = EXP A
30  PRINT B
```

Anzeige

100

EXP

Exponentialfunktionen $e^x = \text{EXP } X$

Beispiel: $e^{2.3}$ ist anzugeben als
EXP 2.3

ABS	Absolutbetrag
INT	ganzzahliger Anteil
SGN	Signum (Vorzeichen)
SQR, $\sqrt{\quad}$	Quadratwurzel

ABS

Die ABS Funktion ermittelt den Absolutbetrag des numerischen Ausdrucks.

Beispiel:

ABS (-6)	6
ABS (6)	6
ABS (-3 * 7)	21

INT

Die INT Funktion ermittelt den ganzzahligen Anteil des numerischen Ausdrucks.

Beispiel:

INT (π)	3
INT (2.15)	2
INT (-3.14)	-4

SGN

Die SGN Funktion ermittelt das Vorzeichen des numerischen Ausdrucks. Das Funktions-
ergebnis hat folgende Bedeutung:

SGN X = 1 für $X > 0$

SGN X = 0 für $X = 0$

SGN X = -1 für $X < 0$

Beispiel:

	Anzeige
SGN (3 + 7 - 50)	-1

SQR oder $\sqrt{\quad}$

Die SQR Funktion berechnet die Quadratwurzel des numerischen Ausdrucks, negative Werte
sind nicht zulässig (ERROR 39).

Beispiel:

	Anzeige
$\sqrt{2}$	1.414213562
SQR (2 * 2)	2
$\sqrt{-1}$	ERROR 39

π (PI)

Die Konstante π ist als

$$\pi = 3.141592654$$

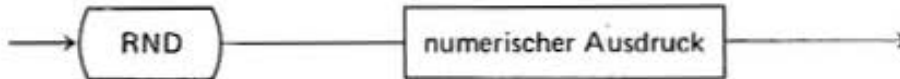
gespeichert.

13.3 Zufallszahlen

RND RANDOM

RND
Die RND Funktion ermittelt eine Zufallszahl.

Syntax



Ergebnis: numerischer Wert

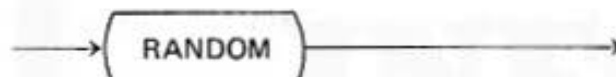
Der Wert der numerischen Variablen bestimmt das Intervall, aus dem die Zufallszahl gezogen wird. Die Zufallszahl wird als Ergebnis ausgegeben. Das Intervall der Zufallszahl ist wie folgt festgelegt:

$$\begin{array}{ll}
 \text{RND } x & \\
 0 \leq x < 1 & \Rightarrow 0 \leq \text{RND } x < 1 \\
 x \geq 1 & \Rightarrow 1 \leq \text{RND } x \leq \text{INT } x
 \end{array}$$

Die Genauigkeit der Zufallszahl beträgt 10 Stellen. Die gleiche Zufallszahlenfolge erhalten Sie jeweils nach dem Einschalten des Computers wieder.

RANDOM
Die Instruktion initiiert eine neue Folge von Zufallszahlen.

Syntax



Die RND Instruktion ermittelt jedesmal nach dem Einschalten des Computers die gleiche Folge von Zufallszahlen. Die RANDOM-Anweisung erlaubt es Ihnen, eine andere zufällige Folge zu initiieren.

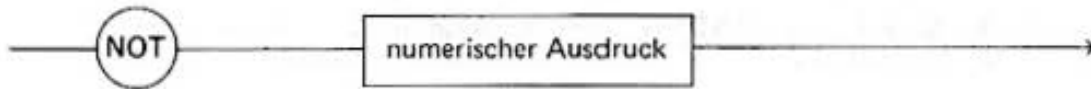
Beispiel:

PRINT "Du hast eine "; RND 6 ;" gewuerfelt".

13.4 NOT-Funktion

Invertieren der Binärdarstellung eines numerischen Ausdrucks.

Syntax



Anwendung auf logische Ausdrücke:

NOT 1 < 2 ergibt 1 (NOT 1 ist -2, die Aussage ist wahr)

NOT 1 = 2 ergibt 0 (NOT 1 ist -2, die Aussage ist falsch)

Anwendung auf numerische Ausdrücke:

Der Wert des numerischen Ausdrucks muß zwischen -32768 und +32767 liegen. Der numerische Ausdruck wird als 16-stellige Binärzahl, d.h. im Zweiersystem, interpretiert. Die Funktion NOT invertiert jede Ziffer der Binärzahl.

Negative Binärzahlen werden durch das "Zweierkomplement" dargestellt, d.h. -X ergibt sich aus (NOT X) + 1

Beispiele:

X (Dezimal)	x (Binär)	NOT X (Binär)	NOT x (Dezimal)
-1	1111111111111111	0000000000000000	0
0	0000000000000000	1111111111111111	-1
1	0000000000000001	1111111111111110	-2
15	0000000000001111	1111111111100000	-16

Hinweis:

Beachten Sie bitte, daß die NOT-Funktion angewandt auf logische und numerische Ausdrücke verschiedene Ergebnisse liefert.

A = 2 NOT A < B ist 1
 B = 3 NOT C ist -2
 C = A < B

Hinweis:

Bei der Benutzung der NOT-Funktion in einer Ausdruck wie z. B.

A OR (NOT B)

wird A als Dezimalzahl und (NOT B) als Binärzahl interpretiert.

durch Eingabe von "Minus, Minus" wird (NOT B) als Dezimalzahl interpretiert.

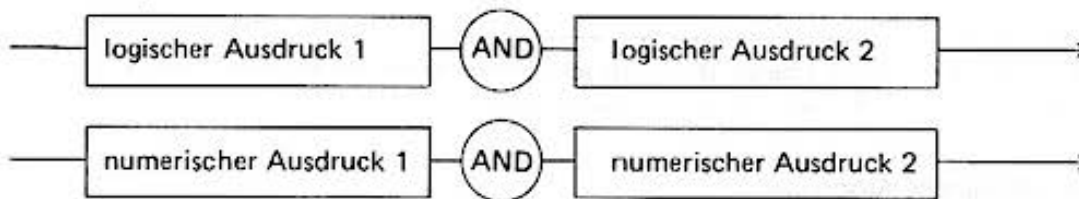
Der o. g. Ausdruck wird dann eingegeben als:

A OR — — (NOT B)

AND-Operator

“UND”-Verknüpfung von zwei logischen Ausdrücken.
Logisches “UND” der binären Darstellung zweier numerischer Ausdrücke.

Syntax



Anwendung auf logische Ausdrücke:

Das Ergebnis ist wahr (1), wenn der Ausdruck 1 und der Ausdruck 2 wahr sind, sonst ist es falsch (0).

Anwendung auf numerische Ausdrücke:

Die Werte der Ausdrücke müssen zwischen -32768 und +32767 liegen und werden als 16-stellige Binärzahlen interpretiert. Das Ergebnis hat in jeder binären Stelle eine 1, in der im Ausdruck 1 und beim Ausdruck 2 eine 1 steht, sonst eine 0:

Beispiel:

X AND 1

Ergebnis:

0 wenn X eine gerade Zahl ist
1 wenn X eine ungerade Zahl ist

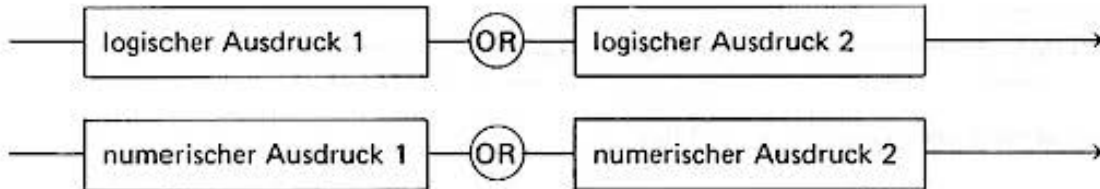
34 AND 70

2
34 \triangleq 0100010
70 \triangleq 1000110
AND 0000010 \triangleq 2

OR-Operator

"ODER"-Verknüpfung von zwei logischen Ausdrücken.
Logisches "ODER" der binären Darstellung zweier numerischer Ausdrücke.

Syntax



Anwendung auf logische Ausdrücke:

Das Ergebnis ist wahr (1), wenn Ausdruck 1 oder Ausdruck 2 oder beide wahr sind, sonst ist es falsch (0):

Die Werte der Ausdrücke müssen zwischen -32768 und +32767 liegen und werden als 16-stellige Binärzahlen interpretiert. Das Ergebnis hat in jeder binären Stelle eine 1, in der im Ausdruck 1 oder im Ausdruck 2 oder in beiden eine 1 steht, sonst eine 0.

Beispiel:

17 OR 3

Ergebnis:

19

$17 \triangleq 10001$

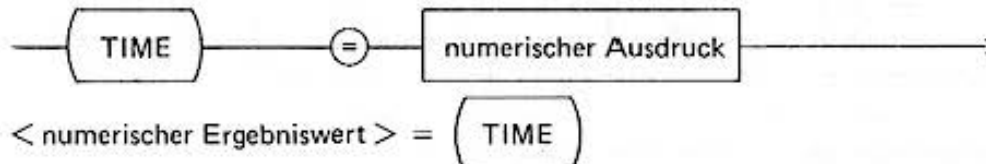
$3 \triangleq 00011$

OR 10011 $\triangleq 19$

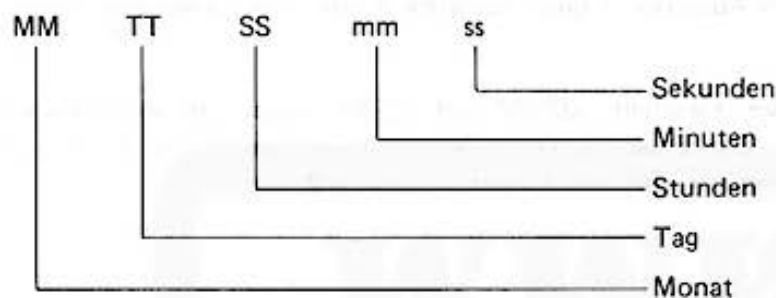
13.5 TIME

Mit der Lese-/Schreibfunktion TIME kann im Computer die Uhrzeit gesetzt bzw. manuell oder programmgesteuert abgefragt werden.

Syntax Lesen und Schreiben



Der PC-1500 hat eine interne Uhr; die Uhrzeit kann mit der TIME-Funktion gesetzt werden. Der Wert hat dabei mit 6 Stellen vor und 4 Stellen hinter dem Dezimalpunkt folgende Bedeutung:



Die Uhrzeit kann ausgelesen werden, wenn TIME als Lesefunktion benutzt wird.

Beispiel:

- (1) 1. Mai 12h 15' 30'
 TIME = 050112.1530

- (2) Ausgabe der Uhrzeit auf dem Display : 3 Sekunden sind vergangen seit der Eingabe laut Beispiel 1
 TIME

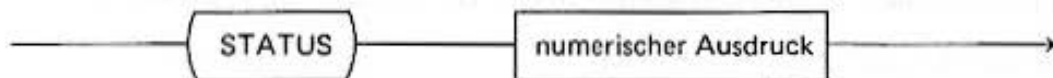
5 0 1 1 2 . 1 5 3 3

- (3) TIME = TIME + 0.0030
 Justierung der Uhr um 30 Sekunden

13.6 STATUS, MEM

Die Statusfunktion gibt Auskunft über die Größe und die Adressen von freiem und belegtem Speicherplatz. Außerdem die letzte bearbeitete Zeilennummer. (s. Speicherorganisation)

Syntax:



Numerischer Ausdruck	Ergebnis
0	Anzahl der unbelegten Bytes (kleinste Speichereinheit) in der "Main Memory"
1	Anzahl der belegten Bytes
2	Adresse des ersten freien Bytes
3	Adresse des letzten freien Bytes
4 – 255	Nummer der vorher bearbeiteten Zeile im Programmablauf. Zuletzt bearbeitete Zeilennummer nach einem STOP oder BREAK.

An Stelle von STATUS 0 kann die Funktion MEM benutzt werden; sie liefert dasselbe Ergebnis.

14. Programmieren in BASIC

Die herausragende Leistungsfähigkeit des PC-1500 wird erst im Programmbetrieb voll ausgeschöpft.

Um Ihnen das Programmieren so einfach wie möglich zu machen, können Sie die weit verbreitete Programmiersprache BASIC verwenden. Das Wort BASIC steht für Beginners All-Purpose Symbolic Instructions Code und wurde 1965 am Dartmouth College entwickelt.

Die großen Vorzüge dieser Sprache sind:

- leichte Erlernbarkeit
- schrittweise Programmentwicklung
- Dialogfähigkeit

Der Leser kann die Sprache fast schon beim Überfliegen der Tabelle verstehen. Es gibt nur einige wenige zusätzliche Punkte, die nicht selbsterklärend sind.

Ein BASIC-Programm kann schrittweise verändert, und Anweisung für Anweisung geprüft werden. Die Dialogfähigkeit gestattet es dem Benutzer, während der Ausführung Daten einzugeben und so den Programmablauf zu beeinflussen.

14.1 BASIC- Übersicht

Wie jede natürliche Sprache, so hat auch die Computer (formale) Sprache BASIC eine Grammatik, wenn auch eine sehr einfache.

Das Alphabet umfaßt die Buchstaben A . . Z, a . . z, die Ziffern 0, 1 . . . 9 sowie die Sonderzeichen. (, ; : + - * \$. & @ % # ! ? < >)

Zur Vermeidung von Verwechslungen wird unterschieden zwischen 0 (Null) und dem Buchstaben O:

Aus diesem Zeichenvorrat setzen sich die BASIC Sprachelemente zusammen. Auf den ersten Blick erscheint es umständlich, daß es keine Funktionstasten für sin, log etc. gibt, wie bei herkömmlichen Taschenrechnern üblich. BASIC bietet jedoch so viele Möglichkeiten, daß einige 100 Tasten nötig wären. Das würde viel mehr Sucharbeit erfordern, als die einfache Eingabe mit der von der Schreibmaschine gewohnten Tastatur. Häufig verwendete Sprachelemente können jedoch unter den Definable Keys direkt abgerufen werden.

BASIC kennt folgende Sprachelemente, die vom Benutzer frei definiert werden:

- numerische Konstanten
- Textkonstanten
- numerische Variablen
- Textvariablen
- dimensionierte Variablen oder Feldvariablen
- und solche, die als Schlüsselwörter vorgegeben sind für:
 - Standfunktionen
 - Zuweisungen
 - Eingabeanweisungen
 - Ausgabeanweisungen
 - Steueranweisungen
 - Kommandoanweisungen
 - und Operationszeichen (+ - / *)

Diese Sprachelemente werden wie Worte zu Satzsätzen zusammengefügt, die der Rechner als Befehlsfolgen interpretiert und Zeile für Zeile abarbeitet.

Wichtig: Damit der Computer die einzelnen Sprachelemente eindeutig erkennen kann, muß zwischen zwei aufeinander folgenden ein Sonderzeichen oder ein Zwischenraum stehen oder eines ein Schlüsselwort sein.

Tabelle:

LET < Variable > = < Ausdruck >
GOTO < Zeilennummer >
GOSUB < Zeilennummer >
IF < Ausdruck > < Vergleichsoperator > < Ausdruck > THEN < Instruktion >
FOR < Variable > = < Ausdruck > TO < Ausdruck > STEP < Ausdruck >
NEXT < Variable >
READ < Variable >, < Variable >, < Variable >
PRINT < Ausdruck >, < Ausdruck >
STOP
END
DIM < Variable > (< Zahl >, < Zahl >)
REM < beliebige Zeichenfolge >

Beispiele von elementaren BASIC Instruktionen.

Die in den spitzen Klammern angegebenen metalinguistischen Variablen stellen Platzhalter für die vom Programmierer benutzten Sprachelemente dar.

LET und GOTO sind die Zuweisungs- bzw. Steuerinstruktionen.

Bei einer Zuweisungsinstruktion wird einer numerischen Variablen der Wert eines arithmetischen Ausdrucks zugewiesen (z.B. $C = A * A + B * B$) bzw. einer Textvariablen eine Zeichenfolge (z.B. $BS = "JUHU"$).

GOSUB ist der Unterprogrammaufruf.

Die Vergleichsoperatoren sind

=, <, <=, >, >= und <>.

Das letzte Symbol steht für ungleich. IF ... THEN erlaubt einen Vergleichsoperator zwischen zwei Ausdrücken (numerische bzw. Text). Die Befehlsfolge wird an der dem THEN folgenden Anweisung fortgeführt, falls der Vergleich das Ergebnis "wahr" liefert.

FOR definiert den Beginn, NEXT das Ende eines Programmteiles, das in einer Schleife durchlaufen wird.

READ weist den aufgelisteten Variablen die Werte zu, die in einer DATA-Anweisung angegeben sind.

PRINT (LPRINT) gibt die Werte der Zeichenketten und Variablen auf der Anzeige bzw. dem Drucker aus.

Der Benutzer kann das Ausgabeformat frei bestimmen.

Mit INPUT kann einer Variablen während des Programmablaufs über die Tastatur ein Wert zugewiesen werden.

STOP oder END beenden den Programmablauf.

Es gibt eine Reihe von Funktionen:

Die bekannten mathematischen Funktionen SIN, COS, TAN etc., zusätzliche arithmetische Funktionen INT, ABS, RND etc. und außerdem Textfunktionen MIDS, LEFTS, RIGHTS etc.

Hinzu kommen Anweisungen zur Steuerung von Anzeige, Drucker und Magnetband.

DIM wird benutzt, um dimensionierte Variablen (Arrays oder Matrizen) zu definieren. REM kennzeichnet eine Kommentarzeile.



(Im Kap. 15 finden Sie eine ausführliche Beschreibung der Instruktionen).

Mit der Instruktion RUN wird die Ausführung des Programms gestartet. LIST (bzw. LLIST) erlaubt das Auflisten des Programms auf der Anzeige, bzw. dem Drucker.

14.2 Programmaufbau

Ein BASIC-Programm besteht aus einer Reihe von Zeilen, die eine oder mehrere BASIC-Anweisungen (Instruktionen) enthalten. Die einzelnen Anweisungen innerhalb einer Zeile werden durch einen Doppelpunkt (:) getrennt.

Jede Zeile kann bis zu 79 Zeichen enthalten. 26 Zeichen werden auf der Anzeige angezeigt.

Durch Bewegungen des Cursors mit den Tasten  und  läßt sich dieses Fenster von 26 Zeichen über die Programmzeile verschieben.

Am Anfang jeder Zeile steht eine Zeilennummer (ganze Zahl zwischen 1 und 65297). Die Programmzeilen werden vom Rechner in aufsteigender Reihenfolge ausgeführt, wenn nicht durch Steuerinstruktionen eine andere Reihenfolge programmiert wird. Die Numerierung muß nicht lückenlos sein, es ist sogar zweckmäßig z.B. Zehnerschritte zu wählen, damit später weitere Zeilen in das Programm eingefügt werden können.



Eingabe eines Programmes:

Die Eingabe eines BASIC-Programmes erfolgt im PRO-Mode. Ein eventuelle noch im Speicher befindliches altes Programm wird mit dem Kommando NEW gelöscht. Danach können die neuen Programmzeilen eingetippt werden.



Beispiel:

NEW

```
10  A = 15
20  B = 3
30  C = A + B
40  PRINT C
50  END
```

Jede Zeile wird mit  abgeschlossen. Nach dem  fügt der Rechner einen Doppelpunkt zwischen der Zeilennummer und der ersten Instruktion ein.

14.3 Überprüfen des Programms und Auflisten

Nachdem Sie Ihr Programm eingegeben haben, können Sie im PRO-Mode mit den Tasten  und  jeweils die vorherige oder nächste Programmzeile auf die Anzeige zurückholen und überprüfen. Durch das Kommando LIST kann die erste Zeile oder durch LIST < Zeilennummer > eine bestimmte Zeile angezeigt werden.

Beispiel:

Durch LIST 30 erscheint auf der Anzeige

30 : C = A + B



40 : PRINT C




14.4 Korrektur einer Zeile

Man kann eine Zeile im PRO-Mode auf zwei verschiedene Arten ändern:


Überschreiben:

- a) Man tippt eine neue Zeile mit der gleichen Nummer ein. Die alte Zeile wird dadurch ersetzt.

Modifizieren:

- b) Man holt eine Zeile auf die Anzeige so wie bei der Programmüberprüfung und betätigt die Cursortaste  oder . Der Doppelpunkt hinter der Zeilennummer verschwindet dann, und man kann die Zeile so verändern, als hätte man sie gerade eingegeben (s. auch S. 42). Die Änderung muß dann wieder mit  abgeschlossen werden.


Beispiel: In dem oben angeführten Beispielprogramm soll in Zeile 10 A = 17 gesetzt werden:

LIST 10  (Zeile auf die Anzeige holen)

 (Beginn Änderung)

   (Zeiger auf die 5)

7 

 RUN  (Programm Start)

14.5 Löschen einer Zeile

Eine Zeile wird im PRO-Mode gelöscht durch

< Zeilennummer > 

14.6 Programmausführung

Zum Ausführen des Programms wird in den RUN-Mode umgeschaltet und das Kommando RUN **[ENTER]** eingetippt. Der Rechner beginnt mit der Bearbeitung des Programms. Während der Programmausführung ist auf der Anzeige das Wort "BUSY" (= beschäftigt) sichtbar.

Dies erlischt mit dem Programmende und das Bereitschaftssymbol > erscheint wieder auf der Anzeige.

Wollen Sie das Programm vorzeitig abbrechen, können Sie das jederzeit durch Drücken der **BREAK**-Taste.

Auf der Anzeige erscheint die Meldung

BREAK IN <Zeilennummer>

BREAK IN 40

Die Ausführung kann durch das Kommando CONT fortgesetzt werden.

14.7 Fehlermeldung / Fehlersuche

Erkennt der PC-1500 während der Programmausführung einen Fehler, so bricht er die Bearbeitung ab und zeigt auf der Anzeige die Fehlermeldung in der Form

ERROR <Fehlernummer> IN <Zeilennummer>

<Zeilennummer> gibt die Nummer der Zeile an, in der die Fehler bemerkt wurden. Durch Drücken der Taste **[↑]** wird diese Zeile angezeigt. Der Cursor zeigt das fehlerhafte Sprachelement an.

Durch die <Fehlernummer> wird der Grund der Unterbrechung angegeben. Im Anhang befindet sich eine Liste der Fehlernummern und ihre Erklärung.

Die Fehlermeldung muß durch die Taste **[CL]** gelöscht werden. Danach kann man in den Programm-Mode umschalten und den Fehler beheben.

Beispiel:

```
10  A = 0
20  B = 10
30  C = B/A
40  PRINT C
50  END
```

Bei Ausführung dieses Programms erscheint

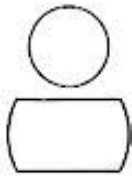
ERROR 38 IN 30

Division durch 0

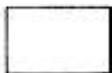
15. PC-1500 BASIC

Das folgende Kapitel beschreibt ausführlich den BASIC Sprachumfang des PC-1500.

Zum leichteren Verständnis wird die Syntax als Graph angegeben. Folgt man beginnend von links nach rechts den Pfeilen des Graphen, erhält man die korrekte Syntaxvariante der einzelnen Instruktionen. Die Wirkung wird jeweils im Anschluß erklärt.



Die eingeschlossenen Zeichen sind Schlüsselworte oder Schlüssel- bzw. Operationszeichen und müssen in dieser Form in das Programm übernommen werden.



Der eingeschlossene Ausdruck beschreibt ein vom Benutzer definiertes Sprachelement und ist im Programm durch das konkrete zu ersetzen.



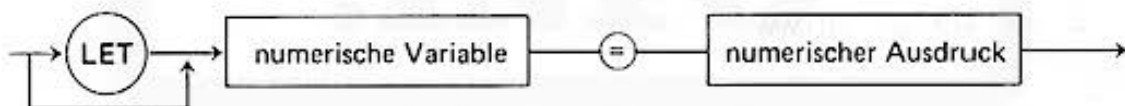
In spitzen Klammern wird die Bedeutung des Sprachelementes angegeben.

Beispiel:

Die LET-Anweisung hat z.B. folgendes Aussehen:

<u>LET</u>	<u>A</u>	=	<u>2 * (B + C)</u>
Schlüsselwort	Variablenname		numerischer Ausdruck

Die Syntax wird dann folgendermaßen als Graph dargestellt:



Um das Schlüsselwort LET gibt es im Graphen eine Umleitung; daher ist die Anweisung

$A = 2 * (B + C)$

ebenfalls korrekt.

Anmerkung

Die **ENTER** Taste muß nach jeder Programmzeile eingegeben werden, obwohl sie im Graphen fortgelassen wird.

16. Sprachelemente

16.1 Numerische Konstante

Ganze Zahl, gebrochene Zahl, Zahl in wissenschaftlicher Darstellung, Hexadezimalzahl.

Beispiele:

513
-256
11.347
-1.234567891 E-12
&FFFF
&1A2B
&1

16.2 Textkonstante

Beliebige Zeichenfolge eingeschlossen in Anführungsstriche.

Beispiel:

"Juhu"
" " (Textkonstante enthält Leerzeichen)
"" (Textkonstante der Länge 0)

16.3 Numerische Variable

Speicherplatz für einen Zahlenwert, auf den über den Variablennamen zugegriffen wird. Der Variablenname besteht aus einem Buchstaben, zwei Buchstaben oder einem Buchstaben und einer Ziffer. Mehr als zwei Zeichen sind erlaubt, solange kein BASIC-Schlüsselwort entsteht. Diese zusätzlichen Zeichen werden vom Rechner ignoriert. Für Variablennamen dürfen nur Großbuchstaben verwendet werden.

Beispiel:

A, AA, A0, BX, NUMM

Ausnahme:

Nicht erlaubt sind die Namen IF, TO, PI, LF, LN

16.4 Textvariable

Speicherplatz für eine Zeichenfolge, auf den über den Variablennamen zugegriffen wird. Der Name hat die gleiche Form wie ein numerischer Variablenname mit einem zusätzlichen \$ am Ende. Eine Textvariable kann maximal 16 Zeichen enthalten. Benötigt man eine größere Zeichenkapazität, kann man ein Textfeld benutzen (s. DIM).

Beispiel:

AS\$, AAS\$, AØ\$, BXS\$, TEXTS\$

Ausnahme:

Nicht erlaubt sind die Namen

IFS\$, TOS\$, PIS\$, LFS\$, LNS\$

16.5 Numerische oder Textvariablen,

deren Name nur einen Buchstaben enthält, sind in einem speziellen Bereich gespeichert. (s. Speicherorganisation). Alle anderen Variablen werden bei ihrer ersten Benutzung erzeugt. Ihr Anfangswert ist 0 bzw. bei Textvariablen das ASCII-Nullzeichen.

16.6 Numerische Funktion, Textfunktionen

Eine Funktion ist ein Operator auf ein oder mehrere Parameter und hat einen Zahlenwert oder eine Zeichenfolge als Ergebnis. Parameter werden hinter dem Funktionsnamen angegeben und können je nach Funktion numerische oder Textausdrücke sein. Sind mehrere Parameter erforderlich, so werden diese in Klammern gesetzt und durch Kommata getrennt.

Beispiele für numerische Funktionen:

Ergebnis:

LN 60

4.094344562

ASC "A"

65

Textfunktionen

CHRS 65

A

MIDS ("JUHU", 2,2)

UH

Hinweis: Denken Sie an die Priorität des Operatoren. Funktionsoperatoren haben eine höhere Priorität als alle anderen Operatoren.

Beispiel:

SIN A + B = (SIN A) + B

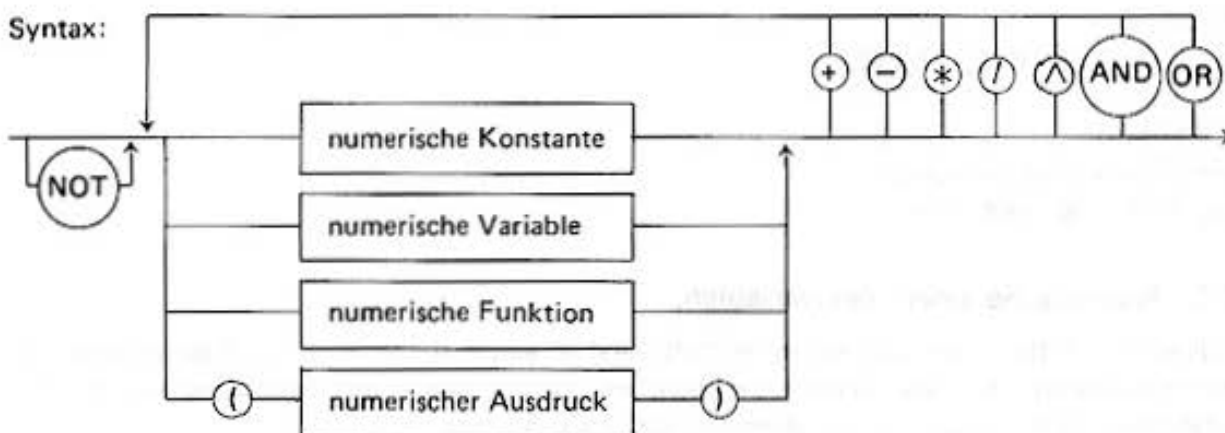
Soll der Sinus der Summen gebildet werden, benutzen Sie die Klammern:

SIN (A + B)

16.7 Numerischer Ausdruck

Ein numerischer Ausdruck besteht aus einer numerischen Konstante-numerischen Variablen oder numerischen Funktionen oder deren Verknüpfung durch die arithmetischen Operatoren $+$, $-$, $/$, $*$, $^$, AND, OR, NOT und die Zusammenfassung durch Klammern. Das Ergebnis eines numerischen Ausdrucks ist ein Zahlenwert.

Syntax:



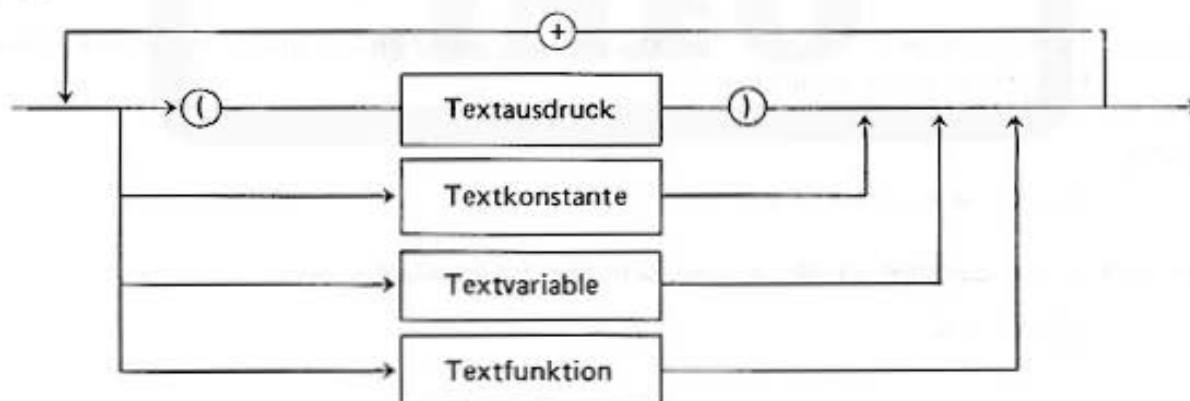
Beispiel:

15
 (-7)
 SIN 45
 A + B/C
 ((A * X + B) * X + C) * X + D

16.8 Textausdruck

Besteht aus einer Text-Konstante, Textvariablen oder einer Textfunktion oder deren Verknüpfung durch das Zeichen $+$. Das Ergebnis eines Textausdruckes ist eine Zeichenfolge.

Syntax:



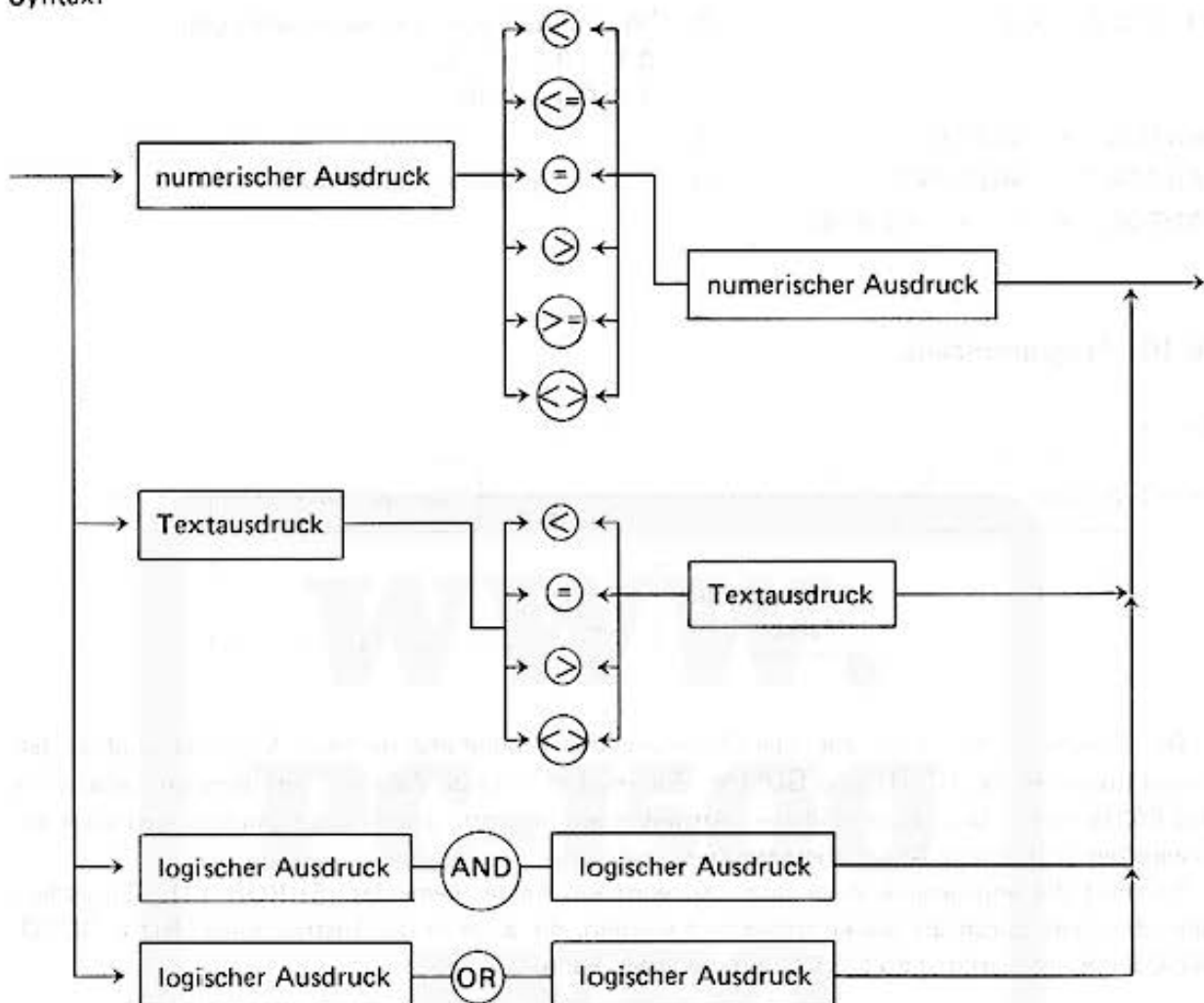
Beispiel:

"AB = " + STR\$ AB
 (A\$ + "PC")

16.9 Logischer Ausdruck

Ein logischer Ausdruck ist ein Vergleich zweier numerischer Ausdrücke durch die Operatoren $<$, $<=$, $=$, $>=$, $>$, $<>$ oder die Verknüpfung solcher Vergleiche durch die Operatoren AND, OR, NOT.

Syntax:



Das Ergebnis eines logischen Ausdrucks ist eine 1, wenn die Vergleichsaussage richtig ist, sonst eine 0. Dieses Ergebnis kann in numerischen Variablen gespeichert werden.

Werden Textausdrücke miteinander verglichen, so ist unter der "Größe" die lexikalische Reihenfolge zu verstehen, d.h. "A" ist kleiner als "Z".

Die Reihenfolge der Zeichen ist aus der ASCII-Code-Tabelle im Anhang zu ersehen. Haben zwei Textausdrücke unterschiedliche Länge, so wird bei einem Vergleich der kürzere Ausdruck mit dem Zeichen ASCII-Null aufgefüllt.

Logische Ausdrücke werden in der BASIC-Anweisung IF verwandt.

Beispiele:

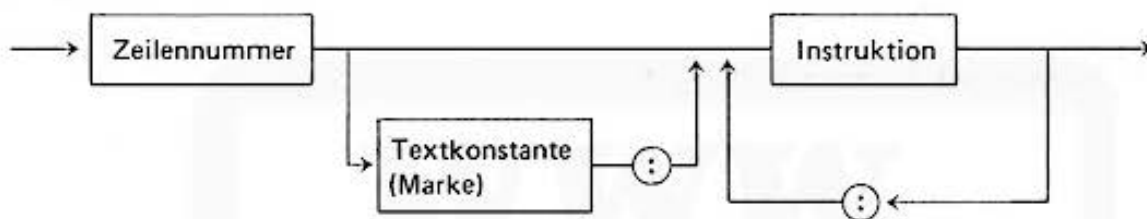
1 < 2
1 + 2 + 3 < 2 + 3 + 4
1 >= 2
1 < 2 < 3
0.1 < 0.2 < 0.3

Ergebnis:

1
1
0
1
0 (Ausführung von links nach rechts gibt
0.1 < 0.2 => 1;
1 < 0.3 => 0)
1
0 ("1" ist größer als ASCII-Null)
1

16.10 Programmzeile:

Syntax:



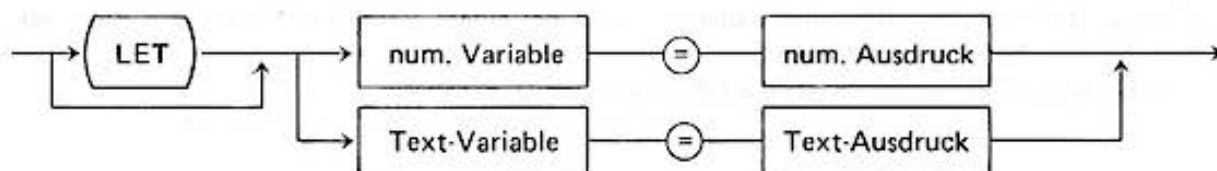
Die Zeilennummer dient zur Identifizierung der Programmzeile beim Editieren und in den Instruktionen RUN, GOTO und GOSUB. Sie muß eine ganze Zahl aus dem Bereich zwischen 1 und 65279 sein. Gibt man in diesen Anweisungen als Sprungziel keinen ganzzahligen Wert an, so wird der gebrochene Anteil abgeschnitten.

Existiert die angegebene Zeile nicht, so wird ein Fehler gemeldet (ERROR 11). Zusätzlich kann die Zeile durch die Marke bezeichnet werden, die auch in den Instruktionen RUN, GOTO, GOSUB und im Kommando LIST benutzt werden kann.

LET

Diese Instruktion weist einer Variablen einen Wert zu.

Syntax:



Mit der LET-Instruktion wird der Wert des numerischen Ausdrucks berechnet bzw. der Textausdruck zusammengestellt und in der numerischen bzw. Textvariablen abgelegt.

Anmerkung:

Das Schlüsselwort LET ist in der ursprünglichen Sprachdefinition von BASIC enthalten. Beim PC-1500 BASIC darf es fortgelassen werden. Wenn es nicht innerhalb einer IF-Instruktion verwendet wird.

Besteht der Variablenname aus mindestens 2 Zeichen, so wird beim ersten Auftreten des Namens im Programmablauf der Speicherplatz für diese Variable reserviert (siehe Speicherorganisation). Ist kein Speicherplatz mehr frei, führt dies zu einem Fehler (ERROR 10).

Beispiele:

- | | |
|----------------------|---|
| (1) 10 LET A = 3.14 | die numerische Konstante 3.14 wird der Variablen A zugewiesen |
| (2) 10 I = I + 1 | der numerische Ausdruck I + 1 wird berechnet und der Variablen I zugewiesen |
| (3) 10 Y = 3 * COS X | der numerische Ausdruck 3 * COS X wird berechnet und der Variablen Y zugewiesen |

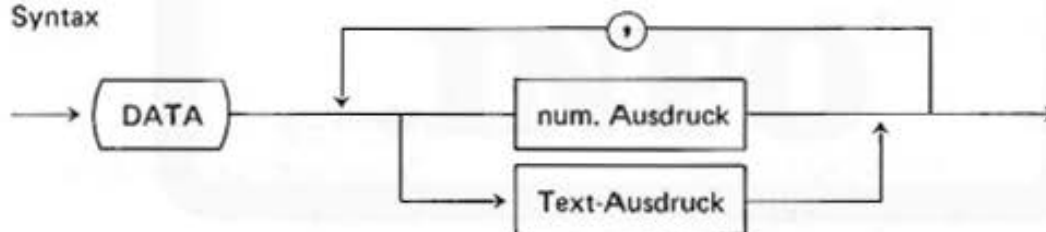
In den Beispielen (2, 3) wurde jeweils das Schlüsselwort LET fortgelassen

- | | |
|-------------------------|---|
| (4) 10 LET AS = "PARIS" | die Textkonstante "PARIS" wird der Stringvariablen AS zugewiesen |
| (5) 10 FRAG\$ = "Was?" | die Textkonstante "Was?" wird der Textvariablen FRAG\$ zugewiesen |

DATA

Diese Instruktion definiert Datenfelder.

Syntax



Die DATA-Instruktion gewinnt ihre Bedeutung erst mit der READ-Anweisung.

Es werden beim READ Variablenwerte aus Datenfeldern zugewiesen. Alle DATA-Instruktionen werden in der Reihenfolge ihrer Anweisungsnummern behandelt und können in beliebiger Anzahl an beliebiger Stelle im Programm auftreten. Sie werden bei der Programmausführung automatisch übersprungen.

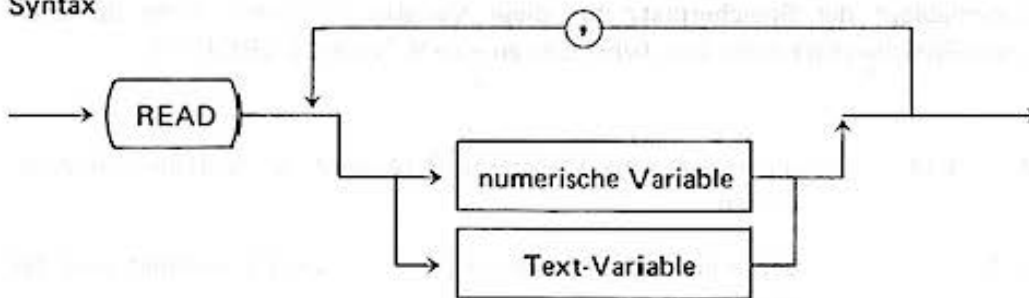
Beispiele:

