

Moppel Software

CP/M mit CF-Karte

(Stand 16.05.2016)

Inhalt:

Seite 2	Vorwort, Speicherausstattung/Banking CF-Karte als Bootlaufwerk
Seite 3	Auszug aus dem Quellcode
Seite 4	DiskparameterHeader
Seite 5	Diskparameterblock
Seite 6	BIOS-Warmstart
Seite 7	BOOT-Vorgang
Seite 9	CF-Hilfsprogramme
Seite 10	Datenaustausch mit 22DISK
Seite 11	IO-Byte der „Gerätmanager“
Seite 13	LST: über schnelle V24
Seite 14	Quellenangaben

Anlagen:

FDC-Routinen	als ASM-Code
BIOS	Version3.0
BOOT-Loader	

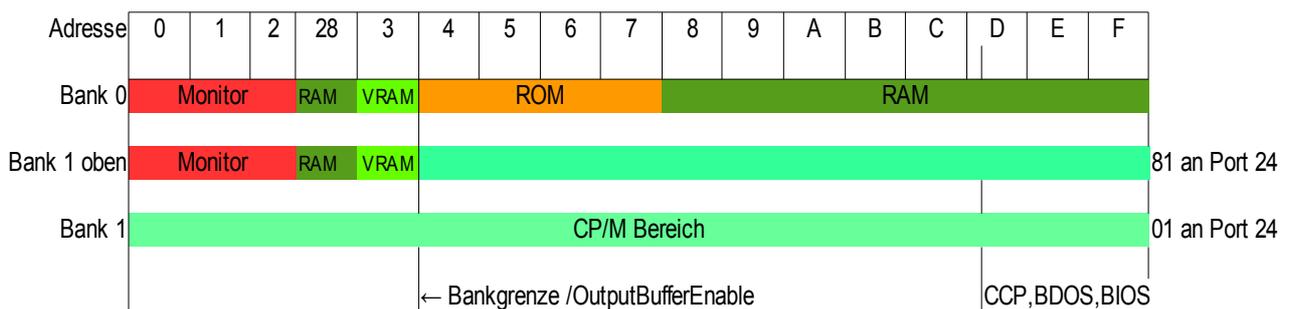
Vorwort:

Der Moppel wurde in erster Linie für die Verwendung der Monitorroutinen, sowie Assembler und Basic entwickelt. Damit ließen sich Anno 83-85 schon weitreichende Steuerungsaufgaben verwirklichen. Mit dem Ausbau zum vollwertigen CP/M-Computer stand dann schließlich die weite Welt der Profi-Software (MBASIC, dBASE, Wordstar etc) zur Verfügung. Für ein „Bastel-PC“ schon eine große Spannbreite.

In dieser Abhandlung beschreibe ich die Benutzung der CF-Karte unter CP/M anhand einiger Auszüge aus dem BIOS-Quellcode.

Speicherausstattung/Banking:

Grundvoraussetzung für den CP/M-Betrieb ist natürlich ein durchgängiger RAM-Bereich von 64kByte, dies konnte damals durch zwei zusätzliche Speicherkarten oder wie bei mir, durch ein entsprechendes Häkelwerk erreicht werden. Dabei ist der Bankumschaltung Beachtung zu schenken. Nicht besonders kompliziert, ein Latch erzeugt die Adressen 16 bis 18 und das Selektsignal /OutputBufferEnable. Dieses Latch wird über die IO-Adresse 24 angesprochen, wobei die Datenleitung 0 bis 3 die Adressen 16 bis 18 widerspiegeln und das Bit 7 die Enable-Leitung.



Mit gesetztem Bit 7 bleiben die ROM-Adressen von 0000H bis 3FFFh aktiv, somit laufen noch alle Monitorprogramme und natürlich die Floppy-Routinen. Die unteren Bits schalten die entsprechende RAM-Bank ab Adresse 4000H bis FFFFH ein. So kann der Bootloader das CP/M, also CCP, BDOS und BIOS ab der Adresse D400H ins RAM schreiben und anschließend den Kaltstart vom BIOS anspringen. Dort ist die erste Aufgabe, den unteren EPROM-Bereich auf die RAM-Bank umzuschalten damit CP/M seinen vollen Lebensraum erhält.

