

# KLEINCOMPUTER

---

## KC85/2



Systembeschreibung HC-CAOS

Beschreibung des  
Betriebssystems

Version 2.1  
Version 2.2

Stand 09/85

**veb mikroelektronik**  
**>wilhelm pieck<**  
**mühlhausen**

**im veb kombinat mikroelektronik**

# I n h a l t

1.	Systemkonzept .....	1
1.1.	Einführung .....	1
1.2.	Merkmale des Betriebssystems .....	1
2.	Speicheraufteilung .....	3
2.1.	Vorhandene Speicher .....	3
2.2.	Verwendbare Speicherbereiche für Programme u. Daten	3
3.	Modulkonzept .....	4
4.	Menütechnik .....	7
5.	Systemschnittstellen .....	9
6.	Systemarbeitszellen .....	11
6.1.	Arbeitszellen im IRM .....	11
6.2.	Arbeitszellen im IX-Bereich .....	13
6.3.	Interrupttabelle .....	13
6.4.	Kellerspeicher .....	14
6.5.	Verändern des Arbeitsspeicherbereiches .....	14
6.6.	Zeichenbildtabellen .....	14
6.7.	Steuercodetabellen .....	16
7.	Funktionstasten .....	17
8.	Magnetbandaufzeichnung .....	18
8.1.	Verfahren .....	18
8.2.	Dateiaufbau .....	18
8.3.	Dateitypen .....	19
9.	Tastencodes .....	20
10.	Spezielle Systembedingungen .....	21
11.	Anlage 1: Unterprogramme .....	22
12.	Anlage 2: Adreßzuweisungen IRM .....	33.1
13.	Anlage 3: Steuercodes CAOS V2.2 und Aufbau der Um- codierungstabelle für die Tastatur .....	34

## 1. SYSTEMKONZEPT

### 1.1. EINFÜHRUNG

Der Kleincomputer KC85/2 ist ein Gerät mit hohen Gebrauchswerteigenschaften. Neben Tonausgabemöglichkeiten können vollgrafische Bilder in 16 Vordergrund- und 8 Hintergrundfarben dargestellt werden.

Der KC85/2 besitzt einen RAM von 16 KByte, einen IRM (Bildwiederholpeicher) von 16 KByte und einen ROM von 4 KByte. Dieser Betriebssystem-ROM enthält nur die wichtigsten Programme zur Bedienung der Peripherie. Eine Übersicht wird im Bild 1 (S. S. 2.1) gegeben.

Das Betriebssystem HC-CAOS (**C**assette **A**ided **O**perating **S**ystem) verwaltet die Gerätetreiber-Routinen mittels Menütechnik. Im Folgenden sollen die einzelnen Software-Baugruppen näher beschrieben werden. Voraussetzung zur Anwendung sind Kenntnisse in Assemblerprogrammierung SYPS K1520.

### 1.2. MERKMALE DES BETRIEBSSYSTEMS

Das Betriebssystem HC-CAOS ist, um vielen Anwendungsbereichen gerecht zu werden, sehr flexibel ausgelegt. Es ermöglicht dem Anwender

- das Ein-Ausgabegerät des Betriebssystems frei zu definieren.
- den Arbeitsspeicher für das Betriebssystem, den Kellerspeicher (STACK) und die Interrupttabellen an beliebige Stellen im RAM anzuordnen.
- leicht, eigene Maschinenprogramme durch Menütechnik in das System einzubinden.
- den eigenen Programmen beim Aufruf über Menü bis zu maximal 10 Parameter zu übergeben.
- die Systemressourcen durch eine große Anzahl von Systemunterprogrammen vollständig zu nutzen.
- Erweiterungsbaugruppen (Module) zu verwalten, d.h., es können somit max. 62 Module quasi gleichzeitig betrieben werden.
- die im Grundgerät enthaltenen Speicher (RAM, IRM, ROM) ein- und auszuschalten.
- das im Grundgerät enthaltene Betriebssystem abzuschalten und mit einem anderen, in einem Modul enthaltenen, zu arbeiten.
- RAM-Speicherblöcke mit einem Schreibschutz zu versehen.
- die sechs auf der Tastatur befindlichen Funktionstasten (F1...F6) in beiden möglichen Belegungen mit beliebigen Codes oder Zeichenketten (z.B. Menü- oder BASIC-Schlüsselwörtern oder Abarbeitungstastensequenzen (Jobs) zu belegen.
- für die Darstellung von Zeichen auf dem Bildschirm beliebige Zeichenbildertabellen (Zeichengeneratoren) zu verwenden, d.h., man kann sich Zeichenbilder frei definieren (z.B. kyrillische Buchstaben, Grafikzeichen) und diese z.B. auf Magnetband abspeichern und

- die Zeichencodes der Tastatur beliebig zuordnen.

Im Bild 2 ist die zentrale Steuerschleife von HC-CAOS angegeben. Daraus wird die Steuerung der Funktionen von CAOS deutlich.

