

#####

P R O M M E R- 8 0 V 2.2

Der EPROM-Programmierer für ECB-Systeme von DK4IE
(c) by Buchhandlung Franke * Steiner Strasse 5 * 7531 Eisingen
Format 8", IBM 3740, Single density, CP/M-80

#####

CP/M: Warenzeichen der Fa. Digital Research

TURBO: Warenzeichen der Fa. BORLAND INC.

Z-80: Warenzeichen der Fa. ZILOG

Handbuch und Programme unterliegen dem Autorenschutz; Kopien fuer den persoelichen Gebrauch sind gestattet. ebenso ist es gestattet, zu modifizieren und/oder im Bekanntenkreis UNENTGELTLICH weiterzugeben. Alle uebrigen Rechte bleiben beim Autor bzw. der Vertriebsfirma. Dieser Vermerk muss in allen Kopien enthalten bleiben.

Autor und Vertriebsfirma uebernehmen keine Haftung fuer Schaeden, die aus der Verwendung der Schaltung oder der Programme entstehen. Aenderungen der Schaltung, der Programme oder des Handbuches bleiben ohne Verpflichtung zur Benachrichtigung vorbehalten.

Inhaltsverzeichnis	Seite
1.) Einleitung.....	4
2.) Hardware.....	4
3.) Software	
3.1) Noetige Dateien.....	5
3.2) Funktionen.....	5
3.3) Bedienung.....	5
3.3.1) Kommandos zur EPROM-Bearbeitung.....	7
- EPROM-Typenwahl.....	7
- EPROM-Leerpruefung.....	7
- EPROM aus Puffer programmieren.....	7
- EPROM mit Puffer vergleichen.....	8
- EPROM in Puffer laden.....	8
3.3.2) Kommandos zur Puffer-Bearbeitung.....	9
- Puffer auf Bildschirm.....	9
- Puffer in Textdatei schreiben.....	9
- Puffer mit Konstante fuellen.....	9
- Pufferbereiche verschieben.....	10
- Puffer editieren.....	10
3.3.3) Kommandos zur HEX- und COM- Datei-Bearbeitung.....	11
- Puffer in HEX- Datei schreiben.....	11
- Puffer aus HEX- Datei laden.....	12
- Puffer in COM- Datei schreiben.....	13
- Puffer aus COM- Datei laden.....	14
3.3.4) Ende, zurueck zu CP/M.....	15
3.3.5) Fehlermeldungen.....	15
4.) Installation der Software.....	16
5.) Aufbau und Inbetriebnahme der Hardware.....	17
 Anlagen	
=====	
A1) Stueckliste.....	18
A2) Stromlaufplan.....	20
A3) Bestueckungsplan Europakarte.....	21
A4) Bestueckungsplan Frontplattenadapter, Lochbild.....	22
A5) Unterschiede zu PROMMER V 1.0.....	23
A6) Nachtraege, Aenderungen.....	24

1.) Einleitung

=====

Der Zusatz besteht aus einer Europakarte mit ECB-Bus-Interface, einem Frontplattenadapter, der den EPROM-Sockel und eine LED traegt, und der zugehoerigen Software fuer CP/M 2.2 - Systeme mit Z-80-CPU. Der Arbeitsspeicherbereich (TPA) muss von Adresse 100H bis C000H verfuegbar sein.

Der Zusatz kann

- EPROMS der INTEL-Typen
 - 2716 (Vpp = 25 V), 2732 (Vpp = 25 V),
 - 2732A (Vpp = 21 V), 2764 (Vpp = 21 V),
 - 2764A (Vpp = 12.5 V), 27128 (Vpp = 21 V),
 - 27128A (Vpp = 12.5 V)

bearbeiten. 2716, 2732 und 2732A werden mit 50 ms/Byte programmiert, 2764, 2764A, 27128 und 27128A adaptiv.

- HEX-Diskettendateien lesen und schreiben.
- COM-Diskettendateien lesen, schreiben und aufdatieren.
- Daten in einem internen Puffer editieren.

Das Betriebsprogramm bietet umfangreiche Bedienerfuehrung, sodass Sie spaeter sicher ohne das Handbuch auskommen werden.

Abschnitte, die Sie unbedingt lesen sollten, bevor Sie das Programm starten:

- Noetige Dateien, (Seite 5)
- Installation der Software (Seite 16) und
- Aufbau und Inbetriebnahme der Hardware (Seite 17), speziell Punkte e und g
- Anlage A5, Unterschiede zu PROMMER V 1.0 (Seite 23)

2.) Hardware:

=====

Die Karte enthaelt (siehe Stromlaufplan)

- das ECB-Bus-Interface, bestehend aus U1 bis U5,
- zwei Parallel-Ein-Ausgabe-Bausteine Z-80-PID-A (4 MHz) U6, U7
- einen Spannungswandler zur Erzeugung der Programmierspannungen aus der 5V-Versorgung des ECB-Bus (U11)
- das EPROM-Interface, bestehend aus
 - den "Open-Collector"-Treibern U8, U9
 - den Programmierspannungsschaltern T1 bis T5
 - der umschaltbaren Spannungsversorgung U10, die 5 oder 6 Volt aus der 12V-Versorgung des ECB-Bus erzeugt
 - den beiden Relais 1 und 2, die die Stromversorgung (Vcc gesch.) fuer das EPROM schalten (Bei EPROM Typ 27128/27128A wird Relais 2 auch zur Adressumschaltung benutzt)
 - dem Sockel fuer das zu bearbeitende EPROM.

Die Karte belegt im Ein/Ausgabe-Adressraum des Rechners 16 Adressen, 8 davon werden benutzt, der Rest ist fuer Erweiterungen freigehalten. Die Adressen muessen vor Benutzung an Ihr System angepasst werden, siehe "Installation der Software", Seite 16 und "Aufbau und Inbetriebnahme der Hardware", Seite 17.

