

Sector Mapper#

Table of Contents

- [Sector Mapper](#)
- [Dokumentation zu MAPPER.SRC](#)
- [BAD SECTORS](#)

Dokumentation zu MAPPER.SRC#

Mit MAPPER.SRC koennen Sie einen Blick hinter die Kulissen von professionellen Programmen und deren Kopierschutz werfen.

MAPPER liest jeden Sektor einer Single-Density Diskette ein, und gibt Ihnen durch eine Sektorkarte einen Uebersicht der Diskette.

Folgende Symbole werden verwendet:

*	belegter Sektor
	(Leerzeichen) leerer Sektor
M	Missing Sector (s. Text)
C	CRC-Error (s. Text)
B	Sonstiger 'Bad-Sector'

Sie erhalten Informationen, ob ein Sektor leer oder belegt ist, oderob es sich um einen sog. 'BAD SECTOR' handelt. Die letzteren geben beim Lesen einen ERROR 144, der gerne fuer den Kopierschutz hergenommen wird.

BAD SECTORS#

Bei einem BAD SECTOR wird Ihnen weiterhin mitgeteilt, um welche Art von BAD SECTOR es sich dabei handelt. Moeglich sind sog. 'Missing Sectors', fehlende Sektoren, die z.B. entstehen, wenn ein Sektor mit zu hoher Drehzahl geschrieben wird, und daher den naechsten Sektor ueberschreibt.

Zweite Moeglichkeit sind sog. 'CRC-Errors', die auf einen Fehler in der internen Pruefsumme hinweisen.

Falls ein 'B' angezeigt wird, so handelt es sich ebenfalls um einen BAD SECTOR, der jedoch in die obigen Kategorien nicht einordnen laesst (z.B. Data Flag).

Das Einlesen einer Diskette kann bei vielen BAD SECTORS laenger dauern, da das Laufwerk bei Lesefehlern mehrmals versucht, den Sektor doch noch zu lesen

```
*****
*   SECTOR-MAPPER fuer Single-Density
*
*   P. Finzel 1985
*****
```

```
MAXSEC   EQU 721   Letzter Sector + 1
MINPOS   EQU 2     Anfangs-Spalte
MAXPOS   EQU 38    End-Spalte +1
DRIVE    EQU 1     Aktuelles Laufwerk
EOL      EQU $9B   End-of-Line Zeichen
```

```

ZEIGER EQU $80 Zeropage-Register
SBUFFER EQU $600 Page 6 als Buffer

LMARGIN EQU $52 linker Rand
COLCRS EQU $55 Cursor-Spalte
ROWCRS EQU $54 Cursor-Zeile
DVSTAT EQU $2EA Geraete-Status
CONSOL EQU $D01F Funtionstasten

KEYBDV EQU $E420 Keyboard-Handler
EDITDV EQU $E400 Editor-Handler
DSKINV EQU $E453 Disk-Handler-Einsprung
*
* DBC-Kontrollblock
*
DDEVIC EQU $300
DUNIT EQU $301
DCOMND EQU $302
DSTATS EQU $303
DBUFLO EQU $304
DBUFHI EQU $305
DBYTLO EQU $308
DBYTHI EQU $309
DAUX1 EQU $30A
DAUX2 EQU $30B
*
ORG $A800 im res. Bereich
*
SMAPPER LDA #0 Linker Rand
STA LMARGIN auf Null
LDA #$7D Clear Screen
JSR SCROUT
LDA #$80 inverse Ausgabe
STA INVREG
JSR PRINT
ASC \\Sector-Mapper P.Finzel 1985\\
LDA #EOL
JSR SCROUT
LDA #EOL
JSR SCROUT
JSR PRINT
ASC \\TR012345678901234567012345678901234567\\
LDA #EOL
JSR SCROUT
JSR PRINT
ASC "00\^[02\^[04\^[06\^[08\^[10\^[12\^[14\^[ "
ASC "16\^[18\^[20\^[22\^[24\^[26\^[28\^[30\^[ "
ASC \\32\^[34\^[36\^[38\\
*
* auf START-Taste warten
*
JSR START
LDX #7
LDY #23
JSR POSITION
JSR PRINT
ASC \\ Diskette wird gelesen \\
LDA #0

