

# EPROM-Handler

Beschreibung zum EPROM-Programmierprogramm

EPROM-Handler V 2.9 (RAM) fuer Z 1013

by Rainer Brosig

Programmstandort: 100H - 8FFH Start: 100H >CRC=96DF

Hardware:

- Z 1013 / Z 9001
- EPROM-Programmiergeraet der IG-HC TU Dresden, nach electronica Bd.227/228 weiterentwickelt.
- in Leipzig (2. Z1013-Tagung) von der IG-HC TU Dresden vorgestellt.
- zusaetzliche PIO im Z 1013 fuer Programmiergeraet keine Interruptfaehigkeit gefordert. (Zusatzbaugruppe Robotron Riesa ab I/O-Adr. 30H)

I/O-Adressen:

```
Port A Daten : 30H
              SW  : 31H
Port B Daten : 32H
              SW  : 33H
```

Beschreibbare EPROM-Typen: 555, 2708, 2716, 2732, 2764, 27128, 27256, 27512 u.a.

Realisierte Funktionen:

- Typenauswahl durch Eingabe der Typenbezeichnung ohne fuehrende Buchstaben; die letzten zwei Zahlen sind signifikant
- Lesen des EPROM-Inhalts und Speichern auf RAM-Bereich; dabei Ausgabe der CRC-Pruefsumme
- Programmieren von beliebigem Speicherbereich, dabei automatischer Loeschtest und bei negativen Ergebnis Vertraeglichkeitstest; waehrend der Programmierung wird der laufende Programmierzyklus bzw. Sicherheitszyklus durch verschiedene Grafikzeichen angezeigt ; es werden so viele Zyklen ausgefuehrt bis alle Zellen fehlerfrei beschrieben sind (max. 100); daran schliessen sich halb so viele Sicherheitszyklen an, wie Programmierzyklen gebraucht wurden, jedoch mindestens 15 ; Zellen mit Inhalt FFH werden nicht mitprogrammiert, dadurch erfolgt Programmierung besonders schnell; nach Beendigung der Programmierung Ausgabe der CRC-Pruefsumme
- Vergleichen des EPROM-Inhalts mit gewaehltem Speicherbereich, dabei Ausgabe ungleicher Zellen mit deren Inhalt
- Loeschtest mit Ausgabe aller nichtgeloeschten Zellen und deren Inhalt; diese Funktion kann auch zum schnellen Betrachten des EPROM-Inhalts genutzt werden
- Wiederholen der zuvor ausgefuehrten Funktion mit gleichen Parametern
- Fortsetzen der vorher ausgefuehrten Funktion mit gleichen Parametern ausser der sich an den alten Bereich anschliessenden RAM-Adresse

Alle Funktionen koennen auf den gesamten EPROM-Speicherbereich sowie auf einen Teilbereich angewendet werden, wobei der kleinste Bereich die Einzelzelle darstellt; zusaetzlich erfolgt eine Bereichskontrolle entsprechend des gewaehlten EPROM-Typs. Aus allen Funktionen kann durch BREAK (CTRL-C,S4-K) auf die Menueebene zurueckgekehrt werden.

Durch BREAK kann auf der Menueebene zum Monitor zurueckgekehrt werden sowie durch Eingabe von N ein anderer EPROM-Typ gewaehlt werden. Sollte bei grossen EPROM-Typen der RAM zu klein sein, kann bereichsweise programmiert werden.

## Leipziger Version

Auf der Leipziger Tagung wurde eine neue Hardware mit integrierter PIO vorgestellt.

Die Umbenennung der Software erfolgte in EPROMMER 2.9/IGD und EPROMMER 2.9/CCL.

Diese beiden Versionen unterscheiden sich nur in den Port-Adressen für die PIO.

EPROMMER 2.9/IGD: Hardwarelösung Eprom-Programmiergerät der IG-HC der TU Dresden, nur in Verbindung mit dem E/A-Modul von Riesa und externer Programmierspannungserzeugung.