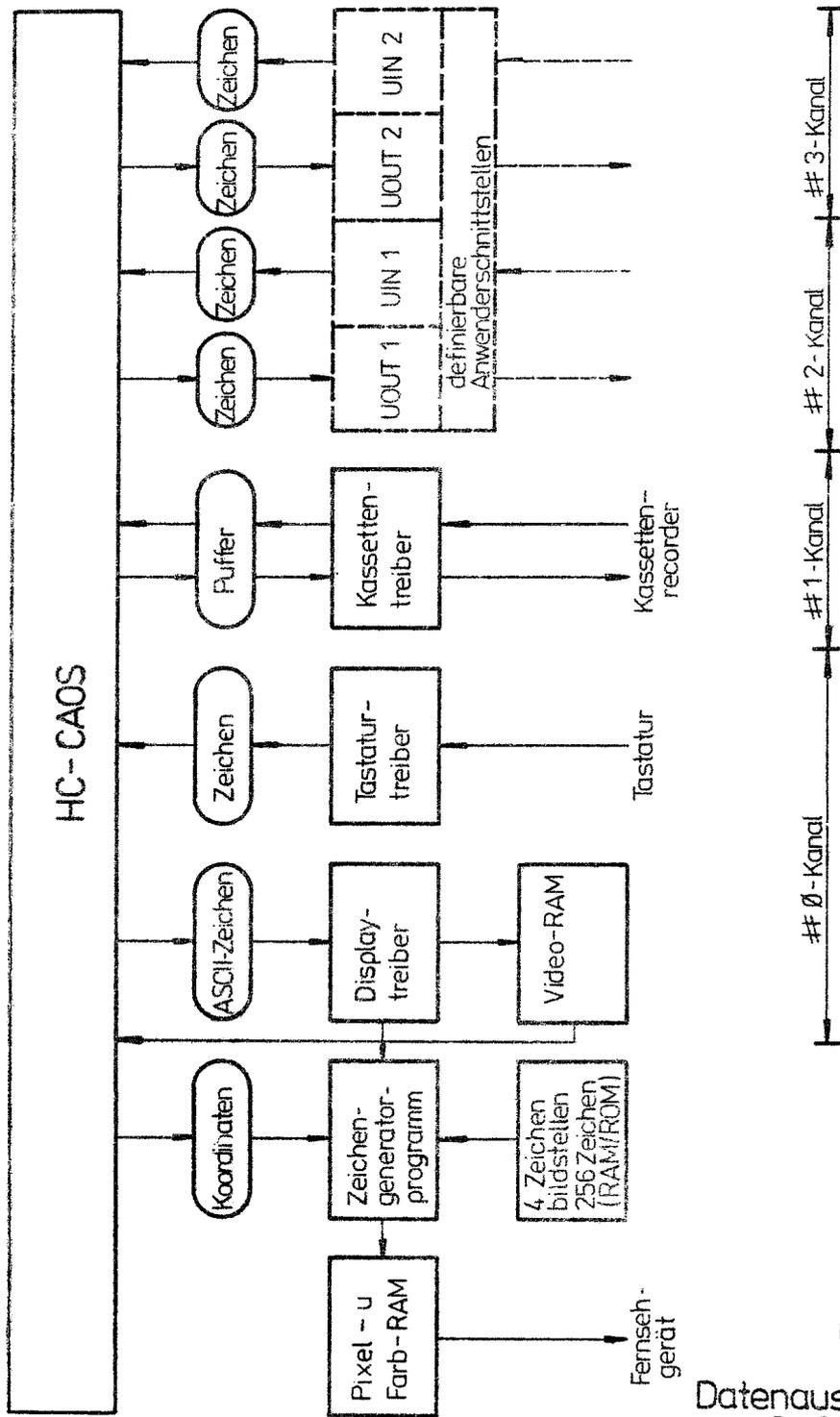


Bild 1

Datenaustausch von CAOS mit der Peripherie

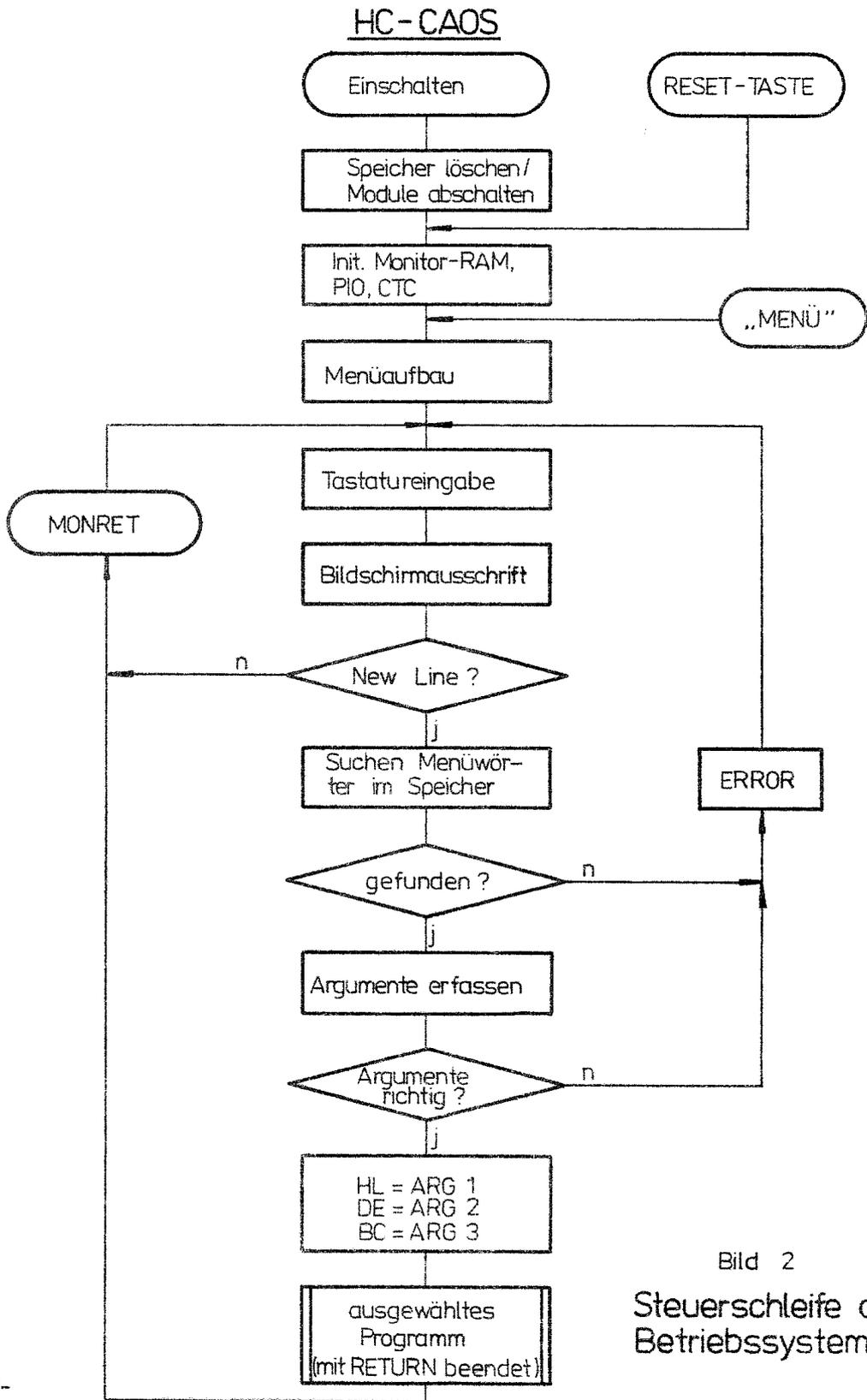


Bild 2  
Steuerschleife des  
Betriebssystem

## 2. SPEICHERAUFTEILUNG

### 2.1. VORHANDENE SPEICHER

RAM: 16K DRAM Adresse 0000H...3FFFH - Anwenderspeicher  
IRM: 16K DRAM Adresse 8000H...8FFFH - Bildwiederholtspeicher  
ROM: 2x2K EPROM Adresse E000H...E7FFH  
und F000H...F7FFH - Betriebssystem

### 2.2. VERWENDBARE SPEICHERBEREICHE FÜR PROGRAMME UND DATEIEN

#### Im Grundgerät

- Adreßbereich 000H...13FH: für Anwender bedingt nutzbar, da diesen Bereich Spezialprogramme (z.B. Testmonitor u.a.) verwenden.
- Adreßbereich 140H...1FFFH: im Normalzustand Monitor-RAM und STACK; kann vom Anwender auf andere Adreßbereiche undefiniert werden (vgl. Unterprogramm SIXD).
- Adreßbereich 200H...3FFFH: frei für Anwender.
- Adreßbereich 8000H...BBFFH: wird vom Betriebssystem genutzt:

8000H...A7FFH	= Pixel-RAM (Bildpunktspeicher)	10	KByte
A800H...B1FFFH	= Color-RAM (Farbattributspeicher)	2,5	KByte
B200H...B6FFFH	= Video-RAM (ASCII-Speicher)	1,25	KByte
B700H...B77FFH	= Kassettenpuffer	128	Byte
B780H...B7FFFH	= Monitor-RAM	128	Byte
B800H...B8FFFH	= Modul-Steuerwort-Speicher	256	Byte
B900H...B97FFH	= Funktionstastenspeicher	128	Byte
B980H...B9FFFH	= Fenstervektorspeicher	128	Byte
BA00H...BBFFFH	= nachladbare Zusatzprogramme	512	Byte

- Adreßbereich BC00H...BFFFFH: frei für Anwender, hierbei ist zu beachten, daß die Zugriffszeit zu diesem RAM relativ groß ist (2,4 µs) und die Zugriffe auf dem Bildschirm sichtbar sind.

(vgl. Bild 3)

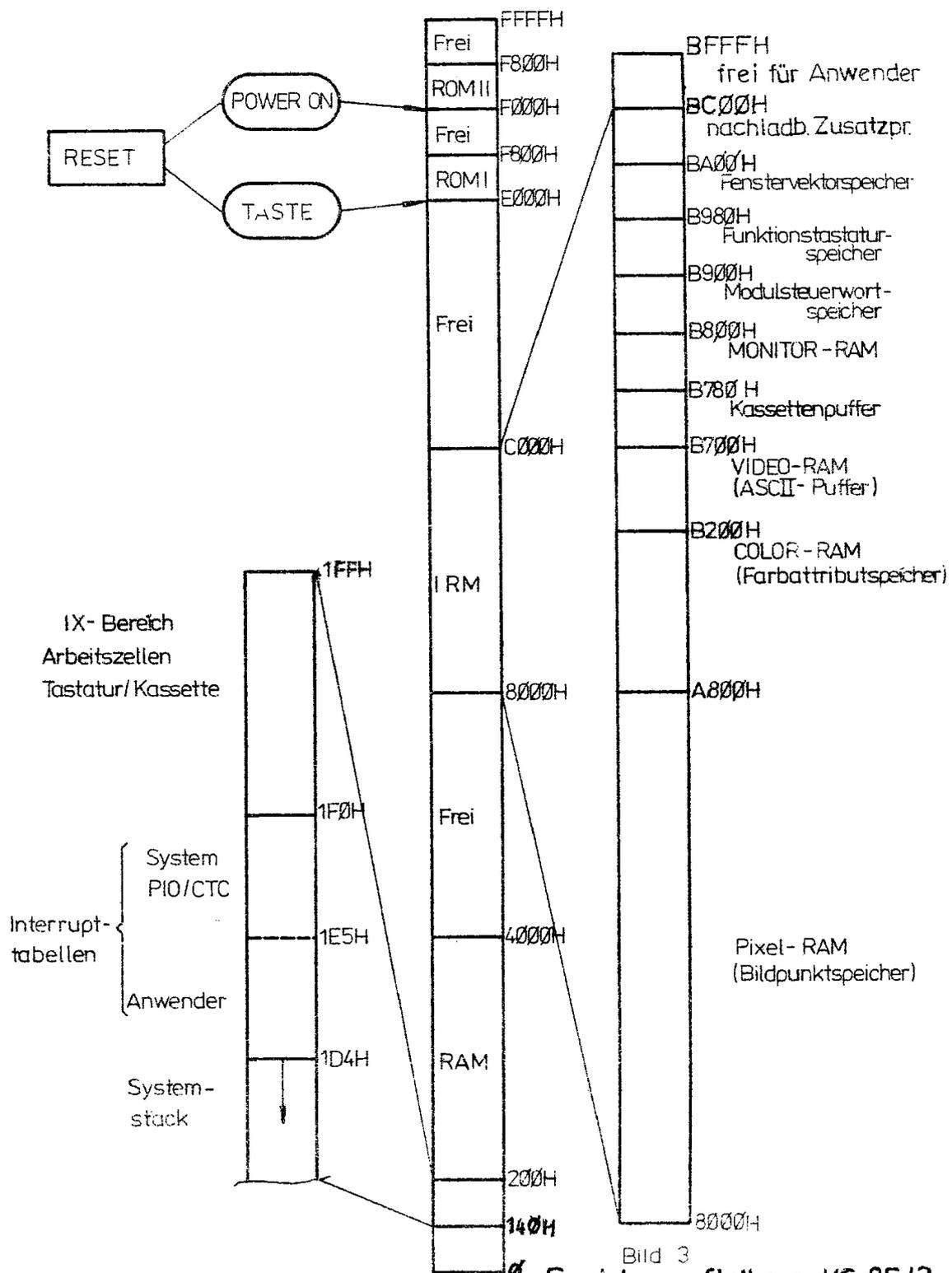


Bild 3

Speicheraufteilung KC 8512

### 3. MODULKONZEPT

Das Grundgerät des KC85/2 erlaubt den Anschluß von zwei Erweiterungsmodulen und mehreren Erweiterungsaufsätzen. Für die Module befinden sich an der Vorderseite des Grundgerätes zwei Modulschächte, in die Module eingesteckt und mit dem Rechnerbus kontaktiert werden.

Als Module sind geplant:

- RAM-Expansion (16 KByte DRAM)
- EPROM-Expansion (8 KByte)
- Anwender-Port digitale und incrementale Ein-/Ausgabe (1 PIO, 1 CTC)
- Serielles Interface V24
- RAM-Expansion 64 KByte DRAM)
- Module mit Festprogrammen (BASIC-Modul) usw.

Für die Erweiterungsaufsätze wurde das gleiche Gefäßsystem wie für das Grundgerät vorgesehen. Die Aufsätze sind stapelbar, d.h., bei entsprechender Verfügbarkeit kann sich der Anwender einen "TURM" zusammenstellen. Die Aufsätze werden an der Rückseite untereinander mit dem im Grundgerät enthaltenen Rechnerbus verbunden. Jeder Aufsatz enthält eine eigene Stromversorgung.

