

# Tastatur

Um die wenig brauchbare Flachfolientastatur des Z1013 zu ersetzen, gibt es in der [Literatur](#) viele Lösungsvorschläge. Durchgesetzt hat sich die Komfortastatur nach Brosig.



Originale Flachfolientastatur des Z1013. Die Anordnung der Tasten in alphabetischer Reihenfolge, die 5fach-Belegung mit Shift1..Shift4 und der nötige hohe Druck auf die Tasten, um einen Kontakt herzustellen, machen das Arbeiten mit dieser Tastatur zur Qual.

## Brosig-Tastatur nach mp

erstmals vorgestellt bei der [1. Z1013-Tagung](#) in Dresden Dez. 1987.

Mikroprozessortechnik 7/1988, Rainer Brosig: „Z1013-Tastatur mit Raffinessen“



Die Leiterplatte wurde so entworfen, dass sie auf einer K7659-Tastatur rechts zwischen Enter und Sondertasten passt.

Beim abgebildeten Exemplar ist links in der Tastatur ein Lautsprecher eingebaut; ganz rechts sind zwei Zusatztasten für Reset und NMI incl. Entprellung.

## Beschreibung

ANMERKUNG: Die hierfolgende Beschreibung ist der sogenannte BROSIG-MONITOR. Diesen gibt es abhängig von zwei Tastaturen, der hier beschriebenen K 7859 sowie der K 7652. Letztere hat die gleiche Matrix, nur dass bei ihr die Koordinaten der Anschlusskabel B sind, wenn sie hier mit A

gekennzeichnet sind und umgekehrt. Aber B10 und B12 sind bei ihr nicht belegt.

(Siehe auch „Mikroprozessortechnik“ Heft 7/88 S.215-218)

## Vorbetrachtung

In zahlloser Folge sind in letzter Zeit Tastaturvarianten fuer den Mikrorechner Z1013 in der Literatur vorgestellt worden. Dies ist eine logische Folge, denn die originale Flachfolientastatur steht ausserhalb jeder Diskussion. Man kann die vorgestellten Loesungen grob in vier Gruppen teilen:

1. Verbesserung der Bedienbarkeit durch bessere Schaltelemente
2. Abfragen einer groesseren Matrix (z.B.K7659→8\*12) und Umsetzen in die originale 4\*8 Matrix (z.B. Prozessorversion v. Riesa)
3. Verzicht auf eine Matrixschnittstelle und Ersatz durch eine parallele oder serielle ASCII-Schnittstelle
4. Aenderung des Monitor-Tastaturprogramms, um eine komfortablere Abfrage mit einer veraenderten Matrix zu realisieren

Alle vier Gruppen haben ihre speziellen Vor- und Nachteile. Es kann festgestellt werden, dass es keine ideale Loesung fuer alle Nutzer gibt, sondern, dass alle vier Gruppen ihren speziellen Nutzerkreis haben.

Die erste Gruppe von Tastaturen wird z.B. besonders die Nutzer interessieren, die noch wenig mit der Computermaterie vertraut sind, da nur allgemeine elektronische Kenntnisse notwendig sind. Fuer fortgeschrittene Nutzer soll folgender Tastaturvorschlag gedacht sein, der sich in die vierte Gruppe einordnet, fuer die es bisher wenige Veroeffentlichungen gab.

## Loesungsprinzip der Tastaturvariante

Es wird eine K 7659\*) mit einer speziellen Hardwareanpassung verwendet, die an den gleichen Hardwareschnittstellen wie die OriginalTastatur betrieben wird. Dadurch gibt es wenig Probleme bei Software, die direkt auf die Originalmatrix zugreift. Die Tastaturabfrage erfolgt durch ein neues komfortables Programm, welches verstaendlicherweise laenger als das originale ist und deshalb ausserhalb des 2K-Monitors generiert wird (ab 0F800H). Um Softwarekompatibilitaet zu sichern, wurde am Anfang der originalen Tastaturroutine ein Sprung zum externen Tastaturprogramm eingetragen.

Um diesen Sprung einzutragen, ist es erforderlich, den originalen Monitor zu aendern. Auf den durch den Wegfall der Originalroutine frei gewordenen Speicherplatz im Monitor wurden kleine Hilfsroutinen, wie die Ausgabe auf einen Port (O) und die Anzeige des Kommandoverteilers auf 0B0H (Z), gelegt. Weiterhin befindet sich dort eine NMI-Unterbrechungsroutine, wo, mit Hilfe einer NMI-Taste, ein Maschinenprogramm mit Registeranzeige abgebrochen werden kann, was sich bisher als eine ausserordentliche Hilfe bei der Testung von Assemblerprogrammen erwiesen hat.

Zur Aenderung des Monitor-EPROMs muss nicht unbedingt ein Eingriff auf der orig. Leiterkarte erfolgen, was evtl. Garantieansprueche loeschen koennte. Es besteht die Moeglichkeit, den Z1013 im Originalzustand zu belassen und den geaenderten Monitor-EPROM von aussen zuzuschalten (MEMDI-Steuerung). Einfacher ist es aber, den originalen Monitor-EPROM auszuloeten und durch einen

anderen EPROM mit Fassung zu ersetzen.