EPROMMER 2.9/CCL: Hardwarelösung Eprom-Programmiergerät Computer-Club robotron Leipzig. Dieses Programmiergerät hat die PIO und die Programmierspannungserzeugung auf der Leiterplatte integriert und ist direkt an den BUS oder den Baugruppenträger (Riesa) anschließbar. Ebenso besteht BUS-Kompatibilität für die Verwendung am KC85/1 bzw. KC87.

Es können die EPROM-Typen(bzw. der dazu Kompatiblen)

2716 / 2732(A) / 2764 / 27128 / 27256

programmiert werden.

Die Programmierspannungen sind 12,5V , 21V und 25V ; sie sind jedoch auch individuell beliebig einstellbar.

Auf Typen kleiner 2k-Speicherkapazität wurde wegen der Bedeutungslosigkeit verzichtet, dafür wurde bisher ein Programmiergerät vertrieben, welches nicht mehr produziert wird.

I/O-Adressen:

```
PORT A Daten - FCH
          St.W. - FDH
PORT B Daten - FEH
          St.W. - FFH
```

## PIN-Belegung

27512 2564 RAM/ROM PIN PIN 2708 2316 2616 6116 2532 2632 2364 6264 2564

2664 27128 27256 27512				U555 2716 6516				2732							
2764				2816											
-----															
-----															
A15	UPR	NC/UPR		1	28		-	-	-	-	-	UCC	UCC	UCC	
UCC	UCC	UCC	UCC												
A12	/CS1	A12		2	27		-	-	-	-	-	CS*	/WE	/CS2	
/PGM	/PGM	A14	A14												
A7	A7	A7		3	26		UCC	UCC	UCC	UCC	UCC	UCC	CS*	CS2	UCC*
N.C.	A13	A13	A13												
A6	A6	A6		4	25		A8	A8	A8	A8	A8	A8	A8	A8	
A8	A8	A8	A8												
A5	A5	A5		5	24		A9	A9	A9	A9	A9	A9	A9	A9	
A9	A9	A9	A9												
A4	A4	A4		6	23		UBB	CS*	UPR	/WE	UPR	A11	A11	A11	A12
A11	A11	A11	A11												
A3	A3	A3		7	22		/CS	/OE	/OE	/OE	PD/PGM/OE	/OE	/OE		
PD/PGM/OE	/OE	/OE	/OE												
A2	A2	A2		8	21		UDD	A10	A10	A10	A10	A10	A10	A10	
A10	A10	A10	A10												
A1	A1	A1		9	20		PR	/CE	/CE	/CE	A11	/CE	/CE	/CE	A11
/CE	/CE	/CE	/CE												
A0	A0	A0		10	19		D7	D7	D7	D7	D7	D7	D7	D7	
D7	D7	D7	D7												
D0	D0	D0		11	18		D6	D6	D6	D6	D6	D6	D6	D6	
D6	D6	D6	D6												
D1	D1	D1		12	17		D5	D5	D5	D5	D5	D5	D5	D5	
D5	D5	D5	D5												
D2	D2	D2		13	18		D4	D4	D4	D4	D4	D4	D4	D4	
D4	D4	D4	D4												
USS	USS	USS		14	15		D3	D3	D3	D3	D3	D3	D3	D3	
D3	D3	D3	D3												
-----															
UBB	USS	UCC	UDD					CS*							
-----															
-5V	0V	5V	12V	maskenprog. Selektionseingaenge (high-, low-, inaktiv moeglich)											

## Aufbauanleitung

Aufbauanleitung des EPROM-Programmiergerät der IG-HC TU Dresden (n.electronica 227/228)  
 © by Dr.U.Hinz 1989

## Einleitung

Obwohl in der letzten Zeit die Bedeutung von Festprogrammen wegen der weitverbreiteten Disketten zurueckgegangen ist, bleibt es vor allem fuer Computeramateure wichtig, EPROMs einzusetzen, zumal ihre Preise stark sinken und die Speicher- kapazitaet pro Schaltkreis steigt.