Als Erweiterungsaufsätze sind geplant:

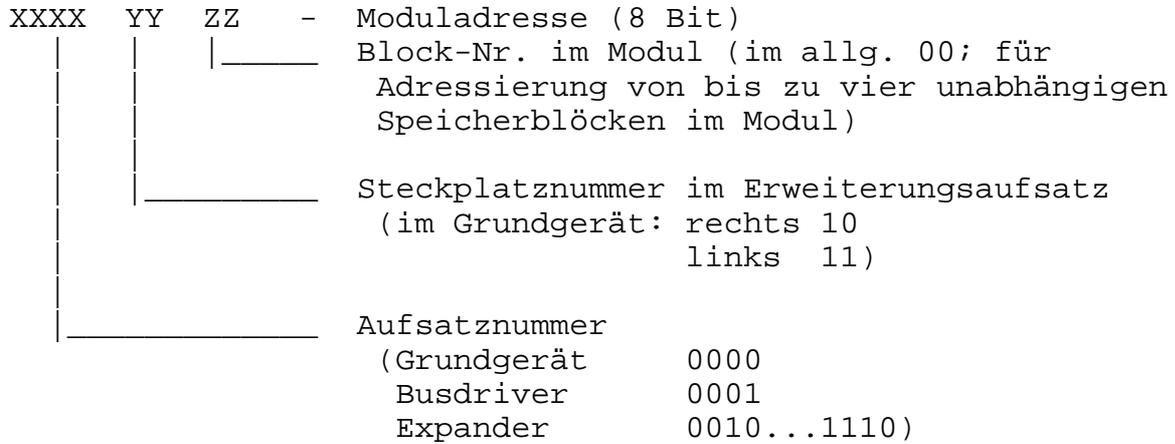
- Aufsatz mit vier weiteren Modulschächten
- EPROM-Programmier- und Löscherät

Durch eine spezielle Steuerung ist es beim KC85/2 möglich, mehrere Module vom gleichen Typ quasi gleichzeitig zu betreiben. Somit kann z.B. der Adreßbereich der Speicher des KC85/2 theoretisch auf maximal 54000 KByte oder der Adreßbereich der Ein-Ausgabebereiche auf maximal 1024 Ports (bei Kombination von Speichern und Ein-Ausgabebereichen entsprechend weniger) ausgedehnt werden.

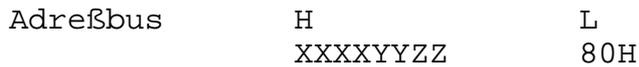
Das Ansprechen der Module erfolgt über vom Steckplatz abhängige Moduladressen. Den im Grundgerät enthaltenen drei Speicherblöcken sind folgende Moduladressen zugeordnet:

- RAM        -     00H
- IRM        -     01H
- ROM        -     02H

Diese Blöcke werden über den internen PIO-Baustein ein- bzw. ausgeschaltet. Beim RAM-Block kann ein Schreibschutz gesetzt werden. Die Moduladressen sind folgendermaßen definiert:



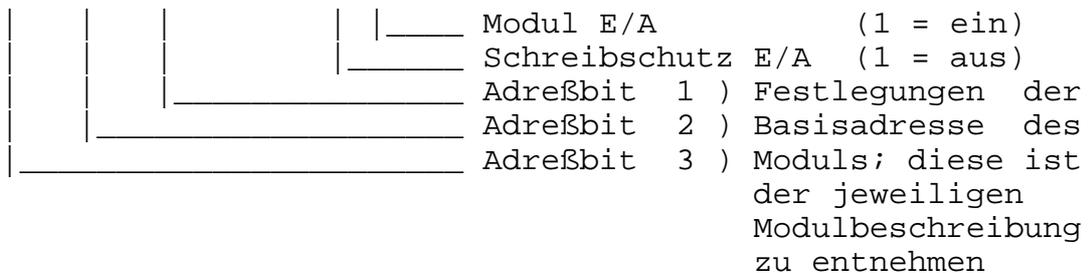
Die Moduladressierung erfolgt über I/O-Adressen:



Beim Lesen der entsprechenden Adresse sendet jedes Modul ein spezielles Strukturkennbyte auf den Datenbus. Die Kennungen der Module sind den Modulbeschreibungen zu entnehmen.

Das Schalten der Module erfolgt über Ausgabe eines Steuerbytes mit der Moduladresse.

A(3)A(2)A(1) XXXX W M



Über A(1), A(2), A(3) kann bei Speichermodulen die absolute Adresse dem Speicherbereich des Moduls zugeordnet werden. Werden mehrere Module mit gleichen Speicher- und E/A-Adressen eingeschaltet, so ist beim Zugriff des Prozessors nur der Modul auf der niedrigsten Moduladresse wirksam (Hardware-Prioritätskette).

(vgl. "SWITCH" in der Bedienungsanleitung)

Die internen Speicher (RAM, IRM, ROM) enthalten keine Modulsteuerung. Sie werden über die Daten des PIO-Ports A gesteuert. Die Speicher sind eingeschaltet, bzw. der Schreibschutz ist ausgeschaltet, wenn das zugehörige Ausgabebit log. 1 ist. Es gibt folgende Zuordnung zum PIO-Port A (Adresse 88H/Daten):

BIT 0 = ROM  
BIT 1 = RAM  
BIT 2 = IRM  
BIT 3 = Schreibschutz RAM  
BIT 4 = frei  
BIT 5 = LED "TAPE" an der Frontplatte  
BIT 6 = Motorschaltspannung (Schnellstop) des Kassetten-  
recorders  
BIT 7 = ROM (Erweiterung)

Für PIO-Port B (Adresse 89H/Daten) gilt:

BIT 0 = )  
BIT 1 = )  
BIT 2 = ) Lautstärkeregelung für Tonausgang  
BIT 3 = )  
BIT 4 = )  
BIT 5 = frei  
BIT 6 = frei  
BIT 7 = Blinken ein/aus

Der PIO-Baustein ist auf Byte-Ausgabe programmiert. Die zugehörigen Strobe-Eingänge werden für Tastatur- und Kassetten-eingabe-Interrupts verwendet.

Der in der gleichen Baugruppe enthaltene CTC-Schaltkreis wird vollständig mit weiteren Systemfunktionen belegt:

CTC-Kanal 0 (Adresse 8CH): Tonausgabe  
CTC-Kanal 1 (Adresse 8DH): Tonausgabe  
CTC-Kanal 3 (Adresse 8EH): Blinksteuerung, Zeitgeber  
für Kassetteneingabe  
CTC-Kanal 4 (Adresse 8FH): Zeitgeber für Tastatureingabe

#### 4. MENÜTECHNIK

Das verwendete Menükonzept ist unabhängig von bestimmten Speicherplätzen, d.h., jedes Programm auf beliebigen Speicherplätzen kann mit entsprechendem "Vorspann" ins Menü eingetragen und über dieses gestartet werden.

```
Vorspann:      7FH )   Prolog
                7FH )

                NN )   max. 32 Zeichen lange Zeichenkette
                  : )   aus Buchstaben, Steuerzeichen
                  : )   und Ziffern (ASCII)
                MM )

                00 bzw. 01 - Epilog
                PP          - 1. Befehlsbyte des Programmes
```

```
Epilog = 00 : mit Abschalten des IRMs beim Programmaufruf
          = 01 : ohne Abschalten des IRMs beim Programmaufruf
```

Bei Großbuchstaben und Ziffern (Code 30H bis 5FH) im Menüwort erfolgt der Eintrag ins Menü auf dem Bildschirm. Bei Sonderzeichen, Groß- und Kleinbuchstaben und Ziffern ist der Aufruf nur über den Namen möglich (Codes 20H bis FFH). Dabei erfolgt kein Eintrag in das Menü, aber das Programm kann über Tastatur aufgerufen werden.

Die mittels Menütechnik zu startenden Maschinenprogramme müssen als Unterprogramme definiert sein, d.h., sie müssen mit RETURN abgeschlossen sein. Die Unterprogramme werden bei Anwahl durch das Menüwort auf dem ersten Byte nach dem Prolog gestartet. Beim Programmaufruf können auf der gleichen Bildschirmzeile bis zu 10 Argumente (ARG1 - ARG10) übergeben werden. Die Anzahl der Argumente wird im Speicherplatz ARGN abgelegt (vgl. Abschnitt 6), der Wert der Argumente als 2-Byte-Zahlen auf den Speicherplätzen ARG1...ARG10.

Gleichzeitig werden die ersten drei Argumente in die Register

HL = ARG1

DE = ARG2

BC = ARG3

geladen und können von dem jeweiligen Unterprogramm verwendet werden.

Beispiel:

Für das Umspeichern von Speicherbereichen in andere ist das Programm "COPY" in das Menü aufzunehmen.

Folgendes "Hand"-Assemblerprotokoll ist die Grundlage:

ADR.	MC	Anweisung	Bemerkung
0000	7F 7F	MENUW: DA 7F7FH	Prolog
0002	43 4F 50 59	DB "COPY"	Menüwort
0005	01	DB 1	Epilog
0006	ED B0	COPY: LDIR	Umspeichern
0008	C9	RET	Rücksprung ins CAOS

Der Maschinencode (MC) ist mittels des Kommandos MODIFY ab Adresse 0 einzugeben und danach das Menü mit "MENU" aufzurufen. Das Kommando "COPY" kann wie folgt verwendet werden:  
Kopieren der Zeichenbildtabelle aus dem ROM (Anfangsadresse 0E600H) in den RAM an Adresse 0BE00H mit einer Länge von 512 Bytes (=200H).

```
COPY E600 8E00 200 <ENTER>
      HL  DE  BC
```

## 5. SYSTEMSCHNITTSTELLEN

Um den Nutzern des Kleincomputers KC85/2 die Arbeit zu ermöglichen, stehen ihm vom Betriebssystem 53 spezielle System-Unterprogramme zur Verfügung.

Dabei wird der Aufruf von Betriebssystem-Unterprogrammen und -Gerätetreiberrountinen (UP) über einen Programmverteiler gesteuert. Das Betriebssystem enthält eine Liste, in der alle UP numeriert sind. Dem Programmverteiler muß als ein Parameter diese UP-Nummer übergeben werden, damit wird das entsprechende UP gestartet.

Für den Anwender sind im wesentlichen sieben Adressen des Betriebssystems interessant:

### **F000H: Reset-Adresse**

Diese Adresse wird beim Einschalten des KC85/2 angesprungen. Der komplette RAM-Speicher wird gelöscht, alle Module werden abgeschaltet (außer 16K-DRAM im rechten Modulschacht des Grundgerätes) und das System wird initialisiert.

### **F003H: Programmverteiler I**

Die Unterprogrammnummer muß im rufenden Programm unmittelbar nach dem CALL-Befehl notiert werden.

Beispiel:      CALL        0F003H  
                  DB         UPNR

Die Parameter für die UP werden in den Registern übergeben. Die Register werden entsprechend der Unterprogramme verändert.

