SHARP

PC-1500

ROM-Unterprogramme I

H.-G. Schlieker

In monatelanger Kleinarbeit wurden die ROM-Routinen des PC-1500 untersucht und die Ergebnisse dann zu dieser Broschüre zusammengefaßt.
Die sinnvolle Nutzung setzt die Kenntnis von zumindest einer der folgenden Publikationen voraus:

1) SHARP Technical Reference Manual <u>oder</u>
2) Pocket Computer PC-1500 Systemhandbuch (deutsche Ausgabe des Technical Reference Manual), Vertrieb durch Fa. Holtkötter, Hamburg
3) PC-1500 Systemhandbuch; RVS-Datentechnik, Hallbergmoos

Trotz aller Sorgfalt bei der Analyse der Unterprogramme ist nicht auszuschließen, daß sich in dem vorliegenden Manuskript noch geringfügige Fehler befinden.
Für alle Hinweise, die eine Korrektur ermöglichen bin ich dankbar.

Ich hoffe, daß Ihnen die Kenntnis dieser Unterprogramme eine erhebliche Zeitersparnis bei der Programmierung einbringt und wünsche Ihnen dazu viel Erfolg.

Hans-Gerhard Schlieker

Bremen, im August 1983

Inhaltsverzeichnis

Unterprogramm	Seite
00	5
02	5
04	6
06	7
CA	8
0C	23
0E	9
10	10
12	10
14	10
16	8
18	33
1E	7
20	11
22	11
24	13
26	12
28	12
2A	13
30	25
32	25
34	14
36	15
38	5
3A	25
3C	7
42	37
48	16
4A	17
4C	16
4E	17
50	32
52	35
54	29
56	24
58	17
5A	18
5E	20
60	19
62	20
64	20
66	19
68	24
6A	19
6C	19
6E	18
70	21
72	21
74	36
76	36
78	36
7A	22
7C	23
7E	22
80	24
82	22
84	25

Unterprogramm	Seite
88	26
8A	26
8C	26
8E	29
90	26
92	27
94	27
96	25
9E	37
A0	28
A2	28
A6	28
AC	28
B2	28
BA	35
BC	29
BE	29
C0	30
C2	30
C4	15
C6	31
C8	31
CA	31
CC	32
CE	9
D2	32
D4	33
D6	34
D8	34
DA	35
DC	34
E6	24
E8	37
EA	36
EC	24
EE	37
F0	21
F2	20
F4	6
F6	6
EFB6	38
F0E9	38
F161	38
F165	38
F1CB	38
F1D4	38
F3A2	38
F39E	38
F49A	38
F496	38
F391	38
F492	38
F531	38
F564	38
F59D	38
F5BE	38
Versch.	37; 1

Erläuterung der verwendeten Fachbegriffe und Abkürzungen:

P1, P2, P3,...,Pn	Datenbytes, die an den Unterprogrammaufruf angefügt sind.
X-Reg, Y-Reg, U-Reg	CPU-Register X, Y, U
A	Akkumulator
AR-X, AR-Y, AR-Z, AR-U, AR-V, AR-W, AR-S	Arithmetikregister X, Y, Z, ...
XH, YH, UH	Höherwertiges Byte der CPU-Register X,Y,U
XL, YL, UL	Niederwertiges Byte " " "
78xx	Der durch "xx" gekennzeichnete Wert ist beliebig wählbar.
(789C)	Inhalt der Speicherzelle 789C
789C	Speicherzelle 789C
character string information	Im Arithmetikregister-X abgelegte Information über Adresse und Länge eines Strings bzw. einer numerischen Variablen. Statt einer Adresse kann dort auch eine 16-Bit-Binärzahl stehen. siehe hierzu: Sharp technical Reference Manual, S. 116

(00) DCB7 Syntaxprüfung. U-reg enthält ein Token oder ein Zeichen, daß vom Basic-Hauptspeicher geladen wurde. Es wird geprüft, ob der Wert in UL innerhalb eines durch die Parameter P 1 und P 2 festgelegten Intervalls liegt.
Steht in U-reg ein Token oder der Wert in UL liegt außerhalb des Intervalls, wird zu einer durch den Parameter P 3 festgelegten Disstanzadresse gesprungen. Liegt UL innerhalb der Intervallgrenzen, wird mit der auf P 3 folgenden Instruktion fortgesetzt.

Syntax des Aufrufes: CD 00 P1 P2 P3

Bedeutung der Parameter:

P1 = kleinster Wert innerhalb des Intervalls
P2 = kleinster Wert oberhalb des Intervalls
P3 = relativer Sprung um P3-Bytes, gerechnet ab der auf P3 folgenden Adresse

veränderte Register:

X = Adresse, zu der verzweigt wird
A

(02) DCB6 Syntaxprüfung wie (00) DCB7, jedoch zuvor Laden von Token oder Zeichen aus dem Basic-Hauptspeicher durch Unterprogramm (C0) DD08.

Eingangsparameter:

Das Y-Reg muß auf das nächste Zeichen im Basic-Hauptspeicher zeigen

Syntax des Aufrufes: CD 02 P1 P2 P3

Bedeutung der Parameter: siehe (00) DCB7

veränderte Register:

X = Adresse, zu der verzweigt wird
Y = zeigt auf das nächste Zeichen im Hauptspeicher
A

(38) CE9F Ermittelt Startadresse des Reservespeichers und lädt diese nach X-Reg. Das L-Byte der Reservespeicherstartadresse ist immer 08.
Die ersten 8 Bytes des Reservespeichers dienen als ROM-Status Information und werden bei Schreiboperationen im Reservespeicherbereich nicht beeinflußt.

Syntax des Aufrufes: CD 38

veränderte Register:

X-Reg enthält die Reservespeicherstartadresse
UH enthält 18
A

(04) DCC6 Syntaxprüfung. Wenn ein Basic-Programm abgearbeitet wird - die Pointer auf die aktuelle Zeilennummer (789C) und (789D) sind dann ungleich Ø, wird geprüft, ob durch das in UL geladene Zeichen, ein Ende der Befehlsfolge (3A) oder ein Zeilenende (OD) signalisiert wird.
Befindet sich der Rechner im Eingabemodus wird geprüft, ob der Wert in UL (OD) entspricht (Enter).
Ist das Prüfergebnis negativ, d.h. weder 3A noch OD sind in UL geladen, wird zu der durch P1 festgelegten Distanzadresse gesprungen, andernfalls wird mit der auf P1 folgenden Adresse fortgesetzt.

Syntax des Aufrufes: CD 04 P1

Bedeutung der Parameter:

P1 = relativer Sprung um P1-Bytes, gerechnet ab der auf P1 folgenden Adresse

veränderte Register:

X = Adresse zu der verzweigt wird
A
Carry-Flag ist gesetzt, falls ein Ende der Befehlsfolge (3A) erkannt wurde.

(F4) DBBC Der Inhalt, der durch die Parameter P1 und P2 gekennzeichneten Adresse, sowie der Inhalt der Folgeadresse wird als 16 Bit-Wert in das U-Reg geladen. Mit diesem Unterprogramm werden Adresspointer oder Daten aus der Memory-Map in die CPU eingelesen.

Syntax des Aufrufes: F4 P1 P2

P1 und P2 kennzeichnen eine 16 Bit-Adresse

veränderte Register:

U-Reg enthält den Inhalt der durch P1 und P2 festgelegten Speicherzellen.
A entspricht dem Inhalt von UH

(F6) DBB5 Der Inhalt des U-Reg wird in die durch die Parameter P1 und P2 festgelegten Adresse, sowie deren Folgeadresse übertragen. Dieses Programm ist die Umkehrung von (F4) DBBC. Der ursprüngliche Inhalt der durch P1 und P2 bestimmten Speicherzellen wird bei der Operation überschrieben.

Syntax des Aufrufes: F6 P1 P2

P1 und P2 müssen einen Adressbereich im RAM definieren

veränderte Register:

X-Reg enthält Adresse der zuletzt beschriebenen Speicherzelle.
A enthält den Wert von UL

(06) D065 Der Basic-Stack-Pointer aus 7882 wird nach XL geladen; XH wird mit 7A geladen. X-Reg zeigt nun auf eine Speicherstelle im Basic-Stack. Der Inhalt der durch X-Reg adressierten Speicherstelle wird nach UH, der Inhalt der Folgeadresse nach UL geladen. U-Reg enthält jetzt eine auf dem Basic-Stack abgelegte Adresse.
X-Reg wurde um 2 inkrementiert. XL wird als neuer Pointer auf den Basic-Stack nach (7882) zurückgeladen.

Syntax des Aufrufes: CD 06

veränderte Register:

X-Reg zeigt auf nächste Adresse des Basic-Stack.
U-Reg enthält die aus dem Basic-Stack entnommene Adresse
A enthält neuen Basic-Stack-Pointer
(7882) wurde mit dem Wert von XL aktualisiert.

(1E) FB2A Der Akkumulator enthält bei Aufruf des Unterprogrammes einen bestimmten Wert. Bit Ø dieses Wertes bestimmt, ob das PV-Banking durchgeführt werden soll.
Ist Bit Ø = 1, wird das PV-Banking aktiviert, andernfalls reaktiviert. Speicherstelle (79DO) wird entsprechend dem Inhalt von Bit Ø des Akku verändert.