```

10 DATA 12, 13, 27
20 DATA "TEXTE", "KOELN"
30 DATA 1, "BEI", "SPIEL"
40 DATA A + B, X/Y
    
```

READ

Diese Instruktion weist Variablen Werte aus Datenfeldern zu.

Syntax

Enthält die READ-Anweisung mehrere Variablen, so hat diese die gleiche Wirkung wie eine Folge von getrennten READ-Instruktionen.

Mit der ersten READ-Anweisung wird der erste Wert aus dem Datenfeld der ersten DATA-Instruktion eingelesen und der Variablen zugewiesen.

Bei der zweiten READ-Instruktion wird der Variablen der zweite Wert aus der gleichen DATA-Anweisung zugewiesen (sofern einer vorhanden ist). Bei den folgenden READ-Anweisungen wird der dritte, vierte usw. Wert zugewiesen. Sind alle Daten der ersten DATA-Anweisung verarbeitet, werden sie automatisch von dem Datenfeld der zweiten, danach von der dritten usw. DATA-Anweisung gelesen usw.

Ist in der DATA-Anweisung ein arithmetischer Ausdruck angegeben, so wird dessen Wert berechnet und der Variablen zugewiesen. Die Variablen der READ-Instruktion und die Datentypen in der DATA-Anweisung müssen zueinander passen, d.h. zu numerischen Variablen numerische Ausdrücke bzw. zu Textvariablen Textausdrücke.

Beispiele:

```
(1) 10 READ Y1, Y2
    20 READ A, B, SS, TS
```

```
(3) 05 CSIZE 1
    10 REM TELEFONVERZEICHNIS
    20 LPRINT "NAME", "TEL."
    30 READ MS
    40 IF MS <> "0" THEN GOTO 70
    50 LPRINT "ENDE DES VERZEICHNISSES"
    60 END
    70 READ NS, TS
    80 IF MS = "M" THEN LPRINT NS, TS
    90 GOTO 30
    100 DATA "M", "MARCUS", "040-441777"
    110 DATA "F", "ARZT", "089-123456"
    120 DATA "M", "FEUER", "112"
    130 DATA "0"
```

Bemerkung

Zeichengröße auf dem Drucker
Ausdruck

Abfrage auf das Ende des
Datenfeldes

Datenfelder

(Falls Sie keinen Drucker haben, verwenden Sie PRINT anstelle von LPRINT)

Die Ausführung des Programmes liefert folgendes Ergebnis:

```

5:CSIZE 1
10:REM TELEFONVERZEICHNIS
20:LPRINT "NAME", "TEL."
30:READ M#
40:IF M# < 0 THEN GOTO 70
50:LPRINT "ENDE DES VERZEICHNISSES"
60:END
70:READ N#, T#
80:IF M# = "M" THEN LPRINT N#, T#
90:GOTO 30
100:DATA "M", "MARCUS", "040-441777"
110:DATA "C", "GRZT", "889-123456"
120:DATA "M", "FEUER", "112"
130:DATA "0"
    
```

```

NAME          TEL.
MARCUS        040-441777
FEUER         112
ENDE DES VERZEICHNISSES
    
```

(2)

```

10  READ  A, B
20  READ  C
30  LPRINT A; B; C
40  END
100  DATA 65
110  DATA 66
120  DATA 67
    
```

Das Ergebnis dieses Programmes lautet:

65 66 67

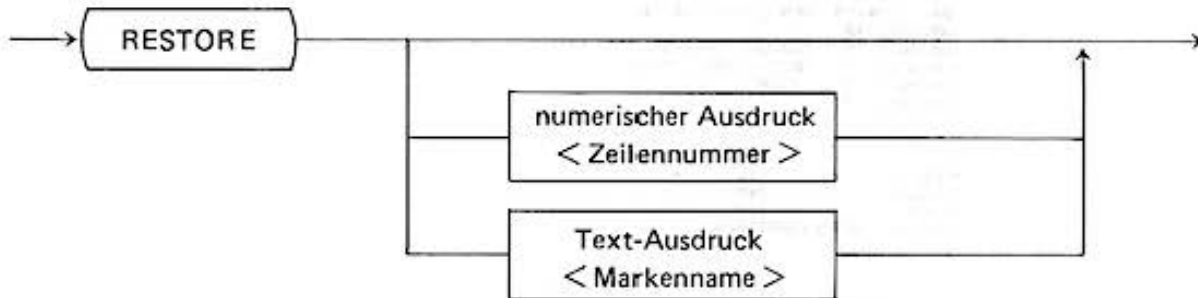
Bemerkungen:

1. Nicht benutzte Daten werden ignoriert.
2. Fehlt eine DATA-Instruktion oder sind zu wenige Daten definiert, so führt das beim READ zu einem Fehler (ERROR 2)
3. Die READ-Instruktion ist der LET-Instruktion verwandt, bietet dem Programmierer durch komplexere Struktur mehr Möglichkeiten.

RESTORE

Diese Instruktion spezifiziert jene DATA-Anweisung, von der mit dem nächsten READ gelesen wird.

Syntax



Der Wert des numerischen Ausdrucks spezifiziert die Zeilennummer, der Textausdruck die Marke einer DATA-Anweisung. Bei der im Programmablauf folgenden READ-Instruktion werden die Werte ab der spezifizierten DATA-Anweisung eingelesen.

Läßt man die Ausdrücke fort, wird die erste mit der kleinsten Zeilennummer versehene DATA-Instruktion bei der darauffolgenden READ-Anweisung benutzt.

Beispiel:

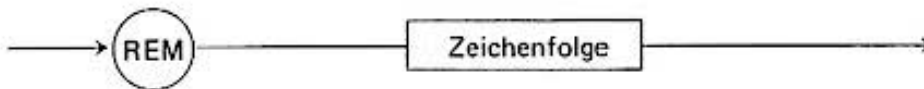
```

(1)  10  READ X      ← 1
      20  RESTORE
      30  READ Y      ← 1
      40  READ Z      ← 2
      50  LPRINT X, Y, Z
      60  END
      100 DATA 1,2
    
```

REM

Diese Instruktion definiert eine Kommentarzeile im Programm.

Syntax



Die REM-Instruktion wird bei der Programmausführung ignoriert. Die Anweisung kann an beliebiger Stelle im Programm erscheinen und definiert den Rest der Zeile als Kommentar.

Hinweis:

- (1) Die REM-Instruktion interpretiert die gesamte restliche Zeile als Kommentar.
 10 REM INITIALISIERUNG : A = 0
 In diesem Fall wird die Instruktion A = 0 als Kommentar aufgefaßt und nie ausgeführt.
- (2) Es ist empfehlenswert mindestens vor größeren zusammenhängenden Programmabschnitten, die Aufgabe jenes Programmteiles zu erläutern.

Beispiele:

- (1)


```

10 REM *****
11 REM *
12 REM * PROGRAMM ZUR BERECHNUNG *
13 REM * DER EINKOMMENSTEUER *
14 REM *
15 REM *****
      
```
- (2)


```

10 READ P : REM PREIS PER STUECK
20 READ N : REM ANZAHL
30 C = P * N
      
```

STOP

Diese Instruktion hält die Programmausführung an.

Syntax



Auf der Anzeige erscheint bei der Ausführung dieser Instruktion:

BREAK IN < Zeilennummer >

Nach Bestätigung dieser Meldung durch Eingabe von **ENTER** erscheint das Bereitschaftszeichen und man kann interaktiv mit dem PC-1500 arbeiten. Es lassen sich beispielsweise die Werte von Variablen anschauen oder Programme nochmal überprüfen.

Mit dem Kommando CONT wird die Ausführung mit der auf STOP folgenden Instruktion fortgesetzt.

Anmerkung:

Drückt man die **↑** Taste, so erscheint die gerade ausgeführte Programmzeile auf der Anzeige.

Die **↓** Taste hat die Wirkung, daß das Programm zeilenweise fortgesetzt wird.

Beispiel:

```
10 INPUT A, B, C
20 M = 3 * A ^ 4
30 N = B/5 - 10
40 STOP
50 L = C * SQR (M-N)
:
```

END

Diese Instruktion beendet die Programmausführung.

Syntax



Diese Instruktion steht gewöhnlich am Ende eines Programmes und beendet die Ausführung. Mit dieser Instruktion kann der Programmablauf an einer beliebigen Stelle im Programm beendet werden.

Anmerkung:

Man kann die END-Instruktion fortlassen. Das Programm wird automatisch nach Ausführung der letzten Zeile beendet.

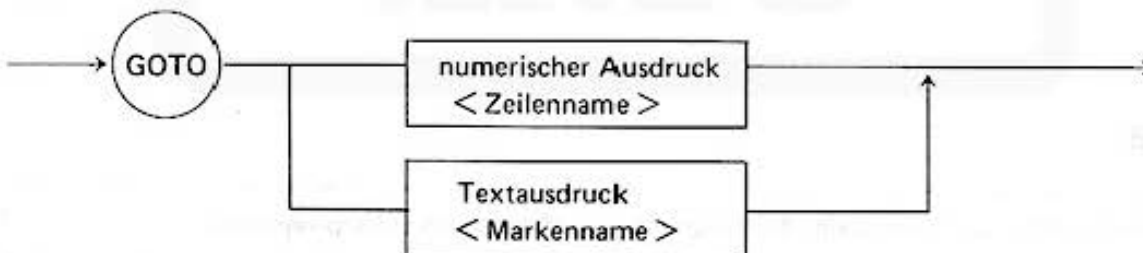
Beispiel:

```
10 PRINT "START"  
20 FOR I = 1 TO 20  
30 PRINT I  
35 IF I = 10 THEN END  
40 NEXT I
```

GOTO

Mit dieser Instruktion lenkt man die Programmausführung zu der spezifizierten Programmzeile.

Syntax



Die Sprunganweisung GOTO lenkt den Programmlauf ohne jede Einschränkung zu einer bestimmten Programmzeile. Der numerische Ausdruck spezifiziert eine Anweisungsnummer (ganzzahliger Anteil), der Textausdruck (16 Zeilen gültig) einen Markennamen als Sprungziel.

Beispiele:

- (1) 50 GOTO 100
- (2) 50 GOTO "ROT"
- (3) 50 AS = "ROT", : GOTO AS
- (4) 50 GOTO 10 ^ 2
- (5) 50 A = 30 : B = 20 : GOTO 2 * (A + B)

⋮

100 "ROT" : PRINT "HIER ZEILE 100 ROT"

- (6) Dieses Programm berechnet die Quadrat-, Kubikzahlen zwischen 1 und 10

```

10 A = 1
20 LPRINT A; A ^ 2; A ^ 3
30 A = A + 1
40 IF A > 10 THEN GOTO 60
50 GOTO 20
60 END
    
```

Bemerkung:

Diese Zeilen 50 haben alle die gleiche Wirkung

Das Programm liefert folgendes Ergebnis:

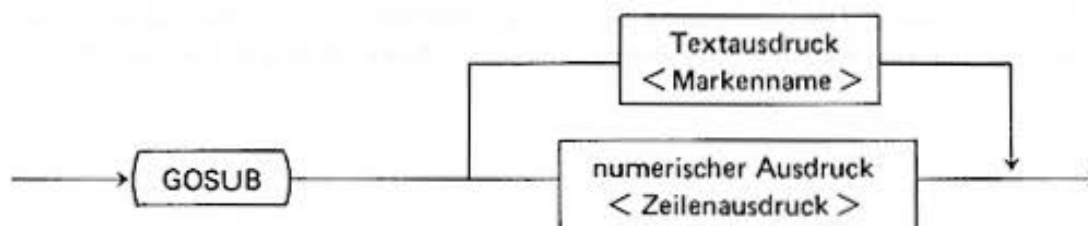
```

1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
    
```

GOSUB

Mit dieser Instruktion verzweigt der Programmablauf zu einem Unterprogramm.

Syntax



Ein häufig benutzter Programmteil wird meist als Unterprogramm geschrieben. Die GOSUB-Instruktion verzweigt den Programmablauf zu jenem Unterprogramm, das mit der Programmzeile beginnt, die durch den numerischen Ausdruck, bzw. Markennamen spezifiziert ist (analog zum GOTO).

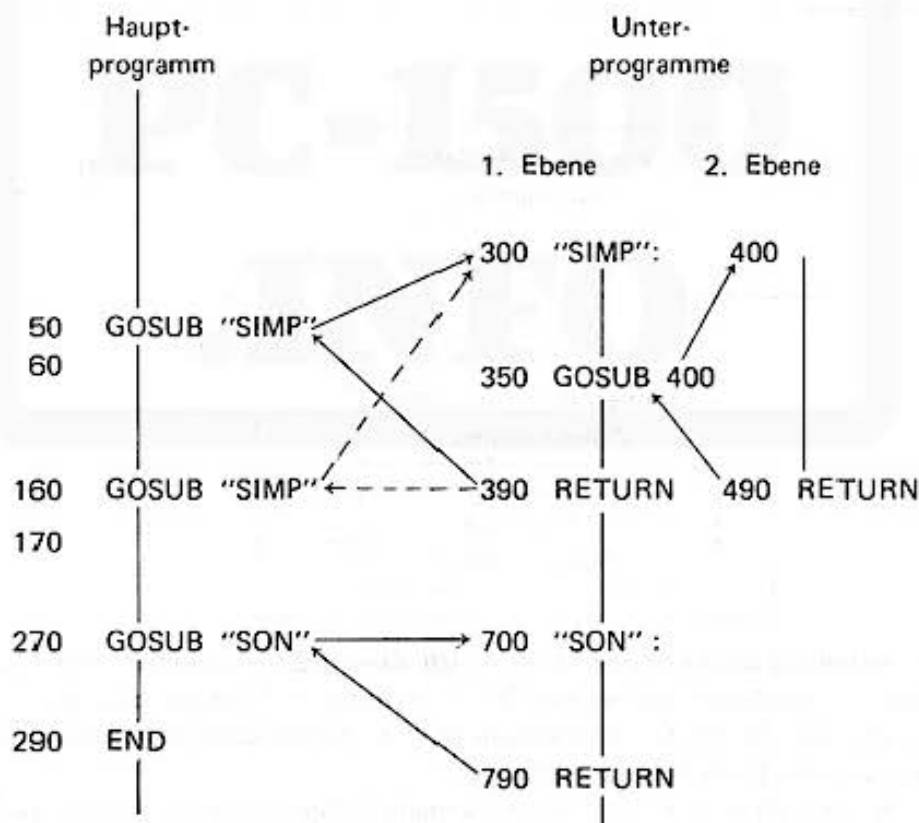
Die Ausführung des Unterprogramms wird durch die RETURN-Instruktion abgeschlossen und mit der auf die GOSUB folgenden Instruktion fortgeführt.

Die GOSUB-Anweisung kann an beliebiger Stelle im Programm stehen und es können mehrere Unterprogrammaufrufe geschachtelt erfolgen (max. 32).

Beispiele:

```
(1)  50  GOSUB 300
(2)  50  GOSUB I + J
(3)  50  GOSUB "SIMPSON"
(4)  50  A$ = "SUB" : GOSUB A$
      ⋮
    100  END
      ⋮
    300
      ⋮
    400  "SIMPSON" :
```

(5)

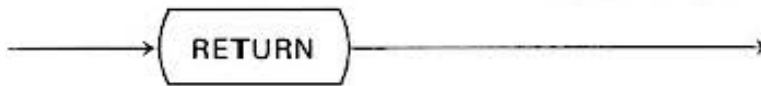


Diese Graphik zeigt die Verwendung von mehreren Unterprogrammen. Zeile 50 und 160 rufen das Unterprogramm "SIMP" auf. Dieses wiederum ruft in der nächsten Ebene das Unterprogramm beginnend mit Zeile 400 auf. Der Rücksprung in Zeile 390 aus dem "SIMP" Unterprogramm erfolgt in Abhängigkeit vom Aufrufort, also zur Zeile, die auf Programmzeile 50 bzw. 160 folgt.

RETURN

Diese Instruktion beendet ein Unterprogramm und setzt die Ausführung mit der Instruktion fort, die auf die GOSUB-Anweisung folgt, von der das Unterprogramm aufgerufen wurde.

Syntax



Bemerkungen:

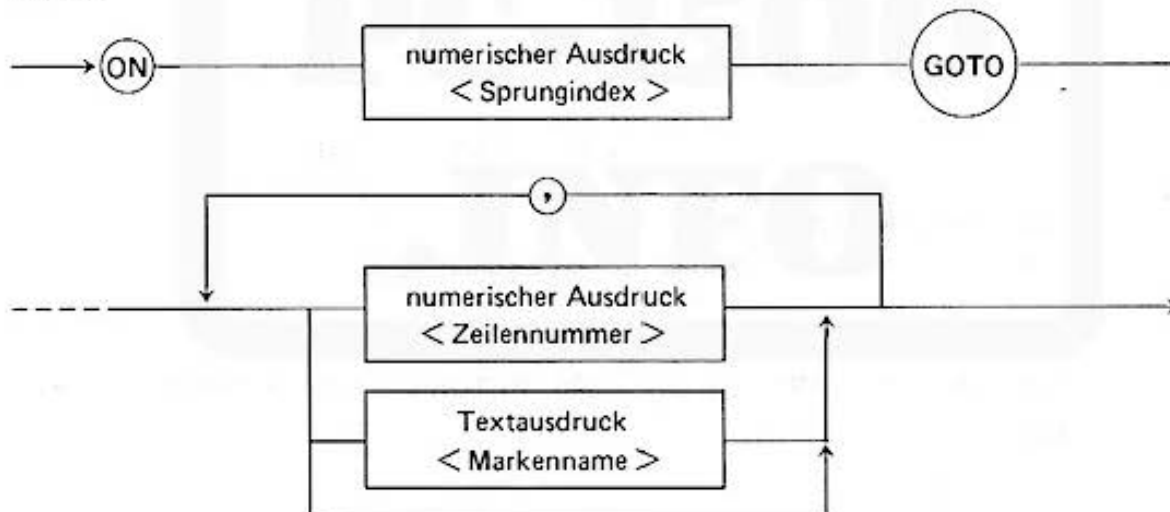
- (1) Ein Unterprogramm muß mit der RETURN-Instruktion beendet werden (nicht GOTO).
- (2) Eine RETURN-Instruktion im Hauptprogramm führt zu einem Fehler (ERROR 2).

Beispiel: siehe GOSUB-Instruktion

ON GOTO

Diese Instruktion verzweigt die Programmausführung abhängig von dem Wert eines numerischen Ausdrucks und der Sprungzielliste.

Syntax



Die Liste, bestehend aus den numerischen Ausdrücken < Zeilennummer > und Textausdrücken < Markenname >, spezifiziert wie bei der GOTO-Instruktion Anweisungsnummern bzw. Markennamen als Sprungziele. Durch den ganzzahligen Wert des numerischen Ausdrucks < Sprungindex > wird festgelegt, welches Element auszuwählen ist.

Hat der Index den Wert 1, wird das erste Element als Sprungziel ausgewählt; beim Wert 2 das zweite usw. beim Wert i das i-te Element.

Beim Wert 0 vom Ausdruck < Sprungindex > findet kein Sprung statt, die Programmausführung wird mit der auf ON . . . GOTO folgenden Instruktion fortgesetzt. Das gleiche ist gültig bei allen anderen Indizes für die kein zugehöriges Listenelement vorhanden ist.

Beispiele:

```
(1)  50 : ON I GOTO 100, 200, 300
      60 : PRINT I
```

In diesem Beispiel kann I die Werte 0, 1, 2, 3 annehmen; hat der Index I den Wert 1, so verzweigt der Programmlauf zu Zeile 100, bei I = 2 zur Zeile 200 usw. Hat I den Wert 0 wird sofort Zeile 60 ausgeführt, ebenso bei allen anderen Werten von I.

```
(2)  10  FOR I = -2 TO 5
      15  PRINT I;
      20  ON I GOTO 50, 30, 60, 70
      30  PRINT " KEIN SPRUNG"
      40  GOTO 200
      50  PRINT " SPRUNG NACH 50"
      55  GOTO 200
      60  PRINT " SPRUNG NACH 60"
      65  GOTO 200
      70  PRINT " SPRUNG NACH 70"
      200 NEXT I
```

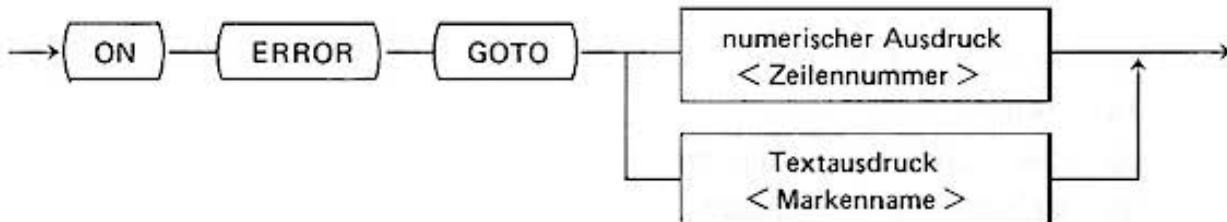
Ergebnis:

```
-2  KEIN SPRUNG
-1  KEIN SPRUNG
 0  KEIN SPRUNG
 1  SPRUNG NACH 50
 2  KEIN SPRUNG
 3  SPRUNG NACH 60
 4  SPRUNG NACH 70
 5  KEIN SPRUNG
```

ON ERROR GOTO

Mit dieser Instruktion verzweigt die Programmausführung zu der spezifizierten Programmzeile, wenn im Ablauf ein Fehler erkannt wird.

Syntax



Die Instruktion hat direkt in der Programmausführung unmittelbar keine Wirkung. Erst, wenn im Ablauf des Programmes ein Fehler auftritt, wird das Programm nicht wie sonst abgebrochen, sondern zu dem spezifizierten Sprungziel verzweigt.

Bemerkungen:

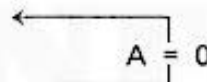
- (1) ON ERROR-Instruktionen können nicht geschachtelt werden. Die zuletzt ausgeführte ist gültig.
- (2) Die standardmäßige Fehlerbehandlung wird wieder eingeschaltet, wenn der numerische Ausdruck den Wert 0 hat. (ON ERROR GOTO 0)
- (3) Hat der auslösende Fehler den Code '1' (Syntax Fehler) '7', '32', so wird die ON ERROR-Instruktion ignoriert, das Programm wie normal abgebrochen und der Fehler angezeigt.

Beispiel:

```

10  ON ERROR GOTO 20
20  INPUT A
30  B = 1/A
40  PRINT B

```



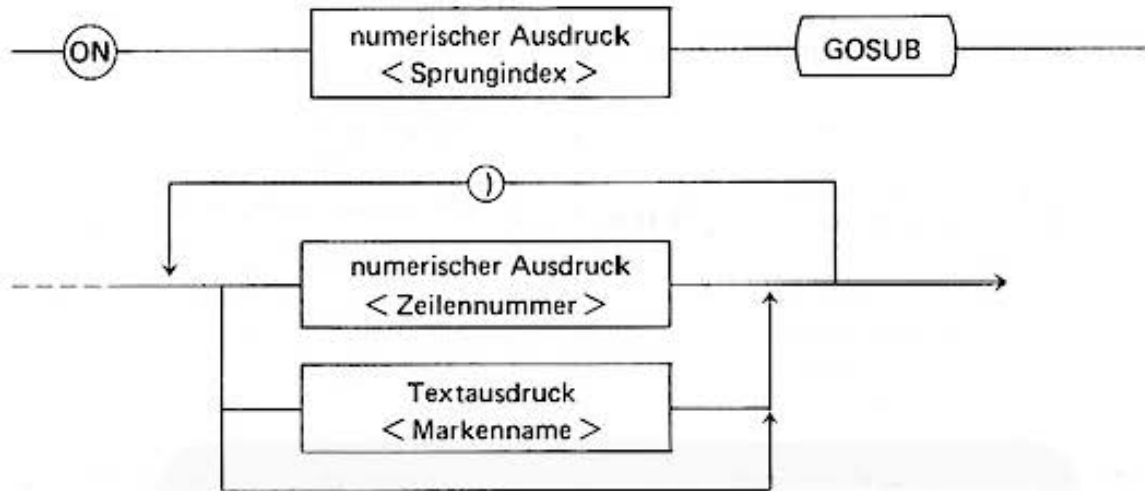
Wird für A der Wert 0 eingegeben, so führt dies in Zeile 30 zu einem Fehler.

Durch die ON ERROR-Instruktion verzweigt das Programm zu Zeile 20 und die Eingabe kann wiederholt werden.

ON GOSUB

Diese Instruktion ruft ein Unterprogramm der Unterprogrammliste abhängig vom Wert des numerischen Ausdrucks auf.

Syntax



Die Programmausführung verzweigt zu jenem Unterprogramm, das aus der Zielliste entsprechend dem Wert des numerischen Ausdrucks $\langle \text{Sprungindex} \rangle$ ausgewählt wird. Analog zur ON GOTO-Instruktion bestimmt der ganzzahlige Wert von $\langle \text{Sprungindex} \rangle$ das Listenelement. In der Liste repräsentieren die Ausdrücke Zeilennummern bzw. Marken.

Ist der Index negativ, 0, größer als die Anzahl der Elemente in der Zielliste oder das Element nicht spezifiziert, wird die nächste Instruktion ausgeführt.

Nach der Ausführung des Unterprogramms wird der Programmlauf mit der auf die ON GOSUB folgenden Instruktion fortgesetzt. (siehe GOSUB).

Beispiel:

```
10 ON (SGN (D) + 2) GOSUB "MINUS", "NULL", "PLUS"
```

SGN (D) kann die Werte -1, 0, 1 annehmen, jeweils für $D < 0$, $D = 0$, $D > 0$.

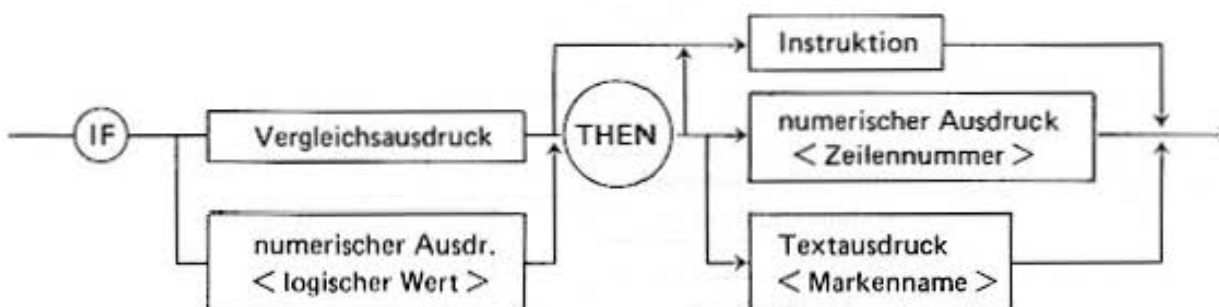
Damit erhält der SGN (D) + 2 die Werte 1, 2, 3 und entsprechend wird zum Unterprogramm "MINUS", "NULL", "PLUS" verzweigt.

IF THEN

Die auf IF folgende Bedingung wird geprüft. Ist sie wahr, so wird die auf THEN folgende Instruktion ausgeführt. Ist die Bedingung falsch, so wird die Programmzeile mit der nächsthöheren Zeilennummer ausgeführt.

Syntax

Die Syntax der IF THEN-Instruktion hat mehrere Formate.



Der Vergleichsausdruck bzw. der numerische Ausdruck <logischer Wert> ist die Bedingung, die geprüft wird. Im Fall des <log. Wert> wird die Bedingung als 'wahr' angesehen, wenn der Wert größer (>) 0 und als 'falsch', wenn er kleiner gleich (\leq) 0 ist. Der Vergleichsausdruck wird nach den aus der Mathematik bekannten Regeln der Operatoren <, <=, =, >=, <> bestimmt und liefert die Werte 'wahr' oder 'falsch'. Bei der Anwendung der Operatoren <, >, =, <> auf Zeichenketten bzw. Textvariablen wird auf die lexikalische Reihenfolge geprüft. Dabei werden nur die ersten 16 Zeichen beachtet.

Ist die Bedingung 'wahr', so werden die auf das THEN (optional) folgenden Instruktionen ausgeführt. Wird der numerische Ausdruck <Zeilennummer> bzw. der Textausdruck <Markenname> angegeben, so stellt dies eine verkürzte Schreibweise für eine GOTO-Instruktion dar.

Die Ausdrücke repräsentieren eine Zeilennummer bzw. einen Markennamen als Sprungziel. In diesem Fall muß das Sprachelement THEN verwendet werden.

Bemerkung:

Instruktionen, die in der gleichen Programmzeile auf die IF THEN-Instruktion folgen, werden nur ausgeführt, wenn die Bedingung des IFs wahr ist.

Beispiele (IF/THEN)

(1) 50 IF A = 10 THEN PRINT "A = 10"

Falls A = 10 erscheint "A = 10" auf der Anzeige

(2) 50 IF A\$ = "JA" THEN GOTO 200

Die Beispiele 2, 3, 4, haben die gleiche Wirkung

(3) 50 IF A\$ = "JA" GOTO 200

(4) 50 IF A\$ = "JA" THEN 200

Wenn die Variable A\$ den Wert "JA" hat, verzweigt das Programm zu Zeile 200

(5) 50 IF A\$ = "JA" THEN "TUWAS"

200 "TUWAS" : PRINT "LOS GENHTS"

(6) 30 X = 100

"
"

50 IF X THEN 200

Der Programmablauf wird bei 200 fortgesetzt, da $X > 0$

"
"

200 PRINT "HIER"

(7) 50 IF 100 < X < 200 THEN GOSUB "INTV"

"
"

1000 "INTV" : Intervallabarbeitung

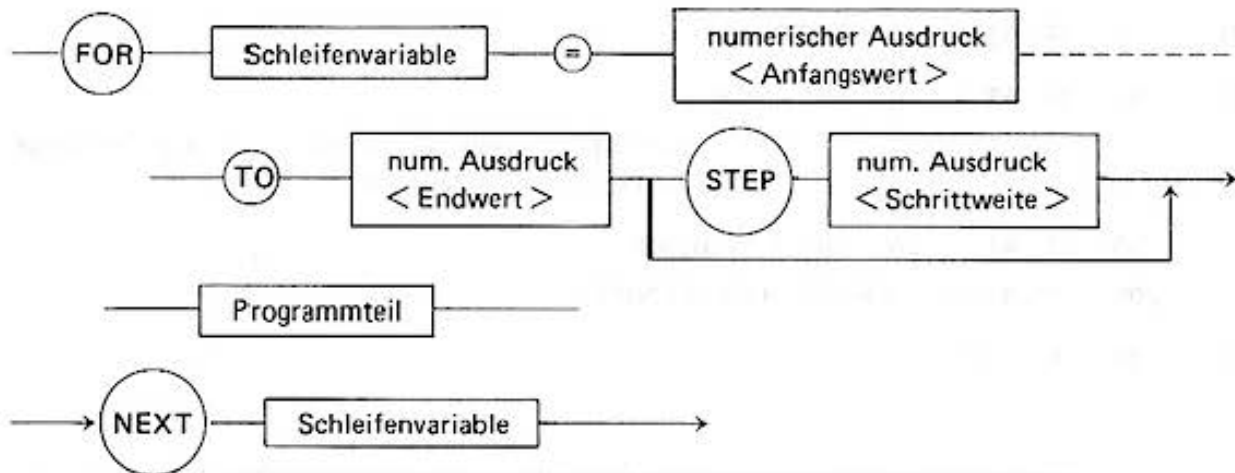
"
"

1950 RETURN

FOR . . . TO . . . STEP / NEXT

Der zwischen der FOR . . . TO und der NEXT-Instruktion eingeschlossene Programmteil wird entsprechend den Schleifenbedingungen mehrfach durchlaufen.

Syntax:



Die FOR . . . TO und die NEXT-Instruktion gehören zusammen. Sie schließen einen Programmteil aus beliebig vielen PC-1500 BASIC-Instruktionen ein, der in einer Schleife mehrfach durchlaufen werden soll. Bei beiden Anweisungen muß die gleiche Schleifenvariable verwendet werden, damit bei geschachtelten Schleifen eindeutig festgelegt ist, welche abgeschlossen wird. Die Schleifenvariable muß eine numerische Variable sein.

Die Werte müssen in dem Intervall $-32768 \dots 32767$ liegen.

Für die Schrittweite ist der Wert 0 nicht zulässig. (Endlosschleife!) (ERROR 19)

Läßt man in der allgemeinen Form den STEP-Teil fort, wird die Schrittweite gleich 1 gesetzt.

Vor dem ersten Durchlauf wird der Schleifenvariablen der Wert vom Ausdruck $\langle \text{Anfangswert} \rangle$ zugewiesen. Erreicht der Programmlauf die NEXT-Instruktion, so wird die $\langle \text{Schrittweite} \rangle$ zum Wert der Schleifenvariablen addiert und der Programmteil mit diesem Wert erneut durchlaufen. Dies wird fortgesetzt, solange der Wert der Schleifenvariablen innerhalb des Intervalles liegt, das festgelegt ist durch:

1) bei positiver Schrittweite

$$\langle \text{Anfangswert} \rangle \leq \langle \text{Schleifenvariable} \rangle < \langle \text{Endwert} \rangle + \langle \text{Schrittweite} \rangle$$

2) oder bei negativer Schrittweite

$$\langle \text{Schrittweite} \rangle + \langle \text{Endwert} \rangle < \langle \text{Schleifenvariable} \rangle \leq \langle \text{Anfangswert} \rangle$$

Das Programm wird mit der auf NEXT folgenden Instruktion fortgesetzt, wenn die Schleifenvariable die Intervallgrenzen überschreitet.

Anmerkung:

Von den Ausdrücken wird jeweils nur der ganzzahlige Anteil verwendet.

Beispiele:

```
(1) 10 FOR I = 1 TO 30 STEP 3
    20 PAUSE I
    :
    50 NEXT I
```

Schleifenprogrammteil

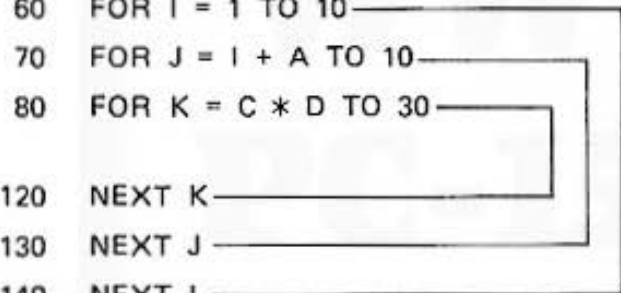
In diesem Beispiel werden die Zeilen 10 bis 50 wiederholt durchlaufen. Beim ersten Durchlauf hat die Variable I den Wert 1. Vor dem nächsten Lauf wird I um die Schrittweite 3 erhöht und hat den Wert I = 4. Dieser Wert liegt innerhalb des Intervalls und der Programmteil von Zeile 10 . . . 50 wird erneut durchlaufen. Beim letzten Durchlauf hat I den Wert 31. Will man erreichen, daß der Endwert 30 nicht überschritten wird, so muß als Endwert 28 in Zeile 10 eingesetzt werden.

```
(2) 10 PAUSE "LISTE DER QUADRATZAHLEN"
    20 PAUSE "VON 10 - 20"
    30 FOR Z = 10 TO 20
    40 Q = Z * Z
    50 PAUSE Q
    60 NEXT Z
    70 END
```

Bemerkungen:

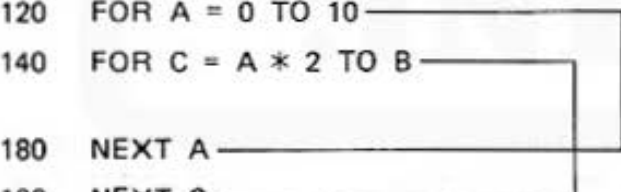
- (1) Wie schon erwähnt, können FOR . . . TO/NEXT Schleifen in bis zu 16 Ebenen geschachtelt werden. Es ist jedoch zu beachten, daß jede Schleife ihre eigene Schleifenvariable benötigt sowie die Schleifengrenzen sich nicht überlappen dürfen.

```
(3) 60 FOR I = 1 TO 10
    70 FOR J = I + A TO 10
    80 FOR K = C * D TO 30
    120 NEXT K
    130 NEXT J
    140 NEXT I
```



richtige
Schachtelung

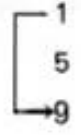
```
(4) 120 FOR A = 0 TO 10
    140 FOR C = A * 2 TO B
    180 NEXT A
    190 NEXT C
```



falsche
Schachtelung

- (2) Auf keinen Fall darf man während der Programmausführung von außen in einen Schleifenprogrammteil hineinspringen (z.B. durch eine GOTO-Instruktion). Dies führt nicht sofort zu einem Fehler, aber spätestens bei der NEXT-Instruktion wird das Programm mit einem Fehlercode abgebrochen. Zulässig ist es jedoch, aus dem Schleifenprogrammteil vor der nächsten NEXT-Instruktion herauszuspringen.

