

```

10 ;
20 ; Error-Routine fuer Lesen/Schreiben/Formatieren/ue
    berpruefen
30 ;
40 ; (c) 1988 by Mercurysoft
50 ;
60         ent    $
70
80 error:  ld     a, (#BE4C)    ; Statusregister 0 (ST0)
                                ; holen
90         bit    2,a          ; Testen, ob Laufwerk
                                ; bereit ist
100        ld     hl,fmdnr     ; Fehlermeldung: Drive not
                                ; ready
110        jr     nz,error1    ; ausgeben
120
130        ld     a, (#BE4D)    ; ST1 holen
140        bit    1,a          ; Testen, ob Diskette
                                ; schreibgeschuetzt
150        ld     hl,fmdwp     ; Disc is write protected
160        jr     nz,error1
170        bit    2,a          ; Testen, ob Sektor
                                ; vorhanden
180        ld     hl,fmnsf     ; No sector found
190        jr     nz,error1
200        bit    0,a          ; Testen, ob Spur
                                ; formatiert ist
210        ld     hl,fmtnf     ; Track is not formatted
220        jr     nz,error1
230
240        ld     a, (#BE4E)    ; ST2 holen
250        bit    6,a          ; Testen, ob geloeschte
                                ; AM vorliegt

```

```

260          ld    hl,fmdam      ; Deleted adress mark
270          jr    nz,error1
280
290 ; Hier koennen Sie noch weitere Bits testen,
300 ; so z.B. wenn Sie SPUR UEBERPRUEFEN verwenden.
310 ; Durch diese Routine werden fast alle Fehler,
320 ; die auftreten koennen, abgefangen.
330
340          ret                  ; zurueck, da kein Fehler
                                ; aufgetreten
350
360 error1:  ld    a,(hl)        ; Fehlermeldung ausgeben
370          or     a              ; Endkennzeichen ?
380          ret    z              ; dann zurueck
390          call  #BB5A          ; Zeichen ausgeben
400          inc   hl              ; Zeiger erhoehen
410          jr    error1         ; ggf. naechstes
                                ; Zeichen

```

```

10 ;
20 ; Error-Routine fuer Laufwerksstatus ermitteln
30 ;
40          ent    $
50
60 error:   ld    a,(#BE4C)      ; ST3 holen
70          bit    5,a            ; Testen, ob Laufwerk
                                ; bereit ist
80          ld    hl,fmdnr      ; Fehlermeldung: Drive not
                                ; ready
90          jr    z,error1      ; ausgeben
100         bit    6,a            ; Testen, ob Diskette
                                ; schreibgeschuetzt
110         ld    hl,fmdwp      ; Disc is write protected
120         jr    nz,error1
130         ret                  ; zurueck, da kein Fehler
                                ; aufgetreten
140
150 error1: .....              ; siehe Listing 1