Syntax des Aufrufes: CD 1E

veränderte Register:

A enthält FE

(3C) FA74 Ermittelt Token-Tabelle. Bei Einsprung enthält der Akku einen bestimmten Wert (in der Regel ist dies der Inhalt von (79D1)="OPN"). Dieser Wert wird verdoppelt (durch linksschieben) und nach XH geladen. Dadurch wird die Speicherseite, bei nach vorhandener Tokentabelle gesucht werden soll selektiert.
Es wird geprüft, ob an der Stelle (XH ØØ) der Wert 55 vorhanden ist. 55 kennzeichnet den Beginn einer Tokentabelle in diesem Bereich. Wurde 55 nicht erkannt, wird durch PV-Banking dieselbe Prüfung im PV-Bereich durchgeführt. Wurde eine Token-Tabelle gefunden, steht in X-Reg die Adresse XH55.
Carry-Flag ist dann gesetzt.

Syntax des Aufrufes: CD 3C

veränderte Register:

X-Reg zeigt ggfls. auf Beginn der Tokentabelle
A

(OA) DE5E AR-X enthält eine character string information. Diese wird in die CPU-Reg X (Adresse bzw. Variablenname) und Akku (Länge) eingelesen.
Wenn der Wert in XH größer 80 ist, wird signalisiert, daß im X-Reg der codierte Variablenname einer Variablen steht, für die noch kein Speicherraum im Variablenspeicher reserviert ist. (Bit 7 des 1. Bytes des Variablennamen ist gesetzt).
Andernfalls enthält X-Reg die Adresse der Variablen.

Wenn im X-Reg die Adresse einer numerischen Variablen steht (Bit 7 der Länge im Akku ist dann gesetzt), wird der Inhalt der Variablen in das AR-X eingelesen und Carry-Flag=Ø gesetzt.
Wenn im X/Reg die Adresse eines Strings steht, wird die Anzahl der Zeichen des Strings ermittelt. Das Ende einer Zeichenkette innerhalb eines String wird daran erkannt, daß eine Speicherstelle den Wert ØØ besitzt. Die Anfangsadresse des Strings, sowie die Anzahl der Zeichen die der String umfaßt, wird als neue character string information in das AR-X zurückgeladen.

Falls im X-Reg ein Variablenname steht, für den noch kein Speicherraum reserviert wurde (B_7 in XH gesetzt) so wird bei einer Stringvariablen die Information über die Länge des Strings (7A07=0) gesetzt.
Bei einer numerischen Variablen wird geprüft, ob der Variablenname zulässig ist. Ergibt die Überprüfung die Zulässigkeit des Variablennamens, so wird der Inhalt von AR-X gelöscht. (XH ist kleiner DB)
Ist der im X-Reg stehende Variablenname oberhalb des zulässigen Wertebereiches (größer DA), so verzweigt (OA) in die Subroutine TIME, d.h. zu Adresse DE82. Hierdurch wird der Inhalt des Timerbausteines als numerischer Wert in das AR-X eingelesen.

Syntax des Aufrufes: CD OA

Eingangsparameter: Das AR-X darf bei Aufruf des Unterprogramms keinen numerischen Wert enthalten, sondern muß mit einer Adresse, bzw. dem codierten Namen einer Variablen, für die noch kein Speicherraum reserviert wurde, geladen sein. Andernfalls ist das Ergebnis im AR-X nicht vorhersehbar.

veränderte Register:

nur der Inhalt von Y-Reg bleibt erhalten

(16) DFF5 Ermittelt die Differenz zwischen dem 16 Bit-Wert im X-Reg und dem Ende des Programmspeichers.
Das Ergebnis entspricht, wenn im X-Reg die Startadresse des Programmspeichers stand, dem Wert für STATUS 1 minus 2.

Syntax des Aufrufes: CD 16

veränderte Register:

U-Reg enthält das Rechenergebnis, hexadezimal
A

(OE) D461 Ermittelt die Adresse einer Variablen, deren codierter Variablenname im U-Reg steht.

Syntax des Aufrufes: CD OE P1 P2

Bedeutung der Parameter:

P1 legt den für die Operation zulässigen Variablentyp fest. Dadurch wird verhindert, daß für Stringoperationen numerische Variable, bzw. für numerische Operationen Stringvariable verwendet werden.
Folgende Werte von P1 sind möglich; bzw. sinnvoll:

58 Alle Variablentypen sind zugelassen
68 Nur numerische Variable sind zugelassen
40 Nur Stringvariable

P2 relativer Sprung um P2 Bytes, falls während der Adressermittlung ein Fehler festgestellt wird, gerechnet ab der auf P2 folgenden Adresse.

Eingangsparameter:

U-Reg muß den codierter Variablennamen enthalten.
Y-Reg muß auf einen Adressbereich zeigen, der die gesuchten Indizes enthalten muß; zeigt auf 1.Index.

Ausgangsparameter:

Falls ein Fehler festgestellt wird, wird ERROR-Code nach UH geladen.

AR-X enthält nach fehlerfreiem Lauf des Unterprogramms character string information über die Anfangsadresse der Variablen sowie deren Länge.
Die Parameter (7881), (7889) und (788C) sind verändert.

veränderte Register:

X-Reg enthält die Rücksprungadresse in das rufende Programm, bei Fehler ist dies die durch P2 festgelegte Distanzadresse.
Y-Reg zeigt auf das nach den durch ")" abgeschlossenen Indiezes folgende Zeichen.
U-Reg enthält ggfls. Fehlercode oder Adresse der gesuchten Variablen
A

(CE) D45D Ermittelt Adresse einer Variablen, deren Variablenname in einem Adressbereich steht, auf den das Y-Reg zeigt. Dieses Unterprogramm unterscheidet sich von (OE) dadurch, daß zusätzlich noch der Variablenname ermittelt wird.

Syntax des Aufrufes: CE P1 P2

Alle Parameter und Veränderungen entsprechen denen in Unterprogramm (OE) beschriebenen.

(10) DD2D U-Reg enthält eine 16 Bit-Zahl. In Abhängigkeit vom Parameter P1 werden folgende Operationen durchgeführt:

ØØ - Die Integerzahl aus U wird im Arithmetikregister X als numerische Zahl im BCD-Format abgespeichert

4Ø - Die Integerzahl aus U wird, beginnend ab der Speicherstelle, auf die das Y-Reg bei Einsprung zeigt, als Ziffernfolge im ASCII-Format abgespeichert. Bei Aussprung zeigt Y-Reg auf die nächste Speicherstelle hinter der der übertragenen Ziffernfolge.

8Ø - U wird im AR-X als numerische Zahl gespeichert, wobei der Bereich 0000-7FFF als positive Werte (7A01 = ØØ) und der Bereich 8ØØØ-FFFF als negative, komplementäre Werte übernommen werden.

EØ - Wirkung wie bei 4Ø, die Ausgabe ist jedoch auf drei Stellen beschränkt, wobei die führende Stelle ein Space ist. Mit diesem Parameter lassen sich nur Binärzahlen im Bereich o - 99 transformieren.

Syntax des Aufrufes: CD 10 P1

veränderte Register:

Das AR-X wird in jedem Falle verändert
ebenso die übrigen CPU-Register
Ausnahme: Bei P1 = ØØ bleibt das Y-Reg unverändert

(12) DF93 Ermittelt Startadresse des Programmspeichers und lädt X-Reg damit.

Syntax des Aufrufes: CD 12

veränderte Register:

X-Reg enthält die Programmspeicherstartadresse
A enthält das H-Byte der Startadresse

(14) DFFA Ermittelt Startadresse des Programmspeichers, lädt das X-Reg damit; sowie die Länge des dort abgelegten Programmes.

Bei Aussprung aus dem Programm steht die Startadresse im X-Reg sowie die Länge in Bytes im U-Reg.

Syntax des Aufrufes: CD 14

veränderte Register:

X, U, A

(22) DF63 Y-Reg zeigt auf den Basic-Hauptspeicher. Die nächste Instruktion wird in U-Reg geladen und Y-Reg entsprechend inkrementiert. Wenn das nach UL geladene Zeichen nicht auf den Beginn eines im Hauptspeicher stehenden Strings hindeutet ("), wird das Carry-Flag zurückgesetzt und es erfolgt ein Rücksprung.
Folgt im Hauptspeicher ein String, so wird dessen Länge ermittelt und die Startadresse sowie die Länge werden in das AR-X als character string information geladen.
In diesem Fall ist das Carry-Flag=1.
Anschließend wird die nächste Instruktion nach U-Reg geladen.