CF-Karte als Bootlaufwerk:

Die Batchroutinen aus dem Monitor funktionieren nur mit dem 1.Laufwerk (A) und CP/M lässt sich auch nicht dazu herab von einem anderen Laufwerk zu starten. Also wird die CF-Karte nun als Laufwerk deklariert.

Hierzu sind einmal Anpassungen in den FDC-Routinen, im BIOS und für den Bootloader erforderlich, sowie die Laufwerkszuordnung von 3 (D) auf 0 (A) geändert und ein eigenständiger Bootloader eingerichtet werden. Im BIOS sind die Tabellen DPH und DPB entsprechend zuzuordnen.

Auszug aus dem Listing FDC-Routinen

```
1089 25A2      ; cfq = Test ob Laufwerk = 0 (geändert 05.05.2016)
1090 25A2      ; wenn nein weiter mit Floppy
1091 25A2      ;   und Test auf rd/wr Sektor aus RWSUB
1092 25A2      ;
1093 25A2 F5    cfq:   push   psw
1094 25A3 3A 73 2F    lda   selbyt      ;
1095 25A6 FE 18    cpi   18h      ; Test ob Laufwerk 0
1096 25A8 C2 B8 25    jnz   cfq_e       ; nein weiter mit Floppy
1097 25AB 3A 7F 2F    lda   rwsubl     ; RWSUB1 testen
1098 25AE FE 40     cpi   40h        ;
1099 25B0 CA C1 25    jz    cfwrbl     ; ja CF-Karte schreiben
1100 25B3 FE 66     cpi   66h        ;
1101 25B5 CA BA 25    jz    cfrdbl     ; ja CF-Karte lesen
1102 25B8 F1      cfq_e  pop    psw
1103 25B9 C9      ret
      ;
```

Hier steht die Laufwerksinformation im SelektByte (18H) für das Laufwerk 0 bzw. A

Auszug aus dem BIOS:

```
0754 ED38      ;
0755 ED38      ; Test ob Laufwerk A ausgewählt wurde
0756 ED38      ;           dann zu den CF-Routinen
0757 ED38      ;
0758 ED38 F5    gettrk:  push   psw      ;
0759 ED39 3A BB F1    lda   aktdrv ;
0760 ED3C FE 00    cpi   00h      ; Laufwerk = A?
0761 ED3E C2 69 ED    jnz   get_fl    ; nein weiter mit Floppy
0762 ED41 F1      pop    psw      ;
0763 ED42 C3 70 F0    jmp   get_cf    ; ==> zu CF-Routinen
0764 ED45
0765 ED45      ;
0766 ED45 F5    puttrk:  push   psw      ;
0767 ED46 3A BB F1    lda   aktdrv ;
0768 ED49 FE 00    cpi   00h      ; Laufwerk = A?
0769 ED4B C2 8F ED    jnz   put_fl    ; nein weiter mit Floppy
0770 ED4E F1      pop    psw      ;
0771 ED4F C3 92 F0    jmp   put_cf    ; ==> zu CF-Routinen
      ;
```

Im BIOS steht die Laufwerksauswahl in der Variablen aktdrv, 00H für LW A, 02H für LW B usw.

Änderungen am DPH :

```
0085 EA33      ;=====
0086 EA33      ;
0087 EA33      ;  DPBase
0088 EA33      ;
0089 EA33      ;=====
0090 EA33      ;
0091 EA33      dpbase equ dpe0
0092 EA33      ; DPH fuer Disk 0 (A)
0093 EA33 00 00  dpe0: dw  xlt0      ; Translate table
0094 EA35 00 00      dw  0000h
0095 EA37 00 00      dw  0000h  ; Sratch area
0096 EA39 00 00      dw  0000h
0097 EA3B D5 F1      dw  dirbuf   ; Dir. Buffer
0098 EA3D A0 EA      dw  dpb3    ; Parameter Block
0099 EA3F BB F3      dw  csv3    ; Check
0100 EA41 BB F2      dw  alv3    ; Allocation Vectors
0101 EA43      ;
          ; DPH fuer Disk 1 (B)
0102 EA43 00 00  dpe1: dw  xlt1
0103 EA45 00 00      dw  0000h
0104 EA47 00 00      dw  0000h
0105 EA49 00 00      dw  0000h
0106 EA4B D5 F1      dw  dirbuf
0107 EA4D 82 EA      dw  dpb1
0108 EA4F 89 F2      dw  csv1
0109 EA51 77 F2      dw  alv1
          ;
```