```
(5) 1 GOTO 9
    5 FOR I = 1 TO 10
    9 PRINT "FALSCH"
    10 NEXT I
```



falsche
Verzweigung

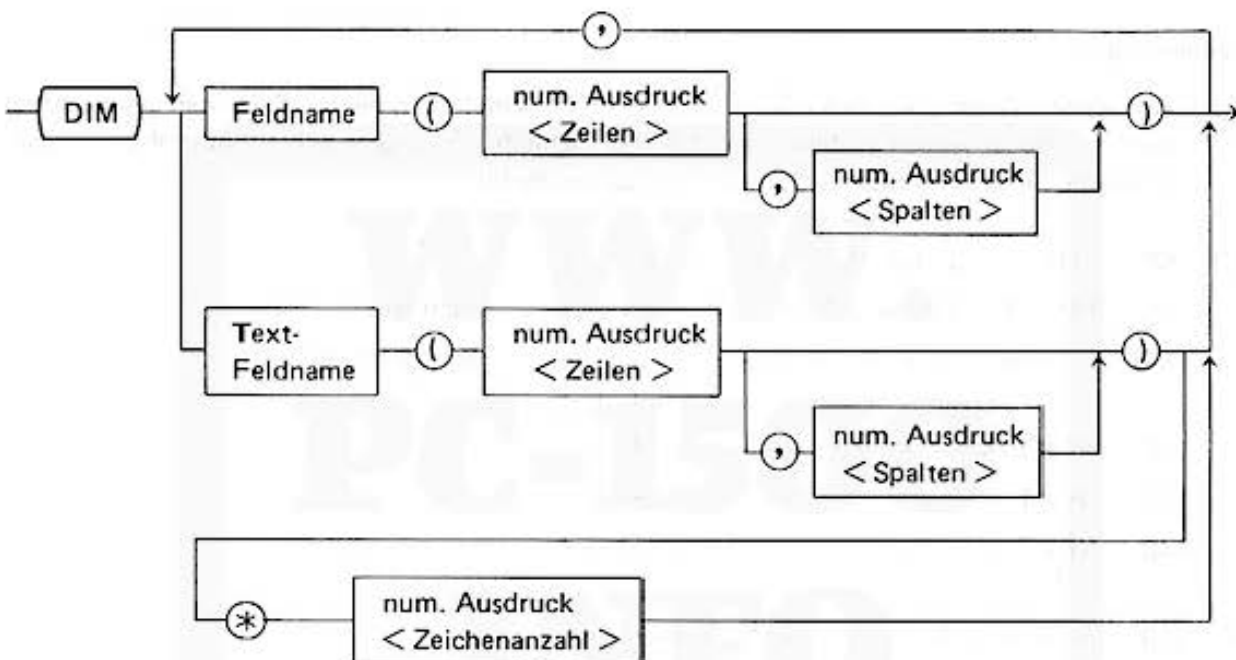
```
(6) 100 FOR I = 1 TO 1000
    110 READ A$
    120 IF A$ = "ENDE" GOTO 190
    180 NEXT I
    190 END
```

richtige
Verzweigung

DIM

Mit dieser Instruktion werden Feldvariablen (Arrays) deklariert.

Syntax



Im Gegensatz zu einfachen Variablen müssen Feldvariablen vor ihrer ersten Benutzung definiert werden. In der Deklaration wird die Größe des Feldes festgelegt und bei Textvariablen zusätzlich die maximale Anzahl von Zeichen.

Mit dieser Instruktion wird gleichzeitig Speicherplatz reserviert.

Der Name der Feldvariablen bezeichnet das gesamte Feld.

Die Werte der numerischen Ausdrücke spezifizieren die Anzahl der Zeilen und Spalten des zweidimensionalen Feldes. Wird der Ausdruck < Spalten > fortgelassen, so erhält man einen eindimensionalen Vektor.

Textfelder können ebenfalls als ein- oder zweidimensionale Felder deklariert werden. Den Ausdruck < Zeichenanzahl > spezifiziert die Zeichenkapazität eines jeden Feldelementes.

Der Zugriff auf ein Feldelement während des Programmlaufs erfolgt durch die Angabe des Namens und ein bzw. zwei Indizes (numerische Ausdrücke) je nach der Deklaration der Feldvariablen und in analoger Form wie in der DIM-Instruktion. Im Unterschied zu einer normalen Variablen sieht die LET-Instruktion beispielsweise wie folgt aus:

LET AR (1, 1) = 3.14

Bemerkungen:

- (1) Die Dimensionierung ist limitiert erstens durch den zur Verfügung stehenden Speicherplatz und zweitens durch die Indizes, die aus dem Intervall 0 – 255 sein müssen. (ERROR 10 bei Speicherplatzmangel und ERROR 8 bei falschem Index)
- (2) Bei Textfeldern darf die Zeichenanzahl eines Elementes 80 nicht überschreiten. Wird der Ausdruck fortgelassen, so erhalten die Elemente eine Kapazität von 16 Zeichen.
- (3) Bei der Definition erhalten numerische Felder die Werte 0 und Textfelder das ASCII-Zeichen Null.
- (4) Ein Feld darf nur einmal im Programmablauf definiert verwendet werden. Sonst erfolgt eine Fehlermeldung. (ERROR 5)
- (5) Da Null ein legaler Wert für den Index ist, hat das Feld jeweils eine um 1 größere Anzahl von Zeilen bzw. Spalten als die Werte der Definition.

Beispiele:

- (1) Die Zeilen 10 und 20 spezifizieren numerische Felder, die folgendermaßen aussehen:

10 DIM X (5)

X (0)	X (1)	X (2)	X (3)	X (4)	X (5)
-------	-------	-------	-------	-------	-------

Das eindimensionale Feld X hat also $5 + 1 = 6$ Elemente.

20 DIM AB (3,4)

AB (0, 0)	AB (0, 1)	AB (0, 2)	AB (0, 3)	AB (0, 4)
AB (1, 0)	AB (1, 1)	AB (1, 2)	AB (1, 3)	AB (1, 4)
AB (2, 0)	AB (2, 1)	AB (2, 2)	AB (2, 3)	AB (2, 4)
AB (3, 0)	AB (3, 1)	AB (3, 2)	AB (3, 3)	AB (3, 4)

Das zweidimensionale Feld AB hat $3 + 1 = 4$ Zeilen,

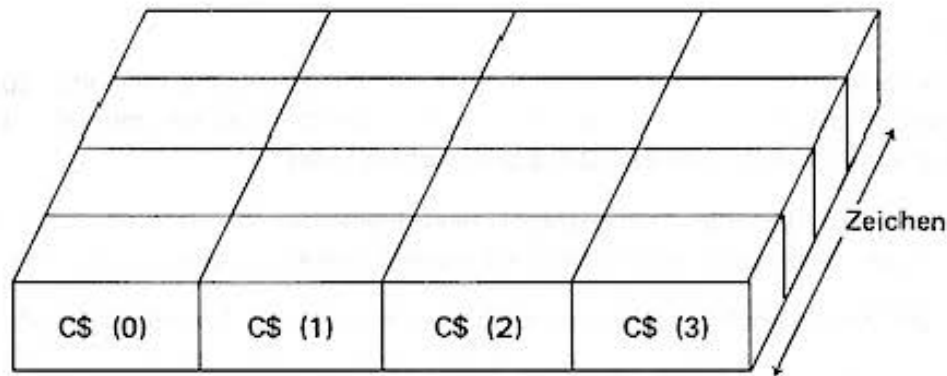
$4 + 1 = 5$ Spalten

und

$4 \times 5 = 20$ Elemente

- (2) 40 DIM C\$ (3) * 4

Zeile 40 definiert einen Textvektor mit $3 + 1$ Elementen und jedes Element hat eine Kapazität von 4 Zeichen.



- (3) 10 DIM VA (3) , PR (2)
 20 I = 4 : J = 5
 30 DIM W\$ (I * J)

Die Definition in Zeile 30 entspricht der Textfelddefinition DIM W\$ (20).

- (4) 10 DIM X (5), ABS (2, 3)
 20 DIM X\$ (2) * 5
 30 DIM X (3, 8)

In Zeile 10 und 20 werden unterschiedliche Namen für die Felder benutzt (X und X\$).
 Zeile 30 gibt einem neuen Feld jedoch einen schon deklarierten Namen X.
 Dies führt zu einem Fehler:

ERROR 5 IN 30

- (5) 10 DIM SN (9, 1)
 20 FOR I = 0 TO 90 STEP 10
 30 SN (I/10, 1) = SIN I
 31 SN (I/10, 0) = I
 40 NEXT I
 50 FOR J = 0 TO 9
 60 PRINT SN (J, 0), SN (J, 1)
 70 NEXT J

In der Programmschleife von Zeile 20 bis 40 nimmt die Schleifenvariable I die Werte 0, 10, 20 . . . etc. an. In den Zeilen 30 und 31 werden die Indizes des Feldes SN daher durch I/10 bestimmt.

Nach einem Durchlaufen der Schleife 20 bis 40 enthält SN in den Zeilen die Winkel und den SINUS dieses Winkels.

- (6) 10 DIM A\$ (3) * 5
 20 A\$ (1) = "ABCDE"

Dem ersten Feld der Textvariablen wird die Zeichenfolge "ABCDE" zugewiesen.

Anmerkung:

- (1) Es ist empfehlenswert, die Felder am Beginn des Programms zu deklarieren.
- (2) Auch ohne DIM-Anweisung ist im Rechner ständig das Array @ (26) definiert. Dieses Array ist den numerischen Variablen A bis Z überlagert (siehe auch Speicherorganisation), d.h.
 - @ (1) ist identisch mit A
 - @ (2) ist identisch mit B usw.

Ebenso ist das Textfeld @\$ (26) definiert, welches die Textvariablen AS bis Z\$ überlagert, d.h.

@\$ (1) ist identisch mit AS

@\$ (26) ist identisch mit Z\$

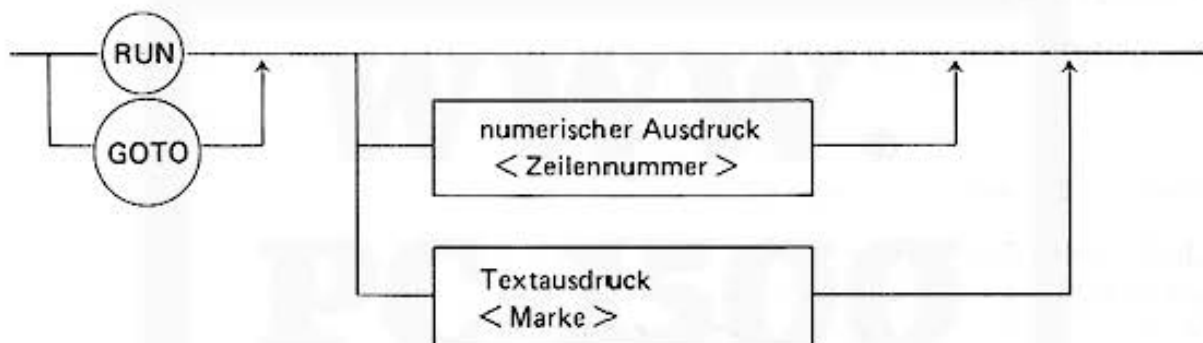
RUN, GOTO

Mit diesen Kommandos wird die Programmausführung begonnen.

Ein Basic-Programm kann im RUN-Mode auf verschiedene Art gestartet werden, durch Eingabe von:

1. RUN
2. GOTO (siehe auch BASIC-Instruktion GOTO)

Syntax



Ist eine Zeilennummer oder eine Marke spezifiziert, so wird die Ausführung mit dieser Zeile begonnen, anderenfalls mit der ersten Zeile des Programms.

3. Drücken der Taste **DEF** und eine der Tasten

A S D F G H J K L Z X C V B N M = SPACE .

Die Ausführung wird dann mit der Zeile begonnen, die das benutzte Zeichen als Marke enthält. Ist die Marke nicht im Programm enthalten, so wird ein Fehler gemeldet (ERROR 11).

Hinweis:

Der Rechner reagiert auf eine Eingabe wie unter 3. beschrieben auch, wenn er durch einen Input-Befehl auf eine Eingabe wartet. Die Input-Instruktion wird dann nicht ausgeführt, sondern ein Sprung zur spezifizierten Zeile gemacht.

4. Durch entsprechende Programmierung der RESERVE-Tasten (Siehe RESERVE.)

Unterschiede zwischen RUN, GOTO und Definable Keys

Beim Start des Programms werden bestimmte Grundzustände des Rechners gesetzt. Hierbei unterscheiden sich die drei Startkommandos.

Zustandsänderung erfolgt bei	RUN	GOTO	DEF
Die Anzeige wird gelöscht (siehe auch AREAD)	ja	ja	nein
Cursor-Position gesetzt	ja	nein	nein
Variable mit zwei Zeichen im Namen werden gelöscht	ja	nein	nein
Der ON ERROR GOTO Befehl wird gelöscht	ja	nein	nein
Das Kommando RESTORE wird ausgeführt (siehe auch BASIC-Befehle READ und DATA)	ja	nein	nein
Das USING Format wird gelöscht	ja	nein	nein

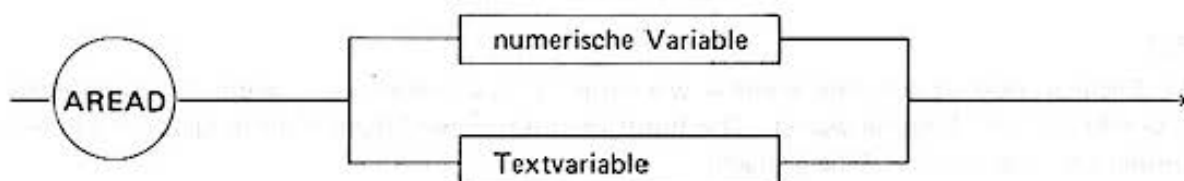
Bei allen drei Startbefehlen wird:

- Die FOR-NEXT Information gelöscht
- Die GOSUB Rücksprungadressen gelöscht
- Variable mit einem Buchstaben im Namen nicht gelöscht
- Das Wait Intervall und der TRACE-Zustand wird nicht geändert.

AREAD

Auslesen der Anzeige nach dem Start eines Programms mit DEF (Taste).

Syntax



Der auf der Anzeige stehende numerische oder Textausdruck wird der numerischen bzw. Textvariablen zugewiesen. Die AREAD-Anweisung muß die erste Programminstruktion sein.

Beispiel:

```
Programm      10  "S" : AREAD X
               20  PRINT "SINH (" ; X ; ") ist";
               30  PRINT (EXP (X) - EXP (-X)) / 2
```

Ausführung

1.5

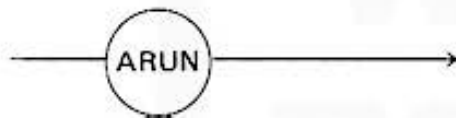
Anzeige

SINH (1.5) ist 2.129279455

ARUN

Automatisches Starten eines Programmes mit dem Einschalten des Rechners.

Syntax

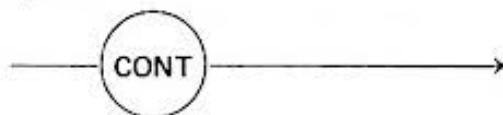


Ist der erste Befehl eines Programms ARUN, so wird das Programm mit dem Einschalten des Rechners gestartet. Der Rechner muß dazu beim Ausschalten im RUN-Mode gewesen sein und darf keine Fehlermeldung auf der Anzeige gehabt haben.

CONT

Nach einem STOP im Programm oder dem Drücken der Break-Taste kann der Programmablauf mit dem Befehl CONT fortgesetzt werden.

Syntax



Alle Zustände des Rechners (FOR-NEXT, GOSUB usw.) bleiben dabei erhalten. Dadurch ist es möglich, den Programmablauf zu unterbrechen, um die aktuellen Werte von Variablen zu überprüfen und das Programm dann weiterlaufen zu lassen.

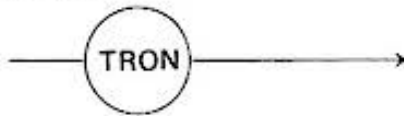
EinzelSchrittverfahren:



Bei einem STOP wird die zuletzt ausgeführte Zeile durch Drücken der Taste gezeigt. Durch die Taste wird nur die nächste Programmzeile ausgeführt. Es ist so möglich, ein Programm schrittweise auszuführen und zu überprüfen.

TRACE-Kommandos: TRON, TROFF

Zum Suchen von Programmfehlern kann der Rechner durch das Kommando TRON in den Trace (Spur-) Zustand gesetzt werden.

Syntax



Das Programm wird dann nach jeder Zeile automatisch gestoppt und die aktuelle Zeilennummer angezeigt. Durch die Tasten  und  wird die Programmzeile angezeigt bzw. das Programm fortgeführt.


TROFF

Durch das Kommando TROFF wird der Trace-Zustand wieder abgeschaltet.

Syntax



Hinweise:

- 1) Die Kommandos TRON und TROFF können auch im Programm stehen. Dadurch ist es möglich, begrenzte Programmteile zu überprüfen.
- 2) Wird die Taste  längere Zeit festgehalten, so wiederholt der Rechner die Tastenfunktionen. Dadurch können bestimmte Programmteile schnell durchlaufen werden.

CLEAR

Löschen aller Variablen und Felder aus dem Hauptspeicher (siehe Speicherorganisation). Die Standardvariablen werden auf 0 gesetzt.

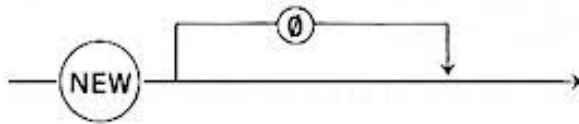
Syntax



NEW

Löschen des Programmspeichers im PRO-Mode, der Tastendefinitionen im RESERVE-Mode. Im RUN-Mode nicht zugelassen.

Syntax



Hinweis: Mit dem Kommando NEW ∅ wird der Rechner nach dem Batteriewechsel initialisiert.

Pro-Mode:

Das gespeicherte Programm und alle gespeicherten Variablen werden gelöscht bzw. auf null gesetzt (siehe auch Clear).

Reserve-Mode:

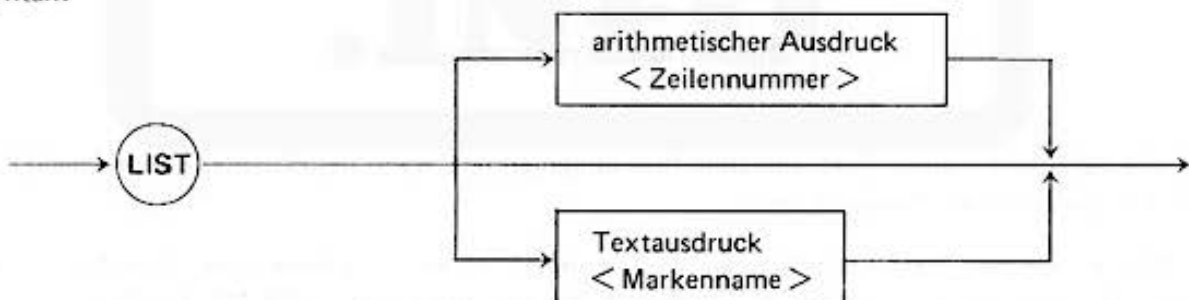
Alle Tastenfunktionen und die Tastenbeschriftungstexte werden gelöscht (siehe Reserve-Mode).

Auflisten eines Programms



LIST

Die LIST-Anweisung holt eine Zeile auf die Anzeige.

Syntax:



Der numerische Ausdruck spezifiziert die Nummer, der Textausdruck den Markennamen der Zeile, die gelistet werden soll. Ist keins von beiden angegeben, so wird die erste Zeile des Programms gezeigt. Ist die angegebene Zeile nicht vorhanden, wird die nächsthöhere gelistet. Ist keine höhere Zeile vorhanden oder fehlt die Marke, so wird ein Fehler gemeldet (ERROR 11).

Mit den Tasten  und  wird die nächste oder vorherige Zeile angezeigt.

Ein- / Ausgabe Anweisungen

Die in folgendem Kapitel beschriebenen Instruktionen gestatten es Ihnen, während des Programmlaufs Werte über die Tastatur einzugeben und auf der Anzeige, bzw. wenn Sie über die Druckeroption verfügen, auf dem Drucker formatiert auszugeben.

Es ist nicht nur möglich, die Zeichen des Alphabets zu benutzen. Auf der Anzeige lassen sich mit einfachen Instruktionen Muster aus einzelnen Punkten erzeugen.

Textausgaben können bis zu 80 Zeichen umfassen, auf der Anzeige werden jedoch nur die ersten 24 Zeichen angezeigt.

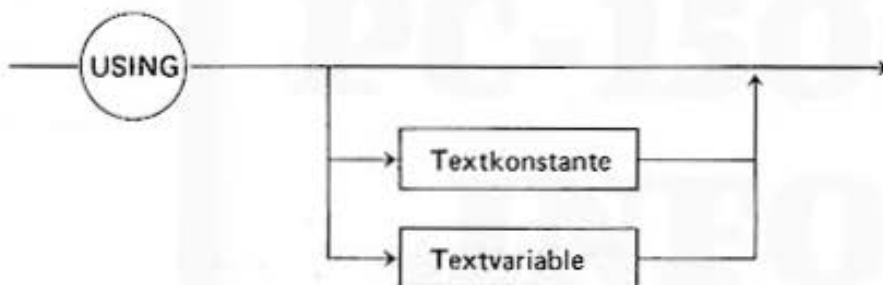
Sie können vierfarbige Graphiken vom Drucker zeichnen und Texte in verschiedenen Größen und Richtungen schreiben lassen.

Zahlen werden im Fließkommaformat ausgegeben, wenn dafür nicht mehr als 10 Ziffern notwendig sind. Sonst wird das wissenschaftliche Format verwendet.

USING

Diese Instruktion spezifiziert das Ausgabeformat von Texten und Werten numerischer Ausdrücke.

Syntax



Wie bereits im Kapitel Taschenrechnerfunktionen erklärt, wählt der PC-1500 entsprechend dem Wert automatisch ein Ausgabeformat.

Mit der USING Instruktion können Sie bestimmen, in welchem Format die Ausgabe erfolgen soll. Die Instruktion wirkt auf alle folgenden Ausgabeanweisungen (PRINT, PAUSE, LPRINT etc.).

Die Formatspezifikation kann als Textkonstante oder als Wert einer Textvariablen angegeben werden.

Wird die USING-Instruktion allein ohne Formatspezifikation gegeben, so wird damit die letzte USING-Anweisung unwirksam und auf die automatische Formatkonvertierung zurückgeschaltet. Das gleiche bewirkt die Ausführung des RUN-Kommandos oder die **SHIFT** **CL** -Taste.

Mit der Textkonstante bzw. der Textvariablen wird das Format spezifiziert. Zur Festlegung des Formats dienen die Sonderzeichen # * & , \ Punkt (.) und +.

bestimmt die Anzahl der Stellen im Ausgabefeld;

- legt die Stelle des Dezimalpunktes fest und \ führt zur Anzeige im wissenschaftlichen Format. Gibt man nur eine Folge der Zeichen # ein, so wird lediglich der ganzzahlige Anteil einer Zahl angezeigt, einschließlich Vorzeichen bei negativen Zahlen. Daher muß das Feld aus den # # eine Stelle mehr besitzen als die größte anzuzeigende Zahl an Ziffern benötigt.
- + spezifiziert die generelle Ausgabe des Vorzeichens.

Die Folge der # Zeichen nach dem Dezimalpunkt bestimmt die Anzahl der Nachkommastellen.

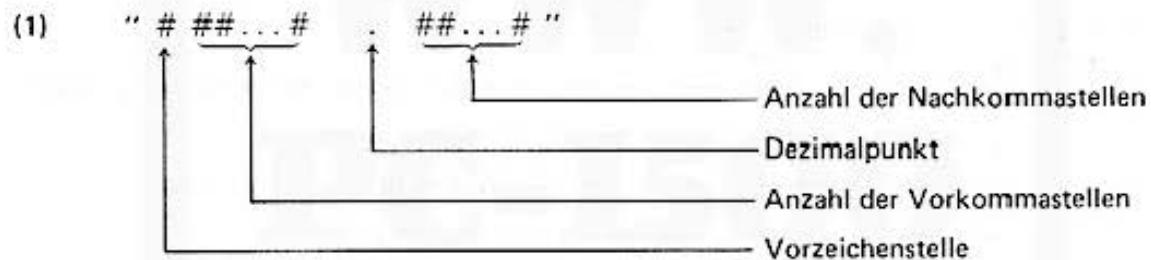
* kann anstelle von # verwendet werden; führende Nullen werden dann nicht als Zwischenraum (' '), sondern als * angezeigt.

In den Nachkommastellen werden fehlende Stellen immer mit Nullen aufgefüllt.

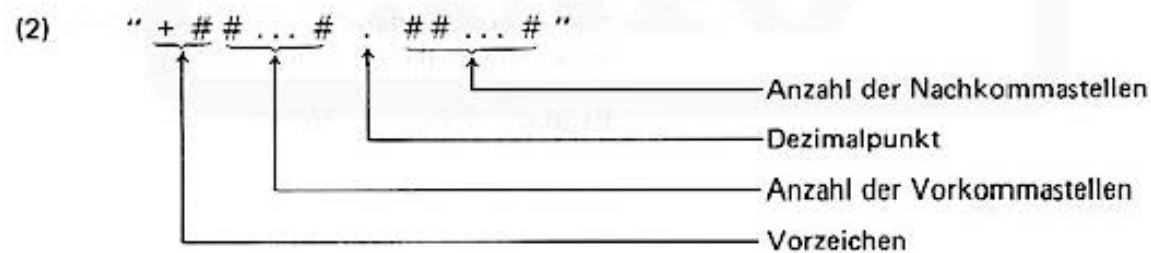
Die Folge der "&" bestimmt die Anzahl der Zeichenstellen von Texten und Textausdrücken.

Das "," setzt die Tausenderstelle ab, wie es in der anglikanischen Schreibweise großer Zahlen üblich ist.

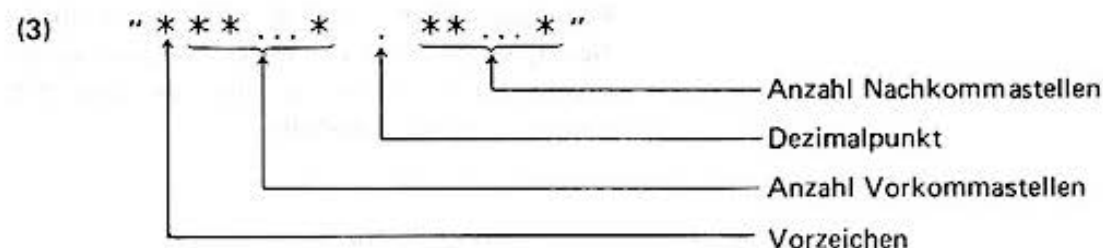
Die unterschiedlichen Formate lauten damit allgemein:



Das Vorzeichen wird nur bei negativen Zahlen ausgegeben.



Das Vorzeichen wird immer direkt vor der ersten Ziffer der Zahl ausgegeben. (im Gegensatz zu Beispiel 1)



Das Vorkommafild wird immer vollständig mit * Zeichen aufgefüllt, z.B. ***1.00

(4) " + * * . . . * . * * . . . * * "

Analog zu (2), das Vorzeichen steht jedoch vor dem Ziffernfeld und fehlende Stellen werden mit * aufgefüllt, z.B. + * * * 1.00

(5) normierte Exponentialdarstellung zur Basis 10

" # # . # # . . . # ^ "

wissenschaftliches Format
Mantisse
Vorzeichenstelle

Das Format ist normiert auf eine Vorkomma-
stelle, die Nachkommastellen und den Ex-
ponenten. Gibt man beim wissenschaftlichen
Format mehr als eine Vorkommastelle, *
anstatt # oder mehrere ^, so hat das keinen
Einfluß auf die Anzeige.

(6) Zahlen in anglikanischer Schreibweise

" # # # . . . # , . # # . . . # "

Anzahl Nachkommastellen (optional)
Dezimalpunkt
Anzahl der Vorkommastellen
Vorzeichen

Ausgabe als Zahl in dem spezifizierten Feld.
Die Tausender- und Million-Stelle wird durch
ein Komma von der vorherigen Stelle abge-
trennt. Die Kommata sind in der Anzahl der
Vorkommastellen zu berücksichtigen.

Beispiel: 10000 als 10,000

(7) Zeichenfelder

" & & . . . & "

Anzahl der Zeichen des Feldes

Die Zeichen eines Textausdrucks werden in das
Feld geschrieben. Ist das Feld kleiner spezi-
fiziert, werden nur die ersten Zeichen ausge-
druckt, ist es größer, so wird der Rest mit
Zwischenräumen aufgefüllt.

Formatfeldspezifikation

#	Repräsentiert die Stellen des ganzzahligen (≤ 15 einschließlich Vorzeichen) und gebrochenzahligen Anteils (≤ 14). Führende Nullen werden in Leerzeichen gewandelt, die Nachkommastellen mit Nullen aufgefüllt.
*	Gleiche Wirkung wie #. Führende Nullen werden als * ausgegeben.
.	Repräsentiert den Dezimalpunkt.
^	Wissenschaftliche Formatausgabe.
+	Ausgabe generell mit Vorzeichen.
,	Anglikanisches Format: Abtrennen von Blöcken aus jeweils drei Ziffern durch Kommata.
&	Repräsentiert die Stellen zur Ausgabe von Zeichenfolgen.

Bemerkungen:

- (1) Reicht das durch die USING-Instruktion festgelegte Ausgabeformat für die Darstellung einer Zahl nicht aus (z.B. Stellenanzahl), so wird ein Fehler gemeldet (ERROR 36).
- (2) Fehlerhafte Formate werden teilweise erst bei den Ausgabeinstruktionen (PRINT, LPRINT, PAUSE) erkannt und angezeigt.

Beispiele:

- (1)


```
10 A = 123.456789
20 USING " #####.#"
30 PRINT A
40 USING
```

die Ausgabe erfolgt in Zeile 30 als 123.4; in Zeile 40 wird wieder zurückgeschaltet auf die automatische Formatierung.
- (2) Die Ausgabe bei unterschiedlichen Formaten in Zeile 20

Format	Anzeige
< Standardformat >	123.456789
" ###.## "	Fehlercode 36
" ##### "	123
" #####. "	123.
" #####.## "	123.45
" *#####.## "	*123.45
" +*****.## "	+*123.45
" ##.###^ "	1.234E 02
" .#####^ "	1.23456E 02

(3) 10 $X = \sqrt{3}$
 20 A\$ = "ABCDEF"
 30 F\$ = " *** "
 40 USING F\$: PRINT -X
 50 USING
 60 PRINT-X

Die Ausgabe erfolgt in Zeile 40 als

- * 1

und Zeile 60 als

- 1.732050808

(4) 10 $X = \sqrt{3}$
 20 A\$ = "ABCDEF"
 30 F\$ = " *** "
 40 USING F\$: PRINT A\$;X

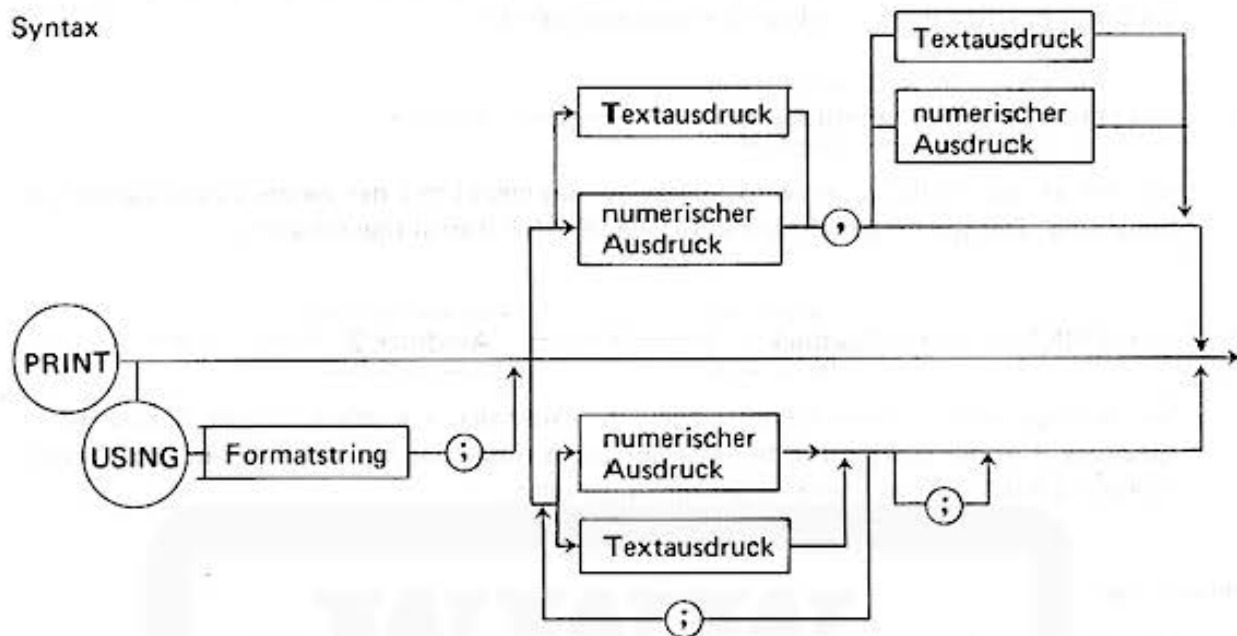
Die Tabelle zeigt die unterschiedlichen Ausgabeformate in Abhängigkeit der Formatspezifikation.

30 F\$ =	40 PRINT	Anzeige
" ##.##^&&& "	X; A\$	1.7E 00ABC
" &&&&&&&&##.## "	A\$; X	ABCDEF...1.73

PRINT

Mit dieser Instruktion werden numerische Werte und Zeichenfolgen auf der Anzeige ausgegeben.

Syntax



Der USING-Teil der Anweisung spezifiziert das Ausgabeformat, so wie in der USING-Instruktion beschrieben. Der Formatstring steht für die dort erwähnte Textkonstante bzw. Textvariable.

Über das Komma wird die Aufteilung der 26-spaltigen Anzeige in zwei gleich große Felder gesteuert.

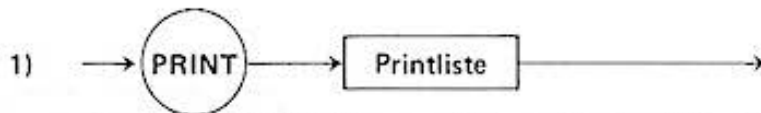
Das Semikolon trennt die einzelnen Elemente einer Liste von Ausdrücken und steuert die Position des Cursors.

Steht am Ende einer PRINT-Instruktion ein Semikolon, so schließt die folgende Ausgabe an die vorangehende an; andernfalls wird die alte Anzeige vorher gelöscht.

Wird die Cursor-Positionierung nicht explizit gesteuert (z.B. durch die CURSOR-Instruktion oder die Verwendung eines Semikolons), so werden die Werte einzelner numerischer Ausdrücke rechtsbündig und Textausdrücke linksbündig in das Feld geschrieben.

Nach dem Ausdrucken der PRINT-Liste (< Ausdrücke >) auf der Anzeige wird der Programmaufhalt bis man ihn durch Drücken von **ENTER** wieder startet. Es sei denn, über die WAIT-Instruktion ist ein Zeitintervall definiert worden, nach dessen Ablauf die Ausführung automatisch wieder aufgenommen wird (siehe WAIT ...).

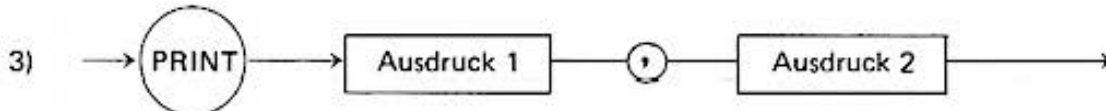
Die einzelnen Formate der PRINT-Instruktion haben folgende Wirkung:



Der in der PRINT-Liste spezifizierte Inhalt wird auf dem Display angezeigt, beginnend mit der momentanen Cursorposition, wenn explizit angegeben, sonst rechts- bzw. linksbündig. Der Inhalt der Anzeige wird mit der nächsten PRINT-Anweisung vollständig gelöscht und die Cursorposition wieder in den Grundzustand gesetzt.



Der Inhalt der PRINT-Liste wird angezeigt, beginnend mit der momentanen Cursorpositionierung. Der Cursor bleibt für die folgende PRINT-Instruktion erhalten.



Die Anzeige wird in zwei Felder eingeteilt. Ausdruck 1 erscheint in der linken Hälfte, Ausdruck 2 in der rechten Hälfte. Die Cursorpositionierung hat keine Bedeutung. Mit der nächsten PRINT-Instruktion wird der Inhalt gelöscht.

Anmerkungen:

- (1) Mit der Cursor-Instruktion kann die Positionierung explizit gesteuert werden.
- (2) Mit der WAIT-Instruktion kann ein Zeitintervall definiert werden, nach dessen Ablauf die Programmausführung automatisch wieder gestartet wird.
- (3) Werden mehr Zeichen ausgegeben, als die Anzeige Schreibpositionen hat, so sind nur die ersten 26 sichtbar.
- (4) Wird die PRINT USING-Instruktion benutzt, so ist das Format für alle PRINT-Instruktionen bis zur nächsten USING- Instruktion wirksam.
- (5) Das USING- Format wird durch das Kommando RUN oder **[SHIFT]** **[CL]** aufgehoben.

Beispiele:

- (1) 10 A = 3.1415
- 20 PRINT A
- 30 SS = "SHARP"
- 40 PRINT SS

Mit Zeile 20 erscheint auf der Anzeige:

3.1415

Zur Fortsetzung des Programmlaufs muß die **[ENTER]** Taste gedrückt werden und mit Zeile 40 erscheint:

SHARP

auf der Anzeige.

