MZ-80B - intern Nr. 13

Mit Hilfe dieses MZ-80B intern können Sie an Ihrem CP/M-Betriebssystem einige Verbesserungen vornehmen. Es bezieht sich auf die Version 2.2.2.08, die am weitesten verbreitet ist. Im Gegensatz zu ersten Annahmen kann man übrigens das CP/M-Betriebssystem selbst, daß sich auf der ersten Spur der Diskette befindet, auch ohne ein Programm, das einzelne Sektoren modifizieren kann, ändern. Dazu wird die Diskette mit dem Betriebssystem wie üblich geladen und dann hinter das 'A>' einige Befehle eingegeben. Zuerst wird mit MOVCPM * * <CR> das Betriebssystem von der Systemspur in die TPA, also den freien Speicherbereich ab 0100H, geladen. Damit wird eine Meldung angezeigt, die für den nächsten Befehl benutzt wird, der nämlich SAVE 48 CPM64.COM <CR> lautet. Damit wird also das soeben geladene Betriebssystem in die Datei CPM64.COM abgelegt. Diese wird nun zur Bearbeitung mit dem Debugger aufgerufen, geben Sie also DDT CPM64.COM ein. Der DDT meldet sich mit einem Punkt. Um nun einzelne Speicherstellen verändern zu können, geben Sie ein S, direkt von der vierstelligen Adresse und <CR> gefolgt, ein. Der DDT zeigt Ihnen dann den derzeitigen Speicherinhalt, den Sie durch Eingabe einer zweistelligen Zahl und <CR> verändern bzw. nur mit <CR> belassen können. Geben Sie stattdessen . sowie <CR> ein, erscheint wieder der Punkt. Mit D, direkt gefolgt von einer vierstelligen Adresse, einem Leerzeichen, einer weiteren vierstelligen Adresse und <CR> können Sie den angegebenen Bereich in Hexzahlen und in ASCII anzeigen. Mit G0 gelangen Sie nach Abschluß und Kontrolle Ihrer Änderungen ins CP/M zurück, wo Sie jetzt sofort SYSGEN <CR> <CR> A <CR> <CR> ausführen, um das geänderte CP/M auf die Systemspur zu bringen. Das geänderte CP/M wird erst durch Urladen desselben auch benutzt!

Sie können anstatt der obigen Vorgehensweise aber auch ein Programm zum Ändern einzelner Sektoren benutzen. Deshalb kurz eine Erklärung der Diskettenparameter. Die Diskette wird mit 35 Spuren (Tracks) beschrieben (eine Spur sitzt je zur Hälfte auf beiden Seiten der Diskette), jede Spur enthält 80 Sektoren (Sectors), die wiederum 128 Byte beeinhalten. 16 Sektoren werden zu einer Gruppe (Group) zusammengefaßt, das entspricht 2 KByte. Deshalb wird der Diskettenspeicherplatz auch in 2-KByte-Sprüngen zugeteilt. Die Direktory faßt maximal 128 Einträge, ein Eintrag belegt 32 Byte, damit belegt die Direktory 4 KByte. Die komplette erste Spur nimmt das Betriebssystem auf, dies sind 10 KByte. Daraus ergibt sich die Speicherkapazität von 336 KByte, 35 Spuren * 80 Sektoren * 128 Byte - 14 KByte = 336 KByte.

Soweit die grundsätzlichen Erklärungen. Nehmen wir also an, Sie befinden sich jetzt nach der Durchführung der oben beschriebenen Prozedur im DDT und wollen Änderungen vornehmen. Dazu finden Sie zwischen 0900H und 0AFF die Informationen für den Urlader (IPLPRO CP/M usw., der Rest sind Nullen) und danach zwischen 0B00H und 30FF das eigentliche CP/M, und zwar zuerst das CP/M-Ladeprogramm, dann das BDOS und ab 2300H das BIOS, der systemspezifische Teil des CP/M, den wir ändern wollen. Dieser Bereich zwischen 2300H und 3041H findet sich, wenn das CP/M normal geladen wurde, zwischen EE00H und FB41H, den Rest bis FFFFH füllen Datenbereiche, die während des Betriebs benutzt werden. Schauen Sie ruhig mit dem DDT nach!

