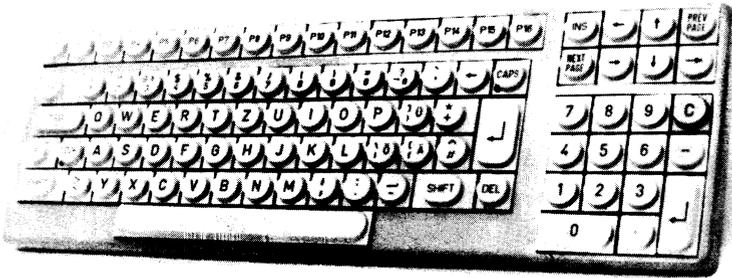


# ELZET 80 DIN-Tastatur



Schreibmaschinentastatur mit seriellem und parallelem Ausgang, durch eigenen Prozessor sehr kompakt aufgebaut: 122 x 392 x 22mm. Das zugehörige Kunststoffgehäuse mißt nur 129 x 400mm. Die Abstände von der Tischplatte zur Tastenmitte in ungedrücktem Zustand betragen: für die Leertaste 25mm. für die Tastenreihe asdf (Mitte) 30mm. für die Zahlenreihe 36mm und für die Funktionstasten 40mm. Damit entspricht die Tastatur dem aktuellen Erkenntnisstand für ermüdungsfreie Arbeit.

Die Tastatur hat 96 Tasten, im Hauptfeld in der deutschen Standardanordnung. Ä, Ö, Ü, § und ß sind auf den Tasten zusätzlich mit der amerikanischen Belegung, also eckige und geschweifte Klammer, Tilde etc. markiert.

Alle Tasten haben eine automatische Dauerfunktion. Dabei kann die Einsatzzeit und Wiederholrate eingestellt werden.

SHIFT-Arretierung kann durch beide SHIFT-Tasten wieder ausgelöst werden, LED-Anzeige des SHIFT-Zustands. CAPS-Funktion zur Eingabe von Großbuchstaben und Zahlen, ebenfalls mit LED-Anzeige. Großflächige Wagenrücklauftaste (ENTER/RETURN), Tasten für TAB, CTRL, ESC, Backspace und DEL.

16 Funktionstasten sind zunächst fest belegt, dann jedoch durch eine bestimmte Tastensequenz mit jedem beliebigen Zeichen programmierbar, also z.B. a, ? oder auch CTRL/B. Die Funktionstasten sind auch mit ESC + Zeichen programmierbar.

Ein 8-Tasten Cursor- und Editierfeld oben rechts ist mit Controlzeichen belegt. Darunter ist ein Zehnertastenfeld mit doppelt breiter Nulltaste, eigener ENTER- und MINUS- Taste und einer mit ^U belegten "C"ancel-Taste. Im Betrieb mit einem Terminal-Rechner können -über einen Jumper selektierbar- die Funktions- und Editiertasten als 8-Bit-Nicht-ASCII-Code ausgegeben werden. Damit können dann die Hex-Codes 80 bis AA und C0 bis EA erzeugt werden.

Die Baudrate der Serienschnittstelle ist einstellbar von 1200 bis 19200 Baud, ein zweiter Serienausgang treibt einen Optokoppler (20mA Stromschleife). Die Tastatur wird ausgeliefert mit der Einstellung auf Stromschleifentreiber und 9600 Baud. Wird eine andere Ausgangsart gewünscht, so muß dies bei Bestellung einer Tastatur im Gehäuse angegeben werden, da eine nachträgliche Änderung nicht möglich ist. Die Einsteller für die Wiederholrate, Terminal-Betriebsart, Stromschleifenphase und Diagnostik sind auch ohne Öffnen des Gehäuses möglich, da sie unter der Leertaste liegen.

Als einzige Versorgungsspannung wird +5V benötigt. Das bedeutet, daß die Tastatur bequem über eine dünne zweiadrige Leitung mit Abschirmung angeschlossen werden kann: Eine Ader für die +5V, eine Ader für den Stromschleifenausgang und die Abschirmung für die Masseverbindung. Damit wird die Tastatur noch handlicher auf dem Schreibtisch.

Die verwendeten SIEMENS-Tasten erlauben 5 Millionen Betätigungen. Bei größeren Abnahmen sind aus dem Siemens-Programm andere Tastenkappen wählbar, sowie auch Sondertasten nach Kundenzeichnung.

## SCHALTUNGSBESCHREIBUNG

Die Tastaturelektronik besteht aus einem Ein-Chip-Mikrocomputer vom Typ 8048 und einem 4 zu 16 -Dekoder 74159. Der 74159 wird über 4 Bits eines Ausgangsports des 8048 angesprochen. Zwischen den beiden Bausteinen ist eine willkürlich verdrahtete Tastenmatrix angeordnet.

Der 8048 erzeugt 'reihum' unter Zuhilfenahme des Dekoders 14 Spaltensignale, und fragt nach dem Ansprechen jeder Spalte die 8 Zeilen über den Datenbus ab. Wird eine Taste gedrückt, erfolgt als nächster Schritt die Abfrage der Shift und Control-Tasten sowie der internen Flip-Flops für SHIFT-LOCK und ALPHA-LOCK. Dann ordnet der Prozessor aus einer Tabelle die Spalte und Zeile der Tastenmatrix dem entsprechenden ASCII-Code zu.

Der Code wird dann, abhängig von der Stellung der Baudratenjumper entweder auf dem Parallelport oder auf dem Serienport in der gewünschten Baudrate ausgegeben. Der parallele und der serielle Ausgang werden als ungepufferte TTL-kompatible Signale direkt vom Prozessor auf die Kabelanschlüsse geführt. Die Belastbarkeit eines solchen Ausgangs liegt bei minimal ca. 10mA für Ausgang Low und bei min. ca. -50uA für Ausgang High. Zu beachten ist jedoch, daß die gesamte Kabellänge 75cm nicht überschreiten sollte. Ist

Belegung der Sondertasten im Terminalbetrieb:

P1 bis P16 80H bis 8FH normal und SHIFT, C0 bis CF bei CTRL

Cursorfeld 1. Zeile von links nach rechts:  
90H bis 93H normal, D0 bis D3 bei SHIFT oder CTRL

Cursorfeld 2. Zeile von links nach rechts:  
94H bis 97H normal, D4 bis D7 bei SHIFT oder CTRL

Dezimalfeld:

Tastenaufdruck	Ausgabe normal	Ausgabe SHIFT oder CTRL
7	9D	DD
8	9E	DE
9	9F	DF
C	A0	E0
4	A1	E1
5	A2	E2
6	A3	E3
-	A4	E4
1	A5	E5
2	A6	E6
3	A7	E7
0	A8	E8
.	A9	E9
<CR>	AA	EA

### Sonderfunktionen:

Rechts neben dem Quarz sind die Jumper für Sonderfunktionen. Stecken des Linken bedeutet 'Terminal-Betrieb' (8 Bit für Sonder-tasten). Der Mittlere startet gesteckt das Diagnostikprogramm. Der rechte Jumper wird für schnelle Wiederholrate gesteckt.

### Phase der Stromschleife:

Rechts neben dem Widerstandsnetzwerk ist der Jumper für die Phaseneinstellung des Stromschleifentreibers. Nach oben gesteckt bedeutet Normalbetrieb, unten gesteckt bedeutet invertierten Betrieb. Für das Zusammenspiel mit der ELZET 80 CPU ist "unten" zu stecken.

### Parallelausgang:

Über der ESC-Taste ist der Parallelausgang. Von links nach rechts (Aufsicht auf Tastatur) sind die 10 Lötäugen mit Bit 1 bis 8, /STROBE und Masse belegt. Der +5V Eingang ist auf den seriellen +5V Anschluß oder Pin 40 des Prozessors zu verdrahten.

### Serieller Ausgang:

Der Seriena Ausgang (Über Backspace und CAPS) ist von links nach rechts belegt mit TTL-Ausgang, Stromschleifen Ausgang, 3x Masse und +5V.

### Belegung der Sondertasten:

Die Programmierung der Funktions- und Editiertasten ist wie folgt:

P1 bis P8:

CTRL S, ESC B, ESC E, ESC F, ESC G, ESC I, ESC J, ESC K

P9 bis P16:

ESC L, ESC M, ESC N, ESC O, ESC P, ESC Q, ESC R, ESC S

Editierfeld, obere Zeile:

CTRL P, CTRL C, CTRL X, CTRL Z

Editierfeld, untere Zeile:

CTRL G, CTRL R, CTRL E, CTRL F

Die farbige 'C'-Taste im Nummernfeld ist mit CTRL U belegt.

eine längere Verbindung zum Prozessor erforderlich, so empfehlen wir den Betrieb über den seriellen Stromschleifenausgang, der zumindest im Bereich bis 10m unkritisch ist. Die Serienschnittstellen der ELZET 80 CPU-Karten verfügen über entsprechende Optokoppler-Eingänge.

Weitere Ausgänge des 8048 treiben die Leuchtanzeigen für SHIFT-LOCK und CAPS und zeigen den Zustand der internen (Software-) Flipflops an. Es werden noch drei Jumper abgefragt, a) für den Betrieb in einem Terminal-Rechner, b) für ein Diagnostikprogramm, das die gerade gedrückte Taste ständig nicht entprellt auf die Schnittstelle gibt und c) für die Wahl der Wiederholrate (Auto-Repeat).

Die Baudrate wird aus der Prozessor-Quarzfrequenz von 5,185 MHz abgeleitet. Die Ausgabe erfolgt asynchron mit 8 Bit Daten, ohne Parity. mit zwei Stop-Bits. Bei Benutzung des Parallelports liegen 8 Bit Daten an und ein STROBE (Übernahme-) Signal mit Aktiv-Low.

### EINSTELLMÖGLICHKEITEN UND AUSGÄNGE

#### Die Programmierung der Funktionstasten:

Den Funktionstasten P1 bis P16 sind bei Einschalten des Tastatur feste, im Prozessor abgelegte Werte zugeordnet. Die Tasten sind jedoch mit einem beliebigen Code zu belegen, der mit einer Tastensequenz eingegeben wird. Dazu ist zunächst CTRL und, während CTRL festgehalten wird, P1 zu drücken.

Dann folgt die Taste, die programmiert werden soll, beispielsweise P5. Das nächste mit der Tastatur erzeugte Zeichen wird nun in P5 gespeichert. Wird vor diesem Zeichen die ESC-Taste gedrückt, so enthält P5 zwei Zeichen, nämlich ESC und den programmierten Wert.