```
(2) 10 PL = 2000 : PD = 1
    20 O$ = "  HH "
    30 S$ = "SHARP"
    40 PRINT S$; "  (EUROPE)"
    50 PRINT PL; O$;
    60 PRINT PD
```

Mit Zeile 40 erscheint auf der Anzeige

SHARP (EUROPE)

nach der Eingabe von **ENTER** wird die Anzeige gelöscht und es ist zu lesen

2000 HH

nach abermaliger Eingabe von **ENTER** wird dies ergänzt zu

2000 HH 1

```
(3) 10 FOR I = 1 TO 5
    20 PRINT I
    30 NEXT I
```

Auf der Anzeige erscheinen der Reihe nach rechtsbündig die Zahlen 1, dann 2, etc.

1

jeweils nach der Eingabe von **ENTER**

```
(4) 10 FOR I = 1 TO 5
    20 PRINT I;
    30 NEXT I
```

Auf der Anzeige erscheinen:

1

1 2

⋮

1 2 3 4 5

(5) 10 PL = 2000, PD = 1
20 PRINT PL, "Hamburg"

Auf der Anzeige erscheint,
linkes Feld

rechtes Feld

2000	Hamburg
------	---------

oder

30 PRINT "HAMBURG", PD

Hamburg	1
---------	---

(6) 10 PD = 1
20 OS = "HAMBURG"
30 PRINT USING "&&&###"; OS, PD

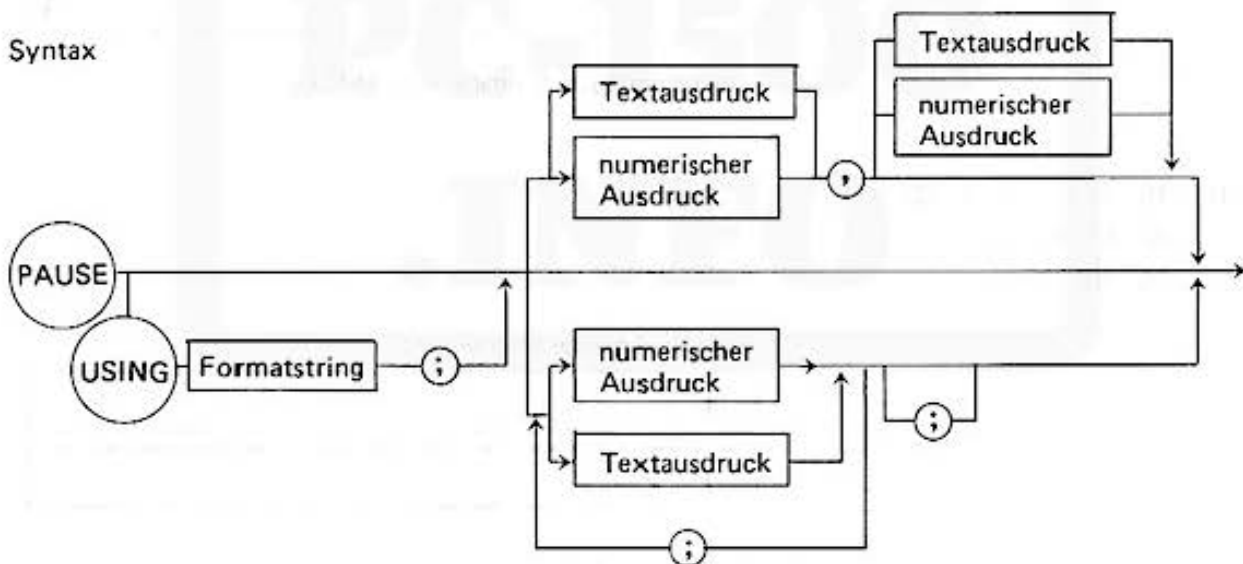
Dies führt zur Anzeige:

HAM	1
-----	---

PAUSE

Mit dieser Instruktion werden numerische Werte und/oder Zeichenfolgen auf der Anzeige ausgegeben und der Programmablauf automatisch fortgesetzt.

Syntax

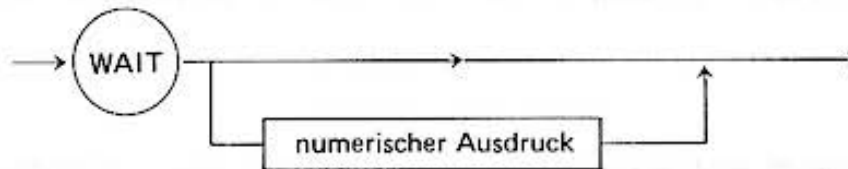


Die Syntax der PAUSE-Instruktion ist so aufgebaut wie bei der PRINT-Anweisung. Die Bedeutung der einzelnen Sprachelemente ist die gleiche. Im Unterschied zur PRINT-Instruktion wird der Programmlauf nach ca. 0.85 sec. wieder gestartet.

WAIT

Diese Instruktion legt das Zeitintervall fest, das vergehen soll, bevor nach einer PRINT-Instruktion der Programmablauf wieder startet.

Syntax



Normalerweise wird der Programmablauf nach einer PRINT-Instruktion erst mit Eingabe der **ENTER** Taste fortgesetzt. Die WAIT-Instruktion gestattet es, ein Zeitintervall vorzugeben, nach dessen Ablauf die Ausführung automatisch wieder gestartet wird.

Der numerische Ausdruck legt das Intervall fest.

Gemessen wird in Einheiten von ca. 1/64 sec. Die Werte des Ausdrucks dürfen variieren von 0 bis 65535. Die Wartezeit ist damit $n \cdot 1/64$ und liegt zwischen 0 sec. und 1023.9 sec.

Die WAIT-Instruktion wirkt auf alle folgenden PRINT- und GPRINT-Anweisungen.

Gibt man keinen Ausdruck an, so wird die Automatik abgeschaltet und die Ausführung muß wieder mit **ENTER** fortgesetzt werden.

Beispiel:

Die PAUSE-Instruktionen haben die gleiche Wirkung wie

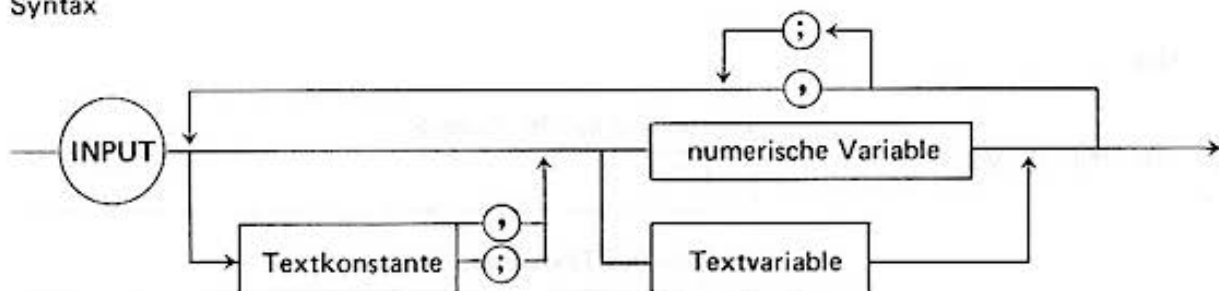
```

10 WAIT 54.4
100 PRINT      anstatt PAUSE
    
```

INPUT

Diese Instruktion weist Variablen Werte zu, die über die Tastatur eingegeben werden.

Syntax



Die Ausführung des Programms wird unterbrochen. Es erscheint auf der Anzeige die Textkonstante oder ein "?", wenn kein Text spezifiziert ist. Ein Hinweis darauf, daß die Werte für die Variablen über die Tastatur eingegeben werden können. Jede einzelne Eingabe wird mit der **ENTER** Taste abgeschlossen. Dies wird wiederholt, bis für alle Variablen die Werte eingegeben sind. Im Anschluß daran wird die Programmausführung fortgesetzt.

Mit der Wahl zwischen Komma und Semikolon steuert man das Ausgabeformat auf der Anzeige. Benutzt man das Semikolon, so erscheint die Textkonstante linksbündig entsprechend der Cursorposition auf der Anzeige. Der Eingabewert wird gleich im Anschluß daran eingetragen; man verschiebt die Cursorposition, und der gesamte Text steht auf der Anzeige. Verwendet man das Komma, so erscheint die Textkonstante an der Cursorposition, ohne diese selbst zu verändern. Das erste Zeichen der Eingabe löscht die Textkonstante und der Eingabewert erscheint linksbündig an der alten Cursorposition.

Bezüglich der Formatsteuerung über das Komma bzw. Semikolon sei an die PRINT-Instruktion erinnert.

Bemerkungen:

- 1) Wenn eine Zeichenfolge eingegeben wird, so braucht sie nicht in Hochkommata eingeschlossen zu werden.
- 2) Wird die **ENTER** Taste ohne zusätzliche Zeicheneingabe gedrückt, so bleibt der Wert jener Variablen unverändert. Variablen, die vorher noch nicht definiert wurden, erhalten den Wert 0 (num. Variablen) bzw. leere Zeichenfolge "" (Textvariablen).
- 3) Fehlerhafte Eingaben können korrigiert werden durch die Editiertasten



Bestätigt man die **ENTER** Taste, so wird die Programmzeile angezeigt, die eine Eingabe erwartet. Die betreffende Variable wird durch Blinken angezeigt.

- 4) Wartet das Programm in der INPUT-Anweisung auf eine Eingabe, so ist es möglich, mit einem Definable Key einen Sprung zu einer spezifizierten Zeile auszuführen und das Programm dort fortzusetzen.

Beispiele

- 1) 10 INPUT A

Darstellung auf der Anzeige

?

Eingabe des numerischen Wertes

1234

20 IF A = 1234 THEN 100

100

- 2) 10 INPUT M\$

Darstellung auf der Anzeige

?

Eingabe des Textwertes

MAI

20 IF M\$ = "MAI" THEN 300

300

- 3) 10 PRINT "DATA 0 0"; A; "0";
 20 INPUT B
 30 A = A + 1: CLS
 40 GOTO 10

Die Eingabe folgender Kommandos zeigt folgendes Ergebnis auf der Anzeige (RUN-Mode):

RUN

DATA 0

DATA 0 ?

123

DATA 0 123

DATA 1

DATA 1 ?

345

DATA 1 345

etc.

- 4) Beispiel für die Wirkung des Cursor

10 PRINT "DATA_1 DATA_2";
 20 CURSOR 7
 30 INPUT A
 40 CURSOR 17
 50 INPUT B

RUN

DATA 1 DATA 2

DATA 1 ? DATA 2

12

DATA 1 12 DATA 2 ?

45

DATA 1 12 DATA 2 45

Die Cursor-Instruktion in Zeile 20, 40 steuern die Eingabeposition.

5) 10 INPUT "GESCHWINDIGKEIT km/h : "; S

50

GESCHWINDIGKEIT km/h :

GESCHWINDIGKEIT km/h : 50

6) 10 INPUT "HOEHE m : ", H

3721

HOEHE m:

3721

7) 10 WAIT 0

20 PRINT "HOEHE ";

30 INPUT " (feet) : "; H

RUN

10000

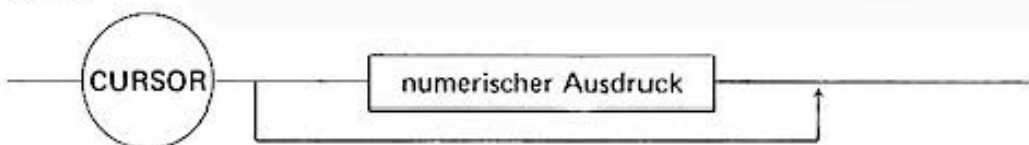
HOEHE (feet) :

HOEHE (feet) : 10000

CURSOR

Mit dieser Instruktion wird die Schreibposition auf der Anzeige festgelegt.

Syntax



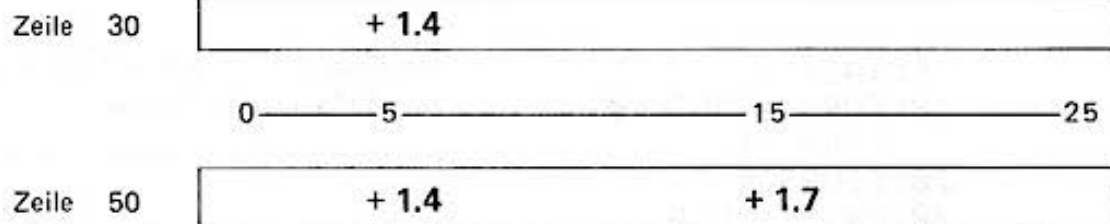
Die Anzeige hat 26 Schreibstellen mit einer Numerierung von links (0) nach rechts (25). Der Wert des numerischen Ausdrucks legt die Cursorposition auf der Anzeige für die folgende PRINT-Instruktion fest. Der Ausdruck kann die Werte 0 bis 25 annehmen.

Wird kein Ausdruck angegeben, so wird die Cursorpositionierung in den Grundzustand gesetzt. Die folgende PRINT-Instruktion löscht die alte Anzeige bevor sie die Printliste wie spezifiziert ausgibt.

Beispiele:

```
(1) 10 USING "+ ##. #"
    20 CURSOR 5
    30 PRINT  $\sqrt{2}$  ;
    40 CURSOR 5 + 10
    50 PRINT  $\sqrt{3}$ 
```

Die Anzeige hat nach den PRINT-Instruktionen jeweils das Aussehen



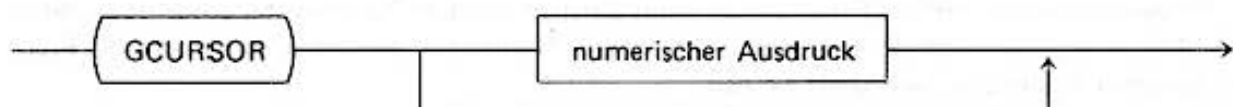
```
(2) 10 USING " ### "
    20 FOR I = 0 TO 23
    30 CURSOR I
    40 PRINT I;
    50 NEXT I
```

Auf der Anzeige erscheinen die Zahlen 0 bis 23, die von links nach rechts laufen. Die alte Anzeige wird in Zeile 40 jedesmal gelöscht, obwohl PRINT I; benutzt wurde.

GCURSOR

Mit dieser Instruktion wird, in Punktraster gemessen, die Schreibposition auf der Anzeige festgelegt.

Syntax:



Die Instruktion hat eine ähnlich Wirkung wie die CURSOR-Instruktion und legt die Schreibposition für die folgende PRINT- oder GPRINT-Anweisung fest.

Im Unterschied zur CURSOR-Instruktion kann jedoch die Position auf jede der 156 Punktspalten der Anzeige festgelegt werden. Die Adressierung erfolgt wieder von links (0) nach rechts (155).

Der Ausdruck kann somit die Werte 0 . . 155 annehmen. Wird der Ausdruck fortgelassen, so hat dies die gleiche Wirkung wie bei der CURSOR Instruktion.

Beispiel:

Für jedes Zeichen auf der Anzeige werden 6 Spalten benötigt.
Somit bezeichnen die Instruktionen

```
CURSOR 5    und
GCURSOR 30
```

die gleiche Position

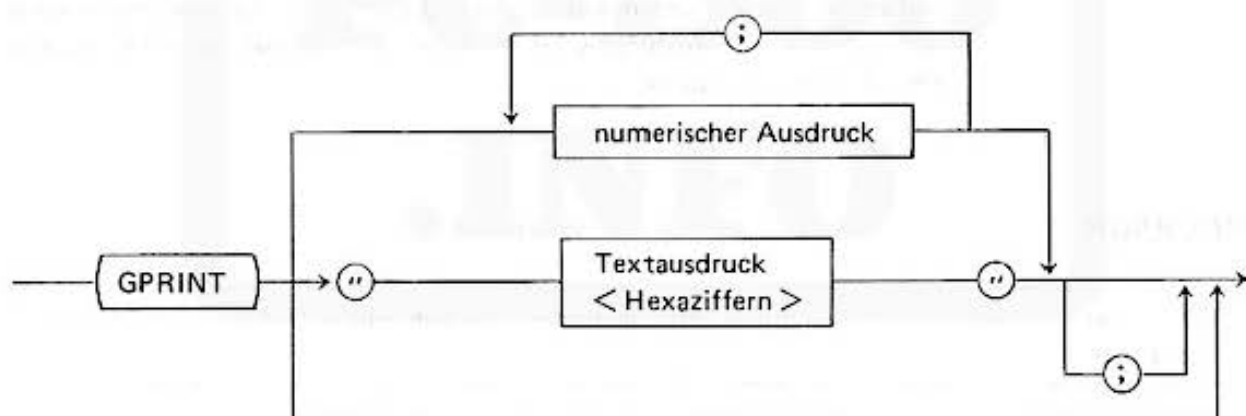
```
5:WAIT 1
10:FOR I=155TO 0
  STEP -1
20:GCURSOR I
30:PRINT "<";
40:NEXT I
```

Das Zeichen "<" wandert von rechts nach links über die Anzeige



GPRINT

Mit dieser Instruktion ist es möglich, graphische Muster auf der Anzeige zu erzeugen.



Es werden bei der GPRINT-Instruktion nicht Zeichen, sondern Punktmuster ausgegeben, die in der Textkonstante durch Folgen von Hexadezimalziffern (0 – 9 und A – F) oder als Folge numerischer Ausdrücke spezifiziert werden.

Sie können also neue Zeichen erfinden oder Graphiken auf der Anzeige darstellen.

Die GPRINT-Instruktion unterbricht den Programmlauf wie die PRINT-Anweisung und die Ausführung wird erst nach Eingabe von **ENTER** fortgesetzt. Mit WAIT kann die automatische Fortführung nach einem Zeitintervall spezifiziert werden.

Die Wirkung des Semikolons am Ende der Zeile ist die gleiche wie bei der PRINT-Anweisung; die folgende PRINT- oder GPRINT-Instruktion löscht die Anzeige nicht, sondern beschreibt die nächste Position.

Die Schreibposition wird entweder explizit über die CURSOR- bzw. GCURSOR-Instruktionen oder durch die bisher erfolgten Ausgaben festgelegt.

In der ersten Form wird das Punktmuster einer Spalte der 7×156 Punktmatrix durch den Wert eines numerischen Ausdrucks bestimmt.

Die Wertigkeiten der einzelnen Punkte ergeben sich aus der folgenden Darstellung:

1							
2							
4							
8							
16							
32							
64							
	0					Position	155

Das Muster einer Spalte ergibt sich aus der Summe der einzelnen Punktwertigkeiten.

Die Folge der Ausdrücke legt die Punktmuster aufeinanderfolgender Schreibpositionen fest. Für jede zu schreibende Spalte ist ein numerischer Ausdruck anzugeben.

In der zweiten Form wird statt der Folge numerischer Ausdrücke ein Textausdruck angegeben, der für jede Spalte zwei hexadezimale Ziffern enthalten muß. Der Wert der aus diesen beiden Ziffern gebildeten Zahl, entspricht jeweils dem numerischen Ausdruck in der ersten Form.

Eine Spalte der Anzeige aus 7 Punkten ist in 2 Teile, 4 obere und 3 untere Punktpositionen aufgeteilt. Das Muster einer Spalte wird durch zwei Hexadezimalziffern bestimmt. Die erste steht für den unteren und die zweite für den oberen Teil der Punktspalten. Der Wert der Hexadezimalzahl bestimmt, welche Punkte der Spalte angeschaltet werden und welche nicht.

4 Punkte	<ul style="list-style-type: none"> • * • * • • 	2. Stelle
3 Punkte	<ul style="list-style-type: none"> • • * • * 	1. Stelle

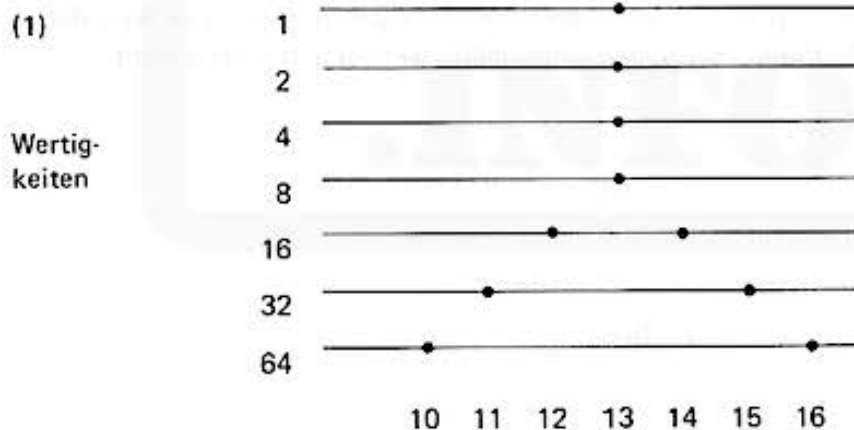
Punktspalten-Position

Das Beispiel entspricht der Zahl 63.

In der folgenden Tabelle finden Sie die Zuordnung von Punktmustern und Zeichen des Textausdruckes:

1	----	--*--	-----	--*--
2	----	----	--*--	--*--
4	----	----	----	----
8	----	----	----	----
	0	1	2	3
1	----	--*--	-----	--*--
2	----	----	--*--	--*--
4	--*--	--*--	--*--	--*--
8	----	----	----	----
	4	5	6	7
1	----	--*--	-----	--*--
2	----	----	--*--	--*--
4	----	----	----	----
8	--*--	--*--	--*--	--*--
	8	9	A	B
1	----	--*--	-----	--*--
2	----	----	--*--	--*--
4	--*--	--*--	--*--	--*--
8	--*--	--*--	--*--	--*--
	C	D	E	F

Beispiel:



Dieses Punktmuster wird durch folgende Werte bestimmt, beginnend mit Position 10

64; 32; 16; 15; 16; 32; 64;

das Muster in Position 13 ergibt sich aus der Summe der Wertigkeiten der einzelnen Punkte

1 + 2 + 4 + 8

10 GCURSOR 10

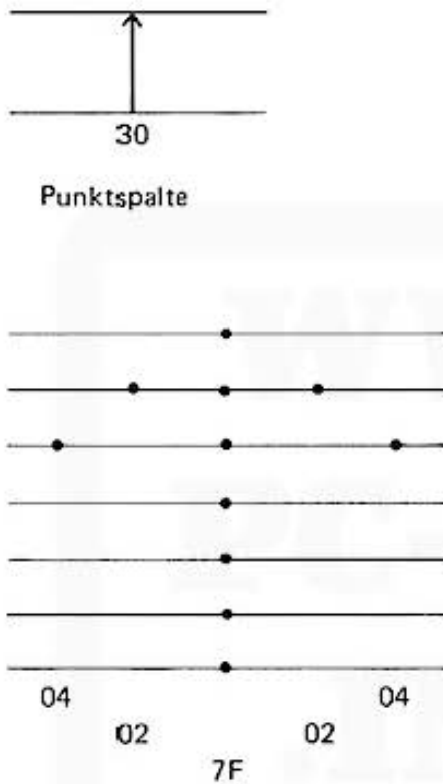
20 GPRINT 64; 32; 16; 15; 16; 32; 64;


```
(2) 10 WAIT 1
    20 FOR I = 0 TO 155
    30 GPRINT I/155 * 127;
    40 NEXT I
```

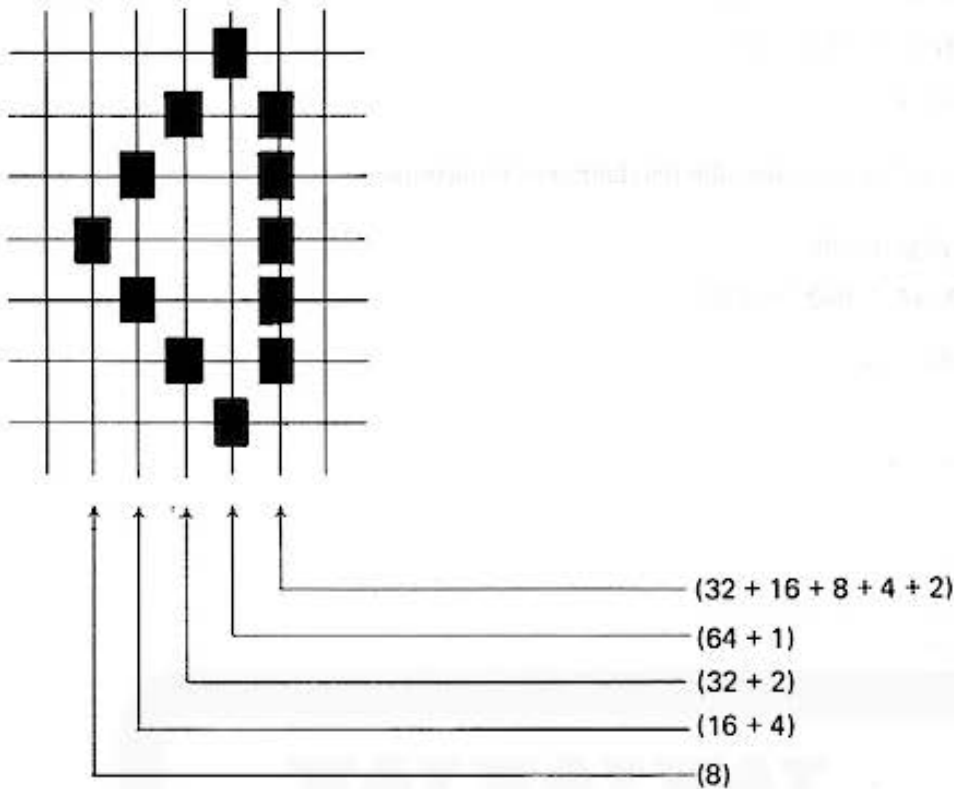
Dieses Beispiel zeigt auf der Anzeige alle darstellbaren Punktmuster.

```
(3) 10 GCURSOR 30
    20 GPRINT " 04027F0204 "
```

führt zur Anzeige



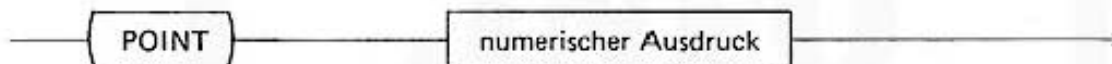
(4) GPRINT 8; 20; 34; 65; 62



POINT

Mit dieser Funktion können die Punktmuster der Anzeige ausgelesen werden.

Syntax



Ergebnis: numerischer Wert

Diese Funktion ist das Gegenstück zur GPRINT Anweisung. Mit der POINT-Funktion können Sie das Punktmuster der Anzeige auslesen.

Der numerische Ausdruck spezifiziert die horizontale Position der 7 * 156 Punktmatrix und kann somit Werte von $0 \leq \text{numerischer Ausdruck} \leq 155$ annehmen. Das Ergebnis ist das vertikale Punktmuster an der Position. Die Wertigkeit der einzelnen Punkte sind bei der GPRINT-Instruktion beschrieben.

Beispiel:

```

10 DIM Z (7)
20 PRINT "X"
30 FOR I = 0 TO 7
40 Z (I) = POINT I
50 NEXT I
60 GCURSOR 7
70 FOR I = 0 TO 7
80 GPRINT Z (I);
90 NEXT I
    
```

In der Schleife 30 – 50 erfolgt das Einlesen des Punktmusters vom Zeichen "X" und die Zuweisung zu der Feldvariablen Z. Mit der Schleife 70 – 90 wird dieses Zeichen an der neuen Position spezifiziert durch Zeile 60 wieder ausgegeben.

Fuchs du hast die Gans

~~Fuchs du hast die Gans~~

(2)

```

100:"A":WAIT 0:REM
      SCHRIFT INVE
      RTIEREN
110:PRINT "Fuchs d
      u hast die Gan
      s"
120:FOR I=0TO 155
130:A=POINT I
140:A=127-A
150:GCURSOR I
170:GPRINT ABS A
200:NEXT I
210:GOTO 210
    
```

Invertieren der Schrift auf der Anzeige.

CLS

Diese Instruktion löscht die Anzeige und setzt den Cursor in die Grundposition.

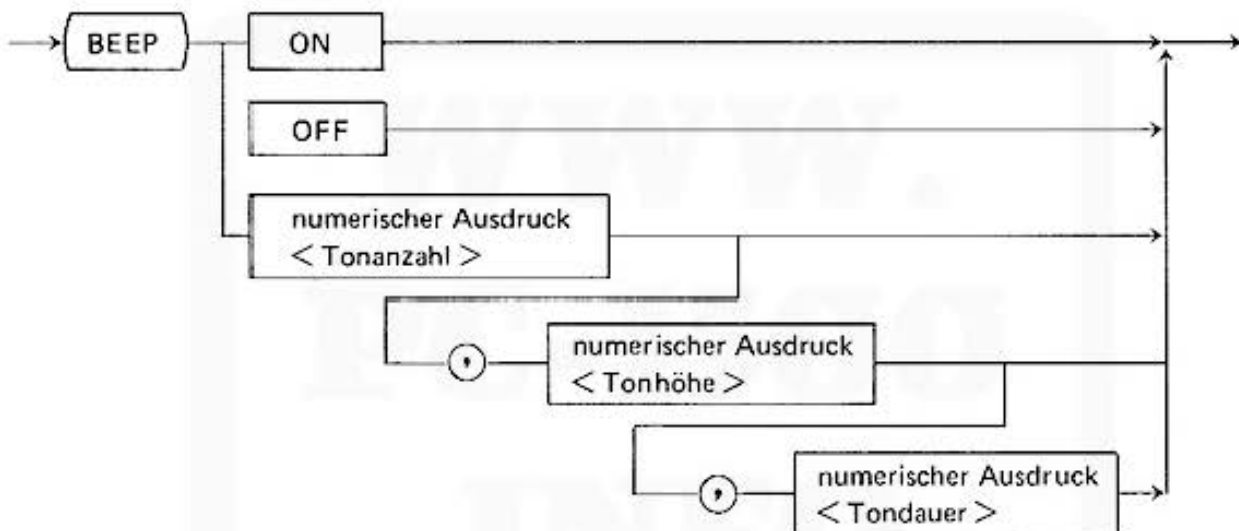
Syntax



BEEP

Mit dieser Instruktion wird der akustische Signalgeber gesteuert.

Syntax



Die BEEP ON Instruktion setzt den Signalgeber in Bereitschaft.

Die BEEP OFF Instruktion schaltet den Signalgeber aus, die folgenden BEEP Instruktionen sind unwirksam bis auf die BEEP ON Anweisung.

Mit der dritten Form der BEEP Instruktion können Sie programmgesteuert akustische Signale mit verschiedenen Parametern geben.

Das Signal besteht aus einer Folge von Tönen gleicher Frequenz unterbrochen von kurzen Ruhepausen.



Die drei Parameter des Signals sind: die Anzahl der Töne (Ausdruck < Tonanzahl >), die Frequenz des Tones (Ausdruck < Tonhöhe >). Die Anzahl der Töne und damit der Wert des numerischen Ausdrucks < Tonanzahl > darf variieren zwischen 0 und 65 535.

Die Tonhöhe wird bestimmt durch den Wert von dem Ausdruck < Tonhöhe >, der zwischen 0 und 255 liegen muß. Die Frequenz liegt dann im Intervall von ca. 230 – 7000 Hz. Der Ausdruck < Tondauer > darf die Werte 0 . . 65279 annehmen. Die Zeit in Sekunden ergibt sich aus < Tondauer + 256 > / Frequenz.

Hinweis:

Bei der Abspeicherung und beim Laden von Informationen vom Magnetband läßt sich mit dem Kommando BEEP OFF der Lautsprecher und damit die Ausgabe der Töne abschalten.

Beispiel:

- (1) 10 BEEP ON
 20 BEEP 1, 1, 1000 – Ausgabe eines kurzen Tones
- (2) s. INKEY\$ Funktion – Beispiel: Elektronische Orgel

Textfunktionen

Mit dem PC-1500 können Sie nicht nur rein numerische Aufgaben lösen, sondern auch Texte verarbeiten. Die im folgenden Kapitel vorgestellten Funktionen sollen Ihnen diese Arbeit erleichtern. So können Sie beispielsweise Zeichenfolgen aus Texten ausschneiden und zu neuen zusammensetzen.

Die Syntax beschreibt die Struktur Ihrer Parameterliste. Die Ergebnisse der Funktionen sind entweder Zeichenketten, die Textvariablen zugewiesen oder in Textausdrücken weiterverarbeitet werden können, oder numerische Werte, die sich in numerischen Ausdrücken verarbeiten oder numerischen Variablen zuweisen lassen.

ASC

Die ASC Funktion bestimmt den ASCII-Code von Zeichen.

Syntax



Ergebnis: numerischer Wert

Der ASCII-Code des ersten Zeichens vom Textausdruck ist der Ergebniswert. Die Umkehrfunktion zu ASC ist die Textfunktion CHR\$.

Beispiel:

- 1) 10 A = ASC "A"
 20 PRINT A

Auf der Anzeige erscheint 65, der ASCII-Code des Zeichens "A"

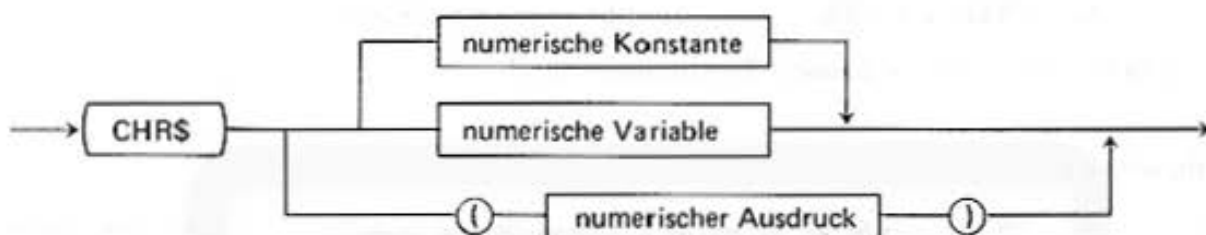
```
2) 10 SS = "SHARP"
    20 A = ASC SS
    30 PRINT A
```

Auf der Anzeige erscheint 83, der ASCII-Code des Zeichens "S", dem ersten Zeichen in der Textvariablen SS (siehe Tabelle im Anhang)

CHR\$

Die Textfunktion CHR\$ ermittelt aus dem ASCII-Code das zugehörige Zeichen.

Syntax



Ergebnis: Textzeichen

Der Wert des numerischen Ausdrucks im Intervall $0 \leq \text{Ausdruck} \leq 127$ definiert den ASCII-Code eines Zeichens. Das Funktionsergebnis ist das zugehörige Zeichen.

Die Funktion CHR\$ ist die Umkehrfunktion zur ASC Funktion.

Beispiel: Aufstellung einer Code-Tabelle

```
1) 5 PAUSE
    10 FOR I = 33 TO 127
    20 PAUSE I; "--- = ---"; CHR$ I
    30 NEXT I
```

Bei I = 65 erscheint auf der Anzeige

65--- = ---A

nicht alle Codes entsprechen darstellbaren Zeichen, wie Sie aus der Tabelle im Anhang ansehen können.

```
2) 5 AS = " "
    10 FOR I = 65 TO 80
    20 AS = AS + CHR$ I
    30 NEXT I
    40 PRINT AS
```

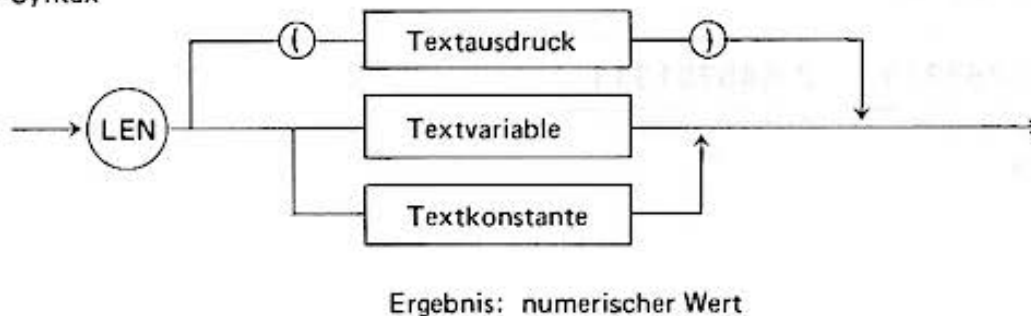
Auf der Anzeige erscheint

ABCDEFGHIJKLMN

LEN

Die LEN Funktion berechnet die Anzahl der Zeichen eines Textausdruckes.

Syntax



Die LEN Funktion zählt die Zeichen eines Textausdruckes, das Ergebnis ist die Zeichenanzahl.

Beispiel:

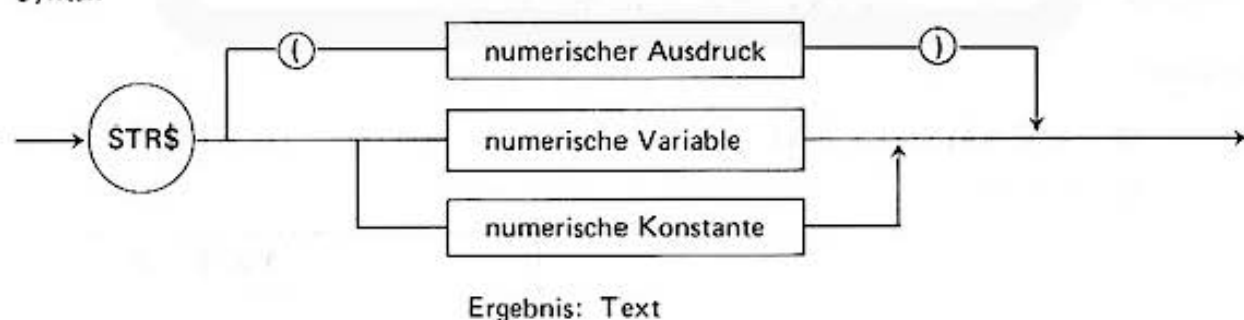
```
1) 10 A = LEN "SHARP"  
    20 SS = "HAMBURG"  
    30 B = LEN SS  
    40 PRINT A; B
```

Die Variablen haben nach dem Programmlauf die Werte A: 5 und B: 7

STR\$

Die Textfunktion STR\$ wandelt den Wert eines numerischen Ausdrucks in die dezimale Zeichenfolge um.

Syntax



Die Zeichenfolge, die bei der Ausgabe des numerischen Ausdrucks durch eine PRINT-Instruktion im Standardformat auf der Anzeige erscheint, ist das Ergebnis der Textfunktion STR\$.

Beispiel:

```
10 SS = STR$  $\sqrt{7}$   
20 PRINT SS,  $\sqrt{7}$ 
```


Bei der Ausgabe mit der Instruktion PRINT $\sqrt{7}$ erscheint auf der Anzeige:

2.645751311

Diese Zeichenfolge wird in Zeile 10 der Textvariablen S\$ zugewiesen.

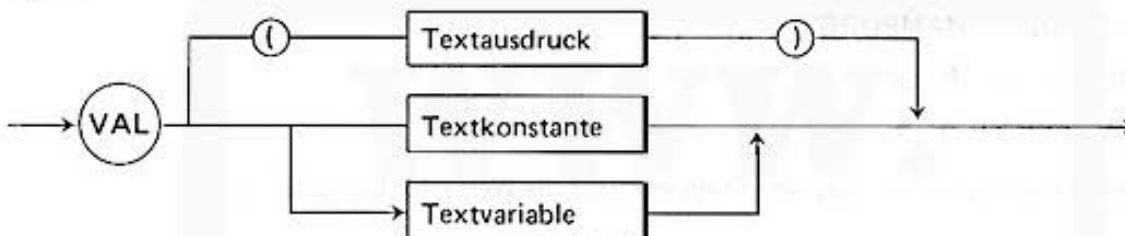
Ergebnis in Zeile 20:

2.645751311	_ _	2.645751311
$S\$$		$\sqrt{7}$

VAL

Die VAL Funktion berechnet den Wert von einer als Zeichenfolge angegebenen Zahl.

Syntax



Ergebnis: numerischer Wert

Die VAL Funktion berechnet die Wert einer Zahl. Diese Zahl kann in den üblichen Formaten in dem Textausdruck als Zeichenfolge angegeben sein.

VAL ist die Umkehrfunktion von STR\$.

Die Konvertierung beginnt mit dem ersten Zeichen des Textausdrucks und wird abgebrochen, wenn ein Zeichen auftritt, das nicht in der Liste {0 . . 9, . . , E} enthalten ist. Die Zeichen +, - sind nur als Vorzeichen zulässig, Zwischenräume werden überlesen.

Beispiele:

- 1) 10 A = VAL "12.3 E 12"
 20 PRINT A

1.23E 13

- 2) 10 A = VAL "1_2_3"
 20 PRINT A

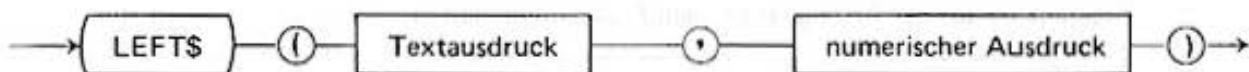
Der Variablen A wird in Zeile 10 der Wert 123 zugewiesen.

123

LEFT\$

Die Textfunktion LEFT\$ entnimmt aus einer Zeichenfolge die ersten Zeichen von links.

Syntax



Ergebnis: Text

Der Wert des numerischen Ausdrucks bestimmt die Anzahl der Zeichen, die von links aus der Zeichenkette des Textausdruckes entnommen werden und die Ergebniszeichenfolge bilden.

Beispiel:

```
1) 10 SS = "SHARP HAMBURG"  
20 TS = LEFT$ (SS, 5)  
30 PRINT TS
```

Aus der Zeichenkette von SS werden in Zeile 20 die ersten 5 Zeichen entnommen ("SHARP") und auf der Anzeige erscheint nach dem Programmablauf als Ergebnis:

SHARP

```
2) 10 PS = "8000 MUENCHEN"  
20 TS = LEFT$ (PS, 4)  
30 PRINT "PLZ"; TS
```

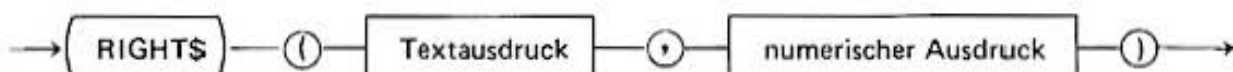
In Zeile 20 werden aus der Zeichenkette von PS die ersten 4 Zeichen entnommen ("8000").

PLZ8000

RIGHT\$

Die Textfunktion RIGHT\$ entnimmt aus einer Zeichenfolge die Zeichen vom rechten Ende.

Syntax



Ergebnis: Text

Wie bei der LEFT\$ Textfunktion bestimmt der Wert des numerischen Ausdrucks die Anzahl der Zeichen, die aus der Zeichenkette des Textausdruckes vom rechten Ende entnommen werden und die Ergebniszeichenfolge bilden.

Beispiele:

```
1) 10 P$ = "8000 MUENCHEN"
    20 T$ = RIGHT$ (P$, 8)
    30 PRINT "ORT _ "; T$
```

Aus der Zeichenkette von P\$ werden in Zeile 20 die letzten 8 Zeichen entnommen, als Ergebnis ist auf der Anzeige nach dem Programmablauf zu sehen:

ORT MUENCHEN

Ausgabe der Uhrzeit (Vergewissern Sie sich vor dem Lauf des Programms, daß Sie Ihre Computer-Uhr gesetzt haben).

<pre>10: "K": T=TIME 20: T=T-INT T 30: T\$=STR\$ T 40: IF LEN T\$<6 THEN LET T\$=T\$ +"0" 50: IF LEN T\$<6 THEN LET T\$=T\$ +"0" 60: T1\$=MID\$ (T\$, 3 , 2) 70: T2\$=RIGHT\$ (T\$, 2) 75: WAIT 0 80: PRINT T1\$;" MI N. UND ";T2\$;" SEC." 100: GOTO 10</pre>	<p>– Speichern der Zeit in T; Abzug von Monat und Tag</p> <p>– Ausgabe von T nach T\$</p> <p>– Anhängen von fehlenden Nullen</p> <p>– Ausblenden der Minutenziffern</p> <p>– Ausgabe von Minuten und Sekunden auf der Anzeige</p>
---	---

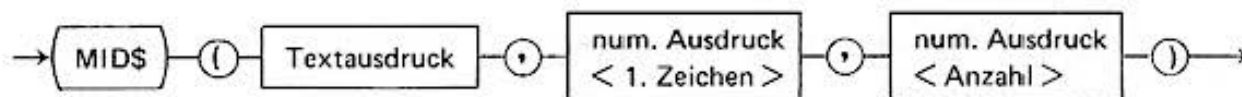
Auf der Anzeige erscheint mit Zeile 80:

12_MIN._UND_13_SEC.

MID\$

Die Textfunktion MID\$ entnimmt einer Zeichenkette den mittleren Teil als Ergebniszeichenkette.

Syntax



Ergebnis: Text

Aus der Zeichenkette des Textausdruckes wird der mittlere Teil entnommen. Der Wert vom numerischen Ausdruck <1. Zeichen> legt die Position innerhalb der Zeichenkette fest von der ab die Zeichen entnommen werden sollen. Der Wert vom Ausdruck <Anzahl> bestimmt die Anzahl der Zeichen, die entnommen werden.

Beispiel:

```

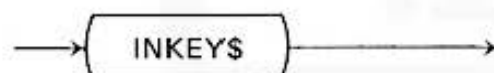
10  SS = "TEL. 049 - 40 - 23775-1"
20  TS = MID$ (SS, 10,2)
30  PRINT "VORW. HH"; "Ø"; TS
    
```

In Zeile 20 werden aus der Zeichenkette von SS beginnend mit Position 10 zwei Zeichen entnommen und der Textvariablen TS zugewiesen.

INKEY\$

Die Zeichenfunktion INKEY\$ liefert das Zeichen einer Taste als Funktionswert.

Syntax:



Ergebnis: Text

Trifft der Programmablauf auf die INKEY\$ Zeichenfunktion, so wird die Ausführung nicht unterbrochen. Das im Moment eingetastete Zeichen wird dem Ergebnis zugewiesen. Wird keine Taste betätigt, so erhält INKEY\$ das ASCII-Code Zeichen 'NULL'.

- Bemerkungen:
- Die **ENTER** Taste entspricht beispielsweise dem ASCII-Code Zeichen 13. Für die obere Reihe der Funktionstasten $F_1 \dots F_6$ gilt, ihr Zeichen hat einen um 16 kleineren ASCII-Code als das über ihnen gedruckte rote Zeichen.
Beispiel: F_1 (!) CHR\$ (ASC "!" - 16)
 - Vorsicht, wenn die INKEY\$ Funktion gleich zu Beginn eines Programmes ausgeführt wird, ist die **ENTER** -Taste vom Programmstart noch gedrückt confirm.
GO TO **ENTER** oder RUN **ENTER**
INKEY\$ wird dann die **ENTER** Taste einlesen

Beispiel:

```
(1) 10  A$ = "  "
    20  A$ = INKEY$
    30  IF A$ = " E " THEN 1000
    40  IF A$ = " S " THEN 60
    50  GOTO 20
    60  PRINT A$, "START < Programm >"
    700 GOTO 20
    1000 PRINT "ENDE"
    1100 END
```

Mit der Taste **S** wird ein Programmdurchlauf gestartet, mit **E** wird das Programm beendet. Nach jedem einzelnen Programmdurchlauf muß es durch **S** wieder neu gestartet werden.

PC-1500 als elektronische Orgel

```
(2) 300: "B": I=ASC
      INKEY$
    310: IF I=0 THEN 300
    320: BEEP 1, I, 6000/
      (1+2)
    330: GOTO 300
```

- Bestimmung des ASCII-Codes aus der gedrückten Taste
- Ist keine Taste gedrückt?
- Ausgabe der Tonfolge

Jeder Taste ist ein Ton zugeordnet.

PC-1500
INFO

17. Der RESERVE-Mode

In der oberen Reihe hat der PC-1500 sechs Tasten, deren Hauptfunktionen nicht beschriftet sind. Diese Tasten können vom Benutzer im Reserve-Mode programmiert, das heißt mit einer neuen Bedeutung belegt werden. Dadurch ist es möglich, häufig benutzte Ausdrücke durch einen Tastendruck einzugeben.

Der Rechner hat drei verschiedene Ebenen für Funktionstasten, so daß insgesamt 18 verschiedene Texte (RESERVE-Ausdrücke) gespeichert werden können. Das Umschalten von einem Zustand in den nächsten geschieht mit der Taste . Auf der Anzeige wird der aktuelle Zustand durch I, II oder III angezeigt.

Eine Tastenbelegung wird folgendermaßen geändert:

1. Mit wird der Reserve-Mode angewählt.
2. Mit wird die gewünschte Ebene angewählt.
3. Die zu belegende Taste wird gedrückt. Auf der Anzeige erscheint je nach Taste $F_1 : \dots$ bis $F_6 : \dots$
4. Der Text kann eingegeben werden. Ist bereits ein Text gespeichert, kann dieser mit den Editierfunktionen verändert werden. Mit wird die Eingabe abgeschlossen. Soll Bestandteil des Textes sein, so muß das Zeichen @ als Platzhalter für eingesetzt werden.

Beispiel:

\$
1)

F_4 : RUN 1000

Nach dieser Eingabe wird im RUN-Mode ein Programm bei Zeile 1000 gestartet durch \$

\$
2)

F_4 : RUN 1000 @

Das Programm wird ab Zeile 1000 gestartet durch \$ Das ist im Text durch das @ schon enthalten.


Bemerkung:

Mit dem Kommando NEW im Reserve-Mode können alle Text gelöscht werden. Die Tasten haben dann dieselbe Funktion wird nach .

Der Speicherbereich für die programmierten Texte ist begrenzt. Versucht man, zu viele Texte zu speichern, meldet der Rechner einen Fehler (ERROR 13). Nach einem ALL RESET sind die Texte undefiniert und möglicherweise sehr lang. Löschen Sie mit NEW.


17.1 Beschriftung der reservierbaren Tasten

Im Reserve-Mode kann zu jeder Tastenebene ein Text gespeichert werden, der die Funktion der Tasten erklärt. Dieser Text kann nur für alle sechs Tasten gleichzeitig eingegeben werden, durch:


1. Anwählen des gewünschten Tastenzustands durch .
2. Eingabe des Textes als Textkonstante, d.h. in Anführungsstrichen.

Beispiel:

"SIN COS ASN ACS TAN ATN"


Beim Wechseln des Tastenzustands wird diese Beschriftung in alle Modes automatisch auf der Anzeige gezeigt. Sie kann außerdem durch die Taste  (Recall) jederzeit aufgerufen werden. Durch nochmaliges Drücken der RCL-Taste wird die letzte Eingabezeile auf die Anzeige zurückgerufen.

17.2 Fest eingebaute Abkürzungen:




Neben den programmierbaren Tasten können einige häufig auftretende BASIC-Schlüsselwörter durch die Tastenfolge  und einen Buchstaben der 2. Tastenreihe abgekürzt werden.

Die Bedeutungen sind:

 A = INPUT	 U = CSAVE
 W = PRINT	 I = CLOAD
 E = USING	 O = MERGE
 R = GOTO	
 T = GOSUB	
 Y = RETURN	 P = LIST

Die Funktionen  U, I, O funktionieren nur bei angeschlossenem Kassetteninterface CE-150, sonst wird ein ~ ausgegeben.

Hinweis:

- 1) Da diese Begriffe im Rechner kodiert als Schlüsselwörter für BASIC-Instruktionen gespeichert werden, ist es nicht möglich, aus    ein GPRINT zusammenzusetzen.
- 2) Eine weitere Möglichkeit zur Reduzierung der Tipparbeit ist durch die Verwendung von Abkürzungen gegeben. Viele BASIC Schlüsselwörter lassen sich abkürzen, z.B. GPRINT durch GP.; eine Liste aller Abkürzungen befindet sich im Anhang.
- 3) Die beigelegte Schablone kann über die Tastatur gelegt werden und beschriftet die Tastendoppelbelegung.

18. CE-150 (Option)

Plotter/Drucker mit integriertem Kassetten-Interface

18.1 Einleitung

Der CE-150 ist ein Zusatzgerät zum PC-1500 und dient als weitere Datenausgabeeinheit zum Ausdrucken von

- a) Programmlisten
- b) Datenlisten
- c) Ergebnisausgaben bei Programmausführung
- d) Ergebnisausgaben bei manuellem Rechnen
- e) Graphiken
- f) Funktionskurven etc.

Darüberhinaus können an den CE-150 bis zu zwei Kassettenrekorder angeschlossen werden, die sowohl manuell als auch unter Programmkontrolle Informationen auf einer Kassette abspeichern und auf Anweisung in den Computer zurückladen.

18.2 Betriebshinweise

- a) Schützen Sie den CE-150 vor Staub, Feuchtigkeit und allzugroßen Temperaturschwankungen.
- b) Zur Reinigung dient ein weiches, trockenes Tuch; keine Reinigungs- oder Lösungsmittel verwenden.
- c) Die Farbpatronen sind mit Tinte gefüllt und können austrocknen. Bei längerer Nichtbenutzung des Druckers sollten die Patronen deshalb aus dem Gerät entfernt und in dem Plastikbehälter aufbewahrt werden.
- d) Achten Sie stets darauf, daß sich im Druckwerk Papier befindet. Das Betreiben des Druckers ohne Papier führt zur Verschmutzung und Verhärtung der Gummiwalze, wodurch der Papiertransport erheblich gestört werden kann.
- e) Drehen Sie niemals manuell an der Farbpatronentrommel.

18.3 Anschluß des CE-150 an den Computer PC-1500

Beim Anschluß des CE-150 an den PC-1500 verfahren Sie unbedingt wie nachstehend beschrieben.

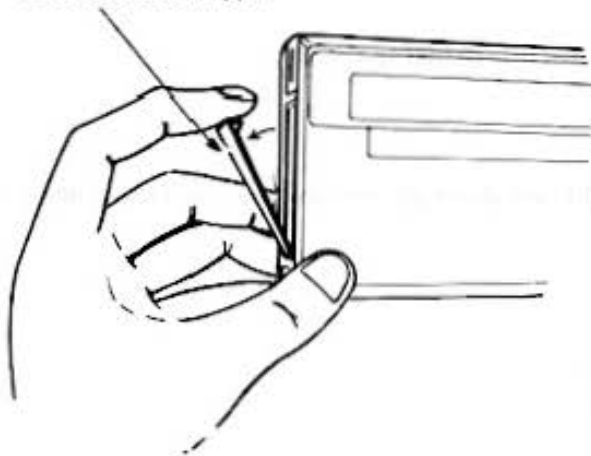
- a) Sie schalten den Computer PC-1500 aus.

Achtung: Es ist unbedingt notwendig, daß der Computer beim Ankoppeln ausgeschaltet ist. Wird dieses nicht beachtet, kann der PC-1500 "abstürzen". Dadurch werden alle Bedienungstasten blockiert und der Computer kann nicht mehr bedient werden.

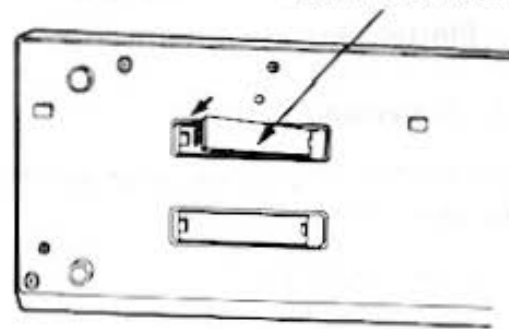
Durch Betätigen des "All-Reset" Schalters bei gleichzeitigem Drücken der ON-Taste kann der Computer initialisiert werden, bzw. müssen die Batterien für längere Zeit ausgebaut werden. (Siehe Kapitel)

- b) Sie entfernen die Anschlußabdeckkappe und rasten sie auf der Unterseite des CE-150 ein. (Abb.)

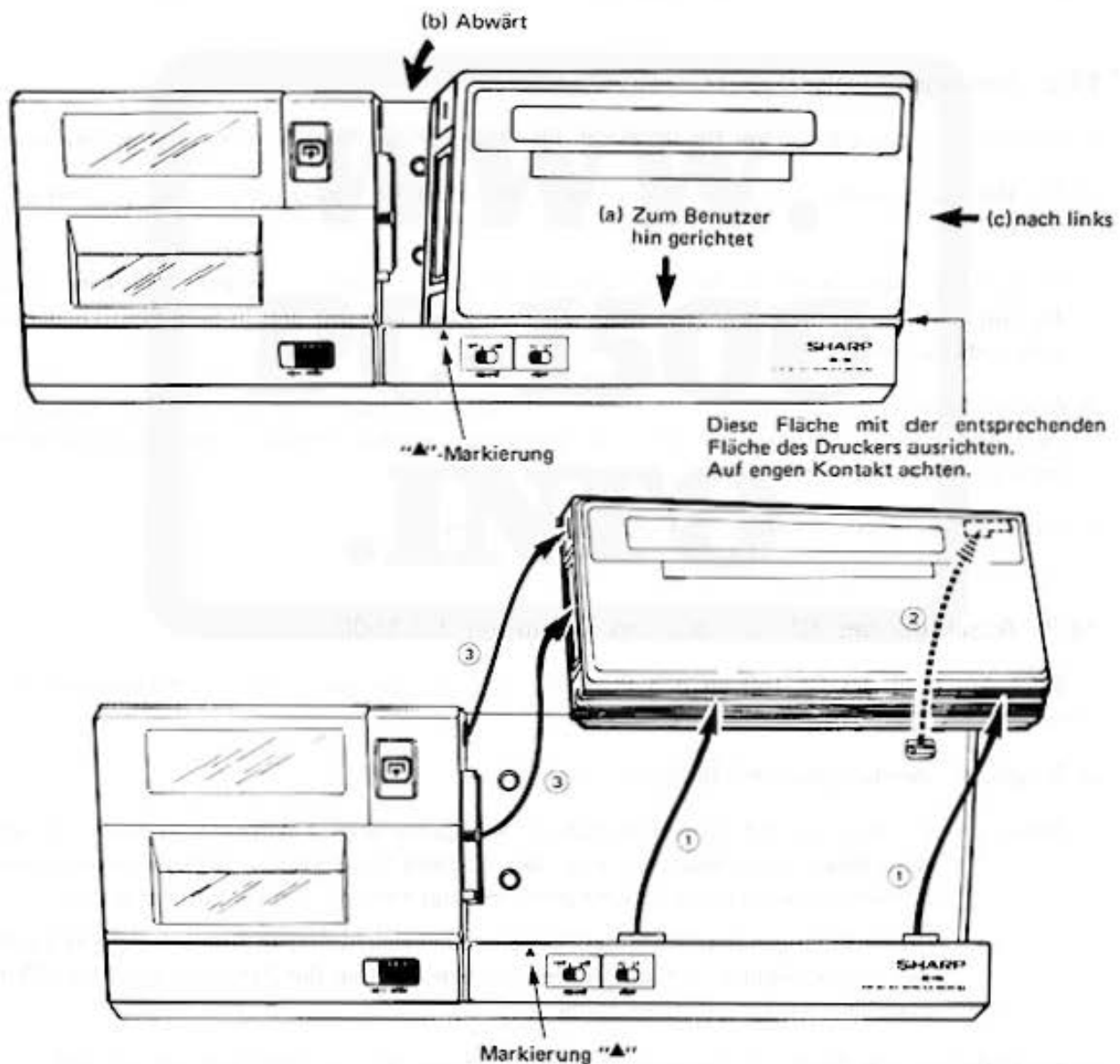
Anschlußabdeckkappe



Hier einrasten lassen



- c) Sie legen den Computer so auf den CE-150, daß die linke Seite des PC-1500 mit der Markierung auf dem CE-150 fluchtet. Dabei sollen die beiden Führungsstege des CE-150 in der Nut des PC-1500 liegen (Pos. 1) und der Haltesttift des CE-150 in die entsprechende Aussparung am PC-1500 eingreifen (Pos. 2).



- d) Durch seitlichen Druck wird der Computer nach links geschoben und die elektrische Verbindung hergestellt. Wenden Sie dabei niemals Gewalt an. Sollte sich ein Widerstand zeigen, verschiebt man den Computer an der vom Pfeil markierten Position geringfügig, bis sich die Verbindung leicht herstellen läßt.

18.4 Betriebsvorbereitung

18.4.1 Einschalten des CE-150

Der CE-150 besitzt keinen eigenen EIN-AUS-Schalter, sondern wird vom PC-1500 kontrolliert.

Beim Drücken der ON-Taste des PC-1500 wird auch automatisch der Drucker aktiviert. Je nachdem, in welchem Status sich Drucker und PC-1500 befinden, kann folgendes eintreten:

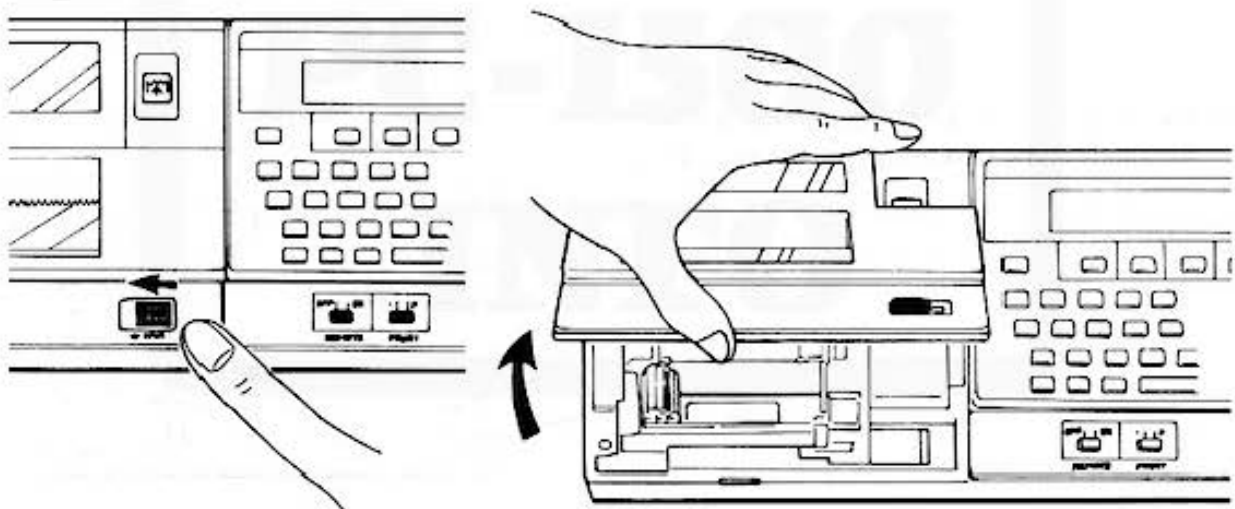
- Das Druckwerk startet mit der Einstellung der Farbpatronentrommel auf die Farbe "Schwarz", einem Papiervorschub und in der Anzeige erscheint das Symbol ">".
- Das Druckwerk startet nicht und in der Anzeige erscheint das Symbol ">".
- In der Anzeige erscheint "ERROR 80" oder "ERROR 78".
- In der Anzeige erscheint
: CHECK 6 oder NEW Ø ? : CHECK 6
(Siehe Kapitel Stromversorgung)
- Das Druckwerk startet nicht, die Anzeige schaltet sich nicht ein, die Tasten sind blockiert und aus dem Computer ist ein Ticken zu hören.

Ist Fall c, d oder e eingetreten, beheben Sie die Störungen gemäß den genannten Anweisungen. In den Fällen a oder b kann die Betriebsvorbereitung fortgesetzt werden.

18.4.2 Auswechseln der Papierrolle

(Siehe auch Bedienungsanleitung CE-150)

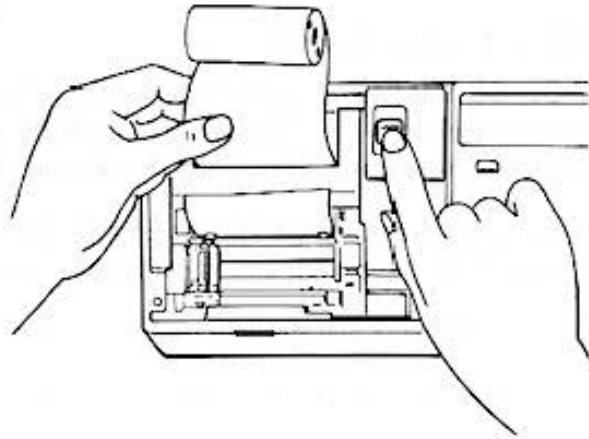
- Sie entriegeln und entfernen die Druckerabdeckung.



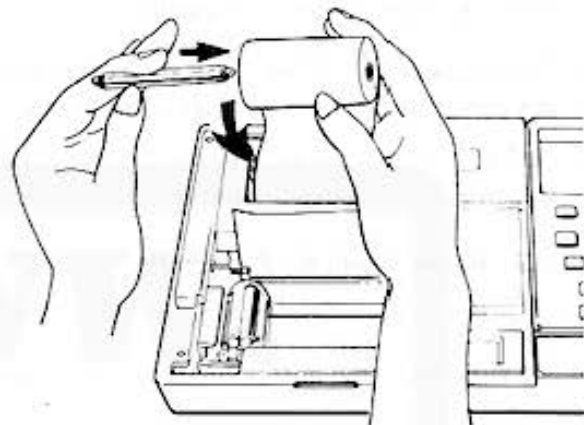
- Sie begradigen den Anfang der Papierrolle, führen ihn in den Papierschlitze ein.



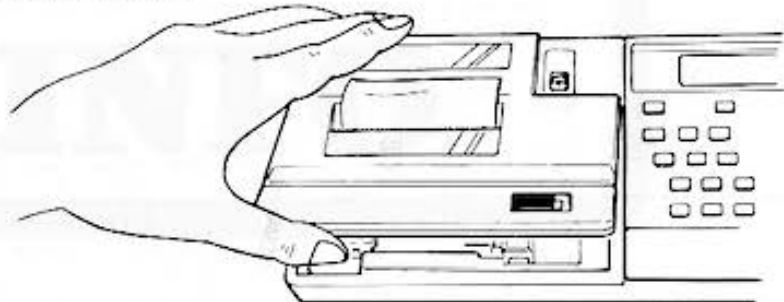
- c) Sie drücken die Papiertransporttaste solange bis ca. 5 cm abgerollt sind (dabei muß der Computer eingeschaltet sein).



- d) Stecken Sie die Achse durch die Rolle und legen Sie sie in die Papierwanne.



- e) Führen Sie den Papieranfang durch die Druckwerkabdeckung und setzen Sie diese wieder auf.

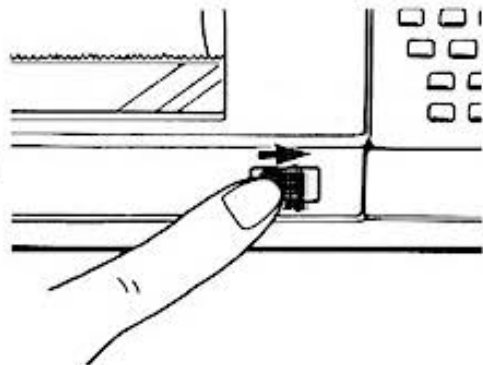


- f) Verriegeln Sie die Abdeckung.

18.4.3 Auswechseln der Farbpatronen

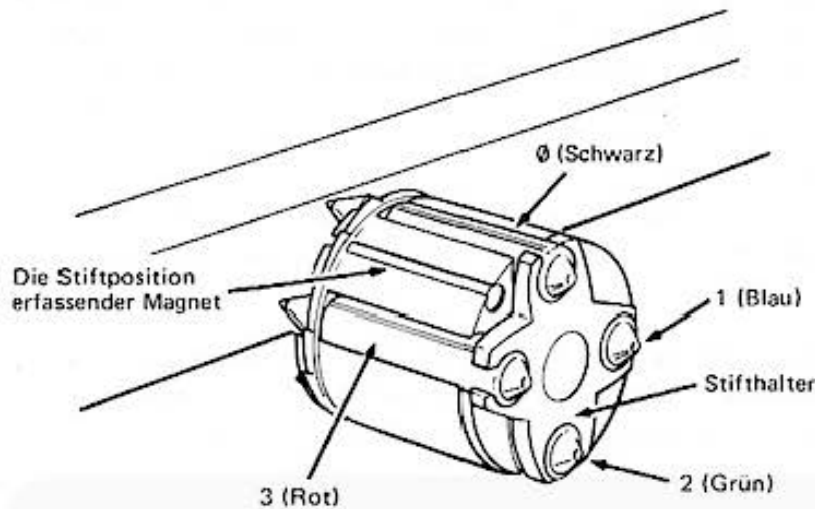
(Einzelheiten siehe Bedienungsanleitung CE-150)

Der Drucker kann mit vier Farben drucken.
Die Farbauswahl erfolgt mit dem Kommando
COLOR 0 für schwarz
COLOR 1 für blau
COLOR 2 für grün
COLOR 3 für rot

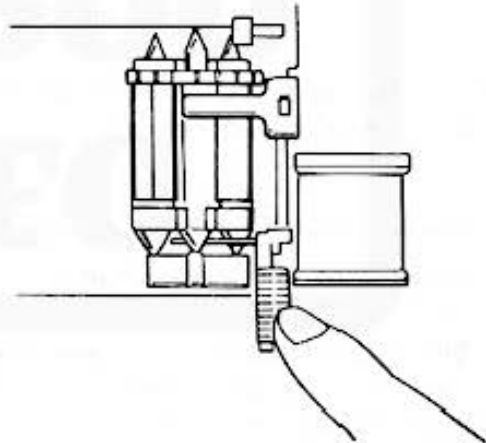


wobei die ausgewählte Farbe durch rechnergesteuertes Linksdrehen der Trommel in die Druckposition gebracht wird.

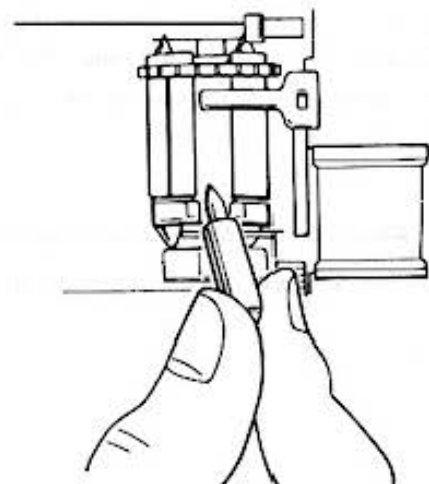
Damit mit dem Kommando COLOR auch die zugeordnete Farbe ausgewählt werden kann, müssen die Farbpatronen, ausgehend vom Positionierungsmagnet in einer bestimmten Folge eingesetzt werden. Um Verwechslungen zu vermeiden, ist die Lochscheibe außerdem mit den entsprechenden Farbmarkierungen zu versehen.




- Sie entriegeln und entfernen die Druckwerkabdeckung (Siehe Auswechseln der Papierrolle).
- Sie geben das Kommando COLOR 3 **ENTER**. Dabei dreht sich die Trommel auf "ROT".
- Sie betätigen bei gedrückter **Ø**-Taste (Null) den Papiervorschub . Die Trommel dreht auf "SCHWARZ" und läuft an den rechten Rand.
- Sofern sich bereits eine Farbpatrone in der Trommel befindet, kann sie durch leichtes Ziehen am Auswurfhebel aus der Trommel gelöst und entfernt werden. Durch leichten Druck mit einem Finger auf die Patrone vermeiden Sie, daß sie aus der Trommel katapultiert wird und ins Gerät fällt.



- Setzen Sie eine neue schwarze Patrone in die Trommel. Beachten Sie, daß die Spitze der Patrone in das obere Loch der Lochscheibe greift. Durch sanften Druck auf das hintere Ende der Patrone rastet sie sicher in die Trommel ein.



- f) Sie drücken die Papiervorschub-Taste  .

Die Trommel läuft nach links, dreht auf die Position "BLAU". Sie ersetzen die Patrone "BLAU".

Durch wiederholten Druck auf die Papiervorschubtaste werden die Patronen "GRÜN" und "ROT" analog zu "BLAU" ausgewechselt.

- g) Nachdem die Patronen eingesetzt bzw. ersetzt sind, wird das Druckwerk durch Betätigen der Papiervorschubtaste bei gedrückter **CL** -Taste in die Ruhestellung gebracht; anschließend montieren Sie die Druckwerkabdeckung. Das Druckwerk ist nun betriebsbereit.

18.4.4 Druckwerktest

Durch Eingabe von TEST **ENTER** wird das Druckwerk auf einwandfreies Funktionieren und richtige Farbauswahl geprüft. Das Testprogramm druckt vier Karos in der Farbfolge schwarz, blau, grün, rot.

Hinweis: Neue oder für längere Zeit nicht benutzte Patronen müssen sich erst "einlaufen" bevor sie sauber schreiben. Daher sollte der Druckwerktest gegebenenfalls wiederholt werden. Beim Druckbetrieb sollen stets alle vier Patronen eingesetzt sein.

18.4.5 Farbpatronen und ihre Handhabung

Die Farbpatronen haben das gleiche Konstruktionsprinzip wie Tintenkugelschreiber, nur sind sie wesentlich kleiner. Sie sind in zwei Zusammenstellungen unter folgender Typenbezeichnung erhältlich

- a) EA-850B (4 Stück Patronen, Farbe schwarz)
- b) EA-850C (4 Stück Patronen in den Farben schwarz, blau, grün, rot).

Die Patronen werden in einem Kunststoffbehälter geliefert, der sie vor dem Austrocknen schützt.

Wird das Druckwerk längere Zeit nicht benutzt, sollten die Patronen ausgebaut und in dem Kunststoffbehälter aufbewahrt werden.

18.4.6 Stromversorgung und Fehlermeldung 80/78

Der CE-150 wird mit wiederaufladbaren Ni-Cd-Batterien betrieben. Mit voll aufgeladenen Batterien kann der CE-150 im Druckbetrieb ca. 50 Minuten unabhängig vom Netzstrom verwendet werden. Sind die Batterien erschöpft, wird ERROR 80 gemeldet.

Wenn nach Löschung der Fehlermeldung 80 (CL-Taste) der Druckbetrieb wieder aufgenommen wird, erscheint Fehlermeldung 78.

Der Druckkopf ist auf einer irregulären Position stehengeblieben und muß zunächst in die Anfangsstellung gebracht werden. Dazu wird der Computer kurz aus- und eingeschaltet. Der Druckkopf geht auf seine Anfangsstellung. Zur Wiederaufladung der Batterien wird das Netz-Lade-Gerät EA-150 angeschlossen. Der Druckbetrieb kann danach wieder aufgenommen werden. Eine Volladung der Batterien dauert ca. 15 Stunden und verlängert sich erheblich, wenn gleichzeitig der Drucker benutzt wird.

18.4.7 Anschluß des Netz-Ladegeräts EA-150

Schließen Sie das Ladegerät gemäß der untenstehenden Abbildung an.

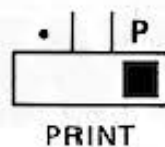


18.5 Druckerausgaben

Die in dem folgenden Kapitel ausgeführten Instruktionen ermöglichen Ihnen die Ausgabe auf dem Drucker, falls Sie über diese Option verfügen.

So können Sie Texte vierfarbig in neun verschiedenen Zeichengrößen programmgesteuert auf den Drucker schreiben. Graphische Zeichnungen in mehreren Farben können erstellt werden.

Ihren PC-1500 können Sie auf dem Drucker Ihre Rechnungen protokollieren lassen. Sie müssen dazu nur den PRINT-Schalter in die Position P bringen.



Die Protokollführung wird unterdrückt, wenn sich der Schalter in der I – Position befindet. Nur explizite Druckerausgaben z.B. über die LPRINT-Anweisung werden dann noch gedruckt.

Auf dem Papierstreifen erscheinen bei der Protokollführung die gleichen Zeichenfolgen in gleicher Form und analog positioniert wie auf der Anzeige. Direkt eingegebene Instruktionen werden allerdings nicht mit protokolliert. Der Druck beginnt erst, nachdem Sie die **ENTER**-Taste betätigt haben, vorher können Sie also noch die Eingabe wie normal korrigieren. Das folgende Bild zeigt eine Protokollführung als Beispiel:

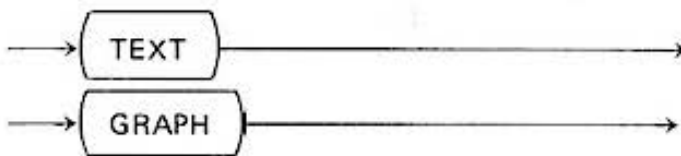
12000*.056	672
780+25.6	805.6
SIN 45	
7.071067812E-01	
TX=1.2356	1.2356
π	3.141592654
P=20000*TX	24712
P/32	772.25

Beim Protokollieren wird die zuletzt angewählte Farbe und Zeichengröße verwendet. Das größte Zeichenformat ist jedoch CSIZE 2. Auf diese wird eventuell automatisch zurückgeschaltet.

TEXT / GRAPH

Steuerung der Druckerbetriebsart.

Syntax

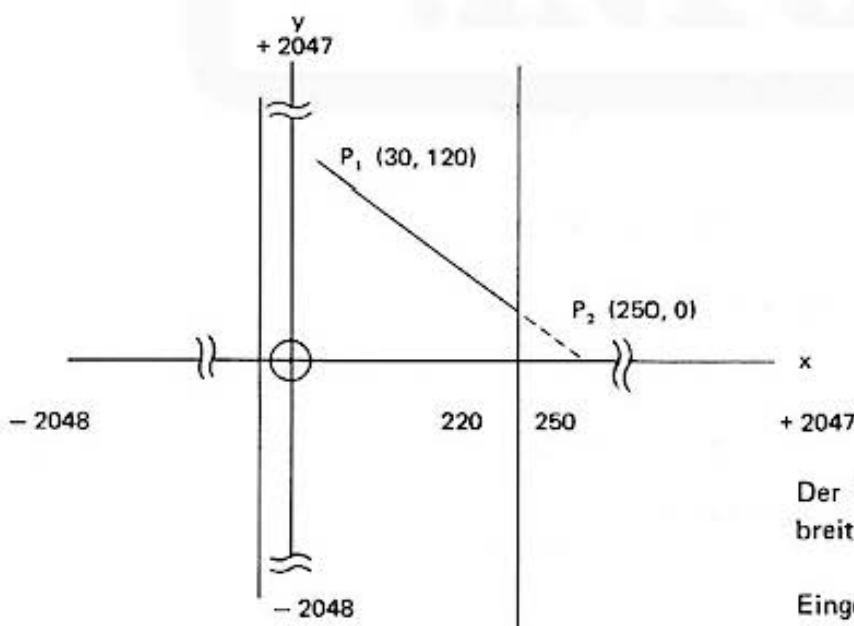


Der Drucker kann in den Betriebsarten (Modes) TEXT oder GRAPH betrieben werden. Die beiden Instruktionen wählen die jeweilige Betriebsart an.

Die folgenden Druckanweisungen sind entweder nur im GRAPHmode, nur im TEXTmode oder in beiden zulässig. Dies ist bei der Beschreibung jeweils vermerkt. Benutzt man eine Instruktion im falschen Mode, so führt dieses zu einem Fehler (ERROR 73).

Der TEXT- Mode wird zum Drucken von alphanumerischen Zeichen benutzt. Es können 9 verschiedene Zeichengrößen angewählt werden. Die Breite des Papierstreifens wird in Spalten aufgeteilt; die Anzahl der Spalten ist abhängig von der Zeichengröße. Sollen Texte ausgegeben werden, die mehr Zeichen enthalten als Spalten vorhanden sind, so werden die folgenden Zeilen automatisch als Fortsetzungszeilen benutzt. Zur Gleiderung der Textinformation können Tabulieranweisungen gegeben werden. Nach dem Einschalten des TEXT-Modes befindet sich der Schreibkopf am linken Papierrand.

Der GRAPH-Mode wird für die Ausgabe von Zeichnungen benutzt. Es gibt Anweisungen für durchgezogene und gestrichelte Linien. Start- und Endpunkte der Linien sind auf ein Koordinatensystem bezogen. Die feinste Einheit in der sich der Schreibkopf bewegen läßt, sind 0,2 mm in x- und y-Richtung. Das virtuelle Koordinatensystem ist in x-Richtung mit $-2048 \leq x \leq 2047$ Einheiten größer als der Papierstreifen. Das Papier enthält somit nur einen Ausschnitt. Nach der Anwahl des GRAPH-Modes wird der Schreibkopf an den linken Papierrand gefahren und dort der Ursprung des Koordinatensystems festgelegt.



Der Papierstreifen hat eine Druckbreite von 220 Einheiten.

Eingezeichnet ist eine Linie von P1 (30, 120) nach P2 (250, 0). Nur die durchgezogene Linie ist auf dem Papierstreifen sichtbar.

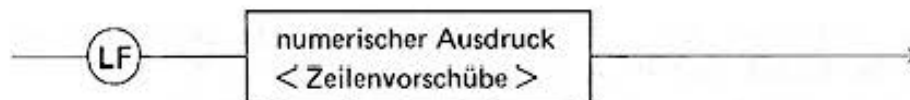
Bemerkung:

Nach dem Einschalten befindet sich der PC-1500 im Text-Mode. Er muß über die **[OFF]** -Taste ausgeschaltet worden sein und nicht über die Energiesparabschaltung.

LF (Textmode)

Die LF-Instruktion steuert den Papiervorschub.

Syntax



Der numerische Ausdruck legt fest, um wieviele Zeilenhöhen entsprechend der Zeichengröße das Papier vorwärts (positiver Wert) oder rückwärts (negativer Wert) bewegt wird. Die Zeichengröße ist mit der CSIZE-Instruktion festgelegt worden.

Bei der Rückwärtsbewegung lassen sich maximal 10 cm wieder aufrollen.

Damit das Papierende nicht eingezogen und somit auf die Rolle geschrieben wird, besitzt der Drucker intern einen Zähler, der automatisch mit jedem Zeilenvorschub inkrementiert wird. Der höchste Zählerstand ist 24.

Bei dem Aufrollen wird dieser Zähler entsprechend dekrementiert.

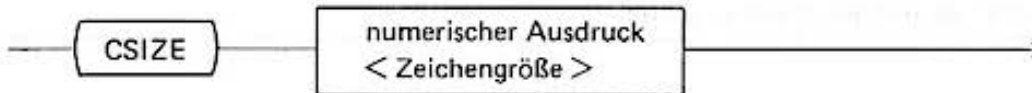
Versucht man bei einem Zählerstand von 0 das Papier rückwärts zu bewegen, wird ein Fehler (ERROR 71) ausgegeben.

Beispiel:

10	LF 5	AUSDRUCK:
20	LPRINT "ENDE"	
30	LF -2	
40	LPRINT "START"	START
		ENDE

CSIZE (Textmode)

Die CSIZE Instruktion legt die Zeichengröße fest.

Syntax

Der numerische Ausdruck darf die Werte 1, 2 bis 9 annehmen und bestimmt damit die Zeichengröße für die folgenden Schreib- und Steueroperationen. Der folgenden Tabelle können Sie Höhe, Breite des Zeichens und Anzahl von Schreibstellen in Abhängigkeit des Wertes entnehmen.

	1	2	3	4	5	6	7	8	9
Schreibstellen	36	18	12	9	7	6	5	4	4
Zeichenhöhe (mm)	1.2	2.4	3.6	4.8	6.0	7.2	8.4	9.6	10.8
Zeichenbreite (mm)	0.8	1.6	2.4	3.2	4.0	4.8	5.6	6.4	7.2

Beispiel:

```

10  FOR I = 1 TO 9
20  CSIZE I
30  LPRINT "A";
40  NEXT I
  
```

Ausdruck:

TEST

Die TEST-Instruktion überprüft die Druckerfunktionen.

Syntax



Die TEST Instruktion zeichnet in den vier Farben 5 mm * 5 mm große Quadrate auf das Papier. Sie können daraus die Reihenfolge der Farbpatronen, die Farbsättigung ansehen und überprüfen, ob Farbpatronen eingetrocknet oder leer sind.

Hinweis:

Die richtige Farbreihenfolge ist wichtig für die programmgesteuerte Farbwahl.

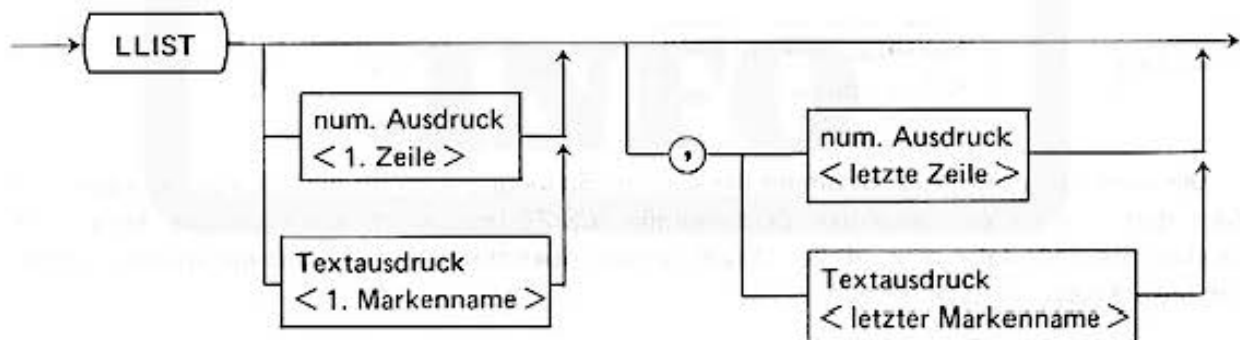
Beispiel:

TEST

LLIST

Mit dieser Instruktion wird das Programm auf dem Drucker gelistet.

Syntax



Die LLIST-Instruktion hat die gleiche Funktion wie die LIST-Anweisung. Das Programm wird jedoch auf dem Drucker und nicht auf der Anzeige aufgelistet.

Die ersten Ausdrücke spezifizieren die Zeile, bei der mit dem Ausdruck begonnen wird. Es kann dabei eine Zeilennummer gewählt werden oder ein Markenname, wie er auch als Sprungzeile in der GOTO-Anweisung verwendet wird.

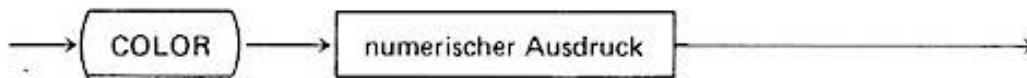
Die zweiten Ausdrücke definieren die letzte Zeile, mit der die Auflistung beendet werden soll. Die Zeile wird markiert durch die Zeilennummer oder den Markennamen.

Werden keine Ausdrücke angegeben, so werden alle gespeicherten Programmzeilen ausgedruckt. Fehlt der erste, wird von der ersten Programmzeile bis zur spezifizierten aufgelistet, fehlt der zweite, so wird nur diese eine Zeile gedruckt.

COLOR (Text-Mode, Graph-Mode)

Die COLOR Instruktion legt die Farbe für die folgenden Schreibanweisungen fest.

Syntax



Die Farbe für die folgenden Schreibanweisungen wird durch den Wert des numerischen Ausdrucks festgelegt. Zulässig sind die Werte 0, 1, 2, 3. Die Zuordnung zu den Farben Schwarz, Blau, Grün, Rot können Sie mit der TEST-Instruktion überprüfen. Das erste Quadrat am linken Rand entspricht dem Farbwert 0, das folgende 1 etc.

Nach dem Einschalten wird die Farbe mit dem Wert "0" ausgewählt.

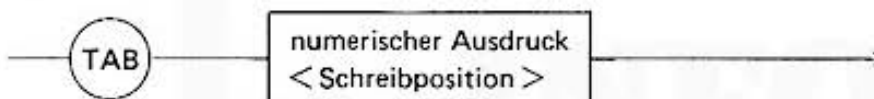
Beispiel:

```
10 COLOR 3 : LPRINT "ROT"
20 COLOR 1 : LPRINT "BLAU"
```

TAB (Text-Mode)

Die TAB Instruktion definiert die Schreibposition.

Syntax



Der numerische Ausdruck bestimmt die absolute Schreibposition für die folgende Ausgabe. Der Wert darf je nach der gewählten Zeichengröße (CSIZE-Instruktion) die maximale Anzahl von Zeichen pro Zeile (also max. 36 bei CSIZE 1) nicht überschreiten. Die Schreibpositionen werden von links 0 nach rechts gezählt.

Mit der TAB Instruktion fährt der Schreibkopf an die spezifizierte Position. (Die Wirkung der Instruktion ist ähnlich der CURSOR-Anweisung für die Anzeige).

Beispiel: (Standard-Zeichengröße)

```
10 FOR I = 16 TO 1 STEP -2
20 TAB I
30 LPRINT "A";
40 TAB (16-I)
50 LPRINT "B"
60 NEXT I
```

Das Ergebnis sieht dann folgendermaßen aus:

Der Schreibkopf fährt auf Position 16 (Zeile 20), druckt das "A", fährt in der gleichen Druckerzeile auf Position 0 (Zeile 40), druckt das "B" und rückt die Zeile vor. Mit dem um 2 verringerten I führt das Programm dann die analogen Aktionen durch.