3.) Software:

=====

3.1) Noetige Dateien

Die Software fuer den EPROM-Zusatz ist fuer Rechner unter CP/M 2.2 mit Z-80-CPU's geeignet. Sie besteht aus den CP/M-Dateien

- PROMMER.COM..... (Betriebsprogramm)
- PROMTEST.COM..... (Test- / Inbetriebnahme-Programm fuer Leiterplatte)
- PROMINST.COM..... (Installationsprogramm zur Anpassung an verschiedene Rechner/Bildschirmgeraete.

Diese Dateien werden auf dem Bezugs-(Default-)Laufwerk vorausgesetzt.

Zusaetzlich sind auf der mitgelieferten Diskette saemtliche PASCAL-Quellendateien und Hinweise zur Kompilierung enthalten. Schauen Sie sich dazu die Textdatei "LESEMICH.TXT" an.

3.2) Funktionen

Das Programm erlaubt es,

- den Typ des zu bearbeitenden EPROMS einzustellen,
- Leerpruefungen von EPROMS auszufuehren,
- EPROMS aus einem rechnerinternen Puffer zu programmieren,
- ihren Inhalt mit dem des Puffers zu vergleichen,
- den Inhalt eines EPROMS in den Puffer zu uebernehmen,
- den Pufferinhalt in eine HEX- oder COM- Datei zu schreiben,
- den Puffer aus einer HEX- oder COM-Datei zu laden.
- den Pufferinhalt zu editieren,
- den Pufferinhalt in HEX und ASCII auf Bildschirm oder in eine Textdatei auszugeben.

3.3) Bedienung

Das Programm wird gestartet durch "PROMMER <CR>". Es meldet sich mit einer Ueberschrift und den bei der Installation eingestellten Parametern (CPU-Taktrate und I/O-Basisadresse). Anschliessend erscheint das Hauptmenu:

```
*****
* EPROM-Typenwahl.....T   EPROM aus Puffer programmieren..P *
* EPROM mit Puffer vergleichen...C   EPROM in Puffer laden.....L *
* EPROM-Leerpruefung.....B                                     *
*                                                                 *
* Puffer auf Bildschirm.....D   Puffer mit Konst. fuellen.....F *
* Puffer editieren.....M       Pufferbereiche verschieben.....S *
*                                                                 *
* Puffer in HEX-Datei schreiben..W   Puffer in COM-Datei schreiben...U *
* Puffer aus HEX-Datei laden.....R   Puffer aus COM-Datei laden.....G *
*                                                                 *
* Puffer in Textdatei schreiben..X   Ende, zurueck zu CP/M.....E *
*****
```

und die Aufforderung, eine Funktion zu waehlen.

Nach Eingabe des Kennbuchstabens verzweigt das Programm in die zugehoerige Funktion und fraegt die noetigen Parameter ab.

Vor der Beschreibung der einzelnen Funktionen zunaechst noch einige generelle Regeln:

- WICHTIG !!! Um Beschaedigungen von EPROM's zu vermeiden, diese nur einsetzen und entnehmen, wenn das Hauptmenu am Bildschirm zu sehen ist. Die LED ueber dem EPROM-Sockel leuchtet dann nicht.
- Alle Eingaben von Kleinbuchstaben werden in Grossbuchstaben umgesetzt.
- Sie koennen alle Funktionen jederzeit durch CONTROL-S unterbrechen / wiederaufnehmen oder durch ESCAPE abbrechen.
- Wenn das Programm eine Eingabe von mehr als 1 Charakter verlangt (also z.B. Zahlenwerte oder Dateinamen), muessen Sie die Eingabe mit der ENTER-Taste abschliessen; vor Druucken der ENTER-Taste haben Sie die Moeglichkeit, mit BACKSPACE oder DELETE (RUBOUT) die Eingabe zu aendern oder die ganze Eingabe zu loeschen (CONTROL-Z oder CONTROL-X), oder auch mit ESCAPE zum Hauptmenu zurueckzukehren.

Eine Ausnahme von dieser Regel gibt es jedoch: waehrend des Editierens im Puffer werden Zahleneingaben aus praktischen Gruenden anders behandelt. (siehe Funktion "M").

Fuer die Eingabe einzelner Zeichen (z.B. Ja/Nein- Fragen) genuegt ein einzelner Tastendruck.

- Zahlenwerte werden in Hexadezimal- oder in Dezimal-Darstellung eingegeben. Standard-Zahlenbasis ist Hexadezimaldarstellung.

Beispiele:

Eingabe 27	entspricht 27 hex	bzw. 39 dezimal
Eingabe 28H	entspricht 28 hex	bzw. 40 dezimal
Eingabe 31T	entspricht 1F hex	bzw. 31 dezimal

- bei allen Parametern, bei denen es sinnvoll ist, bietet das Programm Standard-Zahlenwerte an, die nur mit der ENTER-Taste bestaetigt werden muessen.
- Bereiche im Puffer oder im EPROM werden normalerweise durch Startadresse und Endadresse festgelegt. Sollten Sie anstatt der Endadresse lieber die Laenge eingeben wollen, so stellen Sie der Zahl den Buchstaben "L" voran; das Programm fuehrt dann die Umrechnung in die Endadresse intern aus.
- Dateinamen werden ohne Anfuhrungszeichen eingegeben.

Beispiele: DATEN.HEX B:DATEN1.ABC usw.

- Der Datenaustausch erfolgt immer ueber einen rechnerinternen Puffer, der aus Anwendersicht bei Adresse 0 beginnt, obwohl er im Adressraum des Rechners an anderer Stelle liegt. Er fasst 16k Bytes, ausreichend fuer 27128/27128A.

3.3.1) Kommandos zur EPROM-Bearbeitung:

* EPROM-Typenwahl : Kommando "T" *

Die Funktion gibt an, welcher EPROM-Typ gerade eingestellt ist (nach Kaltstart undefiniert) und bietet eine Liste der moeglichen Typen an.