Die angesprochenen Bereiche lassen sich natürlich auch bestimmten Sektoren zuordnen. Die Informationen für den Urlader liegen in Spur 0 Sektor 1, Sektor 2 ist leer (00H), desgleichen Sektor 3 und 4 (E5H). Von Sektor 5 bis 8 findet sich das CP/M-Ladeprogramm, von Sektor 9 bis 52 das BDOS und von Sektor 53 bis 80 das BIOS. In Spur 1 Sektor 1 bis 31 liegt die Direktory.

Nun zur ersten Änderung. Wenn Sie sich nicht die Sharp-Laufwerke, sondern zum Beispiel die Slim-Line-Laufwerke zugelegt haben, werden Sie festgestellt haben, daß deren Zugriffsgeschwindigkeit, also die Geschwindigkeit der Schreib/Lesekopfpositionierung, nicht ausgenutzt wird. Dies liegt daran, daß die Sharp-Laufwerke schon etwas betagt sind und

diese Geschwindigkeit nicht bringen. Durch die folgende Änderungen nutzen Sie die Slim-Line-Laufwerke voll aus, was sich zum einen in weniger Lärm, zum anderen in fast doppelt so schnellem Zugriff (bei sehr vielen Kopfbewegungen, wie z. B. bei DBase) bemerkbar macht.

Adresse für DDT	(Adresse im CP/M)	alt	neu
2F19	(FA19)	F4	F7
2F38	(FA38)	E0	E3
303A	(FB3B)	F4	02
303B	(FB3B)	1A	06

Desweiteren können die Slim-Line-Laufwerke 40 Spuren pro Seite lesen und schreiben, im Gegensatz zu den Sharp-Laufwerken, die nur 35 Spuren pro Seite schaffen. Dies bringt immerhin weitere 50 KByte Speicherraum pro Diskette, die mit folgenden Änderungen erreicht werden:

Adresse für DDT	(Adresse im CP/M)	alt	neu
2D47	(F847)	95	55
2D57	(F857)	AB	6E
2D65	(F865)	55	15
2D67	(F867)	C1	55
2D75	(F875)	75	35
2D77	(F877)	D7	6E
2D7E	(F87E)	A9	C2

Mit dieser Änderung können Sie nur noch die Laufwerke A und B betreiben, da einige Buffer vergrößert werden mußten, die die von Laufwerk C und D z. T. überschreiben. Außerdem müssen Sie noch das FORMAT-Programm ändern, und zwar bei Adresse 02E8H von DCH auf D7H und bei Adresse 035DH von 23H auf 28H, und auch das BACKUP-Programm bei Adresse 0277H von 23H auf 28H. Diese drei Änderungen sind nötig, um auch 40 Spuren zu formatieren und zu kopieren. Die Vorgehensweise ist ähnlich, DDT FORMAT.COM bzw. DDT BACKUP.COM aufrufen, in diesem Fall die Zahl hinter NEXT merken, Änderungen vornehmen, G0 eingeben, und dann SAVE ??? FORMAT.COM/ BACKUP.COM, wobei an die Stelle der Fragezeichen die Dezimalzahl der vorderen beiden Stellen der hinter NEXT angegebenen Hexzahl gesetzt werden muß. Beispiel: Bei NEXT 3D50 müßte 61 stehen.

Eine weitere nützliche Änderung betrifft die Umlaute. Da CP/M ja im allgemeinen immer mit Kleinbuchstaben bedient wird, ist es praktisch, wenn auch die Umlaute im Normalfall Kleinbuchstaben liefern. Dazu dienen folgende Änderungen:

Adresse für DDT	(Adresse im CP/M)	alt	neu
2823	(F323)	5C	7C
2843	(F343)	7C	5C
2863	(F363)	5C	7C
2895	(F395)	5B	7B
2896	(F396)	5D	7D
28A5	(F3A5)	7B	5B
28A6	(F3A6)	7D	5D

Leider wirkt sich damit SHIFT-LOCK aber immer noch nur auf die Tasten A-Z aus, dabei bleiben die Umlaute unberücksichtigt. Dies wird allerdings nur ganz selten stören, die Lösung dafür ist unbekannt.

