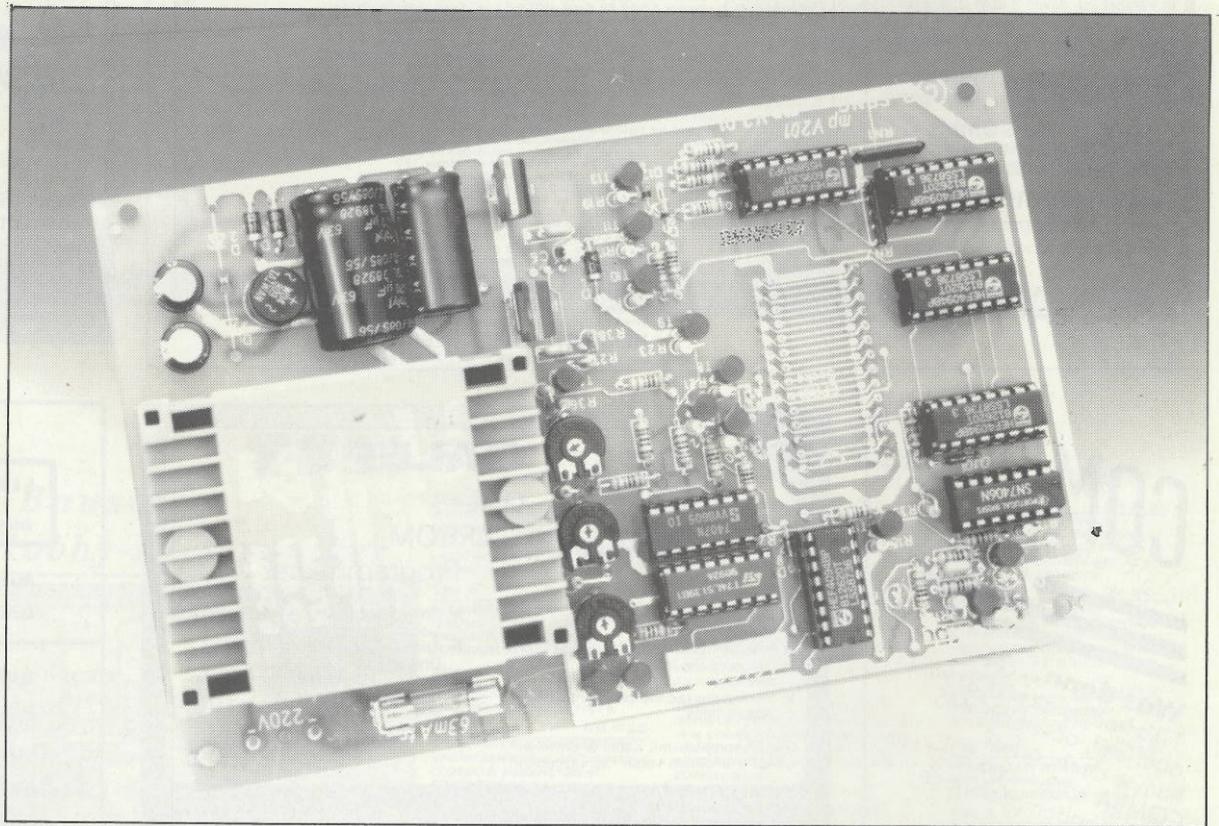


CENTRONICS- EPROMMER

Peter Seng

Ein EPROM-Programmiergerät sollte möglichst viele EPROM-Typen programmieren können und mit handelsüblichen Bauteilen preiswert zu bauen sein. Im Idealfall arbeitet es auch noch an jedem Computer. Die Schaltung? Siehe unten. Außerdem gibts für IBMs und Kompatible eine menü-gesteuerte Software, die keine Wünsche mehr offen läßt.



**brennt
2716 bis
27512**

Es gibt zwei Kategorien von Eprommern: Im professionellen Bereich findet man häufig Stand-alone-Geräte mit eigenem Prozessor und einer Schnittstelle zur Datenübernahme. Üblicherweise setzt man sie zum Duplizieren von Master-Eproms ein, deshalb sind sie auch mit mindestens zwei, teilweise mit bis zu neun Programmierfassungen ausgestattet, damit gleichzeitig mehrere EPROMs kopiert werden können.