```

```

100 ;*****
110 ;Sprungtabelle fuer die
120 ;Window-Routinen
130
140 ORG &A000
150
160 JP INVERS
170 JP INITW
180 JP OPENW
190 JP CLOSEW
200
500 ;*****
510 ;Speicherplatz fuer die
520 ;Window-Formulare
530
540 ANZAHL DS 1 ;akt. Anzahl Windows
550 FORMUL DS 50 ;Platz f. 8 Window-
560 ;Parameterbloেকে
570
1000 ;*****
1010 ;Bildschirmausschnitt invertieren
1020 ;Aufruf:
1030 ;CALL adr,links,rechts,oben,unten
1040
1050 INVERS ;siehe letzte Folge
1060
2000 ;*****
2010 ;Window-Verwaltung initialisieren
2020 ;Aufruf: CALL adr,Startadresse
2030
2040 INITW XOR A ;Akku loeschen
2050 LD (ANZAHL),A ;Anzahl Windows=0
2060 LD L,(IX+0) ;Lobyte Startadr.
2070 LD H,(IX+1) ;Hibyte Startadr.
2080 LD (FORMUL),HL ;Adresse eintragen
2090 RET ;-> Basic
2100
3000 ;*****
3010 ;Fenster oeffnen und Hintergrund
3020 ;abspeichern
3030 ;Aufruf:
3040 ;CALL adr,links,rechts,oben,unten
3050
3060 GPAPER EQU &BB99 ;TXT GET PAPER
3070 ENCODE EQU &BC2C ;SCR INK ENCODE
3080
3090 OPENW CALL WINPAR ;Basic-Params
3100 CALL WINADR ;-> Screenadr.
3110
3120 PUSH HL ;Screenadr. retten
3130 LD HL,ANZAHL
3140 LD A,(HL) ;A = Anz. Windows
3150 INC (HL) ;Anzahl erhoehen
3160 LD E,A
3170 ADD A,A ;* 2
3180 ADD A,E ;* 3
3190 ADD A,A ;* 6
3200 LD L,A ;Ergebnis nach HL
3210 LD H,0
3220 LD DE,FORMUL ;Start Formulare
3230 ADD HL,DE ;addieren
3240 PUSH HL ;Start Param.block
3250 POP IY ;nach IY
3260 LD E,(IY+0) ;Start freier
3270 LD D,(IY+1) ;Speicher nach DE
3280 POP HL ;Screenadr. holen
3290 LD (IY+2),L ;in Param.block
3300 LD (IY+3),H ;eintragen
3310 LD (IY+4),B ;Breite eintragen
3320 LD (IY+5),C ;Hoehe eintragen
3330
3340 CALL GPAPER ;Paper-Nr. holen
3350 CALL ENCODE ;Farbbyte erzeugen

```

```

3360
3370 OLOOP1 PUSH BC ;Breite/Hoehe und
3380 PUSH HL ;Screenadr. retten
3390 LD C,A ;Farbbyte nach C
3400 OLOOP2 LD A,(HL);Screenbyte
3410 LD (DE),A ;speichern und
3420 LD (HL),C ;dafuer Farbbyte
3430 INC HL ;Screenadr.+1
3440 INC DE ;Speicheradr.+1
3450 DJNZ OLOOP2 ;-> naechstes Byte
3460 POP HL ;Startadr. Zeile
3470 CALL LDOWN ;Zeile nach unten
3480 LD A,C ;Farbbyte nach A
3490 POP BC ;Breite/Hoehe
3500 DEC C ;Hoehe - 1
3510 JR NZ,OLOOP1 ;-> noch nicht 0
3520
3530 LD (IY+6),E ;naechste freie
3540 LD (IY+7),D ;Speicheradresse
3550 ;eintragen
3560 RET ;-> Basic
3570
4000 ;*****
4010 ;Fenster schliessen, Hintergrund
4020 ;restaurieren
4030 ;Aufruf:CALL adr (ohne Parameter)
4040
4050 CLOSEW LD A,(ANZAHL);Anz. Windows
4060 OR A ;= 0?
4070 RET Z ;falls ja zurueck
4080 DEC A ;sonst Anzahl-1
4090 LD (ANZAHL),A ;merken
4100 LD E,A
4110 ADD A,A ;* 2
4120 ADD A,E ;* 3
4130 ADD A,A ;* 6
4140 LD L,A ;Ergebnis nach HL
4150 LD H,0
4160 LD DE,FORMUL ;Start Formulare
4170 ADD HL,DE ;addieren
4180 PUSH HL ;Start Param.block
4190 POP IY ;nach IY
4200 LD L,(IY+0) ;Adresse Hintergr.
4210 LD H,(IY+1) ;nach HL
4220 LD E,(IY+2) ;Screenadr. li/ob
4230 LD D,(IY+3) ;nach DE
4240 LD C,(IY+4) ;Breite nach C
4250 LD B,(IY+5) ;Hoehe nach B
4260
4270 CLOOP PUSH BC ;Hoehe/Breite und
4280 PUSH DE ;Screenadr. retten
4290 LD B,0 ;Hibyte BC = 0
4300 LDIR ;Zeile aus Speicher
4310 ;zum Bildschirm
4320 POP DE ;Start Screenzeile
4330 EX DE,HL ;nach HL
4340 CALL LDOWN ;Zeile nach unten
4350 EX DE,HL ;und wieder nach DE
4360 POP BC ;Hoehe/Breite
4370 DJNZ CLOOP ;-> naechste Zeile
4380 RET ;-> Basic
4390
5000 ;*****
5010 ;SUB Window-Grenzen von Basic
5020 ;uebernehmen
5030 ;OUT: H=Textspalte links
5040 ; L=Textzeile oben
5050 ; D=Textspalte rechts
5060 ; E=Textzeile unten
5070
5080 WINPAR ;siehe letzte Folge
5090
5140 ;*****

