

HEWLETT-PACKARD

HP-75

REFERENZHANDBUCH





HP-75

Referenzhandbuch

Oktober 1982

00075-90014

Inhalt

Einführung	5
Abschnitt 1: Betrieb des HP-75	6
Tastefeld- und Anzeigesteuerung	6
Tastenfunktionen	6
Tastenkombinationen	8
Kontrolle der Anzeige	9
Funktionen zur Manipulation von Zeichen	9
Syntax-Richtlinien	9
Filemanipulationen	10
Filetypen	10
Filespezifikatoren	10
Fileparameter	10
Befehle zur Fileorganisation	11
Operatoren und Funktionen	11
Rangfolge von Operatoren	11
Numerische Genauigkeit	11
Zahlenbereich	12
Variablen	12
Numerische Funktionen	12
Numerische Ausdrücke	13
TIME-Modus Operationen	13
APPT-Modus Operationen	13
Kartenleser-Operationen	14
HP-IL Operationen	14
Tastenbelegungen	15
Abschnitt 2: Programmierkonzepte	16
Ausführen von Programmen	16
Editieren von Programmen	16
Grundlegende Anweisungen	16
Verzweigungen, Schleifen und Unterprogramme	16
Programmtimer	17
Felder	17
Strings	17
Stringvariablen	17
Stringausdrücke	17
Stringfunktionen	17
Benutzerdefinierte Funktionen	18
Speichern und Zurücklesen von Daten	18
Daten innerhalb von Programmen	18
Datenfiles	18
Programmaufrufe	18
Lokale und globale Deklarationen	18
Ausgabeformatierung	19
Verfolgungsoperationen	19

Abschnitt 3: Instruktionssatz des HP-75	20
Operatoren	20
Arithmetische Operatoren	20
Vergleichsoperatoren	20
Logische Operatoren	20
Funktionen	20
Numerische Funktionen	21
Stringfunktionen	22
Druckerfunktionen	23
BASIC Anweisungen und Befehle	23
Abschnitt 4: Referenztabellen	45
Zeichensatz	45
Fehlerbedingungen	49
Auftreten eines Fehlers	49
Fehlermeldungen	50
Systemspeicheranforderungen	59
Anzeige-Escapecodes	60
System-Voreinstellungen	61
Abkürzungen	62

Einführung

Dieses Referenzhandbuch soll Ihnen als wichtigste Bezugsquelle dienen, nachdem Sie mit der Benutzung und Programmierung Ihres HP-75 vertraut sind. Es ist so handlich, daß es Ihren Computer überall hin begleiten kann, und so umfassend, daß Sie darin praktisch alle für den Routinegebrauch benötigten Informationen finden werden.

Das Referenzhandbuch ist in vier Abschnitte zu verschiedenen Themenkreisen gegliedert:

- Abschnitt 1, Betrieb des HP-75, enthält Informationen zum Betrieb des Computers und seiner Peripherie. Die Themen dieses Abschnitts sind in der gleichen Reihenfolge wie in den Teilen I und II des Benutzerhandbuchs angeordnet, und bei jedem Thema werden die wichtigsten Fakten der entsprechenden Operation besprochen. Schlagen Sie in diesem Abschnitt nach, wenn Sie Informationen zum Betrieb des Computers benötigen.
- Abschnitt 2, Programmierkonzepte, faßt das Schreiben und Ausführen von Programmen zusammen. Die Themen dieses Abschnitts erscheinen in der gleichen Reihenfolge wie in Teil III des Benutzerhandbuchs. Die zu den thematisch geordneten Programmierkonzepten gehörenden Instruktionen werden aufgelistet und kurz definiert. Schlagen Sie in diesem Abschnitt nach, wenn Sie allgemeine Information zu einem Programmierkonzept benötigen.
- Abschnitt 3, Instruktionssatz des HP-75, ist der Kern des Referenzhandbuches und enthält ein komplettes Nachschlagewerk für alle Operatoren, Funktionen, Anweisungen und Befehle des HP-75 Instruktionssatzes. Schlagen Sie in diesem Abschnitt nach, wenn Sie detaillierte Information über eine bestimmte Instruktion des HP-75 benötigen.
- In Abschnitt 4, Referenztabellen, finden Sie ausführliche Erläuterungen zu Zeichensatz, Fehlerbedingungen, Systemspeicherbedarf, Anzeige-Escapecodes, Voreinstellungen und Ersatzwerten, und Abkürzungen von Schlüsselwörtern.

Durchgehend in diesem Handbuch finden Sie fettgedruckte Seitenzahlen in Klammern. Diese Zahlen spezifizieren die Seiten im Benutzerhandbuch, auf denen Sie umfassendere Informationen zu den entsprechenden Operationen, Konzepten oder Instruktionen finden.

Betrieb des HP-75

Tastefeld- und Anzeigesteuerung

Tastenfunktionen

Schreibmaschinentasten

A bis Z	Buchstaben.
0 bis 9	Ziffern.
<input type="text"/>	Leertaste.
.	Punkt. Wird als Dezimalpunkt in Zahlen und als letztes Zeichen in Abkürzungen verwendet.
+ - * / ^	Arithmetische Symbole: Addition, Subtraktion, Multiplikation, Division, Potenzierung, und Ganzzahldivision.
\ (CTL) /	
,	Komma. Zur Trennung von Parametern in Befehlen, Anweisungen und Funktionen benutzt.
()	Klammern. Zur Eingabe von numerischen Ausdrücken benutzt.
!	Ausrufungszeichen. Für Kommentare am Zeilenende in Programmanweisungen und für Terminotizen verwendet.
" '	Doppelte und einfache Anführungszeichen. Zur Kennzeichnung von Filenamen und anderen Literalen.
#	Nummernzeichen. Zur Spezifikation von BASIC Filenummern in den Anweisungen <code>ASSIGN #</code> , <code>PRINT #</code> , <code>RESTORE #</code> und <code>READ #</code> und zur Zuweisung von Timernummern in den Anweisungen <code>ON TIMER #</code> und <code>OFF TIMER #</code> benutzt. Dient ebenfalls als Ungleichheitsoperator in Vergleichsabfragen.
\$	Dollarzeichen. Dient zur Spezifikation von Stringvariablen und Stringfunktionen.
&	Dient zur Verkettung von Stringausdrücken.
[]	Eckige Klammern. Dienen zur Dimensionierung von Stringvariablen und zur Spezifikation von Teilstrings.
@	Dient zur Bildung von Mehrfachanweisungszeilen.
;	Semikolon. Dient zur Trennung von Datenelementen in den Anweisungen <code>PRINT</code> , <code>DISP</code> , <code>INPUT</code> , <code>PRINT #</code> und <code>READ #</code> .
=	Gleichheitszeichen. Dient zur Wertzuweisung auf Variablen und für Vergleichsabfragen.
<	Kleinerzeichen. Wird in Vergleichsabfragen benutzt.
>	Größerzeichen. BASIC-Eingabeaufforderung für Programme und Befehlstermine. Wird auch in Vergleichsabfragen benutzt.
:	Doppelpunkt. Text-Eingabeaufforderung. Dient auch zur Begrenzung von Einheitscodes.
%	Prozentzeichen.

Ⓜ Fragezeichen. Voreingestellte Eingabeaufforderung für die INPUT Anweisung.

Editiertasten

SHIFT	Umschalttaste. Gleichzeitig gedrückte Tasten führen ihre Alternativfunktion aus. Im Großschrift-Modus erzeugen gleichzeitig gedrückte Buchstabentasten Kleinbuchstaben.
CTL	Kontrolltaste. Erzeugt spezielle Anzeigezeichen und eine Vielzahl von Tastenkombinationen.
LOCK	Dient zur Fixierung des Tastenfelds auf Großschreibung und zur Aktivierung des numerischen Tastenfelds.
◀ ▶	Linkspfeil und Rechtspfeil. Bewegen den Cursor in der Anzeige, und bewirken gegebenenfalls ein Verschieben der Information in der Anzeige.
↑ ↓	Aufpfeil und Abpfeil. Bewegen den Cursor in BASIC-, Text- und Terminfiles, sowie in System- und Massenspeicherkatalogen aufwärts und abwärts.
I/R	Einfügungs-/Ersetzungstaste. Tauscht den Ersetzungscursor und den Einfügungscursor untereinander aus.
CLR	Löschtaste. Löscht die Anzeige.
BACK	Rückschritt-Taste. Setzt den Cursor um eine Stelle zurück und löscht das dort stehende Zeichen.
DEL	Zeichenlöschertaste. Löscht ein Zeichen und verschiebt die nachfolgenden Zeichen in der Anzeige nach links.
TAB	Tabulator. Bewegt den Cursor über die Anzeigefelder in TIME- und APPT-Modus.

Systemtasten

Die Systemtasten und die Tasten ↑ und ↓ senden einen Wagenrücklauf/Zeilenvorschub an die Anzeige und an die DISPLAY IS Einheiten, bevor die eigentliche Funktion ausgeführt wird.

ATTN	Attention. Schaltet den HP-75 ein; unterbricht Programme, Listings, die automatische Zeilennummerierung, Kartenleseroperationen und HP-IL Operationen. Bestätigt fällige Termine.
TIME	Time. Schaltet den HP-75 in den TIME-Modus.
APPT	Appointment. Schaltet den HP-75 in den APPT-Modus. Nicht bestätigte Termine werden angezeigt.
EDIT	Edit. Schaltet den HP-75 in den EDIT-Modus.
FET	Fetch. Dient im EDIT-Modus als Eingabehilfe für den Befehl FETCH.
RTN	Return. Der Ausdruck, die Anweisung oder der Befehl in der Anzeige wird ausgewertet, gespeichert oder ausgeführt.
RUN	Run. Startet im EDIT-Modus das momentane BASIC-Programm. Bearbeitet im APPT-Modus fällige Termine.

Tastenkombinationen

System-Tastenkombinationen

- SHIFT** **CTL** **CLR** Bewirkt ein Zurücksetzen des Systems.*
- SHIFT** **ATTN** Bewirkt ein Ausschalten der Anzeige und die Desaktivierung aller Tasten außer **ATTN**.
- SHIFT** **LOCK** Bewirkt ein Umschalten auf Großbuchstaben. Nochmaliges Drücken von **LOCK** setzt den Computer wieder auf Kleinschreibung zurück.
- CTL** **LOCK** Aktiviert das numerische Tastenfeld. Nochmaliges Drücken von **LOCK** stellt das normale Tastenfeld wieder her.
- SHIFT** **FET** Zeigt die letzte **ERROR**, **WARNING**, Kartenleser-, Programmverfolgungs- oder HP-IL Meldung nochmals an, solange die Taste **FET** gedrückt bleibt. Die Meldung kann durch Drücken von **CLR**, **ATTN**, **TIME**, **APPT**, **EDIT**, **FET**, **↑**, **↓**, **RTN** oder **RUN** gelöscht werden.
- SHIFT** **RUN** Führt ein Programm, beginnend bei der momentanen Zeile, in Einzelschritten aus.

Editier-Tastenkombinationen

- SHIFT** **↑** Stellt den Filepointer auf die erste Zeile des momentanen Files, auf den ersten Eintrag eines **CAT ALL** oder Massenspeicher-Verzeichnisses, oder auf den ersten Termin im **APPT**-Modus.
- SHIFT** **↓** Stellt den Filepointer auf die letzte Zeile des momentanen Files, auf den letzten Eintrag eines **CAT ALL** oder Massenspeicher-Verzeichnisses oder auf den letzten Termin im **APPT**-Modus.
- CTL** **FET** Bringt im **EDIT**-Modus die letzte Eingabe zur Anzeige und macht sie editierbereit. Die Eingabe wird intakt angezeigt, wenn nach dem letzten **RTN** keine weiteren Tasten gedrückt worden sind. Bei den folgenden Befehlen wird der Eingabebuffer benötigt, und die letzte Eingabe ist somit verloren: **FETCH**, **FETCH KEY**, **LIST**, **LOCK**, **PLIST** und **TRANSFORM**.
- SHIFT** **DEL** Löscht im **EDIT**- und im **TIME**-Modus die Zeile von der momentanen Cursorposition bis zum Zeilenende. Löscht im **APPT**-Modus den derzeit angezeigten Termin, oder ersetzt ihn durch einen editierten Termin.
- SHIFT** **TAB** Verschiebt den Cursor nach links über die Anzeigefelder; Umkehrwirkung zu **TAB**.
- SHIFT** **←** Setzt den Cursor an den Anfang der Anzeigzeile.
- SHIFT** **→** Setzt den Cursor ans Ende der Anzeigzeile.
- CTL** **←** Verschiebt den Cursor um 32 Positionen nach links, jedoch nicht über den Anfang der Anzeigzeile hinaus.
- CTL** **→** Verschiebt den Cursor um 32 Positionen nach rechts, jedoch nicht über das Ende der Anzeigzeile hinaus.

Tastenkombination für Anzeigeeichen

- SHIFT** **I/R** Das der nächsten Taste oder Tastenkombination zugeordnete Zeichen wird angezeigt, unabhängig von der aktuellen Tastenbelegung.

* **SHIFT** **CTL** **M**, **I**, **J**, **8** und **I/R** bewirken ebenfalls ein Zurücksetzen des Systems, sobald diese Tastenkombinationen für eine Sekunde oder länger gedrückt werden.

Escape-Tastenkombination

CTL BACK	Erzeugt den Dezimalcode 27 (ESC) als Anfangszeichen von Escapecodefolgen. Der HP-75 reagiert auf 12 Escapecodes. (Siehe Tabelle der Anzeige-Escapecodes in Abschnitt 4, Referenztabellen.)
------------------------	---

Kontrolle der Anzeige

CTL FET	Holt den Inhalt des Eingabepuffers zur Editierung in die Anzeige.
DELAY	Steuert das Anzeigeintervall für Meldungen und Ausgaben.
MARGIN	Bestimmt die Zeichenposition, an der die End-of-Line Sequenz gesendet wird.
PWIDTH	Setzt die Zeilenlänge für Druckerausgaben.
SHIFT I/R	Bewirkt die Anzeige des der als nächstes gedrückten Taste oder Tastenkombination zugeordneten Zeichens.
WIDTH	Setzt die Zeilenlänge für Ausgaben auf die Anzeige.

Funktionen zur Manipulation von Zeichen

CHR\$	Gibt das zu dem spezifizierten Dezimalcode gehörende Zeichen zurück.
NUM	Gibt den Dezimalcode des spezifizierten Zeichens zurück.

Syntax-Richtlinien

In diesem Handbuch werden durchgehend die folgenden Konventionen zur Beschreibung der Syntax verwendet:

Punktmatrix	In Punktmatrix gesetzte Worte (wie LIST) können in Groß- und in Kleinbuchstaben eingegeben werden.
<i>kursiv</i>	Kursiv gesetzte Elemente sind die von Ihnen vorzugebenden Parameter.
' ', " "	Filenamen und andere Zeichenstrings können in einfache oder doppelte Anführungszeichen eingeschlossen und in Groß- und in Kleinbuchstaben eingegeben werden.
[]	Optionale Parameter sind in eckige Klammern gesetzt.
...	Drei Punkte deuten als Auslassungs- oder Wiederholungszeichen an, daß das optionale Element in einer Klammer wiederholt werden kann.
<i>übereinander gesetzt</i>	Wenn zwei oder mehr Elemente übereinander gesetzt sind, kann genau eines dieser Elemente spezifiziert werden.
oder	Sind zwei oder mehr Ausdrücke durch «oder» getrennt, können beide einmal oder mehrmals auftreten.

Filemanipulationen

Filetypen

B	BASIC-File.
PE	Privater BASIC-File.
T	Textfile.
A	Terminfile.
L	LEX-File.
I	Austauschfile oder LIF1 File.
?	Unbekannter File.

Filespezifikatoren

Filespezifikatoren benennen und beschreiben von Files.

Definitionen:

Filename Ein Stringausdruck, der aus maximal acht Zeichen besteht. Das erste Zeichen muß ein Buchstabe oder ein Punkt sein; die folgenden Zeichen können Buchstaben oder Ziffern sein.

Einheitscode Ein Stringausdruck, der aus einem oder zwei Buchstaben, einem Buchstaben und einer Zahl oder einer Zahl und einem Buchstaben bestehen kann.

Paßwort Ein aus bis zu vier Buchstaben oder Ziffern bestehender Stringausdruck.

