

Funktionstastenbelegung unter AC1-BASIC5

Für den AC1/ LLC2 stand ein Tool des SCCH „F1-F8-KEY“ bzw. „Funktionstasten“ zur Verfügung. Dabei handelte es sich um einen "trickreichen" Einzeiler mit angehängtem MC-Teil.

Basic-Teil:

```
1 CALL*6138:KEY:NEW:REM##1612 (C)1988 E.Ludwig HALLE,Tel.41082#
```

- Mittels der Steuerzeichen hinter REM wurde das "Listing" gar nicht angezeigt, sondern nur der Copyright-Vermerk.
- Mit "CALL" erfolgt Aufruf der angehängten MC-Routine, die die im Anschluss abgelegten Werte nach 1F80h kopiert,
- dann Anzeige dieser neuen Tasten (KEY) und
- Löschen des Programms (NEW).

Nach der 3. Null des Basicprogrammes (also hinter dem Programm) steht der eigentliche Inhalt: die Umkopiererroutine:

```
6138h E5      PUSH HL
6139h D5      PUSH DE
613Ah C5      PUSH BC
613Bh 21 4A 61 LD HL,614Ah ;Ablageadresse in F1-F8-Key
613Eh 11 80 1F LD DE,1F80h ;Zieladresse im RAM
6141h 01 80 00 LD BC,0080h ;Menge max. 80h Bytes
6144h ED B0    LDIR          ;umkopieren
6146h C1      POP BC
6147h D1      POP DE
6148h E1      POP HL
6149h C9      RET
```

Anschließend folgen die Binärwerte der Tastenfolgen

```
614AH DEFB      0                      ;EINLEITENDES 0-BYTE
        DEFM      'LIST:',0DH,0        ;TASTENFOLGE KEY1
        DEFM      'RUN:',0DH,0         ;KEY2
        DEFM      'CLOAD""',0DH,0      ;KEY3
        DEFM      'CSAVE""',0          ;KEY4
        DEFM      'AUTO',0DH           ;KEY5
        DEFM      'INKEYA$:IFA$=""THEN',0 ;KEY6
        DEFM      'A=FRE(0):?A:;IFA<0THENA=A+65536:?""=";A',0DH ;KEY7
        DEFM      'NEW:',0DH           ;KEY8
        DEFB      0                      ;ABSCHLUSS 0-BYTE
```

Beachten:

- Ein 0dh nach den Tastenbytes führt Kommando sofort aus, ansonsten wird auf weitere Eingaben (Dateiname) gewartet
- hinter jeder Tastenfolge muss eine 0 stehen
- Gesamtlänge der Einträge (einleitende 0 bis Schluss-0) maximal 128 Bytes
- Wird die Länge nicht ausgeschöpft, ist der Rest mit 00 aufzufüllen.

Will man mit "F1-F8-Key" eine andere Tastenbelegung erzielen, so muss man diese erst in eine geladene (aber nicht gestartete) "Vorlage" eintragen... Der Nutzer muss sich also sein eigenes "F1-F8-Key" Programm bauen.

Inwiefern es dazu damals Anleitungen oder gar ein weiteres (komfortables) Tool gab (was das „F1-F8-KEY“ erzeugte) entzieht sich meiner Kenntnis.

Problem:

Der nach der 3. Null angehängte MC-Teil wird normalerweise nicht mit gesichert (deswegen lässt sich „F1-F8-Key“ auch nicht mit BASCONV5 umwandeln!)

Möglichkeit 1:

- „Vorlage“ laden, AC1-BASIC5 verlassen
- im Monitor die gewünschten Bytefolgen eintragen
- Warmstart
- **DOKE &60A8,&63CF** (Speicherende hinausschieben)
- SAVE"xxx" (Speichern mit „Anhang“)

Möglichkeit 2:

- am PC die "Vorlage" mittels Hex-Editor wie gewünscht patchen
- dann als BASIC-Programm per Stick übertragen.

Möglichkeit 3:

- FKEY nicht als BASIC-Programm, sondern als blanke BIN-Datei, d.h. nur die Tastenfolgen selbst (128 Bytes)
- am PC anfertigen
- mit **BLOAD"xxx",&1F80** laden
- Vorteil: kann zu Beginn jedes BASIC-Programmes benutzt werden, um im Programm selbst die F-Tasten für bestimmte Zwecke zu benutzen

Eine Vorlage "FKEY.ABC" für AC1-BASIC5 anbei, belegt mit folgenden Werten:

```
LIST
RUN
LOAD"
SAVE"
DIR
WINDOW:COLOR:CLS
```

"FKEY.BIN" analog für BLOAD (nur F-Tasten-Einträge)

Anmerkung:

Die Anzeige von Steuerzeichen in einer REM-Zeile habe ich in AC1-BASIC5 nicht realisiert. Es erscheint also mit LIST das „1-Zeiler-Listing“.