

SHARP

PC-1500

UNTERPROGRAMMKONZEPTION

WWW.
PC-1500
.INFO

Berlin, im Dezember 1983

Sehr geehrter Leser,

Sie halten eine Liste von Unterprogrammen in der Hand, die im System-ROM des SHARP PC-1500 verfügbar sind.

Die Unterprogramme sind nach drei verschiedenen Kriterien sortiert:

Vektor-Subroutines
Startadressen
Unterprogramm-Namen

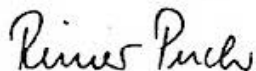
Die Namen wurden von mir willkürlich gewählt und dienen nur der sofortigen Erkennbarkeit von Unterprogrammen in Disassembler-Listings o.ä.

Mit dieser Liste, die im Übrigen keinen Anspruch auf Vollständigkeit erhebt, ist jedoch nur die Bezeichnung und nicht die genaue Funktionsbeschreibung der Unterprogramme verfügbar. Letztere ist im Moment in Arbeit und demnächst in der Reihe

SHARP PC-1500 SPEZIALINFO als INFO 04/83

erhältlich. Bis ich die Funktionen sämtlicher Unterprogramme ermittelt habe, wird noch etwas Zeit vergehen, zumal ich außerdem auch noch studiere. Haben Sie daher bitte etwas Geduld.

Ich wünsche Ihnen nun viel Erfolg bei der Benutzung der Unterprogramme und möchte Sie darauf hinweisen, daß ich bei mißbräuchlicher Benutzung für auftretende Hard- oder Softwareschäden nicht haftbar gemacht werden kann.



Reiner Puch VDI

Hard- und Software Service Berlin

GEORDNET NACH UNTERPROGRAMM-NAMEN:

+	EFBAH	(F0)	2. 1	Addition
*	F01AH	(7E)	2. 1	Multiplikation
-	EFB6H	(--)	2. 1	Subtraktion
/	F084H	(58)	2. 1	Division
^	F89CH	(--)	2. 1	Power Raise
1/X	F080H	(6E)	2. 1	Komplement
10^X	F1D4H	(--)	2. 1	10^x
ABS	F597H	(--)	2. 1	Absolutwert
ACS	F492H	(--)	2. 1	ARCCOS
ADVICE	D03EH	(2A)	8. 3	Behandlung von Systemmeldungen
ARIADDXPT	F7CEH	(72)	2. 4	X+Reg. partiell
ARIADDYXPT	F7CCH	(EE)	2. 4	X+Y partiell
ARICALC	D6DFH	(DE)	2. 3	Funktionsanalyse und -berechnung
ARIFORMAS	F663H	(52)	2. 6	Formieren von Rechenergebnissen
ARIFORMUS	F661H	(E8)	2. 6	Formieren von Rechenergebnissen
ARILOADBAS	DC16H	(30)	2. 5	Wert aus BASIC-Stack laden
ARIOP01	F6B4H	(60)	2. 4	Kopieren X nach Y, Nullsetzen
ARIOP02	F88BH	(62)	2. 4	Laden dezimal 0.6
ARIOP03	F88FH	(6A)	2. 4	Laden dezimal 1.0
ARIOP04	F87FH	(--)	2. 4	Laden dezimal 0.9
ARIOP05	F883H	(--)	2. 4	Laden dezimal 90
ARIOP06	F887H	(--)	2. 4	Laden dezimal 180
ARIRND	F7A7H	(5E)	2. 4	Laden des RND-Registers
ARISAVEBAS	DBF5H	(--)	2. 5	Wert auf BASIC-Stack ablegen
ARISSET	F7B0H	(54)	2. 1	Arithmetikadressen setzen
ARISGNTST	F6E6H	(7C)	2. 4	Vorzeichenstest zwischen X und Y
ARISHFTL	F79CH	(EA)	2. 4	nach links shiften
ARISHFTR	F775H	(74)	2. 4	Nach rechts shiften
ARISUBPT	F7DDH	(7A)	2. 4	Y partiell von X subtrahieren
ARITEST	DB87H	(26)	3. 3	Testen von Arithmetik-Registerinhalten
ARIXCOPY	DC0CH	(--)	2. 5	Übernahme von Werten
ARRAYTEST1	DBB1H	(28)	3. 3	Testen von Felddimensionen
ARRAYTEST2	DBB3H	(--)	3. 3	Testen von Felddimensionen
ASC	D9DDH	(--)	1. 2	ASC
ASCIIDIS	ED5BH	(8A)	5. 4	Anzeige eines ASCII-Zeichens
ASN	F49AH	(--)	2. 1	ARCSIN
ATN	F496H	(--)	2. 1	ARCTAN
AUPOWOFF	E33FH	(--)	5. 1	Auto-Power-Off
BASACCESS	FA89H	(1C)	9. 5	BASIC-Tabellen Zugriff
BASIC1	CFCCH	(--)	8.10	BASIC-Start vorbereiten
BASIC2	CFDOH	(--)	8.10	BASIC-Start vorbereiten
BASINPUT1	CA58H	(42)	4. 5	Standardeingabe mit Prompt ">"
BASINPUT2	CA7AH	(44)	4. 5	Standardeingabe mit bel. Prompt
BASINPUT3	CA80H	(46)	4. 5	Standardeingabe ohne Prompt
BASTEST	DF3BH	(D8)	9. 2	Prüfen, ob BASIC-Programm läuft
BASTRANS	D2DOH	(--)	8. 5	BASIC-Zeile in Input-Buffer
BEEPSET	E655H	(A2)	9. 3	Behandlung des BEEP-PC6
BEEPUX	E66FH	(--)	9. 1	BEEP bezüglich U- und X-Register
BLNKOFF	EF00H	(84)	5. 5	blinkenden Cursor ausschalten
BTABSEAR1	E4A8H	(BE)	4. 2	BASIC-Tabellen suchen
BTABSEAR2	E4B7H	(BC)	4. 2	BASIC-Tabellen suchen
BUFCOPY	FBCBH	(--)	8. 3	Umkopieren zwischen Out- und Stringbuf
CASE	DF23H	(34)	3. 2	Simulation einer Auswahl Schleife
CHANGXS	F7B5H	(64)	2. 2	Austauschen von X und S
CHANGXY	F7B9H	(66)	2. 2	Austauschen von X und Y
CHARCOMP	D0F9H	(--)	3. 1	Vergleich von Strings
CHARLOAD	BDF0H	(A4)	7. 1	Laden eines Zeichens
CHARSAVE	BDCCH	(--)	7. 1	Sichern eines Zeichens
CHR\$	D9B1H	(--)	1. 1	CHR\$
CLEARX	F757H	(EC)	2. 2	Löschen X
CLEARXPT	F75FH	(76)	2. 4	X partiell löschen
CLEARYP	F747H	(70)	2. 4	Y partiell löschen
CLRINP1	D02BH	(--)	8. 7	Eingabepuffer löschen
CLRINP2	D021H	(--)	8. 7	Eingabepuffer löschen