Für jedes LW muss im BIOS ein DiskParameterHeader vorhanden sein, dort stehen dann die Verweise auf der eigentlichen Parametertabell (DiskParameterBlock).

Änderungen am DPB :

```
182 EAA0      ; CF-Karte
183 EAA0      ;
184 EAA0      ; 8MB File auf CF-Karte
185 EAA0      ; Offset für Sektoren sind im AVR-Teil einzurichten
186 EAA0      ;
187 EAA0      ; 32 Records a 128Byte
188 EAA0      ; 4kBlöcke
189 EAA0      ; 512 Dir Einträge
190 EAA0      ; ohne Check (CF-Karte als "Festplatte")
191 EAA0      ; 4 Systemspuren
192 EAA0      ;
193 EAA0      ; Wichtig alv3 anpassen
194 EAA0      ; Parameter stehen im hinteren Bereich
195 EAA0      ; ab Soft-System
196 EAA0      ; alv3 = DSM+1 / 8
;=====
; DPB fuer die CF-Karte LW 0 (A)
2018 EAA0 20 00 dpb3: dw 32          ; Sectors per Track
2019 EAA0 05          db 5          ; Block shift
2020 EAA0 1F          db 31         ; Block Mask
2021 EAA0 01          db 1          ; Extnt Mask
2022 EAA0 FF 07       dw 2047       ; DSM Disk Size-1 = 8MB
2023 EAA0 FF 01       dw 511        ; Directory max.
2024 EAA0 F0          db 11110000b ; Alloc0
2025 EAA0 00          db 0          ; Alloc1
2026 EAA0 00 00       dw 0          ; Check Size
2027 EAA0 04 00       dw 4          ; Offset
2028 EAA0             ;
2029 EAA0 xlt3: equ xlt0 ; keine Sektorübersetzung
20210 EAA0             ;
; DPB fuer Disk 1 (B)
20154 EA82 20 00     dpb1: dw 20h          ; Sectors per Track
20155 EA84 04          db 04h          ; Block shift
20156 EA85 0F          db 0Fh          ; Block Mask
20157 EA86 00          db 0h          ; Extnt Mask
20158 EA87 97 00       dw 97h          ; Disk Size-1
20159 EA89 3F 00       dw 3fh          ; Directory max.
20160 EA8B C0          db 0c0h         ; Alloc0
20161 EA8C 00          db 00h          ; Alloc1
20162 EA8D 10 00       dw 10h          ; Check Size
20163 EA8F 04 00       dw 4h          ; Offset
20164 EA91             ;
20165 EA91 xlt1 equ xlt0
;=====
```

Im Diskparameterblock wird das Diskettenformat festgelegt. CP/M arbeitet mit Records von 128Byte, der Moppel ist auf 256Bytes pro Sektor und 16 Sektoren pro Spur bei 40 bzw. 80 Spuren als einseitiges Format festgelegt. So müssen die Parameter für die jeweilige Disk-Kapazität umgerechnet werden.

Der Moppel hat da noch eine Besonderheit, die maximale Anzahl von Sektoren ist auf 16 begrenzt, was eine Recordzahl von 32 ergibt. Um dies zu erweitern, müssten die internen Umrechnungen neu programmiert werden, ggf. in einer späteren Version ...

Da aber die Spurzahl als Word deklariert ist, sind rund 65000 Spuren möglich und zusammen mit den anderen Parametern kommt man auf die max. Kapazität von 8MB der CF-Karte – mehr als genug.