Um sich fuer ein Programmiergeraet entscheiden zu koennen, sind mehrere Auswahlkriterien zu beachten:

- Anzahl der zu programmierenden Typen (maximal)
- automatische Typerkennung
- Art und Umfang des Interfaces (minimal)
- Portabilitaet bezueglich der Rechnertypen (maximal)
- Softwareunterstuetzung (maximal)
- Preis-Leistungsverhaeltnis (optimal)
- Vorleistungen (maximal)

Bei den zu Programmiergeraeten erschienenen Veroeffentlichungen lassen sich zwei Kategorien unterscheiden. Die erste Kategorie umfasst Programmiergeraete, die auf einen Typ von EPROMs zugeschnitten sind /2/ /3/ /4/. Sie kommen fuer hoehere Ansprueche nicht in Betracht. In der zweiten Kategorie finden sich Beispiele, die den o. g. Kriterien besonders gut gerecht werden /1/ /5/ /6/. Hervorzuheben ist das Programmiergeraet nach /1/. Von V. LUEHNE wurde innerhalb der Z 1013-Gruppe der IG Heimcomputer am Informatikzentrum der TU Dresden eine Leiterplatte entwickelt. R. BROSIG hat ein sehr anspruchsvolles Programmierprogramm namens EPROM-HANDLER V 2.9 beigesteuert.

## Software zum Programmiergeraet

### Allgemeiner Aufbau des EPROM-HANDLER V 2.9 von R. BROSIG

Der EPROM-HANDLER V 2.9 ist ein komfortables, menueorientiertes, fuer den Z 1013 entwickeltes Werkzeug. Er belegt den Adressraum von 100H bis 08FBH, wird mit J 100 gestartet und ist selbst nicht EPROM-faehig. Das im Anhang befindliche HEX-Listing gibt das Programm vollstaendig wieder, wobei darauf zu achten ist, dass es mit einem Adress-Offset von 1000H ausgedruckt wurde.

Die vier I/O-Adressen:

PIO-Port A/Daten	30H	D0...D7
" Port A/Code	31H	
" Port B/Daten	32H	Steuersignale
" Port B/Code	33H	

entsprechen denen des PIO-Zusatzmoduls von robotron Riesa und ordnen sich in die Empfehlung fuer I/O-Adressen fuer den Z 1013 ein, welche auf der Nachfolgetagung der „1. Z 1013-Tagung“ bekanntgegeben wurde.

Der EPROM-HANDLER V 2.9 bietet dem Nutzer insgesamt sieben Funktionen, wie:

1. Typauswahl
2. Lesen
3. Programmieren
4. Vergleichen
5. Loeschtest
6. Wiederholen
7. Fortsetzen

an.

Kernstueck dieser Funktionen sind die Leseroutine (03FAH bis 0422H) und die Zellen-Programmerroutine (060BH bis 063EH). Sie erzeugen unter Nutzung der beschriebenen Hardware ein Signalspiel, das auf die international gebräuchliche EPROM-Familie 27xxx anwendbar ist. Bild 6 gibt einen Ueberblick fuer das Signalspiel der Zellen-Programmerroutine.

### Typauswahl

Nachdem man den EPROM-HANDLER V 2.9 mit J 100 gestartet hat, wird nach dem EPROM-Typ gefragt, was sich jedoch auf die EPROM-Groesse bezieht. Nicht die Typen einzelner Hersteller sind gemeint, denn diese unterscheiden sich meist nach dynamischen Kennwerten, der Programmierspannung oder der Betriebsstromaufnahme. Stellvertretend fuer die EPROM-Groesse werden deshalb die international ueblichen Endnummern:

08 fuer	1 KByte-EPROMs	1)
16 fuer	2 KByte-EPROMs	
32 fuer	4 KByte-EPROMs	
64 fuer	8 KByte-GPROMs	
128 fuer	16 KByte-EPROMs	
256 fuer	32 KByte-EPROMs	

zur Unterscheidung herangezogen.

