

# PC-EPROM-Programmierer

**EPROMs vom 2716 bis hin zu den neuen 2-MBit-Typen sowie die Mikrocontroller der Serien 8748 und 8751 können über diese auch für Profi-Anwendungen konzipierte PC-Einsteckkarte programmiert werden. Ein optional einsetzbarer Gang-Adapter ermöglicht sogar das gleichzeitige Programmieren von bis zu 8 EPROMs. Die komfortable Menü-Führung mit zahlreichen Optionen genügt höchsten Ansprüchen.**

## Allgemeines

Auf dem Markt wird bereits eine Vielzahl unterschiedlichster Programmiergeräte angeboten. Von Einfachst-Versionen, die nur bestimmte EPROMs brennen können, bis hin zu professionellen, sehr umfangreich ausgestatteten und meist auch recht teuren Geräten reicht die Palette. Wenn wir im vorliegenden Artikel nun ein weiteres Programmiergerät vorstellen, kann der interessierte ELV-Leser von den Ingenieuren der ELV-Entwicklungsabteilung zu Recht etwas Besonderes erwarten.

Das vorliegende zukunftsorientierte Konzept dieses neuen PC-EPROM-Programmiergerätes ist aufgrund seiner optimierten Leistungsdaten sowohl für den industriellen Einsatz ausgelegt als auch aufgrund seines günstigen Erstellungspreises für den privaten Einsatz geeignet. Die umfangreiche zugehörige Software ist außerordentlich komfortabel in der Bedienung und ermöglicht so ein müheloses Arbeiten.

Neben den schon hinlänglich bekannten EPROMs 2716 bis 27512 können auch die neueren Typen im 1- und 2-MBit-Bereich programmiert werden. Damit sind praktisch alle heute preiswert verfügbaren EPROMs abgedeckt. Darüber hinaus ist das System so weit vorbereitet, daß zu einem späteren Zeitpunkt auch 4-MBit- und 8-

MBit-Typen sowie die 16-Bit-Versionen bearbeitet werden können.

Für die 8-Bit-Typen ist zusätzlich ein Gang-Adapter erhältlich. Damit können in einem Zug bis zu 8 gleiche EPROMs programmiert werden.

Separate Adapter bieten die Programmiermöglichkeit für die Prozessortypen der Reihe 8748 sowie der Reihe 8751.

Im ersten Teil dieses zweiteiligen, sehr ausführlichen Artikels stellen wir Ihnen zunächst die Karteninstallation vor und beschreiben anschließend detailliert das umfangreiche Software-Paket. Daran schließt sich die Schaltungsbeschreibung an.

Der im zweiten Teil beschriebene Nachbau gestaltet sich recht einfach, da alle wesentlichen Komponenten auf einer einzigen, doppelseitigen, durchkontaktierten Leiterplatte, die als PC-Einsteckkarte ausgeführt ist, untergebracht sind. Lediglich für die Adaptersockel mit ihrer geringfügigen Zusatzbeschaltung sind separate Platinen erforderlich, die je nach Einsatzfall Verwendung finden (Einfach-EPROM, 8fach-EPROM oder Mikrocontroller). Die Adapter sind jeweils in optimal gestalteten Ganzmetallgehäusen mit TEXTTOOL-Sockel(n) untergebracht.

## Karten-Installation

Der EPROM-Programmierer besteht aus

einer PC-Einsteckkarte sowie verschiedenen Adaptern zur Programmierung der einzelnen Speicherbausteine. Einsteckkarte und Adapter werden durch ein 40poliges Flachbandkabel miteinander verbunden.

Die Software befindet sich auf einer mitgelieferten 5 1/4"-Diskette (360 kB). Eine Datei READ.ME befindet sich auf der Diskette, die über aktuelle Ergänzungen und Änderungen informiert. Sie sollte zuerst gelesen werden.

Zur Nutzung muß der EPROM-Programmierer zunächst in einem IBM- oder dazu kompatiblen PC installiert werden (Betriebssystem MS-DOS 3.2 oder höher). Dann sollte die Software auf ein Unterverzeichnis der Festplatte kopiert werden, was durch Einlegen der Diskette und Eingabe des Befehls <A: install> erfolgt.

Das Install-Programm fragt nach dem anzulegenden Unterverzeichnis, legt dieses an (soweit nicht bereits vorhanden) und kopiert dann die Programmier-Software hinein.

Vor dem Einsetzen der Karte sollte man sich auf jeden Fall vergewissern, ob der voreingestellte Adreßbereich (\$300..\$307) nicht schon durch andere Zubehörkarten belegt ist. Eine Mehrfachbelegung macht sich meist durch Funktionsstörungen der Hardware bemerkbar. Die Software versucht nach Programmstart das Vorhandensein der PC-Einsteckkarte zu erkennen und

Tabelle 1

Mögliche freie Adreßbereiche des PC  
Nicht aufgeführte Adreßbereiche sind durch PC-Hardware bzw. „lebenswichtige“ Karten belegt

- \$200..\$207 : Joystick
- \$208..\$20F : nicht dokumentiert 1)
- \$218..\$277 : nicht dokumentiert 1)
- \$278..\$27F : LPT2 2)
- \$280..\$2F7 : nicht dokumentiert 1)
- \$2F8..\$2FF : COM2 2)
- \$300..\$31F : Zusatzkarten
- \$320..\$35F : nicht dokumentiert 1)
- \$360..\$36F : Local Area Network
- \$370..\$377 : nicht dokumentiert 1)
- \$378..\$37F : LPT1 2)
- \$380..\$3AF : nicht dokumentiert 1)
- \$3F8..\$3FF : COM1 2)

- 1) Das Betriebssystem MS-DOS beschreibt diese Adreßbereiche nicht näher. Sollten diese Adreßbereiche benutzt werden, so sollte dies mit genügender Vorsicht geschehen.
- 2) Von einer Verwendung dieser Adressen ist abzusehen, da BIOS und andere Anwenderprogramme hier Standard-Schnittstellen erwarten.

veranlaßt im Fehlerfall eine entsprechende Meldung.

Üblicherweise finden sich die Basisadressen von Zubehörkarten im dazugehörigen Handbuch; dadurch dürfte die Auswahl eines freien Adreßbereichs nicht schwer fallen. Die möglichen bzw. erlaubten Adreßräume sind in Tabelle 1 zusammengefaßt. Auch hier hat die Software bei der Initialisierung eine Hilfe bereit: Nicht erlaubte Adreßbereiche (z. B. für Disk-Controller, Timer, DMA, Grafikkarte) werden nicht akzeptiert, denn oftmals hat schon ein einziger Zugriff auf diese Adressen verheerende Folgen in Form eines Systemabsturzes oder gar Datenverlustes!

Soll der voreingestellte Adreßbereich geändert werden, ist die Software vor Einsetzen der Karte zu starten. Da diese also noch nicht eingebaut ist, ergibt sich die Fehlermeldung „PC-Modul nicht vorhanden“. Das Menü erscheint, worauf der Menüpunkt <OPTION/HARDWARE-ADR> anzuwählen und die Basisadresse einzugeben ist.