Für die Details und Berechnungen der Parameter gibt es auf der Internetseite eine sehr ausführliche Erläuterung:

<http://hc-ddr.hucki.net/wiki/doku.php/cpm:systemdoku>

BIOS-Warmstart:

Hier muss sichergestellt werden, dass nach einem Laufwerkswechsel oder Programmbeendigung das System wieder in einem geordneten Zustand zurückfindet. Im CP/M dürfen Anwenderprogramme den CCP überschreiben um den Arbeitsbereich zu erweitern. Dazu wird der CCP und das BDOS von den Systemspuren komplett neu geladen.

```

1254 F00A 31 80 00 wboot: lxi      sp,buffer
1255 F00D          ;
1256 F00D 3A BA F1      lda      wrtflg  ; wurde alter Buffer auf Disk
1257 F010 B7          ora      a          ; geschrieben ?
1258 F011 CA 17 F0      jz      load      ; write alten Buffer auf Disk
1259 F014 CD 45 ED      call     puttrk   ; weiter mit Floppy
1260 F017          ;
1261 F017 AF      load:  xra      a
1262 F018 2F          cma
1263 F019 32 BA F1      sta      wrtflg   ; Buffer ungueltig
1264 F01C          ;
1265 F01C          ; CCP und BDOS neu laden
1266 F01C          ;
1267 F01C 21 00 D4      lxi      h,ccp    ; Startadresse 60k CP/M
1268 F01F 22 B6 F1      shld     datadr   ; - sichern
1269 F022 3E 00      mvi      a,0h     ; Startspur = 0
1270 F024 32 A9 F1      sta      trknr   ; - sichern
1271 F027 3E 02      mvi      a,2h     ; Startsektor = 2
1272 F029 32 AA F1      sta      secnr   ; - sichern
1273 F02C          ;

```

```

;
1274 F02C CD 1A F1 bootl: call settr ; Spur setzen
1275 F02F CD 26 F1 call setsek ; Sektor setzen
1276 F032 CD 4E F1 call rdsek ; Sektor schreiben
1277 F035 3A AA F1 lda secnr ;
1278 F038 3C inr a ; Sektornummer hochzaehlen
1279 F039 3C inr a ; wg. Sektorberechnung im AVR
1280 F03A 32 AA F1 sta secnr ;
1281 F03D FE 17 cpi 23 ; 11 Sektoren laden
1282 F03F DA 2C F0 jc bootl ; nein - weiter
1283 F042 ;
1284 F042 C3 E9 EF jmp gocpm ; weiter zum ccp
;

```

Hier gibt es die Besonderheit, dass die Sektoren nicht einfach hochgezählt werden können, denn im AVR gibt es die Umrechnung $\text{Sektoren} / 2 + \text{mod}2$. Damit wird die Sektorlänge von 512Byte (CF-Karte) auf 256Byte (Moppel) angepasst. Deshalb hier die scheinbar doppelte Sektorzahl. Ein Fusch kommt selten allein, also muss das hier wieder gerade gebogen werden ;-))

BOOT von CF-Karte:

Schlussendlich wird nun das System aus den Systemspuren ins RAM kopiert. Hierzu habe ich einen sogenannten Bootloader geschrieben. Da der gewohnte Batchbetrieb noch nicht für die CF-Karte angepasst ist, habe ich das Teil im EPROM ab 5000H untergebracht (da stand mal der BASIC-Interpreter).

Wegen der Speicherbankgrenze bei 4000H würde er dort nicht funktionieren, also „Dreisprung“. Eine Kopieroutine schauffelt erstmal den Boot-Loader und ein paar nützliche Hilfsprogramme ins RAM ab 2800H um von dort aus mit der Bankumschaltung die Systemspuren in die Bank1 oberer Teil zu kopieren.

Im Videomonitor ist dann noch das Sprungziel B(Basic) entsprechend umgebogen und das CP/M wird einfach mit B am Monitorprommt gestartet – das geht rats-fat.

