

Für andere DOS Versionen

Um das MRD-Programm mit anderen DOS-Versionen betreiben zu können, muß deren höchste Adresse als Programmbeginn des MRD-Programmes gewählt werden. Dies ist nur durch erneutes Assemblieren des Listings 2 mit geänderter Anfangsadresse möglich. Für DOS-XL (von OSS) verwenden Sie bitte *=\$2200, für DOS-III *=\$1D00 (mit File-Buffer).

Das Assemblerlisting wurde mit MAC/65 erstellt, kann aber auch mit der Editor/Assembler Cartridge assembliert werden.

In der nächsten Assembler-ecke dreht sich wieder alles um Scrolling. Im 2. Teil geht es um die horizontale Bildverschiebung. Dann sind auch wieder die Besitzer der älteren 400/800-Geräte mit von der Partie.

Peter Finzel

Assembler-Listing

```

0100 ;*****
0110 ;
0120 ; Mini-RAMDISK Handler
0130 ; fuer XL-Computer
0140 ; Peter Finzel 1985
0150 ;
0160 ;*****
0170 ;
0180 ;Hardware-Variable
0190 ;
=040E 0200 NMJEN = $D40E
=0301 0210 MEMSEL = $D301
=E000 0220 ZSATZ = $E000 RDM-Zeichensatz
=FFFA 0230 NMIVEC = $FFFA Hardware-Vektoren
=FFFE 0240 IRGVEC = $FFFE
0250 ;
0260 ; O.S. Variable/Konstante.
0270 ;
=00E0 0280 FR1 = $E0 Zwischenspeicher
=00C0 0290 DOSINI = $0C FMS Init.-Vektor
=02E7 0300 LOMEM = $02E7
=031A 0310 HATABS = $031A
=034A 0320 ICAUX1 = $034A AUX1 Byte des IOCB#0
0330 ;
=FFF0 0340 RDTOP = $FFF0 Hoechste benutzbare Adresse
=0001 0350 STOK = 1 Status ist O.K.
=00BB 0360 STEOF = $BB End-of-File Fehlercode
=00A9 0370 STFULL = $A9 DISK-FULL Fehler!
=0008 0380 OPDUT = $08 Open-Output Anweisung
=0100 0390 HI = $0100 Divisor fuer MSB
=00FF 0400 LO = $FF Maske fuer LSB
0410 ;
0420 ;
0430 ;*****
0440 ;MiniRamDisk-Handler wird vorbereitet
0450 ;*****
0460 ;
0470 ; *=$1F00 ;ab DOS-Obergrenze
0480 ;
0490 START JSR DISET DOSINI einrichten
0500 JSR XLINIT Zusatz-RAM vorbereiten
0510 LDX #0 Handler Tabelle nach
0520 LDA HATABS,X Gerateeintraege absuchen
0530 WEITER BEQ EINTRAG leerer Eintrag?
0540 CMP #'X 'X' schon vorhanden?
0550 BEQ FERTIG Ja, dann nicht nochmal -->
0560 INX
0570 INX
0580 INX ; naechsten Eintag
0590 BNE WEITER und weiter geht's-->
0600 RTS Wie bitte? Nicht moeglich!!
0610 ;
0620 EINTRAG LDA #'X MRD wird mit X:
0630 STA HATABS,X angesprochen
0640 LDA #XTAB&LO Adresse der Handler-
0650 STA HATABS+1,X tabelle in HATABS
0660 LDA #XTAB/HI eintragen
0670 STA HATABS+2,X
0680 LDA #0 neues Ende von HATABS
0690 STA HATABS+3,X festlegen
0700 ;
0710 LDA #PGMEND&LD jetzt noch LOMEM
0720 STA LOMEM auf das Ende des
0730 LDA #PGMEND/HI MiniRamDisk Pgms
0740 STA LOMEM+1 setzen
0750 FERTIG RTS
0760 ;
0770 ;*****
0780 ;Die Handler-tabelle...
0790 ;*****
0800 ;
0810 XTAB .WORD OPEN-1
0820 .WORD CLOSE-1
0830 .WORD GET-1
0840 .WORD PUT-1
0850 .WORD STATUS-1
0860 .WORD XIO-1
0870 JMP INIT
0880 ;
0890 ;*****
0900 ;Leere Eintraege
0910 ;*****
0920 STATUS
0930 XIO
0940 INIT LDY #STOK Status ist O.K.
0950 RTS
0960 ;
0970 ;*****
0980 ;Open-Routine
0990 ;*****

1F4B A900 1000 ;
1F4A BD6C20 1010 OPEN LDA #0 Zeiger auf Buffer-
1F4D BD6D20 1020 STA ZEIGER Anfang setzen
1F50 A001 1030 STA ZEIGER+1
1F52 60 1040 LDY #STOK kein Fehler moeglich!