In den Arbeitszellen 0128H und 0129H wird die Byteanzahl, die sich aus der EPROM-Groesse ergibt, abgelegt. Diese Byteanzahl ist die einzige Unterscheidungsmoeglichkeit fuer die EPROM-Groesse innerhalb der internen Ablaeufe des EPROM-Handlers. Sollen andere EPROM-Groessen durch den EPROM-Handler V 2.9 bearbeitet werden, so ist darauf zu achten, dass sie in ihrer Programmiervorschrift dem Signalspiel nach Bild 6 entsprechen. 1/2 KByte- und 64 KByte-EPROMs liessen sich also mit dem EPROM-HANDLER V 2.9 bei vorhandenem Typstecker (vgl. Bild 5.1...5.6) ebenfalls programmieren, doch werden sie im Typauswahldialog nicht beruecksichtigt.

Ehe man die eingegebene Endnummer durch ENTER gueltig macht, sollte man sich in jedem Falle noch einmal ueberzeugen, ob der richtige Typstecker gewaehlt wurde. Hat man irrtuemlicherweise eine falsche Groesse bestaetigt, dann laesst sich mit dem Kommando N (NEU) zur Typauswahl zurueckkehren.

### Lesen

Nachdem die Typauswahl erfolgt ist, bietet der EPROM-Handler ein „Funktionswahlmenue an. Hierin

steht die Kennziffer 1 fuer „Lesen“. Nachdem man die Kennziffer eingegeben hat, stellt der EPROM-Handler die Frage, ob man den steckenden EPROM vollstaendig oder teilweise lesen moechte. Mit ENTER entscheidet man sich fuer vollstaendiges Lesen und mit N (NEIN) fuer bereichsweises.

Nachdem man sich fuer das letztere entschieden hat, ist der interne EPROM-Adressbereich („rel. EPROM-Adresse (HEX):?“) anzugeben. Vor Fehleingabe, wie:

1. Anfangsadresse groesser als Endadresse und
2. Endadresse groesser als EPROM-Groesse,

besteht Schutz. Soll nur eine einzelne EPROM-ZELLE gelesen werden, dann sind die Angaben fuer Anfangs- und Endadresse gleich. Nachdem der einzulesende EPROM-Adressbereich eindeutig und fehlerfrei feststeht, bestaetigt man mit ENTER, sodass das Programm nach dem Ziel des EPROM-Inhaltes fragen kann. Obwohl in der Frage der Text „nach RAM-Adresse (HEX):?“ erscheint, besteht kein Schutz vor Angabe einer ROM-Adresse. Man koennte also als Ziel die Adresse angeben, auf der bereits der Z 1013-Monitor steht.

Auch der RAM-Bereich, in dem der EPROM-Handler selbst steht, kann bei Fehlangabe vom eingelesenen EPROM-Inhalt ueberschrieben werden! Nachdem die (Anfangs-) RAM-Adresse mit ENTER quittiert ist, beginnt der Lesevorgang. Hierfuer wird zunaechst der EPROM-Adresszaehler mit Hilfe des CLEAR-Impulses rueckgesetzt. Gegebenenfalls schliesst sich dann ein Hochzaehlen auf die gewaehlte Anfangsadresse an, wenn man den EPROM nicht vollstaendig auslesen will.

Als naechstes folgt das Einschalten der Betriebsspannungen UCC und UDD (ob beide verwendet werden, bestimmt der Typstecker). Mit Hilfe eines Zeit-Unterprogrammes ab Adr. 0595H bis 05A0H wird eine Wartezeit von ca. 70 ms eingeschoben, in der alle Einschwingvorgaenge, welche mit der Zuschaltung zusammenhaengen, abklingen koennen. Unmittelbar darauf folgt das Auslesen der ersten EPROM-Zelle. Mit Hilfe eines COUNT-UP-Impulses wird der EPROM-Adresszaehler inkrementiert, sodass ein neuer Lesevorgang eingeleitet werden kann. Ein Zellen-Lesezyklus beansprucht insgesamt 190 us. Fuer einen 1 K-EPROM entsteht so eine effektive Lesezeit von 0,2 s. Die anschliessende Checksummenbildung mit etwa gleicher Dauer laesst fuer die Funktion „Lesen“ eines 1 K-EPROMs etwa 0,5 s zusammenkommen. Hierbei ist zu beachten, dass die Checksumme des RAM-Inhaltes gebildet wird. Hat man irrtuemlicherweise als Ziel einen Adressbereich mit Festwertspeicherbelegung angegeben, erhaelt man eine falsche Angabe. Gleiches geschieht natuerlich auch, wenn die Leseoperationen durch Hardwarefehler gestoert sind. Nach Abschluss der Pruefsummenbildung wird das Ergebnis dargestellt und mit OK die abgearbeitete Funktion „Lesen“ fertiggemeldet. Erst mit einer beliebigen Tastenbedienung gelangt man wieder ins Funktionsmenue.