Derzeitig wird fuer Monitor und Tastatur ein 2732 benutzt. Auf diesem EPROM befinden sich weiterhin das Headersave sowie ein Sprungverteiler fuer Peripherie-Schnittstellen.

## Hardware

Die zusaetzliche Hardware beschraenkt sich auf einen 1-aus-16-Decoder, die Statussteuerung und einen 8-auf-4-Coder. Zur Anzeige von Shift-lock und Hardcopy werden zwei Anzeige-Flip-Flops verwendet. Damit 12 Spalten abgefragt werden koennen, muss die 4-bit Spalteninformation neu decodiert werden, da der originale Spaltendecoder nur bis 10 decodiert. Dazu ist es erforderlich, dass die 4-bit-Spalteninformation vom Spaltenlatch A47 ueber die vier freien Kontakte auf dem Loetkamm des Tastaturanschlusses gefuehrt wird. Sollten auch diese vier Draehte zu viel Aenderung auf der Platine darstellen, so kann der Spaltenlatch auch extern ueber den Systemstecker neu ausdecodiert werden.

Die verbleibenden vier Digits vom 1-aus-16-Decoder werden fuer die Ansteuerung der Flip-Flops, der Statussteuerung und des Tastenbeeps (generatorloser Piezoschwinger o.ae.) genutzt.

Die Ausgaenge des Decoders steuern die Spaltenleitungen der Tastaturmatrix, die ueber Dioden entkoppelt sind. Diese Entkopplung ist notwendig, um bei der Betaetigung von zwei Tasten (SHIFT, CTRL) keine Spalten kurzzuschliessen. Die Zeilenausgaenge werden durch einen Coder auf einen negierten binaeren Code gewandelt, um wieder die originalen 4-bit-Zeileneingangsleitungen zu erhalten. Deshalb sind prinzipiell Direktzugriffe auf die Matrix moeglich, die allerdings nicht die gleiche Tastenbelegung wie die Original-Tastatur aufweist (Korrespondenztabelle benutzen). An dieser Stelle soll auch darauf hingewiesen werden, dass solche unnoetigen Praktiken, direkt auf die Matrix zuzugreifen, immer zu Softwareinkompatibilitaeten fuehren und deshalb zu vermeiden sind. Das verwendete Abfrageprinzip weist eine Besonderheit auf, die in einer Statussteuerung liegt.

Diese Statussteuerung hat die Aufgabe, alle Spalten der Tastatur aktiv zu schalten, um einerseits eine schnelle Tastaturabfrage zu realisieren (der Rechner muss nicht staendig auf den Spaltenleitungen herumklappern) und andererseits die Interruptfaehigkeit zu ermoeglichen (wird momentan nicht ausgenutzt). Die Abfrage der Tastatur bei unbetaetigten Tasten wird durch die Statussteuerung etwa 100 mal schneller als die originale. Dieser Zeitvorteil hat aber nicht nur positive Seiten. Programme, die die Abarbeitungszeit fuer bestimmte Zwecke ausnutzen, laufen dadurch zu schnell (z.B. blinkender Cursor). Deshalb kann per Tastendruck fuer solche Anwendungen eine zusaetzliche Zeitschleife eingeschaltet werden.

Die Funktion der Statussteuerung wird mittels LED angezeigt, die damit die Bereitschaft einer Dateneingabe ueber die Tastatur signalisiert.

Die schaltungstechnische Realisierung kann mit verschiedenen Mitteln erfolgen. Die einfachste Moeglichkeit besteht in der Decodierung mit einem 74154, der Codierung mit 2 DL 021 und einem DL 074 fuer die Anzeige. Somit kann die gesamte Hardware auf 4 IC's beschraenkt bleiben. Um aber den arg strapazierten Netzteil nicht noch mehr zu belasten, wurde eine etwas aufwendigere CMOS-Schaltung entwickelt, die dazu noch die hoehere Sicherheit gegen unerwuenschte Uebergangswiderstaende aufweist.

Die Spalte 7 (SHIFT, CTRL) wird durch die Statussteuerung nicht aktiviert (Fehlen einer Diode), damit beim alleinigen Druucken von SHIFT oder CTRL der Tastaturstatus nicht veraendert wird. Diese Spalte wird erst nach Erkennung einer anderen Taste abgefragt, um die SHIFT-Ebene umzuschalten bzw.

eine Control-Maske (1FH) aufzulegen. Die geringe Anzahl zur Verfügung stehender zusätzlicher Steuerleitungen macht es erforderlich, für die Anzeige der Funktionen SHIFT-LOCK und Hardcopy zusätzliche Flip-Flops mit einer gemeinsamen Resetleitung zu verwenden. Bei jeder Neubelegung eines Flip-Flops werden deshalb durch den Rechner immer beide Flip-Flops neu gesetzt. An der Resetleitung ist ebenfalls der Tastenklick angeschlossen, wodurch die Flip-Flops auch nach der Ausgabe des Tones neu gesetzt werden müssen. Der Tastenklick wird softwaremäßig erzeugt, und ist durch Tastendruck ein- und ausschaltbar.