Syntax des Aufrufes: CD 22

Eingangsparameter:

Y-Reg muß auf den Basic-Hauptspeicher zeigen

Ausgangsparameter:

Carry-Flag=1 AR-X enthält character string information über einen String im Hauptspeicher

Carry-Flag=Ø es wurde kein String erkannt

veränderte Register:

X-Reg
Y-Reg zeigt auf die nächste Speicherzelle im Basic-Hauptspeicher
U-Reg enthält das gelesene Zeichen oder Token

AR-X enthält ggfls. character string information, siehe Carry-Flag

(20) DF72 Y-Reg zeigt auf den Basic-Hauptspeicher und und wird soweit inkrementiert, bis es auf das ~~Low~~- HIGH Byte der nächsten Zeilennummer zeigt, jedoch höchstens um 77 Bytes.
Syntax des Aufrufes: CD 20

Ausgangsparameter:

Das Zero-Flag ist gesetzt, wenn die Basic-Zeile innerhalb der zulässigen Länge ist, andernfalls ist Zero-Flag=Ø

Geänderte Register:

X-Reg und Y-Reg enthalten die Adresse der nächsten Zeilennummer
U-Reg
A

Hinweis: Dieses Unterprogramm wird bei der REM-Anweisung benutzt. Das um maximal 77 Bytes inkrementiert wird, liegt daran, daß in der Basic Zeile 3 Bytes für die Zeilennummer und die Länge verbraucht werden und dieses Unterprogramm bei Aufruf nie auf diese ersten 3 Bytes zeigt.

(26) DB87 Es wird geprüft, ob im AR-X eine "character string information" abgelegt ist, oder ob dort eine Zahl im BCD-Format vorhanden ist.
Wenn eine Zahl im BCD-Format vorliegt, ist der Inhalt der Speicherzelle (7A04) kleiner als B2.
Liegt ein BCD-Format vor, wird UH mit 11 geladen (ERROR 17) und ein relativer Sprung zu der durch P1 festgelegten Distanzadresse durchgeführt. In diesem Fall ist das Carry-Flag=Ø.
Ist im AR-X eine character string information enthalten, wird Carry-Flag=1 gesetzt und mit der auf P1 folgenden Instruktion fortgesetzt.

Bedeutung der Parameter:

P1=relativer Sprung um P1-Bytes gerechnet ab der auf P1 folgenden Adresse

Syntax des Aufrufes: CD 26 P1

veränderte Register:

X-Reg enthält die Adresse mit der das Programm fortgesetzt wird.
UH enthält ggfls. ERROR-Code
A
Carry-Flag s.o.

(28) DBB1 Speicherzelle (788C) enthält die Information, ob eine verarbeitete Variable ein- oder zweidimensional dimensioniert war.
Das Unterprogramm prüft den Inhalt dieser Speicherzelle und lädt 12 nach UH (ERROR 18) falls dort ein Wert ≠ 1 vorhanden ist. Anschließend wird mit der durch P1 festgelegten Distanzadresse fortgesetzt.
Steht in (788C) eine 1, wird mit der auf P1 folgenden Adresse fortgesetzt. (Carry-Flag=1)

Syntax des Aufrufes: CD 28 P1

Bedeutung von P1:

relativer Sprung um P1 Bytes, gerechnet ab der auf P1 folgenden Adresse

veränderte Register:

X-Reg enthält die Adresse mit der das Programm fortgesetzt wird.
UH enthält ggfls. ERROR-Code
A=01

(2A) D03E Transfer von Systemmeldungen aus der Speicherseite C3xx.

Syntax des Aufrufes: CD 2A P1 P2

Eingangsparameter:

P1 legt das Low-Byte in der Speicherseite C3xx fest, ab der die zu übertragende Information beginnt.

P2 bezeichnet die Anzahl der zu übertragenden Bytes

Y-Reg muß auf die 1. Speicherstelle zeigen, die die Systemmeldung aufnehmen soll.

Ausgangsparameter:

Y-Reg zeigt auf die nächste freie Stelle nach der übertragenen Systemmeldung

P1	P2	Systemmeldung
5B	09	_BREAK_IN
64	06	_ERROR
61	03 04	_IN

Nach Durchführung des Programmes wird mit der auf P2 folgenden Instruktion fortgesetzt.

veränderte Register:

X, Y, U, A

(24) DEAF X-Reg enthält die 16 Bit-Adresse eines Strings, A die Länge dieses String. Es wird eine character string information in das AR-X geladen.

Syntax des Aufrufes: CD 24

Inhalt des AR-X nach der Operation:

7A00	unverändert
7A01	"
7A02	"
7A03	"
7A04	D0
7A05	H-Byte der Adresse aus X-Reg
7A06	L-Byte der Adresse aus X-Reg
7A07	Länge des Strings aus Akku

veränderte Register:

U-Reg enthält Pointer auf 7A04
A=D0

(34) DF23 Syntaxprüfung und Mehrfachverzweigung. Eine angefügte Tabelle wird nach dem im Akkumulator befindlichen Zeichen durchsucht.
Ist das Zeichen nicht in der Tabelle enthalten, wird das Programm mit der auf die Tabelle folgenden Instruktion fortgesetzt.
War die Suche erfolgreich, d.h., das Zeichen ist in der Tabelle enthalten, wird ein relativer Sprung berechnet der sich aus dem Parameter hinter dem gefundenen Zeichen in der Tabelle ergibt und die Programmausführung wird an der errechneten Adresse fortgesetzt.

Syntax des Aufrufes:

```
CD 34 P1
P2 P3
P4 P5
P6 P7
.. ..
.. ..
.. ..
.. Pn
```

Bedeutung der Parameter:

P1 legt die Anzahl der Zeichentests fest. Damit wird somit auch der Umfang der angefügten Tabelle definiert.
Beispiel: Ist P1 = Ø4; dann enthält die Tabelle 5 Zeichen, die auf Übereinstimmung mit dem Akku getestet werden. Die Anzahl der Datenbytes der Tabelle beträgt in diesem Fall 11.

P2, P4, P6, .., ..: Zeichen (ASCII-Codes), die auf Übereinstimmung mit dem Akku geprüft werden.

P3, P5, P7, .., Pn: relativer Sprung um Px-Bytes, gerechnet ab der diesem Parameter folgenden Adresse.

Beispiel: Wurde festgestellt, daß P4 dem bei Einsprung im Akku befindlichen Zeichen entspricht, dann wird P5 in den Akku geladen. X-Reg zeigt dann auf die Adresse, die dem Datenbyte P5 folgt. Zu der Adresse im X-Reg wird der Akku addiert (X-Reg + P5). Das Programm verzweigt dann zu der Adresse, die im X-Reg steht.

Besonderheit: War bei Einsprung der Wert in UH ≠ ØØ, dann ist bei Aussprung aus dem Unterprogramm Zero-Flag=Ø; war UH=ØØ ist Zero-Flag=1. Sollte sich also im U-Reg bei Aufruf des Unterprogrammes ein Token befunden haben (der Zeichentest wäre dann unsinnig!), dann ist Zero-Flag=Ø.

veränderte Register:
X-Reg enthält die Adresse, zu der verzweigt wird.
Y-Reg unverändert
U-Reg unverändert
A enthält bei Aussprung den Wert von UH; der ursprüngliche Inhalt des Akku geht verloren.
Zero-Flag ist in Abhängigkeit von UH gesetzt, bzw. zurückgesetzt.

(36) DFOF Y-Reg zeigt auf eine Adresse im Input-Buffer oder Basic-Hauptspeicher. Es wird geprüft, ob dort ein Variablenname oder eine Textkonstante abgelegt ist. Sofern weder der Name einer Textvariablen, noch eine Textkonstante erkennbar ist, wird das Carry-Flag auf Ø gesetzt und es erfolgt Rücksprung.

A Textkonstante wurde gefunden:
Die Länge der Textkonstante wird ermittelt und im AR-X als character string information abgelegt.

B Name einer Textvariablen wurde gefunden:
Die Adresse der Textvariablen wird ermittelt und im AR-X als character string information abgelegt.

In beiden Fällen wird das Carry-Flag auf 1 gesetzt.

Syntax des Aufrufes: CD 36

Eingangsparameter: Y-Reg muß bei Einsprung auf eine Adresse im Input-Buffer oder Basic-Hauptspeicher zeigen.

Ausgangsparameter: Das Carry-Flag ist 1, wenn ein ein String, bzw. dessen Startadresse im AR-X als character string information abgelegt wurde. Sofern Carry-Flag Ø ist, kann ggfls. in UH eine Fehlermeldung enthalten sein. Dies ist dann der Fall, wenn ein Variablenname verarbeitet wurde, für den keine Adresse ermittelt werden konnte.

veränderte Register:

X-Reg enthält bei Rückkehr die Adresse, die bei Einsprung im Y-Reg stand.
Y-Reg wird inkrementiert und zeigt auf die nächste Instruktion im Input-Buffer bzw. Basic-Hauptspeicher.
U-Reg enthält ggfls. Fehlermeldung oder Pointer auf Arithmetikregister.
A

(C4) DCD5 Syntaxprüfung. Entspricht Unterprogramm (C2), jedoch ohne Laden des nächsten Zeichen/Token. Für die Bedeutung der Parameter P1, P2, P3 siehe dort angegebene Erläuterung.

Syntax des Aufrufes: C4

veränderte Register:

siehe (C4) DCD6
Y-Reg wird nicht inkrementiert, U-Reg bleibt unverändert.

(48) DCF9 Ein Hauptprogramm hat ein Unterprogramm aufgerufen. Da ein bestimmtes Prozessorflag einen bestimmten Zustand hat, wird aus dem laufenden Unterprogramm in das rufende Hauptprogramm zurückgekehrt. Dem Unterprogrammaufruf sind ein oder mehrere Datenbytes angefügt. Auf dem Stack befindet sich die Adresse des Hauptprogramms, die auf das letzte dieser angefügten Datenbytes zeigt. Dieses Datenbyte ist eine Distanzadresse. Aus der Adresse dieses Datenbytes + 1 und dem Datenbyte wird die absolute Rücksprungadresse ermittelt und die Bearbeitung an der errechneten Adresse fortgesetzt.