Nun zu einem anderen Problem, der Definierung der Funktionstasten. Dies ist mit den Bildschirm-Kontroll-Sequenzen sehr umständlich, außerdem benötigt man dann immer noch eine Hochsprache. Das abgedruckte Maschinenprogramm umgeht diese Sache, indem es direkt das CP/M verändert. Es sollte am besten mit TASTEN.COM benannt werden, dann werden durch bloße Eingabe von TASTEN die Funktionstasten entsprechend belegt, durch Eingabe von Control-Eckige-Klammer-Auf, gefolgt von Semikolon und <CR> (alles direkt hinter A>) wird die Initialisierung des Bildschirms bewirkt, was auch die alten Funktionstasten wiederherstellt. Das Programm wurde mit dem MACRO-80 erstellt (hierfür den ORG-Befehl beseiti-

gen), kann aber auch direkt mit dem DDT eingegeben werden, es benutzt einen freien Bereich im CP/M, der ab Adresse FB42H liegt und 94 Byte umfaßt. Aus diesem Grund darf die Anzahl der Zeichen der Funktionstasten, die mit zwei oder mehr Zeichen belegt werden, 93 nicht überschreiten! In der vorliegenden Version ist das Programm für den Wordstar belegt, es kann aber leicht geändert werden. Dazu wird die 60 Einträge umfassende Tabelle modifiziert, die Anzahl und Reihenfolge der Einträge muß gleich bleiben, da nur daraus die gewünschte Taste ermittelt wird! Es wird immer zuerst die Anzahl der Zeichen und dann die einzelnen Zeichen selbst angegeben. Hinter dem Semikolon stehen die Tasten mitsamt der Belegung in diesem Beispiel.

Besonders günstig ist es, mit dem CONFIG-Programm den nach dem Laden des CP/M ausgeführten Befehl INIT.SYS in z. B. SUBMIT START umzubennen, dann müssen Sie noch eine Datei mit dem Inhalt TASTEN und z. B. WS (für Worstar) erzeugen, und schon haben Sie ein automatisch startendes System, das nach dem Urladen sofort die Tastendefinition vornimmt und dann den Wordstar ausführt.

```
                        ;Definition der Funktionstasten für CP/M 2.2.2.08
                        ;Beispielbelegung für Wordstar
                        ;
                                .Z80                ;Z-80-Opcodes verwenden
                                PAGE    66          ;66 Zeilen pro Seite im Listing
                                ORG     0100H       ;Adresslage
                        ;
0100'  21 0142'         START:  LD      HL,TSTNDF   ;Beginn Tastendefinitionen
0103'  DD 21 FBBC               LD      IX,0FBBCH   ;1. Vektor der Tastentabelle im CP/M
0107'  FD 21 FB42               LD      IY,0FB42H   ;Beginn des 94-Byte-Buffer
010B'  7E               SCHLFE: LD      A,(HL)      ;Länge der Tastendefinition
010C'  23                       INC     HL          ;Zeiger eins weiter
010D'  FE FF                    CP      0FFH        ;Endzeichen?
010F'  C8                       RET     Z           ;ja -> Programmende
0110'  FE 01                    CP      01H         ;Länge=1?
0112'  20 10                    JR      NZ,LANG     ;nein -> Weiter bei LANG
0114'  7E               KURZ:   LD      A,(HL)      ;Zeichen, das die Taste liefern soll
0115'  DD 77 00                 LD      (IX),A      ;1. Byte des Vektors damit laden
0118'  DD 23                    INC     IX          ;Zeiger auf 2. Byte des Vektors
011A'  23                       INC     HL          ;Zeiger eins weiter
011B'  3E 00                    LD      A,00H       ;2. Byte des Vektors mit 0 laden
011D'  DD 77 00                 LD      (IX),A      ;
0120'  DD 23                    INC     IX          ;Zeiger auf 1. Byte des nächsten Vektors
0122'  18 E7                    JR      SCHLFE      ;nächste Taste
0124'  FD E5            LANG:   PUSH    IY          ;DE = IY
0126'  FD 77 00                 LD      (IY),A      ;Länge in Buffer
0129'  FD 23                    INC     IY          ;Zeiger eins weiter
012B'  D1                       POP     DE          ;
012C'  DD 73 00                 LD      (IX),E      ;Vektor mit Adresse, bei der die
012F'  DD 23                    INC     IX          ;Zeichen, die die Taste liefern soll,
0131'  DD 72 00                 LD      (IX),D      ;laden
0134'  DD 23                    INC     IX          ;Zeiger auf 1. Byte des nächsten Vektors
0136'  47                       LD      B,A         ;B=Länge der Tastendefinition
0137'  7E               LANG1:  LD      A,(HL)      ;Zeichen, das die Taste liefern soll
0138'  FD 77 00                 LD      (IY),A      ;in Buffer übertragen
013B'  23                       INC     HL          ;Zeiger eins weiter
013C'  FD 23                    INC     IY          ;Zeiger eins weiter
013E'  10 F7                    DJNZ    LANG1       ;Alle Zeichen übertragen
0140'  18 C9                    JR      SCHLFE      ;ja -> nächste Taste
                        ;
                        ;Anfang der Tabelle, der markierte Teil darf verändert werden
                        ;               !-----------!     !-----------------------------!
                        ;
0142'  02 0B 42         TSTNDF: DEFB    02H,0BH,"B"     ;F1:    ^KB     Block Anfang
0145'  02 0B 52                 DEFB    02H,0BH,"R"     ;F2:    ^KR     Block von Disk lesen
0148'  02 0B 56                 DEFB    02H,0BH,"V"     ;F3:    ^KV     Block verschieben
```