Hier der Auszug aus dem Boot-Loader

```

0209 2F86 org 2C00h ; Startadresse
0210 2C00
0211 2C00 C3 15 2C start: jmp boot ; CP/M von CF-Karte starten
0212 2C03 C3 5C 2C format: jmp form ; Systemspuren und Inhaltsverzeichnis mit E5 beschreiben
0213 2C06 C3 B4 2C sysrd: jmp sread ; Systemspuren 0-3 von Diskette in Buffer laden
0214 2C09 C3 CE 2C syswr: jmp swrite ; Systemspuren auf CF-Karte schreiben
0215 2C0C 21 B9 2D ende: lxi h,msge ; Endemeldung
0216 2C0F CD 21 10 call string ;
0217 2C12 C3 03 10 jmp MOVID ; zurueck zum Monitor
0218 2C15 ;
0219 2C15 ;

```

```

0220 2C15      ;-----
0221 2C15      ;
0222 2C15 CD 42 26 boot: call    cfinit      ; init
0223 2C18      ;
0224 2C18 21 86 2D      lxi    h,msg3      ;
0225 2C1B CD 21 10      call   string      ; Meldung
0226 2C1E 21 00 D4      lxi    h,ccp60k   ; Startadresse 60k CP/M
0227 2C21 22 7A 2F      shld   datadr     ; - sichern
0228 2C24 3E 00      mvi    a,0h      ; Startspur = 0
0229 2C26 32 70 2F      sta    trknr     ; - sichern
0230 2C29 3E 02      mvi    a,2h      ; Startsektor = 2
0231 2C2B 32 71 2F      sta    secnr     ; - sichern
0232 2C2E      ;
0233 2C2E 3E 81      mvi    a,81h     ; Bank 1 ab 4000h
0234 2C30 D3 24      out    bank      ; einschalten
0235 2C32      ;
0236 2C32 CD 76 26 boot: call    settr      ; Spur setzen
0237 2C35 CD 82 26      call   setsek    ; Sektor setzen
0238 2C38 CD AA 26      call   rdsek    ; Sektor schreiben
0239 2C3B 3A 71 2F      lda    secnr     ;
0240 2C3E 3C      inr    a         ; Sektornummer hochzaehlen
0241 2C3F 3C      inr    a         ; wg. Sektorberechnung im AVR
0242 2C40 32 71 2F      sta    secnr     ;
0243 2C43 FE 10      cpi    10h      ; Sektor 16 erreicht
0244 2C45 DA 32 2C      jc     bootl    ; nein - weiter
0245 2C48 3E 01      mvi    a,1h     ; Sektor 1
0246 2C4A 32 71 2F      sta    secnr     ; - sichern
0247 2C4D 3A 70 2F      lda    trknr    ; Spurnummer holen
0248 2C50 3C      inr    a         ;
0249 2C51 32 70 2F      sta    trknr    ; - sichern
0250 2C54 FE 02      cpi    02h     ; Spur 2 erreicht
0251 2C56 DA 32 2C      jc     bootl    ; nein - weiter
0252 2C59      ;
0253 2C59 C3 00 EA      jmp    bios60k   ; jmp bios60k

```

Kopierprogramm für den Bootloader:

```
0396 29C3      ; Ladeprogramm fuer CF-Tools Lader
0397 29C3      ; Lader aus EPROM ab 5200h ins RAM 29C0h
0398 29C3      ;
0399 29C3      ;
0400 29C3      org      5000h      ;
0401 5000
0402 5000 C3 06 50 laders: jmp      lader-      ; "Bootlader"
0403 5003 C3 00 58      jmp      5800h      ; Hexlader 2. Version
0404 5006      ;
0405 5006 21 00 52 lader-: lxi      h,5200h      ; Quelle Startadr.
0406 5009 22 BF 2F      shld     srcbeg      ;
0407 500C 21 FF 53      lxi      h,53ffh      ; Quelle Endadr.
0408 500F 22 C1 2F      shld     srcend      ;
0409 5012 21 C0 29      lxi      h,29C0h      ; Ziel Startadr.
0410 5015 22 C3 2F      shld     dstbeg      ;
0411 5018 CD 88 00      call     copym      ;
0412 501B
0413 501B C3 C0 29      jmp      29C0h      ;
```

Durch die Nutzung der Monitorroutine copym, ist die Laderoutine sehr einfach gehalten.

