

Das Assemblieren dieses Codes geht mit MAKMULTI .BAT :

```
MAKMULTI.BAT
REM GENERIERUNG DES BOOT-PROGRAMMS
REM =====
A51 MULTI.A51
L51 MULTI.OBJ TO MULTI.ABS
OHS51 MULTI.ABS
```

Wir haben jetzt drei HEX-Dateien: FSD51.HEX, MON51.HEX und MULTI.HEX, die zu kombinieren sind, und die ein gemeinsames EPROM ergeben sollen. Die Verbindung der drei Dateien kann etwa mit COPY erfolgen.

Was stört ist, daß jede Datei eine Endezeile (:00000001FF) enthält und beim Verbinden mit COPY diese Zeile natürlich bestehen bleibt und den späteren Programmiervorgang vorzeitig abbrechen würde.

Man kann durchaus mit Mitteln des Betriebssystems eine Lösung herbeiführen: Man vereinigt die Dateien mit COPY und sortiert die Zeilen mit SORT nach den Werten im Adreßfeld (Zeichenpositionen 4, 5, 6 und 7 wenn die Zählung bei 1 beginnt). Man verwendet bei SORT die Option /+n, die den Sortiervorgang ab der n-ten Spalte vornimmt. Dieser Vorgang steht in der BAT-Datei KOMBI.BAT:

```
KOMBI.BAT
COPY MON51.HEX+FSD51V21.HEX+MULTI.HEX UPR51.UNS
SORT /+4 < UPR51.UNS > UPR51.HEX
```

Eine Kleinigkeit, die man noch händisch ausführen muß: Die drei Schlußzeilen kommen jetzt aufgrund des Sortiervorgangs hintereinander zu liegen. Da ihr Adreßfeld 0000 enthält, stehen sie in der kombinierten HEX-Datei am Anfang und nicht am Ende. Man verschiebt also mit einem Texteditor die Zeilen ans Ende der Datei und löscht zwei davon.

Jetzt muß man mit dem Kommando

```
HEX2BIN UPR51
```

eine Binärdatei zum Erstellen eines EPROM generieren.

Fertig!

Selbstverständlich kann man jetzt die einzelnen Schritte in einer einzigen BAT-Datei UPR51.BAT zusammenfassen.

```
UPR51.BAT
CALL TGMPAT
CALL MAKKEIL
CALL MAKMULTI
CALL KOMBI
```

Testen des MULTI-BOOT-Eproms

Trotz der Kürze des Multiboot-Programms sind doch einige Möglichkeiten auf ihre richtige Funktion zu prüfen. Beschränkt man sich auf das Brennen eines EPROM, steht man bei Fehlern oft ziemlich ratlos da, sodaß man diese Versuche alsbald unterläßt.

Der Simulator/Debugger DS51 ist dabei eine vorzügliche Hilfe. Aber auch mit dem Debugger hat man bald genug, denn es sind gar nicht so wenige Eingaben zu machen, bis eine Einstellung korrekt vorgenommen ist. Man behilft sich daher mit einem eigenen INI-File, mit dem der Testlauf gut dokumentiert und gut reproduzierbar wird.

MULTI.INI

Einstellungen des DS51

Die Schalterstellung an P1.0 muß simuliert werden, es wurde nur der Multiboot-Programmteil geladen, daher fehlt der Reset-Vektor des späteren EPROM. Dieser wird mit der asm-Zeile nachgetragen. Die drei Breakpoints bei 1000, A5 und A000 entsprechen den drei Sprungzielen des Boot-Vorgangs. Das gesamte Programm wird jeweils durch g gestartet und nach Auflaufen auf einen der drei Breakpoints wird geprüft, ob die Sprungziele eingehalten werden und ob die Speicherinhalte korrekt sind. Die KEYWAIT-Aufrufe unterbrechen den Ablauf, damit man die Zwischenergebnisse verfolgen kann. In der Anmerkung nach KEYWAIT steht jeweils was getestet wird und was an welchen Speicheradressen stehen soll.

```
/* MULTI.INI */
\r                               /* Register off                */
\u\u\u                           /* Increase Window          */
\o\m                               /* Options Medium (Lines)  */
\v\s                               /* View Serial (Off)       */
load ..\..\ds51\8051.iof          /* IOF driver for 8051     */
map 0,0xffff                       /* XDATA memory 64KByte   */
P1.0=0 /* KEIL-MONITOR */
asm 0
ljmp 0a00H
.
asm 0x1000
nop
.
bs 0x1000
bs 0x00A5
bs 0xA000
KEYWAIT("")
/* Kaltstart ***** */
/* $=1000H [A023]=LJMP 1003 [A0FB]=BB, [A0FE]=C5 */
KEYWAIT("")
g 0
t
d x:0xA000,0xA00F
d x:0xA020,0xA02F
d x:0xA0F0,0xA0FF
KEYWAIT("")
/* Warmstart vom Monitor ***** */
/* $=1000H [A023]=LJMP 1003 [A0FB]=BB [A0FE]=C5 */
KEYWAIT("")
P1.0=1
g 0
t
d x:0xA000,0xA00F
d x:0xA020,0xA02F
d x:0xA0F0,0xA0FF
KEYWAIT("")
/* Warmstart vom Userprogramm ***** */
/* $=A000, [A000]=LJMP A100 [A023]=LJMP 1003 [A0FB]=BB [A0FD]=DD [A0FE]=C5" */
KEYWAIT("")
P1.0=0
g 0
d x:0xA000,0xA00F
d x:0xA020,0xA02F
d x:0xA0F0,0xA0FF
KEYWAIT("")
/* Warmstart vom Userprogramm */
/* Speicher initialisieren ***** */
/* $=A000, [A000]=LJMP A100 [A023]=00 [A0FB]=BB [A0FD]=DD [A0FE]=C5" */
KEYWAIT("")
EC d:0x10
0x12
0x23
0x34
.
EC x:0xA010
0x12
0x23
0x34
.
P1.0=0
d d:0x10,0x1F
d x:0xA000,0xA00F
d x:0xA010,0xA01F
KEYWAIT("")
EC X:0xA0FC
0xCC
.
g 0
d d:0x10,0x1F
d x:0xA000,0xA00F
d x:0xA010,0xA01F
d x:0xA020,0xA02F
d x:0xA0F0,0xA0FF
EC d:0x10
0x12
0x23
0x34
.
EC x:0xA010
0x12
0x23
0x34
KEYWAIT("")
/* Warmstart vom Monitor */
/* Speicher initialisieren ***** */
/* $=A000, [A000]=LJMP A100 [A023]=LJMP 1003 [A0FB]=BB [A0FD]=DD [A0FE]=C5" */
d d:0x10,0x1F
d x:0xA000,0xA00F
d x:0xA010,0xA01F
d x:0xA020,0xA02F
KEYWAIT("")
EC X:0xA0FC
0xCC
.
P1.0=1
g 0
d d:0x10,0x1F
d x:0xA000,0xA00F
d x:0xA010,0xA01F
d x:0xA020,0xA02F
d x:0xA0F0,0xA0FF
```