Für die CMOS-Version wurde eine Leiterplatte entwickelt und digitalisiert, die von der Größe her dem Raum zwischen dem Haupttastenfeld und den rechten Funktionstasten angepasst wurde.

Damit muss die Leiterplatte nicht unterhalb der Tastatur montiert werden, und es ergibt sich eine sehr flache moderne Bauform für das Tastaturgehäuse. Das moderne Äußere wird auch durch das nur 10-adrige dünne Kabel von der Tastatur zum Rechner unterstützt, für das 10-adrige flexible Telefon-Leitung verwendet wurde.

## Software

Für die Tastaturabfrage wurde ein neues Abfrageprogramm entwickelt, welches der verwendeten Hardware angepasst ist. Es hat mit Tabellen eine Länge von einem 3/4 K und kann deshalb nicht den originalen Standort beziehen. Durch den auf der Originalaufrufadresse (F130H) eingetragenen Sprung zum neuen Standort (F800H) wird die Software-Kompatibilität zum Originalmonitor erreicht. Es muss aber eingeschränkt werden, dass Programme, die in die originale Tastaturroutine hineinspringen (eine läienhafte Unsitte), nicht mehr fehlerfrei laufen.

Bei der Realisierung des Programms wurde ein in jeder Hinsicht offenes Prinzip verwendet. Das heißt, dass der Nutzer in der Lage ist, sich die Tastatur nach eigenen Wünschen zusammenzustellen. Bis auf zwei Tasten (SHIFT, CTRL) sind alle Tasten neu belegbar. Weiterhin kann auch wahlweise ein String auf beliebige Tasten gelegt werden. Wenn dafür keine besonderen Tasten verwendet werden sollen, können Strings auch durch gemeinsames Drücken von SHIFT und CTRL sowie einer dritten Zeichentaste erzeugt werden. Der Tastaturtreiber erzeugt dazu intern einen Grafikcode (>80H), mit dem ein mit diesem Zeichen versehener String in einem Stringfeld gesucht und ausgegeben wird. Sollte kein String mit diesem Kennzeichen existieren, wird dieses interne Grafikzeichen ausgegeben. Die Ausgabe von Grafikzeichen im Grafikmode wird dadurch nicht beeinflusst.

Für die einzelnen Felder (Stringfeld, Tastencodefeld, Funktions-tastenfeld und Funktionstastenadressfeld) gibt es ein Pointerfeld, welches auf dem ehemaligen Tastencodefeld ab 3BH angelegt wurde. Damit ist es dem Anwender möglich, sich durch das Verändern dieser Pointer an beliebiger Stelle im RAM ein neues Feld aufzubauen. Diese Möglichkeit hat besonders für das Stringfeld große Bedeutung, da man dadurch in der Lage ist, Strings für verschiedene Anwendungen zu laden (z.B. BASIC-Schlüsselwörter, Assemblermnemonik, Briefköpfe u.ä.). Durch die Möglichkeit, auch die anderen Felder neu zusammenzustellen, kann auch dem Problem begegnet werden, dass z.B. Programme verschiedene Tastencodes für gleiche Steuerfunktionen benutzen.

Das Pointerfeld wird durch den Monitor beim ersten Reset auf die Standardwerte gestellt (nochmaliges Reset wird am C3H auf Adr. 66H erkannt), womit gesichert ist, dass bei jedem weiteren Reset ein evtl. geändertes Pointerfeld erhalten bleibt.

Das hat aber die Konsequenz, dass ein Programmabsturz auch dieses Pointerfeld ungewollt

zerstoeren kann und der Rechner trotz Reset nicht mehr bedienbar ist. Dann hilft nur noch Ausschalten des Rechners! Aendert man folgende vier Zellen, so kann erreicht werden, dass die Neuinitialisierung bei jedem Reset erfolgt.

```
F02AH  02H  --> F2H
F02BH  F2H  --> F1H
F02DH  43H  --> 33H
F030H  0AH  --> 1AH
```

Man sollte dann aber auch darauf achten, dass aus Programmen nicht mit Reset oder Sprung zu F000H herausgegangen wird, sondern mit der Monitorrueckkehrtaste (RST 38H). Ansonsten muss die Pointerinitialisierung neu erfolgen.

Fuer die einzelnen Modes der Tastatur (Grafik, Shift-lock, Hardcopy, Stringmode, Beep, Slow/Fast) existieren Flags, die in der Systemzelle 27H stehen (ehemaliger Schalter Grafik/Alpha). Ein gesetztes Flag (1) bedeutet darin:

```
Bit 0 - lange Repeatverzoegerung durchlaufen
  1 - Stringmode on
  2 - Grafikmode on
  3 - Hardcopy on
  4 - Shift-lock (gilt nur fuer Buchstaben)
  5 - Tastenklick on
  6 - Slow
  7 - Taste war betaetigt
```

Diese Flags werden durch den Tastatortreiber gesetzt bzw. rueckgesetzt. Indem man diese Flags anderweitig setzt oder ruecksetzt, kann die Betriebsart der Tastatur von aussen manipuliert werden. Bei den Funktionstasten fallen fuenf Tasten besonders auf. Mit der Taste F01 laesst sich aus einem beliebigen Anwenderprogramm heraus zum Monitor (RST 38H) zurueckkehren, insofern die Tastatur noch abgefragt wird. Dabei wird auch gleichzeitig eine Reinitialisierung des Zusatz-Kommandoverteilers auf B0H durchgefuehrt, so dass die standardmaessig eingetragenen Kommandos (@L, @S, @D, @I) wieder verfuegbar werden.

