Beschreibung des Betriebssystems CP/M fuer AC1

Version: CPMAC1/V:1.0

Das auf dem AC1 implementierte CP/M entspricht weitgehend der Version CP/M 2.2 . Es wurde ein spezielles BIOS fuer den AC1 geschaffen.
Voraussetzung fuer die Lauffaehigkeit auf dem AC1 ist ein auf 64 kByte ausgebauter RAM und die Speicherverwaltung nach Bauanleitung Modul1/SCCH.
In Version CPMAC1/1.0 genuegt schon die Abschaltbarkeit des Monitors ueber LD A,4 + OUT (14h),A. Das BIOS benutzt den Monitor fuer die Realisierung der Tastatur-,Bildschirm- und Kassettenroutinen. Fuer die Arbeit mit Kassette wurden die residenten Kommandos im CCP erweitert (LOAD,CSAV,TPA). Da die Diskette Voraussetzung fuer eine sinnvolle Arbeit mit CP/M ist wurde ein Diskettenlaufwerk im RAM simuliert.
Die Groesse dieser Ramdisk kann entsprechend den jeweiligen Bedingungen vor dem Start des CP/M modifiziert werden. Nach dem Laden des CP/M kann auf RAM-Adresse F63Eh der Anfang der Ramdiskette festgelegt werden.
Der Start des CP/M erfolgt mit " J F600 " vom Monitor aus.
Bei Erststart muss die Ramdisk initialisiert werden (Y).
Ansonsten bleibt der Inhalt der Ramdisk bei wiederholtem Start erhalten.
Die Ramdisk liegt zwar im 64k-Rambereich und schraenkt so den verfuegbaren Programmspeicher ein, aber sie erlaubt die Abarbeitung vieler Standard-CP/M-Programme, wie POWER,STAT,TURBO-PASCAL .

Das Betriebssystem CP/M besteht aus des drei Hauptteilen BIOS (Basic Input/Output System), BDOS (Basic Disk Operating System) und CCP (Console Command Processor).Dabei sind BDOS und CCP voellig unabhaengig von der konkreten Hardware des Rechners. Die Verbindung zur Hardware wird ausschliesslich durch das BIOS hergestellt.

Aufteilung des 64k RAM's:

0000h - 00FFh : CP/M Verstaendigungsbereich
0100h - : TPA-freier Programmbereich
B000h - DFFFh : Ramdisk
E000h - E800h : CCP
E800h - F5FFh : BDOS
F600h - FFFFh : BIOS

Achtung der freie Programmbereich endet nicht bei Anfangramdisk sondern schon bei (Anfang-Ramdisk) - 80h - 0E00h + 6h, da es CP/M-Anwendungsprogramme gibt, die das BDOS ueberschreiben.

Tastatur: CNTRL+S - Stop der laufenden Bildschirmausgabe
CNTRL+P - Einschalten Drucker parallel zur Bildschirmausgabe

Der Druckerkanal ist in der Version 1.0 nicht realisiert. Zur Demonstration dieser Funktion werden die Zeichen zur Bildschirmroutine geleitet.

Im vorliegenden BIOS wird die Steuerung ueber das I/O-Byte (Adr .0003h) nicht unterstuetzt. Die Bildschirm-Steuerzeichen sind SCP-kompatibel, zusaetzlich existieren einige Erweiterungen:

Steuerzeichen	Wirkung
00h	NOP (keine Wirkung)
01h	Cursor links oben (home)
07h	akustisches Zeichen an Tastatur (i.a. nicht vorh., dann Blinken der Lampen neben Stop-Taste bzw. der Statuszeile beim PC1715)
08h	Cursor zurueck
0ah	Linefeed (neue Zeile)
0ch	Bildschirm loeschen (verzoegert zum Lesen der zuletzt ausgegebenen Bildschirmzeilen), Cursor links oben
0dh	Carriage Return (an Zeilenanfang)
0eh	Umschalten auf 2. Zeichensatz (nur PC1715)
0fh	Umschalten auf 1. Zeichensatz (nur PC1715)
14h	Rest des Bildschirms loeschen
15h	Cursor nach rechts
16h	Rest der Zeile loeschen
18h	Zeile loeschen, Cursor an Zeiilenanfang
1ah	Cursor eine Zeile hoch
1bh	Einleitung Cursorpositionierfolge, die naechsten beiden Bytes beinhalten Zeile und Spalte Offset 00h oder 80h
7fh	Delete (streichen Zeichen links vom Kursor)
04h	nicht invers
05h	invers
06h	nicht invers
87h	invers

Feste Adressen im unteren Hauptspeicher

00h..02h	JP BIOS+3	(Warmstart)
03h	IOBYTE	
04h	User/Defaultdrive	
05h..07h	JP BDOS	
08h..37h	frei	(fuer RST-Routinen nutzbar)
38h	JP Break	
3bh..3fh	reserviert	

Als Scratch-Bereich des BIOS sind in CP/M die Zellen 40H bis 4FH freigehalten.