Syntax des Aufrufes: CD 48 (in der Regel erfolgt Aufruf nicht mit "CD" sondern in Abhängigkeit vom Zustand eines bestimmten Prozessorflags.)

Eingangsparameter: Der Stack muß eine Adresse des Hauptprogramms (in der die Distanzadresse steht) enthalten.

veränderte Register:

X-Reg enthält die Adresse, an der die Programmausführung im Hauptprogramm fortgesetzt wird.
Y-Reg unverändert
U-Reg "
A enthält die geladene Distanzadresse.

(4C) DCE9 Aus einem laufenden Unterprogramm wird in das rufende Hauptprogramm zurückgekehrt. Auf dem Stack liegt die Adresse des Hauptprogrammes, die auf das letzte Datenbyte des aufgerufenen Unterprogrammes zeigt. Diese Adresse wird nach X-Reg geladen und inkrementiert. X-Reg zeigt jetzt auf die Adresse nach dem letzten Datenbyte des Unterprogrammaufrufes. Es erfolgt ein Rücksprung zu dieser Adresse und Fortsetzung des Hauptprogrammes.

Syntax des Aufrufes: CD 4C

Eingangsparameter: Der Stack muß eine Adresse des Hauptprogrammes enthalten, die auf das letzte Datenbyte des Unterprogrammaufrufes zeigt.

veränderte Register:

X-Reg enthält die Adresse, an der das Programm fortsetzt
Y-Reg unverändert
U-Reg "
A "

(4A) DCFD Rücksprung aus einem laufenden Unterprogramm analog zu (48) DCF9, allerdings mit folgendem Unterschied: Das laufende Unterprogramm hat während der Bearbeitung den Inhalt des Y-Reg auf den Stack gerettet. Vor Rücksprung in das Hauptprogramm wird der ursprüngliche Inhalt des Y-Reg wiederhergestellt. Entsprechend dem Datenbyte im Hauptprogramm wird zu einer Distanzadresse zurückgesprungen.

Syntax des Aufrufes: CD 4C

Eingangsparameter: Der Stack muß die Adresse des Hauptprogrammes, an der das letzte Datenbyte nach Unterprogrammaufruf steht und gerettetes Y-Reg enthalten.

veränderte Register:

X-Reg enthält die Rücksprungadresse
Y-Reg enthält die vom Unterprogramm auf den Stack gerettete Adresse
U-Reg unverändert
A enthält das geladene DatenByte

(4E) DCED Rücksprung aus dem laufenden Unterprogramm analog zu (4C) DCE9. Wie bei Unterprogramm (4A) DCFD wird zuvor der Inhalt des Y-Reg wieder vom Stack geholt. Rücksprung in das Hauptprogramm erfolgt zu der dem letzten Datenbyte nach Unterprogrammaufruf folgenden Adresse.

Syntax des Aufrufes: CD 4E

Eingangsparameter: wie (4A)

veränderte Register:

X-Reg enthält Rücksprungsadresse
Y-Reg enthält den,vom Unterprogramm geretteten Inhalt
U-Reg unverändert
A "

(58) F084 Division. AR-X = AR-X / AR-Y
Der numerische Wert im AR-X wird durch den numerischen wert im AR-Y dividiert und das Ergebnis wieder im AR-X abgelegt. Falls durch Ø dividiert wird enthält UH entsprechenden Fehlercode.

Syntax des Aufrufes: CD 58

Eingangsparameter: AR-X und AR-Y müssen numerischen Wert enthalten.

veränderte Register:

X-Reg enthält 7A00
Y-Reg enthält 7A08
U-Reg
A

(5A) E573 Timer-modus-select Routine. Der Timerbaustein µPD1990AC wird über seine Eingänge Co, Cl, C2, die über I/O PORT PC3, PC4, PC5 gespeist werden, in seine verschiedenen Betriebsarten versetzt.

Syntax des Aufrufes: CD 5A

Eingangsparameter:

Der Wert im Akku ist entsprechend u.a. Tabelle vorzugeben. Timer modus select erfolgt dann entsprechend dem Zustand von Bit 3, 4 und 5 des Akku.

C2	Cl	CO	Modus	Wirkung
0	0	0	0	Register hold
0	0	1	1	Register shift
0	1	0	2	Time set
0	1	1	3	Time read
B5	B4	B3		

des Akku

veränderte Register:

U-Reg
Y-Reg enthält F008
X-Reg bleibt erhalten
A

(6E) F080 AR-X = 1 / AR-X oder AR-X = AR-X $^{-1}$
Der Inhalt des AR-X wird nach AR-Y übertragen. Anschließend wird AR-X mit dem numerischen Wert 1 geladen, AR-X wird durch AR-Y dividiert und das Ergebnis in AR-X abgespeichert.

Syntax des Aufrufes: CD 6E

Eingangsparameter: AR-X muß einen numerischen Wert enthalten. Die CPU-Reg X u. Y müssen auf die Speicherseite 7Axx zeigen.

Ausgangsparameter: Wenn AR-X Ø enthält, wird der Fehlercode 26 (ERROR 38) in UH abgelegt und Carry-Flag gesetzt. Bei normaler Beendigung ist Carry-Flagg=Ø

veränderte Register:

AR-X enthält das Ergebnis der Operation
AR-Y enthält den ursprünglichen Wert des AR-X vor der Operation.
X-Reg enthält 7A00
Y-Reg enthält 7A08
AR-Z verändert
U-Reg enthält 0005
A

(60) F6B4 Der numerische Wert im AR-X wird aufgeteilt und zwar so, daß die Vorkommastellen im AR-X stehen und die Nachkommastellen im AR-Y. Die ursprüngliche Position der Nachkommastellen im AR-X bleibt im AR-Y erhalten. Der Exponent wird aus (7A00) nach (7A10) übernommen, wenn der ganzzahlige Anteil aus AR-X 10 Stellen nicht übersteigt.
Das Vorzeichen aus (7A01) wird nicht nach (7A11) übernommen.

Syntax des Aufrufes: CD 60

Eingangsparameter: XH und YH müssen bei Aufruf 7A enthalten. Im Arithmetikregister X muß ein numerischer Wert enthalten sein.

veränderte Register:

X-Reg= 7A08, Y-Reg=7A18, U-Reg, A=Exponent

(6A) F88F Schreibt in AR-Y den numerischen Wert 1 und setzt Carry-Flag.

Syntax des Aufrufes: CD 6A

Eingangsparameter: XH und YH müssen bei Einsprung 7A enthalten

veränderte Register:

X-Reg enthält 7AEF
Y-Reg enthält 7A13
U-Reg UH unverändert, UL enthält FF
A enthält EC

(66) F7B9 Vertauscht AR-X mit AR-Y.

Syntax des Aufrufes: CD 66

Eingangsparameter: XH und YH müssen bei Aufruf 7A enthalten.

veränderte Register:

X-Reg enthält 7A08
Y-Reg " 7A18
U-Reg
A

(6C) F6FB Wenn X-Reg bei Einsprung auf die Speicherseite 7Axx zeigt, wird das Vorzeichen des numerischen Wertes im AR-X in den Akku geladen. Anschließend wird im AR-X der Absolutwert hergestellt.

Syntax des Aufrufes: CD 6C

veränderte Register:

XL = 01; A = Vorzeichen

(62) F88B Löscht AR-Y. Anschließend wird der numerische Wert o,6 dorthin übertragen.

Syntax des Aufrufes: CD 62

Inhalt von AR-Y: 7A10 18
FF 00 60 00 00 00 00 00

Eingangsparameter:

XH und YH müssen bei Einsprung 7A enthalten

veränderte Register:

X-Reg enthält 7AF8
YL " 13
UL " FF
A " F5
Bei Aussprung ist Carry-Flag gesetzt.

(5E) F7A7 Überträgt Zufallszahl, die im Bereich (7B01 - 7B07) abgelegt ist, in das AR-X.

Syntax des Aufrufes: CD 5E

Eingangsparameter: YH muß bei Aufruf 7A enthalten.

veränderte Register:

X-Reg enthält 7A08
y-Reg " "
UL

(64) F7B5 Vertauscht AR-X mit AR-S.

Syntax des Aufrufes: CD 64

Eingangsparameter: XH und YH müssen bei Aufruf 7A enthalten.

veränderte Register:

X-Reg enthält 7A08
Y-Reg " 7A38
U-Reg, A

(F2) EE71 Löscht LCD-Anzeige. Entspricht dem Basic Befehl CLS.

Syntax des Aufrufes: F2

veränderte Register:

U-Reg ent hält 77FF
A
Die System-Ram-Bereiche 7600-764D und 7700-774D werden mit ØØ überschrieben.

(70) F747 Löscht 7 Bytes ab Startadresse (XH 11). Wenn X-Reg auf die Speicherseite 7Axx zeigt, werden Vorzeichen und Mantisse des AR-Y gelöscht. Der Exponent bleibt erhalten.