Während der Tastenprogrammierung wird nichts ausgegeben.

#### Baudrateneinstellung:

Links neben dem Prozessor sind vier Brücken, die durch Kurzschlußstecker oder Draht geschlossen werden können. Sind alle Brücken offen, so ist der parallele Ausgang aktiviert. Bei Stecken mindestens eines Jumpers ist serieller Betrieb eingestellt.

Parallel-Betrieb	offen	offen	offen	offen
1200 Baud seriell	offen	gesteckt	offen	gesteckt
2400 Baud seriell	offen	gesteckt	offen	offen
4800 Baud seriell	offen	offen	gesteckt	gesteckt
9600 Baud seriell	offen	offen	gesteckt	offen
19200 Baud seriell	offen	offen	offen	gesteckt

**ELZET 80** ELEKTRONIKLADEN Giesler & Danne GmbH & Co. KG  
W. Mellies Str. 88 · 4930 Detmold 18 · Tel. 05232-8131 · Tx. 931473

ELZET 80 DIN-TAST

---

Programmierung der  
Cursor Tasten

SHIFT/CAPS

CTRL/LOCK

Cursor Taste

>Text, max. 20 Ch.<

<Cursor Taste>

Multifunktionale, programmierbare Eingabetastatur  
mit parallelem und seriellem ASCII-Ausgang.

**ELZET 80**

Mikrocomputer GmbH & Co. KG

Wilhelm-Mellies-Straße 88

D-4930 Detmold 18

Tel. 0 52 32 - 81 31 · Tx. 931 473 elzet d

Rev. 1.1 Juni 85

**ELZET 80**

9. Aug. 85  
v

Inhalt

Serieller Anschluß.....	5
Paralleler Anschluß.....	6
Belegung der Anschlußkabel.....	7
Steckbrücken für die Funktionsauswahl.....	9
Beschreibung der Betriebsarten.....	12
Schnellspeicher.....	18
Erstellung kundenspezifischer Belegungen.....	19
Tabellenerzeugungsprogramm TAB.....	20
Tastenkappen.....	26
Anhang: Quellprogramm TAB, Standardbelegung EPROM1-4.....	27

## ELZET 80 DIN-TAST

---

Die ELZET 80 DIN-TAST ist eine komfortable Eingabetastatur für alle Rechner, die eine Eingabe im ASCII-Code erwarten, egal ob parallel oder seriell. Ein eigener Prozessor ermöglicht flexible Anpassung an wechselnde Notwendigkeiten.

ELZET 80 DIN-TAST zeichnet sich aus durch:

### Kompaktes Gehäuse

Das Tastaturgehäuse umfaßt nur die äußere Tastenbegrenzung, die Elektronik ist neben der Leertaste untergebracht. Die DIN-TAST nimmt also nicht mehr Platz als unbedingt nötig auf Ihrem Schreibtisch ein. So finden in dem nur 40x13cm großen Gehäuse noch 38 Sondertasten außerhalb des Hauptfelds Platz.

### Mehrfachbelegte Funktionstasten

Die ELZET 80 DIN-TAST kennt 7 Betriebsarten, die durch Tastendruck umschaltbar sind. 4 davon sind durch den Anwender programmierbar, jede in vier Ebenen, durch SHIFT und CTRL umschaltbar. Ausgeliefert wird u.a. mit CP/M- und WordStar-Belegung. Bis zu 20 Zeichen können, jederzeit änderbar, im Schnellspeicher abgelegt werden.

### Ergonomie

Die Tasten sind nach DIN 2137 angeordnet, bei den deutschen Sonderzeichen ä, ö, ü, ß und § ist die US-ASCII-Darstellung ebenfalls mit markiert. Neigung, Bauhöhe und Betätigungsdruck entsprechen den neuesten ergonomischen Erkenntnissen.

## ELZET 80 DIN-TAST

---

### Anschluß

Die Tastatur benötigt zum Betrieb 5V bei ca. 200 mA, für die RS232-Ausführung zusätzlich die Hilfsspannungen  $\pm 12$  bis  $\pm 15$ V. Die Versorgungsspannung kann über das Schnittstellenkabel zugeführt werden.

Entsprechend der universellen Einsatzmöglichkeit verfügt die ELZET 80 DIN-TAST serienmäßig über verschiedene Schnittstellen, die je nach Bedarf (evtl. gleichzeitig) verwendet werden können. Es sind dies:

- Eine 8-Bit Parallelschnittstelle
- Eine serielle Schnittstelle mit TTL-Pegel
- Eine serielle Schnittstelle mit 20mA Strom
- Eine serielle Schnittstelle mit RS232-Pegel

Die Leiterplatte hat verschiedene Anschlußpunkte für die unterschiedlichen Schnittstellen und eine Reihe von Steckbrücken, um Details festzulegen.

Bei Lieferung im Gehäuse wird die Tastatur mit einem Anschluß Ihrer Wahl ausgestattet, d.h. seriell Stromschleife, TTL-Pegel, RS232 oder parallel.

Wenn Sie die ELZET 80 DIN-TAST als Chassis kaufen oder nachträglich modifizieren wollen, finden Sie die Anschlüsse zwischen Hauptfeld und Funktionstasten. Sie sind auf der Lötseite als Leiterbahnfinger ausgeführt und nummeriert.

Serieller Anschluß

Der TTL-Signalausgang ist durch einen Bustreiber 74LS368 gepuffert. Diese Übertragungsart ist für kurze und mittlere Kabelwege möglich. Wo die Hardware des empfangenden Rechners es zuläßt, z.B. also bei allen ELZET 80 Computern, sollte jedoch ein getriebener Ausgang Verwendung finden. 20mA TTY wird als Standard angeboten, ein RS232-Ausgangspegel ist gegen Aufpreis verfügbar, wenn die Hilfsspannungen zugeführt werden.

Die seriellen Schnittstellen sind herausgeführt über der ESC-Taste. Die Zuordnung ist wie folgt:

- 1 Reserve (z.B. für RESET oder Signalgeber)
- 2 RS232 Signalausgang
- 3 Steuereingang, unbeschaltet für Normalbetrieb, TTL-"Low" für Anhalten der Datenübertragung.
- 4 TTL-Signalausgang
- 5 20mA TTY Signalausgang
- 6 Betriebsspannung +5V
- 7 Gemeinsame Masse
- 8 +12V bis +15V
- 9 -12V bis -15V

Die Form der Übertragung, also Baudrate, Anzahl der Datenbits und Paritätsbit, wird über Jumper unter der Leertaste festgelegt (siehe Belegung auf Seite 9). Alle Jumper offen wählt 9600 Baud mit 8 Bit Datenwort ohne Paritätsbit.

## ELZET 80 DIN-TAST

---

### Parallelanschluß

ELZET 80 TAST hat einen 8-Bit parallelen ASCII-Ausgang, dessen Anschlüsse über der CAPS-Taste liegen und mit 1 bis 13 bezeichnet sind.

1	Masse	
2	Betriebsspannung +5V	
3	Datenbit 3	Die Daten sind
4	Datenbit 4	spätestens 30ns
5	Datenbit 2	nach der ersten
6	Datenbit 5	Strobe-Flanke
7	Datenbit 1	gültig.
8	Datenbit 6	
9	Datenbit 7	
10	Datenbit 0	
11	Strobe (positiv)	Pulsbreite 700ns
12	/Strobe (negativ)	Pulsbreite 700ns
13	Datenbusfreigabe	

Die Daten werden in einem 8-Bit-Speicher (3-State, 74LS373) gepuffert, der für hohe kapazitive Last (lange Kabel) geeignet ist und sind bis zum Beginn des nächsten Übernahme-signals (Strobe bzw. /Strobe, über 74LS368 Bustreiber) gültig. In eng gekoppelten Systemen kann der Ausgangs-speicher direkt auf den Datenbus des Hauptprozessors gelegt werden, ein I/O-Chipselect muß dann mit 13 verbunden werden. Strobe könnte z.B. als Interrupt verwendet werden.

## ELZET 80 DIN-TAST

---

### Belegung der Anschlußkabel

Wenn Sie die Tastatur zu einem ELZET 80 -Computer erwerben, ist ein ca. 1,50 langes Kabel bereits angeschlossen. Bei serieller 20mA-TTY-Schnittstelle endet es in einem 3-poligen Klinkenstecker mit 3,5mm Durchmesser, der wie folgt beschaltet ist:

Spitze:	+5V	Aderfarbe braun
Mittelstück:	Signal	Aderfarbe grün
Schaft:	Masse	Aderfarbe weiß

Während des Einsteckvorgangs können Spitze und Mittelstück kurzgeschlossen werden. Wenn der Signaleingang direkt auf die LED eines Optokopplers geht, führt dies zur Zerstörung des Kopplers. Alte Versionen der CPU/IEC waren z.B. ohne Schutzbeschaltung. Bei Verwendung des ansonsten wegen seiner Kleinheit gut geeigneten Klinkensteckers ist daher ein Serienwiderstand von etwa 100 Ohm zwischen Buchse und Optokopplereingang vorzusehen.

Bei seriellem TTL-Ausgang oder bei RS232-Ausgang bleibt das Kabelende offen. Die Aderfarben sind dann wie folgt belegt:

Aderfarbe braun:	+5V
Aderfarbe grün :	Signal
Aderfarbe weiß :	Masse
Aderfarbe gelb :	Signal RS232
Aderfarbe grau :	Hilfsspannung +12V
Aderfarbe blau :	Hilfsspannung -12V
Aderfarbe rosa :	Reserve

## ELZET 80 DIN-TAST

---

Ebenfalls ohne Stecker wird die Parallelversion der Tastatur ausgeliefert. Hier ist die Zuordnung von Aderfarbe zu Signal wie folgt:

Signal	Farbe
Masse	schwarz
+ 5 V	rot
Datenbit 0	weiß
Datenbit 1	lila
Datenbit 2	grün
Datenbit 3	braun
Datenbit 4	rosa
Datenbit 5	grau
Datenbit 6	gelb
Datenbit 7	blau
Strobe	weiß/braun
/Strobe	weiß/grün

ELZET 80 DIN-TAST

---

Steckbrücken (Jumper)

Die zur Einstellung der Betriebsart nötigen Schalter sind als Steckbrücken ausgeführt. Sie befinden sich unter der Leertaste, links zwischen der linken Tastenführung und der integrierten Schaltung 74LS541 und sind daher ohne Öffnen des Gehäuses zugänglich.