CLRLCD	EE71H	(F2)	5. 3	Löschen der Anzeige
CLRMEM	F763H	(BA)	8. 2	Löschen von Speicherbereichen
CLRVAR	D080H	(--)	8. 6	Variablenspeicher löschen
CMDLOAD	DBD3H	(--)	8. 7	BASIC-Token entsichern
CMDSAVE	DBE0H	(--)	8. 7	BASIC-Token sichern
CMTINIT	BBA5H	(B8)	7. 4	CMT-Ablauf initialisieren
CMTIO	BBF5H	(B4)	7. 3	Setzen der CMT I/O Steuerung
COLCALC1	EE22H	(--)	5. 3	Berechnung von Spaltenadressen 1
COLCALC2	EE1FH	(8C)	5. 3	Berechnung von Spaltenadressen 2
COLORSEL	A519H	(--)	6. 1	Farbbestimmung
COMBINE	D925H	(--)	1. 1	Kombination von Strings
COPYSX	F737H	(--)	2. 2	Kopieren S nach X
COPYSY	F715H	(68)	2. 2	Kopieren S nach Y
COPYXS	F707H	(80)	2. 2	Kopieren X nach S
COPYXY	F70DH	(E6)	2. 2	Kopieren X nach Y
COPYXYPT	F729H	(82)	2. 4	X partiell nach Y kopieren
COPYXZPT	F72FH	(78)	2. 4	X partiell nach Z kopieren
COPYYX	F73DH	(56)	2. 2	Kopieren Y nach X
COS	F391H	(--)	2. 1	COS
CURLOAD	EE99H	(--)	5. 5	Entsicherung Cursordaten und Display
CURMOVE	ED4DH	(--)	5. 1	Zeichen anzeigen und Cursor bewegen
CURSAVE	EE80H	(--)	5. 5	Sicherung von Cursordaten und Display
DATALOAD	DED1H	(D6)	8. 3	Entsichern der BASIC-Daten
DATASAVE	DEE3H	(D4)	8. 3	Sichern der BASIC-Daten
DEG	F531H	(--)	2. 1	Umrechnung in GRAD-Form
DEZHEXTEST	D5F9H	(D0)	3. 5	Umwandlung Dezimal -> Hexadezimal
DIMCALC	DCA6H	(2C)	2. 7	Berechnen von Feldindizes
DISP	ED57H	(--)	5. 1	Display
DMS	F564H	(--)	2. 1	Umrechnung in Grad-Minuten-Sekunden
ERROR1	CD89H	(E4)	9. 3	Standardfehlermeldung #1
ERROR2	CD8BH	(E0)	9. 3	Fehlermeldung UH ausgeben
EXP	F1CBH	(--)	2. 1	e^x
FILETRNS	BD3CH	(AA)	7. 2	File transfer
FINDOP	DA98H	(--)	4. 3	Operator-Prioritätszuordnung
GPRNT	EDF6H	(88)	5. 4	grafisches Display
GRAPHIC	EDEFH	(--)	5. 2	grafisches Display (ähnl. GPRINT)
GRAPHSEL	ABEFH	(--)	6. 1	In Graph-Mode umschalten
HEADERCR	BBD6H	(--)	7. 2	Erzeugen des Kopfes
HEADERIO	BCE8H	(B0)	7. 2	Kopf Input/Output
HEXA	ED95H	(--)	5. 2	Hexadezimal 2 Byte -> 1 Byte
HEXDEZ	DD2DH	(10)	2. 3	Umrechnung Hexadezimal -> Dezimal
HEXDEZTEST	DD1AH	(D2)	2. 2	Umrechnung Hexadezimal -> Dezimal
HEXTEST	ED7DH	(--)	3. 1	Überprüfung von Hexzahlen
IFBREAK	E451H	(A6)	4. 5	Break-Test
INCLCPOI	EDB1H	(8E)	5. 3	Erhöhen des Matrixpointer
INITBST	DF93H	(12)	8. 5	Init BASIC-Startadresse
INITPOI	CFFBH	(3A)	8. 5	Init Register und Pointer
INITRSV	CE9FH	(38)	8. 4	Init RESERVE-Startadresse
INITSEAR	D0B4H	(--)	8. 7	Suchspeicher initialisieren
INT	F5BEH	(--)	2. 1	ganzzahliger Anteil
KEYIN	E41AH	(--)	4. 5	Tastentest
KEYSCAN1	E42CH	(--)	4. 4	Keyscan 1
KEYSCAN2	E243H	(--)	4. 4	Keyscan 2
LASTCHAR	DD13H	(C6)	9. 2	Zugriff auf letztes Zeichen
LCDLOAD	EDD8H	(--)	5. 5	Entsicherung des Display-Inhalts
LCDSAVE	EDC1H	(--)	5. 5	Sicherung des Display-Inhalts
LCDSEAR	EE48H	(--)	5. 4	Adresse für LCD-Zeichen suchen
LEFT\$	D9F3H	(--)	1. 2	LEFT\$
LEN	D9DDH	(--)	1. 2	LEN
LFOFF	A306H	(--)	6. 3	Papiervorschub-Interrupt aus
LINEEND	CCDEH	(--)	8. 7	Zeilenende spezifizieren
LINEFEED	A9F1H	(--)	6. 2	Zeilenvorschub
LINESEAR	CC86H	(--)	4. 1	Zeilennummer suchen
LINESERU	D2E6H	(1A)	4. 2	Zeilennummer suchen bei Programmlauf
LN	F161H	(--)	2. 1	LN
LOADSTR	DEBCH	(DC)	8. 2	Stringdatenbehandlung einlesen

LOALCPOI	EDABH	(90)	5. 3	Laden des Matrixpointer
LOG	F165H	(--)	2. 1	LOG
MEMTEST1	DFD9H	(--)	3. 4	Testen auf Speicherüberlappung
MEMTEST2	DFEEH	(--)	3. 4	Testen auf Speicherüberlappung
MEMTEST3	DF3H	(--)	3. 4	Testen auf Speicherüberlappung
MEMTEST4	DF5H	(16)	3. 4	Testen auf Speicherüberlappung
MEMTEST5	DFFAH	(14)	3. 4	Testen auf Speicherüberlappung
MID\$	D9F3H	(--)	1. 2	MID\$
MOTDRIVE	A8DDH	(--)	6. 1	Motorsteuerung
MOTOROFF	A769H	(--)	6. 3	Motor abschalten
MULTIPLY	DA71H	(50)	2. 8	Multiplizieren zweier 16-Bit-Werte
NEWNRCD	F61BH	(5C)	2. 7	Neue Zufallszahl erzeugen
NEXTCHAR	DD08H	(C0)	8. 1	Einlesen von Zeichen
NEXTLINE	DF72H	(20)	8. 2	Pointer auf nächste Zeile setzen
NUMBERCD	D14CH	(--)	2. 4	Zahlen dezimal einlesen
NUMBERY	D14FH	(--)	2. 4	Zahlen dezimal einlesen
NUMCOMP	D0D2H	(--)	3. 1	Numerischer Vergleich
NXBASEXEC	C400H	(E2)	9. 4	Nächstes BASIC-Kommando nach Next
NXDIRECT	DCD4H	(C2)	3. 2	Vergleich mit U mit Einlesen
NXENDE	DCC5H	(C8)	3. 2	Endetest mit U mit Einlesen
NXRANGE	DCB6H	(02)	3. 2	Bereichstest mit U mit Einlesen
OFF	CD71H	(--)	9. 1	Rechner abschalten
OPNACCESS	FA74H	(3C)	9. 2	Zugriff auf BASIC-Tabellen
PAPSTEP	AA04H	(--)	6. 2	Papiervorschub
PARAM	D6C0H	(2E)	8. 6	Parameter vorbereiten
PENUPDOW	AAE3H	(--)	6. 1	Stift auf/ab
POP78	DDC8H	(CC)	8. 1	X-Register lesen
POPU	D065H	(06)	8. 5	Holt U vom BASIC-Stack
PORTINIT	E153H	(--)	9. 4	Portregister-Initialisierung
PRNT	A781H	(--)	6. 3	Textausgabe
PROGDISP	E8CAH	(--)	5. 3	Program Display
PROSEAR	D2EAH	(--)	4. 1	Suche nach einer Programmzeile
PROTOCOL	FB9DH	(3E)	9. 6	Protokollieren und Tracen
PUSH78	C001H	(CA)	8. 1	X-Register schreiben
PUSHU	D071H	(32)	8. 5	Rettet U auf BASIC-Stack
PVADMINSTR	E234H	(A0)	9. 3	PV-Verwaltung
PVRESET	FB2AH	(1E)	8.12	Rücksetzen des PV-Bits
REMOTE	BF28H	(--)	7. 3	Recorder schalten
RETURNJP	DCF9H	(48)	9. 5	Rücksprungroutine
RETURNJPY	DCFDH	(4A)	9. 5	Rücksprungroutine
RETURNNJ	DCE9H	(4C)	9. 5	Rücksprungroutine
RETURNJY	DCEDH	(4E)	9. 5	Rücksprungroutine
RIGHT\$	D9F3H	(--)	1. 2	RIGHT\$
RMTOFF	BF43H	(--)	7. 1	Remote off
RMTON	BF11H	(--)	7. 1	Remote on
RSVFKTCODE	CE87H	(--)	8.10	RESERVE-Tastencode errechnen
RSVFKTLOAD	CED0H	(--)	8. 9	RESERVE-Tasten belegen
RSVFKTSEAR	CEC4H	(--)	8. 9	Suchen RESERVE-Tastenbelegung
RSVTXTLOAD	CF0BH	(--)	8. 9	RESERVE-Texte belegen
RSVTXTSEAR	CEAFH	(--)	8. 9	Suchen RESERVE-Textbeginn
SEARSTR	DE97H	(0C)	8. 4	Stringende suchen
SGN	F59DH	(--)	2. 1	Vorzeichen
SGNX	F6FBH	(6C)	2. 4	Vorzeichen von X in Akku laden
SIN	F3A2H	(--)	2. 1	SIN
SQR	F0E9H	(--)	2. 1	Quadratwurzel
STDBEEP	E669H	(--)	9. 1	Standard-BEEP
STDCOPY	F741H	(--)	2. 1	Standardkopieren
STORESTR	DEAFH	(24)	8. 2	Stringdatenbehandlung abspeichern
STR\$	D9CFH	(--)	1. 1	STR\$
STRCDSPR	C00EH	(DA)	8. 6	Stringdatenverwaltung
STRDISP	ED00H	(92)	5. 1	String-Display
STRFIND	DFB4H	(--)	8. 8	Stringdatenverwaltung
STRPAR	DF0FH	(36)	4. 4	Stringparameter bestimmen
STRSPR	DFC4H	(--)	8. 8	Stringdatenverwaltung
STRSTORE	DFC5H	(--)	8. 8	Stringdatenverwaltung
STRTEST	DB95H	(--)	3. 3	Stringprüfung!!!