Syntax der Filespezifikatoren	
Filetyp	Spezifikatoren
Files im Speicher	' <i>Filename</i> '
	APPT
	KEYS
Kartenfile-Spezifikator	CARD
	' ;CARD '
	' ;PCRD '
	' <i>Filename</i> : CARD '
	' <i>Filename</i> : PCRD '
	' <i>Filename</i> : CARD / <i>Paßwort</i> '
	' <i>Filename</i> : PCRD / <i>Paßwort</i> '
Massenspeicherspezifikator	' <i>Filename</i> : <i>Einheitscode</i> '
	' <i>Filename</i> : <i>Einheitscode</i> / <i>Paßwort</i> '

Fileparameter

Inkrementwerte und Filenummern müssen auf ganze Zahlen von 1 bis 9999 gerundet werden können.

Zeilennummern müssen vorzeichenlose ganze Zahlen von 1 bis 9999 sein.

Filespezifikatoren können auch über Stringvariablen oder Stringausdrücke spezifiziert werden.

Befehle zur Fileorganisation

Nur für den momentan File	Für beliebige BASIC Files im Speicher	Für beliebige Text-Files im Speicher
AUTO DELETE FETCH MERGE NAME RENUMBER	CALL CAT COPY EDIT LIST PLIST PURGE RENAME RUN TRANSFORM	CAT COPY EDIT LIST PLIST PURGE RENAME TRANSFORM

Bei Ausführung von COPY, DELETE, MERGE, NAME, PURGE, RENAME oder TRANSFORM auf einem initialisierten BASIC-File wird das Programm deallokatziert.

Operatoren und Funktionen

Rangfolge von Operatoren (89)

In der folgenden Tabelle sind die HP-75 Operatoren in ihrer Rangfolge aufgelistet. Ausdrücke mit Operatoren der gleichen Rangordnung werden von links nach rechts ausgewertet.

Zuerst ausgeführt

◊
 Funktionen
 ^
 NOT
 *, /, DIV oder \ (CTL \overline{I})
 +, -
 =, >, >=, <, <=, <> oder #
 AND
 OR, EXOR

Verschachtelte Klammern werden von innen nach außen ausgewertet.

Zuletzt ausgeführt

Numerische Genauigkeit (73)

Typ	Genauigkeit	Maximalwerte
REAL	12 Stellen	$\pm 9.999999999999E \pm 499$
SHORT	5 Stellen	$\pm 9.9999E \pm 99$
INTEGER	5 Stellen	± 99999

Allgemeine mathematische Funktionen (83)

SQR(x)	MAX(x, y)	PI
MOD(x, y)	MIN(x, y)	INF
SGN(x)	RMD(x, y)	EPS

Logarithmische Funktionen (84)

LOG(x)	EXP(x)	LOG10(x)
--------	--------	----------

Trigonometrische Befehle (85)

```
OPTION BASE DEGREES
OPTION ANGLE RADIANS
```

Trigonometrische Funktionen (85)

SIN(x)	TAN(x)	SEC(x)
ASIN(x)	ATAN(x)	CSC(x)
COS(x)	ANGLE(x, y)	RAD(x)
ACOS(x)	COT(x)	DEG(x)

Numerische Ausdrücke (87)

Ein numerischer Ausdruck kann die folgenden Formen besitzen:

- Eine numerische Konstante.
- Eine numerische Variable.
- Eine numerische Funktion.
- Eine beliebige Kombination der obenstehenden Formen, verknüpft durch Operatoren oder Klammern.

TIME-Modus Operationen (92)

SET	Stellen der Uhr.
ADJUST	Nachstellen der Uhr.
EXACT	Eichen der Uhr.
RESET	Löschen des Anpassungsfaktors der Ganggeschwindigkeit.
STATS	Spezifikation des Formats auf DMY bzw. MDY, AM/PM bzw. 24-Stunden Format, YEAR bzw. EXTD Terminkalender und Genauigkeit der Uhr.

APPT-Modus Operationen (100)

Alarmarten:	0 Kein Tonsignal.
	1 Ein kurzes Zirpen.
	2 Ein langer, tiefer Ton.
	3 Ein dreimal wiederholtes Zweitonsignal.
	4 Eine Folge von hohen, eindringlichen Tönen.
	5 Ein langer tiefer Ton, gefolgt von einem langen hohen Ton.
	6 Eine Folge von acht Sirenenklängen.
	7 Ein alle 15 Sekunden wiederholter Alarm des Typs 2.
	8 Ein alle 15 Sekunden wiederholter Alarm des Typs 4.
	9 Ein alle 15 Sekunden wiederholter Alarm des Typs 6.

Terminarten: N (normal)
 A (wiederholend nach Bestätigung)
 R (wiederholend)

! oder >	Spezifiziert Notiz- oder Befehlsfeld.
ATTN	Bestätigt fällige Termine.
SHIFT DEL	Löscht Termine.
SHIFT APPT	Zeigt zusätzliche Termininformation an.
SHIFT ATTN	Schaltet den HP-75 aus; fällige Termine werden bearbeitet.
COPY appt TO ' <i>Filespezifikator</i> '	Kopiert den appt File vom Speicher auf eine Magnetkarte oder auf ein Massenspeichermedium.
COPY ' <i>Filespezifikator</i> ' TO appt	Mischt einen Terminfile von Magnetkarte oder Massenspeicher mit dem appt File im Speicher.

Kartenleser-Operationen (114)

CAT CARD	Zeigt den Katalogeintrag eines Kartenfiles an.
COPY [' <i>Filename</i> '] TO ' <i>Kartenfilespezifikator</i> '	Kopiert einen File vom Speicher auf Magnetkarte.
COPY ' <i>Kartenfilespezifikator</i> ' TO ' <i>Filename</i> '	Kopiert einen File von Magnetkarte in den Speicher.
PROTECT	Schützt einen Magnetkartenfile gegen Überschreiben.
UNPROTECT	Entfernt den Schreibschutz von einem Magnetkartenfile.

HP-IL Operationen (124)

ASSIGN TO	Weist die Einheiten in der Schleife zu.
DISPLAY IS ' : <i>Einheitscode</i> '	Deklariert Anzeigeeinheiten.
PRINTER IS ' : <i>Einheitscode</i> '	Deklariert Druckereinheiten.
CLEAR LOOP	Setzt die Einheiten in der Schleife zurück.
OFF IO	Schaltet die Schleifenkommunikation aus.
RESTORE IO	Schaltet die Schleifenkommunikation ein.
INITIALIZE ' : <i>Einheitscode</i> '	Initialisiert ein Massenspeichermedium.

CAT ' : <i>Einheitscode</i> '	Zeigt den Katalog eines Massenspeichermediums an.
COPY [' <i>Filename</i> '] TO ' <i>Filespezifikator</i> '	Kopiert einen File vom Speicher auf Massenspeicher.
COPY ' <i>Filespezifikator</i> ' TO ' <i>Filename</i> '	Kopiert einen File vom Massenspeicher in den Speicher.
RENAME ' <i>Filespezifikator</i> ' TO ' <i>Filename</i> '	Benennt einen File im Massenspeicher um.
PURGE ' <i>Filespezifikator</i> '	Löscht einen File im Massenspeicher.
PACK ' <i>Einheitscode</i> '	Packt ein Massenspeichermedium.

Tastenbelegungen (142)

FETCH KEY ' <i>Taste</i> '	Ruft die Belegung der spezifizierten Taste oder Tastenkombination ab.
DEF KEY ' <i>Taste</i> ', ' <i>Tastenbelegung</i> ' [;]	Definiert die Taste oder Tastenkombination.
SHIFT I/R	Zeigt das Anzeigezeichen der Taste oder Tastenkombination an.
EDIT KEYS	Editiert den <code>keys</code> File.
RENAME KEYS TO ' <i>Filename</i> '	Benennt den <code>keys</code> File um und deaktiviert die momentanen Tastenbelegungen.
RENAME ' <i>Filename</i> ' TO KEYS	Benennt einen Tastenfile in <code>keys</code> um und aktiviert dessen Tastenbelegungen.
COPY KEYS TO ' <i>Filespezifikator</i> '	Kopiert den <code>keys</code> File auf Magnetkarte oder Massenspeicher.
COPY ' <i>Filespezifikator</i> ' TO KEYS	Kopiert einen Tastenfile von Magnetkarte oder Massenspeicher in den Speicher und aktiviert dessen Tastendefinitionen.

Programmierkonzepte

Ausführen von Programmen (158)

RUN, RUN	Startet die Ausführung eines Programms.
ATTN	Unterbricht die Ausführung eines Programms.
CONT	Setzt die Programmausführung nach ATTN oder STOP wieder fort.

Editieren von Programmen (159)

LIST, PLIST	Listet einen Programmfile auf der Anzeige oder einem Drucker.
FETCH	Bringt einzelne Zeilen eines Programmfiles zur Anzeige.

Grundlegende Anweisungen (165)

LET	Weist Variablen Werte zu.
END	Beendet die Programmausführung und deallokiert das Programm.
STOP	Unterbricht die Programmausführung ohne Deallokation des Programms.
WAIT	Unterbricht die Programmausführung für eine bestimmte Zeitspanne.
REM, !	Kennzeichnet Programmkommentare.
DISP, PRINT, TAB	Zeigt Information an bzw. druckt Information.
INPUT	Erlaubt die Eingabe von Daten über das Tastenfeld.

Verzweigungen, Schleifen und Unterprogramme (176)

GOTO	Unbedingte Verzweigung zu einer Zeilennummer.
IF...THEN...ELSE	Bedingte Verzweigung.
FOR...NEXT, STEP	Bewirkt die wiederholte Ausführung einer Folge von Anweisungen für eine gegebene Anzahl von Malen.
GOSUB...RETURN	Verzweigt zu einer Folge von Anweisungen und kehrt danach zurück.
ON...GOTO	Verzweigt in Abhängigkeit vom Wert des spezifizierten Ausdrucks zu einer Zeilennummer.
ON...GOSUB	Verzweigt in Abhängigkeit vom Wert des spezifizierten Ausdrucks zu einem Unterprogramm.
POP	Hebt eine anstehende Rücksprungbedingung auf.

Programmtimer (186)

ON TIMER #	Aktiviert einen Programmtimer.
OFF TIMER #	Desaktiviert einen Programmtimer.
ON TIMER # GOTO	Verzweigt zu einer Anweisung, sobald das Timerintervall abgelaufen ist.
ON TIMER # GOSUB	Verzweigt zu einem Unterprogramm, sobald das Timerintervall abgelaufen ist.

Felder (192)

OPTION BASE	Definiert die Untergrenze(n) für alle in einem Programm deklarierten Felder.
DIM, REAL, SHORT, INTEGER	Deklariert und dimensioniert numerische Felder.

Strings

Stringvariablen (196)

Stringidentifikator: *Buchstabe* [*Ziffer*]

Dimensionierung eines Strings: DIM *Stringidentifikator* [*Index*]

Voreingestellte Länge einer Stringvariablen: 32 Zeichen

Teilstringidentifikator: *Buchstabe* [*Ziffer*] [*Index* [, *Index*]]

Zwei durch Kommata getrennte Indices spezifizieren die Position des ersten bzw. des letzten Zeichens eines Teilstrings. Ein einzelner Index spezifiziert die Position des ersten Zeichens – der Teilstring besteht dann aus dem Rest des Strings ab dieser Position.

Stringausdrücke (196)

Jede der folgenden Formen stellt einen Stringausdruck dar:

- Stringkonstanten
- Stringvariablen
- Teilstrings
- Stringfunktionen
- Jede beliebige Verkettung der obigen Elemente mit dem Operator &.

Stringfunktionen (198)

Die Parameter *s* und *t* können beliebige Stringausdrücke sein.

LEN(<i>s</i>)	UPRC(<i>s</i>)
POS(<i>s</i> , <i>t</i>)	KEY
VAL(<i>s</i>)	CAT(<i>Filenummer</i>)
STR(<i>numerischer Ausdruck</i>)	

Benutzerdefinierte Funktionen (205)

DEF FN	Definiert eine einzeilige benutzerdefinierte Funktion oder die erste Zeile einer mehrzeiligen benutzerdefinierten Funktion.
LET FN	Weist in einer mehrzeiligen benutzerdefinierten Funktion den Funktionswert zu.
END DEF	Markiert das Ende einer mehrzeiligen benutzerdefinierten Funktion.

Speichern und Zurücklesen von Daten

Daten innerhalb von Programmen (210)

DATA	Enthält numerische oder Stringkonstanten, die über READ Anweisungen gelesen werden können.
READ	Weist Variablen die in DATA Anweisungen abgelegten Werte zu.
RESTORE	Setzt den Datenpointer auf die erste oder spezifizierte DATA Anweisung im File.

Datenfiles (216)

ASSIGN #	Ordnet dem spezifizierten File eine Filenummer zu.
PRINT #	Schreibt Datenelemente in einen Datenfile.
READ #	Liest Datenelemente aus einem Datenfile.
RESTORE #	Setzt den Datenpointer auf die erste oder spezifizierte Zeile eines Datenfiles.

Programmaufrufe (230)

CALL	Ruft innerhalb eines Programms ein anderes Programm auf. Nach der Abarbeitung des aufgerufenen Programms kehrt die Programmausführung zu der auf die CALL Anweisung folgende Zeile des aufrufenden Programms zurück.
------	--

Lokale und globale Deklarationen (233)

Eine *globale* Deklaration ist eine Systemeinstellung, die solange wirksam bleibt

- bis die Einstellung durch eine Deklaration (die entweder innerhalb eines ablaufenden Programms oder über das Tastenfeld ausgeführt werden kann) geändert wird,
- oder der HP-75 zurückgesetzt wird.

Lokale Deklarationen wirken nur auf Berechnungen über das Tastenfeld oder auf das Programm, in dem die Deklaration auftritt.

Lokal	Global
DATA DEF FN, LET FN, END DEF DIM INTEGER, SHORT, REAL IMAGE LET OFF ERROR, ON ERROR ON TIMER # OPTION BASE	ALARM OFF, ALARM ON ASSIGN IO, OFF IO, RESTORE IO ASSIGN # BEEP OFF, BEEP ON DEFAULT OFF, DEFAULT ON DEF KEY DELAY DISPLAY IS, PRINTER IS ENDLINE LOCK MARGIN OFF TIMER # OPTION ANGLE DEGREES OPTION ANGLE RADIANS STANDBY ON, STANDBY OFF TRACE FLOW, TRACE VARS, TRACE OFF WIDTH, PWIDTH

Ausgabeformatierung (238)

IMAGE	Spezifiziert das Ausgabeformat für DISP USING und PRINT USING Anweisungen.
DISP USING	Zeigt Information entsprechend dem in der Anweisung oder einer IMAGE Anweisung enthaltenen Formatstring an.
PRINT USING	Druckt Information entsprechend dem in der Anweisung oder einer IMAGE Anweisung enthaltenen Formatstring aus.

Verfolgungsoperationen (252)

TRACE FLOW	Verfolgt den Programmfluß durch Anzeige aller Verzweigungen während der Programmausführung.
TRACE VARS	Verfolgt Programmvariablen durch Anzeige aller Wertänderungen der Variablen während der Programmausführung.
TRACE OFF	Desaktiviert alle Verfolgungsoperationen.
ON ERROR	Ermöglicht benutzerdefinierte Fehlerbehandlung.
OFF ERROR	Desaktiviert benutzerdefinierte Fehlerbehandlung.
ERRN	Zeigt die Fehlernummer des letzten Fehlers an.
ERRL	Zeigt die Zeilennummer des letzten Fehlers an.

Instruktionssatz des HP-75

Operatoren

Arithmetische Operatoren (69)

\oplus	Gleitkomma-Addition.
\ominus	Gleitkomma-Subtraktion.
\otimes	Gleitkomma-Multiplikation.
\oslash	Gleitkomma-Division.
\lfloor oder DIV	Ganzzahlige Division (ohne Rest).
\wedge	Potenzierung.

Vergleichsoperatoren (82)

\equiv	Gleich.
\lt	Kleiner als.
$\lt \equiv$	Kleiner oder gleich.
\gt	Größer als.
$\gt \equiv$	Größer oder gleich.
$\lt \gt$ oder \neq	Ungleich.