Die Tasten F06, F07, F08 dienen fuer DirektsprUenge aus der Tastaturroutine heraus zu festen Adressen (100H, 200H, 300H), da dort die meisten Programme zu starten sind. Der Stack wird dazu vor der Ausfuehrung des Sprungs neu initialisiert.

Eine weitere wichtige Taste ist die Screen-Copy-Taste (F09). Mit ihr ist es moeglich, eine Bildschirmkopie auf dem Drucker zu erzeugen (Druckereinbindung ueber Sprungverteiler DRAKK), ohne das laufende Programm zu unterbrechen. Dabei ist zu beachten, dass waehrend der Bildschirmkopie der Stack des laufenden Programms genutzt wird (Stackbelastung je nach Druckertreiber).

Die Tastenentprellung erfolgt positiv und negativ, d.h., damit eine Taste als betaetigt erkannt wird, muss mindestens 15 ms (2 MHz) lang ein prellfreies Signal anliegen (positive Entprellung). Das gleiche gilt uebertragen fuer das Loslassen der Taste (negative Entprellung). Dadurch kommt es nicht vor, dass selbst bei sehr schlechten Kontakten eine Fehlfunktion auftritt.

Wird eine Taste laenger als 0.5 s (2 MHz) betaetigt, wird die Repeatfunktion (Echofunktion) bei Zeichentasten aktiviert.

Der Normalaustritt aus der Tastaturroutine erfolgt kompatibel zur Originalroutine mit dem Zeichen im Akku und in Zelle 4.

Eine „blinde“ Tastaturabfrage erfolgt, indem vor dem Aufruf der Routine die Zelle 4 auf 0 gesetzt wird.

## Standard-Tastenbelegung

Funktionstasten (erzeugen keinen Code):

```
F01 - RST 38H
F06 - JMP 100H
F07 - JMP 200H
F08 - JMP 300H
F09 - SCREEN-COPY
F53 - GRAFIK ON/OFF
E53 - HARDCOPY ON/OFF
D53 - SLOW/FAST
C53 - BEEP ON/OFF
D00 - SHIFT-LOCK ON/OFF
B11 - SHIFT (+B99)
C00 - CTRL
```

Steuerzeichentasten (derzeitige Belegung)

Pos.	1.Code	2.Code	1.Funktion	2.Fkt(.m.Shift)	1.CTRL	2.CTRL
A00	0AH	0AH	Cursor runter	Cursor runter	J	J
A01	0BH	0BH	Cursor hoch	Cursor hoch	K	K
A10	08	08	Cursor links	Cursor links	H	H
A11	09	09	Cursor rechts	Cursor rechts	I	I
F14	03	03	BREAK	BREAK	C	C
F02	1BH	1BH	ESCAPE	ESCAPE		
B13	0DH	0DH	ENTER	ENTER	M	M
F05	06	1EH	TAB-DUMMY	NL	F	
D95	05	1CH	CRSR ANF.EDIT.	Fettdruck-DUMMY	E	
B95	01	1DH	CRSR END.EDIT.	Norm.druck-DUMMY	A	
E14	14H	13H	LIST-D. (HC-BASIC)		T	S
D13	7FH	02	DEL-MTX	DEL-EDITOR 5.1		B
F10	15H	0FH	RUN-D. (HC-BASIC)		U	0
F11	19H	18H	C.ANF.-HC-BASIC	C.END.-HC-BASIC	Y	X
F12	12H	11H	INSERT HC-BASIC		R	Q
F13	10H	1FH	DEL HC-BASIC		P	
E95	1CH	1CH	Fettdr.DUMMY	Fettdr.DUMMY		
F95	1DH	1DH	Norm.dr.DUMMY	Norm.dr.DUMMY		

Diese Steuerzeichen stehen mit in der Tastenbelegungstabelle und koennen somit anderen Erfordernissen angepasst werden.