Syntax des Aufrufes: CD 70

veränderte Register:

X-Reg enthält 7A18
UL enthält FF
A enthält ØØ

(72) F7CE Dezimale Addition. Der Inhalt einer numerischen Variablen (8-Bytes)bzw. eines Arithmetikregisters, auf dessen letztes Byte das Y-Reg zeigt, wird zum Inhalt des AR-X hinzuaddiert und das Ergebnis dort abgelegt. Es werden nur Vorzeichen u. Mantisse addiert.

Syntax des Aufrufes: CD 72

Eingangsparameter: X-Reg muß auf die Speicherseite 7Axx zeigen. Y-Reg ist frei adressierbar.

veränderte Register:
X-Reg und Y-Reg zeigen bei Aussprung auf das 1. Byte (Exponent) der addierten Dezimalzahlen.
U-Reg
A
Hinweis: Dieses Unterprogramm prüft nicht, ob es während der Durchführung zu einem Übertrag gekommen ist. Dgl. wird auch kein Vorzeichentest durchgeführt.

(FO) EFBA Addition. AR-X= AR-X + AR-Y

Syntax des Aufrufes: FO

Eingangsparameter:

AR-X und AR-Y müssen bei Aufruf numerische Werte enthalten.

Ausgangsparameter:

Das Ergebnis ist in AR-X abgespeichert, ggfls. ERROR-code in UH.

veränderte Register:

X-Reg = 7AO8
Y-Reg = 7AO8
U-Reg
A

(7A) F7DD Subtraktion der Mantissen zweier Dezimalzahlen die im AR-X und AR-Y als numerischer Wert gespeichert sind. GGfls kann ein Übertrag beim Vorzeichen in (7A01) auftreten.
Der Exponent bleibt unberücksichtigt.

Syntax des Aufrufes: CD ~~7C~~ 7A

Eingangsparameter: XH und YH müssen bei Aufruf 7A enthalten.

veränderte Register:

X-Reg zeigt auf 7A00
Y-Reg " " 7A10
U-Reg
A enthält Inhalt der Speicherstelle (7A01) - Vorzeichen
Tritt ein Übertrag im Vorzeichen-Byte auf, ist Carry-Flag Ø; sind die Mantissen gleich, bzw. Mantisse im AR-X ist größer als die im AR-Y, ist Carry-Flag=1

(7E) F6E6 Multiplikation. AR-X = AR-X x AR-Y
Die numerischen Werte in AR-X und AR-Y werden multipliziert und das Ergebnis im AR-X abgespeichert.

Syntax des Aufrufes: CD 7E

Eingangsparameter: AR-X und AR-Y müssen zwei numerische Werte enthalten.

veränderte Register:

X-Reg zeigt auf 7A00
Y-Reg zeigt auf 7A09
U-Reg enthält ggfls. ERROR-Code in UH
A
Kommt es zu einem Ergebnis außerhalb des zulässigen Wertebereiches, ist Overflow-Flag gesetzt.
AR-W, AR-V,AR-U, AR-Z werden verändert.

(82) F729 Vorzeichen und Mantisse des numerischen Wertes im AR-X werden in das AR-Y übertragen.

Syntax des Aufrufes: CD 82

Eingangsparameter: XH und YH müssen bei Aufruf 7A enthalten.

veränderte Register:

X-Reg zeigt auf 7A08
Y-Reg " " 7A18
U-Reg

(7C) F6E6 Vorzeichentest für mathematische Operationen.
Die Vorzeichen der numerischen Werte der AR-X und AR-Y werden mit einer Exklusiv-Oder-Verknüpfung verglichen. Das Ergebnis ist ØØ falls beide Vorzeichen positiv oder negativ sind, bzw. 80 falls die Vorzeichen ungleich sind.
Die Rücksprungadresse wird von Stack geholt und in U-Reg zwischengespeichert. Anschließend wird das Vorzeichen des AR-Y auf den Stack gebracht, sowie das Ergebnis der Vorzeichenverknüpfung aus dem Akku.
Die Vorzeichen der beiden Arithmetikregister werden auf ØØ gesetzt (d.h. positiv). Die Rücksprungadresse wird wieder auf den Stack geladen und es erfolgt Rücksprung.

Syntax des Aufrufes: CD 7C

Eingangsparameter: Bei Aufruf müssen XH u. YH 7A enthalten.

veränderte Register:

X-Reg enthält 7A00
Y-Reg " 7A10
U-Reg
A enthält das Ergebnis der Exklusiv-Oder-Verknüpfung der beiden Vorzeichen.
Stack s.o.

(0C) DE97 Die Länge eines Strings, auf dessen 1.Zeichen das Y-Reg bei Einsprung in das Unterprogramm zeigt wird ermittelt (in UH). Anschließend wird AR-X mit "character string information" geladen (siehe(24)).

AR-X hat bei Ende des Programms folgenden Inhalt:

7A00 unverändert
7A01 "
7A02 "
7A03 "
7A04 D0 ;kennzeichnet die folgende Adresse als String
7A05 H-Byte der Adr. des 1. Byte des Strings
7A06 L-Byte "
7A07 Länge, die in UH ermittelt wurde

veränderte Register:

X enthält den ursprünglichen Wert von Y-Reg vor Durchführung des Unterprogramms (pointer auf das 1.Byte des Strings)
Y zeigt auf das nächste, dem String folgende Zeichen
U zeigt auf 7A04
A

Hinweis: Als Information für das Ende des Strings wird vom Programm sowohl 22, d.h. " als auch 0D, d.h. Zeilenende oder Enter anerkannt. Das zweite Anführungszeichen braucht bei Zeilenende also nicht gesetzt zu werden !

(80) F707 Der Inhalt des AR-X wird in das AR-S übertragen.

Syntax des Aufrufes: CD 80

veränderte Register:

X-Reg enthält 7A08
Y-Reg " 7A38
UH und A bleiben erhalten

(E6) F70D Der Inhalt des AR-X wird in das AR-Y übertragen.

Syntax des Aufrufes: E6

veränderte Register:

X-Reg enthält 7A08
Y-Reg " 7A18
UH und A bleiben erhalten

(EC) F757 Löscht AR-X.

Syntax des Aufrufes: EC

veränderte Register:

X-Reg enthält 7A08
UL = FF

(56) F7B0 Es werden 8 Bytes des Speicherbereiches (XH10 - XH17) in den Speicherbereich (YH00 - YH07) übertragen. Zeigen X-Reg und Y-Reg bei Einsprung auf die Speicherseite 7Axx, dann wird das AR-Y in das AR-X übertragen.

Syntax des Aufrufes: CD 56

veränderte Register:

XL = 18; YL = 08; UL = FF

(68) F715 Es werden 8 Bytes des Speicherbereiches (XH30 - XH38) in den Speicherbereich (YH10 - YH18) übertragen. Zeigen X-Reg und Y-Reg bei Einsprung auf die Speicherseite 7Axx, dann wird das AR-S ind das AR-Y übertragen.

Syntax des Aufrufes: CD 68

veränderte Register:

XL = 38; YL = 18; UL = FF

(84) EF00 Schaltet blinkenden Cursor auf der Anzeige aus.

Syntax des Aufrufes: CD 84

veränderte Register: keine
(787C) Bit Ø u. 1 werden gelöscht

(32) D071 Die 16-Bit Adresse im U-Reg wird auf den Basic-Stack berettet. Der Pointer auf diesen Stack (7882) wird um 2 inkrementiert.

Syntax des Aufrufes: CD 32

veränderte Register:

X-Reg zeigt auf die nächste freie Adresse des Stack.
A enthält das Low-Byte der Adresse im X-Reg

(30) DC16 Der Pointer auf den Basic-Stack (7882) wird um 8 dekrementiert. AR-X wird anschließend auf den Stack gerettet. Carry-Flag wird Ø gesetzt.

Syntax des Aufrufes: CD 30

veränderte Register:
X-Reg enthält nächste freie Stackadresse
UL " FF
A

(3A) CFFB Initialisiert FOR- und GOSUB- Pointer, sowie die Speicherstellen (788A) und (7B0E).

Syntax des Aufrufes: CD 3A

veränderte Register:

(788A) = 00 ?
(7B0E) Bit 5-7 werden gelöscht ?
(7890) = 38
(7891) = FF

(96) EA78 Using-Formatierung. Der numerische Wert im AR-X wird entsprechend dem vorgewählten USING/ Format im AR-Y und AR-U (Vorkommastellen) sowie AR-V und AR-W (Nachkommastellen) zur Ausgabe bereitgestellt.

Syntax des Aufrufes: CD 96

veränderte Register:

X-Reg, Y-Reg, U-Reg, A

(88) EDF6 Ausgabe eines Bitmusters auf der LCD-Anzeige.

Syntax des Aufrufes: CD 88

Eingangsparameter:

X-Reg muß bei Aufruf auf die Matrixspaltenadresse zeigen, in der das Bitmuster angezeigt werden soll.

A muß bei Aufruf einen bestimmten Wert enthalten. Bit Ø bis Bit 6 werden, falls gesetzt in der LCD- Anzeige angezeigt.
Bit Ø - Bit 3 werden in die durch X-Reg angegebene Adresse übertragen;
Bit 4 - Bit 6 kommen in die Adresse (X-Reg + 1)

veränderte Register:

X-Reg enthält die Adresse der nächsten Matrixspalte
UH enthält das übertragene Bitmuster
A enthält das in (X-Reg + 1) übertragene Bitmuster

(8C) EE1F Der Matrixpointer (7875) enthält einen Wert zwischen ØØ und 9C. Aus diesem Wert wird die Adresse der entsprechenden Matrixspalte ermittelt.