Geräte für Software-Entwickler sind grundlegend anders aufgebaut. Gerade die teure Nullkraft-Fassung ist nur einmal vorhanden, und die Elektronik ist auf Signalpufferung, Adressierung und Programmierspannungserzeugung für das

Hardware:

- programmiert EPROMs vom Typ 2716 bis 27512/27513
- Programmierspannung 12,5/21/25 V
- keine Spezialbauteile
- kurzschlußfest
- computerunabhängiger Betrieb an der Centronics-Schnittstelle
- belegt **keinen** internen Slot
- eigene Stromversorgung

Software:

- für XT-/AT-kompatible Rechner (MS-DOS)
- automatische Anpassung an Grafik-Karte und Rechnertakt
- voll Menü-gesteuert
- Makroprogrammierung
- Farbgrafik
- automatische EPROM-Typerkennung
- integrierte Dateiverwaltung
- integrierter Hex/ASCII-Editor
- Standard- und schnelle 1-ms-Algorithmen
- 16-Bit-Programmierung (High/Low-Byte) möglich
- diverse Utilities

EPROM beschränkt. Die eigentliche Steuerung übernimmt ein PC.

Unser Eprommer fällt in Kategorie zwei. Durch die einfachere Schaltung halten sich Preis und Aufwand in Grenzen, während an der Leistungsfähigkeit keine Abstriche gemacht werden müssen.

Die Grundidee der Programmierschaltung geht auf einen Entwurf für den Apple-II aus der Computerzeitschrift MC August '84 zurück. Die Urversion eines PC-Eprommers des Autors wurde in MC Januar '87 abgedruckt. Gegenüber dem damaligen Stand stellt die neue Ausführung in einigen wichtigen Punkten eine wesentliche Verbesserung dar: die Zusammenarbeit mit Computern vom Typ Laptop funktioniert jetzt ohne Schwierigkeiten. Das ehemals einseitige Platinenlayout, das mit einer Menge Drahtbrücken zu bestücken war, ist jetzt doppelseitig ausgeführt. Das erleichtert unter anderem die Bestückung und die Befestigung der Programmierfassung ganz erheblich. Die Software ist in der jetzt erhältlichen Version 3.0 wesentlich komfortabler und übersichtlicher aufgebaut.

Mit der Schaltung lassen sich fast alle erhältlichen EPROM-Typen vom 2716 bis zum 27512/27513 programmieren. Die Auswahl des EPROM-Typs erfolgt rein softwaremäßig, mit Ausnahme des mittlerweile etwas in die Jahre gekommenen 2k·8-Chips 2716. Zu dessen Programmierung ist noch von Hand ein Schalter zu bedienen.

Als Schnittstelle zwischen steuerndem Computer und Programmier haben wir uns für die Standard-Parallelschnittstelle entschieden. Damit besteht nicht nur die Möglichkeit, das Gerät an jedem Rechner mit Centronics-kompatibler Parallelschnittstelle zu betreiben, auch die Belegung eines internen Slots mit einer Interfacekarte entfällt. *Software gibts allerdings nur für IBM-Kompatible.*

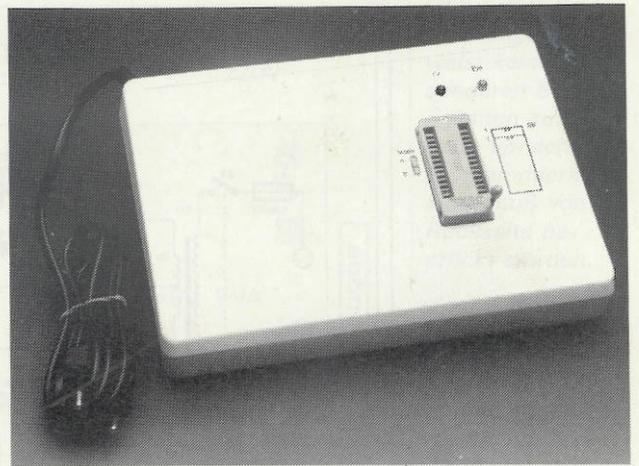
Die Programmierung

eines EPROMs läuft im Prinzip folgendermaßen ab: Zuerst ist die korrekte Versorgungsspannung von 5 V oder 6 V und die richtige Programmierspannung (12,5 V/21 V/25 V) einzustellen. Anschließend muß mit den Adreßleitungen die gewünschte Adresse angewählt werden, und an den Dateneingängen muß das zu programmierende Bitmuster anliegen. Ein Impuls genau definierter Länge am Program-

miereingang bewirkt die Übernahme des Datenbytes in das EPROM. Nach der vollständigen Programmierung liest man die programmierten Bytes üblicherweise wieder aus und vergleicht sie mit den Originaldaten. Wenn der Vergleich positiv ausfällt, ist die Programmierung beendet.