## Tastenbelegungstabelle erste Shiftebene und Zweite Shiftebene

Code	Pos.	Zeichen	Code	Pos.	Zeichen
31H	E1	1	21H	E1	!
51H	D1	Q	71H	D1	q
41H	C1	A	61H	C1	a
59H	B1	Y	79H	B1	y
32H	E2	2	22H	E2	"
57H	D2	W	77H	D2	w
53H	C2	S	73H	C2	s
58H	B2	X	78H	B2	x
33H	E3	3	40H	E3	@
45H	D3	E	65H	D3	e
44H	C3	D	64H	C3	d
43H	B3	C	63H	B3	c
34H	E4	4	24H	E4	\$
52H	D4	R	72H	D4	r
46H	C4	F	66H	C4	f
56H	B4	V	76H	B4	v
35H	E5	5	25H	E5	%
54H	D5	T	74H	D5	t
47H	C5	G	67H	C5	g
42H	B5	B	62H	B5	b
36H	E6	6	26H	E6	&
5AH	D6	Z	7AH	D6	z
48H	C6	H	68H	C6	h
4EH	A6	N	6EH	B6	n
37H	E7	7	2FH	E7	/
55H	D7	U	75H	D7	u
4AH	C7	J	6AH	C7	j
4DH	B7	M	6DH	B7	m
38H	E8	8	28H	E8	(
49H	D8	I	69H	D8	i
4BH	C8	K	6BH	C8	k
2CH	B8	,	3BH	B8	;
39H	E9	9	29H	E9	)
4FH	D9	0	6FH	D9	o
4CH	C9	L	6CH	C9	l
2EH	B9	.	3AH	B9	:
30H	E10	0	3DH	E10	=
50H	D10	P	70H	D10	p
5CH	C10	Ö (Oe)	7CH	C10	ö (oe)
2DH	B10	-	5FH	B10	_
7EH	E11	ß (sz)	3FH	E11	?
5DH	D11	Ü (Ue)	7DH	D11	ü (ue)
5BH	C11	Ä (Ae)	7BH	C11	ä (ae)
3CH	E12	<	5BH	E12	Ä
2BH	D12	+	2AH	D12	*
23H	C12	#	27H	C12	'
5EH	E00	^ (CTRL-ZEICHEN)	7CH	E0	ö
40H	B0	@ (kommerzielles A)	5CH	B0	Ö

3EH	E13	>		5DH	E13	Ü	
0DH	B13	ENTER	CTRL-M	0DH	C,B13	ENTER	
CR							
0BH	A1	Cursor hoch		0BH	A01	Cursor hoch	
8H	A10	<--- (Cursor links) CTRL-H		8H	A10	<---	
20H	A5	Leertaste (Space)		20H	A05	Leertaste	
(Space)							
0	D0	Shift-CAPS-Tast. ON/OFF		0	D00	Sift-CAPS-Tast.	
ON/OFF							
0AH	A0	Cursor runter CTRL-J		0AH	A00	Cursor runter	
9H	A11	---> (Cursor rechts)		9H	A11	--->	
0	B99,B11	SHIFT		0	B99,B11	SHIFT	
0	C0	CTRL-TASTE		0	C00	CTRL-TASTE	
7FH	D13	DEL (rotes C)	DEL	2H	D13	DEL (rotes	
C)							
14H	E14	^T (rotes L)		13H	E14	^T (rotes L)	
0	F01	Monitorrueckkehr		0	F01	Monitor-Ruecksprung	
1BH	F02	ESCAPE		1BH	F02	ESCAPE	
7BH	F03	ä		60H	F03	neg.	
Hochkomma							
7DH	F04	ä		7EH	F04	ß	
0	F06	JMP 100H		0	F06	JMP 100H	
0	F07	JMP 200H		0	F07	JMP 200H	
0	F08	JMP 300H		0	F08	JMP 300H	
0	F09	CALL Screen-Copy (BS-Druck)		0	F09	Screen-Copy (BS-	
Druck)							
15H	F10	^U (RUN-BASIC)		0FH	F10	^0	
19H	F11	^Y		18H	F11	^X (Cursor an	
Zeilenende)							
12H	F12	^R (INSERT-BASIC)		11H	F12	^Q	
10H	F13	^P (DEL-BASIC)		1FH	F13		
3H	F14	^C = BREAK-Taste		03	F14	^C (BREAK-Taste)	
6H	F05	^F (TAB-Editor)		1EH	F05	NL (SIF1000)	
0	F53	GRAFIK ON/OFF		0	F53	GRAFIK ON/OFF	
0	E53	HARDCOPY ON/OFF		0	E53	HARDCOPY ON/OFF	
0	D53	SLOW/FAST		0	D53	SLOW/FAST	
0	C53	BEEP ON/OFF		0	C53	BEEP ON/OFF	
1CH	E95	^Ö (schwarz-rot)		1CH	E95	(schw.-rot)	
1DH	F95	^Ü (rot-schwarz)		1DH	F95	(rot-schw.)	
5H	D,C95	^E (Wippe +)		19H	D,C95	^Y (Wippe +)	
1H	B95	^A (Wippe -)		18H	B95	^X (Wippe -)	

Den Standort der Tastenbelegungstabelle kann man dem Tastenbelegungsfieldpointer (auf 3BH) entnehmen, der auf die Tastenbelegungstabelle im ROM zeigt (standardmaessig).