Geben Sie den gewuenschten Typ ein und druecken Sie die ENTER-Taste.

* EPROM-Leerpruefung : Kommando "B" *

Das Programm erfraegt die Bereichsgrenzen, in denen auf Inhalt = "FFH" (leer) geprueft werden soll. Fuer die Leerpruefung des ganzen EPROM's benutzen Sie die angebotenen Standardwerte.

Falls EPROM nicht leer, zeigt die Funktion Adresse und Inhalt der ersten Zelle <> FFH an und bricht ab; Sollten Sie alle Zellen mit Inhalt ungleich FFH interessieren, benutzen Sie die Funktion "C" (Vergleich des EPROM's mit Puffer), nachdem Sie den Puffer mit FFH gefuellt haben (Funktion "F").

* EPROM aus Puffer programmieren : Kommando "F" *

Das Programm erfraegt zuerst die Startadresse im EPROM, anschliessend den Pufferbereich. Die Laenge des Pufferbereiches bestimmt dabei die Anzahl der zu programmierenden Bytes.

Das Programm arbeitet in den Schritten:

a.) Pruefen, ob alter EPROM-Inhalt mit den zu programmierenden Daten vertraeglich ist, d.h. ob nicht schon Bits im LOW-Zustand sind, die spaeter im HIGH-Zustand sein sollten. Das EPROM muss also vor der Programmierung nicht unbedingt leer sein. Wenn Unvertraeglichkeit festgestellt wurde, wird die Funktion mit Fehlermeldung abgebrochen, die Programmierung unterbleibt.

b.) Programmieren, je nach Typ mit 50 ms pro Byte oder adaptiv. Waehrend des Programmierens erscheinen auf der Konsole zur Fortschrittskontrolle Punkte. (je 16 Bytes 1 Punkt)

c.) Schlusspruefung, Vergleichen Ist-Inhalt mit Soll-Inhalt.

```
*****  
* EPROM mit Puffer vergleichen :      Kommando "C" *  
*****
```

Das Programm erfragt zuerst die Startadresse im EPROM, anschliessend den Pufferbereich. Die Laenge des Pufferbereiches bestimmt dabei die Anzahl der zu vergleichenden Bytes.

Unterschiede zwischen EPROM und Puffer werden auf der Konsole angezeigt.

```
*****  
* EPROM aus Puffer laden :      Kommando "L" *  
*****
```

Das Programm erfragt zuerst die Startadresse im EPROM, anschliessend den Pufferbereich.

3.3.2) Kommandos zur Puffer-Bearbeitung

* Puffer auf Bildschirm : Kommando "D" *

Das Programm erfragt den Pufferbereich, der ausgegeben werden soll.

Der Pufferinhalt wird hexadezimal und, wenn moeglich, als ASCII-Charaktere ausgegeben.

* Puffer in Textdatei schreiben : Kommando "X" *

Das Programm erfragt den Pufferbereich, der ausgegeben werden soll und den Dateinamen.

Sollte eine Datei dieses Namens schon existieren, fragt die Funktion, ob die Datei ueberschrieben werden soll. Antworten Sie mit "J" fuer ja oder mit "N" fuer nein.

Der Pufferinhalt wird hexadezimal und, wenn moeglich, als ASCII-Charaktere ausgegeben.

* Puffer mit Konstante fuellen : Kommando "F" *

Das Programm erfragt den Pufferbereich, der mit einer Konstanten gefuellt werden soll und den Wert der Konstanten .

* Pufferbereiche verschieben : Kommando "S" *

Das Programm erfragt den Ursprungsbereich im Puffer, der zu verschieben ist und die Startadresse des Zielbereichs im Puffer.

Die Bereiche duerfen sich ueberlappen, die Funktion passt ihre Arbeitsweise an.

Beispiele 1: Bereich ab Adresse 0 bis Adresse 3 soll nach Zielbereich beginnend bei Adresse 2 verschoben werden:

Da Zieladresse hoeher als Ursprungsadresse ist,
wird in der Reihenfolge (3) --> (5),
(2) --> (4),
(1) --> (3),
(0) --> (2) verschoben,
um im Ueberlappungsbereich keine Daten zu zerstoeren.

Beispiel 2: Bereich ab Adresse 4 Laenge 4 soll nach Zielbereich beginnend bei Adresse 1 verschoben werden:

Da Zieladresse niedriger als Ursprungsadresse ist,
wird in der Reihenfolge (4) --> (1),
(5) --> (2),
(6) --> (3),
(7) --> (4) verschoben,
um im Ueberlappungsbereich keine Daten zu zerstoeren.

* Puffer editieren : Kommando "M" *

Das Programm erfragt die Startadresse, bei der der Puffer zum Editieren zu oeffnen ist.

Waehrend des Editierens koennen Sie

- durch Eingabe eines Zahlenwertes und Druecken der Leertaste diesen Wert unter der angezeigten Adresse ablegen,
(Alle Zahlenwerte werden modulo 256 behandelt, "DELETE" oder BACKSPACE wirken nicht)
- durch Druecken der Leertaste (ohne vorher eine Zahl einzugeben) zur naechsten Adresse weiterschalten, ohne den Inhalt der momentanen zu veraendern,
- mit dem Minuszeichen (bzw. Bindestrich) zur vorherigen Adresse zurueckschalten,
- und schliesslich mit der ESCAPE-Taste zum Hauptmenu zurueckkehren.

Beachten Sie dabei: NUR das Druecken der Leertaste kann die Uebernahme eines neuen Wertes in den Puffer bewirken. Wenn Sie also einen neuen Wert eingegeben haben und danach ESCAPE oder "-" eingeben, geht dieser Wert verloren !