Eines der wichtigsten Kriterien bei der Programmierung ist die Festlegung der Programmierimpulsdauer. Ältere Programmieralgorithmen arbeiteten generell mit einer Impulsdauer von 50 ms. Das klingt zunächst nicht nach viel, führte aber mit zunehmend größer werdenden EPROMs zu immer längeren Programmierzeiten, beispielsweise fast 7 Minuten für ein 2764. Moderne Eprommer (und EPROMs) beherrschen schnellere Programmierverfahren (2764: etwa 35 s). Hier liegt die Dauer des Programmierimpulses nicht fest, sondern sie ergibt sich aus der Programmierung selbst (sogenannte interaktive Algorithmen). Die Dauer des ersten Impulses ist auf 1 ms beschränkt, anschließend wird sofort ausgelesen. Falls der Inhalt der Speicherzelle noch nicht korrekt ist, wird erneut 1 ms programmiert und anschließend wieder ausgelesen. Die Anzahl der zur erfolgreichen Programmierung erforderlichen Versuche wird registriert. Wenn beispielsweise drei 1-ms-Impulse nötig waren, folgt eine abschließende *Überprogrammierung* mit der dreifachen Dauer (9 ms). Die Zahl der Versuche ist üblicherweise auf 15 oder 25 beschränkt. Wenn sich damit kein Erfolg eingestellt hat, kann man sicher annehmen, daß das EPROM defekt ist.

Es gibt verschiedene schnelle Programmieralgorithmen, die sich im wesentlichen durch die Anzahl der Wiederholungen der 1-ms-Impulse und die Dauer der Überprogrammierung unterscheiden. Im Gegensatz zum Standardalgorithmus arbeiten die schnellen Verfahren auch mit einer auf 6 V erhöhten Versorgungsspannung für das EPROM. Der IC-Hersteller schreibt zu den verschiedenen EPROM-Typen die Programmiermethode vor. Daran sollte man sich auch halten, nicht nur um die Langzeitstabilität der Daten zu gewährleisten, sondern auch um das EPROM nicht zu gefährden. Der Versuch, einen älteren EPROM-Typ mit einem superschnellen Algorithmus zu programmieren, hat schon häufiger mit Totalverlust geendet.



Prinzipielles

Die Schaltung benutzt die Centronics-Parallelschnittstelle zur bidirektionalen, seriellen Datenübertragung. Das ist nur möglich, weil der Centronics-Druckerausgang eines IBM-kompatiblen Rechners auch über einige *Eingangslinien* verfügt, beispielsweise PE (Paper empty), über die ein angeschlossener Drucker ein Signal zum Computer schicken kann, wenn das Papier ausgegangen ist.

Von den acht Ausgangsleitungen D0...D7 der Schnittstelle werden nur die ersten drei verwendet: Eine als Datenausgang (Out), eine als Taktsignal (Clock), und eine weitere als Strobesignal (Strobe). Über den Eingang PE läuft die serielle Information vom Eprommer zum Computer zurück. Das Übertragungsprotokoll ist so einfach wie möglich gehalten. Mit jeder positiven Flanke des Clock-Signals wird der momentane Pegel der In-Leitung in ein 32-bit breites Schieberegister (Seriell → Parallel) übernommen und durchgeschoben. Nach der Übertragung von 4 Byte (32 bit) bewirkt der Strobe-Impuls das Durchschalten der im Schieberegister gespeicherten Daten an die 32 Registerausgänge. Die ersten 16 Bit (Byte 1, Byte 2) liegen direkt an der EPROM-Fassung und übernehmen dort die Funktion der Datenleitungen D0...D8 beziehungsweise der Adreßsignale A0...A7. Von den nächsten 16 Bit (Byte 3, Byte 4) werden die restlichen Adreßsignale A8...A13 und sämtliche Steuersignale für die Spannungseinstellung, -umschaltung und zur EPROM-Selektierung abgeleitet. Um den EPROM-Inhalt auch seriell wieder auslesen zu können, ist ein weiteres 8-bit-Schieberegister (Parallel → Seriell) erforderlich, das mit den Datenanschlüssen D0...D8 der EPROM-Fassung verbunden ist. Es setzt das Daten-