```

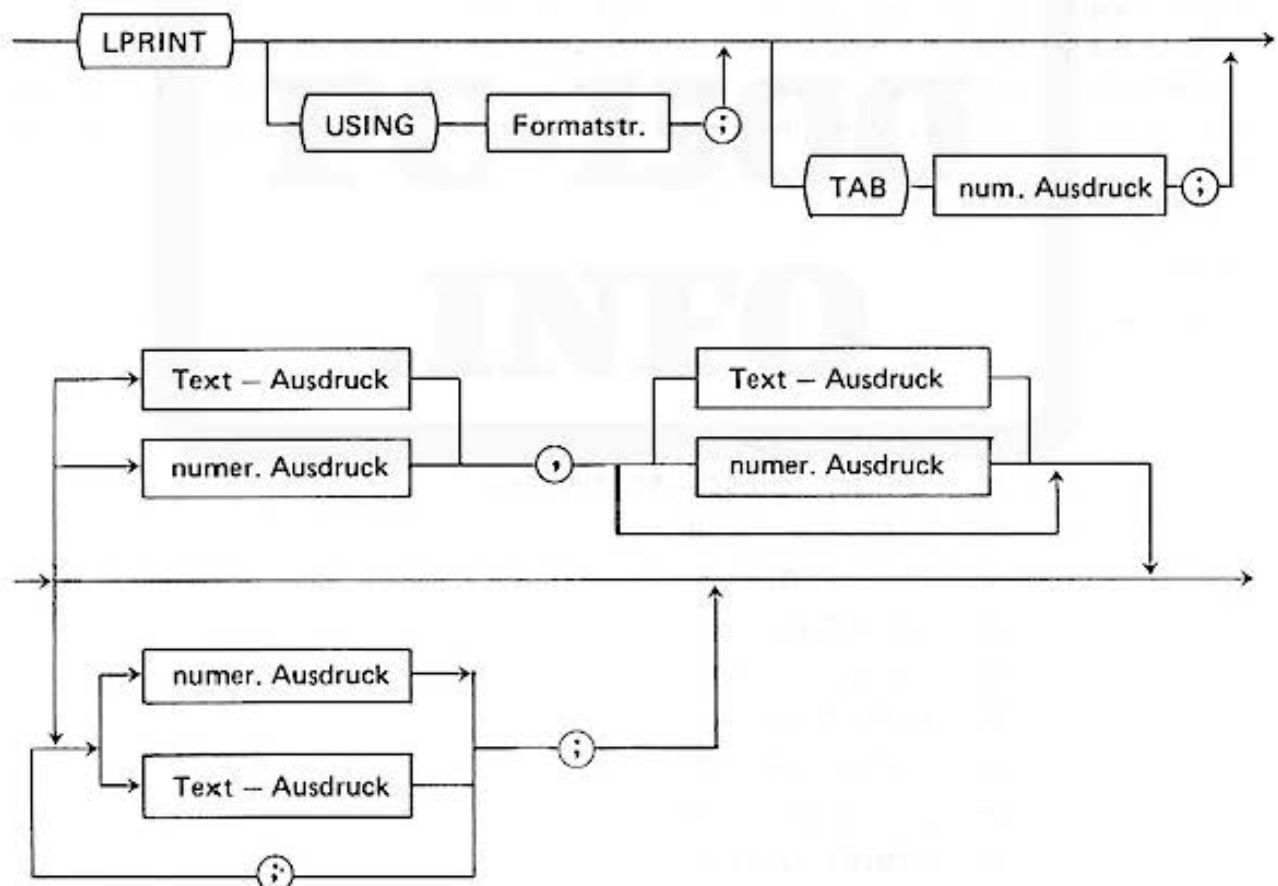
      B           A
    B B         A A
  B B B       A A
B B B B     A A
  A A B     B B
    A A   B B
      A B B
        B B
          B

```

LPRINT (Graphmode und Textmode)

Mit dieser Instruktion werden alphanumerische Texte und numerische Werte auf dem Drucker ausgegeben.

Syntax



Syntax und Wirkung der LPRINT – Instruktion ähneln der PRINT-Instruktion.

Insbesondere führt die Ausgabeformatsteuerung über die USING-Anweisung zu derselben Ausgabeform. Die Anzeige der Druckposition – der Cursor bei der PRINT-Instruktion – wird hier von dem Schreibkopf übernommen. Die aktuelle Position legt den Startpunkt für den Druck fest.

Der Programmlauf wird von der LPRINT-Anweisung jedoch nicht unterbrochen.

Im Textmode werden die Werte einzelner numerischer Ausdrücke rechts- und Texte linksbündig geschrieben. Längere Texte werden automatisch in den folgenden Zeilen fortgesetzt. Durch die Verwendung des Kommas in der Liste von Ausdrücken wird der Papierstreifen in zwei Felder geteilt analog zur PRINT-Instruktion.

Die Verwendung der Semikolons erlaubt es, beginnend mit der Kopfposition zu schreiben.

Die Zeichengröße und damit die Anzahl der Spalten pro Zeile kann von Ihnen über die Instruktion CSIZE gewählt werden. Mit jedem Zeichen rückt der Schreibkopf entsprechend der Zeichengröße ein Stück weiter nach rechts.

Der Teil TAB ist zusätzlich in die LPRINT-Instruktion aufgenommen. Er steuert die Positionierung des Schreibkopfes, so wie in der TAB-Anweisung beschrieben. Steht am Ende der LPRINT-Instruktion kein Semikolon, so beginnt der folgende Ausdruck mit einem Zeilenvorschub.

Die LPRINT-Instruktion GRAPH-Mode hat grundsätzlich die gleiche Wirkung. Es werden neue Zeilen jedoch nie automatisch angesteuert, sondern müssen vom Programmierer explizit angegeben werden.

Aus diesem Grunde ist auch eine Aufteilung in zwei Feldern (Kommasteuerung) nicht möglich.

Da die TAB-Instruktion ebenfalls nur im Textmode ausgeführt werden kann, ist der TAB-Teil der LPRINT-Instruktion im Graphmode nicht erlaubt. Beide erwähnten Fehler führen zu einem Fehler (ERROR 73). Sie haben jedoch zusätzliche Möglichkeiten:

So ist der Schreibkopf in feineren Einheiten steuerbar, d.h. Sie sind mit dem Startpunkt Ihrer Zeichenfolge nicht auf das Zeichenraster beschränkt. Außerdem können Sie Texte in y-Richtung drucken mit um 90° gedrehten Buchstaben zur Beschriftung von beispielsweise graphischen Zeichnungen.

(Siehe ROTATE, GLCOURSE etc.).

Beispiele:

(1) Textmode

```

20  LF 5
30  LPRINT 1.234
35  LPRINT "DIESES IST EIN TEXT"
40  LPRINT  $\pi$  , 10.45
50  ES = "ERGEBNIS : " : A = 3.35996E 32
60  LPRINT ES ; A
70  USING "###.##"
80  LPRINT ES ; A
90  USING "&&###"
100 FOR I = 65 TO 70
110 LPRINT CHR$(I) ; I
120 NEXT I

```


Das Aussehen des Ausdrucks ist abhängig von der gewählten Zeichengröße. Für die kleinste Zeichengröße ist die Spaltenanzahl ausreichend, um die Aufteilung in Programmzeile 40 innerhalb einer Druckzeile vorzunehmen.

```

1.234
DIESES IST EIN TEXT
3.141592654
ERGEBNIS : 3.3599E 32
ERGEBNIS : 3.3E 32
A 65B 66C 67D 68E 69F 70
    
```

Bei der Zeichengröße (CSIZE 2) reicht eine Druckzeile nicht mehr aus, deshalb werden automatisch die folgenden Druckzeilen zur Ergänzung benötigt.

```

1.234
DIESES IST EIN TEX
T
3.141592654
10.45
ERGEBNIS : 3.3599
6E 32
ERGEBNIS : 3.3E
32
A 65B 66C 67D
68E 69F 70
    
```

Programmzeile 40 benötigt für die beiden Felder 2 Zeilen.

Die Aufteilung in zwei Felder ist bei allen Zeichengrößen nicht mehr zulässig, wenn das Einzelergebnis nicht mehr in eine Zeile geschrieben werden kann. (ERROR 76)

Im Beispiel 1 mußte deshalb bei Zeichengröße 4 die Programmzeile 40 gelöscht werden.

```

1.234 - Zeile 30
DIESES IS
T EIN TEX - Zeile 35
T
ERGEBNIS
: 3.3599 - Zeile 50
6E 32
ERGEBNIS
: 3.3E - Zeile 110
32
A 65B 6
6C 67D
68E 69F
70
    
```

- (2) **Graphmode** Das leicht modifizierte Beispielpogramm liefert als Ergebnis eine Spalte; die Breite des Papierstreifens stellt keine Begrenzung mehr dar.

```

10:GRAPH
20:GLCURSOR (20,0
   ):ROTATE 1
30:LPRINT 1.234
35:LPRINT "DIESES
   IST EIN TEXT"
50:E$="ERGEBNIS :
   ":A=3.35996E3
   2
60:LPRINT E$;A
70:USING "###.##"
80:LPRINT E$;A
90:USING "&&###"
100:FOR I=65TO 70
110:LPRINT CHR$ (I
   );I;
120:NEXT I
    
```

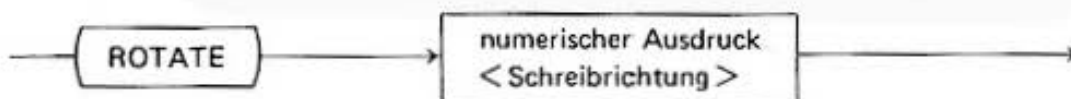
1.234DIESES IST EIN TEXTERGEBNIS : 3.35996E 32ERGEBNIS :

3.3E 32A 65B 66C 67D 68E 69F 70

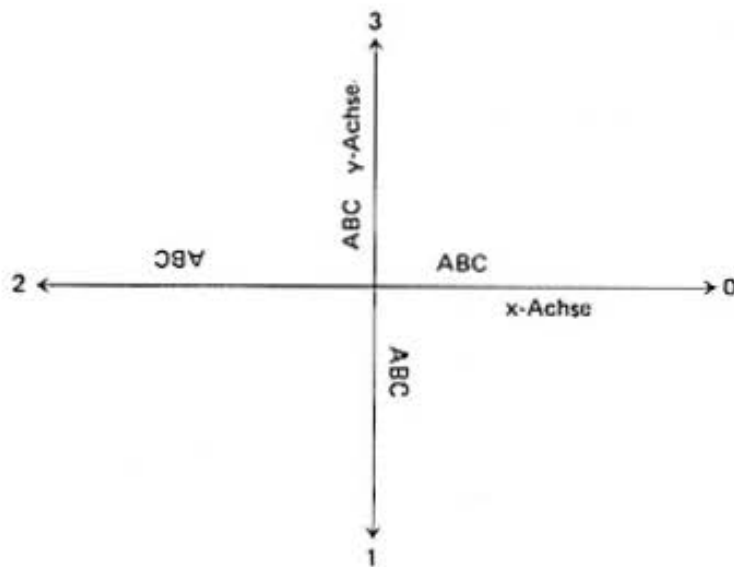
ROTATE (Graphmode)

Die ROTATE-Instruktion legt die Schreibrichtung fest.

Syntax



Der numerische Ausdruck darf die Werte 0, 1, 2, 3 annehmen und legt die Schreibrichtung längs den Achsen eines Koordinatenkreuzes fest. Der Wert 0 definiert die normale Schreibrichtung längs der positiven X-Achse, 1 entlang der negativen Y-Achse, 2 entlang der negativen X-Achse und 3 längs der positiven Y-Achse. Die Zeichen sind dabei so angeordnet, wie aus Bild 1 ersichtlich wird.



Wirksam wird die ROTATE-Instruktion mit dem nächsten Schreibbefehl LPRINT.

```

10:GRAPH
20:ROTATE 0
25:LPRINT " "
30:LPRINT "123456
    78"
40:ROTATE 1
50:LPRINT "123456
    78"
60:ROTATE 2
70:LPRINT "123456
    78"
80:ROTATE 3
90:LPRINT "123456
    78"
    
```

```

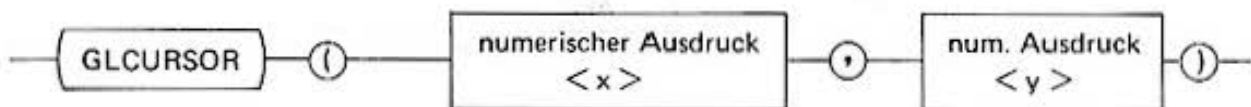
      12345678
87654321
12345678
12345678
    
```

Beispiel zu Rotate

GLCURSOR (Graphmode)

Diese Instruktion positioniert den Schreibkopf im Koordinatensystem.

Syntax



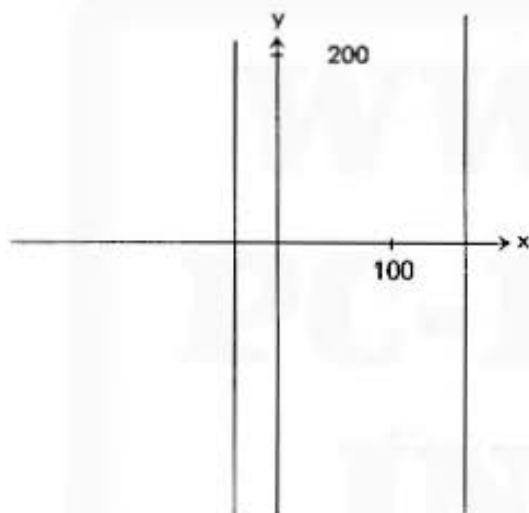
Die Werte der numerischen Ausdrücke x und y bestimmen die x, y -Position im Koordinatensystem. Zu dieser Position wird der Schreibkopf bewegt.

Die Werte müssen in dem Intervall

$-2048 \leq \langle x \rangle \leq +2047$ liegen.

Beispiele:

(1)



GLCURSOR (100, 200)

Der Schreibkopf wird zur Koordinate (100, 200) gefahren.

(2)

```
5: GRAPH
10: FOR I=1 TO 200
  STEP 2
20: GLCURSOR (1,
  SIN (I)*100)
25: LPRINT "/"
30: NEXT I
```

Festlegung der neuen Schreibposition:

— Bezogen auf das Koordinatensystem werden in Zeile 20 die Positionen entsprechend der SINUS-Funktion gewählt.



SORGN (Graphmode)

Mit dieser Instruktion wird der Ursprung des Koordinatensystems für die graphischen Anweisungen gesetzt.

Syntax



Diese Instruktion definiert die aktuelle Position des Schreibkopfes als Ursprung des Koordinatensystems für die folgenden Instruktionen (LINE, GLCURSOR etc.).

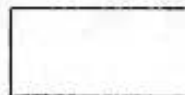
Beispiele:

```
10: GRAPH
20: LINE (0, 0)-(100, 0), 9
30: SORGN
40: LINE (0, 0)-(100, 50), 0, 0, B
50: TEXT
```

Zeile 20 bewegt den Schreibkopf 100 Einheiten nach rechts.

Zeile 30 definiert die Position als Ursprung für die folgenden Anweisungen.

Zeile 40 zeichnet ein Rechteck mit den Längen 100 x 50 Einheiten.



(3)

```
10: GRAPH
20: FOR I=0 TO 3
30: LINE (-30, 0)-(200, 0), 0, 1
40: LINE (0, 0)-(0, 100), 0, 1
50: LINE (0, 0)-(0, -50), 0, 1
60: GLCURSOR (40, 30)
70: SORGN
80: NEXT I
90: TEXT
```

Festlegung des Ursprunges von Koordinatensystemen

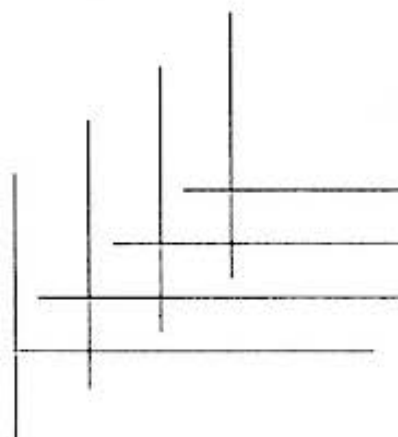
x-Achse zeichnen

y-Achse zeichnen

jeweils in einer anderen Farbe

neuer Ursprung des Koordinatensystems

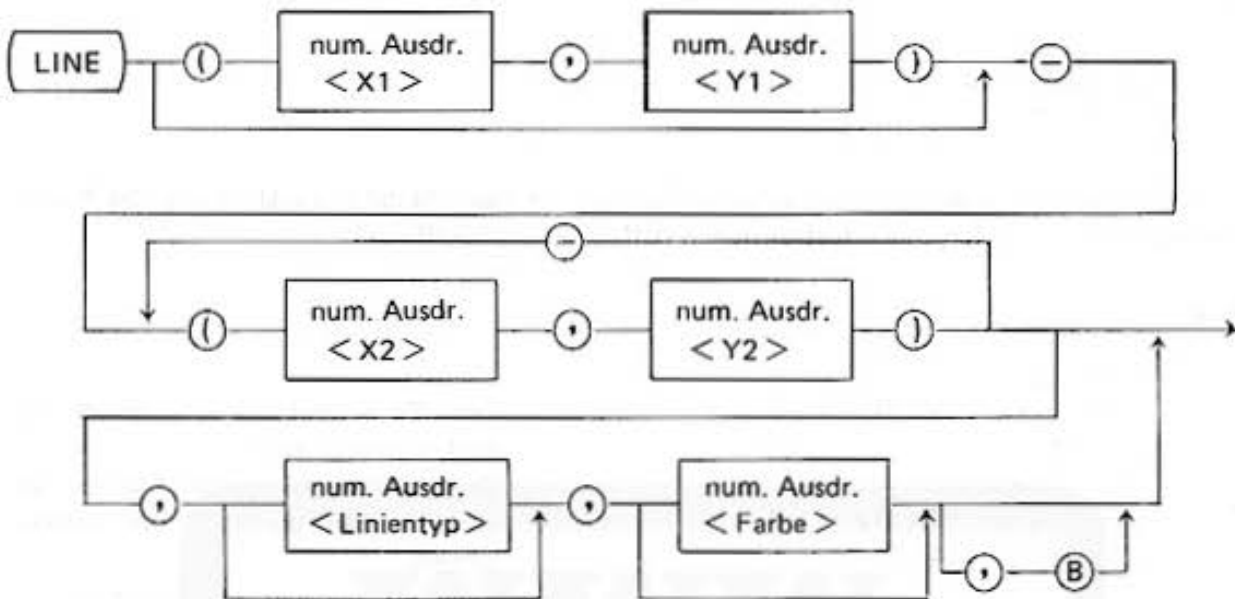
Nacheinander werden in der Schleife Zeile 40 – Zeile 80 der Ursprung des Koordinatensystems neu festgelegt.



LINE (Graphmode)

Mit dieser Instruktion werden Linien gezeichnet.

Syntax



Die LINE-Instruktion ist die Hauptanweisung zur Erzeugung von graphischen Zeichnungen, die sich aus einzelnen Strecken zusammensetzen. Es ist möglich, durchgezogene und gestrichelte Linien zu zeichnen.

Der Anfangspunkt der Strecke wird durch die Werte der numerischen Ausdrücke $\langle X1 \rangle$ und $\langle Y1 \rangle$, der Zielpunkt durch $\langle X2 \rangle$ und $\langle Y2 \rangle$ bestimmt. Die Werte müssen in dem Intervall zwischen

– 2048 und 2047

liegen. Alle Angaben beziehen sich auf den Ursprung eines gedachten Koordinatenkreuzes. (Siehe SORGN –Anweisung).

Sind die Ausgangskordinaten (X1, Y1) nicht angegeben, so wird die augenblickliche Kopfposition als Ausgangspunkt genommen. Wird eine Folge von weiteren Koordinatenpunkten angegeben, so wird ein Linienzug durch all diese Punkte gezeichnet. (Siehe Beispiel Dreieckszeichnung).

Der numerische Ausdruck $\langle \text{Linientyp} \rangle$ legt fest, ob eine durchgezogene oder gestrichelte Linie gezeichnet wird.

Der Wert muß im Intervall

$0 \leq \langle \text{Linientyp} \rangle \leq 9$ liegen

<u>Linientyp</u>	<u>Bedeutung</u>
0	durchgezogene Linie
1	gestrichelt mit je 0,4 mm Strichlänge
2	" 0,6 mm "
3	" 0,8 mm "
4	" 1,0 mm "
5	" 1,2 mm "
6	" 1,4 mm "
7	" 1,6 mm "
8	" 1,8 mm "
9	keine Linie, nur Positionierung

Der numerische Ausdruck < Farbe > wählt den entsprechenden Farbstift aus. Dieser darf die Werte 0, 1, 2, 3 annehmen. Die Zuordnung ist die gleiche wie bei der COLOR-Instruktion. Die Angabe von < Linientyp > und < Farbe > ist optional. Läßt man sie fort, so ist die letzte Festlegung gültig.

Die Angabe von B am Ende der Instruktion ergibt die Zeichnung eines Rechteckes im Koordinatensystem mit den Eckpunkten (X_1, Y_1) , (X_2, Y_1) , (X_2, Y_2) , (X_1, Y_2) .

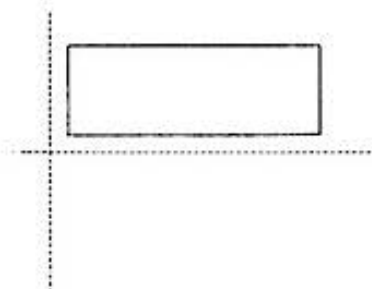
Beispiele:

(1)

```

10: GRAPH
20: LINE (0, 0)-(20
    0, 0), 1
30: LINE (0, 0)-(20
    , 0), 9
40: SORGN
50: LINE (0, 0)-(0,
    80), 1, 0
60: LINE (0, 0)-(0,
    -80), 1, 0
70: LINE (10, 10)-(
    150, 60), 0, 0, B
80: TEXT

```



Demonstration der LINE-Instruktion
Zeichnung eines Rechteckes im
Koordinatensystem

Bedeutung der Instruktion

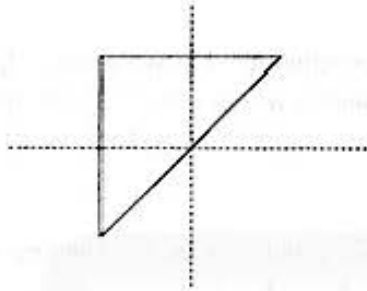
- gestrichelte X-Achse
- Nullpunkt des Koordinatensystems
- gestrichelte positive Y-Achse
- gestrichelte negative Y-Achse
- zeichnet das Rechteck

Rechteck mit den Endpunkten
 $(10, 10)$, $(150, 10)$, $(150, 60)$ und
 $(10, 60)$

Nach der Umschaltung in den Graph-mode ist der Ursprung des Koordinatensystems die Position des Schreibkopfes am linken Papierrand. Zeile 40 verlegt ihn an die in der Zeichnung ersichtlichen Stelle.

(2)

```
10:GRAPH
15:LINE (0,0)-(20
    0,0),1
20:LINE (0,0)-(10
    0,0),9:SORGN
25:LINE (0,0)-(0,
    80),1:LINE (0,
    0)-(0,-80),1
30:LINE (0,0)-(50
    ,50)-(-50,50)-
    (-50,-50)-(0,0
    ),0,0
40:TEXT
```

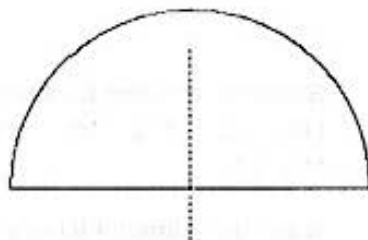


Zeichnung eines Dreiecks in dem Koordinatensystem

- zeichnet die gestrichelte X-Achse
- positioniert den Schreibkopf und legt dieses Position als Ursprung fest
- zeichnet die Y-Achse gestrichelt
- zeichnet das Dreieck als Linien zwischen den Punkten

(3)