## Aufbau des Pointerfeldes

Das Pointerfeld befindet sich auf dem Platz des jetzt nicht mehr benoetigten Tastaturcodefeldes und wird vom Monitor initialisiert. Das originale Tastaturcodefeld beginnt auf Adresse 35H und wird durch

den Monitorsprung RST 38H unterbrochen. Die ersten drei Zellen werden fuer Zaehlzellen des Druckertreibers genutzt und sind mit 0 initialisiert. Das Pointerfeld beginnt erst hinter dem RST 38H auf 3BH mit folgendem Aufbau:

```
3BH - Tastenbelegungsfieldpointer
3DH - Stringfieldpointer
3FH - Pointer auf naechstes auszug. Stringzeichen
41H - Laenge Funktionstastenpos.-feld
43H - Funktionstastenpositionsfieldpointer
45H - Funktionstastenadressfieldpointer
```

## Aufbau des Stringfeldes

Das Stringfeld hat einen sehr einfachen und codeeffektiven Aufbau. Der auszugebende String beginnt mit dem Namen des Strings, welcher nur aus einem Grafikzeichen besteht. Es werden solange Zeichen von diesem String ausgegeben, bis ein neues Grafikzeichen auftritt oder das Ende des Stringfeldes - gekennzeichnet mit einer 0 - erreicht ist. Damit ist auch klar, dass sich im String keine Grafikzeichen befinden duerfen, da diese fuer die Adressierung der Strings reserviert sind. Die Strings koennen eine beliebige Laenge haben, und in beliebiger Reihenfolge angeordnet sein.

Stringfeldaufbau:

```
1. Grfk-Z. 1.String
2. erstes Stringz.
.
.
.
n. letztes Stringz.
n+1 Grfk.Z. 2. String
n+2 erstes Zeichen 2. String
.
.
.
0 ;Ende des Feldes
```

Guenstig ist es, sich auf Kassette einige Felder vorzubereiten und diese mit einer kleinen Umladeroutine zu versehen, welche das Eintragen des Stringfieldpointers entsprechend des Standortes vornimmt. Somit ist das jeweilige Stringfeld nach einem Autostart sofort nutzbar.

Die fuer die Adressierung der Strings benoetigten Grafikzeichen koennen in der Tastenbelegungstabelle stehen, welche bei Betaetigung der entsprechenden Taste wirksam werden, worauf der dazugehoerige String ausgegeben wird. Eine einfachere Moeglichkeit besteht in der gleichzeitigen Betaetigung von SHIFT und CTRL sowie einer Zeichentaste, worauf folgende interne Grafikzeichen erzeugt werden:

```
SHIFT+CTRL+Zeichentaste
A 81H
B 82H
C 83H
D 84H
```

E	85H	
F	88H	
G	87H	
H	8BH	
I	89H	
J	8AH	
K	9CH	
L	8CH	
M	8DH	
N	8BH	;DOPPELBEL. ZU H
O	8FH	
P	90H	
Q	91H	
R	92H	
S	86H	
T	94H	
U	95H	
V	88H	;DOPPELBEL. ZU F
W	97H	
X	86H	;DOPPELBEL. ZU S
Y	99H	
Z	9AH	
Ä	9BH	
Ö	9EH	
Ü	9DH	
0	90H	;DOPPELBEL. ZU P
1	91H	; - " - Q
2	92H	; - " - R
3	93H	
4	94H	; - " - T
5	95H	; - " - U
6	96H	
7	97H	; - " - W
8	98H	
9	99H	; - " - Y

Die Unregelmässigkeiten zu den korrespondierenden CTRL-Zeichen entstehen beim gleichzeitigen Druecken von SHIFT und CTRL durch das Kurzschliessen der Matrixzeile 7 und 8, da diese hardwaremaessig nicht entkoppelt sind.

Im EPROM wurde ein Standardstringfeld mit den am haeufigsten vorkommenden Basic-Schlueselwoertern eingerichtet.

### Direktzugriffe auf die Matrix

Obwohl das direkte Zugreifen auf die Tastaturmatrix von einem Anwenderprogramm aus unnoetig und nur in sehr seltenen Faellen gerechtfertigt ist, gibt es eine Vielzahl von Programmen (meist Spielprogramme), die diese „Spektrum-Manieren“ anwenden. Um solche Programme ohne Aenderung nutzen zu koennen, soll folgende Korrespondenztabelle dienen:

orig.	neu	Pos.
@	1	E01
A	3	E03
B	5	E05
C	7	E07
D	9	E09
E	B	E11
F	>	E13
G		
H	Q	D01
I	E	D03
J	T	D05
K	U	D07
L	0	D09
M	Ue	D11
N	ENT	C/B13
O		
P	Y	B01
Q	C	B03
R	B	B05
S	M	B07
T	.	B09
U	<	E12
V	<-	A10
W		
S1	X	B02
S2	V	B04
S3	N	B06
S4	,	B08
<-	-	B10
SPACE	@	B00
->	->	A11
ENT	CTRL	C00

Drei Tasten koennen nicht konvertiert werden, da diese Positionen bei der K 7659 nicht besetzt sind. Die Korrespondenztabelle kann auch dazu benutzt werden, um die Hardware mit dem originalen Tastaturprogramm zu testen.