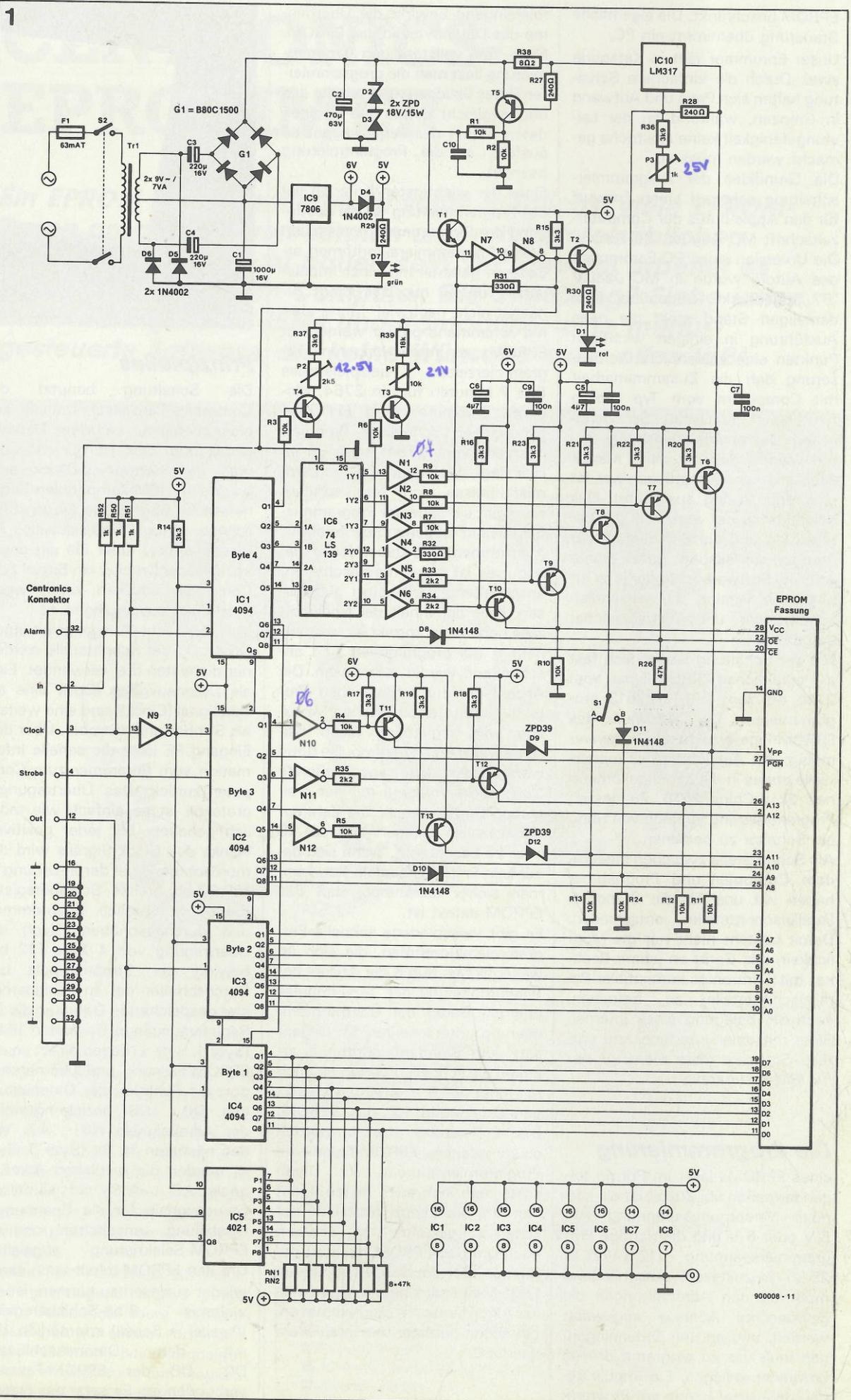


Bild 1. Im Schaltbild ist die zentrale Funktion des 32-bit-Schieberegisters mit IC1...IC4 zu erkennen.