Besonderheiten des CCP

Das CCP enthaelt gegenueber der Version CP/M 2.2 einige Erweiterungen (bei gleichem Hauptspeicherbedarf von 800h Bytes) . Es existieren zusaetzliche residente Kommandos:

LOAD "name" : Es wird die unter name angegebene Datei von Kassette geladen (TURBO-TAPE). Die Dateien werden in den RAM geladen und danach nach TPA (100h) verschoben. Da zum Einlesen der MONITOR zugeschaltet wird, muss die Lade-Adresse >= 2000h sein. Achtung, bei zu grosser Dateilaenge kann die RAM-Disk ueberschrieben werden! Das Programm kann nach fehlerfreier Einlesung mit dem Kommando GO gestartet werden. Es kann auch vorher mit dem SAVE-Kommando auf die RAM-Disk abgespeichert werden und von dort wiederholt gestartet werden .

CSAV nn "name" : Das ab TPA(100h) stehende Programm wird nach 2000h verschoben und in der Laenge nn Bytes auf Kassette ausgegeben.

TPA filename : Die unter filename auf der Diskette befindliche Datei wird nach TPA(100h) geladen und kann mit dem Kommando CSAV auf Kassette ausgegeben werden. Es wird die Laenge der Datei ausgegeben.

GO <beliebige Parameter>
Das letzte geladene Programm wird ohne Neuladen aktiviert. Parameter koennen wie beim Direktaufruf angegeben werden.

EXT [d:]<filename>
Das angegeben COM-File wird zu einem residenten Kommando erklaert, indem es vor BDOS, CCP und vor evtl. schon residenten zusaetzlichen Kommandos im Hauptspeicher abgelegt wird, um bei Aufruf statt von Diskette von dort nach 100h geladen zu werden. Hierdurch verringert sich jedoch der TPA entsprechend. Da residente Kommandos nur maximal 4 Zeichen lang sein duerfen, trifft dies auch auf <filename> zu.

RES Streichen aller zusaetzlich residenten Kommandos

HELP Ausgabe einer Liste aller z.Zt. residenten Kommandos

Die anderen CCP-Kommandos entsprechen den bekannten Funktionen CP/M 2.2:

- REN : Umbennen einer Datei
- ERA : Loeschen von Dateien
- TYPE : Anzeigen des Inhaltes einer Datei

Mit Version CPMAC1/V1.1 sind bisher folgende Programme erprobt:

- POWER,STAT,TURBO-PASCAL,DU(ZSID)

Diese Version stellt mit der relativ kleinen Ramdisk nur einen Einstieg ins CP/M dar. Bei Verfuegbarkeit einer groesseren Ramdisk (>32k) koennte das CP/M auch sinnvoll auf dem AC1 genutzt werden (SUPERCALC,WORDSTAR,ASSEMBLER).

Problematisch ist die Einbindung der AC1-Tastatur, da diese nicht interruptfaehig ist. Dies wirkt sich speziell bei einigen Spielprogrammen stoerend aus.

CP/M Informationen

Reservierte Speicherplaetze

Speicherplaetze	Inhalt

0000H - 0002H	Sprung zum BIOS-Eintrittspunkt WBOOT. Damit ist ein einfacher programmierter Neustart (Sprung zu Adresse 0) moeglich.
0003H	Enthaelt das IOBYTE
0004H	Nummer des aktuellen Laufwerkes und Benutzernummer
0005H - 0007H	Enthaelt eine Sprunganweisung zum BDOS. Die Sprunganweisung liefert einmal den Haupteintrittspunkt in das BDOS und zum anderen stellt die Sprungadresse die niedrigste vom Betriebssystem verwendete Speicheradresse dar. Debugger und einige nachladbare Treiber veraendern die Sprungadresse, um den durch sie reduzierten Speicher zu kennzeichnen.
0008H - 002FH	RST 1 bis RST 5, von CP/M-80 nicht verwandt
0030H - 003AH	RST 6 und RST 7, wird von den CP/M-80 Debuggern DDT und ZSID benutzt
003BH - 005BH	fuer CP/M-80 reserviert, Belegung von der konkreten BIOS-Implementierung abhaengig
005CH - 007FH	durch CCP erzeugter Standard-FCB
0080H - 00FFH	Standard-Datenpuffer

BDOS-Funktionen

Die folgenden Symbole werden nachstehend benutzt und bedeuten:

->	ist zu interpretieren als 'Zeiger auf'
<>	ist zu interpretieren als 'ungleich'
*	Verweis auf nachfolgende Bemerkungen

BDOS-Interface

Eingang:	CPU-Register	C :	BDOS-Funktionsnummer
	- " -	DE :	Feldadressen, Vektoren
	- " -	E :	Eingabezeichen
Ausgang:	- " -	A :	Status, Zeichen
	- " -	HL :	Vektoren, Adressen

Nr.	Bezeichnung	Eingang	Ausgang
0	Warmstart	- - -	- - -
1	Konsoleneingabe+Echo	- - -	A: Zeichen (*1)
2	Konsolenausgabe	E: Zeichen (mit ^S, ^P)	- - -

3	Sequentielle Eingabe	- - -	A: Zeichen
4	Sequentielle Ausgabe	E: Zeichen	- - -
5	List - Ausgabe	E: Zeichen	- - -
6	direkte Konsolen E/A	E = FF: Eingabe E <> FF:Ausgabe	A: Status (*2) - - -
7	IOBYTE abfragen	- - -	A: IOBYTE
8	IOBYTE belegen	E: IOBYTE	- - -
9	Zeichenkette ausgeben	DE -> Kette EKZ=$, mit ^S,^P	- - -
10	Eingabe Konsolpuffer	DE -> Puffer (*3)	- - -
11	Konsolenstatus	- - -	A: Status (*2)
12	Version ermitteln	- - -	HL: Version
13	Diskettensystem zuruecksetzen	- - -	A: Batchmode - Flag (*9)
14	Auswahl Bezugs-LW	E: LW# (0...F)	- - -
15	Datei eroeffnen	DE -> FCB	A: DC (*7) =FF: keine Datei
16	Datei schliessen	DE -> FCB	A: DC (*7) =FF: keine Datei
17	erste Eintragung suchen	DE -> FCB	A: DC (*7) =FF: keine Datei
18	folgende Eintragung suchen	- - -	A: DC (*7) =FF: keine Datei
19	Dateien loeschen	DE -> FCB	A: DC (*7) =FF: keine Datei
20	naechsten Satz lesen	DE -> FCB	A: EC (*5) <>00: EOF
21	naechsten Satz schreiben	DE -> FCB	A: EC (*5) <>00: Disk. voll
22	Datei erzeugen	DE -> FCB	A: DC (*7) =FF: Verz. voll
23	Datei umbenennen	DE -> FCB (*8)	A: DC (*7) =FF: keine Datei
24	Abfrage ange-schlossener LW	- - -	HL: LW - Vektor
25	Abfrage Bezugs-LW	- - -	A: LW# (0 - 15)
26	Datenpuffer adressieren	DE -> Puffer	- - -
27	Belegungstabelle (ALLOC) ermitteln	- - -	HL -> ALLOC
28	Schutz des Bezugs-LW	- - -	- - -
29	Abfrage R/O - LW	- - -	HL: LW - Vektor
30	Dateimerkmale setzen	DE -> FCB	A: DC (*7) =FF:keine Datei
31	Diskettenparameterta-	- - -	HL -> DPB des

```
    belle (DPB) ermitteln                          selekt. LW
32  Nutzernummer            E= FF: Abfrage     A:Nutzer# (0-15)
    abfragen/setzen         E<>FF: Setzen
33  direkt adressierten     DE -> FCB          A: RC      (*6)
    Satz lesen
34  direkt adressierten     DE -> FCB          A: RC      (*6)
    Satz schreiben
35  Dateigroesse berechnen  DE -> FCB            - - -
36  Berechnung der          DE -> FCB          Eintrag in
    aktuellen Satzadresse                      FCB + 33,34,35:
r0-2=fkt(ex,cr)



37  ausgewaehlte Laufwerke  DE: LW-Vektor      A: 00
    zuruecksetzen
40  direkt adressierten     DE -> FCB          A: RC      (*6)
    Satz schreiben und
    Block initialisieren

Fussnoten:
*1: Sonderzeichen
      ^H: Rueckschritt
      CR: Wagenruecklauf
      LF: Zeilenschaltung
      ^I: Tabulation auf naechste 8. Stelle
      ^S: start/stop Konsolenausgabe
      ^P: List - Echo

*2: Status =  00 kein Zeichen im Konsoleneingabepuffer
           <> 00 Zeichen im Konsoleneingabepuffer

*3: Steuerzeichen
      7F: Loeschen letztes Zeichen
      ^C: Sprung zum Warmstart, wenn als 1. Zeichen angeboten
      ^E: Ende der physischen Zeile
      ^H: Rueckschritt um 1 Zeichen
      ^J: (LF) Abschluss der Zeile
      ^M: (Abschlusstaste) Abschluss der Zeile
      ^R: Wiederausgabe der redigierten Konsolzeile
      ^U: ganze Zeile zurueckweisen
      ^X: Ruecksetzen an Zeilenanfang mit Loeschen

*4: Struktur des Puffers
      DE: +0 | +1 | +2 | +3 | ... | +n+1
          mx | nx | Z1 | Z2 | ... |  Zn

            mx - Kapazitaet des Puffers
            nx - Fuellstand des Puffers
             Z - Zeichen im Puffer
```