3.3.3) Kommandos zur HEX- und COM- Datei-Bearbeitung

* Puffer in HEX- Datei schreiben : Kommando "W" *

Das Programm erfragt den Pufferbereich, der als HEX-Datei abgelegt werden soll, den Dateinamen und einen Offsetwert, der zu jeder Adresse addiert werden soll.

Sollte die Datei bereits existieren, können Sie durch "J" fuer ja bzw. "N" fuer nein entscheiden, ob sie ueberschrieben werden soll.

Beispiele:

- Pufferbereich beginnend bei Adresse 0 bis Adresse 0FFH soll in C:DATA.HEX mit Offset 0 abgelegt werden:

Die Funktion legt 256 Bytes in der Datei DATA.HEX auf Laufwerk C: ab.

Da Offset = 0 ist, entsprechen die Ladeadressen, die zu Beginn jedes Satzes in der HEX-Datei geschrieben werden, den Adressen im Puffer.

- Pufferbereich beginnend bei Adresse 100H Leange 200H soll in Datei DIDI mit Offset 1000H abgelegt werden:

Die Funktion legt 512 Bytes in der Datei DIDI auf dem Standard-Laufwerk ab.

Da Offset = 1000H ist, entsprechen die Ladeadressen, die zu Beginn jedes Satzes in der HEX-Datei geschrieben werden, den Adressen im Puffer plus 1000H. Die Ladeadresse des ersten Satzes ist also 1100H.

Vor Oeffnen und nach Schliessen der Datei wird das Disketten-Betriebssystem zurueckgesetzt, um Diskettenwechsel zu ermöglichen.

```
*****  
* Puffer aus HEX- Datei laden : Kommando "R" *  
*****
```

Das Programm erfragt den Namen der HEX-Datei, die in den Puffer geladen werden soll, und einen Offsetwert, der zu jeder Adresse zu addieren ist.

Beispiele:

- Die Datei C:DATA.HEX aus obigem Beispiel soll wieder in den Puffer geladen werden, wobei kein Offset addiert werden soll. Da die Ladeadressen dieser Datei bei 0 beginnen, wird der Pufferbereich ab Adresse 0 der Laenge 256 Bytes durch den Datei-Inhalt ueberschrieben.
- Die Datei DIDI aus obigem Beispiel soll wieder in den Puffer geladen werden, wobei ein Offset von 0F000H zu addieren ist. Da die Ladeadressen dieser Datei bei 1100H beginnen, wird der Pufferbereich ab Adresse

$$(1100H + 0F000H) \text{ modulo } (2 \wedge 16) = 100H$$

der Laenge 256 Bytes durch den Datei-Inhalt ueberschrieben.

Sollten in einer HEX-Datei fehlerhafte Saetze enthalten sein oder Saetze, die mit den Puffergrenzen unvertraegliche Ladeadressen enthalten, so werden nur diese Saetze ignoriert, die Funktion wird dennoch nicht abgebrochen. Auf dem Bildschirm erscheinen entsprechende Warnungen.

* Puffer in COM- Datei schreiben: Kommando "U" *

Das Programm erfragt einen Pufferbereich, der in eine COM- Datei zu schreiben ist, den Namen der COM-Datei und die Zieladresse in der Datei. Die Laenge des Pufferbereiches bestimmt die Laenge des Zielbereiches in der Datei.

Wertebereich fuer Zieladresse: 0 bis 65535T = OFFF5H

Sollte die COM- Datei bereits existieren, so erfragt das Programm, ob sie ueberschrieben oder nur aufdatiert werden soll.

Dabei gelten folgende Regeln:

- Zieladresse = 0 ist gleichbedeutend mit dem ersten Byte des ersten Sektors der Datei. Beachten Sie den Unterschied zu unter CP/M lauffaehigen COM- Dateien, deren erster Sektor gewoehnlich der Adresse 100H im Hauptspeicher zugeordnet ist !

- Die Sektorgrenzen der Datei sind fuer den Anwender ohne Belang, das Programm sorgt durch Sektor-Pufferung dafuer, dass nur in den angegebenen Zielbereich der Datei geschrieben wird.

- Fuer das Neu- oder Ueberschreiben von COM- Dateien gilt:

Bei Zieladresse <> 0 wird der Anfang der Datei vor dem Zielbereich mit 0 gefuellt.

Der eventuell uebrigbleibende Rest des letzten Sektors wird ebenfalls mit 0 gefuellt.

- Fuer das Aufdatieren existierender COM- Dateien gilt:

Bei Zieladresse <> 0 wird der Bereich von Adresse 0 bis zum alten Dateiende bzw. bis vor Zieladresse unveraendert belassen, ein eventuell entstehender Zwischenraum zwischen altem Dateiende und Zieladresse wird mit 0 gefuellt.

Sollte das Ende des Zielbereiches vor dem alten Dateiende liegen, so wird der Rest der alten Datei unveraendert belassen. Sollte dagegen das Ende des Zielbereiches nach dem alten Dateiende liegen, so wird wie beim Neu- oder Ueberschreiben einer Datei der evtl. uebrige Rest des letzten Sektors mit 0 gefuellt.

Vor Oeffnen und nach Schliessen der Datei wird das Disketten-Betriebssystem zurueckgesetzt, um Diskettenwechsel zu ermoeglichen.

```
*****  
* Puffer aus COM- Datei laden: Kommando "G" *  
*****
```

Das Programm erfragt einen Pufferbereich, der aus einer COM-Datei geladen werden soll, den Namen der COM-Datei und die Quelladresse in der Datei.

Wertebereich fuer Quelladresse: 0 bis 65535T = 0FFFFH

Dabei koennen folgende Fehlersituationen entstehen:

- COM- Datei existiert nicht:
==> Fehlermeldung
- COM- Datei ist leer oder Dateiende liegt vor Quelladresse:
==> Warnung, dass keine Daten geladen wurden.
- COM-Dateiende liegt vor Ende des Quellbereiches:
==> Warnung, dass Pufferbereich nur unvollstaendig geladen wurde.