=====

( J 9 ) Automatische Tastenwiederholung aus

=====

( J10 ) Automatische Tastenwiederholung langsam

=====

( J 1 ) Baudrate 1

=====

( J 5 ) Parität ungerade (nur wenn J4 gesteckt)

=====

( J 2 ) Baudrate 2

=====

X ( J 3 ) Betriebsart-Auswahl 1

=====

( J 6 ) Nur 7 Datenbits

=====

X ( J 7 ) Betriebsart-Auswahl 2

=====

( J 4 ) Paritätserzeugung ein

=====

X ( J 8 ) Betriebsart-Auswahl 3

=====

ELZET 80 DIN-TAST

---

Die Steckbrücken erlauben die Auswahl des Datenformats für die serielle Übertragung, die Einstellung der Einschalt-Betriebsart und der Wiederholrate.

Im folgenden bedeutet "1" gesteckt und "0" nicht gesteckt.

Baudrate,            J 1                    J 2  
-----

9600 Baud	0	0
4800 Baud	1	0
2400 Baud	0	1
1200 Baud	1	1

Paritätsbit,        J 5                    J 4  
-----

Keine Parität	0	0
Keine Parität	1	0
Gerade Parität	0	1
Ungerade Parität	1	1

Datenbits,            J 6  
-----

8 Datenbits	0
7 Datenbits	1

## ELZET 80 DIN-TAST

Die Einschalt-Betriebsart, d.h. die Belegung der Funktionstasten, die bei Einschalten des Geräts gültig sein soll, wird über J3, 7 und 8 festgelegt

Betriebsart,	J 3	J 7	J 8	Belegung
1 Eprom 1	1	1	1	CP/M
2 Eprom 2	0	1	1	WS einfach
3 Eprom 3	1	0	1	WS Cursor
4 Eprom 4	0	0	1	Alt-Term.
5 Prog	1	1	0	
6 ESC	0	1	0	
7 80	1	0	0	

Eine Beschreibung der Betriebsarten folgt auf den nächsten Seiten, die Standardbelegungen sind im Anhang aufgeführt.

Die automatische Wiederholfunktion ist ebenfalls über Steckbrücken wählbar.

Wiederholrate,	J 9	J10
Keine Wiederh.	1	0
Langsam	0	1
Schnell	0	0

ELZET 80 DIN-TAST

---

Für den Export oder für an amerikanische Terminals gewohnte Benutzer kann die Z/Y-Anordnung verdrahtungsmäßig vertauscht werden. Die zugehörigen Tastenkappen müssen dann ebenfalls gewechselt werden.

Neben dem Parallelanschluß befindet sich ein Löt punktfeld mit gedruckten Brücken für die Standardanordnung QWERTZ. Diese Vorgaben sind:

I-J, D-E

G-H, C-B

Für die amerikanische Anordnung QWERTY sind die Leiterbahnen zwischen den obigen Punkten aufzutrennen und durch Drahtbrücken wie folgt zu ersetzen:

C-D, A-B

H-J, G-F

### Betriebsarten

Die ELZET 80 DIN-TAST kann in sieben verschiedenen Betriebsarten arbeiten. Die Verschiedenheit bezieht sich auf den Code, der durch die Funktionstasten (P1 bis P16), die Cursorstasten (8'er Feld oben rechts) und das Nummernfeld (14'er Feld unten rechts) erzeugt wird.

In jeder Betriebsart erzeugt die Tastatur auf diesen Tasten, die im folgenden Sondertasten genannt werden, unterschiedliche Codes oder Codefolgen. Innerhalb der sieben Betriebsarten wird teilweise noch unterschieden zwischen der "Normal"-Ebene, d.h. SHIFT und CTRL nicht gedrückt, der "SHIFT"-Ebene, d.h. SHIFT (Umschalt-)Taste gedrückt, der "CONTROL"-Ebene, d.h. CTRL gedrückt und schließlich der "SHIFT/CONTROL"-Ebene, abgekürzt zu "SHCTL", die bei gleichzeitig gedrückten Tasten SHIFT und CTRL wirksam wird.

Zwischen den Betriebsarten kann jederzeit umgeschaltet werden. Die Umschaltung wird eingeleitet durch CTRL/LOCK, also durch gleichzeitiges Drücken von CTRL und LOCK. Die LED in LOCK fängt dann an zu blinken. Mit einer Ziffer zwischen 1 und 7 (im Hauptfeld) wird dann die Betriebsart gleicher Nummer ausgewählt. Die Einschalt-Betriebsart wird durch Jumper festgelegt.

Ein Schnellspeicher für max. 20 Zeichen, die im Tastatur-RAM abgelegt werden, rundet die universelle Funktion ab.

Betriebsart 1 bis 4:

Die "Eprom"-Betriebsarten

Die ersten vier Betriebsarten sind benutzerprogrammierbar, d.h. die Codes können im Tastatureprom abgelegt werden und sind so jederzeit durch den Kunden änderbar. Jede Sondertaste kann pro Ebene mit bis zu 255 Zeichen belegt werden, dabei sind alle Werte zwischen 00 und FE Hex möglich. Denkbare Anwendungen sind z.B. Befehlssequenzen für CP/M, WordStar, dBase oder andere Programme, Schlüsselwörter für BASIC, PASCAL oder Ihre bevorzugte Programmiersprache, Kurztexte für die Korrespondenz oder Steuerzeichen, die ein häufig verwendetes Programm erwartet.

In jeder der Eprom-Betriebsarten wird nochmal zwischen NORMAL, SHIFT, CTRL und SHCTL unterschieden, so daß sich insgesamt 16 kundenprogrammierbare Ebenen pro Sondertaste ergeben. Insgesamt stehen weit über 2000 Speicherstellen zur Verfügung, die mit Zeichenketten unterschiedlicher Länge gefüllt werden können.

Um Speicherplatz zu sparen, können innerhalb einer Betriebsart Ebenen untereinander zugewiesen werden. Beispielsweise kann die SHIFT-Ebene programmiert werden und die CTRL-Ebene wird der SHIFT-Ebene zugewiesen. Dann erzeugt die CTRL-Ebene die gleichen Codes, ohne daß jedoch der doppelte Speicherplatz benutzt wird.

Die Wiederholfunktion ist für jede Ebene und für jeden der drei Blöcke innerhalb der Ebene (Funktions-, Cursor- und Nummernfeld) getrennt abschaltbar. So kann z.B. die automatische Wiederholung im Cursorbereich in Funktion sein, während sie bei den anderen Blöcken abgeschaltet ist.

Da das Erstellen und Ändern der Tabellen für die Belegung etwas mühsam ist, existiert ein CP/M-Programm, der Portabilität halber in MBASIC geschrieben, das die mit einem Texteditor erstellte Tabelle in die für das Eprom nötige Binärdatei umwandelt.

Die Tabelle enthält Zuweisungen folgender Art:

P1(LC0) = "Eine feine Tastatur"

N5(UC2) = "^QC"

C7(CC3) = "\$80AA"

Dabei bezeichnen die ersten beiden Stellen die Sondertaste, der Ausdruck in Klammern die Umschaltart und die Betriebsart (UC2 ist SHIFT-Ebene in Betriebsart 3) und die Zeichenkette in Anführungszeichen den gewünschten Text. Im Text bezeichnet ein \$ einen folgenden Hex-Wert und ein ^ ein Steuerzeichen (CTRL).

Eine Detailbeschreibung finden Sie auf Seite 23 bei der Beschreibung des Tabellenassemblers.

Die vier Eprom-Betriebsarten werden werksseitig mit sinnfälligen Belegungen für CP/M und WordStar ausgeliefert.

## ELZET 80 DIN-TAST

---

Betriebsart 5:

"PROG"

Bei der Betriebsart PROG sind die Tasten P1 bis P16 und das Editierfeld C1 bis C8 während des Betriebs mit je einem beliebigen ASCII-Zeichen oder einer ESC-Sequenz mit einem beliebigen ASCII-Zeichen zu programmieren. Die Zeichen werden im Tastatur-RAM abgelegt, daher ist in der Betriebsart 5 die Schnellspeicherfunktion nicht verfügbar.

Eine Sondertaste kann nicht durch eine andere Sondertaste programmiert werden.

Ohne Programmierung sind die Tasten zwischen P1 und C8 mit CTRL/A bis CTRL/W belegt. Das Numerikfeld entspricht der Tastenkappenmarkierung. Die SHIFT- und CTRL-Tasten haben keinen Einfluß auf den Tastencode.

Die Programmierung erfolgt durch folgende Sequenz:

CTRL und LOCK drücken

> LOCK-Led blinkt

Zu programmierende Taste drücken

> Keine Ausgabe

Gewünschtes Zeichen erzeugen

> Keine Ausgabe, LOCK-Led erlischt

ELZET 80 DIN-TAST

---

Betriebsart 6:

"ESC"

In der Betriebsart "ESC" werden von den Funktions- und Cursortasten je 2 Zeichen ausgegeben: Das stets gleiche Steuerzeichen "ESC" (CTRL/Ä, 1B Hex) gefolgt von einem für jede Taste verschiedenen Zeichen.

Die Zeichenwerte steigen jeweils um eins zwischen P1 und C8. Das Numerikfeld entspricht der Tastenkappenmarkierung. Die Belegung erfolgt in 3 Ebenen: NORMAL, SHIFT und CTRL. Belegung der ESC-Sequenzen in Betriebsart 6:

Umschaltung	P1.....P16		C1.....C8	
NORMAL	ESC/00	ESC/0F	ESC/10	ESC/17
SHIFT	ESC/26	ESC/35	ESC/36	ESC/3D
CTRL	ESC/4C	ESC/5B	ESC/5C	ESC/63

Alle hinter "ESC/" angegebenen Werte verstehen sich als Hex-Werte. CTRL/P1 (ESC/4C) ist dementsprechend ESC "L". Bei SHCTL werden die Codes von SHIFT ausgegeben.

## ELZET 80 DIN-TAST

---

Betriebsart 7:

"80"

Die 80er-Betriebsart erzeugt Zeichen mit gesetztem 8.Bit, entsprechend Zeichenwerten von 80 Hex aufwärts. Sie eignet sich insbesondere für die Arbeit in einem Terminal, wo die von der Tastatur kommenden 80er Zeichen vom Terminalprozessor in Steuerfunktionen oder dynamisch veränderliche Zeichenketten umgesetzt werden.