900008 - 11

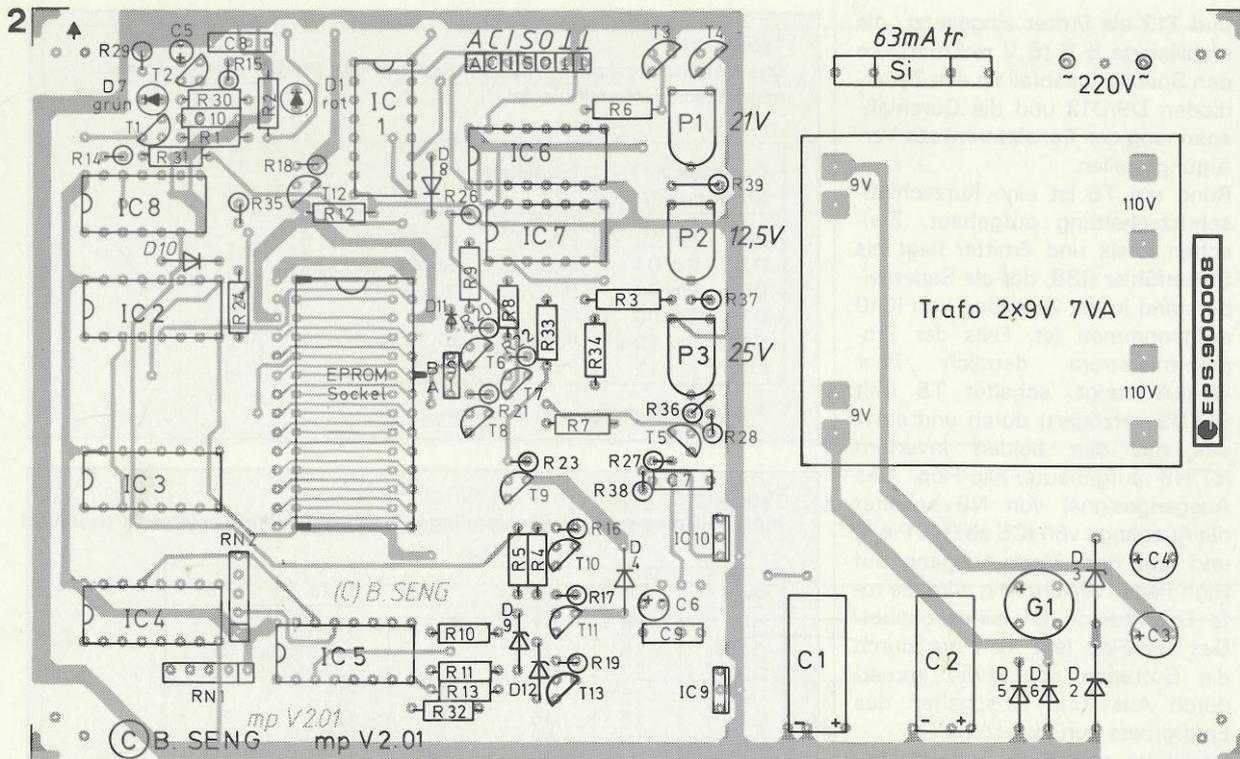


Bild 2. Der Bestückungsplan weist keine besonderen Schwierigkeiten auf, doch Vorsicht, die Programmierfassung muß von der Rückseite bestückt werden.

byte in ein serielles Bitmuster um, das über Pin 12 der Centronics-Schnittstelle zum Rechner gelangt. Auch für die Ausleseprozedur dienen Clock und Strobe wieder als Steuersignale.

Genaugenommen braucht man also für den Anschluß des Programmiergeräts noch nicht mal einen vollständigen Centronics-Anschluß, sondern nur einen 4-bit-Port mit drei Ausgängen und einem Eingang.

Die Schaltung

Ist nach der vorangegangenen Erklärung leicht zu verstehen. Ganz links im Schaltbild (Bild 1) ist der Centronics-Schnittstellenstecker gezeichnet. Gleich daneben ist das mit IC1...IC4 aufgebaute 32-bit-Schieberegister angeordnet. Die seriellen Daten vom Computer liegen direkt am Dateneingang Pin 2 von IC1. Über die Datenausgänge (Q_S, Pin 9) und -eingänge (Pin 2) sind alle vier 4094er kaskadiert. Die ersten 8 Bit die der Computer zum Eprommer schickt werden in IC1 gespeichert. Das zweite Datenbyte ersetzt den Inhalt von IC1, das die erste Bitfolge an IC2 weitergibt. Das Ganze geht so weiter, bis 4 Byte übertragen sind. Byte 1 ist dann in IC 1, Byte 2 in IC2 gespeichert und so weiter. Daher erfolgt die Datenübertragung vom Computer zum Programmiergerät auch dann mit 4-Byte-Paketen, wenn nur der Zustand eines einzigen Bits im er-

sten Byte zu ändern ist. Die Bedeutung der Bytes 1 und 2 und der Bits Q₈...Q₆/Q₄ (Byte 3) als Datenbeziehungsweise Adreßsignal ist im Schaltbild leicht zu erkennen. Etwas unübersichtlicher sieht es mit den restlichen zwölf Bits aus. Die Pin-Nummern und die Bezeichnungen an der Fassung beziehen sich übrigens alle auf 28-polige EPROMs.

Doch zuerst zur Spannungsversorgung. Der Transformator liefert zweimal 9 V, aus denen gleich fünf verschiedene Spannungen abgeleitet werden. IC 9 liefert eine stabilisierte Spannung von 6 V, die D4 auf etwa 5 V reduziert. Aus der 5-V-Leitung werden sämtliche Logik-ICs, das EPROM und die Einschaltanzeige R29 und D7 versorgt. IC10 stabilisiert die mit P3 einstellbare Programmierspannung auf 25 V. Parallel zu R36 und P3 liegen R37/P2/T4 und R39/P1/T3. Wenn einer der beiden Transistoren T3 oder T4 durchgeschaltet ist, liegt das zugehörige Potentiometer am Adjust-Eingang des LM 317. Die Ausgangsspannung von IC 10 ist dann auch von der Einstellung von P2 beziehungsweise P1 abhängig. P1 ist für die Einstellung der Programmierspannung von 21 V, P2 für 12,5 V zuständig. Die Transistoren T6...T8 schalten die eingestellte Spannung an den richtigen Pin der EPROM-Fassung, wobei der Weg des Ausgangssignals von T8 von der Stellung des Umschalters

S1 abhängig ist.