gen), kann aber auch direkt mit dem DDT eingegeben werden, es benutzt einen freien Bereich im CP/M, der ab Adresse FB42H liegt und 94 Byte umfaßt. Aus diesem Grund darf die Anzahl der Zeichen der Funktionstasten, die mit zwei oder mehr Zeichen belegt werden, 93 nicht überschreiten! In der vorliegenden Version ist das Programm für den Wordstar belegt, es kann aber leicht geändert werden. Dazu wird die 60 Einträge umfassende Tabelle modifiziert, die Anzahl und Reihenfolge der Einträge muß gleich bleiben, da nur daraus die gewünschte Taste ermittelt wird! Es wird immer zuerst die Anzahl der Zeichen und dann die einzelnen Zeichen selbst angegeben. Hinter dem Semikolon stehen die Tasten mitsamt der Belegung in diesem Beispiel.

Besonders günstig ist es, mit dem CONFIG-Programm den nach dem Laden des CP/M ausgeführten Befehl INIT.SYS in z. B. SUBMIT START umzubennen, dann müssen Sie noch eine Datei mit dem Inhalt TASTEN und z. B. WS (für Worstar) erzeugen, und schon haben Sie ein automatisch startendes System, das nach dem Urladen sofort die Tastendefinition vornimmt und dann den Wordstar ausführt.

```
                        ;Definition der Funktionstasten für CP/M 2.2.2.08
                        ;Beispielbelegung für Wordstar
                        ;
                                .Z80                ;Z-80-Opcodes verwenden
                                PAGE    66          ;66 Zeilen pro Seite im Listing
                                ORG     0100H       ;Adresslage
                        ;
0100'  21 0142'         START:  LD      HL,TSTNDF   ;Beginn Tastendefinitionen
0103'  DD 21 FBBC               LD      IX,0FBBCH   ;1. Vektor der Tastentabelle im CP/M
0107'  FD 21 FB42               LD      IY,0FB42H   ;Beginn des 94-Byte-Buffer
010B'  7E               SCHLFE: LD      A,(HL)      ;Länge der Tastendefinition
010C'  23                       INC     HL          ;Zeiger eins weiter
010D'  FE FF                    CP      0FFH        ;Endzeichen?
010F'  C8                       RET     Z           ;ja -> Programmende
0110'  FE 01                    CP      01H         ;Länge=1?
0112'  20 10                    JR      NZ,LANG     ;nein -> Weiter bei LANG
0114'  7E               KURZ:   LD      A,(HL)      ;Zeichen, das die Taste liefern soll
0115'  DD 77 00                 LD      (IX),A      ;1. Byte des Vektors damit laden
0118'  DD 23                    INC     IX          ;Zeiger auf 2. Byte des Vektors
011A'  23                       INC     HL          ;Zeiger eins weiter
011B'  3E 00                    LD      A,00H       ;2. Byte des Vektors mit 0 laden
011D'  DD 77 00                 LD      (IX),A      ;
0120'  DD 23                    INC     IX          ;Zeiger auf 1. Byte des nächsten Vektors
0122'  18 E7                    JR      SCHLFE      ;nächste Taste
0124'  FD E5            LANG:   PUSH    IY          ;DE = IY
0126'  FD 77 00                 LD      (IY),A      ;Länge in Buffer
0129'  FD 23                    INC     IY          ;Zeiger eins weiter
012B'  D1                       POP     DE          ;