Stacktiefe des Verteilers: 2

### **F006H: Programmverteiler II**

Dieser Programmverteiler entspricht dem von F003H, die UP-Nr. wird jedoch im RAM auf einer festgelegten Adresse übergeben (ARGC vergl. Punkt 6). Die Register BC, DE, HL werden gerettet.

Stacktiefe des Verteilers: 7

### **F009H: Programmverteiler III**

Funktion wie Programmverteiler II. Die UP-Nr. wird im Register E übergeben. Damit entfällt allerdings das Register E für die Parameterübergabe.

### **F00CH: Programmverteiler IV**

Funktion wie Programmverteiler III, jedoch mit Einschalten des IRMs beim Aufruf und Abschalten des IRMs beim Rücksprung in das Anwenderprogramm.

**F00FH: Relativer Unterprogrammaufruf**

Mit UP-Abstand unmittelbar nach Aufruf

z.B.           RCALL UP1  
entspricht:   CALL 0F00FH  
              DA UP1-NEXT  
NEXT:       (nächster Befehl)  
Das DE-Doppelregister wird nicht übergeben.  
Stacktiefe: 1

**E000H: Reset-Adresse:**

Diese Adresse wird beim Tastenreset angesprungen. Der Systemarbeitsspeicher wird neu initialisiert, der Anwenderspeicher bleibt erhalten.

Die Programmverteiler I bis IV realisieren den Unterprogrammaufruf über eine Tabelle der Anfangsadressen dieser Unterprogramme. Die Anfangsadresse der Tabelle steht in der Speicherzelle ("SUTAB" (vgl. 6.1)). Soll diese Tabelle verändert werden, so muß folgendermaßen vorgegangen werden:

1. Bestimmen der Anfangsadresse aus "SUTAB"
2. Umspeichern der Tabelle in den RAM in der Länge 2 x Anzahl der UP-Nr.
3. Ergänzen/Ändern
4. Eintragen der neuen Anfangsadresse in "SUTAB"

Die Liste der Unterprogramme ist Anlage 1 zu entnehmen.

## 6. SYSTEMARBEITSZELLEN

### 6.1. ARBEITSZELLEN IM IRM

Adresse	Name	Länge in Bytes	Inhalt
B780	ARGC	1	UP-Nr. bei Programmverteiler II (bzw. intern)
B781	ARGN	1	Anzahl der Argumente bei Kommandoeingabe
B782	ARG1	2	1. Argument
B784	ARG2	2	2. Argument
B786	ARG3	2	3. Argument
B788	ARG49	12	4. bis 9. Argument
B794	ARG10	2	10. Argument
B796	NUMNX	1	Anzahl der Zeichen erfaßter HEX-Zahl
B797	NUMVX	2	Wert der erfaßten HEX-Zahl
B799	HCADR	2	Adresse für Hardcopyprogramm Aufruf über Tastatur; Code 0FH (SHIFT INS); DE enthält Cursorposition
B79B	WINNR	1	Nr. des aktuellen Bildschirmfensters
B79C	WINON	2	Fensteranfang: L: Spalte (0...39 BZW. 0H...27H) H: Zeile (0...31 BZW. 0H...1FH)
B79E	WINLG	2	Fenstergröße: L: 0 < Spaltenzahl < 40 - L(WINON) H: 0 < Zeilenzahl < 32 - H(WINON)
B7A0	CURSO	2	Relative Cursorposition im Fenster: L - Spalte H - Zeile
B7A2	STBT	1	Steuerbyte für Bildschirmprogramm: BIT 0 = 0 Schreiben Zeichen ein = 1 Schreiben Zeichen aus  BIT 1 = 0 Schreiben Farbe aus = 0 Schreiben Farbe ein  BIT 3 = 0 Ausführen des Steuercodes (0 = 1FH) = 1 Interpretieren des Steuercodes als Zeichen auf Bildschirm

B7A3	COLOR	1	Farbbyte für Bildschirmprogramm																
			<table border="0" style="margin-left: 40px;"> <tr> <td>B(7)</td><td>B(6)</td><td>B(5)</td><td>B(4)</td><td>B(3)</td><td>B(2)</td><td>B(1)</td><td>B(0)</td> </tr> <tr> <td>A(V)</td><td>X(V)</td><td>G(V)</td><td>R(V)</td><td>B(V)</td><td>G(H)</td><td>R(H)</td><td>B(H)</td> </tr> </table> <p style="margin-left: 40px;">Index V: Vordergrund (Farbe für Bit im Pixel-RAM = 1)  H: Hintergrund (Farbe für Bit im Pixel-RAM = 0)  R: Farbe ROT  G: Farbe GRÜN  B: Farbe BLAU  X: Farbverschiebung im Farbkreis um 30°  A: Alternierende Zeichendarstellung (Blinken der Vordergrundfarbe)  Durch Kombination der Bits ergeben sich Mischfarben.</p>	B(7)	B(6)	B(5)	B(4)	B(3)	B(2)	B(1)	B(0)	A(V)	X(V)	G(V)	R(V)	B(V)	G(H)	R(H)	B(H)
B(7)	B(6)	B(5)	B(4)	B(3)	B(2)	B(1)	B(0)												
A(V)	X(V)	G(V)	R(V)	B(V)	G(H)	R(H)	B(H)												
B7A4	WEND	2	Anfangsadresse des Reaktionsprogrammes auf Erreichen des Fensterendes (z.B. Page-, Scrollmodus usw.)																
B7A6	CCTL0	2	Adresse der Zeichenbildtabelle für Codes 20H - 5FH																
B7A8	CCTL1	2	Adresse der Zeichenbildtabelle für 00H - 1FH und 60H - 7FH																
B7AA	CCTL2	2	Adresse der Zeichenbildtabelle für 80H - 9FH und E0H bis EFH																
B7AE	SYSP	2	Init-Adresse des Systemstackpointers (normal 01D4H)																
B7B0	SUTAB	2	Adresse der Unterprogrammtabelle																
B7B2	CTAB	2	Tabelle der Stammcodes für das Bildschirmprogramm																
B7B9	OUTAB	2	Adresse für Zeiger auf UP-Nr. für Ausgabekanal (normal Bildschirm)																
B7BB	INTAB	2	Adresse für Zeiger auf UP-Nr. für Eingabekanal (normal Tastatur)																
B7BD	UOUTI	3	Sprung in USER-Ausgabekanal 1 (z.B. BASIC #2)																
B7C0	UINI	3	Sprung in USER-Eingabekanal 1 (z.B. BASIC #2)																
B7C3	UOUT2	3	Sprung in USER-Ausgabekanal 2 (z.B. BASIC #3)																

B7C6	UIN2	3	Sprung in USER-Eingabekanal 2 (z.B. BASIC #3)
B7D3	HOR	2	X-Wert für Grafikprogramm (0-319)
B7D5	VERT	1	Y-Wert für Grafikprogramm (0-255)
B7D6	FARB	1	Vordergrundfarbe/Blinken für Grafikprogramm (vgl. COLOR)
B7D7	MIXIT	1	höherwertiger Teil von IX und der Inter- rupttabelle (vgl. Abschn. 6.2.)

## 6.2. ARBEITSZELLEN IM IX-BEREICH

Das IX-Register wird beim RESET/Einschalten auf 01F0H geladen, kann aber, falls dieser Speicherbereich benötigt wird, umgeladen werden, wobei der niederwertige Teil erhalten bleibt.

Wichtige Arbeitszellen:

Kassetten Ein-/Ausgabe

IX + 2 : Blocknummer bei Kassetten-Ein-/Ausgabe  
IX + 5 : Pufferadresse für Kassetten-Ein-/Ausgabe  
IX + 6 : (normal B700H)  
IX + 7 : Bit 0 = 0 - Verify  
          = 1 - Read

Tastatureingabe

IX + 8 : Bit 0 = 1 Tastencode steht zur Verfügung:  
          Übernahmequittierung mit "RES 0.(IX+8)"  
          Bit 7 = 1 Shiftlock  
IX + 13 : Tastaturcode (ASCII)  
IX + 14 : Low Tastaturcodetabelle KTAB  
IX + 15 : High Tastaturcodetabelle KTAB

## 6.3. INTERRUPTTABELLE

Das I-Register der CPU wird beim RESET/Einschalten auf 01 gesetzt, kann aber auch umgeladen werden (vgl. Abschn. 6.2).

01E4	Interrupt PIO-Kanal A - Kassetteneingabe
01E6	Interrupt PIO-Kanal A - Tastatur
01E8	Interrupt CTC-Kanal 0 - frei
01EA	Interrupt CTC-Kanal 1 - Kassettenausgabe
01EB	Interrupt CTC-Kanal 2 - Tondauer
01EC	Interrupt CTC-Kanal 3 - Tastatur
01D4 - 01E3	frei für Anwender-Interrupt-Tabellen

#### 6.4. KELLERSPEICHER

Der Stackpointer (SP) wird beim Einschalten/RESET auf 01D4H gesetzt, kann aber auf jeden anderen freien Speicher gelegt werden.

#### 6.5. VERÄNDERN DES ARBEITSSPEICHERBEREICHES

Im Folgenden soll an einem Beispiel erläutert werden, wie der Arbeitsspeicherbereich im RAM-Block (Stack, Interrupttabellen, IX-Bereich) auf das Ende des RAM-Bereiches verlagert werden kann:

```
DI                ; sperren Interrupt
LD   SP,3FC4H     ; 32 Bytes freihalten
                        ; für USER-Interrupttabelle
LD   (0B7AEH),SP ; merken Stackanfang
LD   A,3FH        ; höherwertiger Teil IX- und I-Register
LD   E,31H        ; UP-Nr. 31H SIXC (vgl. Abschn. 7
                        ; und Anhang)
CALL F009H        ; Verteiler III (vgl. Abschn. 5)
EI                ; Freigabe Interrupt
```

Eine Verlagerung in den IRM ist prinzipiell auch möglich, führt aber ständig zu sichtbaren Speicherzugriffen.