Syntax des Aufrufes: CD 8C

veränderte Register:

X-Reg enthält die errechnete Matrixspaltenadresse
A

(8A) ED5B Ausgabe eines ASCII-Zeichens auf der LCD-Anzeige. Der Wert aus dem Akku wird ab der Matrixspaltenadresse angezeigt, auf die das X-Reg zeigt. Wenn Bit 7 inä Ein zweiter Zeichensatz wird berücksichtigt.

Syntax des Aufrufes: CD 8A

veränderte Register:

X-Reg enthält die Adresse der LCD-Anzeige, an der das nächste Zeichen dargestellt werden kann (ursprüngl. Wert erhöht um 6)
A
U-Reg

(90) EDAB Lädt Matrixpointer (7875) nach A. Vergleich mit zulässigem Wertebereich &9C. Falls Wertebereich überschritten wird Carry-Flag gesetzt.

Syntax des Aufrufes: CD 90

veränderte Register: A enthält Matrixpointer

(92) ED00 Ausgabe eines Textes auf der LCD-Anzeige.

Syntax des Aufrufes: CD 92

Eingangsparameter:

Matrixpointer (7875) zeigt die Matrixspalte an, ab der der Text auf die Anzeige gebracht wird.

U-Reg muß bei Aufruf die Anfangsadresse des Textes enthalten.
A muß bei Aufruf die Anzahl der auszugebenden Zeichen enthalten

Ausgangsparameter:

Wenn der Cursorpointer (787B) dem Wert von UL entspricht, wird die Adresse, auf die der Cursor zeigt nach (787E) und (787F) übertragen. Das entsprechende ASCII-Zeichen wird nach (787D) gebracht.

Ist die LCD-Anzeige voll, erfolgt Rücksprung. In A ist die Anzahl der noch zu übertragenden Zeichen enthalten. In diesem Fall ist Carry-Flag gesetzt.

veränderte Register:

X-Reg enthält die nächste freie Matrixspaltenadresse
U-Reg enthält Adresse, die auf das letzte Zeichen zeigt, falls LCD-Anzeige voll war bzw. auf das dem Ende des Textes folgenden Zeichen, falls der Text auf die Anzeige gepaßt hat.
A

(94) EC5C Überträgt einen String, dessen Anfangadresse im X-Reg steht und dessen Länge in UL abgelegt ist in einen freien Bereich des Output-Buffers.

Syntax des Aufrufes: CD 94

Eingangsparameter:
Bei Auruf müssen die Anfangsadresse an das X-Reg und die Länge des Strings in UL übergeben worden sein.
Der Output-Buffer Pointer muß bei Aufruf das Low-Byte der 1. freien Adresse enthalten.

Ausgangsparameter:

Der Transfer wird abgebrochen, falls das Ende des Output-Buffers erreicht wurde. Carry ist dann gesetzt.

veränderte Register:
X-Reg
Y-Reg zeigt auf die nächste freie Adresse des Output-Buffers.

(A6) E451 Es wird geprüft, ob Bit 1 in .(FOOB) gesetzt ist. Dies ist der Fall, wenn während eines laufenden Programmes die Break/On Taste betätigt wird.

Syntax des Aufrufes: CD A6

veränderte Register:

keine, jedoch ist Zero-Flag=1 falls Break/On betätigt wurde.

(AO) E234 PV-Banking entsprechend dem Zustand von Bit O in (79DO). Ist Bit O = 1, wird das PV-Banking durchgeführt.

Syntax des Aufrufes: CD AO

veränderte Register:

keine
Zero-Flag kann sich ändern

(AC) E88C Zeitverzögerung um 15,625 ms. U-Reg enthält die Anzahl der zu durchlaufenden Verzögerungsschleifen. Der Abbruch durch Drücken der Break/on Taste ist möglich, und führt zum vorzeitigem Aussprung aus dem Programm.

Syntax des Aufrufes: CD AC

Eingangsparameter: U-Reg muß bei Aufruf die Anzahl der zu durchlaufenden Verzögerungsschleifen enthalten.

veränderte Register:
U-Reg enthält OOFF
A

(A2) E655 Beep On/Off. Es wird geprüft, ob in (786B) Bit O gesetzt ist. Ist Bit o = 1, dann wird in I/O PORT .(FOO8), PC6 auf Ø gesetzt, andernfalls auf 1. Ist also (786B)=1, dann wird der Buzzer deaktiviert.

Syntax des Aufrufes: CD A2

veränderte Register:

keine
Carry-Flag wird gesetzt, falls Beep OFF

(B2) B897 Setzt Zero-Flag falls U-Reg = FFFF.

Syntax des Aufrufes: CD B2

veränderte Register: keine
Zero-Flag ist 1 falls U-Reg = FFFF

(BE) E4A8 Ermittelt in Abhängigkeit von der angeschlossenen Peripherie die Token- Tabelle, die im Bereich 8000 - C000 am weitesten "oben" steht.
Ist CE-15o oder CE-158 angeschlossen, wird Carry-Flag auf 1 gesetzt.
UH enthält 2 wenn CE-150 angeschlossen; 1, wenn CE-158 angeschlossen.

Syntax des Aufrufes: CD BE

veränderte Register:

X-Reg enthält ggfls Adresse einer gefundenen Token-Tabelle.
Y-Reg u. UL unverändert.
UH enthält Hinweis auf angeschlossenes Gerät.
A
PV-Banking wird durchgeführt, falls in diesem Bereich Token-Tabelle angelegt ist.

(BC) E4B7 Ermittelt, ausgehend von dem Adressbereich, auf den X-Reg zeigt die nächstniedrigere Token-Tabelle innerhalb der Bereiches 8000 - C000.

Syntax des Aufrufes: CD BC

veränderte Register:

siehe (BE) E4A8

(8E) EDB1 Der Matrixpointer (7875) wird nach A geladen und falls kleiner &9C inkrementiert. Der neue Wert wird nach (7875) zurückgeladen.

Syntax des Aufrufes: CD 8E

veränderte Register:

(7875) enthält einen Wert zwischen Ø1 und 9C
A enthält neuen Matrixpointer
Carry-Flag wird auf 1 gesetzt, falls (7875) den zulässigen Wertebereich überschreitet.

(54) F7BO Setzt in X-Reg und Y-Reg den H-Byte-Pointer auf die Arithmetikregister.

Syntax des Aufrufes: CD 54

veränderte Register:

XH enthält 7A
YH enthält 7A
Alle übrigen Register sowie der Zustand der Prozessor-flags bleiben erhalten.

(CO) DDO8 Lädt nächste Basic-Instruktion oder Zeichen nach U-Reg. Y-Reg enthält bei Aufruf die Adresse der nächsten abzuarbeitenden Instruktion, bzw. die des nächsten Zeichens.

Syntax des Aufrufes: CO

Eingangsparameter:

Y-Reg muß auf den Basic-Hauptspeicher oder den Input-Buffer zeigen

Ausgangsparameter:

A Es folgte eine Basic-Instruktion:
Das Token wird in U-Reg geladen, Carry-Flag wird gesetzt.

B Es folgte ein Zeichen:
Das Zeichen wird nach UL geladen, UH enthält ØØ, Carry-Flag=Ø

veränderte Register:

U-Reg enth. das Zeichen/Token
Y-Reg wurde entsprechend der Anzahl der geladenen Werte (2 oder 1) inkrementiert.
A enthält den Wert von UL

(C2) DCD4 Syntaxprüfung. Es wird geprüft, nachdem mit Unterprogramm ØØ DDO8 ein neues Zeichen oder Token nach U-Reg geladen wurde, ob die nach U-Reg geladene Information mit dem/den folgenden Parameter/n übereinstimmt.
Dieses Unterprogramm kann sowaohl einen Zeichentest, als auch einen Test auf ein bestimmtes Token durchführen. Wird Übereinstimmung festgestellt, erfolgt Fortsetzung des Programms mit der auf P3 folgenden Adresse, andernfalls wird zu der durch P3 festgelegten Distanzadresse verzweigt.

Syntax des Aufrufes: C2 P1 (P2) P3

Bedeutung der Parameter:

P1 Zeichen, bzw. 1. Byte eines Token
P2 2. Byte eines Token
P3 relativer Sprung um P3 Bytes, gerechnet ab der auf P3 folgenden Adresse.

veränderte Register:

Y-Reg zeigt auf die nächste abzuarbeitende Adresse im Basic-Hauptspeicher bzw. Input-Buffer.
X-Reg enthält Adresse zu der Verzweigt wird.
U-Reg enthält das geladene Zeichen/Token
A

(C6) DD13 U-Reg enthält entweder ein Token oder ein Zeichen. Wurde nach U-Reg ein Zeichen geladen, dann ist UH=ØØ. Das Y-Reg zeigt auf die Speicherstelle nach dem in U-Reg geladenen Zeichen.
Für eine nachfolgende Syntaxprüfung wird Y-Reg wieder dekrementiert, sodaß es auf die Speicherstelle des Token oder Zeichen im U-Reg zeigt.
D.h. konkret, daß Y-Reg bei einem Token um 2 und bei einem Zeichen in U-Reg um 1 dekrementiert wird.