In diese Betriebsart wird auch das Numerikfeld einbezogen, das nun nicht die markierten Zeichen abgibt. Die Tastenfolge ist zeilenweise aufsteigend, so daß der Dezimalpunkt im Numerikfeld den höchsten Wert erzeugt.

Umschaltung	P1... P16		C1....C8		N1...N14	
NORMAL	80	8F	90	97	98	A5
SHIFT	A8	B7	B8	BF	C0	CD
CTRL	D0	DF	E0	E7	E8	F5

Bei SHCTL werden Codes wie bei SHIFT generiert.

Bei Verwendung dieser Betriebsart ist darauf zu achten, daß Übertragung mit 8 Bit gewählt wird (J6 offen) und daß die Eingabeschleife des Empfangsrechners das achte Bit nicht maskiert.

### Schnellspeicher

Ein Teil des RAM-Speichers innerhalb des Tastaturprozessors wurde für eine Schnellspeicherfunktion freigehalten. Das ermöglicht, ein innerhalb einer Arbeit häufig gebrauchtes Wort oder eine Folge von Steuerzeichen durch einmaliges Eintippen beliebig oft wiederholbar zu machen.

Für diese Funktion stehen 20 Zeichen zur Verfügung, die beliebig auf die vier oberen Tasten des Cursorfelds verteilt werden können.

Die Einschaltung der Betriebsart Schnellspeicher geschieht durch Betätigung von SHIFT/CAPS, die CAPS-LED leuchtet zur Bestätigung halbhell mit leichtem Flackern. Die Programmierung erfolgt wie gewohnt durch CTRL/LOCK, woraufhin die LOCK-Led blinkt. Nun ist die zu programmierende Taste zwischen C1 und C4 zu drücken, daraufhin der zu speichernde Text einzugeben, der durch ein weiteres Drücken auf C1...C4 abgeschlossen wird. Hört die LOCK-Led auf zu blinken, bevor der Text beendet wurde, so ist die Speicherkapazität überschritten.

Der eben gespeicherte Text ist nun solange abrufbar, bis er geändert wird oder bis die Betriebsspannung der Tastatur abgeschaltet wird. Zwischen der Standardbelegung von C1...C4 und dem Schnellspeicher kann durch SHIFT/CAPS beliebig hin- und hergeschaltet werden. Durch Neuprogrammierung ohne Texteingabe wird der Speicherplatz wieder frei.

Erstellung kundenspezifischer Belegungen

Die Tabelle mit den Einträgen der Benutzerbetriebsarten EPROM1 bis EPROM4 ist im Betriebseprom des Tastaturprozessors hinter dem eigentlichen Programmcode untergebracht, für den ca. 700 (Hex) Byte benötigt werden.

Für jedes Tastenelement ist ein Eintrag vorgesehen, der aus der Anzahl der Zeichen und den Zeichen selbst besteht. Um die Belegung z.B. der Sondertaste P6 ausgeben zu können, springt der Prozessor zunächst auf die Speicherstelle von P1, liest das Byte ein, das die Zeichenmenge von P1 bestimmt, addiert das zur aktuellen Position, erreicht somit den Anfang von P2 usw., bis er bei P5 die Anzahl Zeichen einliest und entsprechend ausgibt.

Die Erstellung der Tabelle kann theoretisch mit einem Dienstprogramm wie DDT oder einem Monitor vorgenommen werden, allerdings ist dies recht zeitraubend. Wir bieten daher ein in BASIC geschriebenes Programm an, das dieser Broschüre als Anlage beigelegt ist. Es erzeugt aus einer Klartextdatei, die mit einem Editor erstellt wird, eine Binärdatei, die mit dem Grundprogramm verknüpft werden kann.

Das Programm ist neuen ELZET 80 -Computern auf der CP/M-Systemdiskette beigelegt, Besitzer von Fremdgeräten können das Quellprogramm auf CP/M Standard-Diskette 8" einfache Dichte zum Preis von DM 25,- erwerben.

Tabellenerzeugungsprogramm TAB

Das Basicprogramm TAB, das als Source-Datei geliefert wird, gestattet es, aus Textdateien binäre Tabellen für den Einsatz in der ELZET-80 Tastatur zu erstellen. Es können die Funktionstasten, die Cursorstasten und die Tasten des Numerikfeldes mit Zeichenketten belegt werden.

Das Programm kann unter CP/M mit MBASIC interpretiert oder mit BASCOM compiliert werden. Bei häufigem Einsatz empfiehlt sich die Compilierung. Bei Verwendung unter anderen Basic-Dialekten sind die Eingabe- und Ausgabefunktionen und Teile der Stringfunktionen eventuell abzuändern.

Das Programm TAB liest immer eine Textdatei ein (erster Abfragetext) und generiert eine Binärdatei. Die Binärdatei kann dann ab Adresse 0700H in ein Tastatureprom der ELZET-80 Tastatur programmiert werden ( Siehe Submitfile-Beispiel am Ende ). Bis Adresse 0700H ist unbedingt der bisherige Eprominhalt zu übernehmen.

Bevor sie das TAB Programm ablaufen lassen können, ist es erforderlich, eine Textdatei zu erstellen. Dies kann mit jedem CP/M Editor geschehen. Im folgenden werden die einzelnen Elemente dieser Textdatei beschrieben.

- (1) Belegen einer der Sondertasten mit Text oder Steuerzeichen.

Mit der folgenden Programmzeile bestimmen Sie den Text, der auf eine Sondertaste der Elzet Tastatur gelegt werden soll:

```
P11(UC0)="Dies ist ein Text"
```

Leer- oder Tabulationszeichen dürfen in die Zeile eingefügt werden. Am Zeilenende kann ein, durch ein ';' -Zeichen abgetrennter, Kommentar stehen. Nun im Einzelnen zu den Teilen der Beipiels-Befehlssequenz. Mit 'P11' wird die zu belegende Taste gekennzeichnet. Hier können die folgenden Namen stehen:

P1	-	P16	Funktionstasten entspr. Namens
C1	-	C8	Cursortasten (oben rechts)
N1	-	N14	Tasten des Numerikfeldes

Beachten Sie, daß alle Tasten von links nach rechts und von oben nach unten durchnummeriert werden.

Nach der Festlegung der Taste wird ihre Betriebsebene angegeben. Dazu haben Sie die folgenden Möglichkeiten:

LC	(NORMAL)	weder SHIFT noch CTRL betätigt
UC	(SHIFT)	SHIFT betätigt, CTRL nicht betätigt
CC	(CTRL)	CTRL betätigt, SHIFT nicht betätigt
SC	(SHCTL)	Sowohl SHIFT als auch CTRL betätigt

Damit können auf jede Taste bis zu vier verschiedene Zeichenketten gelegt werden. Nach den zwei Buchstaben für die Ebene folgt eine Ziffer im Bereich 0 bis 3, die angibt, welche Betriebsart (vier mögliche Anwender-Betriebsarten) programmiert werden soll. Sie haben also beispielsweise die Möglichkeit, die Betriebsart 0 für Basic-Schlüsselworte zu verwenden, während Sie die Betriebsart 3 mit Wordstar-Steuersequenzen belegen. Beachten Sie, daß die Betriebsartangabe immer in Klammern gesetzt werden muß. Das zweimalige Programmieren der gleichen Taste in der gleichen Betriebsart ist nicht zulässig und führt zu einer Fehlermeldung.

Nachdem die Taste exakt definiert ist, können Sie nun die gewünschte Zeichenkette eingeben. Starten Sie mit der Zeichenfolge '='. Geben Sie dann eine Zeichenkette ein. Bei der Interpretation der Zeichenkette, die Sie erneut mit einem '=' Zeichen abschließen müssen, haben einige Zeichen eine spezielle Bedeutung.

- § - Es folgen zwei Hexziffern, mit denen ein beliebiges Ausgabebyte spezifiziert werden kann (Ausnahme OFFH). Soll ein echtes '§'-Zeichen in Ihrem Text erscheinen, so muß es zweifach angegeben werden.
  
- ^ - Das folgende Zeichen wird als 'Control'-Zeichen interpretiert. Die Zeichensequenz '^C' bewirkt also die Ausgabe des Bytes 03H.

Soll das Symbol '^' als Textzeichen ausgegeben werden, so ist es doppelt anzugeben.

Zusätzlich gilt die Konvention, daß das '"' Zeichen als Teil einer Zeichenkette zweifach anzugeben ist, um es von seiner Funktion als Stringbegrenzer zu unterscheiden.

Nachstehend finden Sie einige Zeichenketten und die zugeordneten Ausgabebytes der Tastatur.

Definition	Ausgabebytes
"AAA"	41H 41H 41H
"^C^A"	03H 01H
"\$80AA\$\$"	80H 41H 41H 24H

(2) Kommentarzeilen

Fangen Zeilen mit einem ';' -Zeichen (Semikolon) an, so werden sie beim Einlesen übersprungen. Der Rest der Zeile kann also mit beliebigen Kommentaren versehen sein.

(3) Festlegen der Wiederholfunktion

Zeilen, die mit einem 'M' -Zeichen, gefolgt von einer in Klammern stehenden Betriebsartangabe gefolgt werden, schalten die Wiederholfunktion für alle Tastengruppen der jeweiligen Betriebsart an oder aus.

Beispiel Wiederholfunktion:

$M(LC0)=110$

Als Argument dieser Funktion folgt eine Zeichenkette aus drei Ziffern (nur '0' und '1' zulässig). Die Positionen entsprechen von links nach rechts den Tastengruppen 'Funktionstasten-Cursorstasten-Numerikfeld'. Eine '1' schaltet die Wiederholfunktion für die jeweilige Gruppe ein, eine '0' schaltet sie aus.

(4) Äquivalenz zwischen Funktionsebenen

Einzelne Funktionsebenen lassen sich einander zuordnen. Der Befehl

$E(LC0)=(UC0)$

bedingt, daß in Betriebsart 0 alle 'UC' und 'LC' Eingaben von Sondertasten die gleichen Zeichenketten aussenden. Damit kann man erreichen, daß die Tastatur keinen Unterschied zwischen 'LC' und 'UC' Bedienungen von Sondertasten macht. Der Vorteil dieser Betriebsart ist, daß kein zusätzlicher Speicherplatz benötigt wird. Allerdings können nur ganze Betriebsarten einander zugeordnet werden. Voraussetzung ist weiterhin, daß die zuerst im Kommando spezifizierte Betriebsart noch keine belegten Tasten hat. Ist diese Bedingung nicht erfüllt, so erfolgt eine Fehlermeldung.