Der Einbau der Einsteckkarte wird gemäß dem zum PC mitgelieferten Handbuch vollzogen. Im wesentlichen sind dies folgende Schritte:

- PC stromlos machen, alle Kabel abziehen,
- PC aufklappen oder an der Rückseite aufschrauben - i. a. sind dies 5 dicht an der Gehäusekante angeordnete Schrauben,
- freien Slot auswählen. Die Karte benö-

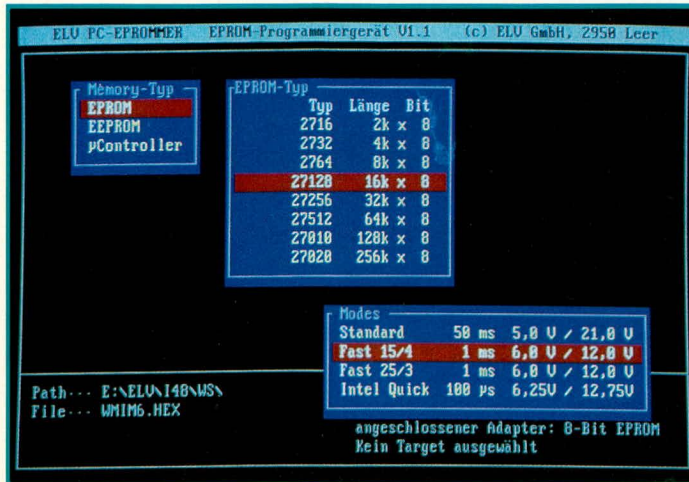


Bild 1:  
Einstellmenü  
des PC-  
EPROM-  
Programmierers  
mit Auswahl-  
möglichkeit  
der verschie-  
denen  
EPROM-,  
EEPROM- und  
Mikrocontrol-  
ler-Typen

- tigt lediglich einen 8Bit-Slot,
- Slotabdeckblech von der Rückwand abschrauben,
- falls notwendig: Adreßbereich auf der Einsteckkarte einstellen,
- Karte vorsichtig in Slot stecken (Stecker zeigt zur Rückseite),
- nach Überprüfung Rechner wieder schließen und Kabel anstecken.

Beim nächsten Programmstart erscheint, wenn alles richtig funktioniert, sofort das Hauptmenü.

Weitere Installationen sind nicht durchzuführen. Selbstverständlich erkennt die Software die installierte Grafikkarte und den Monitor selbsttätig und stellt sich entsprechend darauf ein.

### Betriebsanleitung

Nachdem die Installation erfolgreich abgeschlossen wurde, kann mit dem Aufruf <ELVPROM> der Programmierer gestartet werden (oder in ELV-DOSBATCH unter Menüpunkt 1). Zunächst erscheint ein Pull-Down-Menü, mit dem die Programmierer-Aktionen hauptsächlich gesteuert werden. Einige dieser Aktionen sind zudem über Funktionstasten erreichbar. Verschiedene Verriegelungen sollen Fehlbedienungen vermeiden, die zu Datenverlust oder Beschädigung von Speicherbausteinen führen könnten.

Der Bedienungskomfort und die übersichtliche Führung zeigt sich am Arbeits- und Infofeld. Das Arbeitsfeld befindet sich in den beiden oberen Dritteln des Bildschirms, wo die zur gerade laufenden Aktion benötigten Daten und/oder Operationen angezeigt und abgefragt werden. Im Infofeld befinden sich Angaben über das aktive Unterverzeichnis, die geladene Datei und die aktiven EPROM-Typen. Eine Fußzeile mit Hinweisen auf aktive Tasten erleichtert die Bedienung.

- Zur interaktiven Arbeit sind weitere 4 Typen von Feldern vorgesehen:
- Eingabefeld: Hier wird eine Eingabe, z. B.

der Dateiname, verlangt. Die Aktion wird mit <ESC> abgebrochen, <RETURN> übernimmt den Wert.

- Menüfeld: Hier kann der Anwender aus einer Anzahl vorgegebener Daten oder Funktionen seine Wahl treffen (z. B. EPROM-Auswahl, siehe Bild 1).
- Hinweisfeld: Das Programm weist darauf hin, daß die verlangte Durchführung dieser Aktion ein vielleicht unerwünschtes Ergebnis ergeben könnte - z. B. Überschreiben einer bereits im Speicher befindlichen Datei durch Nachladen einer weiteren.
- Fehlerfeld: Hier liegt ein Fehler oder ein fehlerhaftes Ergebnis vor. Die Aktion kann nicht ordnungsgemäß beendet werden.

Nach der Erläuterung des wesentlichen Software-Konzepts wollen wir uns nun den einzelnen Aktionen zuwenden, die der EPROM-Programmierer bietet:

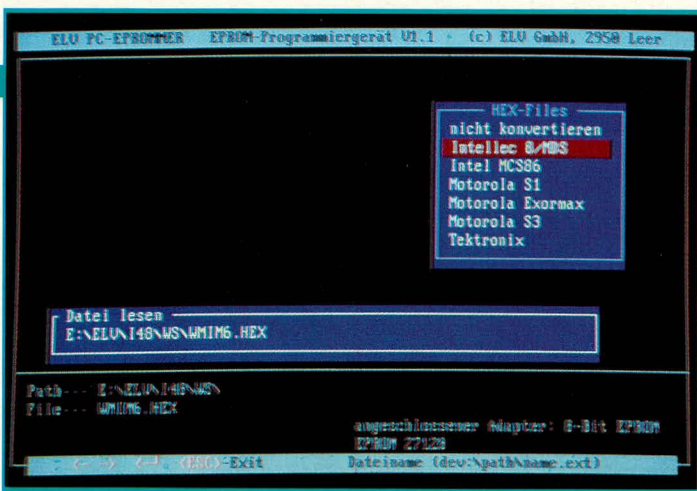
### Menüpunkt FILE

Unter diesem Menüpunkt sind alle Datei-Operationen zusammengefaßt, d. h. es können Dateien von der Magnetplatte geladen, aus einem Speicherbaustein gelesene oder modifizierte Daten auf Diskette/Platte abgelegt, Dateien umbenannt oder gelöscht werden. Die Aktionen im einzelnen:

**LOAD** lädt von der Platte in den Speicherbereich. Hierzu fragt das Programm zunächst nach Pfad, Namen und Extension der zu ladenden Datei. Beim Öffnen des Eingabefeldes erscheint eine durch OPTION/DEFAULT FILE gewählte Vorgabe. Wildcards zur Dateisuche sind erlaubt; so gefundene Dateien zeigt ein Menüfeld an.

Das Programm setzt zunächst das eigene Unterverzeichnis als Path voraus, sofern nicht mit FILE/PATH oder im Eingabefeld konkret ein anderer Pfad angegeben wurde.

Anhand der File-Extension erkennt das Programm, ob die Datei direkt programmierbar (\*.COM, \*.BIN) oder konvertierbar sein könnte. Im letzteren Fall sind in einem Menüfeld mögliche Konversionen



**Bild 2: Einstellung der Konvertierungsmöglichkeiten verschiedener Dateiformate ins binäre EPROM-Format. Diese Funktion wird direkt von der Eprommer-Software bereitgestellt.**

selten Programmieradapter ab, sofern dies nicht automatisch bei Typwechsel erfolgt.

**Menüpunkt ACCESS**

Dieser Menüpunkt verwaltet die gesamten Aktionen, die in bezug auf das Lesen und Schreiben eines EPROMs durchgeführt werden können.