```

```

5150 ;SUB Window-Adresse berechnen
5160 ;IN:  H=Textspalte links
5170 ;    L=Textzeile oben
5180 ;    D=Textspalte rechts
5190 ;    E=Textzeile unten
5200 ;OUT: B=Breite in Bytes
5210 ;    C=Hoehe in Rasterzeilen
5220 ;    HL=Screenadresse oben links
5230 ;
5260 WINADR ;siehe letzte Folge
5270 ;
5650 ;*****
5660 ;SUB Bildschirmadr. eine Raster-
5670 ;zeile nach unten weiterrechnen
5680 ;IN:  HL = alte Adresse
5690 ;OUT: HL = neue Adresse
5700 ;
5710 LDOWN ;siehe letzte Folge

```

10	'***** Window-Demoprogramm *****	[1818]
20	'	[117]
30	MODE 1	[506]
40	'	[117]
50	REM Basiclader aufrufen	[869]
60	GOSUB 1000	[851]
70	'	[117]
80	REM Window-Verwaltung initialisieren	[2894]
90	MEMORY &7FFF:CALL initw,HIMEM+1	[614]
100	'	[117]
110	REM Hintergrund erzeugen	[2003]
120	FOR x=0 TO 640 STEP 10	[1838]
130	MOVE 320,0:DRAW x,400	[1527]
140	MOVE 320,399:DRAW x,0	[1532]
150	NEXT	[350]
160	'	[117]
170	REM 4 Fenster oeffnen	[1776]
180	FOR w=0 TO 3	[662]
190	PAPER 2+(w AND 1)	[908]
200	PEN w AND 1	[392]
210	links=w*8+2:rechts=w*8+15	[1619]
220	oben=w*4+3:unten=w*4+10	[2667]
230	CALL openw,links,rechts,oben,unten	[2970]
240	'	[117]
250	REM Beschriftung	[1238]
260	FOR y=oben+1 TO unten-1	[2049]
270	LOCATE links+3,y	[2110]
280	PRINT"Window";w	[1785]
290	NEXT y	[359]
300	'	[117]
310	REM inverser Balken	[2174]
320	FOR y=oben+1 TO unten-1	[2049]
330	CALL invers,links+1,rechts-1,y,y	[3020]
340	FOR i=1 TO 500:NEXT	[678]
350	CALL invers,links+1,rechts-1,y,y	[3020]