(5) Fehlermeldungen

Alle Fehlermeldungen erfolgen im Klartext und werden deshalb an dieser Stelle nicht beschrieben. In einigen Fällen wird die jeweilige fehlerhafte Zeile und die ungefähre Position des Fehlers angezeigt.

(6) Beispiel

Nachstehend ist die Liste eines kurzen Beispiel gezeigt, mit dessen Hilfe einige Tasten der ELZET-80 Tastatur programmiert werden können.

```
;
; Testdatei fuer TAB
; -----
```

```
P16(LC0) = "DIES IST EIN TEST" ; P16 mit ASCII Text
C7(UC0) = "^X^Y$81" ; Cursor 7 Steuerz.
N5(CC3) = "$$$^" ; Numerik 5 = $^

M(CC3) = 001 ; Repeat belegen

E(lc1) = (lc0) ; zwei Ebenen identisch
```

(7) Submit-Beispiel

XSUB  
DDT KEYGRUND.BIN           (fester Teil des Tastaturprogramms)  
IKEYTABLE.BIN           (von TAB generierter User-Teil)  
R700  
GO  
SAVE 16 KEY.BIN

Tastenkappen

=====

Die Firma SIEMENS als Hersteller der Tastenkappen verfügt über ein reichhaltiges Angebot an Werkzeugen für spezielle Beschriftungen. Wir können spezielle Tastenkappen ab einer Stückzahl von ca. 50 Stück anfertigen lassen, allerdings sind derzeit einige Monate Lieferzeit zu erwarten. Eine Liste mit den Zeichnungen der vorhandenen Werkzeuge kann bei uns gegen eine Gebühr von DM 10,- angefordert werden.

Tastenkappen mit Klarsichtdeckeln sind bei uns erhältlich in quadratischer Form und mit elliptischem Deckel. Die Preise bitten wir zu erfragen.

A N H A N G

Quellprogramm TAB

Tabellenassembler für ELZET 80 DIN-TAST

Beispiel-Textdatei

Standardbelegung der EPROM-Betriebsarten

```

1  REM TABELLENGEM ING ELZET TASTATUR
2  REM
3  REM 31.5.85
4  REM Programmanpassung
5  REM
6  REM
7  REM In der vorliegenden Version ist das Programm fuer MBASIC unter
8  REM CP/M geschrieben.
9  REM
10 REM Tabellenformat
11 REM
12 REM
13 REM Insgesamt werden 2048+256 Byte an Tabelle generiert Sie sind
14 REM in Tastaturepron ab der Adresse 0700H einzuubinden. Auf allen
15 REM durch 256 teilbaren Adressen wird automatisch die Sequenz 0A3H,
16 REM 0B3H eingefuegt. Die ersten 32 Byte sind Zeiger auf die einzel-
17 REM nen Startbereiche der Stringketten. Alle Stringketten bestehen
18 REM aus je 38 Eintraegen.
19 REM
20 REM
21 REM Variablenvereinbarungen
22 REM
23 REM
24 REM DIM ROWZ(2303)
25 REM
26 REM Fehlerbehandlung
27 REM
28 REM
29 REM ON ERROR GOTO 31
30 REM GOTO 32
31 REM ERTYPZ=9:GOTO 71
32 REM
33 REM Hauptprogramm
34 REM
35 REM
36 REM *-----*
37 REM *Tastatur Tabellenerzeugung Version 1.1*
38 REM *(C) 1984 by ELZET 80 GmbH*
39 REM *-----*
40 REM
41 REM GOSUB 338 : REM Initialisiere rowZ
42 REM INPUT "Eingabedatei --> ",EINGABES$
43 REM FOR IZ=1 TO LEN(EINGABES$)
44 REM IF MID$(EINGABES$,IZ,1)<"a" OR MID$(EINGABES$,IZ,1)>"z" THEN GOTO 46
45 REM MID$(EINGABES$,IZ,1)=CHR$(ASC(MID$(EINGABES$,IZ,1))-32)
46 REM NEXT IZ
47 REM INPUT "ROM-Datei --> ",AUSGABES$
48 REM FOR IZ=1 TO LEN(AUSGABES$)
49 REM IF MID$(AUSGABES$,IZ,1)<"a" OR MID$(AUSGABES$,IZ,1)>"z" THEN GOTO 51
50 REM MID$(AUSGABES$,IZ,1)=CHR$(ASC(MID$(AUSGABES$,IZ,1))-32)
51 REM NEXT IZ
52 REM OPEN "0",#2,AUSGABES$
53 REM OPEN "1",#1,EINGABES$
54 REM PRINT
55 REM IF EOF(1) GOTO 62
56 REM LINE INPUT #1,EZEILE$
57 REM PRINT EZEILE$
58 REM EZEILE$=EZEILE$+CHR$(0)
59 REM IF MID$(EZEILE$,1,1)=";" OR MID$(EZEILE$,1,1)=CHR$(0) THEN GOTO 55
60 REM GOSUB 92 : REM Analyse der Eingabezeile
61 REM GOTO 55
62 REM FOR IZ=0 TO 2303 STEP 1
63 REM PRINT #2,CHR$(ROWZ(IZ));
64 REM NEXT IZ
65 REM CLOSE #2
66 REM GOTO 353
67 REM
68 REM Fehlerausgaben
69 REM
70 REM
71 REM PRINT:PRINT
72 REM ON ERTYPZ GOTO 74,75,76,83,84,85,86,87,88,89,90,91
73 REM PRINT "###Illegale Fehlermeldung###:GOTO 353
74 REM PRINT "###Eingabedatei nicht zu offnen###:GOTO 353
75 REM PRINT "###Ausgabedatei nicht zu offnen###:GOTO 353
76 REM PRINT "###Illegale Eingabezeile###"
77 REM PRINT STRING$(79,"-")
78 REM PRINT EZEILE$
79 REM FOR IZ=1 TO PSZ-2:PRINT " ";:NEXT IZ
80 REM PRINT ""
81 REM PRINT STRING$(79,"-")
82 REM GOTO 353
83 REM PRINT "###Taste bereits definiert###:GOTO 353
84 REM PRINT "###Speicherueberlauf###:GOTO 353
85 REM PRINT "###Interner Fehler###:GOTO 353
86 REM PRINT "###Illegales Steuerzeichen###:GOTO 353
87 REM PRINT "###Falsches HEX Digit###:GOTO 353
88 REM PRINT "###Interner Basic Fehler (Datei ueberpruefen)###:GOTO 353

```

```

89 PRINT "###Zeichenkette zu lang###:GOTO 353
90 PRINT "###Zuord auf belegte Betriebsart###:GOTO 353
91 PRINT "###Illeg (zeilenende###:GOTO 353
92 REN
93 REN Analyse einer Eingabezeile
94 REN
95 REN
96 REN Die Eingabezeile in zeile# wird in eine Zeichenkette unge-
97 REN setzt. Die Variable MODEX,KEYZ,LEVELZ werden entsprechend
98 REN gesetzt.
99 REN
100 PSZ=1
101 GOSUB 175
102 IF CH# (<) "P" THEN GOTO 112
103 GOSUB 175
104 GOSUB 187
105 IF IX < 1 OR IX > 16 THEN ERTYPZ=3:GOTO 67
106 KEYZ=IX-1
107 GOSUB 195
108 GOSUB 215
109 GOSUB 257
110 GOSUB 264
111 RETURN
112 IF CH# (<) "E" THEN GOTO 126
113 GOSUB 175
114 GOSUB 195
115 ENDEX=MODEX:ELEVEZ=LEVELZ
116 IF CH#(>)=" THEN ERTYPZ=3:GOTO 71
117 GOSUB 175
118 GOSUB 195
119 GOSUB 257
120 IDIOFZ=2+ELEVELZ*8+EMDEX*2
121 IF (ROMZ(IDIOFZ) AND &H1F)<>0 OR ROMZ(IDIOFZ+1)<>34 THEN ERTYPZ=11:GOTO 71
122 IDIPGZ=2+LEVELZ*8+MODEX*2
123 ROMZ(IDIOFZ)=ROMZ(IDIPGZ)
124 ROMZ(IDIOFZ+1)=ROMZ(IDIPGZ+1)
125 RETURN
126 IF CH# (<) "C" THEN GOTO 136
127 GOSUB 175
128 GOSUB 187
129 IF IX < 1 OR IX > 8 THEN ERTYPZ=3:GOTO 67
130 KEYZ=15+IX
131 GOSUB 195
132 GOSUB 215
133 GOSUB 257
134 GOSUB 264
135 RETURN
136 IF CH#(>)"M" THEN GOTO 157
137 GOSUB 175
138 GOSUB 195
139 IF CH#(>)=" THEN ERTYPZ=3:GOTO 71
140 GOSUB 175
141 MSKZ=224
142 IF CH#="1" THEN MSKZ=MSKZ AND &H7F:GOTO 145
143 IF CH#="0" THEN GOTO 145
144 ERTYPZ=3:GOTO 71
145 GOSUB 175
146 IF CH#="1" THEN MSKZ=MSKZ AND &HBF:GOTO 149
147 IF CH#="0" THEN GOTO 149
148 ERTYPZ=3:GOTO 71
149 GOSUB 175
150 IF CH#="1" THEN MSKZ=MSKZ AND &HDF:GOTO 153
151 IF CH#="0" THEN GOTO 153
152 ERTYPZ=3:GOTO 71
153 GOSUB 175:GOSUB 257
154 IDIOFZ=2+LEVELZ*8+MODEX*2
155 ROMZ(IDIOFZ)=(ROMZ(IDIOFZ) AND &H1F) OR MSKZ
156 RETURN
157 IF CH# (<) "M" THEN ERTYPZ=3 : GOTO 67
158 GOSUB 175
159 GOSUB 187
160 IF IX < 1 OR IX > 14 THEN ERTYPZ=3:GOTO 67
161 KEYZ=23+IX
162 GOSUB 195
163 GOSUB 215
164 GOSUB 257
165 GOSUB 264
166 RETURN
167 REN
168 REN Lese naechstes Zeichen nach CH#
169 REN -----
170 REN
171 IF PSZ > LEN(EZEILE#) THEN ERTYPZ=3 : GOTO 67
172 CH# = MID$(EZEILE#,PSZ,1)
173 PSZ=PSZ+1
174 RETURN
175 REN
176 REN Lese naechstes Zeichen nach CH# (Grosschrift)