Syntax des Aufrufes: C6

veränderte Register:
Y-Reg wird um 1 bzw. 2 dekrementiert
Steht im U-Reg ein Token, dann ist Zero-Flag=Ø
Steht dort ein Zeichen, d.h. UH=ØØ ist Zero-Flag=1

(C8) DCC5 Syntaxprüfung. Das nächste Zeichen oder Token, auf daß das Y-Reg zeigt, wird nach U-Reg geladen.
Es wird mit Unterprogramm (04) DCC6 geprüft, in welcher Betriebsart sich der Rechner befindet und ob das nach U-Reg geladene Zeichen ein Ende der Befehlsfolge oder Zeilenende erkennen läßt. Parameter P1 legt eine Distanzadresse fest, zu der verzweigt wird, falls kein Zeilenende oder Ende der Befehlsfolge festgestellt wurde.

Syntax des Aufrufes: C8 P1

Bedeutung der Parameter:

P1 - relativer Sprung um P1 Bytes, gerechnet ab der auf P1 folgenden Adresse.

veränderte Register:

X-Reg enthält die Adresse, zu der verzweigt wird.
A
Carry-Flag ist gesetzt, falls in einem Basic-Programm ein Ende der Befehlsfolge (:) erkannt wurde.

(CA) COO1 Der 16-Bit Wert aus X-Reg wird in die Speicherseite 78xx übertragen. Das Low-Byte in der Speicherseite 78xx wird durch den Parameter P1 festgelegt. XH wird nach (78P1) und XL in die Folgeadresse übertragen.

Syntax des Aufrufes: CA P1

veränderte Register:
U-Reg enthält die Folgeadresse von 78P1
A enthält den Wert von XL

(CC) DDC8 Das X-Reg wird mit einem 16-Bit Wert geladen, der der Speicherseite 78xx entnommen wird.
Das Low-Byte der Adresse aus der Speicherseite 78xx wird durch den Parameter P1 festgelegt.
Der Inhalt dieser, sowie der Folgeadresse werden nach X-Reg übertragen.

Syntax des Aufrufes: CC P1

veränderte Register:

X-Reg enthält den geladenen Wert
A enthält den Inhalt von (78P1)

(D2) DD1A Das AR-X wird auf die enthaltene Information überprüft.
Ist dort ein numerischer Wert im BCD-Format abgespeichert, wird nichts unternommen und das Programm führt die auf P2 folgende Instruktion durch.
Ist im AR-X eine Zahl im Binärformat abgelegt, d.h. in Speicherstelle 7A04 steht B2 und in den Folgeadressen die Binärzahl, dann verzweigt das Programm zum Unterprogramm (10) DD2D.
Steht im AR-X eine character string information, die auf einen String hinweist, wird ein relativer Sprung um P1 Bytes durchgeführt, gerechnet ab der P1 folgenden Adresse.

Syntax des Aufrufes: D2 P1 P2

Bedeutung der Parameter:

P1 - relativer Sprung um P1-Bytes, gerechnet ab der P1 folgenden Adresse.
P2 - Datenbyte für das ggfls angesprungene Unterprogramm (10) DD2D (siehe dort: Bedeutung der Parameter)

veränderte Register:

X-Reg, U-Reg, A
wenn P2=ØØ bleibt Y-Reg unverändert.

(50) DA71 16 x 16 Bit-Multiplikation. U-Reg und Y-Reg enthalten je eine 16-Bit Zahl die miteinander multipliziert werden. Das Ergebnis der Multiplikation steht nach Beendigung des Programms in X-Reg und Y-Reg.
Wenn das Ergebnis größer als 65535 sein sollte, also mehr als 16 Bit beträgt, wird das Carry-Flag gesetzt.

Syntax des Aufrufes: CD 50

Ausgangsparameter:

X-Reg Rechenergebnis hexadezimal
Y-Reg "
Carry-Flag=Ø wenn das Ergebnis nicht größer als 16 Bit ist
Carry-Flag=1 wenn das Ergebnis 16 Bit übersteigt

veränderte Register:

X, Y, U, A

(D4) DEE3 Die Pointer für den aktuellen Bearbeitungsstand des Programms werden in der Memory-Map gesetzt.
Y-Reg enthält bei Einsprung in das Unterprogramm die aktuelle Programmadresse des Basic-Hauptspeichers.
Der Parameter P1 legt den Bereich innerhalb der Memory-Map fest, der aktualisiert werden soll.

Syntax des Aufrufes: D4 P1

Bedeutung der Parameter: P1 kann sinnvollerweise nur folgende Werte annehmen:

AC - Die Adresse der gerade abzuarbeitenden Zeile wird nach (78A0 - 78A1) übertragen (Inhalt des Y-Reg). Die aktuelle Zeilennummer wird aus (789C - 789D) entnommen und nach (78A2 - 78A3) übertragen. Die Programmstartadresse wird aus (789E - 789F) entnommen und nach (78A4 - 78A5) übertragen.
AC - Die Informationen für einen durchgeführten Break (Adresse, Zeilennummer, Startadresse des Programms) werden ab 78AC - 78B1 übertragen.
B2 - Die Informationen für einen aufgetretenen ERROR (Adresse, Zeilennummer, Startadresse des Programms werden ab 78B2 - 78B7 übertragen.

veränderte Register:

X-Reg
U-Reg enthält 78A0
Y-Reg unverändert
Das Carry-Flag bleibt unverändert

(18) DF80 Die Pointer für die zuletzt abgearbeitete Instruktion werden in der Memory-Map mit Unterprogramm (D4) gesetzt.
Ist bei Einsprung in das Unterprogramm Carry-Flag=Ø, dann zeigt Y-Reg auf die Adresse der nächsten Zeilennummer. Diese Zeilennummer wird nach 789C und 789D übertragen. Anschließend wird Y-Reg inkrementiert, so daß es auf die erste Instruktion der entsprechenden Basic-Zeile zeigt.
Ist bei Einsprung Carry-Flag=1, wird Bit 7 in YH gesetzt und es erfolgt Rücksprung.

Syntax des Aufrufes: CD 18

veränderte Register:

X-Reg, Y-Reg, U-Reg, A

(D6) DED1 Y-Reg wird mit einer neuen Adresse, die auf den Basic-Hauptspeicher zeigt, geladen. Anschließend werden die zu dieser Adresse gehörenden Programmpointer (Zeilennummer und Startadresse) in die Speicherzellen 789C - 789F geschrieben.
Parameter P1 legt fest, von welcher Adresse der Speicherseite 78xx diese Informationen geholt werden sollen (Low-Byte).

Syntax des Aufrufes: D6 P1

Bedeutung der Parameter: P1 nimmt folgende Werte an:

A6 - (78A6 - 78AB) werden wie o.a. übertragen. Dort liegen z.B. Pointer für eine gesuchte Zeilennummer.
AC - (78AC - 78B1) werden übertragen. Dort liegen Pointer, falls ein Break durchgeführt wurde.
B8 - (78B8 - 78BD) werden übertragen. Dort sind die Pointer einer ON ERROR GOTO/GOSUB Anweisung abgelegt.

veränderte Register:

X-Reg, U-Reg, A
Y-Reg enthält eine Adresse, die auf den Basic-Hauptspeicher zeigt.

(D8) DF3B Prüft, ob gerade ein Basic-Programm abgearbeitet wird. In diesem Fall enthalten die Speicherstellen (789C und 789D) die aktuelle Zeilennummer.
Falls kein Programm abgearbeitet wird ist Zero-Flag=1, andernfalls Ø

Syntax des Aufrufes: D8

veränderte Register: A

(DC) DEBC AR-X enthält eine character string information. Die im AR-X abgelegte Adresse (7A05 - 7A06) wird in das X-Reg geladen. Die Information über die Länge des Strings, bzw. 88 bei einer numerischen Variablen (7A07) wird in den Akku und nach UL geladen.

Syntax des Aufrufes: DC

veränderte Register:

X-Reg enthält die Adresse
U-Reg
A

(DA) COOE U-Reg enthält die Anfangsadresse einer Variablen bzw. Konstanten. (7A07) enthält die Länge der Variablen. Die Adresse aus U-Reg wird im Variablen-Pointer (7883 - 7884) zwischengespeichert. Die Länge aus (7A07) wird in (7885) zwischengespeichert.

Syntax des Aufrufes: DA

veränderte Register:

A

(BA) F763 Löschen eines Speicherblocks dessen Anfangsadresse das X-Reg enthält und dessen Länge an UL übergeben wurden.

Syntax des Aufrufes: CD BA

Eingangsparameter: X-Reg muß die Anfangsadresse des zu löschenden Bereiches enthalten, UL die Anzahl der zu löschenden Bytes - 1.

veränderte Register:

X-Reg zeigt auf das erste Byte nach dem gelöschten Speicherblock
UL = FF
A = ØØ

(52) F663 Konvertiert Rechenergebnis im AR-X in die korrekte Form der internen Zahlendarstellung (Exponent, Vorzeichenbyte, Mantisse). Bei Aufruf des Unterprogrammes muß der Akku das spätere Vorzeichenbyte des numerischen Wertes enthalten. Durch internes Verschieben und gleichzeitige Korrektur des Exponenten wird sichergestellt, daß das obere Nibble von (7A02) die 1. Ziffer des numerischen Wertes enthält. Ergibt die Korrektur des Exponenten, daß der obere, zulässige Wertebereich überschritten wird, wird Error-Code 25 nach UH geladen und das Carry-Flag gesetzt. Bei einer Unterschreitung des zulässigen Wertebereiches wird AR-X gelöscht, d.h. der numerische Wert wird Ø.