3.3.4) Ende, zurueck zu CP/M: Kommando "E"

* Ende, zurueck zu CP/M: Kommando "E" *

Das Programm PROMMER wird beendet, CP/M fuehrt Warmstart durch, der Pufferinhalt geht verloren. Um vor versehentlicher Betaetigung zu schuetzen, fraegt das Programm zuvor nochmals nach.

Beantworten Sie die Frage "Ende ? (J/N) mit "J" fuer ja.

3.3.5) Fehlermeldungen und ihr Auftreten

Die meisten Fehlermeldungen, die durch Bedienfehler auftreten koennen, werden vom Programm im Klartext ausgegeben und brauchen daher hier nicht noch einmal erlaeutert werden. Einige seltenere Fehler, vor allem im Zusammenhang mit Disketten-Zugriffen, werden nur als Nummern ausgegeben und deshalb hier noch einmal erlaeutert:

Nr.

99 : Gelesene Datei enthaelt kein CONTROL-Z als Dateiendezeichen. Der Fehler tritt auf, wenn Sie versuchen, mit der "R"-Funktion eine COM-Datei zu lesen.

F0 : Fehler bei Schreibzugriff auf Diskette, da Diskette voll.

F1 : Fehler bei Schreibzugriff auf Diskette, da "Directory" voll.

4.) Installation der Software

=====

Das Programm PROMMER.COM muss in drei Parametern an das jeweilige System angepasst werden:

1.) CPU-Taktrate; Grund:

Der EPROM-Programmierzusatz haelt die fuer die einzelnen EPROM-Typen spezifizierten Programmierpulsbreiten durch Benutzung eines Verzoegerungs-Unterprogrammes ein. Beachten Sie in diesem Zusammenhang auch, dass das Programm NUR fuer CP/M-Systeme OHNE externe Unterbrechungen geeignet ist, NICHT fuer MP/M-Systeme oder z.B. CP/M-Systeme mit Software-Systemuhr !

2.) I/O-Adressen fuer PROMMER-Karte.

Die Karte belegt im I/O-Adressraum des Rechners 16 Adressen, beginnend mit einem ganzzahligen Vielfachen von 16.

Ermitteln Sie einen entsprechend grossen freien Bereich fuer Ihr System (aus Handbuechern, BIOS-Listing usw.)

3.) Anpassung der Bildschirm-Steuerung

Das Programm benoetigt eine Zeichenfolge, die den Bildschirm loescht und den Cursor in die linke obere Ecke setzt. Ermitteln Sie diese Sequenz fuer Ihr Geraet; im einfachsten Falle kann sie aus einem Charakter bestehen, z.B. CONTROL-L (FormFeed, OCH) oder auch aus mehreren, meist eingeleitet durch ein ESCAPE-Zeichen.

Sollten Sie ein Geraet haben, fuer das diese Moeglichkeit nicht besteht, koennen Sie zur Not auch eine Folge aus einem Ruecklauf (CR, ODH) und neun Zeilenvorschueben verwenden. Die Folge darf aus maximal 10 ASCII-Zeichen bestehen.

Wenn Sie diese drei Parameter ermittelt haben, rufen Sie das Programm PROMINST auf und beantworten Sie dessen Fragen. Das Installationsprogramm baut danach die Parameter in PROMMER.COM ein.

Die gleiche Installation wiederholen Sie bitte auch fuer das Hardware-Testprogramm PROMTEST.COM.

5.) Aufbau und Inbetriebnahme der Hardware

=====

Reihenfolge einhalten !

- a) Auf Europakarte alle passiven Bauelemente, die beiden Relais, T1 bis T5 und U10, U11 bestuecken.
- b) 5V- und 12V- Versorgung anlegen.
- c) Mit Potentiometer 1 kOhm (an Pin 1 von U11) die Ausgangsspannung des Wandlers auf 25V einstellen. Die Spannung darf bei Belastung mit ca. 1 kOhm nicht unter 24.5 Volt sinken.
- d) Mit Potentiometer 200 Ohm (beim 7805-Regler) die Ausgangsspannung von U10 auf 6V abgleichen. Bei Ueberbrueckung des Potentiometers muss sie auf 5V absinken.
- e) Alle restlichen Bauelemente bestuecken, Adress-Schalter entsprechend der folgenden Tabelle einstellen (siehe auch "Installation der Software", Seite 16).

I/O-Bank :	DIL-SCHALTER POS				
	1	2	3	4	
00H - 0FH	X	X	X	X	X entspricht "ON"
10H - 1FH	-	x	x	x	
20H - 2FH	x	-	x	x	
30H - 3FH	-	-	x	x	
40H - 4FH	x	x	-	x	
50H - 5FH	-	x	-	x	
60H - 6FH	x	-	-	x	
70H - 7FH	-	-	-	x	
80H - 8FH	x	x	x	-	
90H - 9FH	-	x	x	-	
A0H - AFH	x	-	x	-	
B0H - BFH	-	-	x	-	
C0H - CFH	x	x	-	-	
D0H - DFH	-	x	-	-	
E0H - EFH	x	-	-	-	
F0H - FFH	-	-	-	-	

Bruecke BR zwischen Pin 6 von U9 und Basiswiderstand an T5 einlegen.

- f) Auf Frontplattenadapter ZUERST auf der BESTUECKUNGSSEITE einen Widerstand, 5 Kondensatoren und das Flachbandkabel (oder einen Pfosten-Steckverbinder) bestuecken, DANN auf der LOETSEITE den EPROM-Sockel und eine LED bestuecken. Der Kiphebel des TEXTTOOL-Sockels liegt bei Pin 1. Alle Draehte kurz abschneiden, da Platine dich hinter Frontplatte mit passenden Durchbruechen fuer Sockel und LED zu liegen kommt (ca. 3 bis 5 mm Abstand). Flachbandkabel ueber Steckverbinder mit Europakarte verbinden.
- g) Karte einstecken und Programm PROMTEST starten. (vorher Software installieren, siehe Seite 16 !)