```

```

177 REM -----
178 REM
179 GOSUB 167
180 IF CH#=" " OR CH#-CHR$(9) THEN GOTO 179
181 IF CH#>="a" AND CH#<="z" THEN CH#-CHR$(ASC(CH#)-32)
182 RETURN
183 REM
184 REM Lese Zahl ein
185 REM -----
186 REM
187 IX=0
188 IF CH# >= "0" AND CH# <= "9" THEN IX=10*IX+ASC(CH#)-ASC("0") ELSE RETURN
189 GOSUB 175
190 GOTO 188
191 REM
192 REM Lese ID in Klammern ein
193 REM -----
194 REM
195 IF CH# < > "(" THEN ERTYPZ=3 : GOTO 67
196 GOSUB 175
197 CASE#-CH#
198 GOSUB 175
199 CASE#-CASE#+CH#
200 GOSUB 175
201 IF CASE# = "SC" THEN MDEZ=0 : GOTO 206
202 IF CASE# = "CC" THEN MDEZ=1 : GOTO 206
203 IF CASE# = "UC" THEN MDEZ=2 : GOTO 206
204 IF CASE# = "LC" THEN MDEZ=3 : GOTO 206
205 ERTYPZ=3 : GOTO 67
206 LEVELI-ASC(CH#)-ASC("0")
207 IF LEVELI<0 OR LEVELI>3 THEN ERTYPZ=3 : GOTO 67
208 GOSUB 175 : IF CH# < > ")" THEN ERTYPZ=3 : GOTO 67
209 GOSUB 175
210 RETURN
211 REM
212 REM Lese Zeichenkette in ''' nach STR6#
213 REM -----
214 REM
215 SMDZ=0
216 STR6#=""
217 REM Erstes Zeichen muss Apostroph sein
218 GOSUB 175
219 IF CH# < > CHR$(34) THEN ERTYPZ=3:GOTO 67
220 REM Zeichenleseschleife
221 IF LEN(STR6#)>254 THEN ERTYPZ=10:GOTO 71
222 GOSUB 167
223 REM Normalzeichen
224 IF SMDZ < > 0 THEN GOTO 231
225 IF CH# = "" THEN SMDZ=1:GOTO 220
226 IF CH# = "$" THEN SMDZ=2:GOTO 220
227 IF CH# = CHR$(34) THEN SMDZ=4:GOTO 220
228 IF ASC(CH#) < 32 OR ASC(CH#) > 127 THEN ERTYPZ=7:GOTO 67
229 STR6#-STR6#+CH#
230 GOTO 220
231 REM Control Zeichen
232 IF SMDZ < > 1 THEN GOTO 237
233 IF CH# = "" THEN STR6#-STR6#+CH#:SMDZ=0:GOTO 220
234 STR6#-STR6#+CHR$(ASC(CH#) AND &H1F)
235 SMDZ=0
236 GOTO 220
237 REM Erstes Hex-Digit
238 IF SMDZ < > 2 THEN GOTO 245
239 IF CH# = "$" THEN STR6#-STR6#+CH#:SMDZ=0:GOTO 220
240 IF CH#>="0" AND CH#<="9" THEN HIZ=ASC(CH#)-ASC("0"):GOTO 243
241 IF CH#>="A" AND CH#<="F" THEN HIZ=ASC(CH#)-ASC("A")+10:GOTO 243
242 ERTYPZ=8:GOTO 67
243 SMDZ=3
244 GOTO 220
245 REM Zweites Hex-Digit
246 IF SMDZ < > 3 THEN GOTO 253
247 IF CH#>="0" AND CH#<="9" THEN HIZ=HIZ*16+ASC(CH#)-ASC("0"):GOTO 250
248 IF CH#>="A" AND CH#<="F" THEN HIZ=HIZ*16+ASC(CH#)-ASC("A")+10:GOTO 250
249 ERTYPZ=8:GOTO 67
250 STR6#-STR6#+CHR$(HIZ)
251 SMDZ=0
252 GOTO 220
253 REM Stringdelimitier
254 IF SMDZ < > 4 THEN ERTYPZ=6:GOTO 67
255 IF CH# = CHR$(34) THEN STR6#-STR6#+CH#:GOTO 220
256 RETURN
257 REM
258 REM Test auf korrektes Zeilenende
259 REM -----
260 REM
261 IF CH#=" " OR CH#-CHR$(9) THEN GOSUB 175
262 IF CH#=":" OR CH#-CHR$(0) THEN RETURN
263 ERTYPZ=12:GOTO 71
264 REM

```

```

265 REM Einfuegen ei 'trings an geeigneter Stelle
266 REM -----
267 REM
268 GOSUB 279:REM Unterprogramm LOCATE
269 REM Einfuegen des STRG# ab IDXZ
270 IF ROMZ(1DIPGZ+256+IDIOFZ) (<) 0 THEN ERTYPZ=4:GOTO 67
271 GOSUB 310:REM Unterprogramm MOVE
272 ROMZ(1DIPGZ+256+IDIOFZ)=LEN(STRG#)
273 IF LEN(STRG#)=0 THEN GOTO 278
274 FOR IZ=1 TO LEN(STRG#)
275 IDIOFZ=IDIOFZ+1:IF IDIOFZ=256 THEN IDIOFZ=2:IDIPGZ=IDIPGZ+1
276 ROMZ(1DIPGZ+256+IDIOFZ)=ASC(MID$(STRG#,IZ,1))
277 NEXT IZ
278 RETURN
279 REM
280 REM Unterprogramm LOCATE
281 REM -----
282 REM
283 IDIOFZ=2+LEVELZ+8+MDEZ+2
284 PGZ=ROMZ(IDIOFZ) AND &H1F
285 OFZ=ROMZ(IDIOFZ+1)
286 REM Teste Startbelegung
287 IF (PGZ AND &H1F) = 0 AND OFZ=34 THEN GOTO 297
288 REM Suche den String
289 IF KEYZ=0 THEN GOTO 294
290 FOR IZ=1 TO KEYZ
291 LNZ=ROMZ(PGZ+256+OFZ)+1
292 OFZ=OFZ+LNZ:IF OFZ=256 THEN OFZ=OFZ MOD 256+2:PGZ=PGZ+1
293 NEXT IZ
294 IDIPGZ=PGZ
295 IDIOFZ=OFZ
296 RETURN
297 REM mache Ende der Daten zu neuen Anfang
298 ROMZ(IDIOFZ)=ROMZ(IDIOFZ) OR FIMPZ
299 ROMZ(IDIOFZ+1)=FIMPZ
300 IF FIMPZ+KEYZ=256 THEN GOTO 303
301 IDIOFZ=FIMPZ+KEYZ:IDIPGZ=FIMPZ
302 GOTO 304
303 IDIOFZ=(FIMPZ+KEYZ) MOD 256+2:IDIPGZ=FIMPZ+1
304 IF FIMPZ+38 >= 256 THEN GOTO 307
305 FIMPZ=FIMPZ+38
306 GOTO 309
307 FIMPZ=(FIMPZ+38) MOD 256+2
308 FIMPZ=FIMPZ+1
309 RETURN
310 REM
311 REM Unterprogramm MOVE
312 REM -----
313 REM
314 LNZ=LEN(STRG#)
315 IF LNZ=0 THEN RETURN
316 SOFZ=FIMPZ
317 SPGZ=FIMPZ
318 DPZ=SPGZ
319 DOFZ=SOFZ+LNZ:IF DOFZ=256 THEN DOFZ=DOFZ MOD 256+2:DPZ=DPZ+1
320 IF DPZ+256+DOFZ >= 2304 THEN ERTYPZ=5:GOTO 67
321 DELT=(SPGZ+256+SOFZ)-(1DIPGZ+256+IDIOFZ)
322 FOR IZ=1 TO DELT
323 ROMZ(DPZ+256+DOFZ)=ROMZ(SPGZ+256+SOFZ)
324 SOFZ=SOFZ-1:IF SOFZ<2 THEN SOFZ=255:SPGZ=SPGZ-1
325 DOFZ=DOFZ-1:IF DOFZ<2 THEN DOFZ=255:DPZ=DPZ-1
326 NEXT IZ
327 FOR IZ=2 TO 34 STEP 2
328 IF (ROMZ(IZ) AND &H1F)+256+ROMZ(IZ+1)<=1DIPGZ+256+IDIOFZ THEN GOTO 334
329 IF ROMZ(IZ+1) <=IDIOFZ THEN GOTO 334
330 ROMZ(IZ+1)=ROMZ(IZ)+LNZ
331 IF ROMZ(IZ+1)<256 THEN GOTO 334
332 ROMZ(IZ+1)=ROMZ(IZ+1) MOD 256+2
333 ROMZ(IZ)=(ROMZ(IZ) AND &H1F)+1 OR (ROMZ(IZ) AND &HE0)
334 NEXT IZ
335 FIMPZ=FIMPZ+LNZ
336 IF FIMPZ=256 THEN FIMPZ=FIMPZ MOD 256+2:FIMPZ=FIMPZ+1
337 RETURN
338 REM
339 REM Initialisieren von ROMZ
340 REM -----
341 REM
342 FOR IZ=0 TO 2303
343 IF IZ MOD 256=0 THEN ROMZ(IZ)=&HA3:IZ=IZ+1:ROMZ(IZ)=&HB3:GOTO 345
344 IF IZ<34 THEN ROMZ(IZ)=224 : IZ=IZ+1 : ROMZ(IZ)=34 ELSE ROMZ(IZ)=0
345 NEXT IZ
346 FIMPZ=0
347 FIMPZ=72
348 RETURN
349 REM
350 REM Programmende
351 REM -----
352 END

```

```

;
;
; Tabelle zum generieren eines Eproms
; fuer die ELZET 80 DIN-TAST
; mit benutzer-spezifischen Funktionstasten.
;
;

```

```

; USER 1: CP/M
; USER 2: Einfache WORDSTAR-Belegung
; USER 3: Komfortable WORDSTAR-Belegung
; USER 4: Terminalmode
;

```

```

;
; User 1
;

```

```

; Kein Unterschied bei Shift und Control
;

```

```

;
; H(LCO)=011
;