AUTO läßt die Aktionen „Blanktest“, „Program“ und „Verify“ nacheinander automatisch ablaufen. Hierzu wird über ein Eingabefeld die Start-, End- und Speicherstartadresse abgefragt. Danach kann das EPROM eingesetzt und programmiert werden. Ein auftretender Fehler bricht die Aktion ab.

BLANKTEST prüft den eingesetzten Speicherbaustein von Start-Adresse bis End-Adresse auf das durch den Typen vorgegebene Leerzeichen (bei EPROMs ist dieses z. B. \$FF). Fehler werden angezeigt.

PROGRAM programmiert den eingesetzten Speicherbaustein von Start-Adresse bis End-Adresse mit den im Speicher befindlichen Daten nach dem in der Typliste vorgegebenen Verfahren. Auch hier werden Fehler angezeigt.

**Tabelle 2**

Mögliche Dateikonversionen bei LOAD

- nicht konvertieren
- Intellec 8/MDS
- Intel MCS86
- Motorola S1
- Motorola Exormax
- Motorola S3
- Tektronix

auswählbar (siehe Tabelle 2). Während der Übersetzung in Maschinencode wird gleichzeitig das Prüfbyte mit eigenen Ergebnissen verglichen, und eventuelle Fehler werden anschließend im Fehlerfeld angegeben.

EXE-Dateien sind wegen ihrer Struktur nicht programmierbar und werden daher nicht akzeptiert. Eine solche Datei sollte vorher mit EXE2BIN in eine COM-Datei umgewandelt werden.

NEW initialisiert das Laden oder Speichern einer neuen Datei, sofern nicht automatisch bei FILE/LOAD geschehen.

SAVE sichert den Speicher auf Magnetplatte. Neben der Abfrage des Dateinamens im Eingabefeld (hier werden Wildcards nicht akzeptiert) fragt das Programm noch nach Anfangs- und Endadresse des Programms. Danach wird die Datei als Speicherkopie abgelegt.

PATH ändert die Voreinstellung des aktuellen Pfad-Namens. Wie bereits erläutert, setzt das Programm bei Dateioperationen zunächst immer das Unterverzeichnis voraus, in dem es sich befindet.

DELETE löscht die im Eingabefeld oder über Menüfeld spezifizierte Datei, entspricht also im wesentlichen dem bekannten MS-DOS-Befehl DEL. Im Gegensatz zu diesem wurde hier aus gutem Grund nur ein Löschen einzelner Dateien zugelassen, da im Falle eines versehentlichen Löschens ein automatisches Undelete (durch z. B. PCTools oder XTREE) von über mehrere Cluster ausgedehnten COM- und EXE-Dateien nicht mehr möglich ist.

RENAME benennt die im Eingabefeld oder über Menüfeld spezifizierte Datei in den in einem weiteren Eingabefeld angegebenen Namen um. Das entspricht im wesentlichen dem bekannten MS-DOS-Befehl REN. Auch hier wurden, da diese Aktion

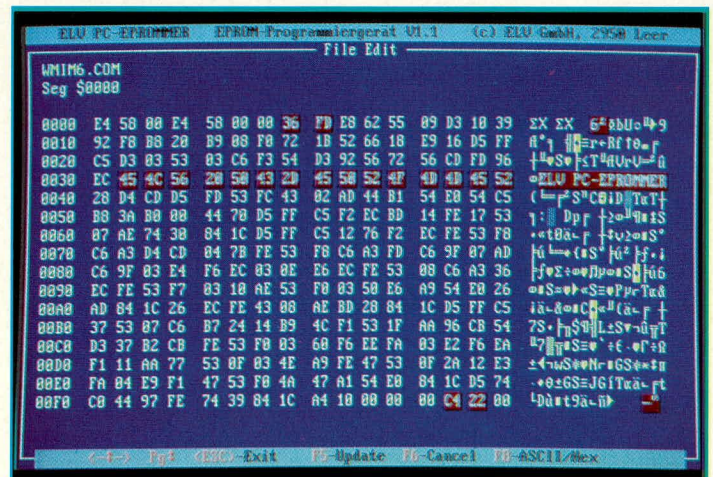
im laufenden Programm allgemein nur sehr sparsam verwendet werden sollte, nur wieder einzelne Dateien zugelassen. Move-Operationen (in anderen Pfad) dürfen innerhalb des gerade aktiven Laufwerks durchgeführt werden.

**Menüpunkt EPROM**

Dieser Menüpunkt verwaltet die gesamten Aktionen, die in bezug auf einen EPROM-Typen durchgeführt werden können.

DEVICE SELECT zeigt im Menüfeld zunächst die möglichen Speicher-Arten EPROM, EEPROM sowie Mikrocontroller an. Danach erscheint das Menüfeld zur

**Bild 3: EDIT-Funktion der EPROM-Programmier-Software anhand eines typischen Datei-Beispiels.**



Auswahl der jeweils unterstützten Typen sowie ihre Speicherkonfiguration. Ist der ausgewählte Speicherbaustein auch noch in mehreren Programmiermodi erhältlich, kann ein solcher im dritten Menüfeld ausgewählt werden. Die hier ausgewählten Daten erscheinen dann im Infobereich (siehe Bild 1).

DEVICE LIST ermöglicht die Wahl des Speicherbausteins anhand seiner aufgestempelten Bezeichnung. Hierzu sind in einer mitgelieferten Datei entsprechende Merkmale anzugeben. Sollte der Typ nicht aufgeführt sein, wird über ein Hinweisfeld das Umschalten auf EPROM/DEVICE SELECT ermöglicht.

DEVICE MODE bietet direkten Zugriff auf die Programmiermodi ohne ausführliche Auswahl über EPROM/DEVICE SELECT.

MODULE SELECT fragt einen gewech-

VERIFY vergleicht den eingesetzten Speicherbaustein von Start-Adresse bis End-Adresse mit den im Speicher befindlichen Daten. Fehler werden angezeigt.

READ liest Daten aus dem eingesetzten Speicherbaustein von Start-Adresse bis End-Adresse in den Speicher.

**Menüpunkt DATA**

Unter diesem Menüpunkt sind die Aktionen zusammengefaßt, die für das Modifizieren der im Speicher befindlichen Datei benötigt werden. Da die Aktionen EDIT, FILL und MOVE bei Unachtsamkeit beträchtlichen Schaden anrichten können, lassen sie sich mit OPTION/DATA MODIFY sperren.

DUMP zeigt eine Datei auf dem Bildschirm an. Änderungen lassen sich hier jedoch nicht durchführen.

EDIT erlaubt die Änderung der im Spei-

cher befindlichen Datei. Jeweils 256 Byte werden in das Aktionsfenster geholt und lassen sich binär oder in ASCII ändern. Die geänderten Bytes werden zur Identifikation markiert (siehe Bild 3). Der Anwender kann dann entscheiden, ob die Änderungen tatsächlich in die Datei übernommen werden sollen.

**FILL** füllt einen Speicherbereich von Start-Adresse bis End-Adresse mit einem vorwählbaren Byte.

**MOVE** verschiebt einen Speicherbereich von Start-Adresse bis End-Adresse in den ab Ziel-Adresse beginnenden Bereich.

**COMPARE** vergleicht einen Datenbereich von Start-Adresse bis End-Adresse mit dem Speicherbereich, der bei der Ziel-Adresse beginnt. Fehler werden angezeigt.