Logische Operatoren (88)

Logische Operatoren wirken auf numerischen Ausdrücken und geben Boole'sche Werte zurück. Wenn die Bewertung eines Ausdrucks 0 ergibt, wird er als falsch betrachtet; wenn sie einen von Null verschiedenen Wert ergibt, wird er als richtig betrachtet. Für alle numerischen Ausdrücke A und B gilt:

$A \text{ AND } B$	ist genau dann wahr, wenn sowohl A als auch B wahr sind.
$A \text{ OR } B$	ist wahr, wenn A oder B oder beide wahr sind.
$A \text{ EXOR } B$	ist wahr, wenn entweder A oder B , nicht aber A und B , wahr ist.

A	B	A AND B	A OR B	A EXOR B	NOT A
T	T	T	T	F	F
T	F	F	T	T	F
F	T	F	T	T	T
F	F	F	F	F	T

Funktionen

In den folgenden Tabellen spezifizieren X und Y zwei beliebige numerische Ausdrücke, $S\#$ und $T\#$ zwei beliebige Stringausdrücke.

Wenn für den Funktionsnamen eine Abkürzung möglich ist, wird sie in Klammern nach der Funktionsbeschreibung angegeben.

Die fettgedruckte Zahl am Ende der Funktionsbeschreibung verweist auf die Seite der Hauptreferenz im Benutzerhandbuch.

Numerische Funktionen

Bei Spezifikationen der geeigneten Argumente (in Typ und Anzahl) gibt jede numerische Funktion eine einzelne numerische Konstante zurück.

ABS(X)	Absolutbetrag von X. (82)
ACOS(X)	Arcus Cosinus von X im ersten oder zweiten Quadranten. (Abkürzung: <i>ac</i>). (86)
ANGLE(X,Y)	Arcus Tangens von Y/X im richtigen Quadranten. Rückgabe des Winkels θ zwischen der x-Achse und dem Punkt (x,y), so daß $-\pi < \theta \leq \pi$ (Abkürzung: <i>an</i>). (86)
ASIN(X)	Arcus Sinus von X, im 1. oder 4. Quadranten. (86)
ATN(X)	Arcus Tangens von X, im 1. oder 4. Quadranten. (86)
CEIL(X)	Kleinste ganze Zahl $\geq X$ (Abkürzung: <i>ce</i>). (82)
COS(X)	Cosinus von X. (86)
COT(X)	Cotangens von X. (86)
CSC(X)	Cosecans von X. (86)
DATE	Datum im Format <i>JJTTT</i> , basierend auf der momentanen Systemzeit. (98)
DEG(X)	Umwandlung des Werts X von Radiant in Altgrad. (86)
EPS	Kleinste Maschinenzahl ($1.E-499$). (83)
ERRL	Zeilennummer des letzten Fehlers oder der letzten Warnung. (260)
ERRN	Nummer des letzten Fehlers oder der letzten Warnung. (Abkürzung: <i>er</i>). (260)
EXP(X)	e^X . (84)
FLOOR(X)	Wie INT(X). (Abkürzung: <i>fl</i>). (82)
FP(X)	Gebrochener Anteil von X. (82)
INF	Größte Maschinenzahl ($9.999999999999999E499$). (83)
INT(X)	Größte ganze Zahl $\leq X$. (82)
IP(X)	Ganzzahliger Anteil von X. (82)
LEN(S#)	Länge des Strings S#. (198)
LOG(X)	Natürlicher Logarithmus von X, $X > 0$. (84)
LOG10(X)	Zehnerlogarithmus von X, $X > 0$ (Abkürzung: <i>lo</i>). (84)
MAX(X,Y)	Wenn $X > Y$, dann X, sonst Y. (83)
MEM	Anzahl noch verfügbarer Bytes im Speicher. (47)
MIN(X,Y)	Wenn $X < Y$, dann X, sonst Y. (83)

MOD(X,Y)	X modulo Y: $X - Y * \text{INT}(X/Y)$. (83)
NUM(S#)	Dezimalcode des ersten Zeichens von S#. (41)
PI	3.14159265359. (83)
POS(S#,T#)	Sucht nach dem ersten Auftreten von T# in S#. Gibt die Anfangsposition oder, falls nicht gefunden, 0 zurück. (198)
RAD(X)	Umwandlung des Werts X von Altgrad in Radiant. (86)
RES	Letztes angezeigtes oder ausgedrucktes numerisches Ergebnis. (71)
RMD(X,Y)	Rest von X/Y: $X - Y * \text{IP}(X/Y)$. (83)
RND	Die nächste Zahl R in einer Folge von Pseudo-Zufallszahlen; $0 \leq R < 1$. (83)
SEC(X)	Sekans von X. (86)
SGN(X)	Vorzeichen von X: -1, falls $X < 0$; 0, falls $X = 0$; 1, falls $X > 0$. (83)
SIN(X)	Sinus von X. (86)
SQR(X)	Positive Quadratwurzel von X. (83)
TAN(X)	Tangens von X. (86)
TIME	Anzahl Sekunden seit Mitternacht. (98)
VAL(S#)	Numerischer Wert eines Strings aus Ziffern, Dezimalpunkt und/oder Exponent. (198)

Stringfunktionen

Stringfunktionen geben null oder mehr Zeichen an Information zurück.

CAT#(X)	Katalogeintrag des spezifizierten Files mit 32 Zeichen Länge. Files werden in der Reihenfolge ihres Eintrags im Systemkatalog numeriert. CAT(0) gibt den Katalogeintrag des momentanen EDIT-Files zurück. Bei $X < 0$ gibt CAT#(X) den Katalogeintrag des derzeit initialisierten BASIC-Files zurück, falls vorhanden (Abkürzung: c.). (198)
CHR#(X)	Zeichen mit dem Dezimalcode MOD(X,256) (Abkürzung: ch.). (41)
DATE#	Datum im Format JJ/MM/TT. (98)
KEY#	Anzeigezeichen der derzeit gedrückten Taste oder Tastenkombination. Falls keine Taste gedrückt ist, wird ein Nullstring zurückgegeben. (Abkürzung: k.). (198)
STR#(X)	Die in den Ziffern, dem Dezimalpunkt, Vorzeichen und Exponenten von X enthaltene Stringinformation. (198)
TIME#	Zeit im Format hh:mm:ss in 24-Stunden-Notation. (Abkürzung: ti.). (98)
UPRC#(S#)	Wandelt die Kleinbuchstaben in S# in Großbuchstaben um. (Abkürzung: upr.). (198)
VER#	Sechsstelliger String mit der aktuellen Betriebssystemversion. (267)

Tabulierungsfunktion

TAB(X) Die nachfolgende DISP oder PRINT Ausgabe wird von Spalte X an dargestellt, wobei Spalte 1 der linke Rand ist. (167)

BASIC-Anweisungen und Befehle

In der folgenden Liste wird hinter jedem Schlüsselwort in Klammern die kürzestmögliche Abkürzung angegeben. Das Wort «keine» hinter dem Schlüsselwort bedeutet, daß das Schlüsselwort keine Abkürzung besitzt.

Die fett gedruckte Zahl in Klammern ganz rechts in der Zeile des Schlüsselworts gibt die Seitenzahl des wichtigsten Bezugs auf dieses Schlüsselwort im Benutzerhandbuch. (Die gleiche Seitenreferenz wird auch im Instruktionssatz-Index im Rückumschlag des Benutzerhandbuches gegeben.)

ALARM (a1.) (106)

```
ALARM OFF
```

Der HP-75 ignoriert nach diesem Befehl fällig werdende Termine.

- Globale Deklaration.

```
ALARM ON
```

Stellt die normale Bearbeitung von fälligwerdenden Terminen wieder her. Alle Termine, die nach dem letzten ALARM OFF fällig geworden sind, werden sofort gemeldet.

- Globale Deklaration.

ASSIGN # (as.#) (216)

```
ASSIGN # Filenummer TO 'Filename' [ ,BASIC ]
                                     [ ,TEXT ]
```

```
100 ASSIGN # 14 TO 'PFEIL'
110 ASSIGN # 08 TO 'RATE',TEXT
```

Weist dem angegebenen File die spezifizierte Filenummer zu und stellt den entsprechenden Datenpointer auf die erste Zeile des Files. Wenn der angegebene File nicht existiert, wird ein neuer File erzeugt. Falls der Filetyp nicht spezifiziert ist, wird ein BASIC-File unterstellt.

- Globale Deklaration.

(219)

```
ASSIGN # Filenummer TO *
                                     ''
                                     '*'
```

```
120 ASSIGN # 14 TO *
130 ASSIGN # 08 TO ''
```

Hebt die Zuordnung zwischen dem spezifizierten Filepointer und dem File auf. Der durch die Zuordnung belegte Speicherplatz wird wieder freigegeben.

- Globale Deklaration.

ASSIGN IO (as . .)

(126)

```
ASSIGN IO [' :Einheitscode [, :Einheitscode...]' ]
```

```
ASSIGN IO
140 ASSIGN IO ':TV, :CA'
```

Initiiert eine interaktive Zuweisung von Einheitscodes an Peripherieeinheiten oder führt die Zuweisung entsprechend der optionalen Einheitsliste durch. Die Einheiten werden in der Reihenfolge ihrer Anordnung in der Schleife zugewiesen.

- Globale Deklaration.

AUTO (a .)

(51)

```
AUTO [Anfangszeilennummer [, Inkrementwert]]
```

```
auto
auto 500
auto 5,5
```

Bewirkt die automatische Zeilennummerierung von Text- oder BASIC-Files; die Numerierung beginnt mit der *Anfangszeilennummer* und wird jeweils um den Inkrementwert erhöht. Kann durch Drücken von **ATTN** aufgehoben werden. Voreinstellung ist die momentane Zeile plus 10 und Inkrementwert 10.

BEEP (be .)

(30)

```
BEEP [Frequenz in Hz [, Dauer in Sekunden]]
```

```
150 BEEP
160 BEEP 400,1,5
170 BEEP W/(2*PI),T*60
```

Erzeugt ein Tonsignal der spezifizierten *Frequenz* und *Dauer*. Voreinstellung ist 1400 Hz und 0,1 Sekunden.

```
BEEP OFF
```

Desaktiviert den Tongenerator, bis ein **BEEP ON** Befehl ausgeführt wird. Die Tonsignale der Alarmtypen 6 und 9 werden auch nach **BEEP OFF** erzeugt.

- Globale Deklaration.

```
BEEP ON
```

Reaktiviert den Tongenerator nach einem **BEEP OFF** Befehl.

- Globale Deklaration.

BYE (b .)

(29)

```
BYE
```

Schaltet den HP-75 aus.

CALL (keine) (230)

```
CALL, 'Filename'
```

```
180 CALL 'TEIL2'
```

Die Programmausführung verzweigt zu dem spezifizierten Programm. Sobald im Verlauf der Ausführung des aufgerufenen Programms ein **END** angetroffen wird, kehrt die Ausführung wieder zum aufrufenden Programm zurück.

CAT ALL (c...) (49)

```
CAT ALL
```

Greift auf den gesamten Systemkatalog zu.

CAT (keine) (49)

```
CAT APPT
```

Zeigt den Katalogeintrag des `appt` Files an.

(117)

```
CAT CARD
```

Zeigt die auf einer Magnetkartenspур enthaltene Kataloginformation an.

(134)

```
CAT ':Einheitscode'
```

```
cat ':C2'
```

Greift auf den Katalog des Mediums in der spezifizierten Massenspeichereinheit zu.

```
CAT ['Filename']
```

```
cat
cat 'namen'
190 CAT F1$
```

Zeigt den Katalogeintrag des momentanen BASIC- oder Textfiles oder des spezifizierten Files im Speicher an.

(134)

```
CAT 'Massenspeicherspezifikator'
```

```
cat 'testdat:c2'
200 CAT T1$ & ':C2'
```

Zeigt den Katalogeintrag des spezifizierten Massenspeicherfiles an.

(144)

```
CAT KEYS
```

Zeigt den Katalogeintrag des `keys` Files an.

CLEAR (`c1.`)

(131)

```
CLEAR ' : Einheitscode [ , : Einheitscode... ] '
```

```
210 CLEAR ' : TP , : TV '
```

Setzt die spezifizierten HP-IL Einheiten in ihren Einschaltungszustand (siehe Benutzerhandbuch des Peripheriegerätes) zurück.

CLEAR LOOP (`c1.1.`)

(131)

```
CLEAR LOOP
```

Setzt alle HP-IL Einheiten in der Schleife in ihren Einschaltzustand zurück.

CLEAR VARS (`c1..`)

(81)

```
CLEAR VARS
```

Löscht die Werte aller Rechner- und Programmvariablen und gibt den von ihnen belegten Speicherplatz wieder frei.

CONT (keine)

(159)

```
CONT [Zeilennummer]
```

```
cont
cont 550
```

Setzt die Ausführung des momentanen Programms bei der nächsten Anweisung oder am Anfang der spezifizierten Zeile fort.

COPY (`c0.`)

(118/135)

```
COPY [ 'Filename' ] TO 'Filename'
      APPT          CARD
      KEYS          'Magnetkartenfilespezifikator'
                  'Massenspeicherspezifikator'
```

```
copy to card
copy 'altfile' to 'neufile'
copy appt to 'app10:ca'
220 COPY 'FILE6' TO ':PCRD'
copy 'files' to 'file5:tp'
230 COPY F1$ TO F2$
```

Kopiert den spezifizierten File im Speicher auf das spezifizierte Zielmedium. Bei fehlender Angabe eines Quellfiles wird der momentane File kopiert.

(119)

```
COPY CARD                TO 'Filename'
      'Magnetkartenfilespezifikator'  APPT
                                      KEYS
```

```
copy card to 'dito'
240 COPY 'DATUM:PCRD' TO 'NEU'
```

Kopiert den spezifizierten Magnetkartenfile auf einen File mit dem spezifizierten Namen im Speicher.

(135)

```
COPY 'Massenspeicherspezifikator' TO APPT
                                      KEYS
                                      'Filename'
                                      'Massenspeicherspezifikator'
```

```
copy 'app10:ca' to appt
copy 'file5:tp' to 'files'
250 COPY 'RECORD5:CA' TO 'BUFFER'
260 COPY F1$ & ':C1' TO F1$ & ':C2'
```

Kopiert den spezifizierten Quellfile vom Massenpeicher auf den spezifizierten Zielfile im Speicher oder Massenspeicher.

DATA (da.)

(210)

```
DATA Zahl oder Text [, Zahl oder Text]
```

```
270 DATA 6, 72.3, DM, 1E44, 'Pfund'
```

Stellt numerische Werte und Strings mit oder ohne Anführungszeichen zur Zuweisung an Variablen in READ Anweisungen zur Verfügung. DATA Anweisungen können an beliebiger Stelle im Programm angeordnet sein. Datenelemente werden von links nach rechts innerhalb einer DATA Anweisung und von der DATA Anweisung mit der niedrigsten Zeilennummer zur DATA Anweisung mit der höchsten Zeilennummer innerhalb des Programms gelesen.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR oder ON TIMER.
- Zeile darf keine weiteren Anweisungen enthalten.

DEFAULT (defa.)

(89)

```
DEFAULT OFF
```

Hebt die Vorgabe von Ersatzwerten für unzulässige mathematische Ausdrücke auf. Die Programmausführung wird angehalten und Sie erhalten eine Fehlermeldung, wenn einer der Fehler 1 bis 8 auftritt. (Siehe Fehlerbedingungen, Abschnitt 4.)

- Globale Deklaration.

```
DEFAULT ON
```

Bedingt die Vorgabe von Ersatzwerten für unzulässige mathematische Ausdrücke. Die Fehler 1 bis 8 erzeugen in diesem Modus Warnungen; die Programmausführung wird fortgesetzt.

- Globale Deklaration.

DEF FN (keine)

(205)

```
DEF FN numerische Variable [Parameter [Parameter...]] = numerischer Ausdruck
DEF FN Stringvariable [Parameter [Parameter...]] = Stringausdruck
```

```
280 DEF FNA (A1, C(4,5)) = A1 * TAN(C(4))
290 DEF FNB (X) = SQR(X^2+3*X+1)
300 DEF FNC$ = CHR$(X4)
```

Definiert eine einzeilige benutzerdefinierte Funktion vollständig. Jede beliebige Kombination von einfachen numerischen und Stringvariablen kann als Parameter benutzt werden; diese Parameter sind allerdings nur lokal innerhalb der Funktion verfügbar.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Zeile darf keine weiteren Anweisungen enthalten.