```

```

        STA INVREG
        LDY #1
        STY SECNUM
        DEY
        STY SECNUM+1
        LDA #2
        STA SPALTE
        LDA #3
        STA ZEILE
CHKSEC  JSR READSEC
        BMI ERRSEC          Fehler-->
*
*  Auswertung der gelesenen Daten
*
        LDX #$7F           128 Bytes pro Sector
        LDA #0
SECLEER ORA SBUFFER,X
        DEX
        BPL SECLEER
        CMP #0
        BEQ LEER
        LDA #'*           Merker 'Voller Sektor'
        JMP NXTSEC
LEER    LDA #$20           Leerzeichen
        JMP NXTSEC
*
*  DISK-FEHLER: Hardware-Status des
*                Disk-Contollers abfragen
*
ERRSEC  JSR STATUS         Disk-Status
        LDA DVSTAT+1       Status des Disk-Cont.
        AND #8             CRC-Error?
        BNE NOCRC         nein-->

        LDA #'C           Merker fuer CRC
        JMP NXTSEC

NOCRC   LDA DVSTAT+1
        AND #16           'Missing Sector'
        BNE NOMISS       nein-->

        LDA #'M
        JMP NXTSEC

NOMISS  LDA #'B           sonst. 'Bad Sector'
*
*  den ermittelten Merker ausdrucken
*
NXTSEC  PHA               Merker aufheben
        LDX SPALTE
        LDY ZEILE
        JSR POSITION
        PLA
        JSR SCROUT
        INC SECNUM         naechsten Sektor
        BNE NXTS1
        INC SECNUM+1
NXTS1   INC SPALTE
        LDA SPALTE

```

```

        CMP #MAXPOS          naechste Zeile?
        BNE NXTS2
        INC ZEILE
        LDA #2
        STA SPALTE
NXTS2   LDA SECNUM
        CMP #MAXSEC:L
        BNE NXTS3
        LDA SECNUM+1
        CMP #MAXSEC:H
        BEQ ENDE fertig-->
NXTS3   JMP CHKSEC
*
* auf START-Taste warten
*
ENDE     JSR START
        JMP SMAPPER          Neu starten
        RTS

*****
* Bereit-Meldung, START-Taste
*****
START    LDX #0
        LDY #23
        JSR POSITION
        LDA #$80
        STA INVREG
        JSR PRINT
        ASC \\              Bitte START druecken      \\
        LDA #8
        STA CONSOL
*
* Auf START-Taste warten
*
WARTE    LDA CONSOL
        AND #1
        BNE WARTE
GEDRKT   LDA CONSOL          immer noch
        AND #1              gedrueckt?
        BEQ GEDRKT          ja -->
        RTS

*****
* Sektor in SBUFFER einlesen
*****
READSEC  LDA #DRIVE          Drive 1
        STA DUNIT
        LDA #'R              Sektor lesen
        STA DCOMND
        LDA #$40             Status fuer
        STA DSTATS          lesen
        LDA #SBUFFER:L
        STA DBUFLO
        LDA #SBUFFER:H
        STA DBUFHI
        LDA #$80             128 Bytes
        STA DBYTLO
        LDA #0

```

```
STA DBYTHI
LDA SECNUM      Nummer des
STA DAUX1      Sektors in
LDA SECNUM+1    AUX-Bytes
STA DAUX2
JSR DSKINV
RTS
```

```
*****
* Status von Disk-Drive anfordern
*****
```

```
STATUS  LDA #DRIVE
        STA DUNIT
        LDA #'S
        STA DCOMND
        JSR DSKINV
        RTS
```

```
*****
* Interne Variable
*****
```

```
SECNUM  DFW 0    aktuelle Sektornummen
ZEILE   DFB 0    momentane Zeile
SPALTE  DFB 0    momentane Spalte
INVREG  DFB 0    fuer PRINT: Invers
```

```
*****
*          Das PRINT-Unterprogramm
*
* Modifiziert zum Ausdruck von
* inversen Zeichen.
* Hierzu muss zuvor INVREG auf $80
* gesetzt werden.
*****
```

```
PRINT   PLA
        STA ZEIGER
        PLA
        STA ZEIGER+1
        LDX #0
PRINT1  INC ZEIGER
        BNE *+4
        INC ZEIGER+1
        LDA (ZEIGER,X)
        AND #$7F
        ORA INVREG
        JSR SCROUT
        LDX #0
        LDA (ZEIGER,X)
        BPL PRINT1
        LDA ZEIGER+1
        PHA
        LDA ZEIGER
        PHA
        RTS
```

```
*****
```

* Ein Zeichen ausgeben *

```
*****
SCROUT  TAY          simuliert
        LDA EDITDV+7 einen indirekten
        PHA          Sprung nach
        LDA EDITDV+6 der SCROUT-Routine
        PHA          im OS-ROM
        TYA
        RTS
```

* POSITION-ROUTINE *

```
*****
POSITION STX COLCRS
        STY ROWCRS
        LDX #0
        STX COLCRS+1
        RTS
```