### Menüpunkt OPTION

In diesem Menüpunkt sind Sonderaktionen zur Programm-Voreinstellung untergebracht, die bereits erwähnt wurden. Nachfolgend nochmals eine kurze Zusammenfassung:

**DEFAULT FILE** gibt eine Standard-Datei oder Wildcards vor (z. B. \*.BIN). **DEFAULT CONV** gibt den Stand des Menübalkens vor, wenn das Konversionsmenü geöffnet wird. Damit gestaltet sich der Zugriff schneller.

**DEFAULT PROM** gibt den Stand des Menübalkens vor, wenn die Art- und Typmenüs geöffnet werden. Damit kann z. B. eine Vorwahl auf ein oft verwendetes EPROM gegeben werden.

**DATA MODIFY** sperrt bzw. entsperrt die Aktionen EDIT, FILL und MOVE.

**HARDWARE-ADR** stellt das Programm auf die auf der Einsteckkarte gewählte Basisadresse um (siehe auch unter „Installation“!).

**SAVE OPTIONS** speichert die neuen Optionen in der .CFG-Datei.

### Menüpunkt QUIT

Hiermit (oder <ESC> im Hauptmenü) wird das Programm verlassen. Geöffnete Dateien werden geschlossen, dynamisch belegte Speicherbereiche (für Daten) werden wieder freigegeben, evtl. zusätzlich angelegte Dateien (zur Verwaltung bei großen EPROMs) werden wieder gelöscht.

### Beschreibung der Hardware

Die Hardware ist, wie bereits erwähnt, hauptsächlich auf einer PC-Einsteckkarte aufgebaut und besteht aus den Gruppen PC-Bus-Anpassung, Adapter-Anpassung,  $V_{cc}$ -Umschaltung und Spannungsgenerator. Diese Gruppen wollen wir im einzelnen näher betrachten:

#### PC-Bus-Anpassung (Bild 4)

Mit IC 1 - IC 5 werden die PC-Busleitungen vom eigentlichen Aufbau der PC-Einsteckkarte abgetrennt. IC 5 (74 LS 245) trennt den Datenbus bidirektional, die Richtungswahl wird über das Signal „ $\overline{IORC}$ “ gesteuert. IC 1 (74 LS 688) sorgt für die Selektion der I/O-Adresse, wobei die untersten 3 Adreßbits (= 8 Adressen) auf der Karte fest für die einzelnen Register belegt sind (siehe Tabelle 3). Damit stellt BR 1 die PC-Adresse A3 als kleinstmögliche Wahl ein. Mit IC 4 (SN 74 LS 00) wird zudem der Kartenzugriff nur bei  $\overline{IORC}$  und  $\overline{IOWC}$  erlaubt.

**Tabelle 3**

#### Adreßbelegung

\$xx0	: EPROM-Adressen 0 - 7
\$xx1	: EPROM-Adressen 8 - 15
\$xx2	: EPROM-Adressen 16 - 19, Versorgungs- und Programmierspannungen
\$xx3	: /OE, /CE, Programmierspannungen
\$xx4	: Leitungsfreigaben, Programmierspannungen
\$xx5	: EPROM-Daten 0 - 7 (Schreib-Lese-Funktion)
\$xx6	: EPROM-Daten 8 - 15 (Schreib-Lese-Funktion)
\$xx7	: nicht benutzt

Über die Signale A0, A1 und A2 ordnen IC 2 und IC 3 (74 LS 138) die einzelnen Register dem entsprechenden PC-Bus-Zugriff zu. IC 2 steuert hierbei die WRITE-Funktion (Adressen- und Datenausgabe), IC 3 ermöglicht das Lesen der beiden Datenregister, was für die Auslesefunktion und das Erkennen der Karte notwendig ist.

Über die Signale A0, A1 und A2 ordnen IC 2 und IC 3 (74 LS 138) die einzelnen Register dem entsprechenden PC-Bus-Zugriff zu. IC 2 steuert hierbei die WRITE-Funktion (Adressen- und Datenausgabe), IC 3 ermöglicht das Lesen der beiden Datenregister, was für die Auslesefunktion und das Erkennen der Karte notwendig ist.

#### Adapter-Anpassung (Bild 4 und 5)

Hierzu sind IC 6 - IC 16 mit der benötigten Zusatzbeschaltung vorgesehen, wobei IC 6 und IC 7 für die unteren 8, IC 8 und IC 9 für die oberen 8 Datenleitungen zuständig sind. Über IC 6 und IC 8 (74 HC 244) wird gelesen, IC 7 und IC 9 (74 HC 373) schreiben. Im Gegensatz zu IC 5 ist hier eine Trennung der Schreib- und Lese-Funktionen erforderlich, da die Daten (wie auch die Adressen) bei den meisten Speicherbausteinen während des Programmierzyklus ständig anstehen müssen. Hier wurde auf eine konventionelle Lösung zurückgegriffen, da spezielle ICs noch recht kostspielig sind.

IC 10 (74 HC 373) gibt die EPROM-Adressen A0 bis A7 an den Adapter aus. Hier sind keine besonderen Ansprüche gestellt, da diese 8 Adreß- und die Datenleitungen seit jeher mit TTL-Pegel bedient wurden.

Ab EPROM-Adresse A8 wird es jetzt spannend: Die Adressen A8 bis A20 und  $\overline{OE}$  spiegeln gleichsam die EPROM-Ent-

wicklung der letzten Jahre wider. Während Anfang der 80er Jahre die EPROMs 2716 und 2732 mit einem DIL24-Gehäuse daherkamen, wurden die Typen 2764 bis 27512 im DIL28-Gehäuse hergestellt. Dann erschien vor wenigen Jahren das 1MBit-EPROM im DIL32-Gehäuse, welches auch noch für das 8MBit-EPROM ausreicht.

Als logische Konsequenz wandert der  $V_{cc}$ -Anschluß von Pin 28 über Pin 30 nach Pin 32, wenn für die Gehäuse DIL24 bis DIL32 ein und derselbe Adapter mit einem („sündig teuren“) Textool-Sockel verwendet werden soll. Ebenso liegen die Programmierspannungsanschlüsse, die von Typ zu Typ auch noch unterschiedlich sein können, an  $\overline{OE}$  und den Adressen A9, A13, A15 und A19. Tabelle 4 bietet dem interessierten Leser einen kleinen Einblick in die Vielfalt der Pinbelegungen, die mit dem EPROM-Programmierer abgedeckt sein müssen.

IC 11 (74 HC 373) schaltet EPROM-Adressen A8 bis A15. Dabei werden A8, A10, A12 und A14 „normal“, also wie A0 bis A7, von den entsprechenden Ausgängen des IC 11 (Pins 2, 6, 12 und 16) angesteuert, wogegen die Adreßleitungen A9, A11, A13 und A15 bei einigen EPROMs jedoch andere Funktionen haben (siehe Tabelle 4).

Für die Leitungen A9, A11 und A15, die im Bedarfsfall  $V_{pp}$  schalten müssen (z. B. EPROM 2716: Leitung A11; 2764 bis 27256: Leitung A15), erklären wir hier stellvertretend den Ausgang von A9:

IC 11, Pin 5 wird vom Ausgang über den Inverter IC 13 (SN 7406), Pin 2 abgetrennt. Hier mußten offene Kollektoren mit einer Spannungsfestigkeit von 30 V gewählt werden, da der Ausgang Pegel bis ca. 25 V führen kann. R 43 hält den Ausgang A9 auf L-Pegel, wenn sich die Karte im Reset-Status befindet (alle Ausgänge hochohmig oder auf definiertem TTL-Pegel).