In der Sprungleiste sind noch ein paar Hilfsprogramme eingerichtet:

format Damit werden die vier Systemspuren mit 00H überschrieben
 dies könnte ich bei Bedarf noch erweitern um das Inhaltsverzeichnis
 ab Spur 4 zu löschen (überschreiben mit E5)

sysrd schreibt die Systemspuren von der Diskette LW B ins RAM ab
 Adresse D400h.

syswr Speichert die Daten ab D400H als Systemspuren auf die CF-Karte.

Mit Hilfe des Intel-Hexladers sind BIOS-Änderung sehr einfach und schnell erledigt:

Sysread aufrufen, dann mit den Hexlader neues BIOS ab EA00H ins RAM laden, der wertet die ORG-Adresszuweisung aus.

Dann alles zusammen mit **syswr** auf die CF-Karte speichern und mit boot CP/M aufrufen.

Fertig !

Programmaustausch CP/M ↔ DOS:

In der jetzigen Ausbaustufe ist der direkte Programmaustausch zwischen den beiden Welten noch nicht möglich. Mit dem Tool 22DISK können Disketten unter DOS auf unterschiedliche CP/M Formate schreiben/lesen, davon gibt es mindestens so viele wie damals Computerhersteller.

Da das Moppelformat nicht enthalten ist, muss hierfür eine Definitionstabelle neu erstellt werden:

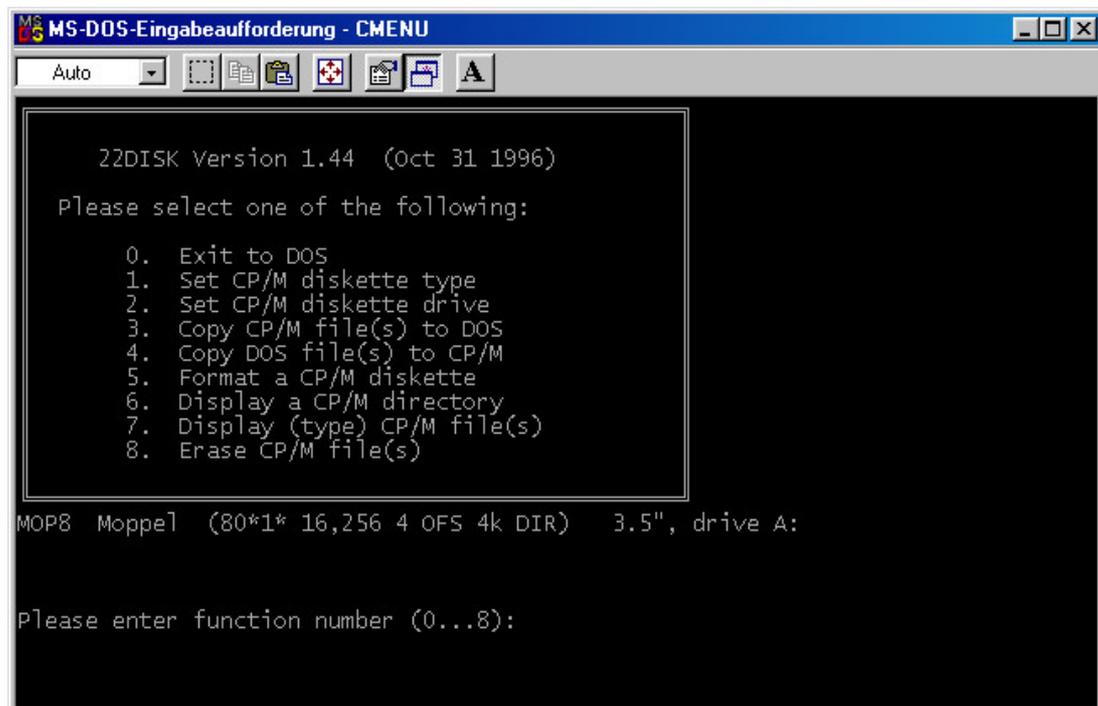
```
NOTE THIS IS THE DATABASE FOR VERSION 1.44 OF 22DISK BY SYDEX.
NOTE
NOTE Edited JULY 17, 1989 for European Purposes
NOTE
NOTE SYDEX
NOTE 153 NORTH MURPHY AVE.
NOTE SUNNYVALE, CA 94086
NOTE (408) 739-4866
```

```
BEGIN MOP8 Moppel (80*1* 16,256 4 OFS 4k DIR) 3.5"
DENSITY MFM,LOW CYLINDERS 80 SIDES 1 SECTORS 16,256 SKEW 1
SIDE1 0 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
ORDER SIDES
LABEL MOPPEL
BSH 4 BLM 15 EXM 0 DSM 151 DRM 63 AL0 192 AL1 0 OFS 4
END
```