(207)

```
DEF FN numerische Variable [Parameter [Parameter...]]
DEF FN Stringvariable [Parameter [Parameter...]]
```

```
310 DEF FNA (B1, B2)
320 DEF FNC$ (N1, R$)
```

Definiert den Anfang einer mehrzeiligen benutzerdefinierten Funktion. Jede beliebige Kombination von einfachen numerischen Variablen und Stringvariablen kann als Parameter benutzt werden; diese Parameter sind nur lokal in der Funktion definiert. Mehrzeilige Funktionen können bis zu 32 Zeichen lange Stringvariablen zurückgeben.

- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.

(207)

```
[LET] FN numerische Variable = numerischer Ausdruck
[LET] FN Stringvariable = Stringausdruck
```

```
330 LET FNA = M
340 FNC$ = 'Laenge'
```

Weist einer mehrzeiligen benutzerdefinierten Funktion einen Wert zu, der an das Hauptprogramm zurückgegeben wird. In jeder mehrzeiligen Funktion muß mindestens eine LET FN Anweisung enthalten sein.

- Nicht über das Tastenfeld ausführbar.

(207)

```
END DEF
```

Markiert das Ende einer mehrzeiligen benutzerdefinierten Funktion.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Muß die letzte Anweisung einer Zeile sein.

DEF KEY (d...)

(143)

```
DEF KEY 'Tastenanzeigezeichen', 'Tastendefinition' [:]
```

```
def key '~', 'cont';
def key '↓', 'CLEAR LOOP'
```

Definiert die durch das *Tastenanzeigezeichen* spezifizierte Taste oder Tastenkombination entweder als Eingabehilfe (mit :) oder als Sofortausführungstaste (ohne :).

- Globale Deklaration.

DELAY (d...)

(39)

```
DELAY numerischer Ausdruck
```

```
delay 3
350 DELAY 0
```

Spezifiziert das Zeitintervall in Sekunden zwischen der Anzeige aufeinanderfolgender Anzeigezeilen (auf die Zehntelsekunde genau). Der *numerische Ausdruck* kann im Bereich von 0 bis 2^{26} Sekunden liegen.

- Globale Deklaration.

DELETE (dele...)

(59)

```
DELETE [Anfangszeilennummer [, Endzeilennummer]]
```

```
delete 120
360 DELETE 250, 280
```

Löscht die spezifizierte Zeile oder Folge von Zeilen aus dem momentanen BASIC- oder Textfile. DELETE allein löscht die momentane Zeile. Bei fehlender Angabe einer *Endzeilennummer* wird nur die durch die *Anfangszeilennummer* spezifizierte Zeile gelöscht.

DIM (keine)

(194)

```
DIM Element [, Element...]
```

```
370 DIM X(20), Y1(5,10), Z#(80)
```

Deklariert die Maximallänge für Stringvariablen sowie die Obergrenzen für ein- und zweidimensionale numerische Felder. Dem Namen einer Stringvariablen muß deren Länge in eckigen Klammern folgen; einem numerischen Feldnamen müssen die Maximalwerte der Indices in runden Klammern folgen (bei zweidimensionalen Feldern durch ein Komma getrennt).

- Lokale Deklaration.
- Nicht zulässig nach THEN, ELSE, ON ERROR oder ON TIMER.
- Muß die letzte Anweisung einer Zeile sein.
- Muß vor einem Aufruf der dimensionierten Variablen auftreten.


```
edit
edit 'datenfile', text
edit basic
```

Positioniert den Filepointer auf die erste Zeile eines neuen Arbeitsfiles oder des spezifizierten Files. Wenn ein neuer File erzeugt wird, hat er entweder den Filetyp des momentanen Files oder den spezifizierten Filetyp. Der Katalogeintrag des Files wird angezeigt.

EDIT KEYS (e . keys) (151)

```
EDIT KEYS
```

Positioniert den Filepointer auf die erste Zeile des `keys` Files.

END (keine) (165)

```
END
```

Beendet die Programmausführung und deallokiert das Programm.

- Nicht über das Tastenfeld ausführbar.

END DEF (end d.) (207)

```
END DEF
```

Siehe `DEF FN`.

ENDLINE (en.) (140)

```
ENDLINE [Stringausdruck mit 0–3 Zeichen]
```

```
440 ENDLINE CHR$(13)&CHR$(10)&CHR$(10)
450 ENDLINE ''
460 ENDLINE
```

Definiert die End-of-Line Zeichen (bis zu 3), die an `PRINTER IS` Einheiten gesendet werden. `ENDLINE` mit dem Nullstring spezifiziert eine Ausgabe in einem kontinuierlichen String ohne End-of-Line Begrenzer. `ENDLINE` ohne Angabe eines Parameters spezifiziert CR/LF (Wagenrücklauf/Zeilenvorschub) als End-of-Line Sequenz.

- Globale Deklaration.

FETCH (f. /`FET`) (53)

```
FETCH [ Zeilennummer
        Suchstring [, Zeilennummer] ]
```

```
fetch 45
fetch 'int'
fetch 'call', 250
```

Ruft die momentane Zeile, die spezifizierte Zeile, die nächste den Suchstring enthaltende Zeile, oder beginnend bei der spezifizierten Zeile die nächste den Suchstring enthaltende Zeile ab. Der Suchstring kann ein beliebiger Stringausdruck sein.

(146)

```
FETCH KEY 'Stringausdruck mit einem Zeichen'
```

```
fetch key 'M'
fetch key chr$(255)
```

Ruft die momentane Definition der spezifizierten Taste oder Tastenkombination ab und macht sie zur Editierung verfügbar.

FOR (keine)

(179)

```
FOR Schleifenzähler = Anfangswert TO Endwert [STEP Inkrementwert]
```

```
470 FOR I=1 TO 15 @ DISP I; X5 @ NEXT I
480 FOR J=2*K TO 15*K STEP K
```

Definiert den Beginn einer FOR...NEXT Schleife. Der *Schleifenzähler* muß ein einfache numerische Variable sein; der *Anfangswert*, der *Endwert* und der optionale *Inkrementwert* können beliebige numerische Ausdrücke sein. Der optionale STEP Parameter bestimmt den *Inkrementwert* für den *Schleifenzähler* bei der Ausführung der nächsten NEXT Anweisung der Schleife.

- Nicht über das Tastenfeld ausführbar (außer wenn die gesamte FOR ... NEXT Schleife in einer Zeile enthalten ist).
- Nicht zulässig nach THEN, ELSE, ON ERROR oder ON TIMER.

GOSUB (gos.)

(182)

```
GOSUB Zeilennummer
```

```
490 GOSUB 1700
```

Die Programmausführung verzweigt zu dem bei der spezifizierten Zeilennummer beginnenden Unterprogramm.

- Nicht über das Tastenfeld ausführbar.

GOTO (g.)

(176)

```
GOTO
```

```
500 GOTO 2000
```

Die Programmausführung verzweigt zur spezifizierten Zeilennummer.

- Nicht über das Tastenfeld ausführbar.
- Dieser Anweisung in der gleichen Zeile folgende Anweisungen werden niemals ausgeführt.

IF...THEN...ELSE (keine ...th...el.)

(177)

```
IF Ausdruck THEN 

|                                                               |
|---------------------------------------------------------------|
| <i>Zeilennummer</i>                                           |
| [ <i>zulässige Anweisung...</i><br>oder<br><i>Befehl...</i> ] |

 ELSE 

|                                                               |
|---------------------------------------------------------------|
| <i>Zeilennummer</i>                                           |
| [ <i>zulässige Anweisung...</i><br>oder<br><i>Befehl...</i> ] |


```

```
510 IF X4 THEN 2350 ELSE RESTORE @ GOTO 200
520 IF M1#[5,6]='LF' THEN C=10 ELSE C=0
530 IF X<Y THEN M1=X @ M2=Y ELSE M1=Y @ M2=X
```

Dient zur Konstruktion von bedingten Verzweigungen. Wenn die Auswertung des Ausdrucks von Null verschiedene Werte ergibt (wahr), wird der THEN Teil der Anweisung ausgeführt. Die Programmausführung verzweigt zu der spezifizierten Zeile oder wird mit den auf THEN folgenden Instruktionen fortgesetzt. Wenn die Auswertung des Ausdrucks den Wert Null ergibt (falsch), wird die Programmausführung mit der nächsten Programmzeile, oder falls die optionale Anweisung ELSE verwendet wird, mit den auf ELSE folgenden Anweisungen fortgesetzt.

- Nicht zulässig nach THEN, ELSE, ON ERROR oder ON TIMER.
- Muß die letzte Anweisung einer Zeile sein.
- Die folgenden Anweisungen sind nach THEN oder ELSE nicht erlaubt:

```
FOR      END DEF    ON TIMER
NEXT     IF        OPTION BASE
DATA     IMAGE     REAL
DEF FN   INTEGER   SHORT
DIM      ON ERROR
```

IMAGE (im.)

(238)

IMAGE *Formatstring*

```
540 IMAGE 4A, 3X, 10A, 5D.3D
550 IMAGE 2X, 'ERGEBNIS'//
560 IMAGE DDDCDDD.DDD, 2X, zzz.d, 3X, 2(*****.dd)
```

Spezifiziert das Ausgabeformat für DISP USING und PRINT USING Anweisungen. Der *Formatstring* besteht aus einem oder mehreren Feldspezifikatoren, die durch Kommata oder Schrägstriche zu trennen sind.

IMAGE Symbol	Ausgabe	Beschreibung	Wiederholung möglich
x, X	Leerstelle	Spezifiziert eine Leerstelle zwischen Datenelementen.	ja
'', " "	Literal	Zur Begrenzung von Strings in Formatstrings oder ganzen Spezifikatoren in DISP USING und PRINT USING Anweisungen.	nein
a, A	Zeichen	Spezifiziert die Position eines Zeichens, Text steht linksbündig.	ja
d, D	Ziffer	Spezifiziert die Position einer Ziffer links oder rechts des Dezimaltrennsymbols; führende Leerstellen, nachlaufende Nullen.	ja
z, Z	Ziffer	Spezifiziert die Position einer Ziffer links des Dezimaltrennsymbols; führende Nullen.	ja
*	Ziffer	Spezifiziert die Position einer Ziffer links des Dezimaltrennzeichens, führende Schutzsterne.	ja
±, S	Vorzeichen	Spezifiziert ein Vorzeichen, + oder -.	nein
m, M	Vorzeichen	Spezifiziert ein Vorzeichen, Leerstelle oder -.	nein
e, E	Exponentialformat	Gibt Zahlen mit Exponenten E, Vorzeichen und drei Stellen aus.	nein
.	Dezimalpunkt	Gibt einen Dezimalpunkt als Dezimaltrennsymbol an dieser Stelle aus.	nein
r, R	Komma	Gibt ein Komma als Dezimaltrennsymbol an dieser Stelle aus.	nein
c, C	Komma	Gibt ein Komma als Zifferntrennsymbol an dieser Stelle aus.	nein
p, P	Punkt	Gibt einen Punkt als Zifferntrennsymbol an dieser Stelle aus.	nein
()	Feld	Ermöglicht die Wiederholung von eingeschlossenen Feldspezifikatoren.	ja
k, K	Kompaktformat	Strings und Zahlen werden ohne führende Daten oder nachlaufende Leerstellen ausgegeben.	nein
/	CR/LF	Erzeugt einen Wagenrücklauf/Zeilenvorschub; kann auch Datenelemente begrenzen.	ja

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Zeile darf keine weiteren Anweisungen enthalten.

INITIALIZE (ini.) (132)

```
INITIALIZE ' :Einheitscode ' [, Anzahl Fileeinträge]
```

```
570 INITIALIZE ' :CA', 300
initialize ' :t1'
```

Bereitet das Medium in der spezifizierten Massenspeichereinheit auf die Speicherung von Information vor. Die Anzahl der Fileeinträge ist auf 453 begrenzt; bei fehlender Angabe dieses Parameters dient 128 als Voreinstellung.

INPUT (i.) (168)

```
INPUT Variable [, Variable...]
INPUT 'Eingabeaufforderung' ; Variable [, Variable...]
INPUT 'Eingabeaufforderung' ; Eingabeaufforderungsstring ; Variable [, Variable...]
```

```
580 INPUT U3, A$
590 INPUT 'Preis'; P1
600 INPUT 'Ihr Name', N$, N$
```

Ermöglicht während der Ausführung eines Programms die Wertzuweisung auf Variablen über das Tastenfeld. Die optionale zweite Eingabeaufforderung kann überschrieben werden und wird gegebenenfalls als Eingabe gelesen.

INTEGER (int.) (194)

```
INTEGER numerische Variable [<Indices>] [, numerische Variable [<Indices>]...]
```

```
610 INTEGER S, D(3,5), K(50)
```

Dimensioniert numerische Variablen in INTEGER Genauigkeit und reserviert Speicherplatz für sie. Sowohl einfache als auch Feldvariablen können deklariert werden.

- Lokale Deklaration.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Zeile darf keine weiteren Anweisungen enthalten.
- Muß vor jedem Bezug auf die deklarierten Variablen auftreten.

LET (keine) (165)

```
[LET] numerische Variable [, numerische Variable...] = numerischer Ausdruck
[LET] Stringvariable [, Stringvariable...] = Stringausdruck
```

```
620 LET X,Y(8) = Z^2-3*Z+8
630 A$[5,9] = CHR$(K)&' ip$'
```

Weist einer oder mehreren Variablen einen Wert zu.

- Lokale Deklaration.

LET FN (l e . .) (207)

```
[LET] FN numerische Variable = numerischer Ausdruck
[LET] FN Stringvariable = Stringausdruck
```

Siehe DEF FN.

LIST (l .) (56)

```
LIST [ 'Filename' ] [ , Anfangszeilennummer [ , Endzeilennummer] ]
      [ KEYS ]
```

```
640 LIST 200, 400
650 LIST 'REST', 150
```

Listet eine oder mehrere Zeilen des momentanen oder des spezifizierten Files auf der Anzeige und den DISPLAY IS Einheiten aus.

LIST IO (l . .) (127)

```
LIST IO
```

Listet die Einheitscodes der momentan zugewiesenen HP-IL Einheiten auf der Anzeige und den DISPLAY IS Einheiten aus.

LOCK (keine) (28)

```
LOCK 'Paßwort'
```

```
lock 'gandalf'
```

Sichert den HP-75 gegen Benutzung ohne das spezifizierte Paßwort. LOCK '' hebt den Paßwortschutz auf.

- Globale Deklaration.

MARGIN (m a .) (40)

```
MARGIN Anzahl Zeichen
```

```
margin 25
660 MARGIN LEN(B$)-5
```

Setzt die Zeichenposition, bei der ein Tonsignal auf das Ende einer Eingabezeile aufmerksam macht.

- Globale Deklaration.

MERGE (m .) (60)

```
MERGE 'Filename' [ , Anfangszeilennummer [ , Endzeilennummer] ]
```

```
merge 'programm2'
670 MERGE 'TEST', 150, 200
```

Mischt die spezifizierte Folge von Zeilen des spezifizierten Files in den momentanen BASIC- oder Textfile. (Die Filetypen müssen übereinstimmen.) Bei fehlender Angabe einer *Endzeilennummer* wird der File von der *Anfangszeilennummer* bis zur End-of-File Marke eingefügt. Bei fehlender Angabe von *Anfangs-* und *Endzeilennummer* wird der gesamte File eingefügt.

NAME (n.) (64)

```
NAME 'Filename'
```

```
NAME 'NEUNAME'
```

Benennt den momentanen File um und erzeugt einen neuen Arbeitsfile des gleichen Typs (BASIC oder Text).

NEXT (ne.) (179)

```
NEXT Schleifenzähler
```

```
680 NEXT J
```

Definiert das Ende einer FOR...NEXT Schleife. Die Ausführung kehrt zu der zugehörigen FOR Anweisung zurück und der *Schleifenzähler* wird um 1 oder um den STEP Wert inkrementiert.

- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.

OFF ERROR (fo.e.) (259)

```
OFF ERROR
```

Hebt die momentane ON ERROR Deklaration auf.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.

OFF IO (keine) (130)

```
OFF IO
```

Unterbindet HP-IL Kommunikation; die Einheitscode-Zuweisungen bleiben jedoch im Speicher erhalten.

- Globale Deklaration.

OFF TIMER # (of.t.#) (187)

```
OFF TIMER # Timernummer
```

```
690 OFF TIMER # 22
```

```
700 OFF TIMER # J7
```

Desaktiviert den spezifizierten Timer.

- Globale Deklaration.
- Nicht über das Tastenfeld ausführbar.