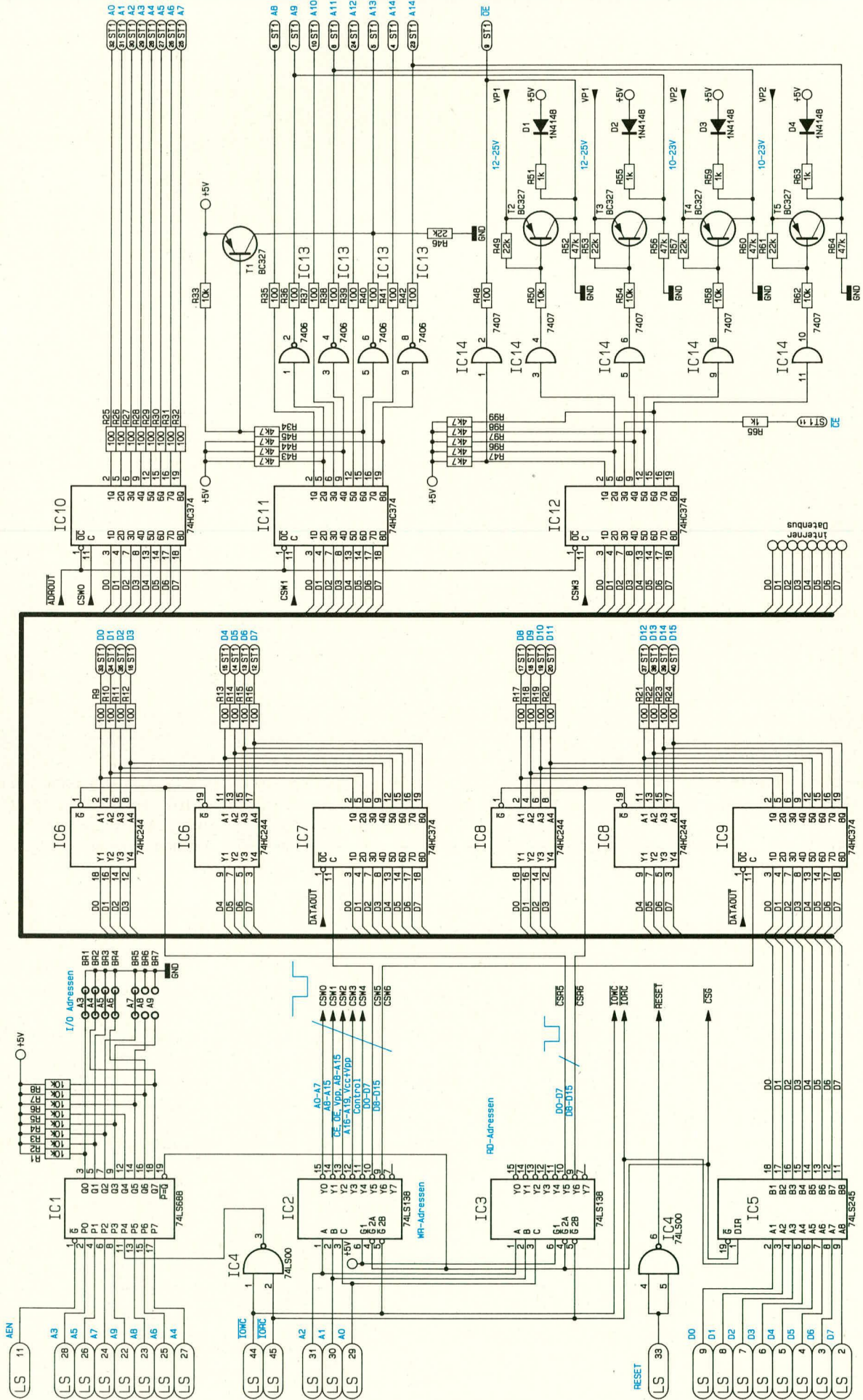
Im Normalfall (Adreß-Leitung A9) wird der benötigte H-Pegel über die Diode D 2 und R 55 vorgegeben. Im Programmierfall ( $V_{pp}$ ) erscheint an IC 12, Pin 9 (74 HC 373) ein L-Signal, das über IC 14, Pin 6 (SN 7407) und R 54 den Transistor T 3 öffnet.

Damit kann die Programmierspannung  $V_{pp}$  an den Ausgang gelangen. Die Ausgänge von IC 14 haben ebenfalls eine Spannungsfestigkeit von 30 V; R 96 hält im Reset-Status das Gatter und damit T 3 gesperrt.

Die Adreßleitung A13 benötigt im Sonderfall (z. B. für die EPROMs 2716 und 2732)  $V_{cc}$  (= 5 V niederohmig) am Ausgang. Hierzu dient T 1, der leitet, wenn IC 11, Pin 15 auf L-Pegel geschaltet wird.

IC 12 schaltet neben den bereits erwähnten Funktionen der Programmierspannungsfreigabe auch die Ausgangsleitungen  $\overline{CE}$  und  $\overline{OE}$ . Während  $\overline{CE}$  für alle EPROMs gleiche Funktion hat, muß  $\overline{OE}$  für verschiedene Typen wieder  $V_{pp}$  bereithalten.

Bild 4:  
PC-Bus-Anpassung  
und Teil-  
schaltbild  
der Adapter-  
Anpassung



Der Schaltungsaufbau mit D 1, R 51, IC 14, Pin 4 und T 2 entspricht dem von A9, A13 und A15. Der Treiber für den L-Pegel ist hier aber auch ein Gatter des Typs SN 7407 (IC 14, Pin 2), da im Ruhezustand  $\overline{OE}$  auf H-Pegel liegen sollte.

Die Adreßleitungen A16 bis A19 sind wiederum ähnlich A8 bis A15 aufgebaut und kommen von IC 15 (74 HC 373). Die nur TTL-Pegel führenden Leitungen (A16, A18) steuern direkt den Ausgang; A17 muß auch  $V_{cc}$  liefern, A19  $V_{pp}$ . Da  $V_{cc}$  bei den

intelligenten Programmiermodi mit Spannungen bis 6,75 V arbeitet, dient hier D 6 (ZPD 3V3) zur Erhöhung der Spannungsdifferenz zwischen Ausgang IC 15, Pin 5 sowie T 6. Die gleiche Schaltungsanordnung gilt für D 7 und T 8, die die  $V_{cc}$ -Leitung des Adapters (Pin 32) schalten.

IC 16, Pin 5 (74 HC 273) schaltet über IC 4, Pin 8 die Adreß- und Programmierleitungen, Pin 2 über IC 4, Pin 11 die Datenleitungen in hochohmigen Zustand. Damit sind im Resetfall oder per Software alle Pegel abtrennbar.

Sämtliche Ausgänge (soweit nicht niederohmig erforderlich) sind mit Widerständen versehen (R 9 - R 24, R 25 - R 32, R 35 - R 42, R 48, R 65, R 68 - R 71), die im Kurzschlußfall die Treiber-ICs kurzzeitig schützen.