1050 RTS
1060 ;
1070 ;*****
1080 ;CLOSE-Routine
1090 ;*****
1100 ;
1110 CLOSE LDA ICAUX1,X Wurde eine neue
1120 AND #OPDUT Datei geschrieben?
1130 BEQ CLENDE Nein, nur gelesen-->
1140 ;
1150 LDA ZEIGER Zeiger als ENDE merken.
1160 STA RDENDE
1170 LDA ZEIGER+1 jetzt MSB
1180 STA RDENDE+1
1190 ;
1200 CLENDE LDY #STOK Alles klar!
1210 RTS
1220 ;
1230 ;
1240 ;*****
1250 ;Ein Byte in MiniRD schreiben
1260 ;*****
1270 ;
1280 PUT PHA ;Datenbyte merken
1290 LDY ZEIGER Zeiger nach Zeropage
12A0 STY FR1
12B0 ;
12C0 ;MSB in physikalische Adresse umrechnen
12D0 ;
12E0 LDA ZEIGER+1 jetzt MSB
12F0 PHA ;MSB-Zeiger
1300 LSR A ;geteilt durch 4
1310 LSR A
1320 TAX ;als Index in Memtab
1330 PLA ;Zeiger
1340 AND #3 Bit 1 & 2 maskieren
1350 ORA MEMTAB,X physikalische Adresse
1360 STA FR1+1
1370 ;
1380 ;Noch Platz in MiniRD?
1390 ;
13A0 CPY #RDTOP&LO MRD voll (LSB)
13B0 BNE PLATZ noch Platz vorhanden-->
13C0 CMP #RDTOP/HI MSB noch pruefen
13D0 BEQ VOLL kein Platz mehr!-->
13E0 ;
13F0 ;Datenbyte ins RAM eintragen
1400 ;
1410 PLATZ LDA MEMSEL auf Zusatzram schalten
1420 AND #FE Bit 0 gleich 0 waehlt RAM
1430 STA MEMSEL
1440 PLA ;Datenbyte wieder holen
1450 LDY #0
1460 STA (FR1),Y uod in MRD schreiben
1470 LDA MEMSEL O.S. wieder ein
1480 ORA #1 Bit 0 auf 1 waehlt ROM
1490 STA MEMSEL
1500 ;
1510 ;File-Zeiger auf naechstes freies Byte richten
1520 ;
1530 PLATZ LDA MEMSEL auf Zusatzram schalten
1540 AND #FE Bit 0 gleich 0 waehlt RAM
1550 STA MEMSEL
1560 PLA ;Datenbyte wieder holen
1570 LDY #0
1580 STA (FR1),Y uod in MRD schreiben
1590 LDA MEMSEL O.S. wieder ein
1600 ORA #1 Bit 0 auf 1 waehlt ROM
1610 STA MEMSEL
1620 ;
1630 ;File-Zeiger auf naechstes freies Byte richten
1640 ;
1650 INC ZEIGER Zeiger weiter
1660 BNE PUT1 MSB ist O.K.-->
1670 ;
1680 INC ZEIGER+1 MSB Uebertrag
1690 ;
16A0 A001 1700 PUT1 LDY #STOK keinerlei Fehler
16A5 60 1710 RTS
1720 ;
1730 VOLL PLA ;Datenbyte vom Stapel
1740 LDY #STFULL MRD ist voll
1750 RTS
1760 ;
1770 ;*****
1780 ;Ein Byte aus MiniRD holen
1790 ;*****
1800 ;
1810 GET LDY ZEIGER Zeiger nach Zeropage
1820 STY FR1
1830 LDA ZEIGER+1
1840 CPY RDENDE letztes Byte in File?
1850 BNE NOCH noch Bytes vorhanden-->
1860 CMP RDENDE+1 MSB noch pruefen
1870 BEQ LEER nichts mehr da!-->
1880 NOCH PHA ;phys. Adresse ermitteln
1890 LSR A ;geteilt durch 4
1900 LSR A
1910 TAX ;als Index in Memtab
1920 PLA ;Zeiger
1930 AND #3 Bit 1 & 2 maskieren
1940 ORA MEMTAB,X physikalische Adresse
1950 STA FR1+1 in Zeropage
1960 LDA MEMSEL Zusatz-RAM ein
1970 AND #FE
1980 STA MEMSEL
1990 LDY #0
2000 LDA (FR1),Y Byte aus MRD holen
2010 PHA ;Datenbyte merken
2020 LDA MEMSEL auf OS zurueckschalten
2030 ORA #1
2040 STA MEMSEL
2050 INC ZEIGER Zeiger weiter
2060 BNE GET1 MSB ist O.K.-->
2070 INC ZEIGER+1 MSB Uebertrag
2080 GET1 PLA ;Datenbyte
2090 LDY #STOK keinerlei Fehler
2100 RTS
2110 ;
2120 LEER LDY #STEOF MRD ist leer!
2130 RTS
2140 ;
2150 ;*****
2160 ;Sicherung gegen SYSTEM RESET
2170 ;*****
2180 DISET LDA DOSINI+1 wurde DOSINI schon
2190 CMP #DIPGM/HI berichtet?
2200 BEQ DISEND ja, stimmt schon!-->
2210 ;