Diese wird mit dem Hilfsprogramm genindex.com für 22DISK bereitgestellt

[C:/22DISK](#)> genindex moppel.txt cpmdisks.def

Damit können nun einseite Disketten mit 80Spuren gelesen und beschrieben werden.



IO-Byte der „Gerätemanager“:

Die Verwaltung der diversen Geräte, Console, Drucker, V24 etc. wird über das sogenannte IO-Byte in der Adresse 0004H gesteuert. Dort gibt es vier Ein-Ausgabekanäle denen jeweils vier Geräte zugewiesen werden können, macht in Summe 16 Geräte.

IO-Byte in Adresse 0004H

Kanal Bit	LST:		PUN:		RDR:		CON:		Gerät
	7	6	5	4	3	2	1	0	
							0	0	TTY:
							0	1	CRT:
							1	0	BAT:
							1	1	UC1:
					0	0			TTY:
					0	1			PTR:
					1	0			UR1:
					1	1			UR2:
			0	0					TTY:
			0	1					PTP:
			1	0					UP1:
			1	1					UP2:
	0	0							TTY:
	0	1							CRT:
	1	0							LPT:
	1	1							UL1:

Mit dem STAT-Befehl kann die Zuordnung gelesen und geändert werden:

STAT DEV: zeigt die Zuordnung Gerät zu Kanal

STAT LST:=LPT: wird dem Listkanal das Gerät LPT: zugeordnet also Drucker über die parallele Schnittstelle

Darüber hinaus gibt es noch für den Kanal CON: und LST: eine Statusabfrage die natürlich auch zum dem gewähltem Gerät eingestellt werden muss. Wenn alle 16 Geräte im BIOS angesprochen werden sollen ist ein entsprechender Treiber erforderlich, die nicht benötigten werden mit einem Dummy belegt.

Die Geräte zu den Kanälen haben im CCP eine feste Bedeutung (siehe oben), was und wie dort etwas gemacht wird, bestimmt alleine das BIOS.

So kann z.B. die Konsole direkt vom Monitorprogramm, wie hier im Moppel oder wie in den meisten anderen CP/M Systemen über eine V24 von einem externen Terminal, bedient werden. Das Gleiche gilt auch für die anderen Kanäle ...

Gerätemanager:

Hört sich großartig an, ist aber nur ein etwas trickreich gestalteter Umschalter.

Am Beispiel der Konsole beleuchte ich hier einzelnen Schritte.

Das BDOS spricht die Kanäle über die am Anfang vom BIOS stehende Sprungtabelle an.

```
0000          org bios
0063 EA00          ;
0064 EA00 C3 B9 EF      jmp boot    ; Einsprungleiste
0065 EA03 C3 0A F0      wboote: jmp wboot
0066 EA06 C3 16 EB      jmp csts   ; Status Console
0067 EA09 C3 C2 EA      jmp ci     ; Input Console
0068 EA0C C3 AF EA      jmp co     ; Output Console
                        ;
```

Hier geht's weiter mit den Vorbereitungen für den Umschalter

```
0218 EAAF 3A 03 00      co:  lda iobyte ; CP/M : Console Output
0219 EAB2 E6 03          ani 03h
0220 EAB4 21 BA EA      lxi h,cotab ; Startadr CO-Tab
0221 EAB7 C3 6A EB      jmp decode
0222 EABA C6 EB      cotab: dw co-mon ; TTY
0223 EABC 36 EC          dw crtout ; CRT
0224 EABE 01 EB          dw list  ; BAT
0225 EAC0 CB F3          dw uc1out ; UC1; Ext.
```