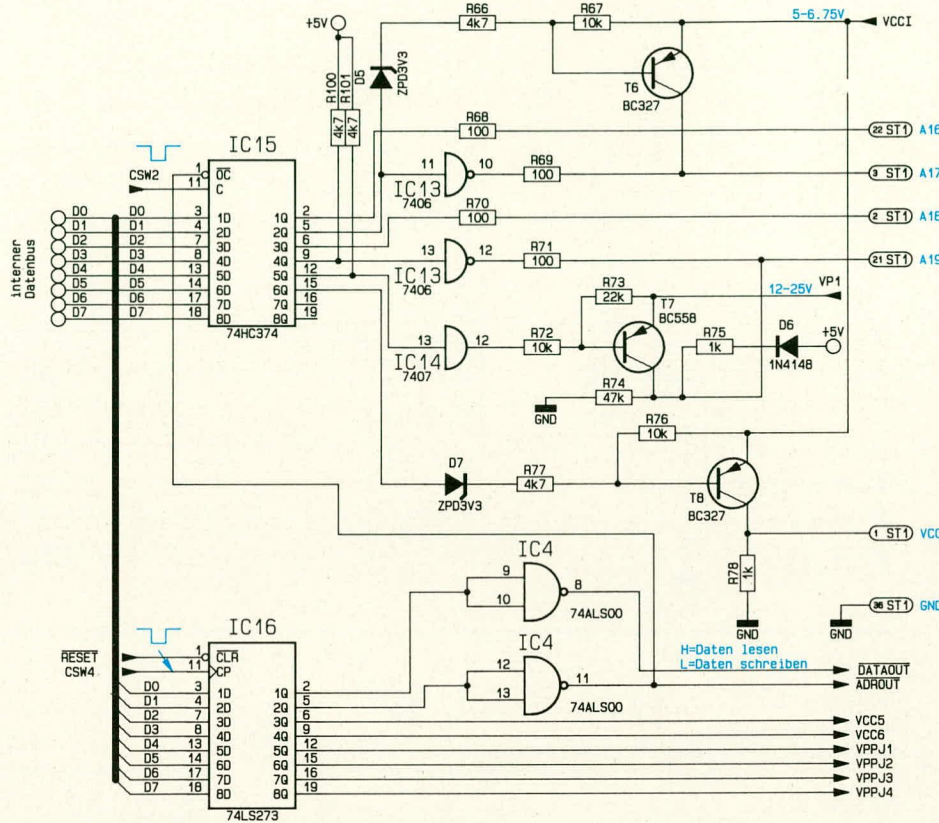


Bild 5: Teilschaltbild der Adapter-Anpassung

### $V_{cc}$ -Umschaltung (Bild 6)

Wie bereits erwähnt, verlangen die intelligenten Programmieralgorithmen unterschiedliche Spannungen im Bereich von 5 bis 6,75 V am  $V_{cc}$ -Anschluß des EPROMs, die also umschaltbar aus den + 12 V des PC gewonnen werden müssen. Mit IC 17 (LM317) und den Widerständen R 79 bis R 82, dem Trimmer R 102 sowie den Pins 6 und 9 von IC 16 wird dieser Zweck erreicht. Wären R 80 und R 81 nicht vorhanden, so würde sich über R 79, R 82 und R 102 eine Spannung von etwa 6,7 V einstellen. Durch Einspeisen des H-Pegels aus IC 16 in Pin 1 des IC 17 bzw. die Parallelschaltung von R 80 und R 81 zu R 82 und R 102 (bei L-Pegel des IC 16) kann die Ausgangsspannung den gewünschten Werten angepaßt werden. R 80 verursacht einen Spannungssprung von etwa 1,4 V, R 81 etwa 0,4 V. Damit werden die benötigten Spannungen 5 V, 6 V und 6,75 V innerhalb der zulässigen Toleranz erzeugt.

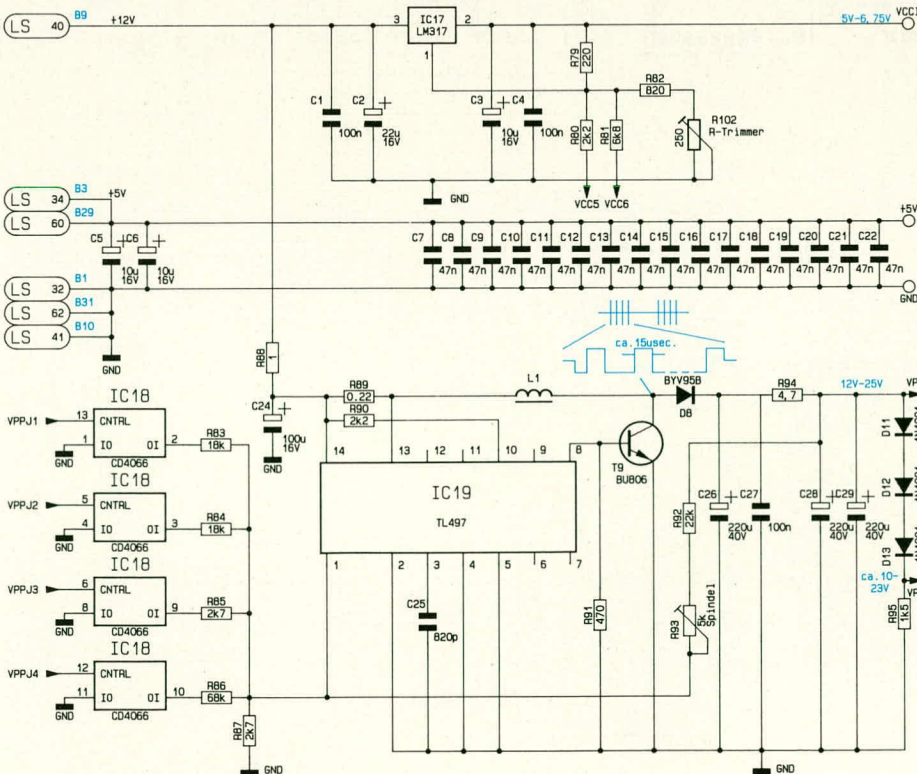
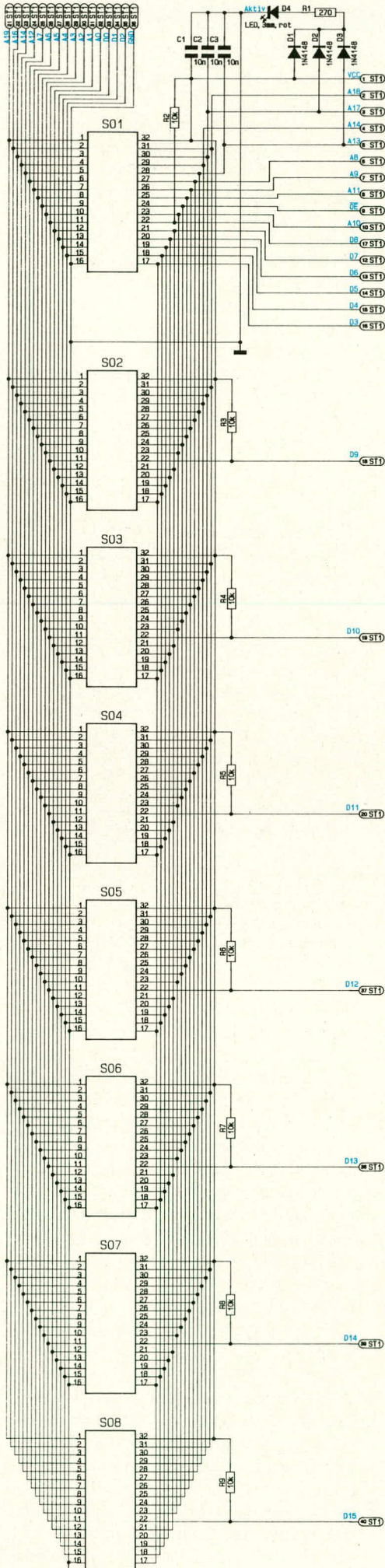