S1 ermöglicht die Umschaltung der Programmierspannung (für 24-polige EPROMs) von Pin 23 auf Pin 24 (A9) der Nullkraft-Fassung. Wenn die Programmierspannung an der Adreßleitung A9 anliegt und gleichzeitig A1...A7 Low-Pegel führen, können mit A1 zwei spezielle Speicherzellen im EPROM adressiert werden, die eine Hersteller- und eine Typ-Kennzeichnung enthalten. Mit dieser, nur in neueren ICs enthaltenen *Signatur* läßt sich der optimale Programmieralgorithmus ermitteln. Wenn S1 in Stellung B steht, kann allerdings der EPROM-Typ 2716 nicht mehr programmiert, sondern nur noch ausgelesen werden, da Pin 23 als Programmierimpulseingang fungiert. Statt des ungleich höheren Aufwands, die Handbetätigung des Schalters durch eine Hardwarelösung zu ersetzen, haben wir diesen kleinen Nachteil in Kauf genommen, da 2716er ohnehin immer seltener zum Einsatz kommen. Die Transistoren T9 und T10 (T12 für 24-polige ICs) schalten die Versorgungsspannung V_{CC} an die EPROM-Fassung. Dabei hat man die Wahl zwischen +5 V mit T9 oder +6 V, wenn T10 durchgeschaltet ist.

An den Anschlüssen 1 (V_{pp}) und 23 (A11) besteht bei einigen Betriebszuständen ein erhöhter Strombedarf, den ein TTL-Signal nicht liefern kann. Daher sind T11

Stückliste

Widerstände:

R1...R13 = 10 k
R14...R23 = 3k3
R24,R26 = 47 k
R27...R30 = 240 Ω
R31,R32 = 330 Ω
R33...R35 = 2k2
R36,R37 = 3k9
R38 = 8Ω
R39 = 18 k
R50...R52 = 1 k
RN1,RN2 = SIL-Netzwerk 4·47 k

Kondensatoren:

C1 = 1000 µ/16 V radial
C2 = 470 µ/16 V radial
C3,C4 = 220 µ/16 V radial
C5,C6 = 4µ/16 V radial
C7...C10 = 100 n keramisch

Halbleiter:

D1 = LED 3 mm rot
D2,D3 = Z-Diode 18 V/0,5 W
D4...D6 = 1N 4002
D7 = LED 3 mm grün
D8,D10,D11 = 1N 4148
D9,D12 = Z-Diode 39 V/0,3 W
G1 = B80 C1500
T1...T4 = BC 546
T5...T13 = BC 327
IC1...IC4 = 4094
IC5 = 4021
IC6 = 74LS139

IC7 = 7407
 IC8 = 7406
 IC9 = 7806
 IC10 = LM 317

Außerdem:

- S1 = Umschalter, einpolig
- S2 = Netzschalter, zweipolig
- F1 = Schmelzsicherung 63 mA träge mit Platinenhalterung
- Tr1 = Flachtrafo 2 x 9 V/7 VA
- 28-poliger-Textool-Sockel
- 228-3345-00-0605 Centronics-Stecker

und T13 als Puffer eingesetzt, die stabilisierte 5 V (6 V reduziert um den Spannungsabfall an den Zenerdioden D9/D12 und die Durchlaßspannung der Transistoren) zur Verfügung stellen.

Rund um T5 ist eine Kurzschlußschutzschaltung aufgebaut. Zwischen Basis und Emitter liegt als Stromfühler R38, der als Serienwiderstand in die Zuleitung von IC10 aufgenommen ist. Falls der Programmierstrom deutlich über 50 mA steigt, schaltet T5 (mit R1/C19 verzögert) durch und setzt das mit den beiden Invertiern N7/N8 aufgebaute Flip-Flop. Das Ausgangssignal von N8 schaltet die Ausgänge von IC6 ab (1G, Pin 1) und legt den Alarm-Ausgang auf High-Pegel. Gleichzeitig wird die rote Leuchtdiode D1 eingeschaltet. Das Flip-Flop läßt sich nur durch die Software oder (wie immer) durch Aus- und Einschalten des Eprommers zurücksetzen.

Damit ist eigentlich auch die Bedeutung der restlichen Bits der vier Steuerbytes geklärt: hiermit werden die diversen Transistoren gesteuert. Eine Ausnahme bilden noch die fünf Bits Q2...Q6 von Byte 4. Q2 bis Q5 liegen an den Eingängen des 2-bit-Binärdekoders IC6, der insgesamt sieben mögliche Bitmuster dekodiert. Q6 liegt direkt am Ausgangs-Freigabeeingang von IC4. Der 4094 braucht eine logische 1 an Pin 15, damit das gespeicherte Bitmuster an den Ausgängen erscheint. Mit einem Low-Pegel bleiben die Ausgangsleitungen hochohmig. Dieser Zustand wird mit Q6 eingeschaltet, wenn das EPROM ausgelesen werden soll.