Das Programm fuehrt Sie durch die noetigen Tests, Sie benoetigen nur ein Voltmeter und einen Widerstand 4.7 kOhm. Damit ist die Hardware geprueft, EPROM-s koennen jetzt bearbeitet werden.

Halbleiter:

- 1- SN 74LS00 U4
- 2- SN 7407 U8,U9
- 1- SN 74LS85 U3
- 1- SN 74LS139 U2
- 2- SN 74LS245 U1,U5
- 1- TL 497 U11
- 2- Z-80 PIO A U6,U7 (fuer Systeme mit 6 MHz-Takt PIO B)
- 1- 7805 U10 (TO-220-Gehaeuse)

- 5- BCY 79 o.ae. T1 bis T5
- 3- 1 N 4148 o.ae.

- 1- ZPD 3.9 (evtl. aussuchen bei Inbetriebnahme)
- 1- ZPD 13 (evtl. aussuchen bei Inbetriebnahme)

- 1- LED 3 mm

Widerstaende: (\geq 1/8 Watt, wenn nicht anders angegeben)

- 1- 1 Ohm \geq 0.25 W
- 1- 4.7 Ohm \geq 0.25 W
- 1- 470 Ohm \geq 0.25 W
- 1- 270 Ohm
- 9- 1 kOhm
- 2- 2.2 kOhm
- 9- 10 kOhm
- 1- 33 kOhm
- 3- 100 kOhm

Potentiometer:

- 1- 200 Ohm beide Rastermass 3.75 mm,
- 1- 1 kOhm z.B. WESTON Typ 566-00HS

Kondensatoren:

- 1- 100 Picofarad (Keramik, Raster 5 mm)
- 15- 0.1 Mikrofarad (Vielschicht-Keramik oder MKT, Raster 5 mm)
- 2 ~~15~~ 10 Mikrofarad (Tantalperlen 10 V) + 1x *10µF/20V*
- 1- 47 Mikrofarad (Tantalperle 10 V)
- 2- 220 Mikrofarad (AL-Elko 30 V)

Spule:

- 1- Siemens-Schalenkern RM 5,N28, AL = 250 mit Wickelkoerper und Klammern, bewickelt mit 13.5 Wdg. 0.5 CuL.

Achtung: beim Verkleben der Kernhaelften KEINEN Klebstoff auf Stirnflaechen auftragen !

Relais:

- 2- DIL-Relais 1 x ein, mit Schutzdiode, 5 Volt Erregerspannung
(z.B. SIEMENS Best.Nr. V23100-V4005-A10)

Anmerkung: Die uebliche Beschaltung ist:

Pin 1,14	Kontakt Eingang
Pin 7,8	Kontakt Ausgang
Pin 13	Spulenanschluss 1
Pin 6	Spulenanschluss 2, Kathode der Schutzdiode
Pin 9	Anode der Schutzdiode

Es koennen auch Relais ohne Schutzdiode verwendet werden, bei diesen liegen die Spulenanschluesse oft an Pins 2 und 6. In diesem Fall ist eine Bruecke zwischen Pin 13 und 2 noetig, ausserdem eine Diode 1 N 4148, Anode an Pin 2, Kathode an Pin 6.

Mech. Teile:

- 1- Europa-Leiterplatte
- 1- Leiterplatte fuer Frontplattenadapter
- 1- DIL-Schalter 4 x EIN
- 1- 34 pol. Pfostenstecker fuer Leiterplattenmontage, gerade
(ggfs. 2 Stueck, wenn auch an Frontplattenadapter gesteckt werden soll)
- 1- 34 pol. Kabelverbinder dazu (ggfs. 2 Stueck)
- 10 cm 34 pol. Flachbandkabel passend zum Kabelverbinder
- 1- VG-Leiste DIN 41612 Bauform C, 64 pol. (nur Reihen A und C)