Bild 6: Spannungsgenerator und  $V_{cc}$ -Umschaltung

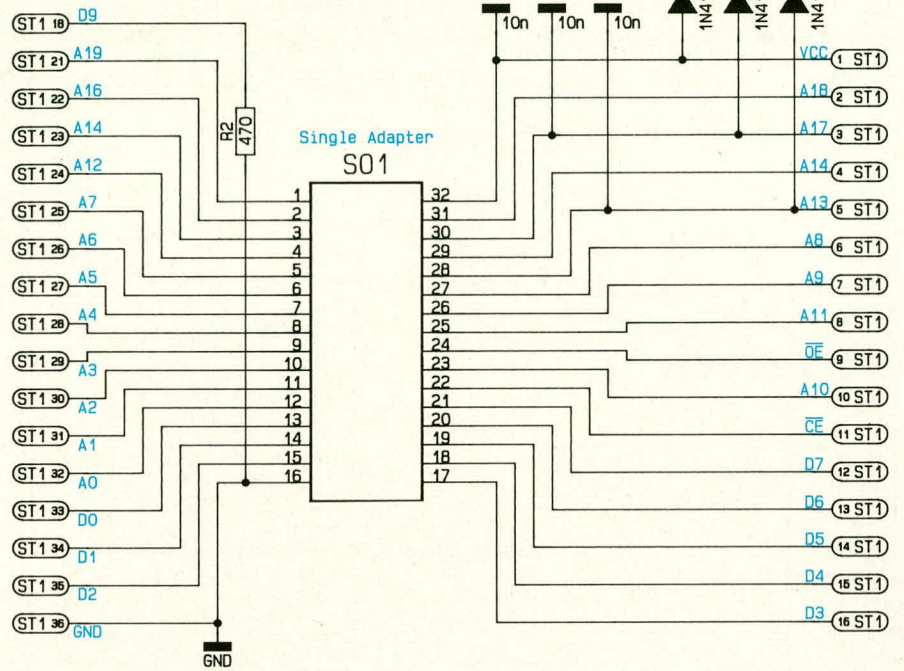
### Spannungsgenerator (Bild 6)

Die Programmierspannung im Bereich von 12 V bis 25 V kann nicht ohne weiteres vom im PC vorhandenen Netzteil abgeleitet werden, denn die höchste vorhandene Spannung mißt hier +12 V. Die Programmierspannung wird daher über einen Sperrwandler, aufgebaut mit IC 19 (TL 497), L 1, T 9 und D 8, auf der Karte generiert. IC 18 dient dabei als Umschalter für die verschiedenen Spannungen.

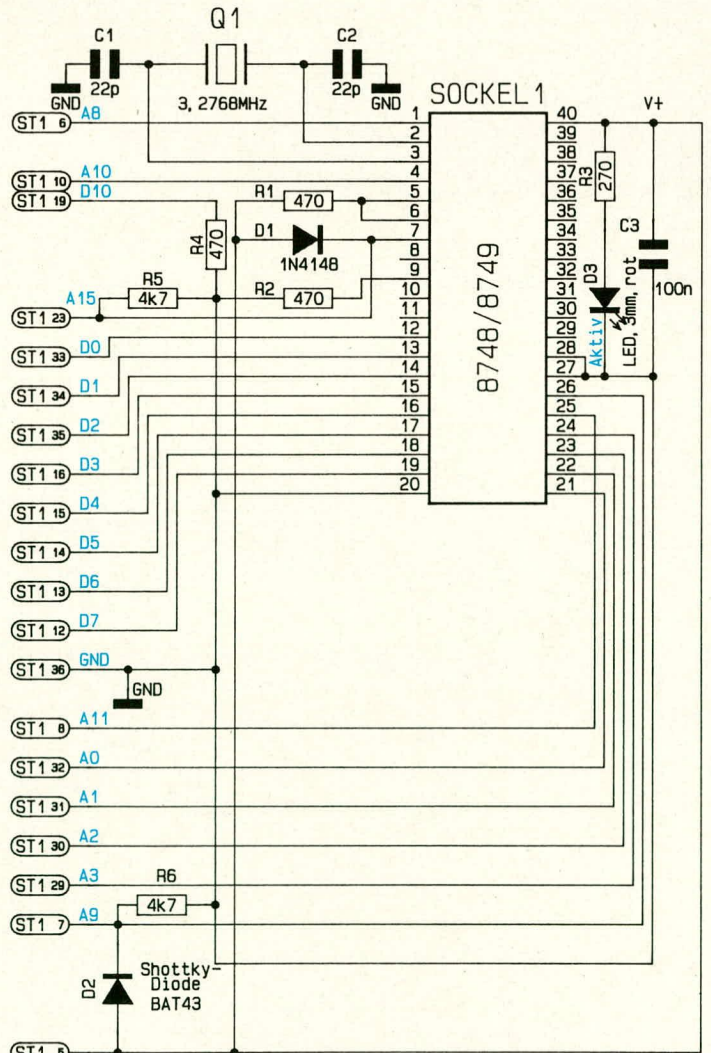
Eine oft unerwünschte Funktionseigenschaft der bei Elektronikern recht ungeliebten Induktivitäten kommt hier hervorragend zur Geltung, nämlich deren Bestreben, den in ihnen herrschenden Stromfluß aufrecht zu erhalten (Induktionsgesetz). Kommt es zum Zusammenbruch des Stromflusses, so entsteht an den Anschlüssen der Spule eine Spannungsspitze entgegengesetzter Polarität. Gehen wir im vorliegenden Fall davon aus, daß L 1 stromlos war.