## Programmieren

Die Funktion „Programmieren“ ruft man mit der Kennziffer 2 auf. In der sich unmittelbar anschliessenden Frage, ob der EPROM vollstaendig oder teilweise zu programmieren ist, muss man sich mit N (NEIN) fuer teilweise entscheiden. Wie beim „Lesen“ folgt dann die Frage nach Anfangs- und Endadresse des Teilbereiches, der nur eine einzige EPROM-Zelle lang sein kann. Ist alles fehlerfrei, mit CTRL C gelangt man wieder ins Funktionsmenue, wird durch ENTER quittiert. Ehe das Programm aber nach der Quelladresse des zukuenftigen EPROM-Inhaltes fragt, wird der genannte EPROM-Adressbereich nach geloeschten Zellen abgefragt, die den Inhalt FFH besitzen. Erst danach erscheint der Text „von RAM-Adresse (HEX):“, wobei das Programm alle Angaben akzeptiert, die

eine Adresse im Adressraum des U880 bezeichnen, gleichgültig, ob mit RAM, ROM oder ohne Speicher besetzt. Ist die Quelladresse mit ENTER bestätigt, wird der Bildschirm gelöscht.

Ehe programmiert werden kann, folgt ein Vertraeglichkeitstest. Waehrend der Loeschtest nach dem EPROM-Inhalt FFH fragt, sucht der Vertraeglichkeitstest alle diejenigen Bits der EPROM-Zelle, die eine 0 haben, obwohl sie nicht programmiert werden sollen, z. B.:

programmierbar	unvertraeglich
----------------	----------------

EPROM FF: 1111 1111 F7: 1111 0 111 RAM 3E: 0011 1110 3E: 0011 1 110

Fehlerhafte EPROMs, Kontaktprobleme, mangelhafte Loeschung und Bedienfehler zeigt dieser Vertraeglichkeitstest mit grosser Sicherheit an. Zusaetzlich wird nicht nur die Unvertraeglichkeit konstatiert, sondern auch alle unvertraeglichen EPROM-Zellen im Tabellenvergleich zu den RAM-Zellen angezeigt. Mit CTRL C kann man anschliessend wieder ins Funktionswahlnuene zurueckkehren. Erst nach erfolgreichem Vertraeglichkeitstest schliesst sich der erste Programmierzyklus an. Er ist im Bild 6 als Zeit-Zustandsdiagramm dargestellt, wobei sich alle Zeitangaben auf den MONITOR Z1013/2.02 beziehen. Zu Beginn sind alle Randbedingungen, wie anliegende erste EPROM-Adresse, anliegende Daten fuer die erste EPROM-Zelle, passiver Programmierimpuls, Umschaltung in den Schreibmodus (LED VB 2 leuchtet, Programmierimpuls IMP noch passiv), erfuellt. Erst wenn all dies geschehen ist (bis zum Zeitpunkt 1), kann der EPROM-Handler den Programmierimpuls durch eine L/H-Flanke des Signales IMP anstossen. Bis zum Zeitpunkt 2 schliesst sich eine Wartezeit (Pauschalzeit) von 256 us an. Unabhaengig vom EPROM-Handler verlaeuft die restliche Zeit des Programmierimpulses bis zum Zeitpunkt 3. Laenger als dieser Impuls dauert aber die Tastaturabfrage, die vor Ende des Programmierimpulses begonnen wird und ca. 10 ms in Anspruch nimmt (Zeitpunkt 4). Vom Zeitpunkt 4 an beginnt bereits der folgende Zyklus, was durch den COUNT-UP-Impuls und die sich einstellende naechste EPROM-Adresse zum Zeitpunkt 5 erkennbar ist. Die Zeitspanne zwischen 5 und 6 benoetigt der EPROM-Handler, um den Inhalt der naechsten Zelle aus dem RAM zu holen, die internen Adressregister zu erhoehen, auf „Ende“ zu testen und zu fragen, ob die Zelle ueberhaupt programmiert werden muss. Ist eine Programmierung nicht noetig (RAM: FFH, EPROM: FFH), dann faellt der eigentliche Programmiervorgang aus und der Uebergang zur naechsten Zelle dauert nur 210 us. Soll keine Zelle uebersprungen werden, dann sind ab Zeitpunkt 7 wieder alle Vorbedingungen erfuellt und der naechste Zellen-Programmierzklus kann mit der L/H-Flanke von IMP beginnen. 2)