Dazu wird das IO-Byte geholt, die für die Konsole zuständigen Bits maskiert und mit dem CO-Tabellenanfang in Register [HL] zur Auswertung gehüpft.

```
0315 EB6A CA 73 EB      decode:      jz found    ; Setzt PC entsprechend IOBYTE
0316 EB6D 23            inx h
0317 EB6E 23            inx h
0318 EB6F 3D            dcr a
0319 EB70 C3 6A EB      jmp decode
0320 EB73 5E      found:  mov e,m    ; Zieladresse auslesen
0321 EB74 23            inx h
0322 EB75 56            mov d,m
0323 EB76 EB            xchg
0324 EB77 E9            pchl      ; Verzweigung
```

Hier dient das maskierte IO-Byte als Zähler für die Tabelle. Sobald er auf 0 steht wird noch die Zieladresse aus der Tabelle geholt und dort im eigentlichem Treiber die Konsolenausgabe verarbeitet. Dieses Spiel ist für alle Kanäle gleich, es müssen nur die zuständigen Bits in die Bit-Position null und eins geschoben und maskiert werden damit der Zähler funktioniert.

LST: über schnelle V24:

Als erstes habe ich den LST: Kanal erweitert um Druckausgaben über die schnelle V24 der CF-Karte zu schicken.

```
0261 EB01 3A 03 00 list: lda iobyte ; CP/M : List Output [C]= Zeichen
0262 EB04 07 rlc
0263 EB05 07 rlc
0264 EB06 E6 03 ani 03h
0265 EB08 21 0E EB lxi h,lotab ; Startadr List-Tab
0266 EB0B C3 6A EB jmp decode
0267 EB0E C6 EB lotab: dw co-mon ; TTY
0268 EB10 36 EC dw crtout ; CRT
0269 EB12 78 EB dw prt ; LPT
0270 EB14 6A F1 dw v24txb ; UL1 ← Sprung zum Gerätetreiber
0271 EB16 ;
```

In der Tabelle ist das Sprungziel einzutragen.

```
1554 F16A ;
1555 F16A ;=====
1556 F16A ; Geraetetreiber fuer schnelle V24 auf CF-Karte
1557 F16A ;
1558 F16A ; Befehle fuer AVR
1559 F16A ;
1560 F16A v24tx equ 52h ; ein byte senden ohne Buffe
1561 F16A v24rx equ 53h ; ein byte empfangen
1562 F16A v24str equ 54h ; String senden Ende 00h mit Buffer
1563 F16A v24zeil equ 55h ; Zeile empfangen bis CR
1564 F16A ;
1565 F16A 3E 52 v24txb: mvi a,v24tx ; Befehl
1566 F16C CD E8 F0 call piowr ;
1567 F16F 79 mov a,c ; Zeichen
1568 F170 CD E8 F0 call piowr ; ausgeben
1569 F173 C9 ret ;
;
```

Mit den im AVR definierten Befehle, hier 52h ein Byte senden, wird nun das anstehende Byte in[C] an den AVR übergeben und dort auf die V24 gesendet.

Bei dieser Methode ergibt das ungefähr die halbe mögliche Datenrate von 115kb/s, wenn's mehr werden soll, muss ein Zeilenbuffer vorgeschaltet werden.

Durch den Befehl STAT LST:=UL1: wird der Treiber aktiviert und mit CTRL-P dann der „Drucker“ zur Protokollierung zugeschaltet.

Quellenangaben:

Dokumentation Moppel-Floppy-Disk-Controller; (C) Cosmos Publikation 1984

CP/M 2.2 Assemblerlisting; ISBN 3-925074-11-2; Klaus Kämpf; Röckrath Mikrocomputer 1985

Beschreibung/Berechnung DPH,DPB: <http://hc-ddr.hucki.net/wiki/doku.php/cpm.systemdoku>

22DISK: <http://www.gaby.de/>

und natürlich die vielen Schnipsel im Internet