Sprungverteiler

Der Sprungverteiler ist kein Programm im eigentliche Sinne. Er wird installiert, um Standard-Routinen (z.B. Headersave, Druckertreiber, aber auch Tastatur- und Spielhebelabfragen) von Anwenderprogrammen (z.B.KC-BASIC+, Microword u. a.) zu erreichen, gleich, wo die jeweilige Routine im Speicher steht. Der Sprungverteiler ist im Speicherbereich FFB7H bis FFFFH einzutragen. In FFB7H steht nur ein Return (C9).

Voraussetzung für die Nutzung des Sprungverteilers ist natürlich ein bestückter Speicher (RAM oder ROM) im o. g. Bereich.

Ein Anwendungsbeispiel soll die Einbindung des Sprungverteilers illustrieren: Es soll mit dem Microword ein Text von der Kassette eingelesen werde. Mit „R„ ruft man LORUF des Headersave über den Sprungverteiler auf (Adr.: FFF1H): Im „Brosig“-Monitor steht hier ein JMP FAACH (C3 AC FA). Kommt das Headersave auf E000 bis E3FFH zum Einsatz, ist auf FFF1H ein JMP E000H (C3 00 E0) einzutragen. Microword findet LORUF in beiden Fällen ohne Änderung.

Wer den Sprungverteiler nutzen will, muß sich diesen in den RAM eintragen oder beim 4-KByte-Monitor in den EPROM brennen. Nicht genutzte Routinen erhalten ein JMP FFB7H (C3 B7 FF), also einen Rücksprung in das aufrufende Programm. Mit diesen Hinweisen sollte es möglich sein, den in [1] ausführlich erläuterten Sprungverteiler praktisch zu nutzen. (K.-J. Hofmann)

DIREKTANSPRÜNGE: Wer keine Möglichkeit hat, sich diesen SV einzutragen (z.B. 16-K), kann sich auch die Aufruf-Adressen der Standard-Routinen (z.B. LORUF/SARUF des HEADERSAVE) direkt eintragen. In allen Beschreibungen stehen die jeweiligen Adressen, wo diese Sprünge eingetragen werden. Hierbei bleiben ein eingetragenes CDH (CALL) oder C3H (JMP) stehen, es kommt der niederwertige, dann der höherwertige Teil der Rufadressen. Die Rufadressen stehen ebenfalls immer in den Beschreibungen zu diesen Standard-Programmen. Es ist deshalb immer sinnvoll, sich alle Beschreibungen gründlich durch zu lesen.

s.a. [1] Lühne, V.: Sprungverteiler für Z 1013-Betriebssystemerweiterungen, FUNKAMATEUR 37 (1988), H.10, S.484

Beschreibung

****

```
            Sprungverteiler für Z 1013-Erweiterungen
```

****

Der Zusatzsprungverteiler wurde entsprechend einer Einigung innerhalb der Interessengemeinschaft Heimcomputer des IZ an der TU Dresden folgendermaßen festgelegt:

Beginnt auf Adresse

FFFFH

abwärtsführend:

FFFDH - JMP INKEY

```
holt ein Zeichen von Tastatur in den Akku;
kommt beim 2. Aufruf nur zurück, wenn Taste
zwischendurch losgelassen wurde
```

FFFAH - JMP POLL

```
bringt immer ein Zeichen im Akku zurück, egal
ob Taste losgelassen wurde oder nicht
```

FFF7H - JMP STAT

```
übergibt Tastaturstatus im Akku
 A=0   - keine Taste gedrückt
 A=FFH - Taste gedrückt
die Abfrage erfolgt ohne Rücksicht, ob die
Taste schon vor dem Aufruf gedrückt war und
hinterlässt trotz gedrückter Taste den Status
'letztes Zeichen war 0' ((Zelle 4)=0) um eine
evtl. nachfolgenden INKEY-Routine nicht zu
sperren
```

FFF4H - JMP SARUF

```
ruft die SAVE-Routine des Headersave
! zerstört 1. Registersatz + AF'
Parameterübergabe:
Zellen 1BH - anfadr.
       1DH - endadr.
       23H - strtadr.
Akku   3AH - Wiederholen der SAVE-Funktion mit
             gleichem Kopf
H(IY)  Typvorgabe (in ASCII), sonst 0
```

FFF1H - JMP LORUF

```
ruft LOAD-Routine des Headersave
! zerstört 1. Registersatz + AF'
Parameterübergabe:
Zellen 1BH - neue Anfangsadresse des Files
             sonst 0
Akku   0   - ohne signifikante Kontrolle
       4EH - mit signifikanter Kopfkontrolle
             (Typ) + Namenabfrage
H(IY)  0   - Typ wird abgefragt
       Typkennzeichen (in ASCII), keine Typabfr.
L(IY)  20H - Freigabe Autostart bei COM-Files
```

FFEEH - JMP ZMINI

```
Initialisierung der Z-Monitorrufe auf B0H
```

FFEBH - JMP DRDEL

```
setzt den logischen Druckertreiber zurück
```

FFE8H - JMP DRAKK

```
übergibt den Akkuinhalt an den logischen
Druckertreiber
```