Fuer die Programmierung einer EPROM-Zelle stehen, wie im Bild 6 zu erkennen ist, ca. 13 ms zur Verfuegung. Fuer einen 1 K-EPROM entsteht eine reine Programmierzeit von 13,4 s, an die sich noch der Erfolgstest anschliesst. Dieser vergleicht den nunmehr vorliegenden EPROM-Inhalt mit dem Quellbereich im RAM. Stellt der Erfolgstest fuer eine EPROM-Zelle Ungleichheit fest, dann unterscheidet er zusaetzlich, ob Vertraeglichkeit vorliegt, also weiterprogrammiert werden kann, oder wegen Unvertraeglichkeit unterbrochen werden muss. Unvertraeglichkeit kann trotz vorheriger ausreichender Loeschung dann auftreten, wenn die Programmierspannung UPRG zu gross gewaehlt wurde und dadurch benachbarte Bitpositionen auf dem EPROM-Chip unbeabsichtigt auf „0“ gebracht werden. Die erste unvertraegliche EPROM-Zelle wird mit Angabe von RAM- und EPROM-Adresse angezeigt, so dass man danach mit CTRL C zum Funktionswahlnuene zurueckkehren oder mit einer beliebigen anderen Taste fortsetzen kann.

Verlaeuft der Erfolgstest so, dass Ungleichheit verbunden mit Vertraeglichkeit festgestellt wird, dann bestaetigt der EPROM-Handler einen vollstaendig abgearbeiteten Programmierzyklus mit einem ungefuellten Kreis als Quittungssymbol. Danach folgen bis maximal 100 Programmierzyklen, ehe gaenzlich abgebrochen wird. Wenn vorher bei einem Erfolgstest Uebereinstimmung aller Zellen

festgestellt wird, schliesst der EPROM-Handler Sicherheits(programmier)zyklen an. Insgesamt die dreifache Anzahl (jedoch mindestens zw||f) der benoetigten Programmierzyklen folgen, bis ein abschlie~ender Erfolgstest und eine Checksummenberchnung das „Programmieren“ abschlie~t. Als zugehoerige Fertigmeldung schlie~t sich ein OK an.

## Vergleichen

Die Funktion „Vergleichen“ wird mit der Kennziffer 3 innerhalb des Funktionswahlmenues aufgerufen. Sie entspricht in ihren Grundlagen dem „Lesen“, ohne dass der EPROM-Inhalt in den Spei- cher des Z 1013 uebergeben wird.

So wie bei allen vorher beschriebenen Funktionen kann man den gesamten EPROM-Inhalt oder Teile davon auswaehlen. Mit N (NEIN) entscheidet man sich fuer Teilbereiche des EPROM. Anschliessend werden die Bereichsadressen, als „rel. EPROM-Adresse“ bezeich- net, abgefordert (rel.= relative). Nach abgeschlossener Eingabe derselben fragt der EPROM-Handler nach der (Anfangs-)RAM-Adres- se, von der an verglichen werden soll. Faellt der Vergleich po- sitiv aus, dann meldet sich der EPROM- Handler mit einem OK, sodass man mit einem beliebigen Tastendruck zum Funktionswahl- menue zurueckkehren kann.