360	NEXT y	[359]
370	NEXT w	[361]
380	'	[117]
390	REM Fenster schliessen	[1350]
400	FOR w=0 TO 3	[662]
410	CALL closew	[537]
420	FOR i=1 TO 500:NEXT	[678]
430	NEXT w	[361]
440	'	[117]
450	GOTO 180	[401]
460	'	[117]
1000	'***** Basiclader fuer die *****	[1257]
1010	'***** Window - Routinen *****	[1404]
1020	'	[117]
1030	'Ermoeeglicht folgende Aufrufe:	[2338]
1040	'	[117]
1050	'CALL initw, SpeicherStart	[1525]
1060	'CALL invers, li, re, ob, un	[1253]
1070	'CALL openw, li, re, ob, un	[1478]
1080	'CALL closew	[813]
1090	'	[117]
1100	MEMORY &9FFF:RESTORE 1220	[889]
1110	FOR adr=&A000 TO &A137	[1091]
1120	READ a\$:a\$="&" + a\$	[595]
1130	v=VAL(a\$):s=s+v	[1644]
1140	POKE adr,v	[136]
1150	NEXT adr	[547]
1160	IF s<>31588 THEN PRINT "DATAFEHLER !!!	[1867]
"	:END	
1170	'	[117]
1180	invers=&A000:initw=&A003	[1991]
1190	openw=&A006:closew=&A009	[2489]
1200	RETURN	[555]
1210	'	[117]
1220	DATA C3,3F,A0,C3,56,A0,C3,64	[1507]
1230	DATA A0,C3,B1,A0,00,00,00,00	[1455]
1240	DATA 00,00,00,00,00,00,00,00	[1005]
1250	DATA 00,00,00,00,00,00,00,00	[1005]
1260	DATA 00,00,00,00,00,00,00,00	[1005]
1270	DATA 00,00,00,00,00,00,00,00	[1005]
1280	DATA 00,00,00,00,00,00,00,00	[1005]
1290	DATA 00,00,00,00,00,00,00,CD	[1027]
1300	DATA EA,A0,CD,F7,A0,C5,E5,7E	[1217]
1310	DATA 2F,77,23,10,FA,E1,CD,29	[850]
1320	DATA A1,C1,0D,20,F0,C9,AF,32	[1085]
1330	DATA 0C,A0,DD,6E,00,DD,66,01	[879]
1340	DATA 22,0D,A0,C9,CD,EA,A0,CD	[591]
1350	DATA F7,A0,E5,21,0C,A0,7E,34	[1149]
1360	DATA 5F,87,83,87,6F,26,00,11	[1306]
1370	DATA 0D,A0,19,E5,FD,E1,FD,5E	[1743]
1380	DATA 00,FD,56,01,E1,FD,75,02	[1326]
1390	DATA FD,74,03,FD,70,04,FD,71	[973]
1400	DATA 05,CD,99,BB,CD,2C,BC,C5	[1707]
1410	DATA E5,4F,7E,12,71,23,13,10	[1306]
1420	DATA F9,E1,CD,29,A1,79,C1,0D	[1409]
1430	DATA 20,ED,FD,73,06,FD,72,07	[2500]
1440	DATA C9,3A,0C,A0,B7,C8,3D,32	[1272]
1450	DATA 0C,A0,5F,87,83,87,6F,26	[1830]
1460	DATA 00,11,0D,A0,19,E5,FD,E1	[1623]
1470	DATA FD,6E,00,FD,66,01,FD,5E	[1336]
1480	DATA 02,FD,56,03,FD,4E,04,FD	[1469]
1490	DATA 46,05,C5,D5,06,00,ED,B0	[1389]
1500	DATA D1,EB,CD,29,A1,EB,C1,10	[1742]
1510	DATA F1,C9,DD,5E,00,DD,6E,02	[1462]
1520	DATA DD,56,04,DD,66,06,C9,25	[976]
1530	DATA 2D,7A,94,47,7B,95,4F,CD	[1623]
1540	DATA 11,BC,38,04,28,06,18,08	[1264]
1550	DATA CB,24,CB,20,CB,24,CB,20	[1035]
1560	DATA CB,21,CB,21,CB,21,7C,26	[1258]
1570	DATA 00,54,5D,29,29,19,29,29	[2061]
1580	DATA 29,29,5F,19,11,00,C0,19	[800]
1590	DATA C9,7C,C6,08,67,D0,D6,40	[1629]
1600	DATA 67,7D,C6,50,6F,D0,24,C9	[1070]