TAN	F39EH	(--)	2. 1	TAN
TEXTSEL	ACBBH	(--)	6. 2	In Text-Mode umschalten
TIME	DE82H	(--)	9. 2	Uhrzeit, Datum ermitteln
TIMER	E573H	(5A)	9. 3	Timer-IC Bedienung über PC
TOKENCR	F957H	(--)	9. 7	Erzeugen von BASIC-Tokens
TOPDISP	ED3BH	(--)	5. 2	Ausgabe vom Displayanfang
TXTEQU	E4A0H	(9E)	3. 4	Vergleichen zweier Texte
TXTPRNT	B4F4H	(--)	6. 4	Druckt einen Text
UHITEST	BBB9H	(B2)	2. 6	U-Register Höchstwerttest
UNEGATE	DAA8H	(--)	2. 6	Negieren von 16-Bit-Werten
URBASEXEC	C401H	(40)	9. 4	Nächstes BASIC-Kommando ohne Next
URDIRECT	DCD5H	(C4)	3. 2	Vergleich mit U ohne Einlesen
UREAD	DBBCH	(F4)	8. 1	U-Register lesen
URENDE	DCC6H	(04)	3. 2	Endetest mit U ohne Einlesen
URRANGE	DCB7H	(00)	3. 2	Bereichstest mit U ohne Einlesen
USINGCR	DAB2H	(--)	9. 6	Erzeugen der USING-Kodierung
UWRITE	DDB5H	(F6)	8. 1	U-Register schreiben
VAL	D9D7H	(--)	1. 2	VAL
VARIAB	D45DH	(CE)	4. 3	Suche von Variablenadressen
VARLOAD	DC20H	(--)	8.10	Wert aus Variable laden
VARNAME	D407H	(--)	8. 8	Variablenname-Ermittlung
VARSEAR	D461H	(0E)	4. 1	Suche von Variablenadressen
VARSPEACE	D371H	(--)	8.11	Reservieren von Variablen-Speicherplatz
VARSTORE	DDD9H	(08)	8.11	Werte in Variablen speichern
WAIT64	E88CH	(AC)	9. 1	Warteschleife
WROUT	EC5CH	(94)	8. 4	Schreiben in den Ausgabepuffer
YSAVE	BDC7H	(A8)	7. 3	Y-Register sichern

WWW.
PC-1500
.INFO

GEORDNET NACH VEKTOR-SUBROUTINES:

COMBINE	D925H	(--)	1. 1	Kombination von Strings
CHR\$	D9B1H	(--)	1. 1	CHR\$
STR\$	D9CFH	(--)	1. 1	STR\$
VAL	D9D7H	(--)	1. 2	VAL
LEN	D9DDH	(--)	1. 2	LEN
ASC	D9DDH	(--)	1. 2	ASC
RIGHT\$	D9F3H	(--)	1. 2	RIGHT\$
MID\$	D9F3H	(--)	1. 2	MID\$
LEFT\$	D9F3H	(--)	1. 2	LEFT\$
-	EFB6H	(--)	2. 1	Subtraktion
VARSAPCE	D371H	(--)	8.11	Reservieren von Variablen-Speicherplatz
TOKENCR	F957H	(--)	9. 7	Erzeugen von BASIC-Tokens
VARLOAD	DC20H	(--)	8.10	Wert aus Variable laden
LN	F161H	(--)	2. 1	LN
LOG	F165H	(--)	2. 1	LOG
EXP	F1CBH	(--)	2. 1	e^x
10^X	F1D4H	(--)	2. 1	10^x
SIN	F3A2H	(--)	2. 1	SIN
COS	F391H	(--)	2. 1	COS
TAN	F39EH	(--)	2. 1	TAN
ASN	F49AH	(--)	2. 1	ARCSIN
ACS	F492H	(--)	2. 1	ARCCOS
ATN	F496H	(--)	2. 1	ARCTAN
DEG	F531H	(--)	2. 1	Umrechnung in GRAD-Form
DMS	F564H	(--)	2. 1	Umrechnung in Grad-Minuten-Sekunden
ABS	F597H	(--)	2. 1	Absolutwert
SGN	F59DH	(--)	2. 1	Vorzeichen
INT	F5BEH	(--)	2. 1	ganzzahliger Anteil
^	F89CH	(--)	2. 1	Power Raise
SQR	F0E9H	(--)	2. 1	Quadratwurzel
USINGCR	DAB2H	(--)	9. 6	Erzeugen der USING-Kodierung
BEEPUX	E66FH	(--)	9. 1	BEEP bezüglich U- und X-Register
BASIC2	CFD0H	(--)	8.10	BASIC-Start vorbereiten
BASIC1	CFCCH	(--)	8.10	BASIC-Start vorbereiten
RSVFKTCODE	CE87H	(--)	8.10	RESERVE-Tastencode errechnen
RSVTXTLOAD	CF0BH	(--)	8. 9	RESERVE-Texte belegen
RSVFKTLOAD	CED0H	(--)	8. 9	RESERVE-Tasten belegen
COPYSX	F737H	(--)	2. 2	Kopieren S nach X
RSVFKTSEAR	CEC4H	(--)	8. 9	Suchen RESERVE-Tastenbelegung
STDCOPY	F741H	(--)	2. 1	Standardkopieren
RSVTXTSEAR	CEAFH	(--)	8. 9	Suchen RESERVE-Textbeginn
PORTINIT	E153H	(--)	9. 4	Portregister-Initialisierung
LINEEND	CCDEH	(--)	8. 7	Zeilenende spezifizieren
NUMCOMP	D0D2H	(--)	3. 1	Numerischer Vergleich
CHARCOMP	D0F9H	(--)	3. 1	Vergleich von Strings
INITSEAR	D0B4H	(--)	8. 7	Suchspeicher initialisieren
CLRVAR	D080H	(--)	8. 6	Variablenspeicher löschen
CLRINP2	D021H	(--)	8. 7	Eingabepuffer löschen
BUFCOPY	FBCBH	(--)	8. 3	Umkopieren zwischen Out- und Stringbuf
ARIOP04	F87FH	(--)	2. 4	Laden dezimal 0.9
STDBEEP	E669H	(--)	9. 1	Standard-BEEP
CLRINP1	D02BH	(--)	8. 7	Eingabepuffer löschen
HEXTEST	ED7DH	(--)	3. 1	überprüfung von Hexzahlen
VARNAME	D407H	(--)	8. 8	Variablenname-Ermittlung
PROSEAR	D2EAH	(--)	4. 1	Suche nach einer Programmzeile
FINDOP	DA98H	(--)	4. 3	Operator-Prioritätszuordnung
KEYSCAN1	E42CH	(--)	4. 4	Keyscan 1
KEYSCAN2	E243H	(--)	4. 4	Keyscan 2
KEYIN	E41AH	(--)	4. 5	Tastentest
MEMTEST3	DFF3H	(--)	3. 4	Testen auf Speicherüberlappung
DISP	ED57H	(--)	5. 1	Display
CURMOVE	ED4DH	(--)	5. 1	Zeichen anzeigen und Cursor bewegen
AUPOWOF	E33FH	(--)	5. 1	Auto-Power-Off
MEMTEST2	DFEEH	(--)	3. 4	Testen auf Speicherüberlappung