#### 6.6. ZEICHENBILDTABELLEN

Zur Ergänzung des internen Zeichenbildvorrats (Großbuchstaben, Ziffern, Sonderzeichen - Codes  $\geq 0H - 5FH$ ) können eigene Zeichenbildtabellen (Zeichengeneratoren) erstellt werden. Je Zeichen werden 8 Bytes benötigt:

Bildpunkte = Bits : seitenrichtig, nicht negiert, oberste Bildpunktzeile = niedrigste Adresse. Die Anfangsadressen der Tabellenzeiger müssen entsprechend den zugehörigen Codes in die Speicherzellen CCTL0 - CCTL3 eingetragen werden. In der Grundinitialisierung ist in allen vier Adressen ein Zeiger auf die interne Zeichenbildtabelle im ROM eingetragen (Adresse 0E600H).

Beispiel 1:

Für die Zeichen mit dem Code 0A0H - 0DFH sollen spezielle Zeichen definiert werden. Die Zeichenbildtabelle wird im Speicherbereich 0BC00H - 0BDFFH abgelegt.

1. Umschalten des Zeigers auf die Zeichenbildtabelle

```
MODIFY B7AAH
B7AA 00
B7AB E6 ändern in BC.
```

2. Generieren eines neuen Zeichens mit dem Code 0A0H

Zeichenbild	Bild-Code	HEX-Code	z.B. griechisches Omega "Ω"
-----	0000	0000 00	)
XXX	0011	1000 38	)
X X	0100	0100 44	)
X X	1000	0010 82	) 8 Bytes
X X	0100	0100 44	)
X X	0010	1000 28	)
XX XX	0110	1100 6C	)
	0000	0000 00	)

Wenn der HEX-Code ab Adresse BE00H mit MODIFY abgelegt wird, wird das Zeichen 0A0H mit diesem Bild so auf dem Bildschirm dargestellt.

Beispiel 2:

Für die Zweitbelegung der Tastatur sollen Kleinbuchstaben generiert werden. Dazu wird die im ROM vorhandene Zeichenbildtabelle mit dem Kommando COPY (vgl. Abschn. 4) in den RAM ab Adresse BE00H eingeladen.

Der Buchstabe "a" hat den Code 61H und ist in der Zeichenbildtabelle das 34. Zeichen (=22H) (00H - 1FH = 32 Zeichen, 60H = 33. Zeichen, 61H = 34. Zeichen).

Da für jedes Zeichen 8 Bytes reserviert werden, berechnet sich die Adresse für "a" wie folgt:

$$\text{Adresse Zeichenbild} = \text{Anfangsadresse der Tabelle} + 8 * (\text{Stellung in Tabelle} - 1)$$

$$\text{Adresse} = \text{BE00H} + 8 * (22\text{H} - 1) = \text{BF08H}.$$

Auf die Adressen BF08H...BF0FH wird also eingetragen:

BF08 00	0000 0000	
BF09 00	0000 0000	
BF0A 00	0000 0000	
BF0B 16	0001 1100	XXX
BF0C 24	0010 0100	X X
BF0D 24	0010 0100	X X
BF0E 1E	0001 1110	XXXX
BF0F 00	0000 0000	

Nun muß noch der Zeiger in CCTL1 geändert werden:

```
B7A8 00
B7A9 8E
```

### 6.7. STEUERCODETABELLE

In der Speicherzelle "CTAB" /vgl. Abschn. 6.1) ist ein Zeiger auf eine Programmverteiltertabelle abgelegt, welche die Zuordnung der Steuercodes zu den einzelnen Bildschirmprogrammsteuerfunktionen organisiert. Die Zuordnung ist in folgender Reihenfolge realisiert:

Nr.	Funktion	Code
1	PAGE	11H
2	SCROL	12H
3	HOME	10H
4	CUR	9
5	CUU	0BH
6	ESC	2
7	CCR	19H
8	INS	1AH
9	CUD	0AH
10	DEL	1FH
11	CUL	8
12	CLS	0CH
13	CR	0DH 1)
14	HCOPY	0FH
15	BEEP	7
16	CLR	1

1) Nr. 13 (NEWLINE) ist funktional identisch zu Nr. 9 (Cursor Down) bei der Bearbeitung durch das Bildschirmprogramm.

Soll die Zuordnung der Steuercodes geändert werden, muß eine neue Tabelle gleichen Aufbaus im RAM abgelegt werden, und ihre Anfangsadresse muß in "CTAB" eingetragen werden.

## 7. FUNKTIONSTASTEN

Die F-Tasten liefern von den Tastaturprogrammen KBDS, KBDZ folgende Codes:

Taste	Code: 1. Belegung	2. Belegung
F1	F1H	F7H
F2	F2H	F8H
F3	F3H	F9H
F4	F4H	FAH
F5	F5H	FBH
F6	F6H	FCH

Beim Betätigen einer F-Taste wird vom Tastaturprogramm KBD die Zeichenübergabe auf Zeichen aus dem zugehörigen Puffer (ab B900H) umgeschaltet. Der Pufferaufbau ist dynamisch, d.h., die Zeichenanzahl zu den einzelnen F-Tasten liegt nicht fest, sondern wird nur von der Puffergröße begrenzt. Der Puffer darf maximal 128 Bytes betragen (Adresse B900H ... B97FH). Der Puffer muß mit 00 beginnen und mit 00 abgeschlossen werden. Die Zeichenketten für die einzelnen F-Tasten werden ebenfalls durch ein 00-Byte getrennt. Es sind als Code alle Codierungen zugelassen.

Bei Betätigen einer F-Taste werden die Trenn-Nullen vor und nach der eigentlichen Zeichenfolge mit übergeben, aber vom Bildschirmprogramm ignoriert. Eine nicht belegte F-Taste liefert also zweimal den Code 00.

Dabei ist es möglich, auf den F-Tasten "Jobs" abzulegen, deren Abarbeitung mittels BRK-Taste abgebrochen werden kann.

Beispiel:

Mit der Taste F1 soll die Zeichenkette "MENU" und mit F2 die Kette COLOR erzeugt werden (vgl. Kommando "MODIFY").

```
MODIFY B900 <ENTER>
B900 00 <ENTER> Startzeichen
B901 ,M,E,N,U <ENTER> Zeichenkette für F1
B905 0D <ENTER> ENTER-Taste
B906 00 <ENTER> Trennzeichen zwischen F1 und F2
B907 ,C,O,L,O,R <ENTER> Zeichenkette für F2
B90C 00 <ENTER> Trennzeichen zwischen F2/F3
B90D 00 <ENTER> Trennzeichen zwischen F3/F4
...
...
...
B914 00 <ENTER> Trennzeichen zwischen FB/FC
B915 00 <ENTER> Endezeichen
```

## 8. MAGNETBANDAUFZEICHNUNG

### 8.1. VERFAHREN

Die Aufzeichnung auf Kassette erfolgt nach einem neuentwickelten Verfahren, welches Vorteile bezüglich Übertragungsrates und Synchronisation gegenüber bekannten Verfahren bietet. Zur Aufzeichnung dienen drei verschiedene Frequenzen, wobei jeweils eine komplette Schwingung für eine logische Einheit dient:

Nullbit:            f = 1200 Hz

Einsbit:            f = 2400 Hz

Trennzeichen:    f = 600 Hz

Byteaufbau:        8 Datenbits (je 0 oder 1 - Bit)  
                  1 Trennzeichen  
                  mit Bit 0 beginnend

Blockaufbau - Vorton: aus Schwingungen mit 1200 Hz (Einsbit) bestehend

- erster Block langer Vorton, etwa 8000 Schwingungen
- folgende Blöcke je nach Verarbeitungszeit (für MC-Programme 160 Schwingungen)
- 1 Trennzeichen
- 1 Byte Block-Nr.. vgl. Punkt 8.2
- 128 Datenbytes            vgl. Punkt 8.2
- 1 Byte Datensumme vgl. Punkt 8.2

### 8.2. DATEIAUFBAU

Die Daten werden auf dem Magnetband als sequentielle Datei mit Blöcken zu je 128 Datenbytes abgespeichert.

Jeder Block besteht aus

- 1. Byte: Blocknummer (erster Block: Nr. 01; folgende Blöcke aufsteigend numeriert; letzter Block Nr. FFH)
- 2. bis 128. Byte: Daten
- 130. Byte: Prüfsumme über die Daten

Jede Datei besteht aus einem Vorblock (Block Nr. 01) und nachfolgenden Datenblöcken.

Der Vorblock ist wie folgt aufgebaut:

1. - 8. Byte: Name, besteht aus alphanumerischen Zeichen
9. - 11. Byte: Dateityp, vgl. Abschnitt 8.3
12. - 16. Byte: Reservierte Bytes für Hersteller  
für Anwenderprogramme müssen diese 00 enthalten
17. Byte: Anzahl der nachfolgenden 2-Byte-Argumente, für ladbare Maschinenprogramme und Speicherabzüge (DUMP) muß dieses Byte einen Wert zwischen 02H und 0AH enthalten.  
Dafür gilt:  
Byte = 02H: Programm wird geladen, danach Rückkehr in das aufrufende Programm.  
Byte = 03H: Programm wird geladen, danach Start des Programms bei angegebener Startadresse. Wird das Programm relativ geladen, so erfolgt der Start bei umgerechneter Startadresse.  
Byte = 04H...07H: Wie bei Byte = 03H, jedoch ohne Umrechnung der Startadresse beim relativen Laden.

Die im Vorblock enthaltenen restlichen Datenbytes enthalten Parameter zur genaueren Definition der Datei.

Für Maschinenprogramme und Speicherabzüge gilt folgende Festlegung:

- 18./19. Byte: Ladeadresse
- 20./21. Byte: Endadresse + 1
- 22./23. Byte: Startadresse

### 8.3. DATEITYPEN

Im 9. bis 11. Byte des Vorblockes ist der Dateityp anzugeben. Dafür gelten folgende Festlegungen:

- COM = Maschinenprogramm
- DUM = Speicherabzüge
- TXT = Textdateien
- ASM = Quelltextdateien für Assemblerprogramme
- LBL = Markentabellen von Übersetzern (z.B. Assembler)
- (F) = Forth-Quellprogramme

## **9. TASTENCODES**

Der Tastencode wird über eine Tabelle (KTAB vgl. Punkt 6.1) aus den seriellen Impulsfolgen der Fernsteuer-IS U807D gewonnen. Eine Änderung der Codes zu den einzelnen Tasten ist durch Aufbau einer neuen Umcodierungstabelle und eintragen deren Anfangsadresse in KTAB möglich. Diese Tabelle umfaßt 128 Bytes, wobei jeder Taste 2 Bytes zugeordnet sind:

1. Byte Erstbelegung der Taste
2. Byte Zweitbelegung (über SHIFT)

Anlage 3 enthält die Beschreibung der Steuercodes, Anlage 4 und Bild 4 enthalten die Steuercodes.