012C'  DD 73 00                 LD      (IX),E      ;Vektor mit Adresse, bei der die
012F'  DD 23                    INC     IX          ;Zeichen, die die Taste liefern soll,
0131'  DD 72 00                 LD      (IX),D      ;laden
0134'  DD 23                    INC     IX          ;Zeiger auf 1. Byte des nächsten Vektors
0136'  47                       LD      B,A         ;B=Länge der Tastendefinition
0137'  7E               LANG1:  LD      A,(HL)      ;Zeichen, das die Taste liefern soll
0138'  FD 77 00                 LD      (IY),A      ;in Buffer übertragen
013B'  23                       INC     HL          ;Zeiger eins weiter
013C'  FD 23                    INC     IY          ;Zeiger eins weiter
013E'  10 F7                    DJNZ    LANG1       ;Alle Zeichen übertragen
0140'  18 C9                    JR      SCHLFE      ;ja -> nächste Taste
                        ;
                        ;Anfang der Tabelle, der markierte Teil darf verändert werden
                        ;               !-----------!     !-----------------------------!
                        ;
0142'  02 0B 42         TSTNDF: DEFB    02H,0BH,"B"     ;F1:    ^KB     Block Anfang
0145'  02 0B 52                 DEFB    02H,0BH,"R"     ;F2:    ^KR     Block von Disk lesen
0148'  02 0B 56                 DEFB    02H,0BH,"V"     ;F3:    ^KV     Block verschieben
```

```
014B'  02 0B 48          DEFB   02H,0BH,"H"     ;F4:   ^KH     Blockhervorhebung weg
014E'  02 11 46          DEFB   02H,11H,"F"     ;F5:   ^QF     Suchen
0151'  01 14             DEFB   01H,14H         ;F6:   ^T      Wort löschen
0153'  02 11 59          DEFB   02H,11H,"Y"     ;F7:   ^QY     Bis Zeilende löschen
0156'  01 19             DEFB   01H,19H         ;F8:   ^Y      Zeile löschen
0158'  01 02             DEFB   01H,02H         ;F9:   ^B      Absatz formatieren
015A'  01 11             DEFB   01H,11H         ;F10:  ^Q      Merker aufsuchen
015C'  01 38             DEFB   01H,"8"         ;8:    8       Zahl 8
015E'  01 39             DEFB   01H,"9"         ;9:    9       Zahl 9
0160'  02 30 30          DEFB   02H,"0","0"     ;00:   00      Zahlen 0 0
0163'  01 2E             DEFB   01H,"."         ;.:    .       Dezimalpunkt .
0165'  01 2B             DEFB   01H,"+"         ;+:    +       Pluszeichen +
0167'  01 2D             DEFB   01H,"-"         ;-:    -       Minuszeichen -
0169'  01 30             DEFB   01H,"0"         ;0:    0       Zahl 0
016B'  01 31             DEFB   01H,"1"         ;1:    1       Zahl 1
016D'  01 32             DEFB   01H,"2"         ;2:    2       Zahl 2
016F'  01 33             DEFB   01H,"3"         ;3:    3       Zahl 3
0171'  01 34             DEFB   01H,"4"         ;4:    4       Zahl 4
0173'  01 35             DEFB   01H,"5"         ;5:    5       Zahl 5
0175'  01 36             DEFB   01H,"6"         ;6:    6       Zahl 6
0177'  01 37             DEFB   01H,"7"         ;7:    7       Zahl 7
0179'  01 05             DEFB   01H,05H         ;^:    ^E      Cursor eine Zeile höher