Treten jedoch Abweichungen auf, dann wird die erste festge- stellte Zelle mit RAM-Adressangabe, dazugehoerige EPROM-Adresse und deren nicht uebereinstimmenden Inhalten in einer Bild- schirmzeile angezeigt und der Vergleichsvorgang danach unter- brochen. Mit jeder beliebigen Taste faehrt das Programm mit der Arbeit fort, bis die naechste Abweichung auftritt. Nachdem alle Zellen ueberprueft sind, meldet sich der EPROM-Handler mit OK zurueck. Durch Druck einer beliebigen Taste erreicht man, wie oben gesagt, das Funktionswahlmenue. Vorher laesst sich mit CTRL C (S4-K) ein Zwangsabbruch der Funktion „Vergleichen“ herbei- fuehren.

## Loeschtest

Die Funktion „Loeschtest“ wird ueber die Kennziffer 5 erreicht. Ebenso wie beim „Lesen“, „Programmieren“ und „Vergleichen“ sind Bereichsangaben moeglich. Bei einem erfolgreichen Test, der Loeschtest ist ein abgewandeltes „Vergleichen“, wobei alle EPROM -Zellen den Inhalt FFH haben sollen, folgt das uebliche OK. Andernfalls wird die erste ungeloeschte Zelle mit EPROM-Adresse und dem Hinweis „no blank“ angezeigt. Nach jedem Tastendruck wird der Loeschtest fortgefuehrt, bis die naechste ungeloeschte Zelle ermittelt ist oder bis alle Zellen ueberprueft worden sind. Nach OK kehrt man mit beliebigem Tastendruck zum Funk- tionswahlmenue zurueck.

## Wiederholen

Alle oben beschriebenen Funktionen lassen sich unter Aufruf der Kennziffer 5 wiederholen. Hierbei bleiben alle Parameter, also:

gewaehlte Kennziffer,  
Typ (EPROM-Groesse),  
EPROM-Anfangsadresse,  
EPROM-Endadresse und

## RAM-Adresse

unveraendert erhalten. Das ist besonders vorteilhaft, wenn man groessere Serien von EPROMs bearbeitet, oder ein EPROM-Exemplar }berprogrammieren will. Die Fehler durch falsche Eingabe aller Art reduzieren sich merklich.

### Fortsetzen

Besonders bei den Funktionen „Lesen“ und „Programmieren“ ist es guenstig, beim Bearbeiten von EPROM-Saetzen den Parameter

## RAM-Adresse

fortlaufend vor jedem naechsten EPROM des Satzes automatisch weiterzustellen. Unter Kennziffer 6 ist dies moeglich und kann jederzeit mit CTRL C abgebrochen werden.

## Hardware des EPROM-Programmiergeraetes

### Allgemeiner Aufbau

Das EPROM-Programmiergeraet findet auf einer Zweiebenenleiter- karte von Format 135 mm x 72,5 mm Platz. Der Leiterbildentwurf stammt von V. LUEHNE und weist bereits Aenderungen zum Vorbild aus /1/ auf. Insgesamt fuenf Baugruppen

1. Fassung und Typsteckverbinder
2. Adresszaehler
3. Read/Write-Logik
4. Programmierspannungserzeugung und
5. Relaisblock

finden auf der Leiterkarte Platz.

Im Bild 1 ist der Stromlaufplan des EPROM-Programmiergeraetes ausgewiesen. Er stellt ausser Netzteil und Verbindungselemente zum Z 1013 alle oben genannten Baugruppen dar. Die Bestueck\_ kungszeichnung ist unter Bild 2 angegeben.