## 10. SPEZIELLE SYSTEMBEDINGUNGEN

- Das IX-Register wird für die Adressierung der Tastatur/Kassetten-Interruptprogramme benötigt und darf bei freigegebenem Interrupt nicht verändert werden.
- Es ist IM2 vorgeschrieben.
- IX-Register und Interrupttabelle können über das Unterprogramm SIXD (UP-Nr. 31) auf andere Speicherbereiche gelegt werden.
- Die I/O-Adresse 80H ist für die Modulsteuerung reserviert.
- Interne I/O-Adressen: 88H - 8BH PIO  
8CH - 8FH CTC
- Für den Anwender stehen die I/O-Adressen 0C0H...0FFH zur Verfügung, die anderen I/O-Adressen sind für Module bzw. Aufsätze des Herstellers reserviert.
- Bei Anwenderprogrammen, welche mit eigenem Stackbereich arbeiten und mit Zusatz-RAM-Modulen den Speicherbereich 8000H - BFFFH bei abgeschaltetem IRM nutzen, ist es notwendig, entweder den Stack in den Bereich < 8000H zu legen oder vor Aufruf des Betriebssystems den Stackpointer in diesen Bereich zu verlegen und den IRM einzuschalten.

## 11. ANLAGE 1

### UNTERPROGRAMME DES BETRIEBSSYSTEMS HC-CAOS

Aufruf der CAOS-UP über Programmverteiler (PV)

**PV1:** (Adresse 0F003H)

Die UP-Nr. wird nach dem CALL definiert.

Bsp.:           CALL    0F003H  
                  DB      UP-Nr.

**PV2:** (Adresse 0F006H)

Die UP-Nr. wird im RAM auf (ARGC) übergeben; Register werden gerettet.

**PV3:** (Adresse 0F009H)

Die UP-Nr. wird im Register E übergeben.

**PV4:** Wie PV3, jedoch mit Ein-/Ausschalten des IRMs.

### Legende

Name:           Name des UPs

UP-Nr.          Nummer des UPs

Fkt.:           Beschreibung der Funktion

PE:            Parameterübergabe vor dem Programmaufruf

PA:            Parameterübergabe nach RETURN des UPs

VR:            Veränderte Register

Stack:         Stack-Tiefe des UPs

In Klammern stehende Werte entsprechen RAM-Speicherzellen.

Name: **CRT** UP-Nr.: 00H  
Fkt.: Zeichenausgabe auf den Bildschirm  
PE: Register A = Zeichencode (ASCII); Stack = 16

---

Name: **MBOT** UP-Nr.: 01H  
Fkt.: Ausgabe eines Datenblocks auf Kassette  
PE: Register BC = Länge Vorton  
(IX+5) = L (Pufferadresse)  
(IX+6) = H (Pufferadresse)  
PA: Register HL = Pufferende + 1  
(IX+2) = Block-Nr.  
VR: AF,BC,DE,HL Stack: 3

---

Name: **OUT1** UP-Nr.: 02H  
Fkt.: Ausgabe auf Anwenderkanal 1  
PE: Register A = Zeichencode  
PA/VR = entsprechend der Routine

Bemerkung: Adresse der selbstzuerstellenden Routine muß auf  
(OUT1) eingetragen werden.

---

Name: **OUT2** UP-Nr.: 03H  
Fkt./P = vgl. OUT1, Adresse der Routine muß auf (OUT2) einge-  
tragen werden.

---

Name: **KBD** UP-Nr.: 04H  
Fkt.: Tasteneingabe mit Einblendung des Cursors, wartet bis  
Taste gedrückt, bzw. liefert Codefolge vorher betätigter  
F-Taste  
PA: Register A = Zeichencode (ASCII)  
VR: AF, HL Stack: 7

---

Name: **MBI** UP-Nr.: 05H  
Fkt.: Einlesen eines Datenblocks von der Kassette in den  
Puffer (128 Bytes)  
PE: (IX+5) = L (Pufferadresse)  
(IX+6) = H (Pufferadresse)  
PA: CY = 1 = Block fehlerhaft  
(IX+2) = Block-Nr.  
VH: AF Stack: 4

---

Name: **USIN1** UP-Nr.: 06H  
Fkt.: Eingabe Anwenderkanal 1  
Bem.: Adresse des selbstzuerstellenden Programms muß in (UIN1)  
eingetragen werden.

---

Name: **USIN2** UP-Nr.: 07H  
Fkt.: vgl. USIN1, Adresse der Routine muß in (UIN2) eingetra-  
gen werden.

---

Name: **ISRO** UP-Nr.: 08H  
Fkt.: Initialisierung der Magnetbandausgabe, Ausgabe des 1.  
Blockes (Block-Nr. 01H)  
P.: vgl. MBOT; UP-Nr.: 01H; Stack: 4

---

Name: **CSRO** UP-Nr.: 09H  
Fkt.: Abschluß-(CLOSE)-Routine für Magnetbandausgabe, Ausgabe  
des letzten Blocks (Block-Nr.: 0FFH)  
P.: vgl. MBOT; Stack: 3

---

Name: **ISRI** UP-Nr.: 0AH  
Fkt.: Initialisierung der Magnetbandeingabe, Einlesen des 1.  
Blockes  
P.: vgl. MBIN; Stack: 4

---

Name: **CSRI** UP-Nr.: 0BH  
Fkt.: Abschluß der Magnetbandeingabe  
P.: keine  
VR: AF, HL Stack: 8

---

Name: **KBDS** UP-Nr.: 0CH  
Fkt.: Tastaturabfrage ohne Quittierung der Taste  
PA: CY = 1 --> Taste gedrückt, dann  
Register A = Zeichencode (ASCII)  
VR: AF Stack: 0

---

Name: **BYE** UP-Nr.: 0DH  
Fkt.: Sprung auf RESET  
Bem.: Adresse 0E000H

---

Name: **KBDZ** UP-Nr.: 0EH  
Fkt.: Tastenstatusabfrage mit Quittierung der Taste  
(Autorepeat)  
PA: CY = 1 --> Taste gedrückt, dann  
Register A = Zeichencode (ASCII)  
VR: AF Stack: 1

---

Name: **COLOR** UP-Nr.: 0FH  
Fkt.: Farbe einstellen  
PE: Register E = Hintergrundfarbe (0...7)  
Register L = Vordergrundfarbe (0...1F)  
(ARGN) = 1 = nur Vordergrundfarbe  
2 = Vorder- und Hintergrundfarbe  
VR: AF, L Stack: = 0

---

Name: **LOAD** UP-Nr.: 10H  
Fkt.: Einlesen von Maschinenprogrammen von Kassette  
PE: (ARGN) = 0 Load ohne Offset  
= 1 Load mit Offset  
(ARG1) = Ladeoffset  
VR: AF, BC, DE, HL Stack: 18

---

Name: **VERIF**           UP-Nr.: 11H  
Fkt.: Überprüfen von Kassettenaufzeichnungen  
VR:   AF,BC,DE,HL   Stack: 18

---

Name: **LOOP**            UP-Nr.: 12H  
Fkt.: Rückgabe der Steuerung an CAOS ohne Speicherinitialisierung

---

Name: **NORM**            UP-Nr.: 13H  
Fkt.: Rückschalten des Ein- und Ausgabekanals auf CRT und KBD  
PA:   Register HL = alter Ausgabezeiger  
VR:   HL            Stack: 2

---

Name: **WAIT**            UP-Nr.: 14H  
Fkt.: Warteschleife  
PE:   Register A    T = A\*6ms  
VR:   AF, B         Stack: 1

---

Name: **LARG**            UP-Nr.: 15H  
Fkt.: Lade Register mit Argumenten  
PA:   Register HL = (ARG1)  
      Register DR = (ARG2)  
      Register BC = (ARG3)  
VR:   BC, DE, HL   Stack: 0

---

Name: **INTB**            UP-Nr.: 16H  
Fkt.: Eingabe eines Zeichens vom aktuellen Eingabekanal [über (INITTAB) definiert].  
PA:   Register A = Zeichencode (ASCII)

---

Name: **INLIN** UP-Nr.: 17H  
Fkt.: Eingabe einer Zeile mit Funktion aller Cursortasten,  
Abschluß mit <ENTER>  
PA: Register DE = Adresse des Zeilenanfangs  
VR: AF, DE Stack: 20

---

Name: **RMEX** UP-Nr.: 18H  
Fkt.: Umwandlung einer Zeichenkette (Hexadezimalzahl) in in-  
terne Darstellung  
PE: Register DE = Anfangsadresse der Zeichenkette  
PA: Register DE = Ende der Zeichenkette  
(NUMNX) = Länge der Zeichenkette  
(NUMVX) = umgewandelte Zahl  
CY = 1 = Fehler (Zeichenkette enthält falsche Hexazif-  
fern, Länge zu groß usw.)  
VR: AF, DE, HL Stack: 0

---

Name: **ERRM** UP-Nr.: 19H  
Fkt.: Ausschrift des Textes "ERROR"  
VR: - Stack: 18

---

Name: **HLHX** UP-Nr.: 1AH  
Fkt.: Ausgabe des Wertes des Registers HL als Hexazahl  
PE: Register HL  
VR: - Stack: 20

---

Name: **HLDE** UP-Nr.: 1BH  
Fkt.: Ausgabe der Register HL und DE als Hexazahlen  
PE: Register HL, Register DE  
VR: AF Stack: 22

---

Name: **AHEX** UP-Nr.: 1CH  
Fkt.: Ausgabe Register A als Hexazahl  
PE: Register A  
VR: A Stack: 20

---

Name: **ZSUCH** UP-Nr.: 1DH  
Fkt.: Suche nach Zeichenkette (Menüwort)  
PE: Register A = Prolog (für CAOS-Menü: 7FH)  
Register BC = Länge des Suchbereiches  
Register DE = Anfang der Vergleichskette  
Register HL = Anfang des Suchbereiches  
PA: Register DE = ENDE+1 Vergleichskette  
Register HL = ENDE+1 gefundene Kette  
CY = 1 = Kette gefunden  
VR: AF, BC, DE, HL Stack: 3