ON ERROR (o..) (258)

```
ON ERROR [zulässige Anweisung...] oder [Befehl...]
```

```
710 ON ERROR GOSUB 500
720 ON ERROR X2=X2-1 @ RESTORE @ OFF ERROR
```

Spezifiziert die Ausführungsreihenfolge bei Auftreten eines Fehlers, ohne dabei das Programm anzuhalten.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Folgende Anweisungen sind nach ON ERROR nicht zulässig:

```
FOR          END DEF    ON TIMER
NEXT         IF         OPTION BASE
DATA        IMAGE     REAL
DEF FN      INTEGER    SHORT
DIM         ON ERROR
```

ON...GOSUB (keine) (183)

ON...GOTO (keine) (182)

```
ON numerischer Ausdruck GOSUB Zeilennummer [, Zeilennummer...]
ON numerischer Ausdruck GOTO Zeilennummer [, Zeilennummer...]
```

```
730 ON F9 GOSUB 3100,3200,3400
740 ON MOD(N5,5) GOTO 3650,3660,3670,3680,3690
```

Bewirkt in Abhängigkeit vom Wert des numerischen Ausdrucks Verzweigungen zu den in der Anweisungsliste spezifizierten Anweisungen oder Unterprogrammen. Der numerische Ausdruck muß auf die Werte 1, 2, 3, ..., n gerundet werden können.

- Nicht über das Tastenfeld ausführbar.
- Nach ON...GOTO in der gleichen Zeile stehende Anweisungen werden niemals ausgeführt.

ON TIMER # (o.t.) (186)

```
ON TIMER # Timernummer, Sekunden [zulässige Anweisung...] oder [Befehl...]
```

```
750 ON TIMER #21, 3600 RESTORE @ GOSUB 450
760 ON TIMER #K2, 60*M9 DISP 'Zeit' @ BEEP
```

Setzt den spezifizierten Timer (*Timernummer*) und dessen Unterbrechungsintervall (*Sekunden*) sowie die Ausführungsreihenfolge des Programms bei einer Timerunterbrechung.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Folgende Anweisungen sind nach ON TIMER nicht zulässig:

```
FOR          END DEF    ON TIMER
NEXT         IF         OPTION BASE
DATA        IMAGE     REAL
DEF FN      INTEGER    SHORT
DIM         ON ERROR
```

OPTION ANGLE (op.a.d./op.a.r.) (85)

```
OPTION ANGLE DEGREES
              RADIANS
```

Setzt den trigonometrischen Modus auf Altgrad oder Radiant.

- Globale Deklaration.

OPTION BASE (op..) (193)

```
OPTION BASE 0
              1
```

Spezifiziert die Untergrenze von numerischen Feldern.

- Lokale Deklaration.
- Nicht über das Tastenfeld ausführbar.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Muß vor einem Bezug auf Feldvariablen auftreten.
- Ein Programm darf nicht mehr als eine OPTION BASE Anweisung enthalten.

PACK (pa.) (137)

```
PACK ' :Einheitscode '
```

```
pack ' :CA '
```

Packt das Medium in der spezifizierten Massenspeichereinheit.

PLIST (pl.) (56)

```
PLIST [ 'Filename' ] [ :Anfangszeilennummer [ :Endzeilennummer] ]
      [ KEYS ]
```

```
770 PLIST F1$
plist 'namen'
```

Listet eine oder mehrere Zeilen des momentanen oder spezifizierten BASIC- oder Textfiles. Die Auslistung erfolgt über alle PRINTER IS Einheiten, oder falls keine PRINTER IS Einheiten zugewiesen worden sind, über die Anzeige.

POP (p.) (183)

```
POP
```

Hebt die aus der letzten GOSUB Anweisung resultierende anstehende Rücksprungbedingung auf.

- Nicht über das Tastenfeld ausführbar.

PRINT (pri.) (167)

```
PRINT [Ausgabeliste] [ : ]
```



```
780 PRINT B4$;
790 PRINT 16,T2(4*I),H5*E2,53
```

Sendet die Elemente der Ausgabeliste an die PRINTER IS Einheiten oder, falls keine PRINTER IS Einheiten zugewiesen worden sind, an die Anzeige.

PRINT # (Pri.#) (217)

```
PRINT # Filenummer [, Zeilennummer] ; Ausdruck [, Ausdruck...]
```

```
800 PRINT # 6; H$, 'Tage', 05
810 PRINT #J7, I+1; D$&E$, 120
```

Schreibt die aufgeführten Elemente in die nächste bzw. die spezifizierte Zeile des der Filenummer zugeordneten Files.

(221)

```
PRINT # Filenummer ; Zeilennummer
```

Stellt den Filepointer des der *Filenummer* zugeordneten Files auf die spezifizierte DATA Anweisung, *löscht* die Zeile und läßt den Filepointer auf dieser Zeile stehen.

- Nicht über das Tastenfeld ausführbar.

PRINTER IS (Pri.) (128)

```
PRINTER IS ' ; Einheitscode [, ; Einheitscode... ]'
```

```
820 PRINTER IS ' ; PR'
```

Bestimmt die spezifizierte Einheit als Druckereinheit.

- Globale Deklaration.

(129)

```
PRINTER IS *
;'
```

Hebt alle Zuweisungen von Druckereinheiten auf.

- Globale Deklaration.

PRINT USING (Pri.us.) (238)

```
PRINT USING Zeilennummer ; [Ausgabeliste]
Formatstring
```

```
830 PRINT USING 540; 'Daten', D1$, X5
840 PRINT USING " 'neuer Wert' /10(' - ')" ;
```

Sendet die Elemente der Ausgabeliste entsprechend dem in der Anweisung oder dem in der über die *Zeilennummer* spezifizierten IMAGE Anweisung enthaltenen *Formatstring* an die Druckereinheiten.

PROTECT (PR.) (121)

```
PROTECT
```

Schützt eine Magnetkarte vor Überschreiben.

PURGE (PU.) (50/137)

```
PURGE [ 'Filename '
        KEYS
        APPT
        'Filespezifikator ' ]
```

```
850 PURGE 'TEMP:CA'
purge appt
purge 'file12'
```

Löscht den spezifizierten File aus dem Speicher oder von einem Massenspeichermedium und gibt den Speicherplatz an das System zurück.

PUT (keine) (204)

```
PUT Stringausdruck mit einem Zeichen
```

```
860 PUT CHR$(172)
```

Simuliert das Drücken der entsprechenden Taste oder Tastenkombination. Speichert den *Zeichencode des Stringausdrucks* im Wartetastenbuffer.

PWIDTH (PW.) (39)

```
PWIDTH Anzahl Zeichen
```

```
870 PWIDTH 10*I
pwidth 24
```

Setzt die Zeilenlänge für PRINT und PLIST Anweisungen.

- Globale Deklaration.

RANDOMIZE (RA.) (83)

```
RANDOMIZE [numerischer Ausdruck]
```

```
880 RANDOMIZE
890 RANDOMIZE 7^(9*Z2)
```

Benutzt den spezifizierten numerischen Ausdruck oder die Uhrzeit der Systemuhr zur Berechnung eines neuen Startwerts für den Zufallszahlengenerator.

- Globale Deklaration.

READ (keine) (211)

```
READ Variablenname [, Variablenname...]
```

```
900 READ R$,W3(7)
910 READ V(,)
```

Weist die spezifizierten Variablen numerische Werte oder Stringkonstanten aus den DATA Anweisungen des Programms zu. Bei Feldvariablen ohne Indices in den Klammern werden allen Elementen des Feldes Werte zugewiesen.

- Nicht über das Tastenfeld ausführbar.

READ # (r, #) (219)

```
READ # Filenummer [, Zeilennummer] ; Variable [, Variable...]
```

```
920 READ #2,10*X; S4$,M
930 READ #I+1; A,B( )
```

Weist den spezifizierten Variablen die Konstanten der folgenden oder der spezifizierten DATA Anweisung(en) in dem über die *Filenummer* spezifizierten Datenfile zu.

- Nicht über das Tastenfeld ausführbar.

(221)

```
READ # Filenummer ; Zeilennummer
```

Stellt den Filepointer auf den Anfang der spezifizierten DATA Anweisung in dem über die *Filenummer* spezifizierten Datenfile.

- Nicht über das Tastenfeld ausführbar.

REAL (re.) (194)

```
REAL numerische Variable [(<Indices>)] [, numerische Variable [(<Indices>)]...]
```

```
940 REAL X,H7(30),05(15,15)
```

Dimensioniert REAL Variablen (volle Genauigkeit) und reserviert Speicherplatz für sie. Gültig für einfache und für Feldvariablen.

- Lokale Deklaration.
- Nicht zulässig nach THEN, ELSE, ON ERROR und ON TIMER.
- Muß die letzte Anweisung in der Zeile sein.
- Muß vor einem Bezug auf die deklarierten Variablen auftreten.

REM (keine) (166)

```
REM [Zeichen[Zeichen...]]
!
```

```
950 REM - DDW Unterprogramm
960 ! Warteschleife
970 LET L = L + 1 ! Erhöht den Schleifenzähler.
```

Dient der Dokumentation von Programm listings. Alle Zeichen nach der REM Anweisung werden als Kommentar behandelt. Zusätzlich kann auch das Ausrufungszeichen (!) als Kommentareinleitung benutzt werden. REM muß am Zeilenanfang stehen; ! kann an beliebiger Stelle in der Zeile benutzt werden (alle Zeichen nach ! werden als Kommentar aufgefaßt).

RENAME (ren.) (59)

```
RENAME ['alter Filename'] TO 'neuer Filename'
```

```
980 RENAME TO 'ALTDAT'
990 RENAME 'DIESEN' TO 'JENEN'
rename keys to 'tasten3'
```

Ändert den Namen des momentanen oder des spezifizierten Files im Speicher.

(137)

```
RENAME 'Filespezifikator' TO 'Filespezifikator'
      'Filename'
RENAME 'Filename' TO 'Filespezifikator'
```

```
1000 RENAME 'FILE20:CA' TO 'FILE30'
rename 'namfile' to 'neuname:tp'
```

Ändert den Namen des spezifizierten Massenspeicherfiles.

RENUMBER (renu.) (57)

```
RENUMBER [Anfangszeilennummer [, Inkrement [, alte Anfangszeilennummer
      [, alte Endzeilennummer]]]]
```

```
1010 RENUMBER
renumber 100,5
renumber 100,5,10,250
```

Bewirkt eine Umnummerierung des momentanen BASIC- oder Textfiles bzw. des spezifizierten Teilbereichs.

RESTORE (res.) (213)

```
RESTORE [Zeilennummer]
```

```
1020 RESTORE
1030 RESTORE 250
```

Stellt den Datenpointer auf den Anfang der spezifizierten DATA Anweisung in dem BASIC-File, der die RESTORE Anweisung enthält. Wird keine Zeilennummer spezifiziert, wird der Datenpointer auf die DATA Anweisung mit der niedrigsten Zeilennummer gestellt.

- Nicht über das Tastenfeld ausführbar.

RESTORE # (res.#) (221)

```
RESTORE # Filenummer [, Zeilennummer]
```

```
1040 RESTORE #3
1050 RESTORE #I+1,L-100
```

Stellt den Datenpointer auf den Anfang der spezifizierten oder der ersten DATA Anweisung in dem durch die *Filenummer* zugeordneten Datenfile.

- Nicht über das Tastenfeld ausführbar.

RESTORE IO (res..) (130)

```
RESTORE IO
```

Stellt die HP-IL Kommunikation nach einer Unterbrechung mit `OFF IO` oder durch bestimmte Schleifenfehler wieder her. Es wird die Schleifenkonfiguration der letzten `ASSIGN IO` Deklaration zugrundegelegt.

- Globale Deklaration.

RETURN (ret.) (182)

```
RETURN
```

Das Programm verzweigt aus dem momentanen Unterprogramm zu der Anweisung, die der das Unterprogramm aufrufenden Anweisung folgt.

- Nicht über das Tastenfeld ausführbar.
- In der gleichen Zeile nachfolgende Anweisungen werden nie ausgeführt.

RUN (keine) (158)

```
RUN [ Zeilennummer  
      'Filename' [, Zeilennummer]
```

```
run 150  
run 'finanz', 1000
```

Bewirkt die Ausführung des momentanen oder spezifizierten Programms beginnend bei der ersten oder der spezifizierten *Zeilennummer*.

SHORT (sh.) (194)

```
SHORT numerische Variable [<Indices>] [, numerische Variable [<Indices>]...]
```

```
1060 SHORT B(4,4),D8,C(3)
```

Dimensioniert Variablen der Genauigkeit `SHORT` und reserviert den benötigten Speicherplatz. Kann zur Deklaration von einfachen und Feldvariablen verwendet werden.

- Lokale Deklaration.
- Nicht zulässig nach `THEN`, `ELSE`, `ON ERROR` und `ON TIMER`.
- Muß die letzte Anweisung in einer Zeile sein.
- Muß vor einem Bezug auf die deklarierten Variablen auftreten.

STANDBY (s.) (29)

```
STANDBY OFF
```

Setzt den HP-75 auf automatisches Ausschalten nach fünfminütiger Inaktivität. Zusätzlich wird die Anzahl erfolgloser Kommunikationsversuche mit HP-IL Einheiten begrenzt.

- Globale Deklaration.

```
STANDBY ON
```

Desaktiviert die Ausschaltautomatik und setzt den HP-75 auf unbegrenzt lange Wartezeit auf Reaktionen von HP-IL Einheiten.

- Globale Deklaration.

STOP (keine)

(165)

```
STOP
```

Unterbricht die Programmausführung, ohne das Programm zu deallokalisieren.

- Nicht über das Tastenfeld ausführbar.
- Auf diese Anweisung in der gleichen Zeile folgende Anweisungen werden niemals ausgeführt.

TRACE FLOW (t . .)

(252)

```
TRACE FLOW
```

Setzt den HP-75 auf Anzeige der Ursprungs- und Zielzeilennummer jeder Verzweigung während der Ausführung eines Programms.

- Globale Deklaration.

TRACE OFF (tr . o .)

(253)

```
TRACE OFF
```

Hebt alle Verfolgungsoperationen auf.

- Globale Deklaration.

TRACE VARS (tr . v .)

(253)

```
TRACE VARS
```

Setzt den HP-75 auf Anzeige der Zeilennummer und des Variablennamens (und der aktuellen Werte numerischer Variablen) bei jeder Wertänderung einer Variablen.

- Globale Deklaration.

TRANSFORM (tr .)

(275)

```
TRANSFORM ['Filename'] INTO BASIC
                        TEXT
                        LIF1
```

```
transform into basic
1070 TRANSFORM 'DATA10' INTO LIF1
```

Bewirkt die Umwandlung des Filetyps (BASIC-, Text- oder Austauschfile) eines Files im Speicher.

UNPROTECT (u.)

(121)

```
UNPROTECT
```

Entfernt den Schreibschutz auf einer Magnetkarte.

WAIT (wa.)

(166)

```
WAIT Anzahl Sekunden
```

```
1080 WAIT 2  
1090 WAIT M/60
```

Verzögert die Programmausführung um die spezifizierte Zeitspanne.

WIDTH (w.)

(39)

```
WIDTH Anzahl Zeichen
```

```
width 32  
1100 WIDTH 2*B2
```

Setzt die Zeilenlänge für DISP und LIST Anweisungen.

- Globale Deklaration.

Referenztabellen

Der Zeichensatz

Die Funktion `CHR$` gibt das Anzeigezeichen eines gegebenen Dezimalcodes von 0 bis 255 zurück. Die Argumente werden auf ganzzahlige Werte gerundet und in den richtigen Bereich transformiert (modulo 256). Der Dezimalcode eines unterstrichenen Anzeigezeichens ist um 128 größer als der des gleichen, nicht unterstrichenen Zeichens.

Die Funktion `NUM` erzeugt den Dezimalcode des ersten Zeichens eines gegebenen Zeichenstrings.

Zeichen mit den Dezimalcodes von 32 bis 126 sind Standard-Druckzeichen nach der ASCII (American Standard for Information Interchange) Konvention zugeordnet. Die Reaktion von Peripherieeinheiten auf die Dezimalcodes von 0 bis 31 und von 127 bis 255 finden Sie im Benutzerhandbuch der entsprechenden HP-IL Einheiten. Wenn Sie Sonderzeichen (zum Beispiel das Zeichen Form-Feed, FF) darstellen wollen, sollten Sie nicht das Anzeigezeichen selbst benutzen – Sie müssen sonst mit unerwartetem Verhalten der Peripherieeinheit bei Programmlistings rechnen; verwenden Sie stattdessen die Funktion `CHR$` (zum Beispiel `CHR$(12)` an Stelle von `μ`).