Die Bedeutung aller Bits ist in Tabelle 2 noch einmal zusammengefaßt.

Die Mechanik

und die Bestückung der Platine sind nicht weiter schwierig. Alle Bauteile werden wie gewohnt bestückt und verlötet, lediglich die Nullkraft-Fassung kommt auf die Lötseite der doppelseitigen, durchkontaktierten Platine. Die große, (leider) rechteckige Ausparung für die Fassung läßt sich mit Bohrer und Feile herstellen. Die Platine wird mit der Lötseite und der Fassung nach oben im Gehäusedeckel befestigt. Die restlichen Löcher sind wieder rund: zwei für die beiden Leuchtdioden, eins für den Schalter S1 und zwei für die Durch-

Tabelle 1

Das Verbindungskabel zwischen Rechner und Programmiergerät muß mindestens sechsadrig sein.

Centronics-Schnittstelle		Eprommer	
PE	Pin 12	Pin 12	Out (O)
D2	Pin 04	Pin 04	Strobe (S)
D0	Pin 02	Pin 02	In (I)
D1	Pin 03	Pin 03	Clock (C)
Gnd	Pin 18-25	Pin 16, 19-30, 33	Gnd
Error	Pin 15	Pin 32	Alarm* (A)

*die Error-Leitung kann entfallen.

Tabelle 2

Mit vier Bytes werden alle Vorgänge im EPROM-Programmiergerät gesteuert

Byte 1	Pin	Byte 2	Pin	Byte 3	Pin
Q1	19	Q1	3	Q1	1
Q2	18	Q2	4	Q2	27
Q3	17	Q3	5	Q3	26
Q4	16	Q4	6	Q4	2
Q5	15	Q5	7	Q5	23
Q6	13	Q6	8	Q6	21
Q7	12	Q7	9	Q7	24
Q8	11	Q8	10	Q8	25

Byte 4	Pin
Q1	High: 12,5 V _{pp} /Low: 21 V _{pp}
Q2	*
Q3	*
Q4	*
Q5	*
Q6	High: Lesen/Low: Schreiben
Q7	22
Q8	20

* Mit S1 und Q1...Q5 wird die Betriebsart festgelegt. Die Zuordnung ist in der unteren Tabelle separat angegeben.

S1	Q1	Q2	Q3	Q4	Q5	Funktion
B	H	H	H	X	L	12 V AN A9 (Identifizier-Mode)
A	X	H	H	X	X	Pin 23 V _{pp}
X	X	H	L	X	X	Pin 1 V _{pp}
X	X	L	H	X	X	Pin 22 V _{pp}
X	X	L	L	X	X	V _{pp} ausgeschaltet
X	L	X	X	H	H	25 V V _{pp}
X	X	X	X	H	L	Pin 28 5 V V _{cc}
X	X	X	X	L	H	Pin 28 6 V V _{cc}
X	X	X	X	L	L	Pin 28 0 V/Schutzschaltungsreset

H: High-Pegel; L: Low-Pegel; X: beliebig

führung von Netz- und Schnittstellenkabel. Die Anschlüsse auf der Platine sind im üblichen Rastermaß ausgeführt. Wenn man Steckverbinder verwenden will, hat man mit einreihigen Pfostenfeldsteckern und -buchsen keine Probleme. Für den 230-V-Anschluß nimmt man am besten eine dreipolige Lötlüsterklemme. Auf eine solide Zugentlastung des Netzkabels ist trotzdem zu achten.

Die Software

beschränkt sich nicht nur auf das eigentliche Steuerprogramm, sondern umfaßt auch noch einige Hilfsdateien. Alle Programme sind in Turbo-Pascal Version 4.0 erstellt. Nachdem der Programmierer mit dem Druckerkabel an den Rechner angeschlossen und eingeschaltet ist, startet man zuerst das Programm MP.exe. Es ermittelt die Schnittstel-



Bild 3. Die Bildschirmdarstellung des Eröffnungsmenus.