FFE5H - JMP BSDR

```
druckt den Inhalt des BWS und kehrt in das
rufende Programm zurück
```

FFE2H - JMP HARDC

```
übergibt den Akkuinhalt an logischen Drucker-
treiber wenn ein Flag im Rechner gesetzt ist;
wandelt CR (0DH) in NL (1EH/0DH-0AH)
! nur verwenden, wenn Programm eine eigene
Bildschirmverwaltung hat
```

FFDFH - JMP DRZEL

```
wie DRAKK, nur das der Inhalt von 1BH übergeben
wird (vorgesehen, um im BASIC mit POKE zu drucken)
```

FFDCH - JMP BEEP

```
erzeugen eines kurzen Signals
```

FFD9H - JMP ASTA

```
Ausgabe Akkuinhalt als ASCII-Zeichen an PUNCH
```

FFD6H - JMP BSTA

```
Ausgabe Akkuinhalt als Byte an PUNCH
```

FFD3H - JMP AIN

```
Eingabe eines ASCII-Zeichens vom LBL in den Akku
```

FFD0H - JMP BIN

```
  Eingabe eines Bytes vom LBL in den Akku
```

FFCDH - JMP DRINI

```
  Initialisierung des logischen Druckertreibers
```

FFCAH - JMP ZEIDR

```
  übergibt ein Zeichen im Akku an physischen
  Druckertreiber
```

FFC7H - JMP BLMK

```
  Lesen eines Blocks vom Headersave
  Parameterübergabe:
   Zellen 25H/26H * Kopfinhalt des zu lesenden Bl.
   HL             * Ladeadresse des Blocks
  Return:
   Zellen 25H/26H * Kopfinhalt + 20H
   HL             * HL:=HL+20H
  Abbruch des Lesens bei Kopfinhalt=0FFFFH
  oder DMA > Endadr in ARG2 (1DH)
```

FFC4H - JMP BSMK

```
  Schreiben eines Blocks im Headersave
  Parameterübergabe:
   HL  * Quelladresse Block
   IX  * Kopfinhalt
   DE  * Anzahl der Sync.-Bits
  Return:
   HL  * HL:=HL+20H
```

FFC1H - JMP SUCHK

```
  Suchen eines Kopfblocks und Übergabe des Inhalts
  im Kopfpuffer (E0-FF), keine Auswertung
```

FFBEH - JMP AKP

```
  Aufbereitung Kopfpuffer mit Namenabfrage
  Parameterübergabe wie bei SARUF
```

FFBBH - JMP GETST

```
  Abfrage der Joysticks und Übergabe des Er-
  gebnisses in BC (B-links,C-rechts) mit folgen-
  den Bit-Bedeutungen (Belegung mit 1):
    Bit 0 - links
```

```
       1 - rechts
       2 - runter
       3 - hoch
       4 - Aktionstaste
    Z-Flag=1, wenn keine Betätigung vorliegt
   CY-Flag=1, wenn Spielhebel nicht angeschlossen
```

FFB8H - JMP SOUND

```
 Ausgabe einer vollen Periode auf die Tonband-
 buchse, sowie auf Bit 7 vom Systemport (User-P)
 Übergabe der Periodendauer in C mit
 T=n*33mks+20mks (2MHz)
```

Anmerkungen

DRZEL - eigentlich, um aus BASIC heraus zu drucken; wird m.E. nicht genutzt

Innerhalb des BASIC-Interpreters KC-BASIC+ kann man mit dem Unterprogramm

```
    xxxx POKE 27,ASC(A$):CALL *FFDF:RETURN
                                  xxxx = Zeilennummer
                                    A$ = Stringvariable (ein Zeichen lang)
```

einzelne Zeichen an einen eingebundenen Druckertreiber übermitteln

BSDR - Bildschirmkopie drucken. im BROSIG-Monitor ist bereits eine Routine enthalten, die allerdings den Bildschirm nur bis zum Cursor bzw. dem Graphikzeichen 0FFH druckt und außerdem die Verarbeitung von Graphikzeichen dem Drucker überläßt.

HARDC - soll A drucken, wenn Flag im Speicher gesetzt ist. welche Speicheradresse gemeint ist, wird leider nicht spezifiert. Ich denke, daß es 006EH, Bit 0 sein könnte. Das ist das Druckflag für den Original-10k-BASIC-Interpreter von Riesa, allerdings müßte dann das Zeichen im Register E übergeben werden, nicht in A !?

In practic 1/90, S. 41 wurde eine Erweiterung des Sprungverteilers für BASICODE-Save und -Load vorgeschlagen. Das hat aber keinerlei praktische Anwendung mehr erfahren. Es gab auch keine Hinweise zu Aufrufparametern und auch keine Referenzimplementierung.

FFB5H - JMP BSAVE:

```
 Saven im BASIC-Code-Format
```

FFB2H - JMP BLOAD:

```
 Laden im BASIC-Code-Format
```

(vp)

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/z1013/software/sprungverteiler?rev=1732544548

Last update: **2024/11/25 14:22**