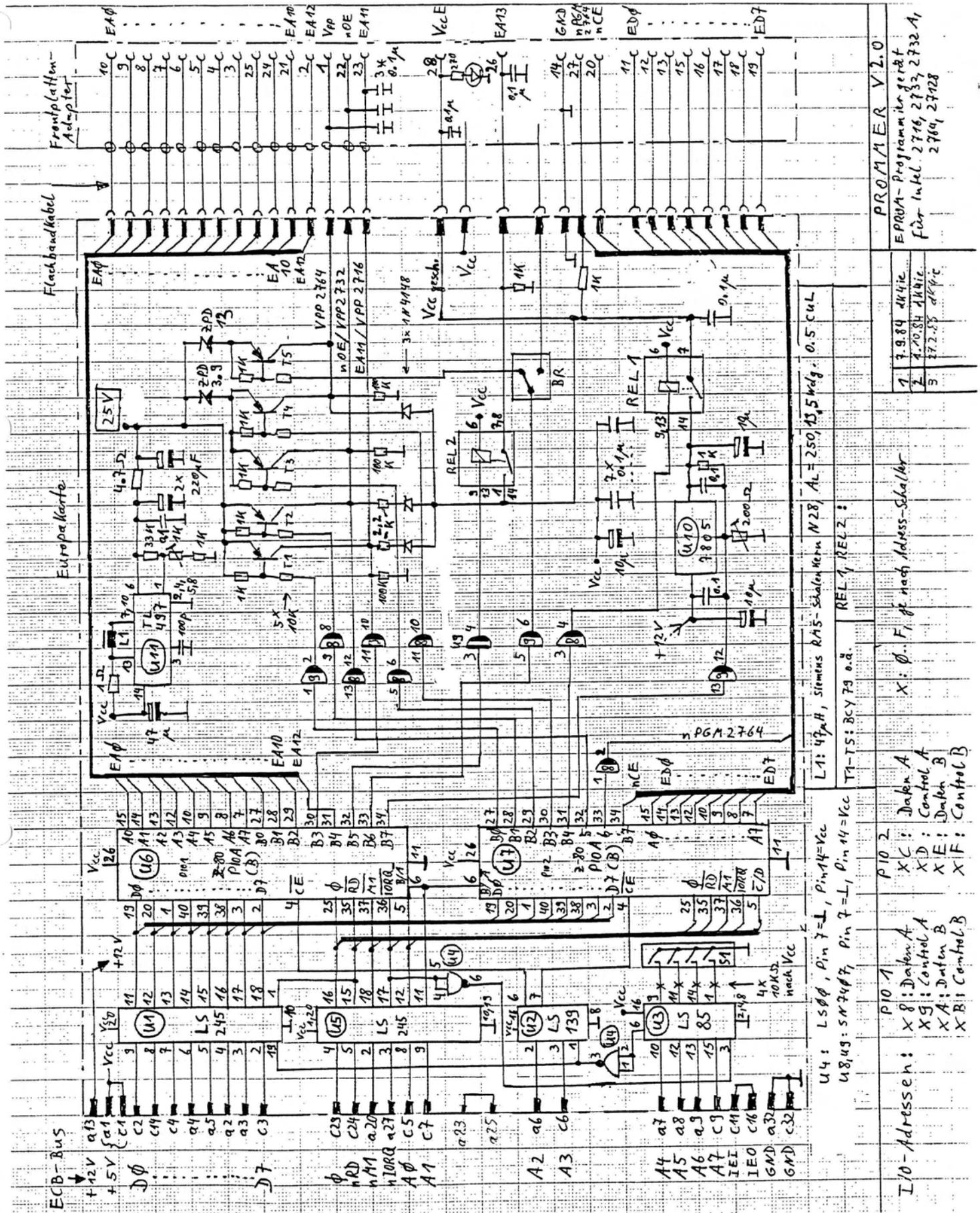
- 4- DIL-Sockel 14pol. (U4,U8,U9,U11)
- 2- DIL-Sockel 16pol. (U2,U3)
- 2- DIL-Sockel 20pol. (U1,U5)
- 2- DIL-Sockel 40pol. (U6,U7)

- 1- TEXTTOOL-DIL-Sockel 28pol. (fuer EPROM)

- div. Schrauben und Kleinteile

bei Bedarf:

- 1- Frontplatte ausgesaegt, mind. 6 TE, mit Montagezubehoer

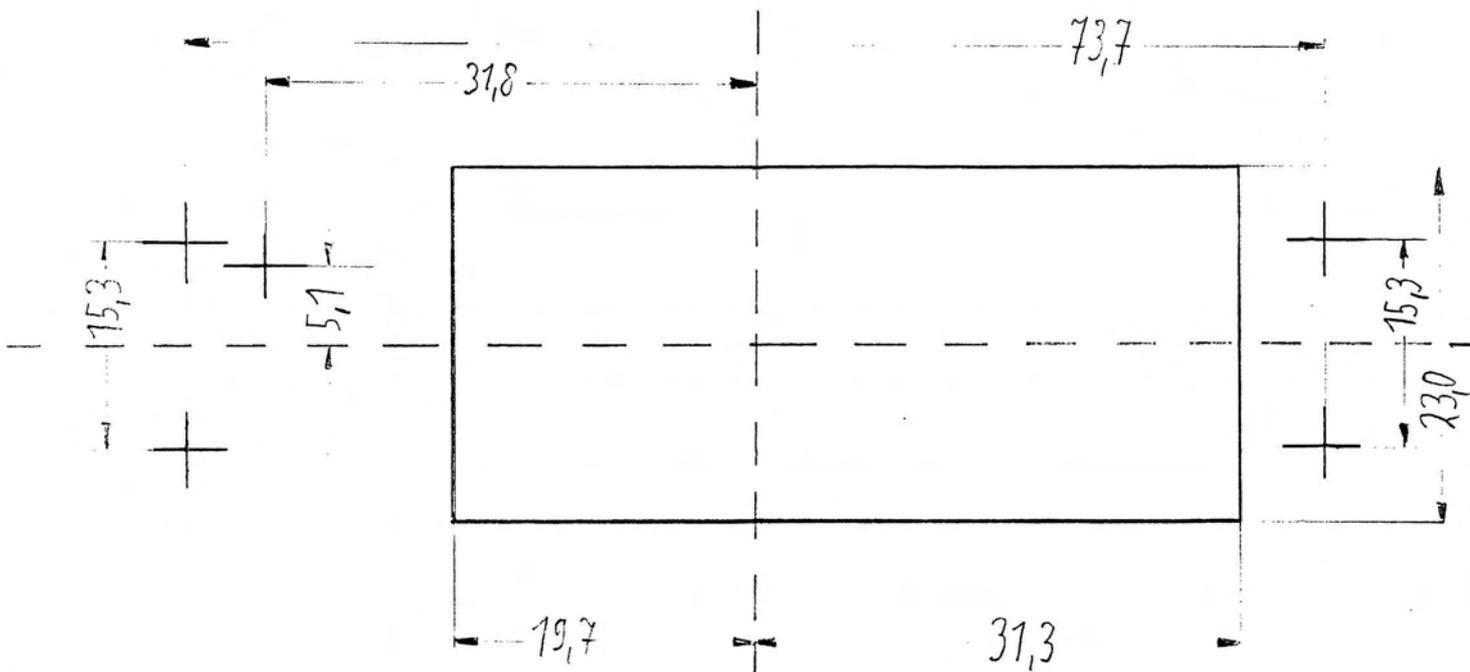
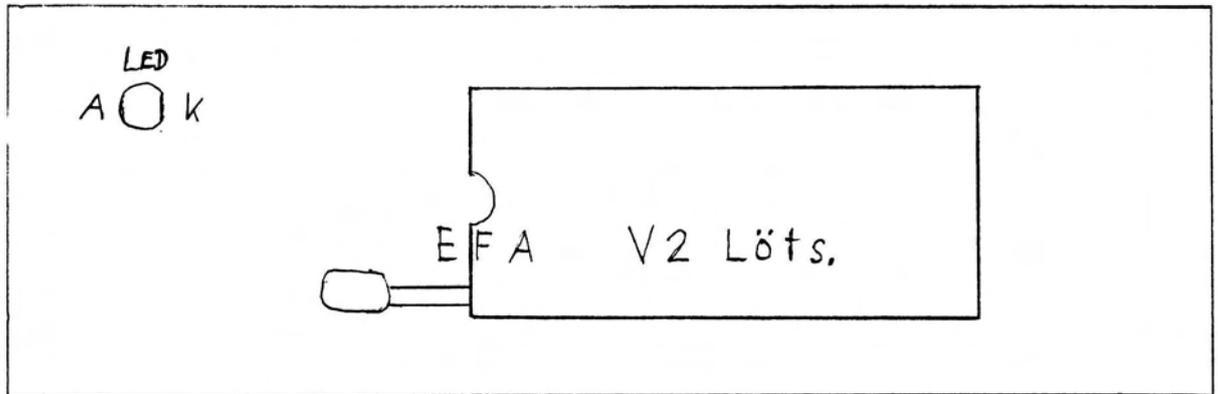
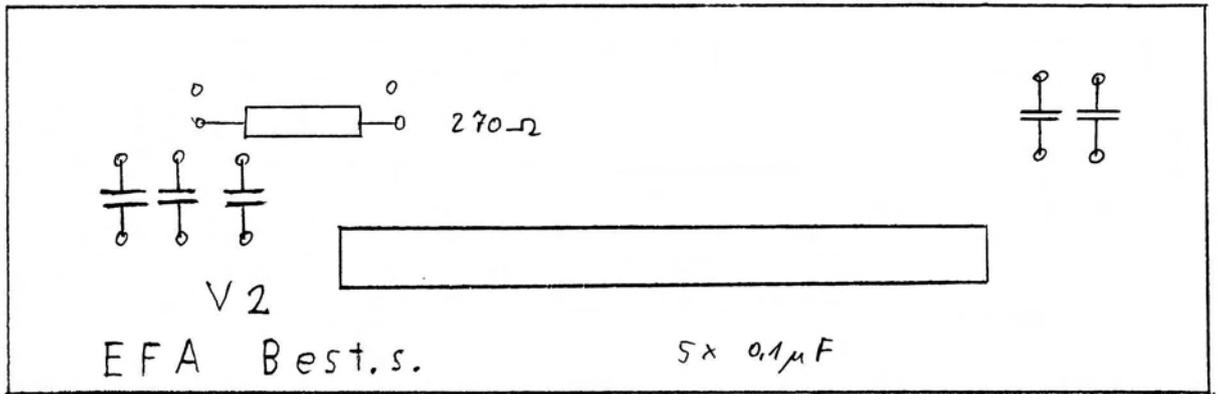