```

```

; P1(LCO)="^S"
; P2(LCO)="DIR^M"
; P3(LCO)="DIR B:^M"
; P4(LCO)="PIP^M"
; P5(LCO)="^I"
; P6(LCO)="[VOR]^M"
; P7(LCO)="FORMAT "
; P8(LCO)="SYSGEN^M"
; P9(LCO)="MAKE "
; P10(LCO)="STAT "
; P11(LCO)="DDT "
; P12(LCO)="ERA "
; P13(LCO)="SAVE "
; P14(LCO)="TYPE "
; P15(LCO)="REN "
; P16(LCO)="B^M"
;

```

```

; C1(LCO)="^A"
; C2(LCO)="^S"
; C3(LCO)="^E"
; C4(LCO)="^R"
; C5(LCO)="^C"
; C6(LCO)="^T"
; C7(LCO)="^D"
; C8(LCO)="^F"
;

```

```

; N1(LCO)="7"
; N2(LCO)="8"
; N3(LCO)="9"
; N4(LCO)="U"
; N5(LCO)="4"
; N6(LCO)="5"
; N7(LCO)="6"
; N8(LCO)="."
; N9(LCO)="1"
; N10(LCO)="2"
; N11(LCO)="3"
; N12(LCO)="^M"
; N13(LCO)="0"
; N14(LCO)=","
;

```

```

; J0=(LCO)
; E(LCO)=(LCO)
; E(SC0)=(LCO)
;

```

```

;
; User 2
;

```

```

; Kein Unterschied zwischen Shift und Control
;

```

```

;
; H(LC1)=011
;

```

```

; P1(LC1)="^KS^QP"
; P2(LC1)="^KD"
; P3(LC1)="^DX"
; P4(LC1)="^DF"
; P5(LC1)="^BA"
; P6(LC1)="^KB"
; P7(LC1)="^KY"
; P8(LC1)="^KC"
; P9(LC1)="^KV"
; P10(LC1)="^KR"
; P11(LC1)="^DL"
; P12(LC1)="^DP"
;

```

```

P13(LC1)="^B"
P14(LC1)="^Y"
P15(LC1)="^G"
P16(LC1)="^T"
;
C1(LC1)="^A"
C2(LC1)="^S"
C3(LC1)="^E"
C4(LC1)="^R"
C5(LC1)="^C"
C6(LC1)="^I"
C7(LC1)="^D"
C8(LC1)="^F"
;
N1(LC1)="7"
N2(LC1)="8"
N3(LC1)="9"
N4(LC1)="^U"
N5(LC1)="4"
N6(LC1)="5"
N7(LC1)="6"
N8(LC1)="."
N9(LC1)="1"
N10(LC1)="2"
N11(LC1)="3"
^ (LC1)="^M"
(LC1)="0"
N14(LC1)="."

```

```

;
E(UC1)=(LC1)
E(CC1)=(LC1)
E(SC1)=(LC1)
;

```

-----

User 3

weder Shift noch Control

-----

```

;
N(LC2)=001
;

```

```

P1(LC2)="^OB"
P2(LC2)="^OX"
P3(LC2)="^OI^["
P4(LC2)="^OI"
P5(LC2)="^OL^["
P6(LC2)="^OR^["
P7(LC2)="^OF"
P8(LC2)="^L"
P9(LC2)="^OE-^OE"
P10(LC2)="^OJ"
P11(LC2)="^M"
P12(LC2)="^PS"
P13(LC2)="^PD"
P14(LC2)="^PT"
P15(LC2)="^B"
P16(LC2)="^T"
;

```

```

C1(LC2)="^KS^QP"
^ (LC2)="^KD"
^ (LC2)="^KC"
^ (LC2)="^KR"
C5(LC2)="^QC"
C6(LC2)="^K1"
C7(LC2)="^KB"
C8(LC2)="^KX"
;

```

```

N1(LC2)="^BS"
N2(LC2)="^E"
N3(LC2)="^BD"
N4(LC2)="^R"
N5(LC2)="^S"
N6(LC2)="^V"
N7(LC2)="^D"
N8(LC2)="^C"
N9(LC2)="^A"
N10(LC2)="^Y"
N11(LC2)="^E"
N12(LC2)="^N"
N13(LC2)="^B"
N14(LC2)="."
;

```

User 3

Shift betaetigt, Control nicht

M(UC2)=001

P1(UC2)="^DC"  
 P2(UC2)="^DF"  
 P3(UC2)="^DM^C"  
 P4(UC2)="^DM^H"  
 P5(UC2)="^DL"  
 P6(UC2)="^DR"  
 P7(UC2)="^DA"  
 P8(UC2)="^DT"  
 P9(UC2)="^DD"  
 P10(UC2)="^DM"  
 P11(UC2)="^Z"  
 P12(UC2)="^P0"  
 P13(UC2)="^PH"  
 P14(UC2)="^PV"  
 P15(UC2)="^Q\$7F"  
 ^4(UC2)="^QV"

^1(UC2)="^KS"  
 C2(UC2)="^KV"  
 C3(UC2)="^KW"  
 C4(UC2)="^KW"  
 C5(UC2)="^QR"  
 C6(UC2)="^Q1"  
 C7(UC2)="^QB"  
 C8(UC2)="^QK"

M1(UC2)="^QS"  
 M2(UC2)="^E"  
 M3(UC2)="^QD"  
 M4(UC2)="^R"  
 M5(UC2)="^S"  
 M6(UC2)="^V"  
 M7(UC2)="^D"  
 M8(UC2)="^C"  
 M9(UC2)="^A"  
 M10(UC2)="^X"  
 M11(UC2)="^F"  
 M12(UC2)="^P^M"  
 M13(UC2)="^DH"  
 M14(UC2)="^QV"

User 3

Control betaetigt, Shift nicht

M(CC2)=001

^C2)=(UC2)

User 3

Shift und Control betaetigt

M(SC2)=001

E(SC2)=(UC2)

User 4

weder Shift noch Control

M(LC3)=111

P:(LC3)="^\$B0"

P2(LC3)="\$81"  
 P3(LC3)="\$82"  
 P4(LC3)="\$83"  
 P5(LC3)="\$84"  
 P6(LC3)="\$85"  
 P7(LC3)="\$86"  
 P8(LC3)="\$87"  
 P9(LC3)="\$88"  
 P10(LC3)="\$89"  
 P11(LC3)="\$8A"  
 P12(LC3)="\$8B"  
 P13(LC3)="\$8C"  
 P14(LC3)="\$8D"  
 P15(LC3)="\$8E"  
 P16(LC3)="\$8F"

C1(LC3)="\$90"  
 C2(LC3)="\$91"  
 C3(LC3)="\$92"  
 C4(LC3)="\$93"  
 C5(LC3)="\$94"  
 C6(LC3)="\$95"  
 C7(LC3)="\$96"  
 C8(LC3)="\$97"

(LC3)="\$9D"  
 (LC3)="\$9E"  
 M3(LC3)="\$9F"  
 M4(LC3)="\$A0"  
 M5(LC3)="\$A1"  
 M6(LC3)="\$A2"  
 M7(LC3)="\$A3"  
 M8(LC3)="\$A4"  
 M9(LC3)="\$A5"  
 M10(LC3)="\$A6"  
 M11(LC3)="\$A7"  
 M12(LC3)="\$AA"  
 M13(LC3)="\$AB"  
 M14(LC3)="\$A9"

-----  
 User 4

Shift betaetigt, Control nicht  
 -----

M(UC3)=111

E(UC3)=(LC3)  
 -----

User 4

Control betaetigt, Shift nicht  
 -----

M(UC3)=111

P1(CC3)="\$C0"  
 P2(CC3)="\$C1"  
 P3(CC3)="\$C2"  
 P4(CC3)="\$C3"  
 P5(CC3)="\$C4"  
 P6(CC3)="\$C5"  
 P7(CC3)="\$C6"  
 P8(CC3)="\$C7"  
 P9(CC3)="\$C8"  
 P10(CC3)="\$C9"  
 P11(CC3)="\$CA"  
 P12(CC3)="\$CB"  
 P13(CC3)="\$CC"  
 P14(CC3)="\$CD"  
 P15(CC3)="\$CE"  
 P16(CC3)="\$CF"

C1(CC3)="\$D0"  
 C2(CC3)="\$D1"  
 C3(CC3)="\$D2"  
 C4(CC3)="\$D3"  
 C5(CC3)="\$D4"  
 C6(CC3)="\$D5"  
 C7(CC3)="\$D6"

C8(CC3)="907"

N1(CC3)="900"

N2(CC3)="90E"

N3(CC3)="90F"

N4(CC3)="9E0"

N5(CC3)="9E1"

N6(CC3)="9E2"

N7(CC3)="9E3"

N8(CC3)="9E4"

N9(CC3)="9E5"

N10(CC3)="9E6"

N11(CC3)="9E7"

N12(CC3)="9EA"

N13(CC3)="9EB"

N14(CC3)="9E9"