Ein Stern (*) deutet an, daß Sie zuerst `SHIFT` `I/R` drücken müssen, um das mit der Taste oder Tastenfolge verknüpfte Zeichen anzuzeigen. Wenn keine Tastenfolge angegeben ist, kann das entsprechende Zeichen nur mit dem Befehl `CHR$` erzeugt werden.

Dezimal-code	Anzeige-zeichen	Tastensequenz	Dezimal-code	Anzeige-zeichen	Tastensequenz
0	␣	<code>CTL</code> <code>Leertaste</code>	128	␣	* <code>ATTN</code>
1	␣	<code>CTL</code> <code>A</code>	129	␣	* <code>TIME</code>
2	␣	<code>CTL</code> <code>B</code>	130	␣	* <code>APPT</code>
3	␣	<code>CTL</code> <code>C</code>	131	␣	* <code>EDIT</code>
4	␣	<code>CTL</code> <code>D</code>	132	␣	* <code>↑</code>
5	␣	<code>CTL</code> <code>E</code>	133	␣	* <code>↓</code>
6	␣	<code>CTL</code> <code>F</code>	134	␣	* <code>←</code>
7	␣	<code>CTL</code> <code>G</code>	135	␣	* <code>→</code>
8	BS	<code>CTL</code> <code>H</code>	136	␣	* <code>I/R</code>
9	␣	<code>CTL</code> <code>I</code>	137	␣	* <code>FET</code>
10	LF	<code>CTL</code> <code>J</code>	138	␣	* <code>DEL</code>
11	␣	<code>CTL</code> <code>K</code>	139	␣	* <code>CLR</code>
12	μ	<code>CTL</code> <code>L</code>	140	␣	* <code>LOCK</code>
13	CR	<code>CTL</code> <code>M</code>	141	␣	* <code>RUN</code>
14	␣	<code>CTL</code> <code>N</code>	142	␣	* <code>TAB</code>
15	␣	<code>CTL</code> <code>O</code>	143	␣	
16	␣	<code>CTL</code> <code>P</code>	144	␣	
17	␣	<code>CTL</code> <code>Q</code>	145	␣	
18	␣	<code>CTL</code> <code>R</code>	146	␣	

Dezimal-code	Anzeige-zeichen	Tastenfolge
19	€	CTL S
20	π	CTL T
21	µ	CTL U
22	§	CTL V
23	ö	CTL W
24	ö	CTL X
25	o	CTL Y
26	ü	CTL Z
27	ESC	CTL BACK
28	~	CTL +
29	#	CTL =
30	£	CTL ;
31	⌘	CTL 8
32		space bar
33	!	SHIFT 1
34	"	SHIFT 2
35	#	SHIFT 3
36	\$	SHIFT 4
37	%	SHIFT 5
38	&	SHIFT 6
39	'	SHIFT 7
40	(SHIFT 8
41)	SHIFT 9
42	*	*
43	+	+
44	,	,
45	-	-
46	.	.
47	/	/
48	0	0
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	SHIFT ;
59	;	;
60	<	SHIFT ,

Dezimal-code	Anzeige-zeichen	Tastenfolge
147	€	
148	π	
149	µ	
150	§	
151	ö	
152	ö	
153	o	
154	ü	
155	ESC	
156	~	
157	#	
158	£	
159	⌘	
160		* SHIFT ATTN
161	!	* SHIFT TIME
162	"	* SHIFT APPT
163	#	* SHIFT EDIT
164	\$	* SHIFT ↑
165	%	* SHIFT ↓
166	&	* SHIFT ←
167	'	* SHIFT →
168	(* SHIFT I/R
169)	* SHIFT FET
170	*	* SHIFT DEL
171	+	* SHIFT CLR
172	,	* SHIFT LOCK
173	-	* SHIFT RUN
174	.	* SHIFT TAB
175	/	
176	0	* CTL 0
177	1	* CTL 1
178	2	* CTL 2
179	3	* CTL 3
180	4	* CTL 4
181	5	* CTL 5
182	6	* CTL 6
183	7	* CTL 7
184	8	* CTL 8
185	9	* CTL 9
186	:	
187	;	
188	<	

Dezimal-code	Anzeige-zeichen	Tastenfolge
61	=	[=]
62	>	[SHIFT] [.]
63	?	[SHIFT] [/]
64	@	[SHIFT] [+]
65	A	[SHIFT] [A]
66	B	[SHIFT] [B]
67	C	[SHIFT] [C]
68	D	[SHIFT] [D]
69	E	[SHIFT] [E]
70	F	[SHIFT] [F]
71	G	[SHIFT] [G]
72	H	[SHIFT] [H]
73	I	[SHIFT] [I]
74	J	[SHIFT] [J]
75	K	[SHIFT] [K]
76	L	[SHIFT] [L]
77	M	[SHIFT] [M]
78	N	[SHIFT] [N]
79	O	[SHIFT] [O]
80	P	[SHIFT] [P]
81	Q	[SHIFT] [Q]
82	R	[SHIFT] [R]
83	S	[SHIFT] [S]
84	T	[SHIFT] [T]
85	U	[SHIFT] [U]
86	V	[SHIFT] [V]
87	W	[SHIFT] [W]
88	X	[SHIFT] [X]
89	Y	[SHIFT] [Y]
90	Z	[SHIFT] [Z]
91	[[SHIFT] [0]
92	/	[CTL] [/]
93]	[SHIFT] [-]
94	>	[SHIFT] [-]
95	[[CTL] [-]
96	7	[CTL] [7]
97	a	[A]
98	b	[B]
99	c	[C]
100	d	[D]
101	e	[E]
102	f	[F]
103	g	[G]

Dezimal-code	Anzeige-zeichen	Tastenfolge
189	=	
190	>	
191	?	
192	@	* [CTL] [ATTN]
193	A	* [CTL] [TIME]
194	B	* [CTL] [APPT]
195	C	* [CTL] [EDIT]
196	D	* [CTL] [↑]
197	E	* [CTL] [↓]
198	F	* [CTL] [←]
199	G	* [CTL] [→]
200	H	* [CTL] [I/R]
201	I	* [CTL] [FET]
202	J	* [CTL] [DEL]
203	K	* [CTL] [CLR]
204	L	* [CTL] [LOCK]
205	M	* [CTL] [RUN]
206	N	* [CTL] [TAB]
207	O	
208	P	
209	Q	
210	R	
211	S	
212	T	
213	U	
214	V	
215	W	
216	X	
217	Y	
218	Z	
219	[
220	/	
221]	
222	>	
223	[
224	7	* [SHIFT] [CTL] [ATTN]
225	a	* [SHIFT] [CTL] [TIME]
226	b	* [SHIFT] [CTL] [APPT]
227	c	* [SHIFT] [CTL] [EDIT]
228	d	* [SHIFT] [CTL] [↑]
229	e	* [SHIFT] [CTL] [↓]
230	f	* [SHIFT] [CTL] [←]
231	g	* [SHIFT] [CTL] [→]

Dezimal-code	Anzeige-zeichen	Tastensequenz	Dezimal-code	Anzeige-zeichen	Tastensequenz
104	h	[H]	232	h	* [SHIFT] [CTL] [I/R]
105	i	[I]	233	i	* [SHIFT] [CTL] [FET]
106	j	[J]	234	j	* [SHIFT] [CTL] [DEL]
107	k	[K]	235	k	* [SHIFT] [CTL] [CLR]
108	l	[L]	236	l	* [SHIFT] [CTL] [LOCK]
109	m	[M]	237	m	* [SHIFT] [CTL] [RUN]
110	n	[N]	238	n	* [SHIFT] [CTL] [TAB]
111	o	[O]	239	o	
112	p	[P]	240	p	
113	q	[Q]	241	q	
114	r	[R]	242	r	
115	s	[S]	243	s	
116	t	[T]	244	t	
117	u	[U]	245	u	
118	v	[V]	246	v	
119	w	[W]	247	w	
120	x	[X]	248	x	
121	y	[Y]	249	y	
122	z	[Z]	250	z	
123	<	[CTL] [,]	251	<	
124	=	[SHIFT] [=]	252	=	
125	>	[CTL] [.]	253	>	
126	~	[CTL] [*]	254	~	BYE↑
127	␣	[CTL] [9]	255	␣	

Fehlerbedingungen

Auftreten eines Fehlers

Beim Auftreten eines Fehlers erzeugt der HP-75 ein Tonsignal, setzt die Statusanzeige **ERROR** und zeigt eine **ERROR** oder **WARNING** Meldung mit der derzeitigen Verzögerungsrate an. Wenn der Fehler während einer Programmausführung auftritt, wird auch die Nummer der den Fehler erzeugenden Zeile angezeigt. Mit [SHIFT] [FET] wird die Meldung noch einmal angezeigt, solange die Taste [FET] gedrückt gehalten wird. Mit **errn** [RTN] erhalten Sie die Identifikationsnummer des Fehlers oder der Warnung. Wenn Sie [CLR], [ATTN], [TIME], [APPT], [EDIT], [FET], [↑], [↓], [RTN] oder [RUN] drücken, wird die Statusanzeige zurückgesetzt und die Fehlermeldung gelöscht.

Nach einer **WARNING** gibt der HP-75 Ersatzwerte vor, und die Ausführung des Programms wird fortgesetzt (falls nicht **ON ERROR** deklariert worden ist). Eine Fehlerbedingung unterbricht die Programmausführung bei der den Fehler erzeugenden Anweisung. Wenn das unterbrochene Programm gleichzeitig der momentane File ist, wird der Filepointer auf die Zeile gestellt, in der der Fehler aufgetreten ist; mit [FET] und dann [RTN] können Sie die Zeile zur Anzeige bringen. Das Programm bleibt initialisiert, bis Sie eine Zeile editieren oder ein anderes Programm starten.

Mathematische Warnungen und Fehler		
Nummer	Meldung und Ursache	Ersatzwert*
1	num too small Zahl liegt zwischen \pm EPS.	0
2	num too large <ul style="list-style-type: none"> • größer als \pmINF. • größer als in SHORT möglich. • größer als in INTEGER möglich. 	$\pm 9.999999999999999E499$ $\pm 9.9999E99$ ± 99999
3	COT or CSC inf COT oder CSC ist $n \times 180^\circ$; n = ganzzahlig	$9.999999999999999E499$
4	TAN or SEC inf TAN oder SEC ist $n \times 90^\circ$, n ganzzahlig. und ungerade.	$9.999999999999999E499$
5	0^neg Null zu einer negativen Potenz.	$9.999999999999999E499$
6	0^0 Null hoch null.	1
7	no value. <ul style="list-style-type: none"> • Der Wert einer Stringvariablen wird abgerufen, bevor ein Wert zugewiesen wurde. • Der Wert einer einfachen numerischen Variablen oder eines numerischen Feldelements wird abgerufen, bevor ein Wert zugewiesen wurde. 	' ' 0
8	/zero Division durch Null.	$\pm 9.999999999999999E499$
Die übrigen mathematischen Fehler besitzen keine Ersatzwerte.		
9	neg^non-integer Negativwert zu einer negativen Potenz.	
10	SQR(neg number) Quadratwurzel einer negativen Zahl.	
11	arg out of range Argument zu groß oder zu klein.	
12	LOG(0) Logarithmus von Null.	
13	LOG(neg number) Logarithmus einer negativen Zahl.	

* bei DEFAULT ON.

Systemfehler (14-18)	
Nummer	Meldung und Ursache
14	Low batteries Ersetzen Sie die Batterien oder stecken Sie das Netzadapter/Ladegerät ein.
15	system error Der HP-75 muß evtl. zurückgesetzt werden.
16	not enough memory <ul style="list-style-type: none"> • nicht genügend Speicherplatz verfügbar, um einen Magnetkarten- oder Bandkassettenfile in den Speicher einzulesen. • Ein File oder Programm benötigt zu viel Speicher. Mögliche Lösungen: <ul style="list-style-type: none"> • Löschen Sie eine oder mehrere Zeilen des momentanen Files. • Führen Sie CLEAR VARS aus, um die Rechnervariablen zu löschen. • Verkleinern Sie die Genauigkeit oder Dimensionen von Variablen. • Löschen Sie einen oder mehrere Files aus dem Speicher. • Spezifizieren Sie unbenutzte Datenfilepointer in ASSIGN # TO *. • Führen Sie CATALOG aus und drücken Sie dann <input type="checkbox"/> oder <input type="checkbox"/>, um nicht mehr benötigte Files zu finden, und löschen Sie diese mit <input type="checkbox"/> PURGE. • Desaktivieren Sie die HP-IL Schleife mit ASSIGN IO *.
17	RAM is invalid <ul style="list-style-type: none"> • Zeigt einen defekten Schaltkreis an. Das Gerät muß gegebenenfalls repariert werden. • Zeigt einen Speicherverlust an.
18	ROM missing Ein notwendiges Einsteck-ROM oder LEX-File fehlt.

Warnungen des Kartenlesers (19-26)	
Nummer	Meldung und Ursache
19	write protected Versuch, eine schreibgeschützte Magnetkarte zu beschreiben. Führen Sie zuerst UNPROTECT durch.
20	not this file Eine Spur gehört nicht zum derzeit eingelesenen File.
21	verify failed Eine Spur muß noch zwei weitere Male durch den Magnetkartenleser geführt werden, einmal zum Beschreiben der Spur und einmal zur Prüfung. Reinigen Sie die Magnetkarte zuerst.
22	unknown card Die Information auf der Spur wird vom HP-75 nicht erkannt.
23	bad read/write Ein Fehlversuch, eine Spur richtig zu beschreiben oder einzulesen. Reinigen Sie die Magnetkarte und versuchen Sie es noch einmal.
24	pulled too fast Ziehen Sie die Magnetkarte langsamer durch den Magnetkartenleser.
25	pulled too slow Ziehen Sie die Magnetkarte schneller durch den Magnetkartenleser.
26	wrong name Der Filespezifikator in einem COPY Befehl stimmt mit dem Namen des Magnetkartenfiles nicht überein.