TOPDISP	ED3BH	(--)	5. 2	Ausgabe vom Displayanfang
HEXA	ED95H	(--)	5. 2	Hexadezimal 2 Byte -> 1 Byte
GRAPHIC	EDEFH	(--)	5. 2	grafisches Display (ähnlt. GPRINT)
PROGDISP	E8CAH	(--)	5. 3	Program Display
COLCALC1	EE22H	(--)	5. 3	Berechnung von Spaltenadressen 1
MEMTEST1	DFD9H	(--)	3. 4	Testen auf Speicherüberlappung
ARRAYTEST2	DBB3H	(--)	3. 3	Testen von Felddimensionen
STRTEST	DB95H	(--)	3. 3	Stringprüfung
NUMBERY	D14FH	(--)	2. 4	Zahlen dezimal einlesen
NUMBERCD	D14CH	(--)	2. 4	Zahlen dezimal einlesen
UNEGATE	DAA8H	(--)	2. 6	Negieren von 16-Bit-Werten
LCDSEAR	EE48H	(--)	5. 4	Adresse für LCD-Zeichen suchen
LCDSAVE	EDC1H	(--)	5. 5	Sicherung des Display-Inhalts
LCDLOAD	EDD8H	(--)	5. 5	Entsicherung des Display-Inhalts
CURSAVE	EE80H	(--)	5. 5	Sicherung von Cursordaten und Display
CURLOAD	EE99H	(--)	5. 5	Entsicherung Cursordaten und Display
COLORSEL	A519H	(--)	6. 1	Farbbestimmung
MOTDRIVE	A8DDH	(--)	6. 1	Motorsteuerung
MOTOROFF	A769H	(--)	6. 3	Motor abschalten
PENUPDOW	AAE3H	(--)	6. 1	Stift auf/ab
GRAPHSEL	ABEFH	(--)	6. 1	In Graph-Mode umschalten
TEXTSEL	ACBBH	(--)	6. 2	In Text-Mode umschalten
PAPSTEP	AA04H	(--)	6. 2	Papiervorschub
LINEFEED	A9F1H	(--)	6. 2	Zeilenvorschub
PRNT	A781H	(--)	6. 3	Textausgabe
LFOFF	A306H	(--)	6. 3	Papiervorschub-Interrupt aus
TXTPRNT	B4F4H	(--)	6. 4	Druckt einen Text
RMTON	BF11H	(--)	7. 1	Remote on
RMTOFF	BF43H	(--)	7. 1	Remote off
CHARSAVE	BDCCH	(--)	7. 1	Sichern eines Zeichens
STRSTORE	DFC5H	(--)	8. 8	Stringdatenverwaltung
HEADERCR	BBD6H	(--)	7. 2	Erzeugen des Kopfes
STRSPR	DFC4H	(--)	8. 8	Stringdatenverwaltung
STRFIND	DFB4H	(--)	8. 8	Stringdatenverwaltung
CMDLOAD	DBD3H	(--)	8. 7	BASIC-Token entsichern
REMOTE	BF28H	(--)	7. 3	Recorder schalten
CMDSAVE	DBE0H	(--)	8. 7	BASIC-Token sichern
ARIXCOPY	DC0CH	(--)	2. 5	Übernahme von Werten
ARISAVEBAS	DBF5H	(--)	2. 5	Wert auf BASIC-Stack ablegen
BASTRANS	D2D0H	(--)	8. 5	BASIC-Zeile in Input-Buffer
LINESEAR	CC86H	(--)	4. 1	Zeilennummer suchen
OFF	CD71H	(--)	9. 1	Rechner abschalten
ARIOP06	F887H	(--)	2. 4	Laden dezimal 180
ARIOP05	F883H	(--)	2. 4	Laden dezimal 90
TIME	DE82H	(--)	9. 2	Uhrzeit, Datum ermitteln
URRANGE	DCB7H	(00)	3. 2	Bereichstest mit U ohne Einlesen
NXRANGE	DCB6H	(02)	3. 2	Bereichstest mit U mit Einlesen
URENDE	DCC6H	(04)	3. 2	Endetest mit U ohne Einlesen
POPU	D065H	(06)	8. 5	Holt U vom BASIC-Stack
VARSTORE	DDD9H	(08)	8.11	Werte in Variablen speichern
SEARSTR	DE97H	(0C)	8. 4	Stringende suchen
VARSEAR	D461H	(0E)	4. 1	Suche von Variablenadressen
HEXDEZ	DD2DH	(10)	2. 3	Umrechnung Hexadezimal -> Dezimal
INITBST	DF93H	(12)	8. 5	Init BASIC-Startadresse
MEMTEST5	DFFAH	(14)	3. 4	Testen auf Speicherüberlappung
MEMTEST4	DFF5H	(16)	3. 4	Testen auf Speicherüberlappung
LINESEAR	D2E6H	(1A)	4. 2	Zeilennummer suchen bei Programmablauf
BASACCESS	FA89H	(1C)	9. 5	BASIC-Tabellen Zugriff
PVRESET	FB2AH	(1E)	8.12	Rücksetzen des PV-Bits
NEXTLINE	DF72H	(20)	8. 2	Pointer auf nächste Zeile setzen
STORESTR	DEAFH	(24)	8. 2	Stringdatenbehandlung abspeichern
ARITEST	DB87H	(26)	3. 3	Testen von Arithmetik-Registerinhalten
ARRAYTEST1	DBB1H	(28)	3. 3	Testen von Felddimensionen
ADVICE	D03EH	(2A)	8. 3	Behandlung von Systemmeldungen
DIMCALC	DCA6H	(2C)	2. 7	Berechnen von Feldindizes
PARAM	D6C0H	(2E)	8. 8	Parameter vorbereiten