**Bild 7 (unten):  
Schaltbild für  
Single-EPROM-Adapter**



**Bild 8  
(links):  
Schaltung des 8fach-  
Multi-  
Eprom-  
Adapters  
(Gang-  
Adapter)**



**Bild 9  
(rechts):  
Schaltung des  
Adapters  
für Mikro-  
controller  
Serie  
8748/8749**

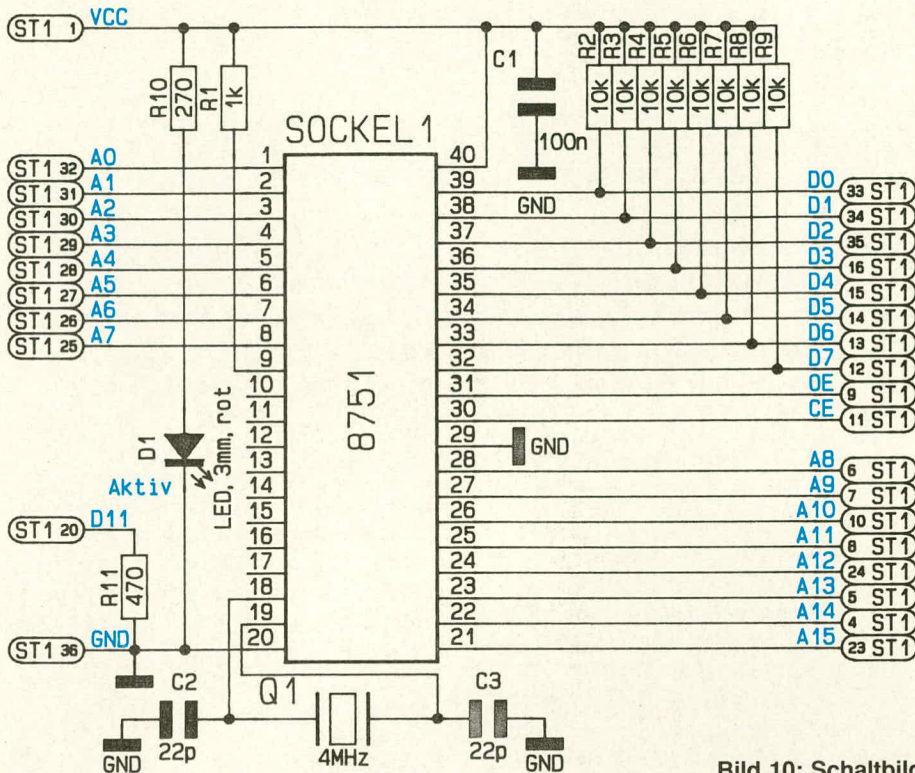


Bild 10: Schaltbild des Adapters für Mikrocontroller der Serie 8751

T9 wird nun eingeschaltet, und es fließt ein Strom, dessen Dauer durch C 25 an IC 19 bestimmt wird. Dadurch hat die Induktivität Gelegenheit, ein Magnetfeld aufzubauen. Wird T 9 abgeschaltet, so verharrt das Magnetfeld, und es baut sich eine positive Spannung am Kollektor von T 9 auf. Da der nun negative Pol bereits an +12 V liegt, addiert sich die Spulenspannung dazu, die nun über D 8 in den Speicherelko C 26 fließen kann. Dieser Vorgang läuft mit etwa 10 kHz ab.

Im Gegensatz zu den meisten Spannungswandlerbausteinen steuert der TL 497 nicht die Einschaltdauer des Transistors und hält dabei die Frequenz konstant, sondern hier wird die Einschaltdauer konstantgehalten und die Frequenz variiert. Bei kleineren und recht konstanten Lasten hat das i. a. den Vorteil, daß die Induktivität stets optimal geladen werden kann. Damit sind auch kaum Regelschwingungsprobleme zu erwarten.

R 92, R 93 und R 87 nehmen eine Spannungsteilung vor, deren Ausgangswert in IC 19 (über Pin 1) mit einer hochkonstanten Referenzspannung verglichen wird, wonach dann das Puls-Pausen-Verhältnis von T 9 gesteuert wird. Mit IC 18, Pin 2, 3, 9 und 10 können noch Parallelwiderstände an diesen Spannungsteiler geschaltet werden, so daß  $V_{pp}$  auf 12 V, 12,5 V, 21,5 V, 23 V und 25 V eingestellt werden kann. D 11 bis D 13 sowie R 95 bilden eine Grundlast und bieten eine Hilfsspannung zur Programmierung der Reihe 8748.

Speiseleitung (Leitungen  $V_{cc}$ : A 17 und A 13) versorgt und weist auf den nicht spannungsfreien Adapter hin.

Der Gang-Adapter (Bild 8) enthält 8 nebeneinander angeordnete Textool-Sokkel. Alle Leitungen außer  $\overline{CE}$  (zur Selektion der einzelnen EPROMs) sind parallelgeschaltet.

Der Adapter für Serie 8748 (Bild 9) besitzt als Hauptmerkmal eine Oszillatorschaltung mit einem Quarz 3,2768 MHz, da diese Controller dynamisch arbeiten und somit auch beim Programmieren ein Takt anliegen muß. R 1 und R 2 initialisieren den Baustein, zusammen mit entsprechenden Pegelvorgaben an den Adreßleitungen. D 1 und D 2 legen +5 V (über A 13) an Pin 7 und Pin 26, wenn nicht programmiert wird.

Auch der Adapter für Serie 8751 (Bild 10) benötigt einen Takt, daher ist hier ein 4 MHz-Quarz vorgesehen. Im Gegensatz zum Adreßblatt des 8748/8749 müssen hier die Adressen direkt angelegt werden.

**Abgleich der Trimmer R 102 und R 93**

Nach Aufbau und Inbetriebnahme der Karte wird ein Programm ABGLEICH von der Diskette gestartet. Damit kann an Pin 32 des EPROM-Single-Adapters  $V_{cc}$ , an Pin 1 des Adapters  $V_{pp}$  gemessen werden. Bezugspunkt ist hierzu Pin 16 (GND). Der Abgleich von  $V_{cc}$  wird mit R 102, von  $V_{pp}$  mit R 93 durchgeführt. Danach kann sofort mit dem Einsatz des Programmierers begonnen werden.

Im zweiten, abschließenden Teil dieses Artikels stellen wir Ihnen Nachbau und Inbetriebnahme ausführlich vor. **ELV**

**Adapter-Sockel (Bild 7 bis 10)**

Auf den Adaptern selbst, die jeweils mit einem optimierten, standfesten Ganzmetallgehäuse ausgeliefert werden, sind neben einer Steckerleiste zum Aufstecken des 40poligen Flachbandkabels ein Textool-Stecksockel für den zu bearbeitenden Speicherbaustein und eine LED vorhanden. Diese wird über Dioden aus der jeweils aktiven

Tabelle 4: Anschlußbelegung der verschiedenen EPROM-Typen

																				+5 V										
																				PGM	PGM	A18	A18							
A19	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>							1																				
A16	A16	A16	A16							2																				
A15	A15	A15	A15	A15	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>			3	1																			
A12	A12	A12	A12	A12	A12	A12	A12			4	2																			
A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	5	3	1	24	26	28	30					+5 V	+5 V	n.c.	A 13	A13	A13	A13	A13	A13	A13
A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	6	4	2	23	25	27						A8	A8	A8	A8	A8	A8	A8	A8	A8	A8
A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	7	5	3	22	24	26						A9	A9	A9	A9	A9	A9	A9	A9	A9	A9
A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	8	6	4	21	23	25						V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>	V <sub>pp</sub>
A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	9	7	5	20	22	24						OE	OE <sub>np</sub>	OE	OE	OE	OE <sub>np</sub>	OE	OE	OE	OE <sub>np</sub>
A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	10	8	6	19	21	23						A10	A10	A10	A10	A10	A10	A10	A10	A10	A10
A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	11	9	7	18	20	22						CE	CE	CE	CE	CE	CE	CE	CE	CE	CE
A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	12	10	8	17	19	21						D7	D7	D7	D7	D7	D7	D7	D7	D7	D7
D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	13	11	9	16	18	20						D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	14	12	10	15	17	19						D5	D5	D5	D5	D5	D5	D5	D5	D5	D5
D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	15	13	11	14	16	18						D4	D4	D4	D4	D4	D4	D4	D4	D4	D4
Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	16	14	12	13	15	17						D3	D3	D3	D3	D3	D3	D3	D3	D3	D3
																				2 KB										
																				4 KByte										
																				8 KByte										
																				16 KByte										
																				32 KByte										
																				64 KByte										
																				128 KByte										
																				256 KByte										
																				512 KByte										
																				1MByte										