Programmfehler (27-54)	
Nummer	Meldung und Ursache
27	<p><code>invalid subscript</code></p> <ul style="list-style-type: none"> • Ein Feldindex liegt außerhalb des erlaubten Bereichs. • Ein Teilstring einer Stringvariablen liegt außerhalb der dimensionierten Stringlänge.
28	<p><code>record overflow</code></p> <p>Eine direkte <code>PRINT #</code> Anweisung bewegt den Datenpointer über das Zeilenende hinaus.</p>
29	<p><code>ON ERROR overflow</code></p> <p>Eine <code>ON ERROR</code> Routine sollte den Fehler 49 – <code>GOSUB overflow</code> – bearbeiten. Der HP-75 stoppt die Programmausführung und zeigt diese Meldung an, anstatt mit einem weiteren Unterprogramm den Fehler noch zu vergrößern.</p>
30	<p><code>OPTION BASE</code></p> <ul style="list-style-type: none"> • Ein unzulässiger <code>OPTION BASE</code> Parameter wurde spezifiziert. • Die Anweisung <code>OPTION BASE</code> erscheint nach einem Bezug auf eine Feldvariable. • Mehrfache <code>OPTION BASE</code> Anweisungen in einem Programm.
31	<p><code>CONT before RUN</code></p> <p>Es wurde vor der Initialisierung oder nach der Deallokation eines Programms <code>CONT</code> eingegeben oder <code>SHIFT</code> <code>RUN</code> gedrückt.</p>
32	<p><code>missing line</code></p> <ul style="list-style-type: none"> • Eine Zeile, die mit <code>GOTO</code> oder <code>GOSUB</code> adressiert wird, ist nicht vorhanden. • Eine Zeile, die in <code>DISP USING</code> oder <code>PRINT USING</code> adressiert wird, fehlt.
33	<p><code>data type</code></p> <p>Eine <code>READ</code> oder <code>READ #</code> Anweisung versucht, Stringinformation in eine numerische Variable einzulesen.</p>
34	<p><code>no data</code></p> <ul style="list-style-type: none"> • In einer <code>READ</code> oder <code>READ #</code> Anweisung (seriell) wird versucht, über das Zeilenende hinaus zu lesen. • Es wird eine <code>RESTORE</code> oder <code>RESTORE #</code> Anweisung ohne Daten im File ausgeführt. • Eine existierende Zeile, die nicht das Schlüsselwort <code>DATA</code> enthält, wird in einer der Anweisungen <code>RESTORE, RESTORE #, READ #</code> oder <code>PRINT #</code> (direkter Zugriff) spezifiziert.
35	<p><code>DIM exist var</code></p> <p>Genauigkeitsdeklaration für eine Variable, die schon zuvor im Programm benutzt wurde.</p>
36	<p><code>Invalid DIM</code></p> <p>Unzulässiges Dimensionieren eines Feldes nach einer <code>OPTION BASE</code> Deklaration.</p>
37	<p><code>duplicate FN</code></p> <p>Der gleiche Funktionsname erscheint in zwei oder mehreren <code>DEF FN</code> Anweisungen.</p>
38	<p><code>no END DEF</code></p> <p>Eine benutzerdefinierte Funktion mit mehreren Zeilen enthält keine <code>END DEF</code> Anweisung.</p>
39	<p><code>FN missing</code></p> <p>Versuchte Verzweigung in die Mitte einer mehrzeiligen benutzerdefinierten Funktion.</p>

Programmfehler (27-54) (Fortsetzung)	
Nummer	Meldung und Ursache
40	FN parameter <ul style="list-style-type: none"> Die formale Parameterliste in der Funktionsdefinition stimmt nicht mit der gegebenen Parameterliste im Hauptprogramm überein. Versuch, eine nicht vorhandene benutzerdefinierte Funktion aufzurufen.
41	FN calls itself Eine benutzerdefinierte Funktion ist rekursiv definiert.
42	string too long <ul style="list-style-type: none"> Versuch, einer Stringvariablen zu viele Zeichen zuzuweisen. Schreiben eines Strings mit mehr als 251 Zeichen in einen BASIC-File, oder mit mehr als 255 Zeichen in einen Textfile. Erzeugt eine Warnung und schneidet den String ab.
43*	numeric input <ul style="list-style-type: none"> Versuch, bei einer INPUT Anweisung Alpha-Zeichen einzugeben. Drücken von RTN, bevor alle erforderlichen Werte für die numerischen Variablen der Eingabeliste eingegeben worden sind.
44*	too many inputs Eingabe zu vieler Daten nach einer INPUT Anweisung.
45	missing ASSIGN # <ul style="list-style-type: none"> Ausführung einer der Befehle PRINT #, READ #, RESTORE # ohne vorherige Zuweisung einer Filenummer.
46	missing NEXT Fehlendes NEXT in einer FOR...NEXT Schleife.
47	no matching FOR Ein NEXT ohne dazugehöriges FOR wurde angetroffen. Vermutlich unrichtig verschachtelte Schleifen.
48	FOR overflow Mehr als 255 Verschachtelungen von FOR...NEXT Schleifen.
49	GOSUB overflow Mehr als 255 Verschachtelungen von Unterprogrammen.
50	RETURN w/o GOSUB Ein RETURN ohne anstehende Rücksprungbedingung wurde angetroffen.
51	PRINT# to runfile Ein Programm bezieht sich in einer PRINT # Anweisung auf sich selbst.
52	invalid IMAGE Mindestens einer der Spezifikatoren in IMAGE, DISP USING oder PRINT USING ist unrichtig. Kann durch ein unzulässiges Zeichen im IMAGE String verursacht sein.
53	invalid USING Einer der Spezifikatoren in IMAGE, DISP USING oder PRINT USING kann ein anzuzeigendes oder zu druckendes Datenelement nicht darstellen.
54	invalid TAB Ein TAB Argument wird auf einen Wert kleiner 1 gerundet. Sie erhalten eine Warnung, und der Ersatzwert 1 wird zur Verfügung gestellt.

* Bei einem Eingabefehler fordert der HP-75 noch einmal zur Eingabe auf. Wenn bei einem INPUT Fehler ON ERROR deklariert ist, werden den INPUT Variablen so viele Werte wie möglich zugewiesen; danach wird die ON ERROR Anweisung ausgeführt.

HP-IL Fehler (55 bis 61)	
Nummer	Meldung und Ursache
55	<p>ASSIGN IO needed Ausführung von DISPLAY IS, PRINTER IS, LIST IO, OFF IO oder RESTORE IO ohne vorhergehende Zuweisung der Einheiten in der Schleife.</p>
56	<p>no loop response Die HP-IL Schleife ist angeschlossen, aber keine Einheit reagierte auf den Befehl ASSIGN IO.</p>
57	<p>bad transmission</p> <ul style="list-style-type: none"> • Hardware-Fehler. • Drücken von ATTN während einer Übertragung.
58	<p>loop timeout</p> <ul style="list-style-type: none"> • Eine Unterbrechung oder Stromausfall eines Peripheriegerätes während einer Schleifenübertragung. • Ein Gerät benötigt bei STANDBY OFF Einstellung mehr als 10 Sekunden, um eine einzelne HP-IL Instruktion auszuführen. <p>Versichern Sie sich, daß alle Einheiten richtig angeschlossen und mit Strom versorgt sind, und führen Sie dann RESTORE IO aus. Falls die Unterbrechung beabsichtigt war, verbinden Sie die Buchsen IN und OUT des HP-75 mit einem HP-IL Kabel und führen Sie OFF IO aus. Setzen Sie STANDBY ON, falls eine Einheit länger als 10 Sekunden zur Reaktion auf einen Befehl benötigt.</p>
59	<p>too many names</p> <ul style="list-style-type: none"> • Mehr Namen als Einheiten in ASSIGN IO. Erzeugt ein Warnung; es werden alle existierenden Geräte zugewiesen. • Nach einem RESTORE Befehl befinden sich mehr Einheitsnamen im Speicher als Einheiten in der Schleife gefunden wurden. Erzeugt einen Fehler; es werden keine Einheiten zugewiesen. Verbinden Sie die ursprünglichen Einheiten und führen Sie nochmals RESTORE IO aus, oder deklarieren Sie erneut mit ASSIGN IO.
60	<p>RESTORE IO needed Der HP-75 versucht, nach einem OFF IO Befehl eine HP-IL Instruktion auszuführen.</p>
61	<p>>31 devices Es befinden sich einschließlich des HP-75 mehr als 31 Einheiten in der HP-IL Schleife.</p>

File- und Einheitsfehler (62-69)	
Nummer	Meldung und Ursache
62	<p>file not found Der spezifizierte File existiert nicht im Speicher oder auf dem Massenspeichermedium.</p>
63	<p>invalid filespec</p> <ul style="list-style-type: none"> • Ungültiger Name für einen File im Speicher. Namen sind hier auf ein bis acht Zeichen beschränkt; davon muß das erste ein Buchstabe oder Punkt und die folgenden Buchstaben oder Ziffern sein. • Ungültiger Name für einen File auf Massenspeicher oder Magnetkarte. Namen sind hier auf ein bis maximal acht Zeichen beschränkt, das erste muß ein Buchstabe sein (Punkte sind nicht erlaubt). • Ungültiger HP-IL Einheitscode. Auf einen oder zwei Buchstaben oder einen Buchstaben und eine Ziffer beschränkt. • TRANSFORM INTO LIF1 wird auf einen Arbeitsfile angewandt. • Ein Anzeige-Einheitscode wird in einem Massenspeicherbefehl spezifiziert.
64	<p>duplicate name</p> <ul style="list-style-type: none"> • Doppelter Filename. Es existiert schon ein File dieses Namens im Speicher oder auf einem Massenspeichermedium. • Doppelter Einheitscode in einer ASSIGN IO Deklaration. Wenn ASSIGN IO Einheiten nacheinander vom Tastenfeld aus zugewiesen werden, erhalten Sie eine Warnung; der HP-75 fordert einen anderen Einheitscode an.

File- und Einheitsfehler (62 bis 69) (Fortsetzung)	
Nummer	Meldung und Ursache
65	<p>access restricted</p> <ul style="list-style-type: none"> • Versuch, einen Termin-, Text-, LEX oder LIF1 File zu starten. • Versuch, einen privaten BASIC-File zu editieren, zu listen oder zu kopieren. • Versuchtetes PRINT # oder READ # auf einen privaten BASIC-File, einen LEX File oder einen LIF1 File. • Versuch, mit PRINT # einen numerischen Wert in einen Textfile zu schreiben.
66	<p>invalid password</p> <ul style="list-style-type: none"> • Das Paßwort für einen Massenspeicher- oder Magnetkartenfile ist beim Kopieren des Files nicht korrekt spezifiziert. • Ein Paßwort wird beim Kopieren eines LIF1 Files oder eines nicht vom HP-75 erzeugten Files auf oder von einem Massenspeichermedium spezifiziert. Erzeugt eine Warnung; Kopieren ist möglich, aber es wird kein Paßwort angefügt.
67	<p>line too long</p> <ul style="list-style-type: none"> • Eine von einem File gelesene oder aufgelistete Zeile enthält mehr als drei Anzeigefenster lange Information. Erzeugt eine Warnung, und die ersten 94 Zeichen der Zeile werden angezeigt. Beim Drücken von [RTN] wird die abgerufene Zeile beim 94. Zeichen abgeschnitten, falls dies möglich ist; andernfalls wird ein Syntaxfehler gemeldet . • Abrufen einer Tastendefinition mit mehr als 80 Zeichen. Nur die ersten 80 Zeichen der Tastendefinition werden angezeigt. • Versuch, einen LIF1 File mit außergewöhnlich langen Zeilen in Text oder BASIC umzuwandeln; erzeugt diese Warnung und setzt ! ? direkt hinter der Zeilennummer ein.
68	<p>wrong file type</p> <ul style="list-style-type: none"> • Versuch, einen Nichttextfile in keys umzubenennen. • Versuch, einen Nichtterminfile in appt umzubenennen. • Versuch, einen File in einen File eines anderen Typs zu mischen. • Versuch, einen Massenspeicherfile unbekanntem Typs (?) in den Speicher einzulesen. • Versuch, einen Text-, Termin-, LEX oder LIF1 File auf einen Privatfile zu kopieren; erzeugt eine Warnung und eine nichtprivate Kopie.
69	<p>workfile name</p> <p>Versuch, während der Editierung eines nichtleeren, unbenannten Arbeitsfiles einen anderen File zu editieren. Führen Sie die Operationen NAME, RENAME oder PURGE auf dem Arbeitsfile durch, bevor Sie den EDIT Befehl ausführen oder bei CATALOG die Taste [EDIT] drücken.</p>

Warnung im TIME-Modus (70)	
Nummer	Meldung und Ursache
70	<p>time adjust bad</p> <p>Versuch, die Ganggeschwindigkeit um mehr als ±10% zu verändern. Erzeugt eine Warnung; der Anpassungsfaktor bleibt erhalten, und eine neue Anpassungsperiode beginnt.</p>

Fehler im APPT-Modus (71 bis 77)	
Nummer	Meldung und Ursache
71	<p><code>duplicate APPT</code> Versuch, einen doppelten Termin einzugeben. Termine müssen sich in mindestens einem Zeichen unterscheiden.</p>
72	<p><code>day/date mismatch</code> Wochentag und Datum stimmen nicht überein.</p>
73	<p><code>bad day field</code> Falsche Buchstabierung des Wochentags.</p>
74	<p><code>bad date field</code> Unzulässiger Parameter für Monat, Tag oder Jahr. Kann durch ein unerlaubtes Zeichen in den Feldern <code>Mo</code>, <code>Dy</code> oder <code>Yr</code> verursacht werden.</p>
75	<p><code>bad time field</code> Unzulässiger Parameter für Stunde oder Minute. Kann durch ein unerlaubtes Zeichen in den Feldern <code>Hr</code>, <code>Mn</code> oder <code>AM</code> verursacht werden.</p>
76	<p><code>bad rep field</code></p> <ul style="list-style-type: none"> • Unzulässiger Parameter oder nicht erlaubtes Zeichen in der <code>Rept</code> (<i>repeat</i>) Maske. • Drücken einer Systemtaste, während die <code>Rept</code> Maske in der Anzeige steht.
77	<p><code>bad alarm spec</code></p> <ul style="list-style-type: none"> • Unzulässige Alarmart. Muß eine Ziffer von 0 bis 9 sein. • Unzulässige Terminart. Muß ein <code>N</code>, <code>A</code> oder <code>R</code> sein.

Syntaxfehler (78 bis 91)	
Nummer	Meldung und Ursache
78	<p><code>syntax</code></p> <ul style="list-style-type: none"> • Unzulässige Zwischenräume oder Zeichen in der Zeile. Der Cursor steht auf dem Zeichen, wo der Fehler gefunden wurde. • Bei <code>TRANSFORM INTO BASIC</code> konnte eine Zeile nicht interpretiert werden. Der Fehler wird nach der Umformung des gesamten Files gemeldet; nicht interpretierte Zeilen werden in Programmkommentare, die mit <code>! ?</code> beginnen, umgeformt.
79	<p><code>; expected</code> Fehlendes Semikolon zwischen Parametern.</p>
80	<p><code>) expected</code> Fehlende schließende Klammer in einem Ausdruck.</p>
81	<p><code>comma expected</code> Fehlendes Komma zwischen Parametern.</p>
82	<p><code>string expected</code> Nicht gelungener Versuch, einen Ausdruck als String zu interpretieren.</p>
83	<p><code>missing TO</code> Das Schlüsselwort <code>TO</code> ohne Zwischenräume muß in den Befehl eingefügt werden.</p>
84	<p><code>extra characters</code> Überflüssige Zeichen am Zeilenende. Kann durch falsche Eingabe einer Instruktion entstanden sein.</p>

Syntaxfehler (78 bis 91) (Fortsetzung)	
Nummer	Meldung und Ursache
85	<code>expr too big</code> Der Ausdruck ist zu lang, um vom HP-75 ausgewertet zu werden. Kann durch zu viele verschachtelte Klammern oder zu viele Operationen im Ausdruck entstanden sein.
86	<code>illegal context</code> Nach <code>THEN, ELSE ON TIMER #</code> oder <code>ON ERROR</code> nicht erlaubte Anweisung.
87	<code>bad expression</code> <ul style="list-style-type: none"> • Syntaxfehler in einem Ausdruck; zum Beispiel zu viele Operatoren zwischen Operanden. • Versuch, einen LIF1 File ohne Zeilennummern in einen Text- oder BASIC-File umzuwandeln. Der File bleibt ein LIF1 File.
88	<code>bad statement</code> <ul style="list-style-type: none"> • Eine unzulässige Abkürzung. • Eine unvollständige Anweisung. • Versuch, eine Anweisung, die nur in Programmen erlaubt ist (zum Beispiel <code>GOTO</code>) über das Tastenfeld auszuführen.
89	<code>bad parameter</code> <ul style="list-style-type: none"> • Falscher Parametertyp; zum Beispiel ein Stringargument für eine Funktion mit numerischem Argument. • Eingabe von Parametern, die außerhalb des Definitionsbereichs liegen; Eingabe unzulässiger Zeichen oder zu vieler Parameter.
90	<code>bad line number</code> <ul style="list-style-type: none"> • Versuch einer unzulässigen Umnummerierung. Erzeugt eine Warnung; der spezifizierte Fileteil wird mit Ersatzwerten numeriert. • Versuch, mit einer <code>PRINT #</code> Anweisung eine größere Zeilennummer als 9999 zu erzeugen. Dabei entsteht ein Fehler; der Datenpointer bleibt am Fileende stehen, die Schreiboperation wird nicht ausgeführt. • Versuch, im <code>AUTO</code> Befehl eine Anfangszeilennummer größer als 9999 vorzugeben.
91	<code>missing parameter</code> Fehlen eines notwendigen Parameters in einer Funktion, einer Anweisung oder einem Befehl.

Massenspeicherfehler (92 bis 97)	
Nummer	Meldung und Ursache
92	<code>dev not mass mem</code> Versuch einer Massenspeicheroperation mit einer nicht zulässigen Peripherieinheit.
93	<code>mass mem error</code> Die Massenspeichereinheit funktioniert nicht einwandfrei, möglicherweise wegen zu niedriger Batteriespannung.
94	<code>no medium</code> Kein Medium in der Massenspeichereinheit vorhanden.
95	<code>medium full</code> Das Medium in der Massenspeichereinheit kann keine weiteren Files aufnehmen. Löschen Sie einen oder mehrere Files, packen Sie das Medium oder setzen Sie ein neues Medium ein.
96	<code>invalid medium</code> Die Massenspeichereinheit kann nicht vom Medium lesen oder darauf schreiben. Möglicherweise ist das Medium nicht formatiert; initialisieren Sie das Medium. Möglicherweise ist das Medium beschädigt oder defekt, oder die Aufnahmeköpfe sind verschmutzt. Reinigen Sie die Köpfe entsprechend der Anleitung im Handbuch des Peripheriegeräts.
97	<code>invalid pack</code> Unterbrechung der <code>PACK</code> Operation. Möglicherweise muß das Medium mit <code>INITIALIZE</code> neu formatiert werden.