ARILOADBAS	DC16H	(30)	2. 5	Wert aus BASIC-Stack laden
PUSHU	D071H	(32)	8. 5	Rettet U auf BASIC-Stack
CASE	DF23H	(34)	3. 2	Simulation einer Auswahlsschleife
STRPAR	DF0FH	(36)	4. 4	Stringparameter bestimmen
INITRSV	CE9FH	(38)	8. 4	Init RESERVE-Startadresse
INITPOI	CFFBH	(3A)	8. 5	Init Register und Pointer
OPNACCESS	FA74H	(3C)	9. 2	Zugriff auf BASIC-Tabellen
PROTOCOL	FB9DH	(3E)	9. 6	Protokollieren und Tracen
URBASEXEC	C401H	(40)	9. 4	Nächstes BASIC-Kommando ohne Next
BASINPUT1	CA58H	(42)	4. 5	Standardeingabe mit Prompt ">"
BASINPUT2	CA7AH	(44)	4. 5	Standardeingabe mit bel. Prompt
BASINPUT3	CA80H	(46)	4. 5	Standardeingabe ohne Prompt
RETURNJP	DCF9H	(48)	9. 5	Rücksprungroutine
RETURNJPY	DCFDH	(4A)	9. 5	Rücksprungroutine
RETURNNJ	DCE9H	(4C)	9. 5	Rücksprungroutine
RETURNNJY	DCEDH	(4E)	9. 5	Rücksprungroutine
MULTIPLY	DA71H	(50)	2. 8	Multiplizieren zweier 16-Bit-Werte
ARIFORMAS	F663H	(52)	2. 6	Formieren von Rechenergebnissen
ARISSET	F7B0H	(54)	2. 1	Arithmetikadressen setzen
COPYXX	F73DH	(56)	2. 2	Kopieren Y nach X
/	F084H	(58)	2. 1	Division
TIMER	E573H	(5A)	9. 3	Timer-IC Bedienung über PC
NEWRNDCD	F61BH	(5C)	2. 7	Neue Zufallszahl erzeugen
ARIRND	F7A7H	(5E)	2. 4	Laden des RND-Registers
ARIOP01	F6B4H	(60)	2. 4	Kopieren X nach Y, Nullsetzen
ARIOP02	F88BH	(62)	2. 4	Laden dezimal 0.6
CHANGXS	F7B5H	(64)	2. 2	Austauschen von X und S
CHANGXY	F7B9H	(66)	2. 2	Austauschen von X und Y
COPYSY	F715H	(68)	2. 2	Kopieren S nach Y
ARIOP03	F88FH	(6A)	2. 4	Laden dezimal 1.0
SGNX	F6FBH	(6C)	2. 4	Vorzeichen von X in Akku laden
1/X	F080H	(6E)	2. 1	Komplement
CLEARYP	F747H	(70)	2. 4	Y partiell löschen
ARIADDXPT	F7CEH	(72)	2. 4	X+Reg. partiell
ARISHFTR	F775H	(74)	2. 4	Nach rechts shiften
CLEARXPT	F75FH	(76)	2. 4	X partiell löschen
COPYXZPT	F72FH	(78)	2. 4	X partiell nach Z kopieren
ARISUBPT	F7DDH	(7A)	2. 4	Y partiell von X subtrahieren
ARISGNTST	F6E6H	(7C)	2. 4	Vorzeichentest zwischen X und Y
*	F01AH	(7E)	2. 1	Multiplikation
COPYXS	F707H	(80)	2. 2	Kopieren X nach S
COPYXYPT	F729H	(82)	2. 4	X partiell nach Y kopieren
BLNKOFF	EF00H	(84)	5. 5	blinkenden Cursor ausschalten
GPRNT	EDF6H	(88)	5. 4	grafisches Display
ASCIIDIS	ED5BH	(8A)	5. 4	Anzeige eines ASCII-Zeichens
COLCALC2	EE1FH	(8C)	5. 3	Berechnung von Spaltenadressen 2
INCLCPOI	EDB1H	(8E)	5. 3	Erhöhen des Matrixpointer
LOALCPOI	EDABH	(90)	5. 3	Laden des Matrixpointer
STRDISP	ED00H	(92)	5. 1	String-Display
WROUT	EC5CH	(94)	8. 4	Schreiben in den Ausgabepuffer
TXTEQU	E4A0H	(9E)	3. 4	Vergleichen zweier Texte
PVADMINSTR	E234H	(A0)	9. 3	PV-Verwaltung
BEEPSET	E655H	(A2)	9. 3	Behandlung des BEEP-PC6
CHARLOAD	BDF0H	(A4)	7. 1	Laden eines Zeichens
IFBREAK	E451H	(A6)	4. 5	Break-Test
YSAVE	BDC7H	(A8)	7. 3	Y-Register sichern
FILETRNS	BD3CH	(AA)	7. 2	File transfer
WAIT64	E88CH	(AC)	9. 1	Warteschleife
HEADERIO	BCE8H	(B0)	7. 2	Kopf Input/Output
UHITEST	BBB9H	(B2)	2. 6	U-Register Höchstwerttest
CMTIO	BBF5H	(B4)	7. 3	Setzen der CMT I/O Steuerung
CMTINIT	BBA5H	(B8)	7. 4	CMT-Ablauf initialisieren
CLRMEM	F763H	(BA)	8. 2	Löschen von Speicherbereichen
BTABSEAR2	E4B7H	(BC)	4. 2	BASIC-Tabellen suchen
BTABSEAR1	E4A8H	(BE)	4. 2	BASIC-Tabellen suchen
NEXTCHAR	DD08H	(C0)	8. 1	Einflesen von Zeichen

NXDIRECT	DCD4H	(C2)	3. 2	Vergleich mit U mit Einlesen
URDIRECT	DCD5H	(C4)	3. 2	Vergleich mit U ohne Einlesen
LASTCHAR	DD13H	(C6)	9. 2	Zugriff auf letztes Zeichen
NXENDE	DCC5H	(C8)	3. 2	Endetest mit U mit Einlesen
PUSH78	C001H	(CA)	8. 1	X-Register schreiben
POP78	DDC8H	(CC)	8. 1	X-Register lesen
VARIAB	D45DH	(CE)	4. 3	Suche von Variablenadressen
DEZHEXTEST	D5F9H	(D0)	3. 5	Umwandlung Dezimal -> Hexadezimal
HEXDEZTEST	DD1AH	(D2)	2. 2	Umrechnung Hexadezimal -> Dezimal
DATASAVE	DEE3H	(D4)	8. 3	Sichern der BASIC-Daten
DATALOAD	DED1H	(D6)	8. 3	Entsichern der BASIC-Daten
BASTEST	DF3BH	(D8)	9. 2	Prüfen, ob BASIC-Programm läuft
STRCDSPR	C00EH	(DA)	8. 6	Stringdatenverwaltung
LOADSTR	DEBCH	(DC)	8. 2	Stringdatenbehandlung einlesen
ARICALC	D6DFH	(DE)	2. 3	Funktionsanalyse und -berechnung
ERROR2	CD8BH	(E0)	9. 3	Fehlermeldung UH ausgeben
NXBASEXEC	C400H	(E2)	9. 4	Nächstes BASIC-Kommando nach Next
ERROR1	CD89H	(E4)	9. 3	Standardfehlermeldung #1
COPYXY	F70DH	(E6)	2. 2	Kopieren X nach Y
ARIFORMUS	F661H	(E8)	2. 6	Formieren von Rechenergebnissen
ARISHFTL	F79CH	(EA)	2. 4	nach links shiften
CLEARX	F757H	(EC)	2. 2	Löschen X
ARIADDYXPT	F7CCH	(EE)	2. 4	X+Y partiell
+	EFBAH	(F0)	2. 1	Addition
CLRLCD	EE71H	(F2)	5. 3	Löschen der Anzeige
UREAD	DBBCH	(F4)	8. 1	U-Register lesen
UWRITE	DDB5H	(F6)	8. 1	U-Register schreiben

WWW.
PC-1500
.INFO

GEORDNET NACH STARTADRESSEN:

LFOFF	A306H	(--)	6. 3	Papiervorschub-Interrupt aus
COLORSEL	A519H	(--)	6. 1	Farbbestimmung
MOTOROFF	A769H	(--)	6. 3	Motor abschalten
PRNT	A781H	(--)	6. 3	Textausgabe
MOTDRIVE	A8DDH	(--)	6. 1	Motorsteuerung
LINEFEED	A9F1H	(--)	6. 2	Zeilenvorschub
PAPSTEP	AA04H	(--)	6. 2	Papiervorschub
PENUPDOWN	AAE3H	(--)	6. 1	Stift auf/ab
GRAPHSEL	ABEFH	(--)	6. 1	In Graph-Mode umschalten
TEXTSEL	ACBBH	(--)	6. 2	In Text-Mode umschalten
TXTPRNT	B4F4H	(--)	6. 4	Druckt einen Text
CMTINIT	BBA5H	(B8)	7. 4	CMT-Ablauf initialisieren
UHITEST	BBB9H	(B2)	2. 6	U-Register Höchstwerttest
HEADERCR	BBD6H	(--)	7. 2	Erzeugen des Kopfes
CMTIO	BBF5H	(B4)	7. 3	Setzen der CMT I/O Steuerung
HEADERIO	BCE8H	(B0)	7. 2	Kopf Input/Output
FILETRNS	BD3CH	(AA)	7. 2	File transfer
YSAVE	BDC7H	(A8)	7. 3	Y-Register sichern
CHARSAVE	BDCCH	(--)	7. 1	Sichern eines Zeichens
CHARLOAD	BDF0H	(A4)	7. 1	Laden eines Zeichens
RMTON	BF11H	(--)	7. 1	Remote on
REMOTE	BF28H	(--)	7. 3	Recorder schalten
RMTOFF	BF43H	(--)	7. 1	Remote off
PUSH78	C001H	(CA)	8. 1	X-Register schreiben
STRCDSPR	C00EH	(DA)	8. 6	Stringdatenverwaltung
NXBASEXEC	C400H	(E2)	9. 4	Nächstes BASIC-Kommando nach Next
URBASEXEC	C401H	(40)	9. 4	Nächstes BASIC-Kommando ohne Next
BASINPUT1	CA58H	(42)	4. 5	Standerdeingabe mit Prompt ">"
BASINPUT2	CA7AH	(44)	4. 5	Standerdeingabe mit bel. Prompt
BASINPUT3	CA80H	(46)	4. 5	Standerdeingabe ohne Prompt
LINESEAR	CC86H	(--)	4. 1	Zeilennummer suchen
LINEEND	CCDEH	(--)	8. 7	Zeilenende spezifizieren
OFF	CD71H	(--)	9. 1	Rechner abschalten
ERROR1	CD89H	(E4)	9. 3	Standardfehlermeldung #1
ERROR2	CD8BH	(E0)	9. 3	Fehlermeldung UH ausgeben
RSVFKTCODE	CE87H	(--)	8.10	RESERVE-Tastencode errechnen
INITRSV	CE9FH	(38)	8. 4	Init RESERVE-Startadresse
RSVXTSEAR	CEAFH	(--)	8. 9	Suchen RESERVE-Textbeginn
RSVFKTSEAR	CEC4H	(--)	8. 9	Suchen RESERVE-Tastenbelegung
RSVFKTLOAD	CED0H	(--)	8. 9	RESERVE-Tasten belegen
RSVXTLOAD	CF0BH	(--)	8. 9	RESERVE-Texte belegen
BASIC1	CFCCH	(--)	8.10	BASIC-Start vorbereiten
BASIC2	CFD0H	(--)	8.10	BASIC-Start vorbereiten
INITPOI	CFFBH	(3A)	8. 5	Init Register und Pointer
CLRINP2	D021H	(--)	8. 7	Eingabepuffer löschen
CLRINP1	D02BH	(--)	8. 7	Eingabepuffer löschen
ADVICE	D03EH	(2A)	8. 3	Behandlung von Systemmeldungen
POPU	D065H	(06)	8. 5	Holt U vom BASIC-Stack
PUSHU	D071H	(32)	8. 5	Rettet U auf BASIC-Stack
CLRVAR	D080H	(--)	8. 6	Variablenspeicher löschen
INITSEAR	D0B4H	(--)	8. 7	Suchspeicher initialisieren
NUMCOMP	D0D2H	(--)	3. 1	Numerischer Vergleich
CHARCOMP	D0F9H	(--)	3. 1	Vergleich von Strings
NUMBERCD	D14CH	(--)	2. 4	Zahlen dezimal einlesen
NUMBERY	D14FH	(--)	2. 4	Zahlen dezimal einlesen
BASTRANS	D2D0H	(--)	8. 5	BASIC-Zeile in Input-Buffer
LINESEARU	D2E6H	(1A)	4. 2	Zeilennummer suchen bei Programmlauf
PROSEAR	D2EAH	(--)	4. 1	Suche nach einer Programmzeile
VARSPACE	D371H	(--)	8.11	Reservieren von Variablen-Speicherplatz
VARNAME	D407H	(--)	8. 8	Variablenname-Ermittlung
VARIAB	D45DH	(CE)	4. 3	Suche von Variablenadressen
VARSEAR	D461H	(0E)	4. 1	Suche von Variablenadressen
DEZHEXTEST	D5F9H	(D0)	3. 5	Umwandlung Dezimal -> Hexadezimal
PARAM	D6C0H	(2E)	8. 6	Parameter vorbereiten

ARICALC	D6DFH	(DE)	2. 3	Funktionsanalyse und -berechnung
COMBINE	D925H	(--)	1. 1	Kombination von Strings
CHR\$	D9B1H	(--)	1. 1	CHR\$
STR\$	D9CFH	(--)	1. 1	STR\$
VAL	D9D7H	(--)	1. 2	VAL
ASC	D9DDH	(--)	1. 2	ASC
LEN	D9DDH	(--)	1. 2	LEN
LEFT\$	D9F3H	(--)	1. 2	LEFT\$
MID\$	D9F3H	(--)	1. 2	MID\$
RIGHT\$	D9F3H	(--)	1. 2	RIGHT\$
MULTIPLY	DA71H	(50)	2. 8	Multiplizieren zweier 16-Bit-Werte
FINDOP	DA98H	(--)	4. 3	Operator-Prioritätszuordnung
UNEGATE	DAA8H	(--)	2. 6	Negieren von 16-Bit-Werten
USINGCR	DAB2H	(--)	9. 6	Erzeugen der USING-Kodierung
ARITEST	DB87H	(26)	3. 3	Testen von Arithmetik-Registerinhalten
STRTEST	DB95H	(--)	3. 3	Stringprüfung
ARRAYTEST1	DBB1H	(28)	3. 3	Testen von Felddimensionen
ARRAYTEST2	DBB3H	(--)	3. 3	Testen von Felddimensionen
UREAD	DBBCH	(F4)	8. 1	U-Register lesen
CMDLOAD	DBD3H	(--)	8. 7	BASIC-Token entsichern
CMDSAVE	DBE0H	(--)	8. 7	BASIC-Token sichern
ARISAVEBAS	DBF5H	(--)	2. 5	Wert auf BASIC-Stack ablegen
ARIXCOPY	DC0CH	(--)	2. 5	Übernahme von Werten
ARILOADBAS	DC16H	(30)	2. 5	Wert aus BASIC-Stack laden
VARLOAD	DC20H	(--)	8.10	Wert aus Variable laden
DIMCALC	DCA6H	(2C)	2. 7	Berechnen von Feldindizes
NXRANGE	DCB6H	(02)	3. 2	Bereichstest mit U mit Einlesen
URRANGE	DCB7H	(00)	3. 2	Bereichstest mit U ohne Einlesen
NXENDE	DCC5H	(C8)	3. 2	Endetest mit U mit Einlesen
URENDE	DCC6H	(04)	3. 2	Endetest mit U ohne Einlesen
NXDIRECT	DCD4H	(C2)	3. 2	Vergleich mit U mit Einlesen
URDIRECT	DCD5H	(C4)	3. 2	Vergleich mit U ohne Einlesen
RETURNNJ	DCE9H	(4C)	9. 5	Rücksprungroutine
RETURNNJY	DCEDH	(4E)	9. 5	Rücksprungroutine
RETURNJP	DCF9H	(48)	9. 5	Rücksprungroutine
RETURNJPY	DCFDH	(4A)	9. 5	Rücksprungroutine
NEXTCHAR	DD08H	(C0)	8. 1	Einlesen von Zeichen
LASTCHAR	DD13H	(C6)	9. 2	Zugriff auf letztes Zeichen
HEXDEZTEST	DD1AH	(D2)	2. 2	Umrechnung Hexadezimal -> Dezimal
HEXDEZ	DD2DH	(10)	2. 3	Umrechnung Hexadezimal -> Dezimal
UWRITE	DDB5H	(F6)	8. 1	U-Register schreiben
POP78	DDC8H	(CC)	8. 1	X-Register lesen
VARSTORE	DDD9H	(08)	8.11	Werte in Variablen speichern
TIME	DE82H	(--)	9. 2	Uhrzeit, Datum ermitteln
SEARSTR	DE97H	(0C)	8. 4	Stringende suchen
STORESTR	DEAFH	(24)	8. 2	Stringdatenbehandlung abspeichern
LOADSTR	DEBCH	(DC)	8. 2	Stringdatenbehandlung einlesen
DATALOAD	DED1H	(D6)	8. 3	Entsichern der BASIC-Daten
DATASAVE	DEE3H	(D4)	8. 3	Sichern der BASIC-Daten
STRPAR	DF0FH	(36)	4. 4	Stringparameter bestimmen
CASE	DF23H	(34)	3. 2	Simulation einer Auswahl Schleife
BASTEST	DF3BH	(D8)	9. 2	Prüfen, ob BASIC-Programm läuft
NEXTLINE	DF72H	(20)	8. 2	Pointer auf nächste Zeile setzen
INITBST	DF93H	(12)	8. 5	Init BASIC-Startadresse
STRFIND	DFB4H	(--)	8. 8	Stringdatenverwaltung
STRSPR	DFC4H	(--)	8. 8	Stringdatenverwaltung
STRSTORE	DFC5H	(--)	8. 8	Stringdatenverwaltung
MEMTEST1	DFD9H	(--)	3. 4	Testen auf Speicherüberlappung
MEMTEST2	DFEEH	(--)	3. 4	Testen auf Speicherüberlappung
MEMTEST3	DFFFH	(--)	3. 4	Testen auf Speicherüberlappung
MEMTEST4	DFFAH	(16)	3. 4	Testen auf Speicherüberlappung
MEMTEST5	DFFAH	(14)	3. 4	Testen auf Speicherüberlappung
PORTINIT	E153H	(--)	9. 4	Portregister-Initialisierung
PVADMINSTR	E234H	(A0)	9. 3	PV-Verwaltung
KEYSCAN2	E243H	(--)	4. 4	Keyscan 2
AUPOWOFF	E33FH	(--)	5. 1	Auto Power Off