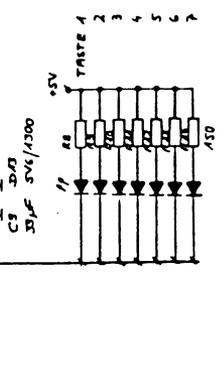
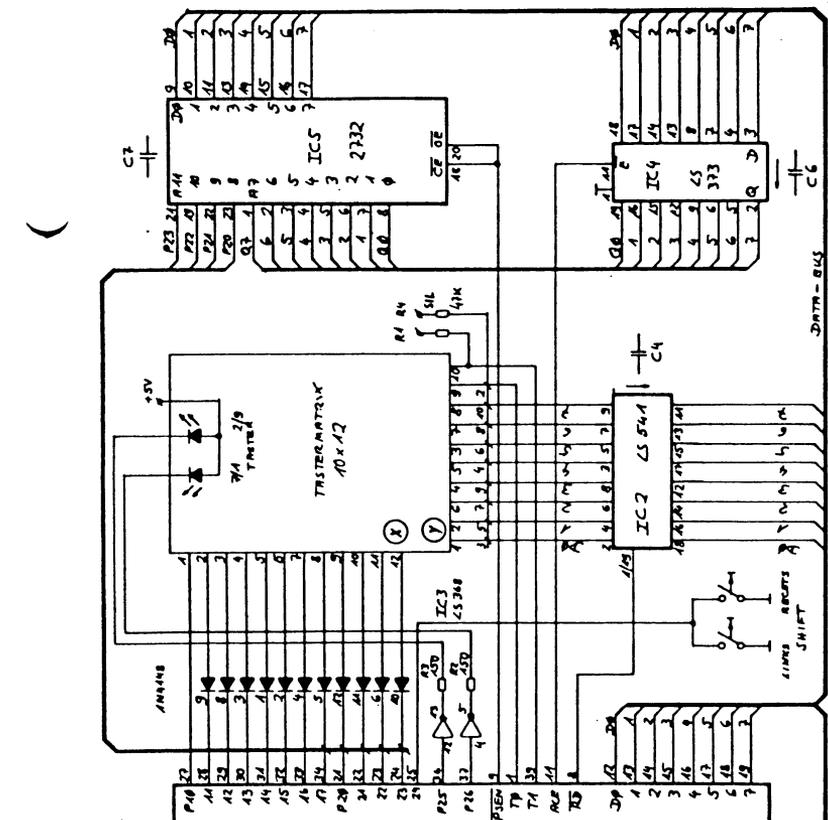
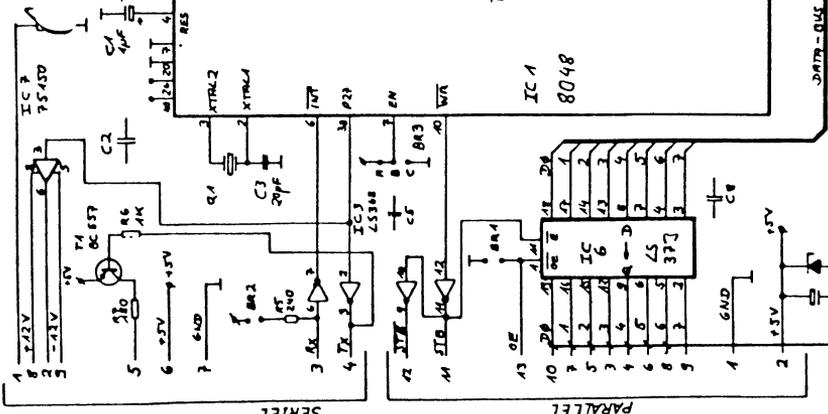
-----  
User 4

Shift und Control betätigt  
-----

(SC3)=111

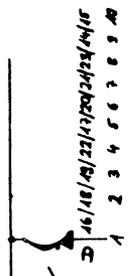
(SC3)=(CC3)

1 7545D  
 2 7545D  
 3 7545D  
 4 7545D  
 5 7545D  
 6 7545D  
 7 7545D  
 8 7545D  
 9 7545D  
 10 7545D  
 11 7545D  
 12 7545D  
 13 7545D  
 14 7545D  
 15 7545D  
 16 7545D  
 17 7545D  
 18 7545D  
 19 7545D  
 20 7545D  
 21 7545D  
 22 7545D  
 23 7545D  
 24 7545D  
 25 7545D  
 26 7545D  
 27 7545D  
 28 7545D  
 29 7545D  
 30 7545D  
 31 7545D  
 32 7545D  
 33 7545D  
 34 7545D  
 35 7545D  
 36 7545D  
 37 7545D  
 38 7545D  
 39 7545D  
 40 7545D  
 41 7545D  
 42 7545D  
 43 7545D  
 44 7545D  
 45 7545D  
 46 7545D  
 47 7545D  
 48 7545D  
 49 7545D  
 50 7545D  
 51 7545D  
 52 7545D  
 53 7545D  
 54 7545D  
 55 7545D  
 56 7545D  
 57 7545D  
 58 7545D  
 59 7545D  
 60 7545D  
 61 7545D  
 62 7545D  
 63 7545D  
 64 7545D  
 65 7545D  
 66 7545D  
 67 7545D  
 68 7545D  
 69 7545D  
 70 7545D  
 71 7545D  
 72 7545D  
 73 7545D  
 74 7545D  
 75 7545D  
 76 7545D  
 77 7545D  
 78 7545D  
 79 7545D  
 80 7545D  
 81 7545D  
 82 7545D  
 83 7545D  
 84 7545D  
 85 7545D  
 86 7545D  
 87 7545D  
 88 7545D  
 89 7545D  
 90 7545D  
 91 7545D  
 92 7545D  
 93 7545D  
 94 7545D  
 95 7545D  
 96 7545D  
 97 7545D  
 98 7545D  
 99 7545D  
 100 7545D



Part	Qty	Part No.	Description
81	30	41	RES
82	1	30	41
83	1	30	41
84	1	30	41
85	1	30	41
86	1	30	41
87	1	30	41
88	1	30	41
89	1	30	41
90	1	30	41
91	1	30	41
92	1	30	41
93	1	30	41
94	1	30	41
95	1	30	41
96	1	30	41
97	1	30	41
98	1	30	41
99	1	30	41
100	1	30	41

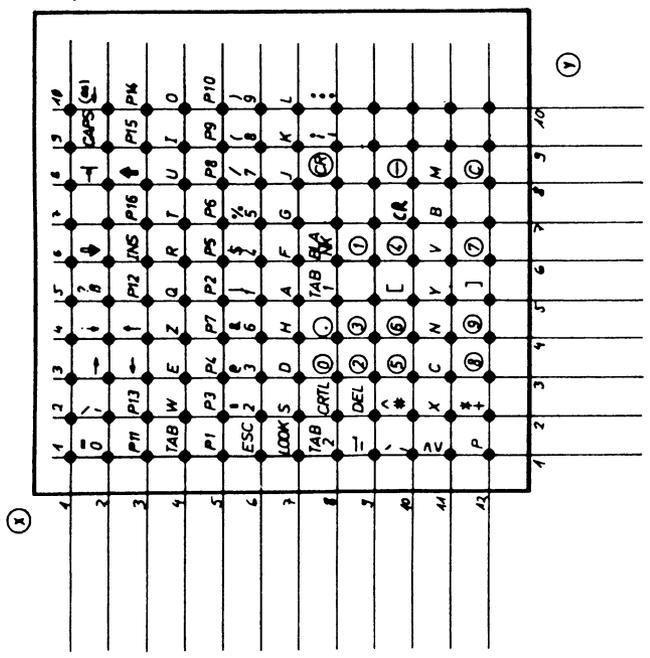
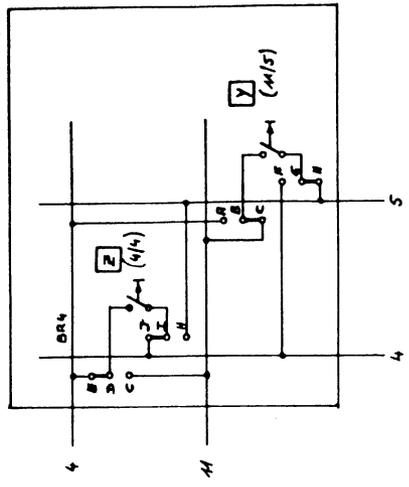
Part	Qty	Part No.	Description
101	1	30	41
102	1	30	41
103	1	30	41
104	1	30	41
105	1	30	41
106	1	30	41
107	1	30	41
108	1	30	41
109	1	30	41
110	1	30	41
111	1	30	41
112	1	30	41
113	1	30	41
114	1	30	41
115	1	30	41
116	1	30	41
117	1	30	41
118	1	30	41
119	1	30	41
120	1	30	41



JUMPER  
TASTER (SCHLEIBER)

⑤ = ZEHNERSTATUR

QUERT 1/2

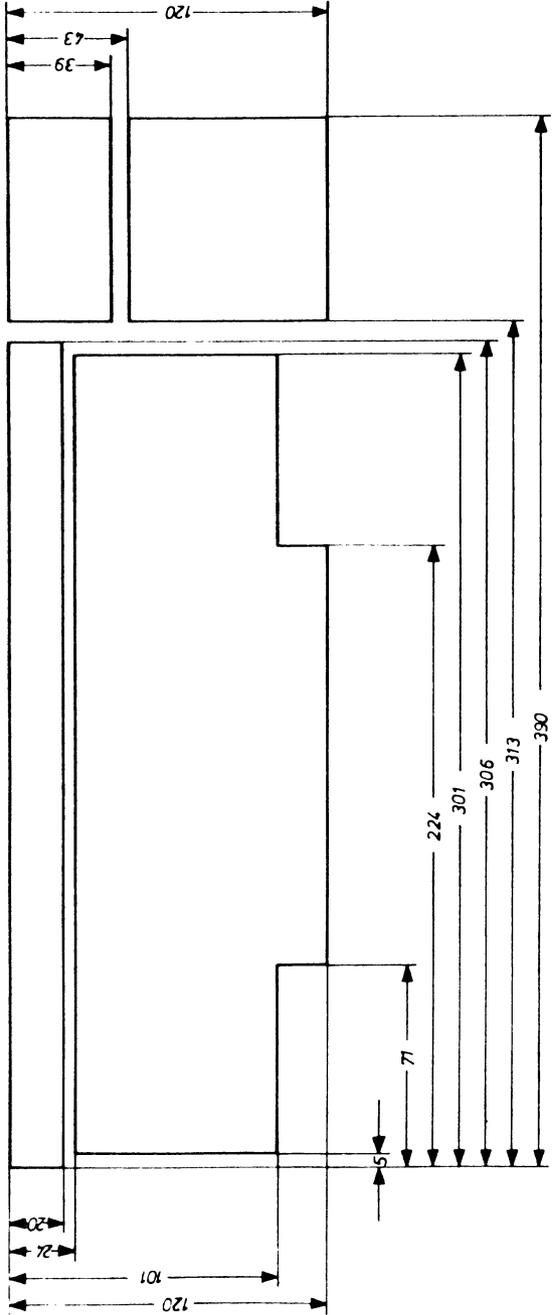


MDF/Hub			
83	Down	Norm	
83	Board	42	AT
	Color	Norm	

# ELZET 80 DIN TAST

Erforderliche Ausbrüche für den Tastatureinbau

M 1:2



- Maße der DIN-TAST - Platine:** Länge 393 ± 0,5 mm  
Breite 121 ± 0,5 mm  
Höhe 21,5 ± 0,5 mm (Oberkante Tastenkappen - Tastenzentrierstifte)  
Länge 409 ± 0,8 mm  
Breite 137 ± 0,8 mm  
Neigungswinkel 10°  
h F-T 42,6 ± 0,5 mm  
h LT 26,2 ± 0,5 mm
- Maße der DIN-TAST/G:**
- Gewicht der ELZET 80 DIN-TAST/G o. Anschlußkabel: 728 ± 2 g