```

Listing 1

```

1FF2 8D0520      2220      STA DIPGM+2      sonst die Adresse in
1FF5 A50C        2230      LDA DOSINI        den JSR-Befehl ein-
1FF7 BD0420      2240      STA DIPGM+1      setzten!
1FFA A903        2250      LDA #DIPGM&L0  DOSINI auf neue
1FFC B50C        2260      STA DOSINI        Routine richten
1FFE A920        2270      LDA #DIPGM/Hi
2000 B50D        2280      STA DOSINI+1
2002 60          2290 DISEND  RTS
                2300 ;
                2310 ;modifizierte DOSINI-Routine
                2320 ;
2003 204015      2330 DIPGM  JSR $1540      ;!! Hier wird DOSINI
2006 4C061F      2340 JMP DIEIN      ;eingesetzt!!
                2350 ;
                2360 ;
                2370 ;*****
2009 7B          2420 XLINIT  SEI          ;alle Interrupts
200A A900        2430      LDA #0          ;ausschalten
200C BD0ED4      2440      STA NMIEIN
200F AD01D3      2450      LDA MEMSEL      Zusatz-Ram anwaehlen
2012 29FE        2460      AND #FE
2014 BD01D3      2470      STA MEMSEL
2017 A96B        2480      LDA #INTEND&255 Interruptvektoren
2019 BDAFF        2490      STA NMIVEC      auf 'Dummy' RTI-
201C BDFEFF      2500      STA IRQVEC      Befehl richten
201F A920        2510      LDA #INTEND/256
2021 BDFBFF      2520      STA NMIVEC+1
2024 BDFFFF      2530      STA IRQVEC+1
                2540 ;
                2550 ; ROM-Zeichensatz ins RAM kopieren
                2560 ;
2027 A200        2570      LDX #0
2029 AD01D3      2580 ZCOPY   LDA MEMSEL      jetzt wieder ROM
202C 0901        2590      ORA #1
202E BD01D3      2600      STA MEMSEL
2031 BD00E0      2610      LDA ZSATZ,X      vier Bytes Zeichensatz
2034 4B          2620      PHA          ;auf den Stack
2035 BD00E1      2630      LDA ZSATZ+256,X
2038 4B          2640      PHA
2039 BD00E2      2650      LDA ZSATZ+512,X
203C 4B          2660      PHA
203D BD00E3      2670      LDA ZSATZ+768,X
2040 4B          2680      PHA
2041 AD01D3      2690      LDA MEMSEL      RAM einschalten
2044 29FE        2700      AND #FE
2046 BD01D3      2710      STA MEMSEL
2049 6B          2720      PLA
204A 9D00E3      2730      STA ZSATZ+768,X
204D 6B          2740      PLA
204E 9D00E2      2750      STA ZSATZ+512,X
2051 6B          2760      PLA
2052 9D00E1      2770      STA ZSATZ+256,X
2055 6B          2780      PLA
2056 9D00E0      2790      STA ZSATZ,X
2059 EB          2800      INX
205A D0CD        2810      BNE ZCOPY      ;256 Durchlaeufer-->
                2820 ;
205C AD01D3      2830      LDA MEMSEL      fertig, OS-ROM ein
205F 0901        2840      ORA #1
2061 BD01D3      2850      STA MEMSEL
2064 A940        2860      LDA #*40      NMI wieder ein
2066 BD0ED4      2870      STA NMIEIN
2069 5B          2880      CLI          ;IRQ wieder an
206A 60          2890      RTS
                2900 ;
                2910 ;
                2920 INTEND  RTI          ;fuer Interrupts in RAM-Phase
                2930 ;
                2940 ;*****
206B 40          2950 ;interne Variablen
                2960 ;*****
                2970 ;