KEYIN	E41AH	(--)	4. 5	Tastentest
KEYSCAN1	E42CH	(--)	4. 4	Keyscan 1
IFBREAK	E451H	(A6)	4. 5	Break-Test
TXTEQU	E4A0H	(9E)	3. 4	Vergleichen zweier Texte
BTABSEAR1	E4A8H	(BE)	4. 2	BASIC-Tabellen suchen
BTABSEAR2	E4B7H	(BC)	4. 2	BASIC-Tabellen suchen
TIMER	E573H	(5A)	9. 3	Timer-IC Bedienung über PC
BEEPSET	E655H	(A2)	9. 3	Behandlung des BEEP-PC6
STDBEEP	E669H	(--)	9. 1	Standard-BEEP
BEEPUX	E66FH	(--)	9. 1	BEEP bezüglich U- und X-Register
WAIT64	E88CH	(AC)	9. 1	Warteschleife
PROGDISP	E8CAH	(--)	5. 3	Program Display
WROUT	EC5CH	(94)	8. 4	Schreiben in den Ausgabepuffer
STRDISP	ED00H	(92)	5. 1	String-Display
TOPDISP	ED3BH	(--)	5. 2	Ausgabe vom Displayanfang
CURMOVE	ED4DH	(--)	5. 1	Zeichen anzeigen und Cursor bewegen
DISP	ED57H	(--)	5. 1	Display
ASCIIDIS	ED5BH	(8A)	5. 4	Anzeige eines ASCII-Zeichens
HEXTEST	ED7DH	(--)	3. 1	Überprüfung von Hexzahlen
HEXA	ED95H	(--)	5. 2	Hexadezimal 2 Byte -> 1 Byte
LOALCPOI	EDABH	(90)	5. 3	Laden des Matrixpointer
INCLCPOI	EDB1H	(8E)	5. 3	Erhöhen des Matrixpointer
LCDSAVE	EDC1H	(--)	5. 5	Sicherung des Display-Inhalts
LCDLOAD	EDD8H	(--)	5. 5	Entsicherung des Display-Inhalts
GRAPHIC	EDEFH	(--)	5. 2	grafisches Display (ähnl. GPRINT)
GPRNT	EDF6H	(88)	5. 4	grafisches Display
COLCALC2	EE1FH	(8C)	5. 3	Berechnung von Spaltenadressen 2
COLCALC1	EE22H	(--)	5. 3	Berechnung von Spaltenadressen 1
LCDSEAR	EE48H	(--)	5. 4	Adresse für LCD-Zeichen suchen
CLRLCD	EE71H	(F2)	5. 3	Löschen der Anzeige
CURSAVE	EE80H	(--)	5. 5	Sicherung von Cursordaten und Display
CURLOAD	EE99H	(--)	5. 5	Entsicherung Cursordaten und Display
BLNKOFF	EF00H	(84)	5. 5	blinkenden Cursor ausschalten
-	EFB6H	(--)	2. 1	Subtraktion
+	EFBAH	(F0)	2. 1	Addition
*	F01AH	(7E)	2. 1	Multiplikation
1/X	F080H	(6E)	2. 1	Komplement
/	F084H	(58)	2. 1	Division
SQR	F0E9H	(--)	2. 1	Quadratwurzel
LN	F161H	(--)	2. 1	LN
LOG	F165H	(--)	2. 1	LOG
EXP	F1CBH	(--)	2. 1	e^x
10^X	F1D4H	(--)	2. 1	10^x
COS	F391H	(--)	2. 1	COS
TAN	F39EH	(--)	2. 1	TAN
SIN	F3A2H	(--)	2. 1	SIN
ACS	F492H	(--)	2. 1	ARCCOS
ATN	F496H	(--)	2. 1	ARCTAN
ASN	F49AH	(--)	2. 1	ARCSIN
DEG	F531H	(--)	2. 1	Umrechnung in GRAD-Form
DMS	F564H	(--)	2. 1	Umrechnung in Grad-Minuten-Sekunden
ABS	F597H	(--)	2. 1	Absolutwert
SGN	F59DH	(--)	2. 1	Vorzeichen
INT	F5BEH	(--)	2. 1	ganzzahliger Anteil
NEWRNDCD	F61BH	(5C)	2. 7	Neue Zufallszahl erzeugen
ARIFORMUS	F661H	(E8)	2. 6	Formieren von Rechenergebnissen
ARIFORMAS	F663H	(52)	2. 6	Formieren von Rechenergebnissen
ARIOPO1	F6B4H	(60)	2. 4	Kopieren X nach Y, Nullsetzen
ARISGNTST	F6E6H	(7C)	2. 4	Vorzeichentest zwischen X und Y
SGNX	F6FBH	(6C)	2. 4	Vorzeichen von X in Akku laden
COPYXS	F707H	(80)	2. 2	Kopieren X nach S
COPYXY	F70DH	(E6)	2. 2	Kopieren X nach Y
COPYSY	F715H	(68)	2. 2	Kopieren S nach Y
COPYXYPT	F729H	(82)	2. 4	X partiell nach Y kopieren
COPYXZPT	F72FH	(78)	2. 4	X partiell nach Z kopieren
COPYSX	F737H	(--)	2. 2	Kopieren S nach X

COPYXX	F73DH	(56)	2. 2	Kopieren Y nach X
STDCOPY	F741H	(--)	2. 1	Standardkopieren
CLEARPT	F747H	(70)	2. 4	Y partiell löschen
CLEARX	F757H	(EC)	2. 2	Löschen X
CLEARXPT	F75FH	(76)	2. 4	X partiell löschen
CLRMEM	F763H	(BA)	8. 2	Löschen von Speicherbereichen
ARISHFTR	F775H	(74)	2. 4	Nach rechts shiften
ARISHFTL	F79CH	(EA)	2. 4	nach links shiften
ARIRND	F7A7H	(5E)	2. 4	Laden des RND-Registers
ARISSET	F7B0H	(54)	2. 1	Arithmetikadressen setzen
CHANGXS	F7B5H	(64)	2. 2	Austauschen von X und S
CHANGXY	F7B9H	(66)	2. 2	Austauschen von X und Y
ARIADDYXPT	F7CCH	(EE)	2. 4	X+Y partiell
ARIADDXPT	F7CEH	(72)	2. 4	X+Reg. partiell
ARISUBPT	F7DDH	(7A)	2. 4	Y partiell von X subtrahieren
ARIOP04	F87FH	(--)	2. 4	Laden dezimal 0.9
ARIOP05	F883H	(--)	2. 4	Laden dezimal 90
ARIOP06	F887H	(--)	2. 4	Laden dezimal 180
ARIOP02	F88BH	(62)	2. 4	Laden dezimal 0.6
ARIOP03	F88FH	(6A)	2. 4	Laden dezimal 1.0
^	F89CH	(--)	2. 1	Power Raise
TOKENCR	F957H	(--)	9. 7	Erzeugen von BASIC-Tokens
OPNACCESS	FA74H	(3C)	9. 2	Zugriff auf BASIC-Tabellen
BASACCESS	FA89H	(1C)	9. 5	BASIC-Tabellen Zugriff
PVRESET	FB2AH	(1E)	8.12	Rücksetzen des PV-Bits
PROTOCOL	FB9DH	(3E)	9. 6	Protokollieren und Tracen
BUFCOPY	FBCBH	(--)	8. 3	Umkopieren zwischen Out- und Stringbuf