017B'  01 18             DEFB   01H,18H         ;v:    ^X      Cursor eine Zeile tiefer
017D'  01 13             DEFB   01H,13H         ;<:    ^S      Cursor ein Zeichen links
017F'  01 04             DEFB   01H,04H         ;>:    ^D      Cursor ein Zeichen rechts
0181'  02 0B 53          DEFB   02H,0BH,"S"     ;HOME: ^KS     Text sichern, zum Anfang
0184'  01 7F             DEFB   01H,7FH         ;DEL:  DEL     Zeichen löschen
0186'  02 0B 4B          DEFB   02H,0BH,"K"     ;S-F1: ^KK     Block Ende
0189'  02 0B 57          DEFB   02H,0BH,"W"     ;S-F2: ^KW     Block auf Disk schreiben
018C'  02 0B 59          DEFB   02H,0BH,"Y"     ;S-F3: ^KY     Block löschen
018F'  02 0B 43          DEFB   02H,0BH,"C"     ;S-F4: ^KC     Block kopieren
0192'  02 11 41          DEFB   02H,11H,"A"     ;S-F5: ^QA     Austauschen
0195'  01 0C             DEFB   01H,0CH         ;S-F6: ^L      Suche/Austausch wiederh.
0197'  02 0F 44          DEFB   02H,0FH,"D"     ;S-F7: ^OD     Steuerzeichenanzeige e/a
019A'  02 11 51          DEFB   02H,11H,"Q"     ;S-F8: ^QQ     Befehl wiederholen
019D'  01 15             DEFB   01H,15H         ;S-F9: ^U      Befehl abbrechen
019F'  01 0B             DEFB   01H,0BH         ;S-F10: ^K     Merker setzen
01A1'  02 0F 47          DEFB   02H,0FH,"G"     ;S-8:  ^OG     Absatz einrücken
01A4'  02 0F 43          DEFB   02H,0FH,"C"     ;S-9:  ^OC     Zeile zentrieren
01A7'  02 10 59          DEFB   02H,10H,"Y"     ;S-00: ^PY     Breitschrift e/a
01AA'  02 10 53          DEFB   02H,10H,"S"     ;S-.:  ^PS     Unterstreichen e/a
01AD'  02 0F 52          DEFB   02H,0FH,"R"     ;S-+:  ^OR     Rechten Rand setzen
01B0'  02 10 4F          DEFB   02H,10H,"O"     ;S--:  ^PO     Untrennbarer Leerschritt
01B3'  02 10 42          DEFB   02H,10H,"B"     ;S-0:  ^PB     Fettdruck e/a
01B6'  02 10 4E          DEFB   02H,10H,"N"     ;S-1:  ^PN     Normale Schriftdichte
01B9'  02 10 41          DEFB   02H,10H,"A"     ;S-2:  ^PA     Alternative Schriftdichte
01BC'  02 10 52          DEFB   02H,10H,"R"     ;S-3:  ^PR     Selbstdef. Steuerzeichen
01BF'  02 0F 49          DEFB   02H,0FH,"I"     ;S-4:  ^OI     Tabulator setzen
01C2'  02 0F 4E          DEFB   02H,0FH,"N"     ;S-5:  ^ON     Tabulator löschen
01C5'  02 10 48          DEFB   02H,10H,"H"     ;S-6:  ^PH     überdrucken e/a
01C8'  02 0F 4C          DEFB   02H,0FH,"L"     ;S-7:  ^OL     Linken Rand setzen
01CB'  01 12             DEFB   01H,12H         ;S-^:  ^R      Fenster höher
01CD'  01 03             DEFB   01H,03H         ;S-v:  ^C      Fenster tiefer
01CF'  01 01             DEFB   01H,01H         ;S-<:  ^A      Fenster links
01D1'  01 06             DEFB   01H,06H         ;S->:  ^F      Fenster rechts
01D3'  04 0B 53 11       DEFB   04H,0BH,"S",11H,"P";CLR:^KS^QP  Text sichern, alte Position
01D7'  50
01D8'  01 16             DEFB   01H,16H         ;INST: ^V      Einfügen e/a
                         ;
                         ;Ende der Tabelle
                         ;
01DA'  FF                DEFB   0FFH            ;Kennzeichnung des Endes
                         ;
                         END
```