Alphabetische Auflistung

Meldung	Nummer	Meldung	Nummer
access restricted	65	medium full	95
arg out of range	11	missing ASSIGN#	45
ASSIGN IO needed	55	missing line	32
bad alarm spec	77	missing NEXT	46
bad date field	74	missing parameter	91
bad day field	73	missing TO	83
bad expression	87	neg^non-integer	9
bad line number	90	no data	34
bad parameter	89	no END DEF	38
bad read/write	23	no loop response	56
bad rep field	76	no matching FOR	47
bad statement	88	no medium	94
bad time field	75	no value	7
bad transmission	57	not enough memory	16
comma expected	81	not this file	20
CONT before RUN	31	num too large	2
COT or CSC inf	3	num too small	1
		numeric input	43
data type	33	ON ERROR overflow	29
day/date mismatch	72	OPTION BASE	30
dev not mass mem	92	PRINT# to runfile	51
DIM exist var	35	pulled too fast	24
duplicate APPT	71	pulled too slow	25
duplicate FN	37		
duplicate name	64	RAM is invalid	17
		record overflow	28
expr too big	85	RESTORE IO needed	60
extra characters	84	RETURN w/o GOSUB	50
		ROM missing	18
file not found	62	SQR(neg number)	10
FN calls itself	41	string expected	82
FN missing	39	string too long	42
FN parameter	40	syntax	78
FOR overflow	48	system error	15
GOSUB overflow	49	TAN or SEC inf	4
illegal context	86	time adjust bad	70
invalid DIM	36	too many inputs	44
invalid filespec	63	too many names	59
invalid IMAGE	52	unknown card	22
invalid medium	96	verify failed	21
invalid pack	97	workfile name?	69
invalid password	66	write protected	19
invalid subscript	27	wrong file type	68
invalid TAB	54	wrong name	26
invalid USING	53		
line too long	67	/zero	8
LOG(neg number)	13	0^neg	5
LOG(0)	12	0^0	6
loop timeout	58	>31 devices	61
low batteries	14) expected	80
mass mem error	93	; expected	79

Systemspeicheranforderungen

Sie können die Größe eines initialisierten Programmes bestimmen, indem Sie zuerst die Ausführung unterbrechen (mit `ATTN`) und den verfügbaren Speicher bestimmen (mit `MEM RTN`). Deallokalisieren Sie danach das Programm, indem Sie eine Zeile editieren. Der dadurch verfügbar gemachte zusätzliche Speicherplatz plus die Größe des deallokierten Files (wie im Katalogeintrag angegeben) ergeben die Gesamtgröße.

Speicherobjekt	Benötigter Speicherplatz
Termine	15 Bytes für den Terminfile + 7 Bytes/Termin + 1 Byte/Zeichen im Notizfeld (NOTE) + 5 Bytes/Wiederholungstermin.
Datenelemente	5 Bytes/DATA Anweisung + 4 Bytes/INTEGER Wert + 9 Bytes/REAL -Wert + 2 Bytes/Datenelement + 1 Byte/Zeichen in Strings.
Datenpointer	15 Bytes/Datenpointer, der mit <code>ASSIGN #</code> erzeugt worden ist.
Files (BASIC und Text)	15 Bytes/File + 3 Bytes/Zeile + 1–3 Bytes/Schlüsselwort (BASIC) oder + 1 Byte/Zeichen (Text).
Files (LEX und Austausch-)	Siehe Katalogeintrag des Files
HP-IL Zuweisungen	7 Bytes/deklarierte Einheit (in <code>ASSIGN IO</code>).
Tastendefinitionen	3 Bytes/Definition + 1 Byte/Zeichen im Tastendefinitionsstring + 1 Byte für das End-Semikolon.
Massenspeicherbefehle (CAT, COPY, PURGE, RENAME)	43 Bytes/Befehl.
Massenspeichereinheiten	105 Bytes/Einheit.
PACK Befehl	256 Bytes + 6 Bytes/File.
Einsteck-ROMs	Siehe Benutzerhandbuch des ROMs.
Programmaufrufe	30 Bytes/CALL Anweisung + 2 Bytes/Variable des aufrufenden Programms + Anzahl Bytes, die die Variablengenauigkeit erfordert.
Timer	63 Bytes/Timer (der in einer <code>ON TIMER</code> Anweisung gesetzt wurde) + ein oder mehr Bytes pro Timeranweisung.
TRANSFORM Befehl	Maximal 255 Bytes während der Umformung.
Variablen (Rechner- und initialisierte Programmvariablen)	
Einfache numerische Variablen	
REAL	12 Bytes/Variable.
SHORT	8 Bytes/Variable.
INTEGER	7 Bytes/Variable.
Numerische Feldvariablen	
REAL	10 Bytes/Feldvariable plus 8 Bytes/Element.
SHORT	4 Bytes/Element.
INTEGER	3 Bytes/Element.
Stringvariablen	8 Bytes/Variable + dimensionierte Länge.

Anzeige-Escapecodes

ESC symbolisiert das Escapezeichen, Dezimalcode 27. Wenn Sie ESC durch `CTL BACK` oder `CHR$(27)` erzeugen, wird ein Escapezeichen an die Anzeige des HP-75 und an die anderen `DISPLAY IS` Einheiten gesandt. Wenn `CHR$(27)` ausgedruckt wird, wird ein Escapezeichen an die `PRINTER IS` Einheiten gesandt. Die Anzeige des HP-75 und die meisten HP-IL Einheiten zeigen spezielle Reaktionen auf vordefinierte *Escapecodes*; das sind Anzeige- oder Druckeranweisungen, die aus dem Escapezeichen und den direkt darauf folgenden Zeichen bestehen. Ein gegebener Escapecode kann von einer Einheit ignoriert werden, während er von einer anderen als Systembefehl verstanden wird. Ein ESC `C` zum Beispiel wird vom Thermodrucker HP82162A ignoriert, das Video-Interface HP82163 bewegt dagegen den Cursor um eine Stelle nach rechts. Ein ESC `C` kann zum Beispiel mit der Eingabe `DISP CHR$(27)&'C'` an die Anzeige des HP-75 und an andere Anzeigeeinheiten gesandt werden.

Die Anzeige des HP-75 reagiert auf 12 Escapecodes. Alle anderen Escapecodes, die in `DISP` oder `PRINT` Anweisungen benutzt werden können, werden von der Anzeige ignoriert.

Die 32 Fensterpositionen werden durch die Nummern 0 bis 31 identifiziert.

Escapecode	Instruktion	Beschreibung
ESC <code>C</code>	Cursor nach rechts*	Bewegt den Cursor um eine Stelle nach rechts.
ESC <code>D</code>	Cursor nach links*	Bewegt den Cursor um eine Stelle nach links.
ESC <code>E</code>	Lösche alles*	Setzt den Cursor auf Position 0 und löscht die Anzeige.
ESC <code>G</code>	Cursor Return	Setzt den Cursor auf Position 0.
ESC <code>H</code>	Cursor Home*	Setzt den Cursor auf Position 0.
ESC <code>J</code>	Lösche nach Cursor*	Löscht die Anzeige ab der momentanen Cursorposition.
ESC <code>K</code>	Lösche bis Zeilenende	Löscht die Anzeige ab der momentanen Cursorposition.
ESC <code>O</code>	Lösche Zeichen mit Rundumverschiebung	Löscht das Zeichen an der Position des Cursors und schiebt alle nachlaufenden Zeichen mit Rundumverschiebung nach links.
ESC <code>F</code>	Lösche Zeichen	Löscht das Zeichen an der Position des Cursors und schiebt alle nachlaufenden Zeichen nach links.
ESC <code><</code>	Cursor aus*	Schaltet den Cursor aus.
ESC <code>></code>	Cursor ein*	Schaltet den Cursor ein.
ESC <code>%cr</code>	Cursor an Adresse*	Bewegt den Cursor an die vom Dezimalcode des ersten Anzeigezeichens modulo 32 spezifizierte Anzeigeposition. Der Dezimalcode des zweiten Zeichens wird vom Video-Interface als Zeilenparameter benutzt, vom HP-75 aber ignoriert. Mit <code>DISP CHR\$(27);'%';CHR\$(16);CHR\$(8);'hier'</code> zum Beispiel wird der Cursor auf Spalte 16 (und Zeile 8 des Video-Interface) gesetzt und von dieser Position an <code>hier</code> angezeigt.

Ein Stern (*) deutet an, daß das Video-Interface HP82163 vergleichbar reagiert, wenn es als `DISPLAY IS` Einheit erklärt worden ist. Schlagen Sie auch im Benutzerhandbuch des Video-Interface nach.

Geben Sie nach einem Escapecode ein Semikolon ein, um den Wagenrücklauf/Zeilenvorschub zu unterdrücken, der normalerweise eine DISPLAY oder PRINT Anweisung abschließt.

Mit **SHIFT** **↑** erzeugen Sie ein ESC T (Rollen nach oben um eine Zeile) und mit **SHIFT** **↓** ein ESC S (Rollen nach unten um eine Zeile) auf DISPLAY IS Einheiten. Das Anzeigefenster des HP-75 ignoriert diese Codes.

System-Voreinstellungen

Bedingung	Nach einem Zurücksetzen	Beim Einschalten nach einem automatischen Ausschalten
ALARM	ON	Vorherige Einstellung.
ASSIGN IO	Keine HP-IL Zuweisungen.	Vorherige HP-IL Zuweisungen.
ASSIGN #	Keine Zuweisung von Filenummern.	Vorherige Zuweisung der Filenummern.
BEEP	ON	Vorherige Einstellung.
CTL FET	Zeitsetzmaske.	Leere Zeile.
DEFAULT	ON	Vorherige Einstellung.
DEF KEY	Identitätszuweisungen.	Vorherige Tastendefinitionen.
DELAY	1 Sekunde.	Vorherige Einstellung.
DISPLAY IS	* (Anzeige des HP-75).	Vorherige Anzeigeeinheiten.
ENDLINE	CR/LF.	Vorherige Einstellung.
ERRL	0	Vorherige Fehlerzeile.
ERRN	0	Vorherige Fehlernummer.
EXACT	EXACT Marken sind gelöscht.	EXACT Marken sind unverändert.
FETCH	Zeile 0 des Arbeitsfiles.	Vorherige anstehende Zeile.
Files	BASIC Arbeitsfile (0 Bytes, Zeit und Datum der Uhr.)	Vorherige BASIC Files, Textfiles, APPT-Files, Keys-Files, LEX Files, LIF1 Files.
Tastenfeld	Nicht umgeschaltet.	Vorheriger Tastenfeldmodus.
LOCK	Kein Paßwort.	Vorheriges Paßwort.
MARGIN	91	Vorherige Einstellung.
Modus	TIME	EDIT
PRINTER IS	* (Anzeige des HP-75).	Vorherige Druckereinheiten.
PWIDTH	32 Spalten.	Vorherige Einstellung.
RES	0	Letztes numerisches Ergebnis.
RND	.529199358633	Nächste Zahl in der Folge.
SHIFT FET	Leere Anzeige.	Leere Anzeige.
STANDBY	OFF	Vorherige Einstellung.
STATS Maske	MDY, ~, AM, YEAR	Vorherige Einstellungen.
TIME Anzeige	Zeitsetzwerte.	Aktuelle Zeit.
Trigonometrischer Modus	OPTION ANGLE RADIANS	Vorherige Einstellung.
Variablen: Rechner- Programm-	Keine. Keine.	Vorherige Zuweisungen. Vorherige Zuweisungen bei allokatisiertem Programm.
WIDTH	32 Spalten	Vorherige Einstellung.

Abkürzungen

Die folgenden Abkürzungen stellen die kürzesten vom HP-75 noch getrennt erkennbaren Kurzformen dar. Längere Abkürzungen sind erlaubt, solange der Punkt nicht für das letzte Zeichen des Schlüsselwortes steht. Abkürzungen dürfen im Wort selbst keine Leerzeichen enthalten.

Bei Listings und beim Abrufen von Zeilen setzt der HP-75 die vollständigen Schlüsselworte ein.

Schlüsselwort	Abkürzung	Schlüsselwort	Abkürzung
ACOS()	ac.()	IMAGE	im.
ALARM OFF	al.off	INITIALIZE	ini.
ALARM ON	al.on	INPUT	i.
ANGLE()	ang.()	INTEGER	int.
ASSIGN #	as.#	KEY\$	k.
ASSIGN IO	as..	LIST	l.
AUTO	a.	LIST IO	l..
BEEP	be.	LET FN	le..
BEEP OFF	be.off	LOG10()	lo.()
BEEP ON	be.on	MARGIN	ma.
BYE	b.	MERGE	m.
CAT#()	c.()	NAME	n.
CAT ALL	c..	NEXT	ne.
CEIL()	ce.()	OFF ERROR	of.e.
CHR#()	ch.()	OFF TIMER #	of.t.#
CLEAR <i>Einheitscode</i>	cl.'iec'	ON ERROR	o..
CLEAR LOOP	cl.l.	ON TIMER #	o.t.#
CLEAR VARS	cl..	OPTION ANGLE DEGREES	op.a.d.
COPY	co.	OPTION ANGLE RADIANS	op.a.r.
DATA	da.	OPTION BASE	op..
DEFAULT OFF	defa.off	PACK	pa.
DEFAULT ON	defa.on	PLIST	pl.
DEF KEY	d..	POP	p.
DELAY	d.	PRINT	pr.
DELETE	dele.	PRINT #	pr.#
DISP	di.	PRINTER IS	pr..
DISPLAY IS	di..	PRINT USING	pr.us.
DISP USING	di.us.	PROTECT	pr.
EDIT	e.	PURGE	pu.
EDIT BASIC	e.ba.	PWIDTH	pw.
EDIT TEXT	e.t.	RANDOMIZE	ra.
ELSE	el.	READ #	r.#
END DEF	end d.	REAL	re.
ENDLINE	en.	RENAME	ren.
ERRN	er.	RENUMBER	renu.
FETCH	f.	RESTORE	res.
FETCH KEY	f..	RESTORE #	res.#
FLOOR()	fl.()	RESTORE IO	res..
GOSUB	gos.	RETURN	ret.
GOTO	g.	RUN	r

Schlüsselwort	Abkürzung
SHORT	sh.
STANDBY OFF	s.off
STANDBY ON	s.on
THEN	th.
TIME#	ti.
TRACE FLOW	t..
TRACE OFF	tr.o.
TRACE VARS	tr.v.
TRANSFORM	tr.

Schlüsselwort	Abkürzung
UNPROTECT	u.
UPRC#()	up , ()
VER#	v.
WAIT	wa ,
WIDTH	w.

VERKAUFSNIEDERLASSUNGEN:**Hewlett-Packard GmbH:**

6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 50 04-1
 7030 Böblingen, Herrenbergerstraße 110, Tel. (07031) 667-1
 4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 59 71-1
 2000 Hamburg 60, Kapstadtring 5, Tel. (040) 6 38 04-1
 8028 Taufkirchen, Eschenstraße 5, Tel. (089) 61 17-1
 3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 46 60 01
 8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 52 20 83/87
 1000 Berlin 30, Keithstraße 2-4, Tel. (030) 24 90 86
 6800 Mannheim, Roßlauer Weg 2-4, Tel. (0621) 7 00 50
 7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 7 02 41
 7517 Waldbronn 2, Hewlett-Packard-Straße, Tel. (07243) 602-1

Hewlett-Packard (Schweiz) AG:

Allmend 2, CH-8967 Widen, Tel. (057) 31 21 11

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:

Wagamerstraße-Lieblgasse 1, A-1220 Wien

Hewlett-Packard S.A., Europa-Zentrale:

7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz

SERVICENIEDERLASSUNGEN:**Hewlett-Packard GmbH:**

6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 50 04-1

Hewlett-Packard (Schweiz) AG:

Allmend 2, CH-8967 Widen, Tel. (057) 31 21 11

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:

Wagamerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 23 65 11