PROMMER V2.0
 EPROM-Programmierzusatz
 für Intel 2716, 2732, 2732 A1,
 2764, 27128

1	7.9.84	4K42c
2	1.10.84	14K1c
3	27.2.85	2K4c

L1: 47µH, Siemens RAS-Schleifen N28, AL = 250, 35,5 µg, 0.5 CuL
 T1-T5: BCY 79 o.ä.
 REL 1, REL 2:
 X: Ø. F. je nach Address-Schalter
 Daten A
 Control A
 Daten B
 Control B

U4: LSØØ, Pin 7=L, Pin 14=Vcc
 U8,U5: SR24ØØ, Pin 7=L, Pin 14=Vcc
 P10 1
 X 8: Daten A
 X 9: Control A
 X A: Daten B
 X B: Control B
 P10 2
 X C: Daten A
 X D: Control A
 X E: Daten B
 X F: Control B



Diejenigen, die Version 1.0 des Programmes bereits kennen und die Schaltung in Betrieb genommen haben, sollten unbedingt beachten:

- die LED, die bisher den Programmiervorgang anzeigte, ist zu Gunsten der Programmierung von EPROM's 2764A und 27128A entfallen; siehe auch geaenderte Brueckenbelegung in Anlage A2.

Daraus folgt: NIE die Schaltung Version 2.1 mit dem Betriebsprogramm Version 1.0 zusammen verwenden, da sonst waehrend jedem Programmiervorgang 12.5 Volt an EPROM Pin 1 angelegt werden ! Dies koennte ein EPROM oder den PROMMER beschaedigen !

Sonstige wesentliche Neuerungen:

- Menu-Steuerung
- Funktionen jederzeit abbrechbar
- COM-Files koenne jetzt gelesen und geschrieben werden.

Bei einigen Exemplaren des PROMMER kam es zu einem kleineren Problem:

Die Integrierte Schaltung U8, ein Open-Collector-Buffer 7407, fiel aus;

Die Analyse ergab, dass der Ausgang Pin 12 Masseschluss hatte, offenbar aus Ueberlastungsgruenden. Die Schaltung und das Programm wurden nochmals geprueft, Fehler wurden keine gefunden. Als einzige Erklaerung bleibt, dass die betroffenen Exemplare nicht die Spezifikation (30 Volt, 40 mA) erfuehlten, oder dass durch undefinierte Verhaeltnisse waehrend des Einschaltens T2 oder T3 aktiv waren, waehrend eine Null am Eingang des Puffers anlag.

Um solchen Problemen vorzubeugen, empfehlen wir, die Abgaenge U8 Pin 12 und U9 Pin 10 direkt unter den beiden IC's auf der Loetseite zu unterbrechen und einen Widerstand von jeweils ca. 100 Ohm in Reihe zu schalten. Sollten Sie aehnliche Probleme haben, lassen Sie es uns wissen.