Umsteiger von Sharp-BASIC auf CP/M mit seinen wesentlich leistungsfähigeren Hochsprachen stehen oft vor der Situation, Ihre unter Sharp-BASIC erstellt Software neu und leicht verändert neu eintippen zu müssen, um sie dann z. B. mit MBASIC-80 laufen zu lassen. Zumindest das Eintippen nimmt einem die folgende Routine ab, sie umgeht den Nachteil des CMT-Programms, keine BASIC-Programme lesen zu können, in dem das BASIC-Programm als sequentielle Datei auf Kassette aufgezeichnet wird. Laden Sie also zuerst BASIC SB-6510, dann laden Sie das zu übertragende BASIC-Programm, es kann natürlich auch mit BASIC SB-5510 geschrieben worden sein, danach tippen Sie die fünf abgedruckten Zeilen ein (sie überschreiben hoffentlich nichts) und starten das Programm mit RUN 65531. Es zeichnet das Programm und auch sich selbst als sequentielle Datei auf Kassette auf, dies dauert eine Weile, Sie können den Vorgang am Bildschirm verfolgen.

Nun können Sie CP/M laden und CMT Name/N eingeben. Damit wird die Datei gelesen und unter Name auf Diskette abgespeichert. Zuletzt nehmen Sie einen Editor (z. B. Wordstar) und löschen die erste Zeile der auf Diskette übertragenen Datei, die ist nämlich leer, sowie die fünf Zeilen und die beiden letzten Zeilen (eine ist leer, eine besteht aus einem ß). Damit ist die übertragung komplett. Ein kleiner Hinweis zum CMT-Programm: Der angegebene Name bezieht sich auf den Namen, den die Datei auf der Diskette erhalten soll. Der Schrägstrich und das N sind unbedingt nötig, damit überhaupt etwas geladen wird. Dies wird oft vergessen.

```
 65531 D=5588:A=26460:B=65280:LIMITB:POKED+4,0:POKED+5,255:PRINT:PRINT"BASIC ->
BSD":PRINT:INPUT"Dateiname: ";N$:PRINT"übertragung beginnt!":WOPEN/TN$
 65532 A2=INT((A+2)/256):A1=A+2-INT(A2)*256:POKED+1,A1:POKED+2,A2:A2=INT((A+4)/2
56):A1=A+4-INT(A2)*256:POKED+16,A1:POKED+17,A2:USR(D):P=B:Z$=""
 65533 Z=PEEK(P):Z$=Z$+CHR$(Z):IFZ<>13THENP=P+1:GOTO65533
 65534 PRINTZ$:PRINT/TCHR$(10)+Z$:IFPEEK(A+2)+PEEK(A+3)*256<>65535THENA=PEEK(A)+
PEEK(A+1)*256:GOTO65532
 65535 PRINT/TCHR$(10):CLOSE/T:POKED+1,16:POKED+2,97:POKED+4,13:POKED+5,96:POKED
+16,18:POKED+17,97:PRINT"übertragung ist abgeschlossen!":PRINT:END
```

Zum Schluß noch ein kleiner Tip für die Benutzer von Turbo-PASCAL: Wenn ein Programm auf Diskette zugreift bzw. die Diskettenstation noch läuft, werden während des Programmlaufs diese nicht mehr abgeschaltet, was ja normalerweise passiert. Abhilfe schafft ein winziges Maschinenprogramm, nämlich XOR A: OUT (ODCH),A. Damit werden die Motoren abgeschaltet. Sie können natürlich auch das vordefinierte Port-Array benutzen.

* * *

SB 6510 — SB 5510

D = 5588 = $ 15D4 → (5376 = $ 15ØØ) = Zeile umstellen von Zwischencode in ASCII

A = 26460 = $ 675C → (20764 = $ 511C)

B = 65280 = $ FFØØ