```

PAGE 6

```

206C 0000      2980 ZEIGER  .WORD 0      Zeiger auf aktuelles Byte
206E 0000      2990 RDEnde  .WORD 0      log. Endadresse
                3000 ;
                3010 ;*****
2070 C0C4CBCC   3020 ;Umrechnungstabelle zur Bestimmung der physikalischen
2074 DBDCE4EB   3030 ;Adresse ($C000 - $FFFF, mit Unterbrechungen) aus
2078 ECF0F4FB   3040 ;der logischen Position im File ($0000-$33F0)
207C FC         3050 ;*****
                3060 ;
                3070 MEMTAB  .BYTE $C0,$C4,$CB,$CC
                3080      .BYTE $DB,$DC,$E4,$EB
                3090      .BYTE $EC,$F0,$F4,$FB
                3100      .BYTE $FC
                3110 ;
                3120 ;
                3130 PGMEND  = *          ;LOMEM-Grenze liegt hier
                3140 ;
                3150 ;RUN-Adresse zum selbstaendigen Programmstart
                3160 ;
207D           3170      == $02E0
02E0 001F      3180      .WORD START
02E2           3190      .END

```

ASSEMBLY ERRORS: 0 2012B BYTES FREE

```

10 REM ** AUTORUN.SYS GENERIERUNG **
20 REM ** P. FINZEL 1985
100 DIM D$(80):OPEN #1,8,0,"D:AUTORUN.
SYS"
110 ? "AUTORUN.SYS wird generiert...":
?
120 READ D$:READ P:IF D$="" THEN 200
130 S=0:?"*";
140 FOR I=1 TO LEN(D$) STEP 2
150 H=ASC(D$(I,I))-48:L=ASC(D$(I+1,I+1))-48
160 D=(H-(H>9)*7)*16+L-(L>9)*7:S=S+D:P
UT #1,D
170 NEXT I:IF S=P THEN 120
180 ? :? :? "DATENFEHLER IN ZEILE ";PE
EK(183)+PEEK(184)*256:CLOSE #1:STOP
200 REM * FILE SCHLIESSEN
210 CLOSE #1
220 ? :? :? "AUTORUN.SYS ordnungsgemae
ss erzeugt!"
230 END
1100 DATA FFFF001FFB1F20EC1F200920A200
BD1A03F00AC958F024E8EB,2854
1110 DATA E8D0F260A9589D1A03A9369D1B03
A91F9D1C03A9009D1D03A9,2546
1120 DATA 7D8DE702A9208DE80260471F521F
A91F6B1F441F441F4C451F,2095
1130 DATA A00160A9008D6C208D6D20A00160
BD4A032908F00CAD6C208D,2267
1140 DATA 6E20AD6D208D6F20A0016048AC6C
2084E0AD6D20484A4AAA68,2545
1150 DATA 29031D702085E1C0F0D004C9FFFF0
20AD01D329FE8D01D368A0,3244
1160 DATA 0091E0AD01D309018D01D3EE6C20
D003EE6D20A0016068A0A9,2775
1170 DATA 60AC6C2084E0AD6D20CC6E20D005
CD6F20F02D484A4AAA6829,2805
1180 DATA 031D702085E1AD01D329FE8D01D3
A000B1E048AD01D309018D,2736
1190 DATA 01D3EE6C20D003EE6D2068A00160
A08B60A50DC920F0108D05,2746
1200 DATA 20A50C8D0420A903FC1F7C20850C
A920850D602040154C061F,1815
1210 DATA 78A9008D0ED4AD01D329FE8D01D3
A96B8DFAFF8DFEFA9208D,3603
1220 DATA FBFF8DFFFFA200AD01D309018D01
D3BD00E048BD00E148BD00,3227
1230 DATA E248BD00E348AD01D329FE8D01D3
689D00E3689D00E2689D00,3055
1240 DATA E1689D00E0E8D0CDAD01D309018D
01D3A9408D0ED458604000,2951
1250 DATA 000000C0C4C8CCD8DCE4E8ECF0F4
FBFCE002E102001F,3392
1260 DATA *,0

```