WWW.
PC-1500
.INFO

BASIC ERWEITERUNGEN VERSION 02.06 BUCHSTABENPOINTER:

Bst. Adr. Point to

A	8820	8856
B	8822	8871
C	8824	8893
D	8826	88E1
E	8828	8937
F	882A	8969
G	882C	89DC
H	882E	89E8
I	8830	8A17
J	8832	0000
K	8834	8A4B
L	8836	8A54
M	8838	8A6A
N	883A	8A97
O	883C	8AA1
P	883E	8AAC
Q	8840	0000
R	8842	8ACA
S	8844	8B11
T	8846	8B54
U	8848	8B68
V	884A	8B87
W	884C	8B9C
X	884E	8BB9
Y	8850	0000
Z	8852	8BC1

FREIGABEMITTEILUNG VERSION 3.04:

a) geänderte und korrigierte Befehle:

DATE\$ Jahresfunktion ergänzt
UPPER\$ }
LOWER\$ } behandeln ab sofort kompletten String
UPBK\$ }
INC } Variable muß definiert sein
DEC }
ZERO korrekte Grenzenberechnung
MIN } Werte jetzt korrekt
MAX }
FSAVE Sicherere Datenausgabe

b) neue Befehle:

YEAR Jahreseingabe
DAY TageskennzahlAusgabe
STD Tageskennzahleingabe

BASIC-ERWEITERUNGEN VERSION 02.06 TABELLE:

Adr.	L	Name	Tok.	Adr.	Opr.	byte	Bemerkung
8854	4	ACOS	E16B	877F	1010	A	Function
885D	4	ASIN	E16A	877F	1010	A	Function
8866	4	ATAN	E16C	877F	1011	B	Function
886F	3	BAS	E183	8783	1000	8	Procedure
8877	4	BIN\$	E172	80B1	1010	A	Function
8880	4	BOOT	E1A7	8784	1100	C	Procedure
8889	3	BYE	E1B7	82CA	1001	9	Procedure
8891	4	CASE	E18D	8784	1010	A	Procedure
889A	7	CATALOG	E1A9	8783	1000	8	Procedure
88A6	6	CHANGE	E1A2	8783	1000	8	Procedure
88B1	5	CHECK	E1A8	8783	1000	8	Procedure
88BB	6	CIRCLE	E1B2	8784	1110	E	Procedure
88C6	4	COPY	E182	8783	1000	8	Procedure
88CF	3	CSC	E166	877F	1010	A	Function
88D7	3	CRT	E16E	81FA	1011	B	Function
88DF	5	DATES	E15F	8039	1100	C	Function
88E9	3	DEC	E18C	8232	1100	C	Procedure
88F1	5	DEFFN	E197	8784	1100	C	Procedure
88FB	5	DEEK#	E179	8185	1010	A	Function
8905	4	DEEK	E178	8185	1010	A	Function
890E	6	DELETE	E1A0	8783	1000	8	Procedure
8919	4	DISP	E180	87BC	1100	C	Procedure
8922	5	DOKE#	E18A	8151	1100	C	Procedure
892C	4	DOKE	E189	8150	1101	D	Procedure
8935	5	ECONT	E188	8783	1000	8	Procedure
893F	7	ELLIPSE	E1B3	8784	1110	E	Procedure
894B	4	ENG\$	E176	877F	1100	C	Function
8954	4	ETEN	E16D	81F3	1010	A	Function
895D	5	EXIST	E17D	8787	1011	B	Function
8967	3	FAC	E17A	8438	1010	A	Function
896F	4	FAST	E1B0	8328	1010	A	Procedure
8978	6	FCHAIN	E192	85C4	1010	A	Procedure
8983	4	FIND	E1A1	8783	1000	8	Procedure
898C	5	FLOAD	E191	8480	1100	C	Procedure
8996	6	FMERGE	E193	84F3	1000	8	Procedure
89A1	2	FN	E198	8784	1100	C	Procedure
89A8	4	FRAC	E162	805B	1010	A	Function
89B1	5	FREE#	E157	877F	1100	C	Function
89BB	4	FREE	E158	81E0	1100	C	Function
89C4	5	FSAVE	E190	8363	1100	C	Procedure
89CE	7	FVERIFY	E194	860B	1101	D	Procedure
89DA	7	GETKEY\$	E15C	8000	1101	D	Function
89E6	4	HEX\$	E171	808B	1010	A	Function
89EF	6	HCLIST	E1B8	8784	1110	E	Procedure
89FA	4	HCOS	E168	877F	1010	A	Function
8A03	4	HSIN	E167	877F	1010	A	Function
8A0C	4	HTAN	E169	877F	1011	B	Function
8A15	3	INC	E188	822B	1100	C	Procedure
8A1D	8	INITLOOP	E195	8783	1000	8	Procedure
8A2A	4	INPT	E1A8	8784	1010	A	Procedure
8A33	6	INSTR\$	E170	877F	1100	C	Function
8A3E	6	INVERT	E19B	825E	1101	D	Procedure
8A49	4	KEEP	E1A4	8783	1001	9	Procedure
8A52	6	LABELS	E1AA	8783	1000	8	Procedure
8A5D	6	LOWER\$	E174	80E4	1001	9	Function
8A68	3	MAX	E164	832F	1100	C	Function

8A70	7	MIDSET\$	E175	877F	1100	C	Function
8A7C	3	MIN	E163	832F	1100	C	Function
8A84	3	MOD	E150	877F	1100	C	Function
8A8C	4	MOVE	E181	8783	1001	9	Procedure
8A95	5	NAME\$	E17F	877F	1011	B	Function
8A9F	6	OTHERS	E18E	8784	1011	B	Procedure
8AAA	3	POL	E17C	877F	1100	C	Function
8AB2	3	POP	E1B6	82BB	1010	A	Procedure
8ABA	2	PU	E199	CD89	1100	C	Procedure
8AC1	2	PV	E19A	CD89	1101	D	Procedure
8AC8	3	RCP	E16F	8084	1010	A	Function
8AD0	3	REC	E17B	877F	1100	C	Function
8AD8	8	RENUMBER	E1A3	8783	1000	8	Procedure
8AE5	6	REPEAT	E184	8784	1010	A	Procedure
8AF0	5	RESET	E1A5	8212	1000	8	Procedure
8AFA	6	RESULT	E17E	877F	1000	8	Function
8B05	5	ROUND	E160	877F	1101	D	Function
8B0F	3	SEC	E165	877F	1010	A	Function
8B17	6	SELEND	E18F	8784	1010	A	Procedure
8B22	4	SHOW	E196	8784	1100	C	Procedure
8B2B	7	SIMPSON	E19D	8784	1100	C	Procedure
8B37	4	SLOW	E1B1	831C	1010	A	Procedure
8B40	4	SORT	E19C	8784	1100	C	Procedure
8B49	4	SWAP	E1B5	86F0	1101	D	Procedure
8B52	5	TIMD\$	E15E	8013	1100	C	Function
8B5C	5	TOKEN	E161	877F	1001	9	Function
8B66	5	UNTIL	E185	8784	1010	A	Procedure
8B70	5	UPBK\$	E177	8100	1000	8	Function
8B7A	6	UPPER\$	E173	80CB	1001	9	Function
8B85	7	VARLIST	E1B9	8783	1000	8	Procedure
8B91	4	VER\$	E15D	81A0	1101	D	Function
8B9A	5	WHEND	E187	8784	1010	A	Procedure
8BA4	5	WHILE	E186	8784	1010	A	Procedure
8BAE	4	WIDE	E1B4	8784	1101	D	Procedure
8BB7	3	XOR	E151	877F	1101	D	Function
8BBF	4	ZERO	E1A6	82F6	1001	9	Procedure

WWW.
PC-1500
.INFO