## Aenderungen am Monitorprogramm

Eine der wichtigsten Aenderungen ist das Eintragen des Aussprunges aus dem originalen Tastaturprogramm und der veraenderten Initialisierung des ehemaligen Tastaturcodefelds. Alles andere koennte original bleiben. Es werden aber weitere Aenderungen vorgenommen, die nicht die Softwarekompatibilitaet verringern.

Solche Veraenderungen sind:

- NMI-Unterbrechung mit Registeranzeige (Init.66H bei Reset)

- Hardcopyfunktion des Bildschirmtreibers
- Beschreiben eines IO-Ports vom Monitor aus („O“ ehemals „H“)
- Auflistung der im Zusatzkommandoverteiler stehenden Kommandos („Z“ ehemals „A“)
- Initialisierung des Zusatzkommandoverteilers mit Standardwerten (@L->Headersave load, @S->Headersave save, @D->Druckerreset, @I-> Druckerinitialisierung)

Da der zusätzliche Speicher mit dem Tastaturprogramm noch nicht ausgelastet ist, wurden auf dem Rest des Speichers das Headersave, das Screencopyprogramm, die Joystickabfrage sowie der Sprungverteiler fuer die Systemerweiterung fest installiert.

Fuer die Uebernahme der Software ist es guenstig, die kompletten 4K ab F000H zu uebernehmen.

Die umfassende Vorstellung des Sprungverteilers wuerde hier zu weit fuehren. Deshalb sollen nur die wichtigsten Spruenge beschrieben werden:

```
FFF7H - JMP STAT uebergibt Tastaturstatus im Akku
        A=0   - keine Taste gedr.
        A=FFH - Taste gedr. (ausser SHIFT o. CTRL)

FFF4H - JMP SARUF ruft SAVE-Routine des Headersave
        Parameteruebergabe entspr. Beschr. Headersave

FFF1H - JMP LORUF
        Parameteruebergabe entspr. Beschr. Headersave

FFE8H - JMP DRDEL setzt den logischen Druckertreiber zurueck
        (Zaehlzellen fuer Zeile- und Spaltenposition werden zu 0)

FFE5H - JMP BSDR druckt den Inhalt des BWS bis zur Cursorposit.
        und kehrt in das aufrufende Programm zurueck

FFDFH - JMP DRZEL wie DRAKK, nur dass das Zeichen in Zelle 1BH
        uebergeben wird (vorgesehen, um im Basic mit POKE Zelle
        und CALL 0FFDFH zu drucken)

FFCDH - JMP DRINI Initialisierung des logischen Druckertreibers

FFCAH - JMP ZEIDR uebergibt ein Zeichen im Akku an physischen
        Druckertreiber

FFBBH - JMP GETST Abfrage der Joysticks und Uebergabe des Ergebnis-
        ses in BC (B-links,C-rechts) mit folgender Bit-Bedeutung (1)
        Bit 0 - links
            1 - rechts
            2 - runter
            3 - hoch
            4 - Aktionstaste
```

Z-Flag=1, wenn keine Betaetigung vorliegt  
 CY-Flag=1, wenn Spielhebel nicht angeschlossen sind

FFB8H - JMP SOUND Ausgabe einer vollen Periode auf die Tonband-  
 buchse, sowie auf Bit 7 vom Systemport  
 Uebergabe der Periodendauer in C mit  $T=n*33mks+20mks$  (2MHz)

Die Spruenge DRINI, DRZEL, DRAKK und ZEIDR sind nicht in den oberen 4K realisiert und sind mit JMP 0E800H, 0E803H, 0E806H und E809H initialisiert, wo man einen Unterverteiler fuer den verwendeten Druckertreiber installieren kann.

## Zusammenfassung

Es wurde eine alternative Tastaturvariante beschrieben, die professionellen Anspruechen gerecht wird und durch folgende Vorteile gekennzeichnet ist:

- geringer leistungsarmer Hardwareaufwand
- zwischen Tastatur und Rechner nur 10-adriges Kabel erforderlich
- Anzeige der Tastaturmodes durch LED's
- schaltbarer Tastenklick
- schnelle und sichere Abfrage
- Interruptfaehigkeit
- nur noch zwei SHIFT-Ebenen
- hohe Softwarekompatibilitaet
- freie Belegbarkeit der Tasten
- frei programmierbare Stringausgabe mit Quittung
- Repeatfunktion auf allen Zeichentasten
- einfache Monitorrueckkehr durch RST 38H-Taste
- direkter UP-Aufruf Screencopy aus der Tastaturroutine heraus
- Direktspruenge zu festen Adressen aus der Tastaturroutine heraus
- es koennen gegenueber der alten Routine alle Steuerzeichen erzeugt werden
- Abfrage der vollen 8\*12 Matrix
- einfache Erweiterung der Tastatur auf 96 Tasten, indem die nicht besetzten Positionen aufgefuellt werden

Dem gegenueber stehen die Nachteile:

- es wird zusaetzlicher Adressraum des Hauptspeichers benutzt
- es muss eine Aenderung des originalen Monitorprogramms erfolgen
- nur bedingte Kompatibilitaet bei Direktzugriffen auf die Matrix und bei Spruengen in die originale Routine hinein