```
5:J=1
10:GRAPH
20:R=100
40:GLCURSOR (110,
    0):SORGN
50:FOR I=0TO 2*R
    STEP J
55:X=R-I
60:Y=SQR (R*R-X*X
    )
70:LINE -(X,Y),0,
    0
80:NEXT I
90:LINE -(0,0)
100:LINE (0,0)-(0,
    80),1
110:LINE (0,0)-(0,
    -30),1
```

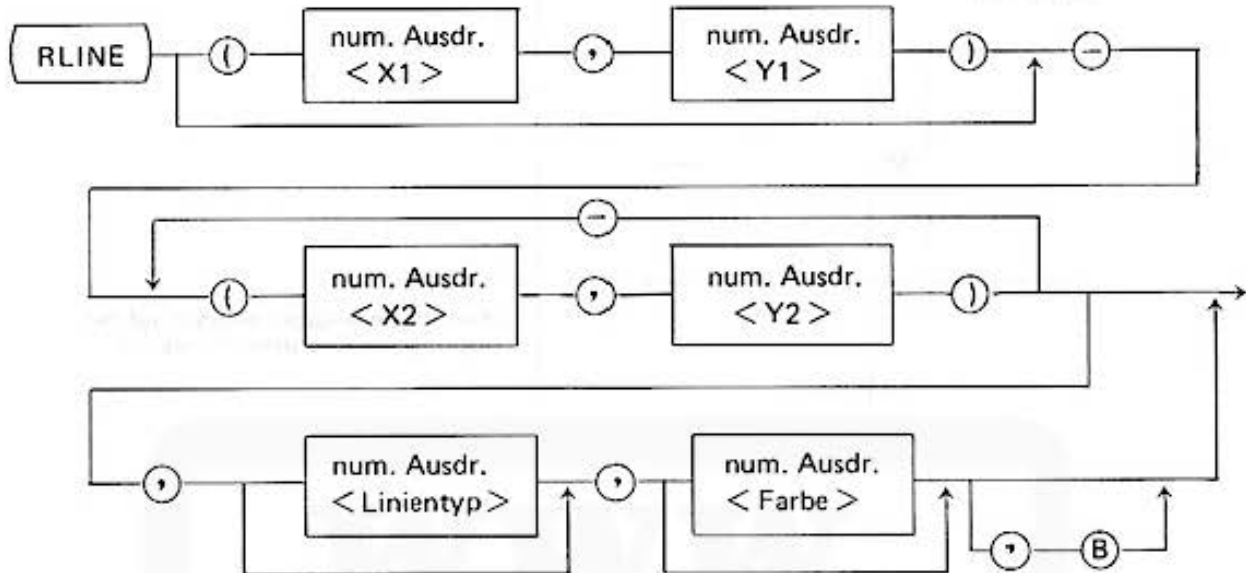


Zeichnung eines Halbkreises

- Radius
- Ursprung eines Koordinatensystems
- Linien zeichnen von der Position des Schreibkopfes bis zum Punkte (X, Y)
- gestrichelte positive Y-Achse
- gestrichelte negative Y-Achse

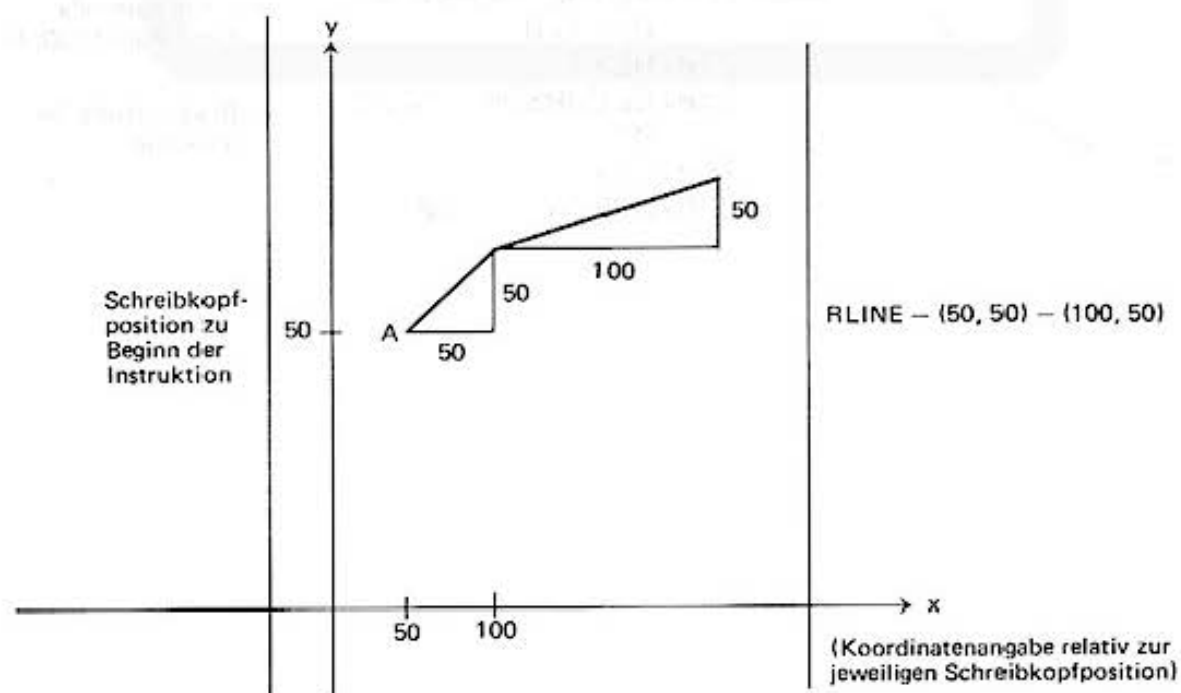
RLINE

Mit dieser Instruktion werden Linien gezeichnet, bezogen auf ein Koordinatensystem, dessen Ursprung die augenblickliche Schreibposition ist.

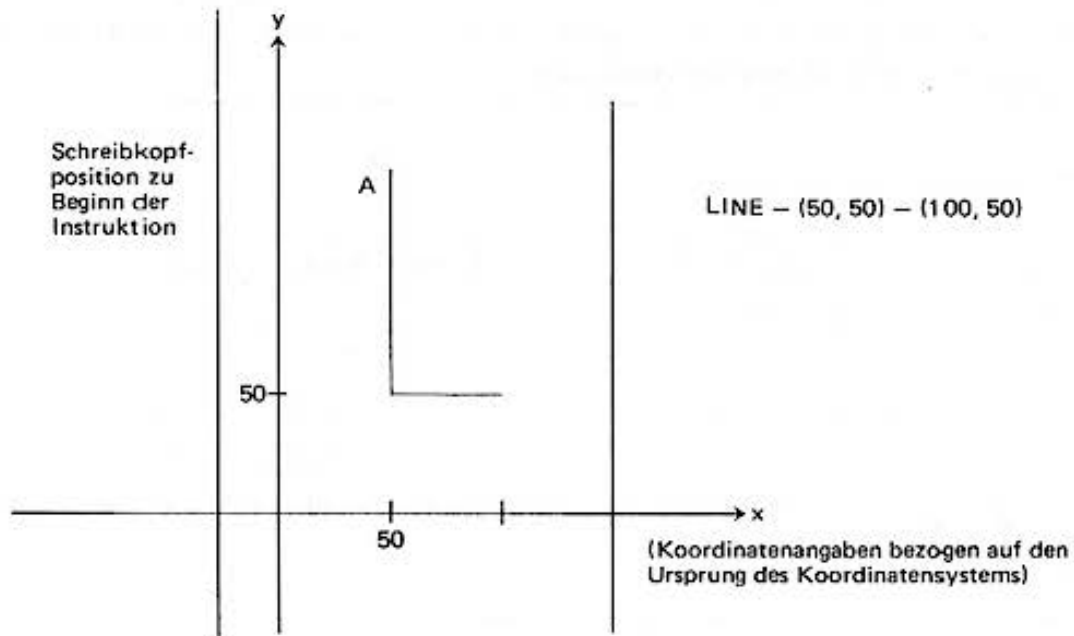
Syntax

Die RLINE-Instruktion hat die gleiche Form und Wirkung wie die LINE-Anweisung. Im Unterschied zu dieser werden die Koordinatenangaben jedoch immer relativ auf die aktuelle Schreibkopfposition bezogen. Dadurch bleibt das durch SORG festgelegte absolute Koordinatensystem unberührt. Wird eine relative Anfangskoordinate (X_1 , Y_1) angegeben, so wird der Kopf auf diese Position bewegt, ohne zu zeichnen.

Beispielsweise führt die RLINE – (50, 50) – (100, 50) Instruktion zu folgendem Linienzug:

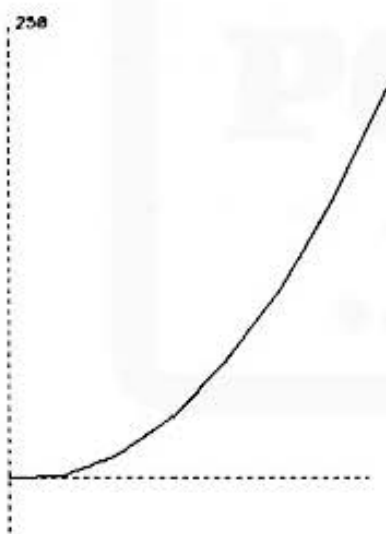


Während die entsprechende LINE-Anweisung zu dem Ergebnis führt:



Beispiel:

(1) 250



200:GRAPH

202:LINE (0,0)-(20,0),2

204:LINE (0,250)-(0,-30),2

206:GLCURSOR (0,0)

210:FOR I=1 TO 60
STEP 10

220:RLINE (0,0)-(30,1),0

230:NEXT I

250:GLCURSOR (5,250)

260:CSIZE 1

270:LPRINT "250"

- Zeichnung des absoluten Koordinatensystems

- Linie zum relativen Punkt (30, 1)

- Beschriftung der Y-Achse

(2)

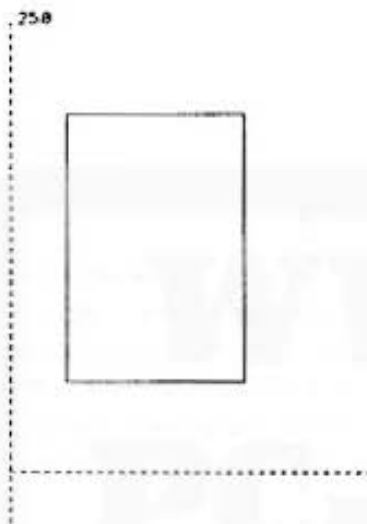
```
200:GRAPH
202:LINE (0,0)-(20
      0,0),2
204:LINE (0,250)-(
      0,-30),2
206:GLCURSOR (30,5
      0)
210:I=100:J=150
220:RLINE -(I,J),0
      ,,B
250:GLCURSOR (5,25
      0)
260:CSIZE 1
270:LPRINT "250"
```

Zeichnung eines Rechteckes

— x, y-Koordinatensystem in gestrichelten Linien

— Zeichnung des Rechteckes

beginnend mit der Schreibkopfposition und den Kantenlängen 100 und 150



19. Datenspeicherung auf Magnetband

19.1 Einleitung

In Verbindung mit dem Plotter/Drucker CE-150 können Kassettenrekorder als externe Programm-Daten-Speicher an den Computer PC-1500 angeschlossen werden.

Der Kassettenbetrieb ermöglicht folgende Betriebsarten:

- a) manuelles Abspeichern von Programmen
- b) manuelles und programmkontrolliertes Rückladen von Programmen
- c) manuelles und programmkontrolliertes Abspeichern von Daten
- d) manuelles und programmkontrolliertes Rückladen von Daten
- e) manuelles Rückladen von zwei oder mehr Programmen
- f) Vergleichen von Programmen und Dateien
- g) Abspeichern von Reserveausdrucken
- h) manuelles Rückladen von Reserveausdrucken

19.2 Auswahlkriterien für einen Kassettenrekorder

Für eine problemlose Aufzeichnung und Wiedergabe von Daten ist die Anschaffung eines neuen Rekorders zu empfehlen. Um den Verschleiß an den Tonköpfen und der übrigen Mechanik so klein wie möglich zu halten, soll der Rekorder ausschließlich für die Aufzeichnung und Wiedergabe von Daten verwendet werden. Sehr gut geeignet für den Bandbetrieb sind die kompakten monofonen Minirekorder, die mit Standardkassetten arbeiten. (Stereo-Geräte sind weniger geeignet).

Der Minirekorder soll folgende Ausstattung haben:

- 1) Fernbedienung (Remote)
- 2) Bandzählwerk
- 3) Eingang (MIC) 3,5 mm Ø Klinkenbuchse
- 4) Ausgang (EAR) 3,5 mm Ø Klinkenbuchse
- 5) Fernbedienung 2,5 mm Ø Klinkenbuchse

Daneben sind folgende technischen Daten gefordert:

- | | |
|----------------------------------|---------------------------------------|
| 1) Eingangsimpedanz (MIC) | 200 ~ 1000 OHM |
| 2) Eingangsempfindlichkeit (MIC) | Kleiner als 3 mV (-50 dB) |
| 3) Ausgangsimpedanz (EAR) | Kleiner als 10 OHM |
| 4) Ausgangsleistung (EAR) | Größer als 1 Vss |
| 5) Klirrfaktor | Kleiner als 15 %
zwischen 2 ~ 4KHZ |
| 6) Gleichlaufschwankung | 0,3 % max. (W.R.M.S.) |

19.3 Betriebshinweise

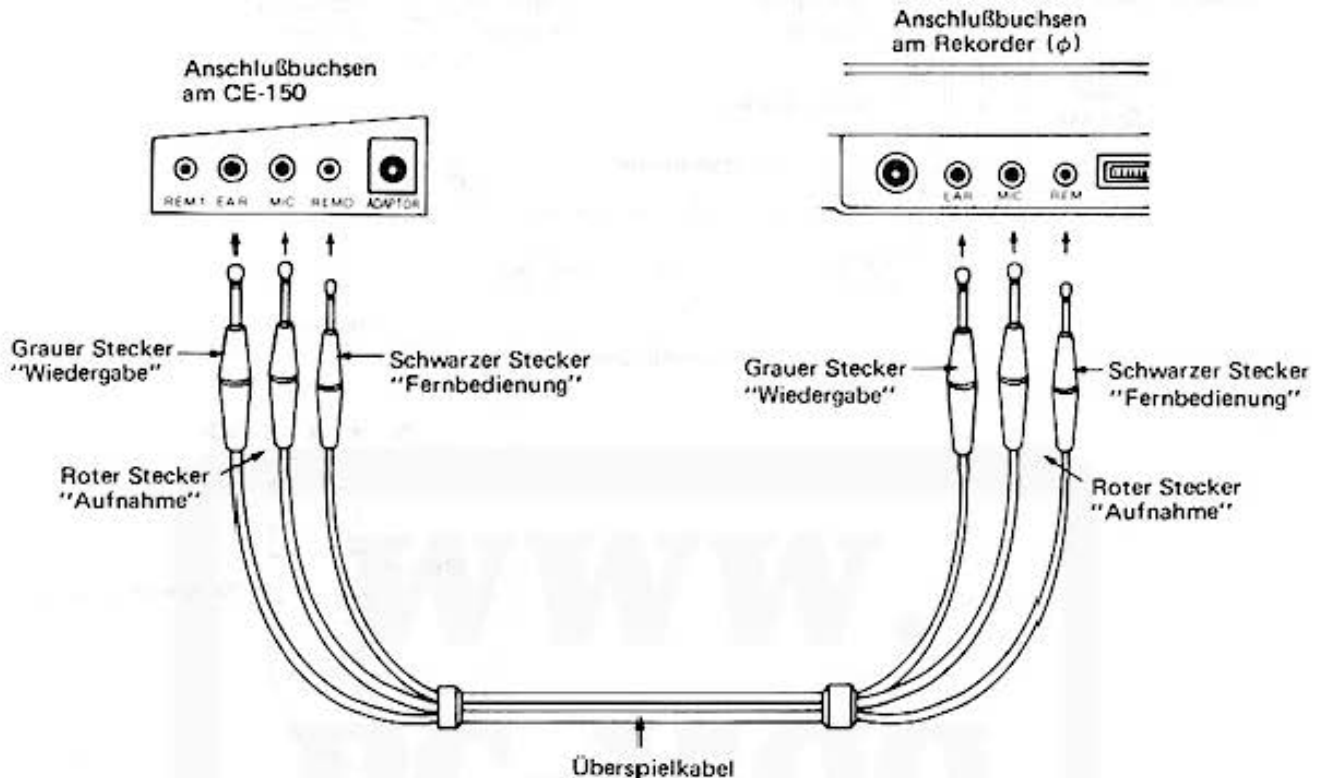
- 1) Verwenden Sie den Rekorder nur zur Programm-/Datenaufzeichnung
- 2) Verwenden Sie nur Qualitätskassetten (z.B. TDK SA C60/90, Maxell UDXL-II C60/90 etc.)
- 3) Benutzen Sie den Rekorder nach Möglichkeit mit Batterien (bei Netzbetrieb können Knackstörungen zu Datenverlusten führen).
- 4) Beachten Sie die Betriebsanleitung Ihres Rekorders.

19.4 Anschluß der Rekorder an den CE-150

a) Anschluß eines Kassettenrekorders (0)

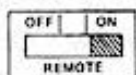
Sie schalten den Computer PC-1500 aus.

Bei der Verwendung eines Rekorders schließen Sie die Geräte gemäß untenstehender Abbildung an.




Hinweis:

Die Anschlüsse REM \emptyset am CE-150 und REM am Kassettenrekorder bezeichnen die Fernbedienung der Geräte. Sobald die Verbindung hergestellt ist, und der Schiebeschalter des CE-150



auf "ON" steht, sind alle Rekorderlauffunktionen wie RECORD/PLAY, REW (Rücklauf) und FF (schneller Vorlauf) gesperrt.

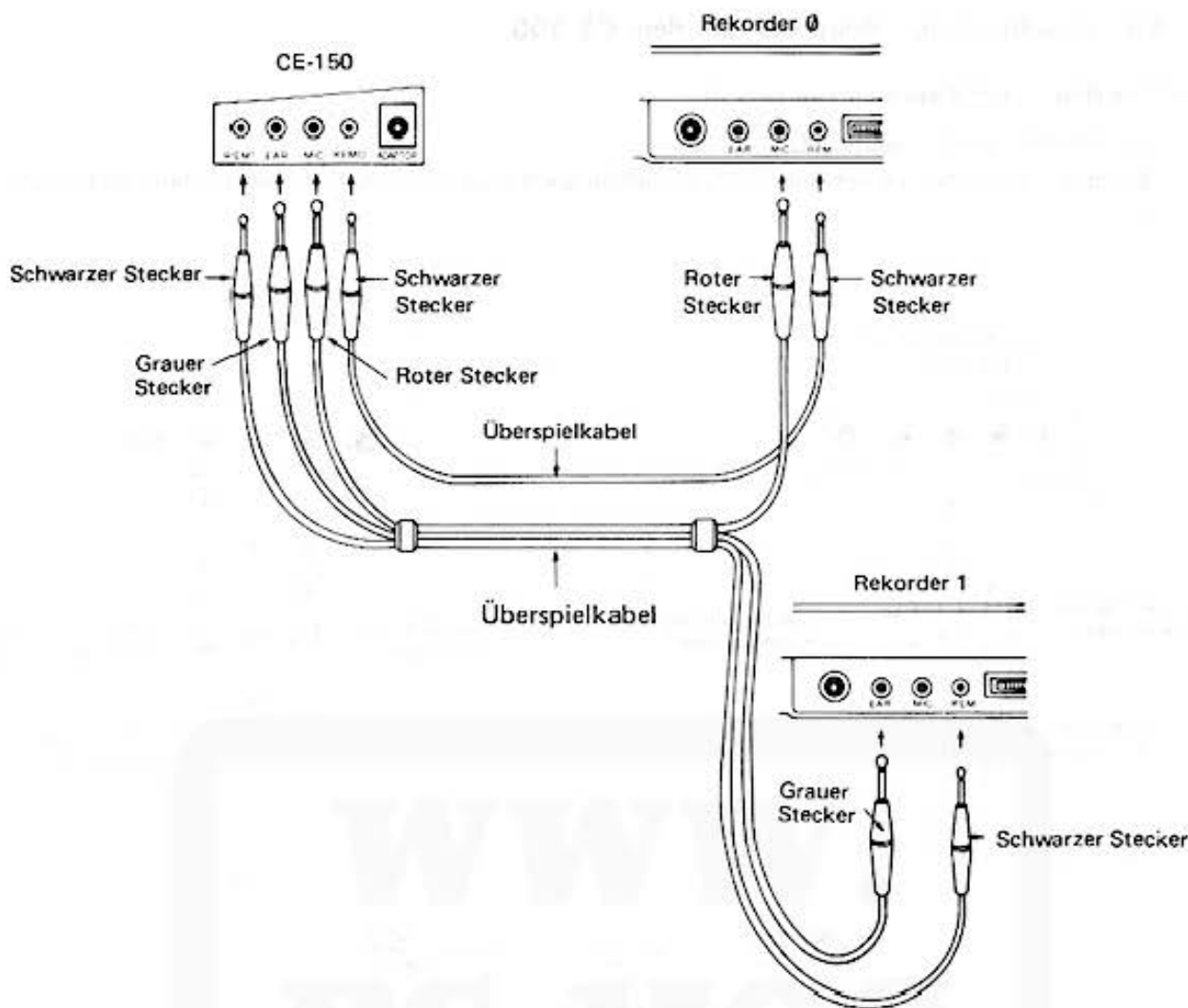
Wird der Schalter  auf "OFF" gestellt, sind obige Lauffunktionen wieder manuell möglich.

Beachten Sie, daß die BASIC-Instruktionen CHAIN, CLOAD, CLOAD?, CSAVE, INPUT#, MERGE, PRINT# das Bandgerät \emptyset , angeschlossen über die Anschlußbuchse REM \emptyset , bedienen.

b) Anschluß von zwei Kassettenrekordern (0 und 1)

Sie schalten den Computer PC-1500 aus.

Bei der Verwendung zweier Rekorder (0 und 1) schließen Sie die Geräte gemäß untenstehender Abbildung an.



Hinweis:

Bei der Verwendung von zwei Rekordern kann der erste (Rekorder 0) für das Abspeichern von Informationen und der zweite (Rekorder 1) für das Rückladen von Informationen benutzt werden. Anders als bei REM 0 wird REM 1 vom Computer unter Verwendung des Befehls RMT ON **ENTER** bzw. RMT OFF **ENTER** kontrolliert. Der Anschluß REM 1 wird über die Instruktion RMT ON bzw. RMT OFF gesteuert.

Beachten Sie, daß die BASIC-Instruktionen CHAIN-1, CLOAD-1, CSAVE-1, INPUT #-1, MERGE-1, PRINT-1 das Bandgerät 1 (REM 1) bedienen.

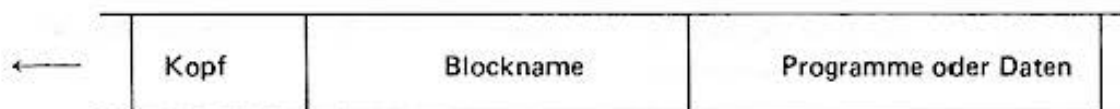
19.5 Datenspeicherung auf dem Magnetband

Auf dem Magnetband können Sie Daten, Programme und Reserve-Ausdrücke abspeichern oder von dort zurückladen. Außerdem können Sie das Magnetband als externen Programmspeicher verwenden, um große Programme ablaufen zu lassen.

Die Informationen aus Daten und Programmen werden in Blöcken nach dem Zweittonverfahren auf das Magnetband geschrieben.

Auf einem Magnetband können abhängig vom Kassettentyp sehr viele Informationen gespeichert werden. Damit Sie die Daten oder Programme wiederfinden und unterscheiden können, beginnt jeder Block mit einem Blocknamen, der aus maximal 16 Zeichen besteht. Abgeschlossen wird der Block, wenn die gesamte Information abgespeichert ist.

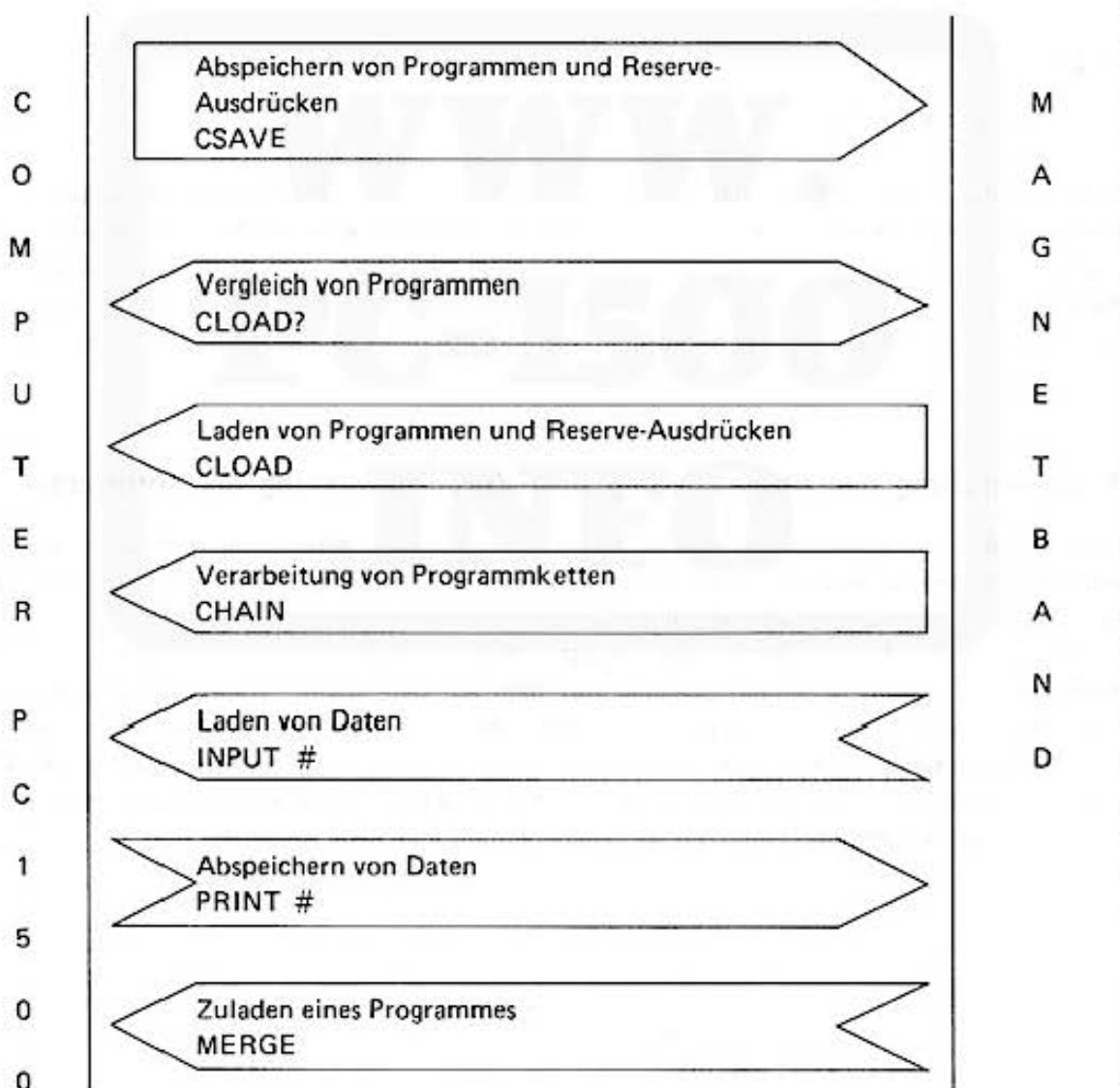
Um eine sichere Trennung der Blöcke zu gewährleisten, schreibt der Rechner vor dem Blocknamen automatisch etwa 5 bis 7 Sekunden einen konstanten Signalton. Ein kompletter Block hat damit folgende Form:



Hinweis:

Auf die Informationen auf dem Band können Sie nur sequentiell zugreifen. Damit Sie Ihre Daten und Programme schneller wieder auffinden, sollten Sie sich vor dem Abspeichern von Informationen den Zählerstand des Kassettenrekorders und den Blocknamen der Daten aufschreiben. Sie werden sehr schnell eine umfangreiche Bibliothek von Daten erhalten.

Folgende Möglichkeiten werden beim Bandbetrieb geboten:



19.6 Benutzung des Magnetbandes

Schließen Sie Ihr Magnetbandgerät an den PC-1500 an, so wie in Kap. 19.4 beschrieben.

Bitte benutzen Sie für Ihre ersten Versuche den REM 0 Anschluß. Nachdem Sie Ihr Programm so wie normalerweise auch eingegeben haben, können Sie den Rekorder für die Abspeicherung vorbereiten. Sie müssen dafür folgendes tun:

- 1) Stellen Sie den REMOTE Schalter des Interfaces CE-150 auf OFF.
- 2) Legen Sie ein Magnetband in den Rekorder ein, und vergewissern Sie sich, daß es zurückgespult ist.
- 3) Stellen Sie den Zähler auf Null.
- 4) Suchen Sie eine geeignete Stelle auf dem Band.
Merken Sie sich den Zählerstand (ist das Band leer, zu Beginn des Bandes).
- 5) Stellen Sie den REMOTE Schalter des Interfaces auf ON.
- 6) Bereiten Sie die Aufnahme vor:
Drücken Sie die "RECORD"- und "PLAY"-Taste.
Lautstärkeregler auf Mitte bis Maximum stellen, Tonhöhenregler auf "hoch".
- 7) Kommando Eingabe:
Mit der Abspeicher-Anweisung CSAVE müssen Sie Ihr Programm gleichzeitig mit einem Namen, dem Blocknamen, versehen.

Geben Sie jetzt ein:

CSAVE "BEISPIEL"

Wenn Sie die **ENTER** Taste betätigt haben, setzt sich das Magnetband in Bewegung. Ihr Programm wird jetzt übertragen und unter dem Namen "Beispiel" abgespeichert. Zu Beginn hören Sie, sofern der Lautsprecher eingeschaltet ist, 5–7 Sekunden den konstanten Signalton, anschließend eine Folge von Tönen. Sobald der Computer am Ende des Programmes angelangt ist, hört das Band auf zu laufen; das Bereitschaftssymbol erscheint wieder auf der Anzeige.

19.7 Überprüfung der Abspeicherung und Zurückgewinnung der Information

Nach der Abspeicherung Ihres Programmes mit der CSAVE-Anweisung und bevor Sie das Programm in Computer löschen, ist es empfehlenswert, zu überprüfen, ob es fehlerfrei übertragen wurde. Das Band kann beispielsweise eine schadhafte Stelle haben.

Die Überprüfung ist ganz einfach und erfolgt mit der CLOAD? Anweisung. Der Computer vergleicht das Programm in seinem Speicher mit dem auf dem Band. Wenn alles geklappt hat, wird der Blockname angezeigt und anschließend das Bereitschaftszeichen ausgegeben. Wenn der Rechner einen Unterschied feststellt, wird er einen Fehler anzeigen (normalerweise ERROR 43). Löschen Sie dann diesen Teil des Bandes, überprüfen Sie alle Verbindungen zwischen dem Interface und dem Magnetband und benutzen Sie gegebenenfalls eine neue Kassette.

19.8 Vorbereitung für die CLOAD?-Anweisung

- 8) Lautstärkeregler auf Mitte bis Maximum stellen, Tonhöhenregler auf "hoch".
- 9) Schalten Sie das Magnetband auf "STOP".
- 10) Stellen Sie den REMOTE-Schalter des Interface auf OFF.
- 11) Lassen Sie das Band bis zum Programmanfang entsprechend dem Zählerstand zurücklaufen.
- 12) Halten Sie das Band an dieser Stelle an.
- 13) Stellen Sie den REMOTE-Schalter auf ON.
- 14) Drücken Sie den PLAY-Schalter.
- 15) Geben Sie ein:

CLOAD?

Wenn Sie die **ENTER** Taste gedrückt haben, setzt sich das Band in Bewegung, Sie hören die gleiche Signalfolge wie bei der Abspeicherung, und auf der Anzeige erscheint der Blockname

BEISPIEL

Die Programme werden verglichen und anschließend erscheint das Bereitschaftszeichen > auf der Anzeige. Ist ein Fehler aufgetreten, versuchen Sie die Abspeicherung noch einmal.

Jetzt können Sie sicher sein, zu einem späteren Zeitpunkt das Programm wieder vom Magnetband zurückgewinnen zu können.

Die Handgriffe hierzu sind die gleichen: anstatt CLOAD? geben Sie jetzt unter Punkt 15)

CLOAD

oder

CLOAD "BEISPIEL"

ein.

Es wird das Programm vom Magnetband in den Speicher des Rechners übertragen. Die beiden CLOAD-Anweisungsformen unterscheiden sich nur dadurch, daß bei der zweiten Instruktion der Blockname zusätzlich mit der aufgeführten Zeichenfolge verglichen wird. Nur wenn diese identisch sind, findet die Übertragung statt.

Stimmen der eingegebene Programmname und der gespeicherte Blockname nicht überein, so werden alle auf dem Band folgenden Blöcke nach diesem Programmnamen abgesucht. Wird er gefunden, findet die Übertragung des Programmes in den Computer statt.

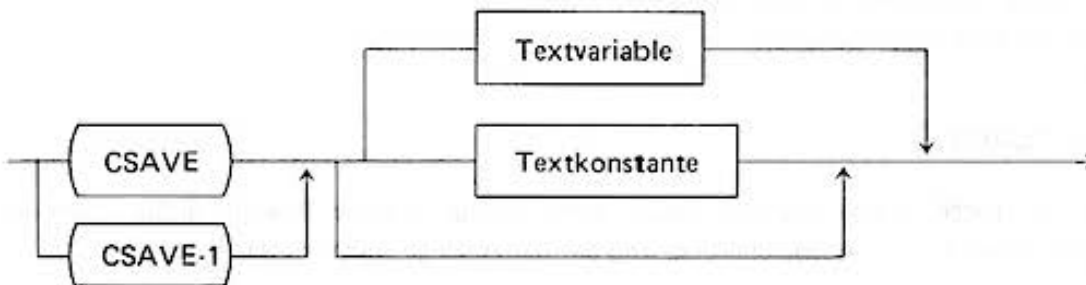
Die Benutzung des 2. Magnetbandes (REM 1 Anschlußbuchse) unterscheidet sich nur dadurch, daß anstatt der Betätigung des REMOTE-Schalters die Instruktion RMT ON bzw. RMT OFF eingegeben und an das Kommando (CSAVE, CLOAD) ein -1 angefügt werden muß (CSAVE-1, CLOAD-1). Diese einfachen Handgriffe sollten Sie sich einprägen. Sie müssen beim Abspeichern und Zurückladen von Daten und Programmen immer wieder ausgeführt werden.

Den Lautsprecher können Sie vor der Übertragung mit der BEEP OFF Instruktion ausschalten.

CSAVE

Mit dieser Instruktion werden Programme und RESERVE-Ausdrücke auf das Magnetband geschrieben.

Syntax



Mit der Anweisung CSAVE werden Programme und RESERVE-Ausdrücke auf das Magnetband – 0, mit CSAVE-1 auf das Band-1 geschrieben. Die Textkonstante oder Textvariable legt den Blocknamen auf dem Band fest. Dabei werden die ersten 16 Zeichen berücksichtigt. Unter diesem Namen können Sie später das Programm auf der Kassette wieder auffinden.

Die Überspielung dauert, abhängig von der Blocklänge, einige Minuten.

Es werden in PRO- und RUN-Mode das gesamte Programm, im RESERVE-Mode alle Reserve-Ausdrücke abgespeichert. Der Informationstyp (Programm, Reserve-Ausdruck) wird auf dem Band vermerkt. Im Anschluß an das Speichern des Blockes sollte man mit der Anweisung CLOAD? überprüfen, ob nicht etwa bedingt durch Bandfehler Information verloren gegangen ist.

Beispiele:

- (1) direkte Kommandos im PRO-Mode

CSAVE "STEUERBERECHNUNG"

CSAVE-1 "EINKOMMENST"

CSAVE AS

CSAVE "RESEV-1" im RESERVE-Mode

- (2) im Programm

900 B\$ = "PROG-BEISPIEL"

1000 PRINT "ACHTUNG : AUFNAHME BEREIT ?"

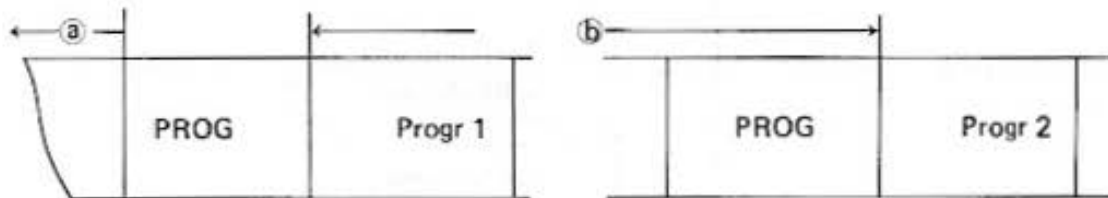
1010 CSAVE B\$

Wenn man die Angabe in Zeile 1000 mit **ENTER** bestätigt, wird das Programm abgespeichert unter dem Namen:

PROG-BEISPIEL

Hinweise:

- Wird ein neues Programm so auf dem Band gespeichert, daß es einen anderen Block auch nur teilweise überlappt, wird die ursprüngliche Information teilweise gelöscht. Dies führt zu einem Fehler beim Laden dieses Blockes.
- Haben zwei Blöcke die gleichen Namen, etwa "PROG", so lädt der Rechner das zuerst aufgefundene Programm. Befindet sich der Tonkopf in der Darstellung im Bereich a, so wird Progr. 1 geladen. Befindet er sich jedoch im Bereich b, so wird Progr. 2 geladen.



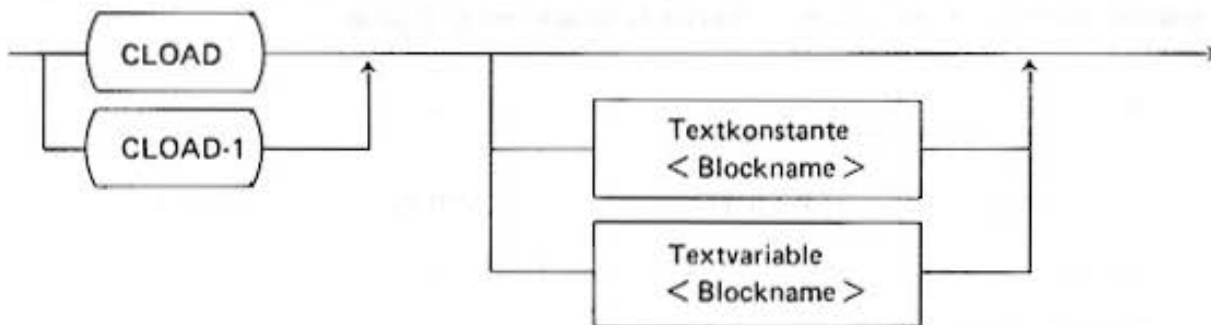
← Bandlaufrichtung

- Zweckmäßigerweise legt man sich für jedes Band der Programmbibliothek ein Karteiblatt an mit Namen, Hinweis auf Informationstyp (Progr., Reserv, Data), Zählerstand und kurze Programmbeschreibung. Hat man den Namen und Zählerstand vergessen, so kann es sehr mühselig und aufwendig sein, sich diese Information wieder zu besorgen. Kennen Sie den Zählerstand, so können Sie mit dem schnellen Vorlauf den Tonkopf schon richtig positionieren und gewinnen Zeit.

CLOAD (direktes Kommando)

Mit diesem Kommando werden Programme und Reserve-Ausdrücke vom Band geladen.

Syntax



Mit dem Lade-Kommando CLOAD werden Programme und Reserve-Ausdrücke vom Rechner gesucht und in den Programmspeicher geladen. CLOAD benutzt den Anschluß REM 0 und CLOAD-1 REM 1. Das Laden ist nur manuell als direktes Kommando möglich. Programme oder Reserve-Ausdrücke, die sich im Speicher befinden, werden vor der Ladeoperation gelöscht. Die Textkonstante bzw. die Textvariable bezeichnen den Blocknamen. Nach diesem Blocknamen wird auf dem Band automatisch gesucht. Ist der Block gefunden, wird er in den Rechner geladen und der Rechner gibt das Bereitschaftssymbol > aus. Während dieses Suchvorganges werden Ihnen die Namen der Blöcke, die jeweils gelesen werden, auf der Anzeige angezeigt. Wird keine Textvariable und Textkonstante angegeben, so wird der folgende Block unabhängig vom Blocknamen geladen.

Abhängig von der gewählten Betriebsart wird der Block als Programm (RUN- bzw. PRO-Mode) oder als RESERVE-Ausdruck (Reserve-Mode) aufgefaßt.

Hinweise:

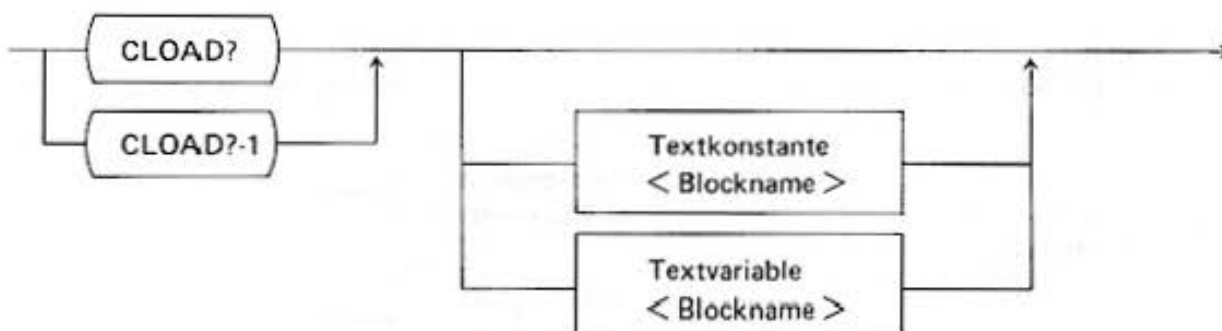
Befindet sich der gesuchte Blockname nicht auf dem eingelegten Band, sucht der Computer auch dann nach dem fehlenden Namen weiter, wenn das Band abgelaufen ist. In diesem Falle muß man mit **BREAK** diese Operation abbrechen.

Wurde die falsche Betriebsart gewählt, Laden von Reserve-Ausdrücken im PRO-Mode oder umgekehrt, so wird dieser Block überlesen.

CLOAD? (direktes Kommando)

Mit diesem Kommando werden Band- und Rechnerinformationen verglichen.

Syntax



Mit dem Kommando CLOAD? vergleicht man die auf dem Band gespeicherte Information des spezifizierten Blocknamens (Textvariable und Textkonstante) mit dem im Computer gespeicherten Programm (RUN- bzw. PRO-Mode) oder den RESERVE-Ausdrücken. Der Vergleich kann nur manuell abgerufen werden.

Der Ablauf der CLOAD? Instruktion ist identisch mit der CLOAD-Anweisung. Werden bei dem Vergleich Fehler festgestellt, so erscheint eine Fehlermeldung (ERROR 43); sonst das Bereitschaftssymbol >.

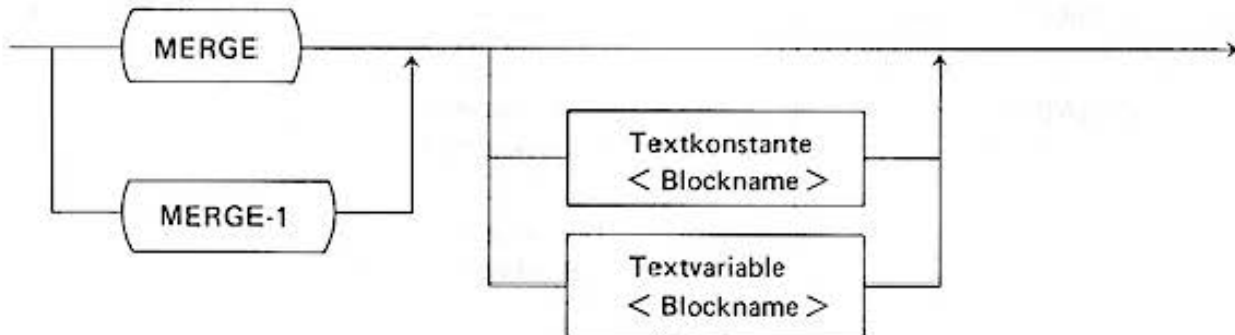
Beispiel:

```
CLOAD?  
CLOAD? "PROG 1"  
CLOAD?-1
```


MERGE (direktes Kommando)

Mit diesem Kommando werden Programme vom Band zu den im Computer gespeicherten hinzugeladen.

Syntax



Die MERGE-Instruktion hat die gleiche Funktion wie die CLOAD-Anweisung, allerdings wird das alte Programm nicht gelöscht, sondern das neue hinzugefügt. Auf diese Weise wird es Ihnen ermöglicht, 2 oder mehrere Programme im Computer zu speichern.

Die Anweisungsnummern der beiden Programme werden nicht verändert. Es kann also passieren, daß die gleiche Nummer zwei Mal vorhanden ist. Bei der Programmausführung erfolgen Sprünge über GOTO < Zeilennummer > nur im selben Programm.

Mit GOTO < Markenname > kann man von einem Programm ins andere springen. Wird ein Programmende erreicht, wird automatisch der Programmablauf beendet.

Beispiel:

Ein Auszug aus dem Programmspeicher

CLOAD "PROG-1"

Programm PROG-1

MERGE "PROG-2"

Programm PROG-1
Programm PROG-2

MERGE "PROG-3"

Programm PROG-1
Programm PROG-2
Programm PROG-3

Befinden sich mehrere Programme im Programmspeicher, so werden die Programme einzeln ausgeführt. Damit die Programme PROG-2, PROG-3 usw. ablaufen können, müssen sie in der ersten Programmzeile jeweils mit einem Namen versehen werden. Über RUN < Markenname > oder GOTO < Markenname > (siehe Beispiel) wird mit ihrer Ausführung begonnen.

```
10:"PROG1":P$="PR _____
   OG1"
20:LPRINT "START
   VON ";P$
290:LF 2
300:LPRINT " << PR          PROG 1
   OGRAMM AUSFUEH
   RUNG >>"
310:LF 2
400:LPRINT "ENDE V
   ON ";
410:LPRINT P$ _____
10:"PROG2":P$="PR
   OG2"
20:LPRINT "START
   VON ";P$
290:LF 2
300:LPRINT " << PR          PROG 2
   OGRAMM AUSFUEH
   RUNG >>"
310:LF 2
400:LPRINT "ENDE V
   ON ";
410:LPRINT P$ _____
10:"PROG3":P$="PR
   OG3"
20:LPRINT "START
   VON ";P$
290:LF 2
300:LPRINT " << PR          PROG 3
   OGRAMM AUSFUEH
   RUNG >>"
310:LF 2
400:LPRINT "ENDE V
   ON ";
410:LPRINT P$ _____
```

Das Kommando RUN führt zu folgendem Ablauf: (ebenso GOTO 10)

START VON PROG1

Ausgeführt wird PROG 1 der erste Zeilen-
block von Zeile 10 bis 410.

<< PROGRAMM AUSFU
EHRUNG >>

ENDE VON PROG1

Das Kommando RUN "PROG-2" liefert:

START VON PROG2

Ausgeführt wird das Programm PROG 2, der zweite Block mit den Zeilennummern 10 bis 410.

<< PROGRAMM AUSFÜHRUNG >>

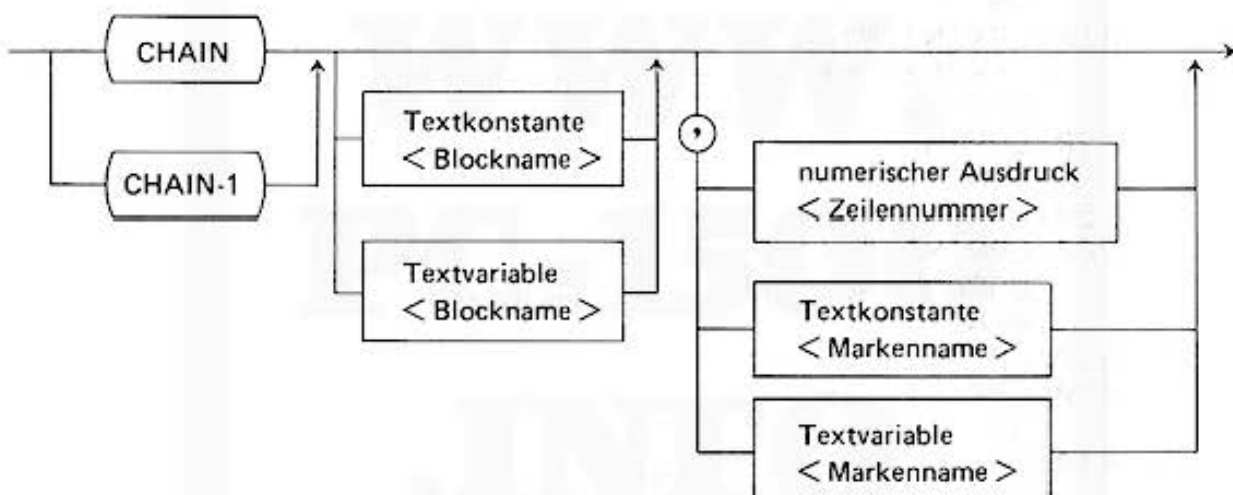
ENDE VON PROG2

und das Kommando gilt für RUN "PROG-3".

CHAIN (Programminstruktion)

Mit dieser Instruktion wird eine Folge von auf dem Magnetband gespeicherten Programme ausgeführt.

Syntax



CHAIN ist eine Programmanweisung und kann nicht als manuelles Kommando verwendet werden. Die CHAIN-Instruktion ermöglicht es Ihnen, ein Programm laufen zu lassen, das so umfangreich ist, daß es nicht auf einmal in den Programmspeicher paßt. Solche Programme müssen unterteilt und mit der CSAVE-Anweisung einzeln auf das Band übertragen werden. Am Ende eines jeden Abschnitts steht die CHAIN-Instruktion. Sie legt fest, welcher Block als nächster Programmteil ausgeführt werden soll. Die Programmteile müssen somit in sequentieller Reihenfolge auf dem Band abgespeichert sein. Denken Sie vor dem Eingeben des nächsten Programmblockes daran, den alten mit NEW zu löschen.

CHAIN lädt den Programmblock von Band 0, CHAIN-1 von Band 1. Textkonstante und Textvariable < Blockname > spezifizieren wie bei der CLOAD-Instruktion den Namen des Blockes, der als nächstes zu laden ist.

Der numerische Ausdruck < Zeilennummer >, sowie die Textkonstante bzw. Textvariable < Markenname > geben an, mit welcher Zeile bzw. an welcher Marke die Ausführung des nächsten Programnteils begonnen werden soll.

Vor dem Laden des Programmblocks wird der Programmspeicher gelöscht, Variablen bleiben jedoch erhalten. Sprünge von Block zu Block sind somit nicht möglich.

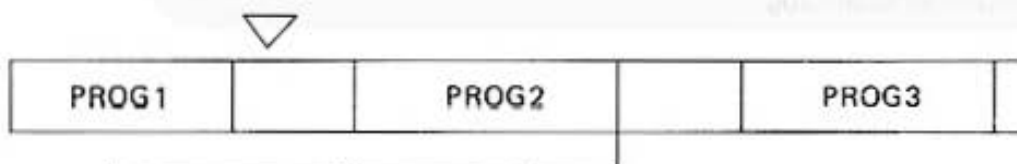
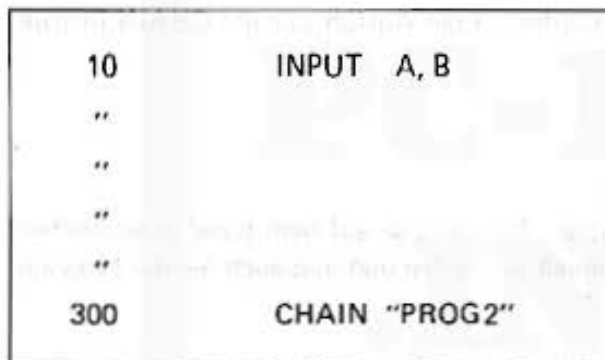
Beispiel:

Die folgenden Diagramme veranschaulichen das Zusammenfügen der Programmblocke:

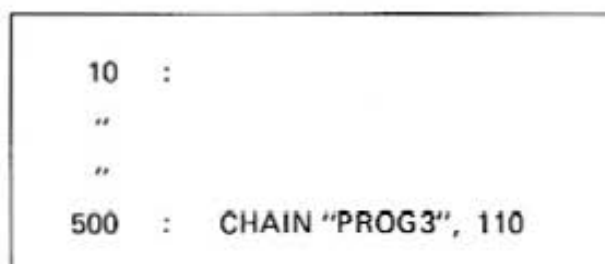
Auf dem Band befinden sich die drei Programnteile mit den Blocknamen: PROG 1, PROG 2, PROG 3

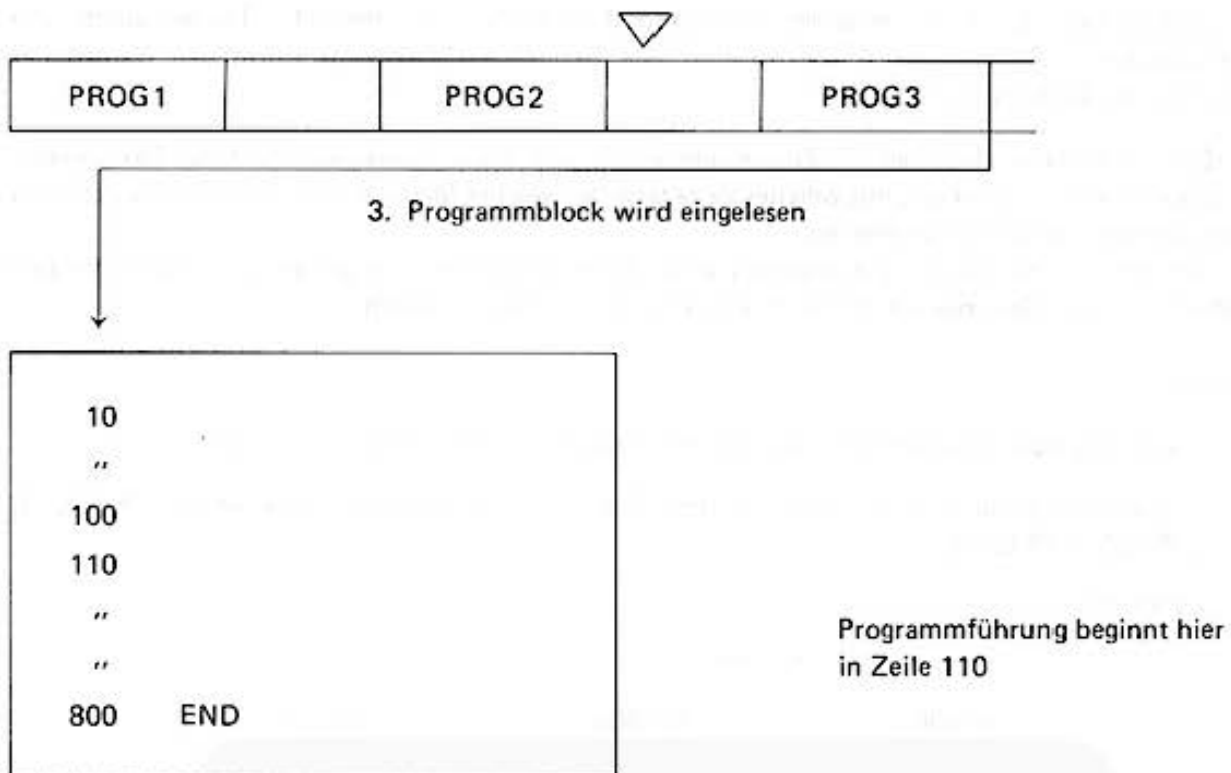


Mit CLOAD "PROG 1" wird der erste Programnteil geladen.



2. Programmblock wird eingelesen





Die Programmteile werden automatisch miteinander verkettet und ausgeführt.

▽ kennzeichnet jeweils die Position des Tonkopfes vor der Ausführung der CHAIN-Instruktion.

Hinweise:

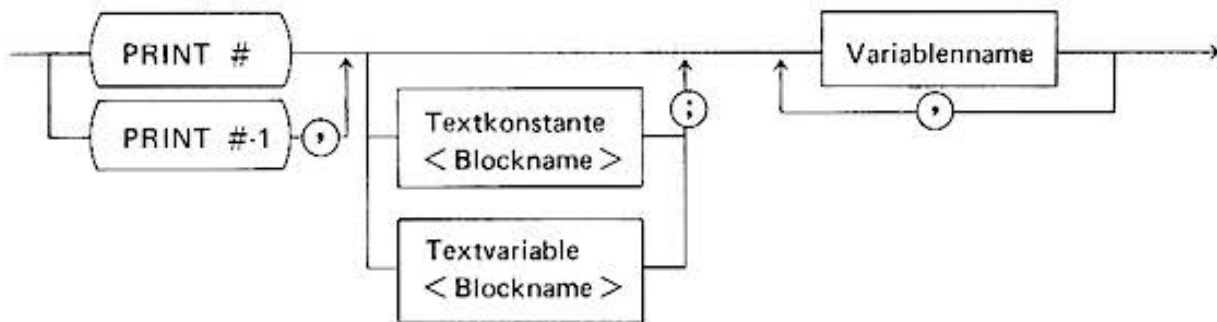
Man sieht, daß die Programmblöcke in der richtigen Reihenfolge auf dem Band abgespeichert sein müssen: Sonst ist es erforderlich, daß Sie manuell eingreifen und dies auch bei der Programmierung berücksichtigen.

Außerdem ist es notwendig, den Tonkopf vor den ersten Programmteil zu bringen. Alles weitere steuert der Rechner selbsttätig.

PRINT

Mit dieser Instruktion werden die Werte von Variablen auf dem Band abgespeichert.

Syntax

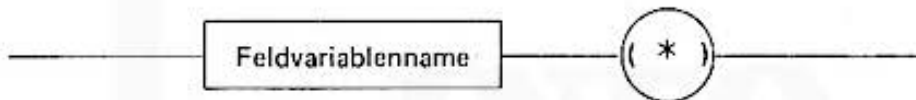


Der PC-1500 unterscheidet zwischen dem Programm- und Datenspeicher. Mit der PRINT # oder PRINT #-1 Instruktion werden die Werte einer Variablen oder einer Gruppe von Variablen auf dem Band 0 oder Band 1 abgespeichert. Die Textkonstante und Textvariable < Blockname > haben die gleiche Bedeutung wie bei der CSAVE-Anweisung und legen den Namen des Datensatzes auf dem Band fest.

Die Speicherung kann nur in der Betriebsart RUN sowohl manuell als auch programmgesteuert erfolgen.

Der Rechner kann Datenblöcke von Programmblocken unterscheiden. Es ist daher möglich, die zu einem Programm gehörenden Daten mit dem gleichen Namen zu versehen.

Geben Sie einen einzelnen Variablennamen oder eine Folge von Variablennamen an, werden die Inhalte dieser aufgeführten Variablen abgespeichert. Die Namen von Feldvariablen sind einzugeben als:



z.B. B (*) bzw. \$\$ (*)

Es werden dann alle Elemente des Feldes auf dem Band aufgezeichnet. Die Variablen A – Z können als Feld mit @ (*) und die Textvariablen A\$ – Z\$ mit @\$ (*) übertragen werden.

Hinweise:

- Denken Sie bitte rechtzeitig daran, daß das Magnetband für die Aufnahme vorbereitet sein muß.
- Einzelne Elemente eines Feldes können nicht direkt abgespeichert werden.

Beispiele:

```
(1) 10 DIM B (10)
    20 FOR I = 1 TO 10
    30 B (I) = I
    40 NEXT I
    50 PRINT # "DATENSATZ"; I, B (*)
```

Die Inhalte der Variablen I und des Feldes B werden auf das Band unter dem Namen DATENSATZ abgespeichert.

```
(2) 10
    :
    :
    1000 PRINT # "ERGEBNISSE"; A, B, C, SS
```

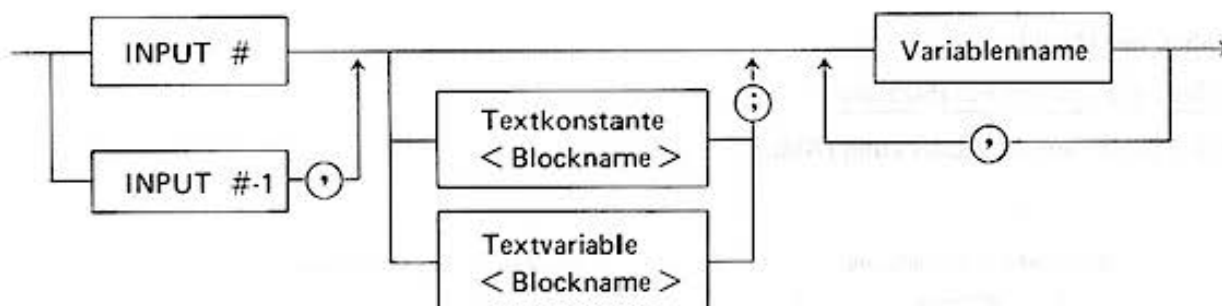
Alle Variablen werden unter dem Datensatz mit dem Namen ERGEBNISSE abgespeichert.

```
(3) .
    .
    1000 PRINT # A, B, C, SS, TS
```


INPUT

Mit dieser Instruktion werden Daten vom Band gelesen und Variablen zugewiesen.

Syntax



Mit der INPUT #-Anweisung kann man die Daten, die mit der PRINT # Instruktion auf das Band geschrieben wurden, wieder lesen und sie Variablen zuweisen.

Die Sprachelemente haben die gleiche Bedeutung wie in der PRINT # Anweisung. INPUT # liest von Band 0, INPUT #-1 von Band-1, Textkonstante und Textvariable bezeichnen den Namen des zu lesenden Datensatzes. Der einzelnen Variablen oder der Folge von Variablen und Feldern werden die Werte zugewiesen, die bei der PRINT# Instruktion abgespeichert wurden. Felder sind so wie bei der PRINT #-Instruktion anzugeben.

Wenn sich die Anzahl der Variablen zwischen der INPUT # und entsprechenden PRINT #-Anweisung unterscheiden, tritt folgende Aktion ein:

- Die Anzahl der INPUT # Variablen ist größer:
Die restlichen Variablen erhalten den Wert 0 bzw. leeren Text.
- Die Anzahl der PRINT # Variablen ist größer:
Die restlichen Variablen auf dem Band werden ignoriert.

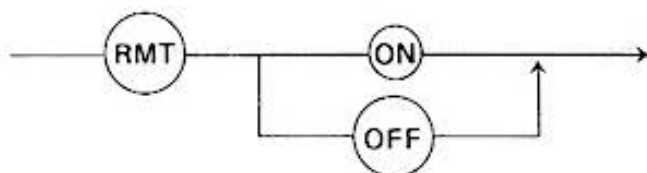
Beispiele:

- (1) 10 INPUT # A, B, C
- (2) 10 INPUT # "ERGEBNISSE"; A, B, C, SS

RMT

Mit dieser Instruktion wird der REMOTE-Schalter des zweiten Bandgerätes gesteuert.

Syntax



Die Instruktion übernimmt die Funktion des REMOTE-Schalters des CE-150 für Bandgerät-1. Anstatt den Schalter zu bedienen, so wie beim Bandgerät-0, geben Sie jetzt die Instruktionen RMT ON oder RMT OFF ein.

An der Buchse REM 1 liegt dann das entsprechende Remote-Signal an.

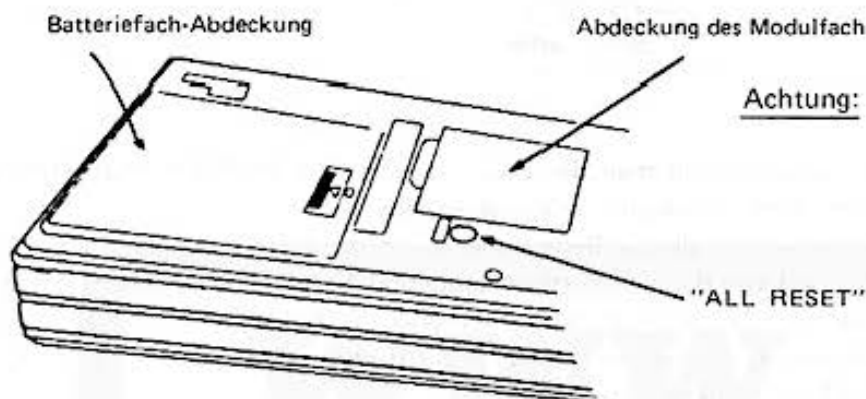
20. CE-151 / CE-155 (Option)

Die Steckmodule CE-151/CE-155 dienen zur Hauptspeichererweiterung um 4096 bzw. 8192 Bytes. Sie können nicht zusammen verwendet werden.

Die Module haben keine eigene Stromversorgung und können deshalb im ausgebauten Zustand keine Informationen speichern.

Einbau der Module

- 1) Sie schalten den PC-1500 aus.
- 2) Sie entfernen die Abdeckung (Abb.).



Achtung: Berühren Sie nicht die Kontaktleisten am PC-1500 und CE-151/155.

- 3) Sie legen das CE-151/155 (Beschriftung sichtbar) in das Modulfach und schieben es vollständig in die Kontaktleiste.
- 4) Sie schließen das Modulfach.
- 5) Sie schalten den PC-1500 ein.
- 6) Der Computer meldet sich mit

NEW Ø ? : CHECK

- 7) Sie geben ein: **CL** NEW Ø **ENTER**
- 8) Sie geben ein: STATUS Ø **ENTER**

Auf der Anzeige erscheint 5946 bzw. 10042 (Die Aufrüstung ist vom Computer erkannt worden.).

Ausbau des Moduls

- 1) Sie verfahren sinngemäß nach den Punkten 1) bis 7) von "Einbau der Module".
- 2) Sie geben ein: STATUS Ø **ENTER**
Auf der Anzeige erscheint 1850 (Die Abrüstung ist vom Computer erkannt worden.).

ANHANG

A. ASCII-Code Tabelle

ASCII Zeichen	Entsprechender Code		
	Binär	Dezimal	Hexadezimal
space	00100000	32	20
!	00100001	33	21
"	00100010	34	22
#	00100011	35	23
\$	00100100	36	24
%	00100101	37	25
&	00100110	38	26
'	00100111	39	27
(00101000	40	28
)	00101001	41	29
*	00101010	42	2A
+	00101011	43	2B
,	00101100	44	2C
-	00101101	45	2D
.	00101110	46	2E
/	00101111	47	2F
0	00110000	48	30
1	00110001	49	31
2	00110010	50	32
3	00110011	51	33
4	00110100	52	34
5	00110101	53	35
6	00110110	54	36
7	00110111	55	37
8	00111000	56	38
9	00111001	57	39
:	00111010	58	3A
;	00111011	59	3B
<	00111100	60	3C
=	00111101	61	3D
>	00111110	62	3E
?	00111111	63	3F

ASCII Zeichen	Entsprechender Code			ASCII Zeichen	Entsprechender Code		
	Binär	Dezimal	Hexadezimal		Binär	Dezimal	Hexadezimal
@	01000000	64	40		01100000	96	60
A	01000001	65	41	a	01100001	97	61
B	01000010	66	42	b	01100010	98	62
C	01000011	67	43	c	01100011	99	63
D	01000100	68	44	d	01100100	100	64
E	01000101	69	45	e	01100101	101	65
F	01000110	70	46	f	01100110	102	66
G	01000111	71	47	g	01100111	103	67
H	01001000	72	48	h	01101000	104	68
I	01001001	73	49	i	01101001	105	69
J	01001010	74	4A	j	01101010	106	6A
K	01001011	75	4B	k	01101011	107	6B
L	01001100	76	4C	l	01101100	108	6C
M	01001101	77	4D	m	01101101	109	6D
N	01001110	78	4E	n	01101110	110	6E
O	01001111	79	4F	o	01101111	111	6F
P	01010000	80	50	p	01110000	112	70
Q	01010001	81	51	q	01110001	113	71
R	01010010	82	52	r	01110010	114	72
S	01010011	83	53	s	01110011	115	73
T	01010100	84	54	t	01110100	116	74
U	01010101	85	55	u	01110101	117	75
V	01010110	86	56	v	01110110	118	76
W	01010111	87	57	w	01110111	119	77
X	01011000	88	58	x	01111000	120	78
Y	01011001	89	59	y	01111001	121	79
Z	01011010	90	5A	z	01111010	122	7A
√	01011011	91	5B	{	01111011	123	7B
¥	01011100	92	5C	:	01111100	124	7C
π	01011101	93	5D	}	01111101	125	7D
^	01011110	94	5E	~	01111110	126	7E
-	01011111	95	5F	■	01111111	127	7F

Hinweis:

Folgende Zeichen werden vom Druckwerk nicht ausgegeben:

□ - { : } ■

B. Rechenbereich

Funktionen	Rechenbereich
$y \wedge x (y^x)$	$-1 \times 10^{100} < x \log y < 100$ $\left(\begin{array}{l} y = 0, x \leq 0 : \text{ERROR 39} \\ y = 0, x > 0 : 0 \\ y < 0, x \neq \text{Ganzzahl} : \text{ERROR 39} \end{array} \right)$
SIN x COS x TAN x	<p>Die folgenden Werte sind für TAN x nicht zulässig</p> <p>DEG: $x < 1 \times 10^{10}$ RAD: $x < \frac{\pi}{180} \times 10^{10}$ GRAD: $x < \frac{10}{9} \times 10^{10}$</p> <p>DEG: $x = 90 (2n-1)$ RAD: $x = \frac{\pi}{2} (2n-1)$ GRAD: $x = 100 (2n-1)$ (n: Ganzzahl)</p>
SIN ⁻¹ x COS ⁻¹ x	$-1 \leq x \leq 1$
TAN ⁻¹ x	$ x < 1 \times 10^{100}$
LN x LOG x	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
EXP x	$-1 \times 10^{100} < x \leq 230.2585092$
\sqrt{x}	$0 \leq x < 1 \times 10^{100}$

Andere Funktionen können nur berechnet werden, wenn der Wert für x im angegebenen Bereich bleibt:

$$1 \times 10^{-99} \leq |x| < 1 \times 10^{100} \text{ and } 0$$

(Ex.)

$0 \wedge 0$	ENTER	→	ERROR 39
$0 \wedge 5$	ENTER	→	0
$(-4) \wedge 0.5$	ENTER	→	ERROR 39
$-4 \wedge 0.5$	ENTER	→	-2

In der Regel ist der Fehler kleiner als ± 1 an der letzten angezeigten Stelle innerhalb des oben genannten Rechenbereichs.

Bei der wissenschaftlichen Darstellung gilt dies für die letzte Stelle der Mantisse.

C. Fehlermeldungen

1. Syntax – Fehler,

z.B.

```
10 : GOTO  
      (Sprungziel fehlt)
```

```
10 : NEW (keine programmierbare Instruktion)
```

< Anzeige > ERROR 1 IN 10

2. Zu einer NEXT-bzw. RETURN-Instruktion existiert keine entsprechende FOR- bzw. GOSUB-Instruktion.

z.B. 10 : FOR A=1 TO 10
 100 : NEXT B

< Anzeige > ERROR 2 IN 100

4. Zu einer READ-Instruktion existieren keine entsprechenden Daten bzw. die DATA-Instruktion fehlt ganz.

z.B. 10 : READ X, Y
 20 : DATA 10
 30 : END

< Anzeige > ERROR 4 IN 10

5. Die Bezeichnung einer Feldvariablen wurde bereits verwendet.

z.B. 10 : DIM A (10, 10)
 20 : DIM A (5)

< Anzeige > ERROR 5 IN 20

6. Eine Feldvariable wurde ohne DIM-Instruktion benutzt.

z.B. 10 : CLEAR
 20 : A (3) = 1

< Anzeige > ERROR 6 IN 20

7. Einer Textvariablen wurde ein numerischer Wert zugewiesen oder umgekehrt.
 z.B. 10 : A\$ = 10
 10 : FOR A\$ = 1 TO 10

 < Anzeige > ERROR 7 IN 10
8. Bei der Vereinbarung der Feldvariablen sind mehr als 2 Dimensionen angegeben.
 z.B. 10 : DIM A (3,4,5)

 < Anzeige > ERROR 8 IN 10
9. Der Index überschreitet das vereinbarte Datenfeld.
 z.B. 10 : DIM A (3)
 20 : A (4) = 1

 < Anzeige > ERROR 9 IN 20
10. Zur Einrichtung neuer Variablen steht nicht mehr genügend Speicherplatz zur Verfügung.
 z.B. Eingabe Anzeige
 MEM 7
 AB = 10 ERROR 10
11. Die Sprungzieladresse existiert nicht.
 z.B. 10 : INPUT "X"; X: GOTO5

 < Anzeige > ERROR 11 IN 10
12. Das Format einer USING-Instruktion ist falsch.
 z.B. 100 : PRINT USING " ###A #"; 10

 < Anzeige > ERROR 12 IN 100
13. Beim Programmieren wird die Kapazität des Haupt- bzw. RESERVE-Speichers überschritten.
 z.B. Eingabe Anzeige
 MEM 7
 15 A = A + 1 ERROR 13
14. Die Kapazität des BASIC-Stack wurde durch eine FOR-NEXT-Schleife bzw. durch ein Zwischenergebnis eines arithmetischen Ausdrucks überschritten.

15. Die Kapazität des BASIC-Stack wurde durch eine GOSUB-Ebene bzw. beim Zerlegen eines mathematischen Ausdrucks überschritten.
16. Der angegebene Wert liegt über 1E 100 oder unter -1E 100.
z.B. 123E99

Der äquivalente Dezimalwert zu einer Hexadezimalzahl überschreitet 65535.
z.B. &1FFAB

< Anzeige > ERROR 16
17. Datenart (Zahlen, Zeichenfolgen) ist für den Rechenausdruck nicht zulässig.
z.B. 1 + "A"
18. Die Zahl der Argumente entspricht nicht der Definition der Funktion.
z.B. LEFT \$ ("ABC")

 SIN (30, 60)
19. Die Feldvereinbarung liegt außerhalb des zulässigen Wertes.
z.B. 10 : DIM A (256)

< Anzeige > ERROR 19 IN 10
20. Der Variablenname einer Standardvariablen wurde unvollständig eingegeben.
z.B. 10 : @ = 123
 10 : @\$ = "ABC"

< Anzeige > ERROR 20 IN 10
21. Die Instruktion erfordert eine Variable.
z.B. 10 : FOR 1 = 0 TO 10

< Anzeige > ERROR 21 IN 10
22. Ein Programm wird eingeladen, und es ist keine Speicherkapazität mehr vorhanden.
23. Der numerische Ausdruck zum Einlesen der Uhrzeit ist fehlerhaft.
z.B. TIME = 131005.10

26. Die Instruktion ist in der gewählten Betriebsart unzulässig.

z.B. RUN MODE NEW **ENTER**

27. Es wurden Schlüsselwörter über Definable Keys für nicht angeschlossene Peripherie abgerufen.

z.B.	Eingabe	Anzeige
	DEF I ENTER	ERROR 27

28. Nach INPUT oder AREAD wurde der Variablen ein BASIC-Schlüsselwort über Definable Key zugewiesen.

z.B.	10 : INPUT AS	
	RUN MODE	Anzeige
	RUN ENTER	?
	DEF W ENTER	

30. Eine Zeilennummer wurde eingegeben, die größer ist als 65535.
(65280 bis 65539 : ERROR 1)

z.B. 1022235 A = 10 **ENTER**

32. Der Cursor ist soweit nach rechts gerückt, daß die Variablen-Eingabe nach INPUT nicht mehr angezeigt werden kann

z.B. 10 : G_CURSOR 152
 20 : INPUT X

< Anzeige > ERROR 32 IN 20

36. Das Format ist für den auszugebenden Ausdruck zu klein gewählt.

z.B. 10 : USING "####.##"
 20 : PRINT 12345

< Anzeige > ERROR 36 IN 20

37. Das Ergebnis eines arithmetischen Ausdrucks überschreitet den Wert 9,999999999E99.

38. Es wurde eine Division durch 0 durchgeführt.

z.B. 10 : PRINT 5/0

< Anzeige > ERROR 38 IN 10

- 39.** Der numerische Ausdruck ist für die Funktion unzulässig.
- z.B. LN (-10)
 ASN (1.5)
 SQR (-1)
- Fehlermeldungen bei Kassettenbetrieb
- 40.** Ungenügend spezifizierter Ausdruck.
- 42.** Die auf Kassette gespeicherten Dateien sind zu umfangreich und können nicht eingeladen werden.
- 43.** Die auf Kassette gespeicherten Programminformationen stimmen bei Überprüfung mit CLOAD? nicht mit den im Rechner befindlichen überein.
- 44.** Prüfnummerfehler
(Tritt nur beim Rückladen in den Rechner auf).
Information auf dem Band ist nicht korrekt lesbar.
- Fehlermeldungen beim Ausgeben über Druckwerte
- 70.** Der zulässige Koordinatenbereich von
- 2048 <= (X, Y) <= 2047
ist überschritten worden.
- 71.** Das Papier wurde im TEXT-Mode um mehr als 10,24 cm oder um zuviele Zeilen zurückbewegt.
- 72.** Der bei TAB oder LCURSOR definierte Ausdruck ist nicht zulässig.
- 73.** Die Instruktion kann in der gewählten Betriebsart (GRAPH/TEXT) nicht ausgeführt werden.
- 74.** Die Anzahl der auf LINE oder RLINE folgenden Kommas ist zu groß.
- Hinweis:
Die Eingabe von mehr als 7 Kommas führt zu einem Fehler. Mehr als 6 Kommas haben einen Fehler zur Folge, wenn als erstes eine Leerstelle eingegeben wird.
- 76.** Das Ergebnis eines arithmetischen Ausdrucks kann nach einer LPRINT-Instruktion nicht in einer Zeile ausgedruckt werden (im TEXT-Mode).

78. (a) Das Druckwerk befindet sich in der Position Farbpatronenwechsel.

(b) Die Fehlermeldung 80 ist nicht behoben worden.

Der Fehler tritt auf, wenn aus einem dieser beiden Gründe eine Druckanweisung nicht angeführt werden kann.

Resultiert Fehlermeldung 78 aus Fehlermeldung 80, muß der Computer aus- und wieder eingeschaltet werden. Durch den Anschluß des Ladegerätes EA-150 ist der Druckerbetrieb wieder möglich. Gleichzeitig werden die Batterien aufgeladen.

79. Der Schreibkopf stand in einer irregulären Position, als das COLOR-Kommando gegeben wurde.

80. Die Batterien im CE-150 sind erschöpft und müssen durch Anschluß des EA-150 wieder aufgeladen werden.

177. – 181. Beim Programmieren überschneiden sich Programm und Datenfeld.

0 224.–241. Den Variablen nach der INPUT oder AREAD-Instruktion sind falsche Daten zugewiesen worden.

z.B.

10 : INPUT A

RUN MODE

Eingabe

Anzeige

RUN
123 PRINT

?
ERROR 240

D. Liste der Funktionen und Instruktionen

Funktionen

Funktion	Abkürzung	Bemerkungen	Seite
ABS	AB.	Absolutbetrag	24
ACS	AC.	Arcuscosinus (\cos^{-1})	22
AND	AN.	Logischer Operator "UND"	28
ASC		Umwandlung von Zeichen in ASCII-Code (Umkehrfunktion von CHR\$)	89
ASN	AS.	Arcussinus (\sin^{-1})	22
ATN	AT.	Arcustangens (\tan^{-1})	22
CHR\$	CH.	Wandelt ASCII-Code in Zeichen um. (Umkehrfunktion von ASC)	90
COS		Cosinus X	22
DEG		Umrechnungsfunktion Dezimalsystem \Rightarrow Sexagesimalsystem	23
DMS	DM.	Umrechnungsfunktion Sexagesimalsystem \Rightarrow Dezimalsystem	23
EXP	EX.	e^x ; Exponentiation zur Basis e	24
INKEY\$	INK. INKE. INKEY.	Zeichenfunktion; das Zeichen einer gedrückten Taste ergibt den Funktionswert.	95
INT		Integerfunktion; zur Ermittlung des ganzzahligen Anteils eines numerischen Ausdrucks.	25
LEFT\$	LEF. LEFT.	Textfunktion; entnimmt aus einer Zeichenfolge eine spezifizierte Zeichenanzahl von links.	93
LEN		Ermittlung der Zeichenanzahl eines Textaus- druckes.	91
LOG	LO.	$\log_{10} x$; allgemeiner Logarithmus zur Basis 10.	24
LN		$\log_e x$; natürlicher Logarithmus zur Basis e.	24
MEM	M. ME.	Ermittlung der Anzahl der freien Bytes im Haup- tspeicher (s. STATUS 0)	31
MID\$	MI. MID.	Textfunktion; entnimmt eine spezifizierte An- zahl von Zeichen aus der Mitte einer Zeichen- kette.	95
NOT	NO.	logische Negation	27
OR		logischer Operator "ODER"	29

Funktion	Abkürzung	Bemerkungen	Seite
π /PI		Abruf der Kreiszahl (= 3.141592654)	25
POINT	POI. POIN.	Auslesen von Punktmustern der Anzeige; der Funktionswert ist numerisch.	86
RIGHT\$	RI. RIG. RIGH. RIGHT.	Textfunktion; entnimmt aus einer Zeichenfolge die Zeichen vom rechten Ende.	93
RND	RN.	Erzeugung einer Zufallszahl	26
SGN	SG.	Ermittlung des Vorzeichens eines numerischen Ausdrucks.	25
SIN	SI.	Sinus; $\sin x$	22
$\sqrt{\quad}$ /SQR	SQ.	Quadratwurzel	25
STATUS	STA. STAT. STATU.	Angabe von freiem und belegtem Speicherplatz und der letzten bearbeiteten Zeilennummer (STATUS 0, 1, 2 oder 3)	31
STR\$	STR.	Textfunktion; wandelt den Wert eines numerischen Ausdrucks in die dezimale Zeichenfolge um.	91
TAN	TA.	Tangens; $\tan x$	22
TIME	TI. TIM.	Lesen und Schreiben der Zeit (Monat, Tag, Stunde, Minute, Sekunde)	30
VAL	V. VA.	Textfunktion; wandelt eine als Zeichenfolge eingegebene Zahl in ihren numerischen Wert um.	92
\wedge		allgemeine Exponentialfunktion.	16

B. Instruktionen

Instruktion	Abkürzung	Bemerkungen	Seite
AREAD	A. AR. ARE. AREA.	Automatische Wertzuweisung zu einer spezifizierten Variablen nach Programmstart über Definable Keys.	64
ARUN	ARU.	Automatisches Starten eines Programms beim Einschalten des PC-1500.	65
BEEP	B. BE. BEE.	Akustisches Signal in Lautstärke, Frequenz und Länge veränderbar.	88
CLEAR	CL. CLE. CLEA.	Löschen aller Variablen und Felder aus dem Hauptspeicher. Die Standardvariablen werden auf 0 gesetzt.	66
CLS		Löschen der Anzeige	88
CONT	C. CO. CON.	Fortsetzung der Programmausführung nach STOP oder BREAK (nur im RUN-Mode).	65
CURSOR	CU. CUR. CURS. CURSO.	Bestimmt die Schreibposition auf der Anzeige.	80
DATA	DA. DAT.	Definition von Datenfeldern in Verbindung mit READ.	43
DEGREE	DE. DEG. DEGR. DEGRE.	Setzen der Winkleinheit GRAD.	22
DIM	D. DI.	Deklaration von Feldvariablen (Arrays)	60
END	E. EN.	Abschlußkommando eines Programms	49
FOR TO STEP NEXT	F. FO. STE. N. NE. NEX.	Einrichtung von Programmschleifen	58
GCURSOR	GC. GCU. GCU. GCUR. GCURS. GCURSO.	Bestimmt die Schreibposition auf der Anzeige bezogen auf das Raster der 7 x 156 Punktmatrix.	81
GOSUB	GOS. GOSU.	Aufruf eines Unterprogramms. Rücksprung mit RETURN.	50

Instruktion	Abkürzung	Bemerkungen	Seite
GOTO	G. GO. GOT.	Verzweigt das Programm zu einer spezifizierten Zeile. Im RUN-Mode: Start des Programms an einer spezifizierten Zeile.	49
GPRINT	GP. GPR. GPRI. GPRIN.	Ausgabe von spezifizierten Mustern auf der Anzeige.	82
GRAD	GR. GRA.	Setzen der Winkeleinheit NEUGRAD	22
IF		Prüft nachfolgende Bedingung auf "wahr" oder "unwahr" (wird in Verbindung mit THEN benutzt)	56
INPUT	I. IN. INP. INPU.	Weist Variablen Werte zu, die über die Tastatur eingegeben werden.	77
LET	LE.	Wertzuweisung für Variablen	42
LIST	L. LI. LIS.	Auflisten von Programmzeilen in der Anzeige (nur im PRO-Mode)	67
LOCK	LOC.	Blockiert die Mode-Taste	12
NEXT	N. NE. NEX.	S. FOR ... TO ... STEP NEXT	58
NEW		Im PRO-Mode: löscht den Inhalt des Hauptspeichers. Im RESERVE-Mode: löscht den Inhalt des RESERVE-Speichers. NEW Ø: Initialisierung des PC-1500	67
ON ERROR	O. ER. ERR. ERRO.	Verzweigt den Programmablauf bei Fehlermeldungen zu einer spezifizierten Zeile (wird in Verbindung mit GOTO verwendet) Ausnahme: Syntaxfehler	54
ON GOSUB	O. GOS. GOSU.	Ruft in Abhängigkeit vom ON-Argument ein spezifiziertes Unterprogramm auf.	55
ON GOTO	O. G. GO GOT.	Verzweigt den Programmablauf in Abhängigkeit vom ON-Argument zu einer spezifizierten Programmzeile.	52
PAUSE	PA. PAU. PAUS.	Ausgabeeinweisung; nach Ablauf von ca. 0,85 Sek. wird das Programm fortgesetzt.	76

Instruktion	Abkürzung	Bemerkungen	Seite
POINT	POI. POIN.	Auslesen der Punktmuster auf der Anzeige (Umkehrfunktion von GPRINT)	86
PRINT	P. PR. PRI. PRIN.	Ausgabeanweisung	73
RADIAN	RAD. RADI. RADIA.	Setzen der Winkeleinheit RADIAN	22
RANDOM	RA. RAN. RAND. RANDO.	Initiiert eine neue Folge von Zufallszahlen (in Verbindung mit RND)	26
READ	REA.	Weist Variablen Werte aus einem Datenfeld zu (in Verbindung mit DATA).	44
REM		Definiert den Rest der Zeile als Kommentar.	47
RESTORE	RES. REST. RESTO. RESTOR.	Definieren der nächsten zu bearbeitenden DATA Anweisung.	46
RETURN	RE. RET. RETU. RETUR.	Rücksprung in die auf den Unterprogrammaufruf folgende Zeile (in Verbindung mit GOSUB).	52
RUN	R. RU.	Start eines Programms (nur im RUN-Mode).	63
STOP	S. ST. STO.	Unterbrechung der Programmausführung	48
THEN	T. TH. THE.	S. IF-Instruktion	56
TRON	TR. TRO.	Automatischer Stopbefehl nach Abarbeiten jeder Zeile (Debugging, Trace)	66
TROFF	TROF.	Aufhebung von TRON	66
UNLOCK	UN. UNL. UNLO. UNLOC.	Aufheben des LOCK-Befehls	12
USING	U. US. USI. USIN.	Bestimmung des Ausgabeformats.	68
WAIT	W. WA. WAI.	Festlegen einer Zeitspanne, nach der nach einer PRINT-Instruktion der Programmablauf auto- matisch startet.	77

3. Instruktionen für Druckerausgabe

Instruktion	Abkürzung	Bemerkungen	Seite
COLOR	COL. COLO.	Farbstiftauswahl für den Druck	110
CSIZE	CSI. CSIZ.	Bestimmung der Zeichengröße und damit der Anzahl der Zeichen pro Zeile auf dem Drucker.	108
GLCURSOR	GL. GLC. GLCU. GLCUR. GLCURS. GLCURSO.	Bestimmung der Startposition des Drucks in einer spezifizierten X/Y-Koordinate (nur im GRAPH-Mode).	116
GRAPH	GRAP.	Wahl der Betriebsart GRAPH.	106
LF		Zeilenvorschub.	107
LINE	LIN.	Drucken von spezifizierten Strecken (nur im GRAPH-Mode).	118
LLIST	LL. LLI. LLIS.	Drucken von Programmlisten.	109
LPRINT	LP. LPR. LPRI. LPRIN.	Ausgabeinstruktion für den Drucker.	111
RLINE	RL. RLI. RLIN.	Ausgabe von Strecken, wobei die augenblickliche Position des Druckkopfes als Startposition angenommen wird (nur GRAPH-Mode).	121
ROTATE	RO. ROT. ROTA. ROTAT.	Bestimmung der Druckrichtung auf einer X/Y-Achse (nur GRAPH-Mode).	114
SORGN	SO. SOR. SORG.	Festlegung des Koordinatensprungs für eine LINE-Instruktion (nur GRAPH-Mode)	117
TAB		Bestimmung der Druckposition (nur TEXT-Mode).	110
TEST	TE. TES.	Ausführung eines Druckertests.	109
TEXT	TEX.	Wahl der Betriebsart TEXT.	106

Instruktionen für den Kassettenbetrieb

Instruktion	Abkürzung	Bemerkungen	Seite
CHAIN	CHA. CHAI.	Programmgesteuertes Rückladen von spezifizierten Programmblöcken.	136
CLOAD	CLO. CLOA.	Rückladen von spezifizierten Programmen.	132
CLOAD?	CLO.? CLOA.?	Vergleich eines im PC-1500 befindlichen Programms mit dem gleichen Kassettenprogramm.	133
CSAVE	CS. CSA. CSAV.	Abspeichern eines Programms auf Kassette.	130
INPUT #	I. # IN. # INP. # INPU. #	Rückladen von Daten.	141
MERGE	MER. MERG.	Rückladen von Programmen, wobei die im Hauptspeicher befindlichen Programme erhalten bleiben.	134
PRINT #	P. # PR. # PRI. # PRIN. #	Abspeichern von Daten.	139
RMTOFF	RM.OF. RMTOF.	Aufheben der Fernbedienungsfunktion REM 1 (Rekorder-1).	141
RMTON	RM.O. RMTO.	Aktivieren der Fernbedienungsfunktion REM 1 (Rekorder-1).	141

E. Referenzliste BASIC

Bent, Robert J. et al., Business BASIC. New York: Brooks-Cole, 1980.

Bitter, Gray G./Wilson Y. Gateley, BASIC for Beginners. 2nd ed. New York: McGraw-Hill Book Co., 1978.

Chou, George T., Computer Programming in BASIC with Business Applications. New York: Harper & Row, 1978.

Dwyer, Thomas A./Margot A. Critchfield, BASIC & the Personal Computer. New York: Addison-Wesley, 1978.

Feichtinger, Herwig, BASIC für Mikrocomputer. München: Franzis, 1980.

Gottfried, B.S., Programmieren in BASIC. Düsseldorf: McGraw-Hill Book Co., 1978.

Sack, John/Judith Meadows, BASIC – eine Einführung. München: Oldenbourg, 1979.

Schärf, Julius, BASIC für Anfänger. München: Oldenbourg, 1979.

Schneider, W., Einführung in BASIC, 2. Aufl., Wiesbaden: Vieweg-Verlag, 1980.

Schwill, Wolf D./Roland Weibezahn, Einführung in die Programmiersprache BASIC. Wiesbaden: Vieweg, 1979.

WWW.
PC-1500
.INFO

F. PC-1500 Spezifikationen

Modell:	PC-1500 Taschencomputer
Rechenstellen:	10 Stellen Mantisse, 2 Stellen Exponent
Rechensystem:	AEL mit Hierarchie
Anzeige:	Flüssigkristall-Anzeige 26 Stellen in 7 x 156 Matrix
Tastatur:	65 Tasten Alphatastatur (Schreibmaschine) Zifferntastatur (10er Block) Funktionstasten Programmierbare Tasten Vorprogrammierte Tasten
Programmiersprache:	BASIC
CPU:	CMOS 8 BIT
Betriebssystem:	ROM 16 KBYTES
Speicherkapazität:	RAM 3.5 KBYTES, davon 900 Bytes System 80 Bytes Eingabepuffer 196 Bytes Stackregister 1.850 Bytes BASIC-Programm 624 Bytes Standard-Variable 188 Bytes RESERVE-Speicher
Speicherschutz:	Programm-Daten-RESERVE-Speicher sind beim Ausschalten geschützt.
Stromversorgung:	6 Volt, 4 Stück Trockenbatterien (SUM-3, AA, R 6) oder Netzadapter
Stromverbrauch:	0,13 W bei 6 Volt
Betriebszeit:	ca. 50 Stunden mit einem Batteriesatz
Betriebstemperatur:	0°C bis 40°C
Abmessungen:	195 x 86 x 25,5 (B, T, H in mm)
Gewicht:	ca. 375 Gramm (mit Batterien)
Zubehör:	Schutzbehälter 4 Batterien 2 Tastaturschablonen 1 Betriebsanleitung 1 Programmsammlung
Optionen:	CE-150 : Plotter/Drucker mit integrierter Kassettengerät-Schnittstelle CE-151 : 4 KBYTES-RAM Speichererweiterung CE-155 : 8 KBYTES-RAM Speichererweiterung

G. Technische Daten des Druckers (Option)

Zeichen/Zeile:	4, 5, 6, 7, 9, 12, 18, oder 36 abhängig von der Formatwahl
Zeichengröße:	10,8 x 7,2 mm bis 1,2 x 0,8 abhängig von der Formatwahl
Druckgeschwindigkeit:	max. 11 Zeichen/sek. bei 36 Zeichen/Zeile
Druckrichtung:	auf x/y-Achse mit jeweils 180° Zeichendrehung
Druckfarben:	4 (schwarz, blau, grün, rot)
Druckprinzip:	x-y plotting
Papiertransport:	manuell oder programmkontrolliert
Papier:	58 x 30 (mm) holzfrei auf Kern, EA-1500P
Farbpatronen:	EA-850B oder EA-850C







SHARP CORPORATION

OSAKA, JAPAN