---

Name: **SOUT** UP-Nr.: 1EH  
Fkt.: Setze neuen Zeiger auf Ausgabetabelle; auf Adresse (HL)  
steht neue UP-Nr.  
PE: Register HL = neuer Zeiger auf OUTAB  
PA: Register HL = alter Zeiger  
VR: HL Stack: 1

---

Name: **SIN** UP-Nr.: 1FH  
Fkt.: Setze neuen Zeiger auf Eingabetabelle; auf Adresse (HL)  
steht UP-Nr.  
PE: Register HL = neuer Zeiger auf INTAB  
PA: Register HL = alter Zeiger  
VR: HL Stack: 1

---

Name: **NOUT** UP-Nr.: 20H  
Fkt.: Setze Zeiger für Ausgabe auf Normalausgabe (CRT)  
PA: Register HL = alter Zeiger  
VR: HL Stack: 1

---

Name: **NIN** UP-Nr.: 21H  
Fkt.: Setze Zeiger für Eingabe auf KBD  
PA: Register HL = alter Zeiger  
VR: HL Stack: 1

---

Name: **GARG** UP-Nr.: 22H  
Fkt.: Erfassen von maximal 10 Hexazahlen und Wandlung in die interne Darstellung  
PE: Register DE = Adresse des ersten Zeichens  
PA: Register DE = Adresse des letzten Zeichens+1  
(ARGN) = Anzahl der erfaßten Zahlen  
(ARG1)...(ARG10) = Werte der Zahlen  
CY = 1 bei Fehler  
VR: AF, BC, DE, HL Stack: 1  
Bem.: zulässige Ziffern in Zeichenkette 0...9,A...F

---

Name: **OSTR** UP-Nr.: 23H  
Fkt.: Ausgabe einer Zeichenkette, die nach UP-Aufruf steht, Abschluß mit 00H  
VR: AF Stack: 22  
Bsp.: CALL F003  
DB 23H ; UP-Nr.: OSTR  
DB "ERROR" ; Ausgabe "ERROR"  
DA 0D0AH ; Newline  
DB 0 ; Ende

---

Name: **OCHR** UP-Nr.: 24H  
Fkt.: Zeichenausgabe an Gerät, das über Ausgabetabelle eingestellt werden kann (vgl. UP-Nr. 1EH, 20H)  
PE: RG, A = Zeichencode (ASCII)  
VE: AF Stack: 21

---

Name: **CUCP** UP-Nr.: 25H  
Fkt.: Komplementiere Cursor  
PE: (CURSO) = Cursorposition  
VR: - Stack: 8

---

Name: **MODU** UP-Nr.: 26H  
Fkt.: Modulsteuerung  
- Lesen des Modultyps, wenn Register A  $\leq$  1  
- Aussenden des Steuercodes und Eintragung in den Modul-  
Steuerwort-Speicher, wenn RG, A  $\geq$  2  
PE: RG, A - Anzahl der Parameter: = 1 - RG, L  
= 2 - RG, D und L  
RG, L - Modulsteckplatz  
RG, E - Modulsteuerbyte

PA: RG, H - Modultyp  
RG, E - Modulsteuerbyte  
VR: AF, H Stack: 2

---

Name: **JUMP** UP-Nr.: 27H  
Fkt.: Sprung in neues Betriebssystem, Abschalten von CAOS und  
Eintragung des Steuerwortes FFH in den Modul-Steuerwort-  
Speicher  
PE: RG, A - Modulsteckplatz  
Bem.: Startadresse neues Betriebssystem auf 0F012H

---

Name: **LDMA** UP-Nr.: 28H  
Fkt.: LD (HL),A ; (für Aufruf über PV IV)  
PE: RG, A - Byte  
RG, HL - Adresse  
VR: - Stack: 0

---

Name: **LDAM** UP-Nr.: 29H  
Fkt.: LD A,(HL) ; (für Aufruf über PV IV)  
PE: RG, HL - Adresse  
PA: RG, A - Byte auf ADR, (HL)  
VR: A Stack: 0

---

Name: **BRKT** UP-Nr.: 2AH  
Fkt.: Test auf Unterbrechnungsanforderung (Betätigung BRK-  
Taste)  
PA: CY = 1 Taste gedrückt  
RG, A - Tastencode  
Stack: 1

---

Name: **SPAC** UP-Nr.: 2BH  
Fkt.: Ausgabe eines Leerzeichens über UP-Nr.: 24H  
VR: AF Stack: 18

---

Name: **CRLF** UP-Nr.: 2CH  
Fkt.: Ausgabe von "Newline" (Codes 0AH und 19H)  
VR: AF Stack: 18

---

Name: **HOME** UP-Nr.: 2DH  
Fkt.: Ausgabe des Steuerzeichens "HOME" (Code 10H)  
VR: AF Stack: 18

---

Name: **MODI** UP-Nr.: 2EH  
Fkt.: Aufruf des Systemkommandos "MODIFY"  
PE: RG, HL - Anfangsadresse Stack: 24

---

Name: **PUDE** UP-Nr.: 2FH  
Fkt.: Löschen eines Bildpunktes  
PE: (HOR) - Horizontalkoord. (0...13H)  
(VERT) - Vertikalkoord. (0...FFH)  
PA: RG, A - Farbbyte  
CY = 1 - Punkt außerhalb (Fehler)  
Z = 1 - Punkt war gesetzt  
VR: AF Stack: 7  
Bem.: (HOR) = (VERT) = 0 entspricht linker unterer Ecke

---

Name: **PUSE** UP-Nr.: 30H  
Fkt.: Setzen eines Bildpunktes  
PE: (HOR) - Horizontalkoord. (0...13H) (0...19)  
(VERT) - Vertikalkoord. (0...FFH) (0...255)  
(FARB) - Bildpunktfarbe (0...1FH)  
PA: CY = 1 - Punkt außerhalb (Fehler)  
VR: AF Stack: 7

---

Name: **SIXD** UP-Nr.: 31H  
Fkt.: Verlagerung des Arbeitsbereiches von CAOS  
- Initialisierung Interrupttabelle  
- Initialisierung RG, IX  
- Setzen IM2  
- Initialisierung PIO, CTC  
PE: RG, A - höherwertiger Adreßteil  
PA: (MIXIT) - höherwertiger Adreßteil  
VR: AF, BC, DE, HL, IX Stack: 5

---

Name: **DABR** UP-Nr.: 32H  
Fkt.: Berechnung Video-RAM-Adresse aus der Cursorposition  
PE: RG, D - Zeile auf dem Bildschirm  
E - Spalte auf dem Bildschirm  
PA: CY = 1 - außerhalb (Fehler)  
HL = Adresse im Speicher  
VR: AF, BC, HL Stack: 4

---

Name: **TGIF** UP-Nr. 33H  
Fkt.: Test, ob Cursorposition im definierten Fenster  
PE: RG, D - Zeile der Cursorposition  
E - Spalte der Cursorposition  
PA: CY = 1 - Cursor außerhalb  
VR: AF Stack: 0

---

Name: **PADR** UP-Nr.: 34H  
Fkt.: Berechne Pixel- und Farbadresse aus Position  
PE: RG, H - Vertikalposition (0...FFH)  
L - Horizontalposition (0...27H)  
PA: RG, DE - Farbadresse  
RG, HL - Zeichenadresse  
CY = 1 - außerhalb  
VR: F, HL, DE Stack: 2

**!!! Bemerkung: Aufruf nur über Adresse 0F003H möglich !!!**  
HL = 00 entspricht linker oberer Ecke

---

Name: **TON** UP-Nr.: 35H  
Fkt.: Tonausgabe  
PE: (ARG1) - Tonhöhe 1 (Zeitkonstante für CTC0)  
(ARG1+1) - Vorteiler 1 (0, 1) <Systemstart 16 bzw. 256>  
(ARG2) - Tonhöhe 2 (CTC1)  
(ARG2+1) - Vorteiler 2 (0, 1)  
(ARG3) - Lautstärke (0...1FH)  
(ARG3+1) - Tondauer (0...FFH)  
<in 20ms-Schritten bzw. 0 Dauerton>  
VR: AF, BC, DE, HL Stack: 7

---

Name: **SAVE** UP-Nr.: 36H  
Fkt.: Ausgabe von Maschinenprogrammen auf Kassette  
PE: RG, HL - Anfangsadresse des Dateinamens  
(ARG1) - Anfangsadresse des Programms  
(ARG2) - Endadresse des Programms  
(ARG3) - Startadresse des Programms  
(ARGN) - Anzahl der Parameter  
(2 - ARG1, ARG2)  
(3 - ARG1...ARG3 bei selbststartenden Programmen)

**12. ANLAGE 2**

**Adreßzuordnungen IRM (Pixel-Farbe)**

Zuordnung IRM-Adresse (Adr.-Bits 0 ... 15)

und Bildschirmposition: horizontal Bits  $h_0 \dots h_5$

vertikal Bits  $V_0 \dots V_7$

Bildschirmzeile -  $V_0 \dots V_7$  - Pixelzeilennummer

Bildschirmspalte -  $h_0 \dots h_5 \leq 1FH$

= Zeichenspaltennummer,  $\leq 31$

$h_0 \dots V_5 \geq 20H$

= Zeichenspaltennummer,  $\geq 32 \dots 39$

**Adresse im Video-RAM (ASCH - Puffer)**

=  $B200H + \text{Zeichenspalte} + 40 * \text{Zeichenzeile}$

=  $B200H + \text{Zeichenspalte} + 5 * \text{Pixelzeile}$

Farbauflösung: 4 Pixelzeile = 1 Farbbyte

(bei Farbadressen  $V_0 + V_1 = 0$ )