Betrachtet man Vor- und Nachteile, so kann festgestellt werden, dass es sich lohnt, die Nachteile in Kauf zu nehmen, zumal eine Aenderung des Monitors nur einmal erfolgen muss. In jedem Fall stellt diese Tastatur eine echte Alternative zu der Prozessortastatur von Riesa dar, die in jedem Fall einen

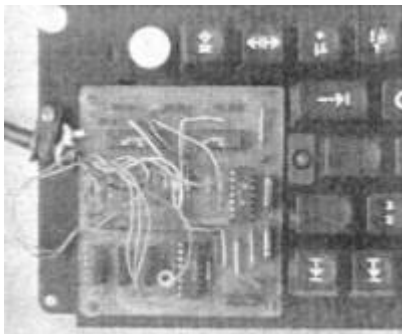
hoeheren Hardwareaufwand erfordert und eine geringere Leistungsfahigkeit besitzt.

\*) ZUSATZ: Es gibt diesen Monitor auch fuer die Tastatur der elektronischen Schreibmaschine 'S3006', der K 7652. Bei der Bestueckung der Leiterplatte ist nur darauf zu achten, dass Spalte 10 und 12 nicht belegt, und dass bei der K7659 und K7652 Spalten und Zeilen (A u. B) umgekehrt sind.

## Version practic

Von Gerhard Eisenkolb stammt eine einfacher aufzubauende Schaltung, die 100% kompatibel zu obiger Lösung ist. Anstelle einer zweiseitigen Leiterplatte wird nur eine einseitige Platine genutzt.

[practic 3/89, S. 137-138](#)



## K7669 Anschluss nach Brosig

Seit 1985 befindet sich ein 4-K-Monitor fuer die Tastatur K 7659 und spaeter auch fuer K 7652 in der Anwendung der Z-1013-User (2.028 RB). Durch die Entwicklung der Amateurtastatur K 7669 wurde es notwendig, diesen 4-K-Monitor auch fuer diese Tastatur anzupassen.

Der originale Monitor-EPROM ist auszuloeten und durch einen anderen EPROM mit Fassung zu ersetzen. Die zusaetzliche Hardware beschraenkt sich auf einen 1-aus-16-Decoder, die Statussteuerung und einen 8-auf-4-Coder. Zur Anzeige von CAPS-lock und Hardcopy werden zwei Anzeige-Flip-Flops verwendet. Die fuer die Tastaturtypen K7659 und K7652 eingesetzten Zusatzleiterplatten fuer die Ansteuerschaltung koennen unveraendert auch fuer die K7669 eingesetzt werden. Die nichtvorhandenen Spalten werden einfach nicht angeschlossen (Achtung: Die Anschlussbelegung an den Tastatursteckern ist bei allen Tastaturen unterschiedlich!)

- [k7669.zip](#) Monitor 2.029 K7669, Quellcode, Beschreibung

## K7659 Alphatastatur mit U880

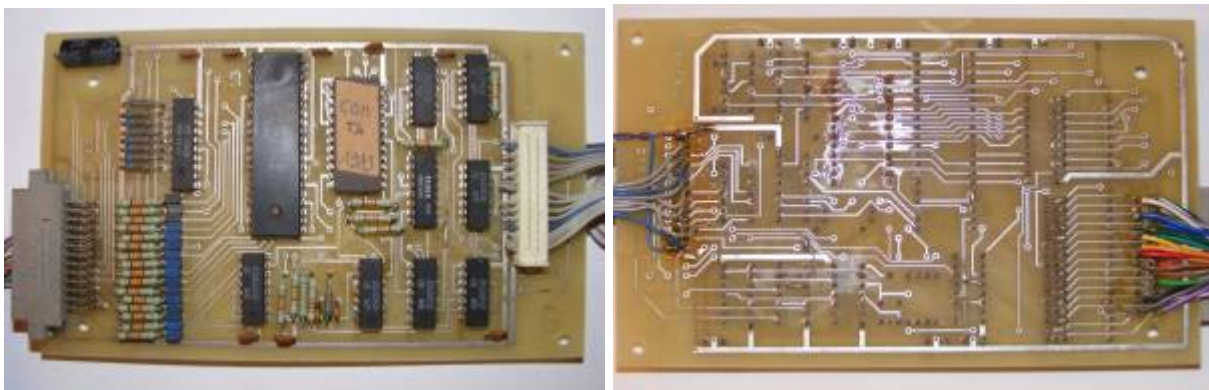
Anschluss einer K7659-Tastatur mit eigenem Prozessor, von Riesa



- [riesa-alphatastatur.pdf](#)

## Commodore-Tastatur mit U880

analog zum Anschluss einer K7659-Tastatur gibt es auch eine Variante zum Anschluss einer Commodore-Tastatur.



- [com-ta.pdf](#) Beschreibung
- [com-ta.zip](#) EPROM-Inhalt

Bild des Z1013 mit der 6x16-Tastatur



*Vielen Dank an Matthias*

From:

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

Permanent link:

<https://hc-ddr.hucki.net/wiki/doku.php/z1013/erweiterungen/tastatur?rev=1337754762>

Last update: **2012/05/23 06:32**