*5: Fehler-Code
A = 00 : Operation ok
<> 00 : Operation nicht ausfuehrbar

*6: Direktzugriffs-Code
A = 01 : Lesen ungeschriebener Daten
= 03 : Extent-Wechsel nicht moeglich
= 06 : Zugriff auf EOD-Satz der Datei

*7: Verzeichnis-Code
A = FF : Fehler
= 0,1,2,3 : Nummer des Eintrags im aktuellen Verzeichnissatz

*8: Aufbau des Dateibeschreibers fuer Umbenennung
FCB +0 - +15 : Datei alt (LW = 0,1, ... 16)
+16 - +31 : Datei neu (LW = 0 oder LW-alt)

*9: Batchmode
A = 00 : keine Batchmodedatei
A = FF : Batchmodedatei vorhanden
Aufbau des FCB:

+00	dr	Laufwerkcodierung dr = 0 : Bezugs-LW 1 : LW A 16 : LW P
+01	f1 - f8	Dateiname in ASCII Flags in den 8. Bits f1' - f8' fuer Nutzerflags reserviert
+09	t1 - t3	Dateityp in ASCII Flags in den 8. Bits t1' - t3' t1' = 1/0 : geschuetzte/ungeschuetzte Datei t2' = 1/0 : SYSTEM-/Verzeichnis-Datei t3' : reserviert
+12	ex	laufende Nummer des Dateiteils (Extent) 0..31
+13	s1 - 2	reserviert fuer internen Gebrauch
+15	rc	Satz-zaehler innerhalb laufenden Extens
+16	d0 - 15	Blockverzeichnis des Extents
+32	cr	laufende Satznummer der sequentiellen Datei innerhalb des Extents

```
    +33  |   r0 - 2  |  Satznummer der Datei fuer Direktzugriff
    +36  |           |  Ende FCB
```

Aufbau der LW-Vektoren

```
Vektor im CPU-Doppelregister
          bit 0 : LW A
              1 : LW B
                :::: :
             15 : LW P
```

BIOS-Aufrufe

BIOS-Interface

```
  Eingang: CPU-Register C  :  8-Bit-Werte
               - " -     BC : 16-Bit-Werte
               - " -     DE :  ev. 2. 16-Bit-Wert
  Ausgang:     - " -     A  :  8-Bit-Werte
               - " -     HL : 16-Bit-Werte
```

Nr.	Bezeichnung	Eingang	Ausgang
1	BOOT (Kaltstart)	- - -	- - -
2	WBOOT (Warmstart)	* * *	* * *
3	CONST (Status-Abfrage CON:)	- - -	A=00: kein Zeichen =FF: sonst
4	CONIN (Konsol-Eingabe)	- - -	A: Zeichen
5	CONOUT(Konsol-Ausgabe)	C:Zeichen	- - -
6	LIST (Ausgabe =Drucker! nach LST:)	C: Zeichen	* * *
7	PUNCH (Ausgabe nach PUN:)	C: Zeichen	- - -
8	READER (Eingabe von RDR:)	- - -	A: Zeichen
9	HOME (Positionieren Spur 0)	- - -	- - -
10	SELDSK (LW auswaehlen)	C:LW-Nummer	HL -->Diskparameterkopf DPH
11	SETTRK (Spur auswaehlen)	BC: Spurnummer	- - -

12	SETSEC (Sektor auswaehlen)	BC: Sektornummer	- - -	
13	SETDMA (Puffer-Adr.einstellen)	BC: DMA-Adresse	- - -	
14	READ (Sektor lesen)	- - -	A=00: Lesen OK =01: Lesefehler	
15	WRITE (Sektor schreiben)	- - -	- " - - " -	
16	LISTST (Status Kanal LST:	- - -	A=FF: Bereit-schaft =00: sonst	
17	SECTRAN(Wandeln Sektornummer	BC: Sektornummer DE: Tab-Adresse	HL: Sektornummer	

(Vom AC1 ausgelesen und entsprechend Original-Bildschim
formatiert von Norbert Z80-Nostalgiker 05/2009)