Syntax des Aufrufes: CD 52

Eingangsparameter: Akku muß bei Aufruf das Vorzeichen des numerischen Wertes enthalten. AR-X muß das Rechenergebnis enthalten.

veränderte Register:

X-Reg, Y-Reg, U-Reg, A

(EA) F79C Der Inhalt des Speicherbereiches xx01 - xx07 wird um ein Nibble nach links geschoben. Durch XH wird das H-Byte dieses Bereiches bestimmt. In der Regel ist dies 7A, d.h. die Operation wird im AR-X durchgeführt.
Der Exponent (7A00) bleibt erhalten. Nach Durchführung enthält A den Inhalt der Speicherstelle (7A01).

Syntax des Aufrufes: EA

veränderte Register:

X-Reg zeigt auf xx01
Y-Reg und UH bleiben erhalten
A enthält den ursprünglichen Wert von (xx01)

(74) F775 Der Inhalt des Speicherbereiches xx01 - xx07 wird um ein Nibble nach rechts geschoben. Durch XH wird das H-Byte dieses Bereiches bestimmt.
Nach Durchführung enthält der Akku den ursprünglichen Wert der Speicherzelle xx07 vor Durchführung der Operation.

Syntax des Ausrufes: CD 74

veränderte Register:
X-Reg enthält xx07
Y-Reg und UH bleiben unverändert
A enthält den ursprünglichen Wert von (xx07)

(76) F75F Der Inhalt des Speicherbereiches (xx01 - xx07) wird gelöscht (ØØ). XH legt das H-Byte dieses Bereiches fest. Zeigt X-Reg auf die Speicherseite 7Axx, wird der entsprechende Bereich im AR-X gelöscht. Der Exponent in (7A00) bleibt erhalten.

Syntax des Aufrufes: CD 76

veränderte Register:

Y-Reg und UH bleiben erhalten.
X-Reg zeigt auf xx08
A enthält ØØ

(78) F72F X-Reg und Y-Reg zeigen bei Einsprung auf einen bestimmten Speicherbereich. Es werden 7 Bytes des Bereichs (XH01 - XH07) in den Bereich (YH09 - YH0F) übertragen. Zeigen X-Reg und Y-Reg auf die Speicherseite 7Axx, dann wird aus dem AR-X das Vorzeichen und die Mantisse eines dort abgelegten Wertes in das AR-Z übertragen.

Syntax des Aufrufes: CD 78

veränderte Register:

X-Reg = XH08, Y-Reg = YH10, A und UH unverändert

(EE) F7CC Vorzeichen und Mantisse der numerischen Werte in den AR-X und AR-Y werden addiert und das Ergebnis in AR-X abgelegt.

Syntax des Aufrufes: EE

Eingangsparameter:
XH und YH müssen bei Aufruf 7A enthalten.
Bei unterschiedlichen Exponenten muß eine Mantisse stellengerecht korrigiert sein, da sonst das Ergebnis falsch wäre.

veränderte Register:

X-Reg zeigt auf 7A00
Y-Reg zeigt auf 7A10
UL, A

(42) CA58 RUN-Modus. Mit diesem Unterprogramm springt der Rechner aus einem laufenden Maschinen- oder Basic-Programm zurück in den Eingabemodus.
Die Stacks werden initialisiert und die LCD-Anzeige gelöscht. Auf der Anzeige erscheint das Prompt-Zeichen. Der Rechner erwartet eine Eingabe.

Syntax des Aufrufes: CD 42

(E8) F661 Konvertiert das Rechenergebnis im AR-X in die korrekte Form der internen Zahlendarstellung, wie in (52) F661.
Unterschied: Als Vorzeichenbyte wird ØØ also positiv in den numerischen Wert eingesetzt.

Syntax des Aufrufes: E8

veränderte Register:

X-Reg, A, U-Reg

(9E) E4A0 Es wird geprüft, ob ein String auf dessen 1. Byte Y-Reg zeigt mit einem String, dessen character string information im AR-Y abgelegt ist, übereinstimmt.
Falls beide Strings gleich sind, wird der numerische Wert 1 in das AR-X geladen und Zero-Flag ist 0.

Syntax des Aufrufes: CD 9E

Eingangsparameter:
Y-Reg muß auf das 1. Byte des zu prüfenden Strings zeigen.
AR-Y muß character string information des 2. Strings enthalten.

veränderte Register:
X-Reg, Y,Reg, U-Reg a

Die folgenden Unterprogramme beziehen sich auf das "Kassetteninterface". Sie lassen sich im Rahmen von Maschinenspracheprogrammierung nur schwer sinnvoll einsetzen. Das Laden von Programmen erfolgt nicht mit einem Uart sondern durch Portabfragen. Mit der eingebauten Hardware läßt sich das Laden von Programmen lediglich um den Faktor 2 beschleunigen und dies auch nur, wenn das Format der abgespeicherten Daten geändert wird. Für Interessenten seien hier dennoch die entsprechenden Unterprogramme aufgeführt:

(A4)	B888	Laden eines Zeichens von Kassette
(A8)	B88B	Speichern eines Zeichens auf Kassette
(AA)	BD3C	Filetransfer auf/von Kassette
(AE)	B891	Vergleicht Prüfsumme der empfangenen Daten mit der empfangenen Prüfsumme
(BO)	BCE8	Lädt/empfängt Kopf eines Programmes von/auf Band
(B6)	B89D	Entwirft Kopf für ein abzuspeicherndes Programm
(B8)	B8AO	Syntaxprüfung. Wenn einer Kassettenoperation "-1" folgt, wird RMT-Parameter (7879) entsprechend verändert.

Die folgenden Unterprogramme waren bei Manuskripterstellung noch nicht voll analysiert. Für Interessenten hier ein kurzer Hinweis über die Wirkung:

(1A)	D2E6	Sucht Programmzeile und setzt Parameter in der Memorymap
(1C)	FA89	Ermittelt aus dem in U-Reg geladenen Token, und der durch X-Reg festgelegten Token-Tabelle die Startadresse des Basic-Befehls im ROM.
(3E)	FB9B	Trace ?
(4O)	C4O1	Einsprung in den Basic-Interpreter
(5C)	F61B	Generiert Zufallszahlen
(EO)	CD8B	Gibt ERROR 1 aus und führt in den RUN-Modus zurück
(E2)	C4OO	Basic-Interpreter
(E4)	CD89	Gibt ERROR aus UH aus und führt in den RUN-Modus zurück. (Hinweis: Die in der Tokentabelle auf diese Adresse gerichteten Basic-Befehle werden hier <u>nicht</u> bearbeitet, sondern an anderer Stelle. Zu dieser Adresse wird nur dann verzweigt, wenn (AND, ERROR, OR, OFF, THEN) als <u>erste</u> Instruktion eines zusammengesetzten Befehls genannt werden.)
(F8)	E171	Maskierbarer Interrupt
(FA)	E22C	Timer-Interrupt
(FC)	E22B	Nicht-maskierbarer Interrupt
(FE)	EOOO	Reset-Routine

Arithmetikroutinen des PC-1500

Eingangsparameter: AR-X muß numerischen Wert enthalten
AR-Y muß, falls für die Operation zwei Variable verwendet werden, den zweiten numerischen Wert enthalten.

Ausgangsparameter: Das Ergebnis der Rechnung steht im AR-X zur Verfügung.

Funktion	Aufruf	Wirkung	Operationen ca. pro/sec
Addition	FO	AR-X = AR-X + AR-Y	1100
Subtraktion	BE EF B6	AR-X = AR-X - AR-Y	860
Multiplikation	CD 7A	AR-X = AR-X x AR-Y	130
Division	CD 58	AR-X = AR-X / AR-Y	50
SQR X	BE FO E9	AR-X = SQR AR-X	43
LN X	BE F1 61	AR-X = LN AR-X	8
LOG X	BE F1 65	AR-X = LOG AR-X	9
EXP X	BE F1 CB	AR-X = EXP AR-X	13
10^X	BE F1 D4	AR-X = 10^{AR-X}	53
SIN X	BE F3 A2	AR-X = SIN AR-X	8
TAN X	BE F3 9E	AR-X = TAN AR-X	14
ASN X	BE F4 9A	AR-X = ASN AR-X	8
ATN X	BE F4 96	AR-X = ATN AR-X	10
COS X	BE F3 91	AR-X = COS AR-X	7
ACS X	BE F4 92	AR-X = ACS AR-X	7
DEG X	BE F5 31	AR-X = DEG AR-X	23
DMS X	BE F5 64	AR-X = DMS AR-X	56
ABS X	CD 6C	AR-X = ABS AR-X	56
SGN X	BE F5 9D	AR-X = SGN AR-X	2300
INT X	BE F5 BE	AR-X = INT AR-X	1100
1 / X	CD 54 CD 6E	AR-X = 1 / AR-X	50