**Adreßzuordnungstabelle**

Adreßbits	Position bis 31. Zeichen		32. - 39. Zeichen	
	Pixelbyte	Farbbyte	Pixelbyte	Farbbyte
15	1	1	1	1
14	0	0	0	0
13	0	1	1	1
12	$V_7$	0	0	1
11	$V_6$	1	0	0
10	$V_5$	$V_7$	$V_7$	0
9	$V_4$	$V_6$	$V_6$	0
8	$V_1$	$V_5$	$V_1$	$V_7$
7	$V_0$	$V_4$	$V_0$	$V_6$
6	$V_3$	$V_3$	$V_3$	$V_3$
5	$V_2$	$V_2$	$V_2$	$V_2$
4	$h_4$	$h_4$	$V_5$	$V_5$
3	$h_3$	$h_3$	$V_4$	$V_4$
2	$h_2$	$h_2$	$h_2$	$h_2$
1	$h_1$	$h_1$	$h_1$	$h_1$
0	$h_0$	$h_0$	$h_0$	$h_0$

**13. ANLAGE 3: STEUERCODES CAOS V.2.2**

Code	Name	Funktion (speziell für CRT)
00	DUMMY	Füllzeichen keine Funktion
01	CLEAR	Löschen eines Zeichens: Auf aktuelle Position wird SPACE eingetragen und der Cursor eine Position nach links verschoben
02	ESC	Löschen einer Zeile: Die aktuelle Bildschirmzeile wird mit SPACE gefüllt und der Cursor auf den Anfang dieser Zeile gestellt
03	BREAK	Programmende: Keine Funktion in der CRT-Routine; Abbruch der Zeichenübergabe von einer F-Taste
04	-	n.b.
05	-	n.b.
06	-	n.b.
07	BEEP	Signaltonausgabe: Ausgabe eines kurzen Tones z.B. zur Fehlersignalisierung
08	CUL	CURSOR LEFT: Cursor um eine Position innerhalb des Fensters nach links verschieben bis max. auf HOME-Position
09	CUR	CURSOR RIGHT: Cursor um eine Position innerhalb des Fensters nach rechts verschieben ggf. Rollen des Fensters nach oben
0A	CUD	CURSOR DOWN: Cursor um eine Zeile nach unten, bei Fensterende Rollen des Fensters
0B	CUU	CURSOR UP: Cursor um eine Zeile nach oben bis max. in die Zeile 0 des Fensters
0C	CLS	CLEAR SCREEN: Löschen des Fensters und Eintragen des Codes 00 in den Video-RAM des Fensters
0D	CR	NEW LINE: Funktion wie CUO
0E	-	n.b.
0F	HCOPY	Aufruf Sonderprogramm (z.B.: HARDCOPY): Anfangsadresse des Sonderprogramms auf B799H
10	HOME	CURSOR HOME: Cursor auf Fensteranfang (Zeile 0, Spalte 0); Fensterinhalt unverändert

11	PAGE	Umschaltung PAGE-Modus: Modus bewirkt, daß nach Erreichen des Fensterendes der Cursor bei unverändertem Fensterinhalt auf HOME-Position gestellt wird. (In diesem Modus ist im CAOS keine Kommandoeingabe auf der untersten Zeile möglich!)
12	SCROL	Umschalten SCROLL-Modus: Modus bewirkt, daß nach Erreichen des Fensterendes alle Zeilen des Fensters um eine Zeile nach oben geschoben werden, wobei die oberste Zeile verloren geht. Als unterste Zeile wird eine mit dem Code 20H gefüllte Leerzeile eingefügt und der Cursor auf deren Anfang positioniert. (Dieser Modus entspricht der Grundeinstellung)
13	STOP	Keine Funktion in der CRT-Routine *
14	-	n.b. (Verwendung in CAOS V3.0 und V3.1)
15	-	n.b.
16	-	n.b.
17	-	n.b. (Verwendung in CAOS V3.0 und V3.1)
18	-	n.b.
19	CCR	CURSOR TO BEGIN OF LINE: Cursor auf den Anfang der aktuellen Zeile setzen, ohne diese zu verändern
1A	INS	INSERT: Einfügen eines Leerzeichens (Code 20H) auf aktueller Position und Rechtsverschieben aller rechts davon stehenden Zeichen innerhalb einer Textzeile (nicht unbedingt identisch mit Bildschirmzeile); d.h., es werden so viele Zeichen verschoben, bis der Code 00 erkannt wird, auch über die Bildschirmzeile hinaus. Dabei gehen diese DUMMY-Zeichen verloren, solange mehr als ein DUMMY-Zeichen vorhanden ist. Wenn nur ein DUMMY-Zeichen vorhanden ist, so bleibt dieses als Trennung stehen, und es gehen die rechten Textzeichen verloren.
1B	-	n.b.
1C	LIST	) In der CRT-Routine nicht benutzt;
1D	RUN	) Verwendung in BASIC
1E	CONT	)
1F	DEL	DELETE: Löschen des Zeichens auf der Cursorposition und verdichten des Textes durch Linksverschieben aller Zeichen bis zu einem DUMMY-Zeichen und Einfügen eines weiteren DUMMY-Zeichens (vgl. INS)

**AUFBAU DER UMCODIERUNGSTABELLE FÜR DIE TASTATUR**

(vgl. Bild 4)

Nr.	Taste	Erstbelegung/Code	Zweitbelegung/Code	
1	W	57	w	77
2	A	41	a	61
3	2	32	"	22
4	CUL	06	CCR	19
5	HOME	10	CLS	0C
6	-	2D	=	3D
7	F2	F2	F8	F8
8	Y	59	y	79
9	E	45	e	65
10	S	53	s	73
11	3	33	#	23
12	^	5E	~	5D
13	CLR	01	HCOPY	0F
14	:	3A	*	2A
15	F3	F3	F9	F9
16	X	58	x	78
17	T	54	t	74
18	F	46	f	66
19	5	35	%	25
20	P	50	p	70
21	DEL	1F	ESC	02
22	0	30	@	40
23	F5	F5	FB	FB
24	V	56	v	76
25	U	55	u	75
26	H	48	h	68
27	7	37	'	27
28	O	4F	o	6F
29	INS	1A	INS	1A
30	9	39	)	29
31	BRK	03	BRK	03
32	N	4E	n	6E
33	I	49	i	69
34	J	4A	j	6A
35	8	38	(	28
36	SPACE	20	■	58
37	K	4B	k	68
38	,	2C	<	3C
39	STOP	13	STOP	13
40	M	4D	m	6D
41	Z	5A	z	7A
42	G	47	g	67
43	6	36	&	26
44	Taste nicht vorhanden			
45	L	4C	l	6C
46	.	2E	>	3E

47	F6	F6	FC	FC
48	B	42	b	62
49	R	52	r	72
50	D	44	d	64
51	4	34	\$	24
52	_	5F		5C
53	+	2B	;	3B
54	/	2F	?	3F
55	F4	F4	FA	FA
56	C	43	c	63
57	Q	51	q	71
58	SHIFT LOCK	16	SHIFT LOCK	16
59	1	31	!	21
60	CUD	0A	SCROL	12
61	CUU	08	PAGE	11
62	CUR	09	CUR	09
63	F1	F1	F7	F7
64	CR	0D	CR	0D

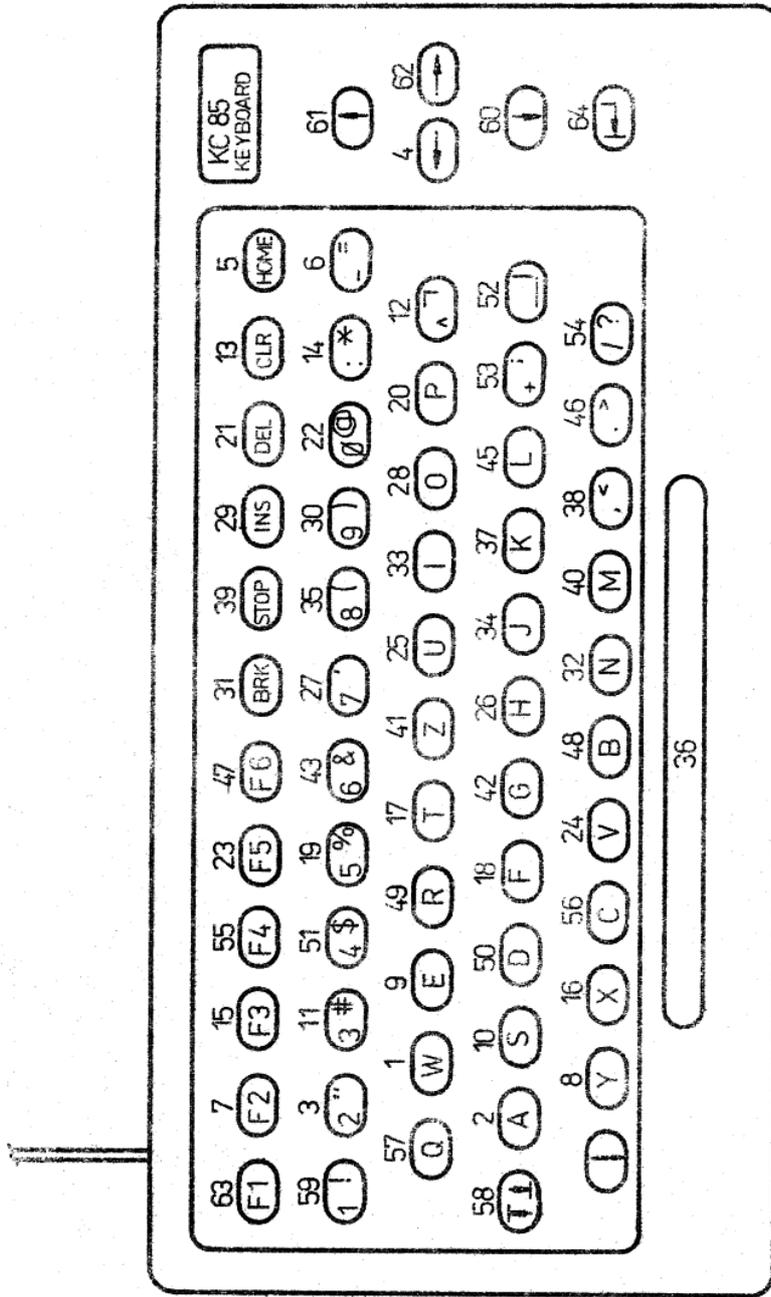


Bild 4  
 Ansicht der Tastatur des KC 85  
 Reihenfolge der Tasten in der Umkodierungstabelle