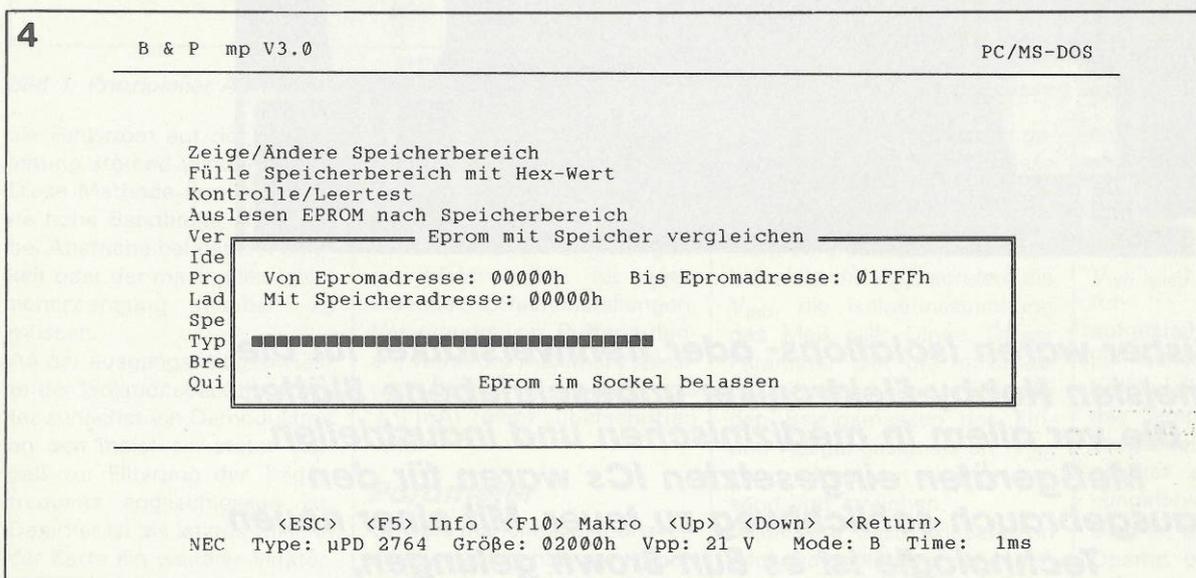


Bild 4. Der Stand der Dinge wird bei allen Schreib- und Lesevorgängen jederzeit sichtbar gemacht.

le, an die der Programmierer angeschlossen ist (LPT1, LPT2 oder LPT3), die Art und Taktfrequenz der CPU im Computer und den aktiven Bildschirmadapter. Daraufhin konfiguriert sich das Programm selbständig; ein wie auch immer geariteter Setup ist also nicht erforderlich.

Die Bedienung ist auf zweierlei Arten möglich: ein Menüpunkt wird durch Eingabe des Anfangsbuchstabens, oder durch Positionierung eines farbigen Balkens mit den Cursorstasten ausgewählt und durch die Return-Taste bestätigt. Die verschiedenen Menüpunkte erklären sich eigentlich von selbst:

- Zeige/Ändere Speicherbereich
- Fülle Speicherbereich mit HEX-Wert
- Kontrolle/Leertest
- Identifiziere EPROM
- Programmiere mit Speicherbereich
- Lade Datei in Speicherbereich
- Speichere Datei von Speicherbereich

- Typwahl EPROM
- Brenne Security-Bit
- Quit

Es ist lediglich darauf zu achten, ob vor der Programmierung oder dem Auslesen eines EPROMs der korrekte EPROM-Typ eingestellt ist. Schreib- oder Lesevorgänge werden am Bildschirm visualisiert, so daß man sie jederzeit mitverfolgen kann.

Im Steuerprogramm ist ein komfortabler Bildschirmeditor enthalten, mit dem ausgelesene EPROM-Dateien oder auf Diskette abgelegte Datenfiles schnell und einfach verändert werden können. Auf der Diskette sind verschiedene Dateikonvertierungsprogramme enthalten, beispielsweise, um HEX-Dateien in High- und Lowbyte aufteilen zu können (für den Einsatz von byte-wide organisierten EPROMs in 16-bit-Systemen). Das Programm **Exchange** konvertiert bestehende Dateien in die verschiedensten Datenformate. MP.EXE benötigt immer das Binär-

format, gegebenenfalls ist also zuerst eine Konvertierung durchzuführen. Die Syntax dazu lautet beispielsweise: Exchange Infile.hex Outfile.bin.

Makros erlauben es, Tastatureingaben in eine Datei abzuspeichern und selbstständig auszuführen. Zur Makroerstellung ist ein integrierter Editor vorhanden. Auf der Diskette sind drei Makrobeispiele, eine Batchdatei und verschiedene Testdaten gespeichert. Anhand dieser Beispiele ist die Makroprogrammierung leicht nachzuvollziehen. Mit der Batchdatei selbst kann man das Steuerprogramm aus der DOS-Ebene starten und gleichzeitig ein Makro aufrufen.

*Hinweis des Autors:
 Fragen zur Software und zur Hardware können direkt an den Autor gerichtet werden:
 B. Seng
 Ludwig-Dürr-Straße 10
 7320 Göppingen*