So wie in beiden Abbildungen dargelegt, ist das EPROM-Program- miergeraet bereits funktionsfaehig. Ueber einen 26poligen EFS- Steckverbinder (X1) kann es an eine PIO gemaess der Tabellen 3 und 4 angeschlossen werden, wobei es nicht zwingend ist, einen Z 1013 als Master zu benutzen. Lediglich alle betriebssystembe- dingten Besonderheiten des unter Punkt 1 beschriebenen EPROM- Handlers sind zu beruecksichtigen.

### Fassung und Typsteckverbinder

Den EPROM nimmt eine mindestens 28polige Schwenkhebelfassung auf. Sie ist so angeordnet, dass

der EPROM einen sicheren Halt hat und vom Bediener leicht zugaenglich bleibt. Besonders empfehlenswert ist es, die Fassung nicht direkt auf die Leiterplatte zu loeten, sondern sie etwas erhoehert anzubringen, damit sie von keinem anderen Bauelement auf der Leiterplatte ueberragt werden kann. Wenn man auf eine gesonderte Befestigung der Fassung verzichten moechte, ist es unbedingt notwendig, die verlaengerten Zuleitungen auch auf der Leiterseite anzuloeten. Auf die Fassung werden acht festverdrahtete Adressleitungen und acht festverdrahtete Datenleitungen gefuehrt. Das sind diejenigen Signale, deren Anschlusslage bei allen EPROMs von 2708 bis 27512 gleich ist (vgl. Tab. 1 und 2). Alle anderen Leitungen, ausser dem Masseanschluss, haben von Typ zu Typ andere Signalbelegung oder andere elektrische Kennwerte. Um diese Hauptprobleme der Mehrtypen-EPROM-Programmiergeraete zu loesen, benutzen die Autoren aus /1/ einen Typstecker als variablen Leitungsverteiler fuer Adress-Signale, Betriebsspannungen, Programmierimpulse, Programmierspannung und Schaltkreissteuerung.

Bild 3 zeigt die Belegung der Signale, die bei geeigneter Verbindung fuer die jeweilige EPROM-Groesse die geforderten Anschlussbedingungen ergeben. In den Bildern 4.1 bis 4.6 sind die Typstecker aufgefuehrt. Ihre Schaltungen sind teilweise der Literatur entnommen. Die Typstecker der Bilder 4.1...4.2. stammen aus /1/, wobei jedoch zu bemerken ist, dass der Typstecker des 4 K-EPROMs (2732) auf S. 84 fehlerhaft ist. Der im Bild 4.2. wurde korrigiert. Fuer die Typen 2764 und 27128 stammt die Schaltung aus /7/ und musste ebenfalls berichtigt werden. Die uebrigen zwei Typstecker wurden vom Autor entworfen, wobei jedoch nur der fuer 32 K-EPROMs getestet werden konnte. Zum Typstecker nach Bild 4.6. sei vermerkt, dass er vom EPROM-HANDLER V 2.9 nur indirekt gestuetzt wird (siehe 2.3). Als Typsteckverbindungstyp wurde ein 26poliger direkter Steckverbinder (TGL 29331/01) gewaehlt, wie ihn auch die Autoren in /1/ vorschlagen. Dies gestattet es, sich die Typstecker relativ leicht selbst herzustellen. Solche Abweichungen koennen Potentiometer sein, die anstelle des Festwiderstandes stehen und so die Programmierspannung kontinuierlich zu veraendern gestatten. Auch zusaetzliche Leuchtdioden, die die anliegende Betriebsspannung signalisieren, sind denkbar, wenn man VB 4 in Bild 1 nicht auf der Leiterplatte unterbringen will. Selbstverstaendlich kann auf der Typsteckerleiterplatte eine Hilfsspannungsquelle (Batterie, Transverter) untergebracht werden.

## Adresszaehler

Die EPROM-Adress-Signale von einem Zaehler zu erzeugen, ist ein erheblicher Vorteil des EPROM-Programmierens nach /1/. Gegenueber anderen Varianten aus /2/, /3/, /4/ reduzieren sich die Anzahl Steuerleitungen von mindestens 16 auf nur zwei, naemlich COUNT-UP und /CLEAR. Ehe die Signale auf die vier Zaehlerschaltkreise D1, D2, D3 und D4 gelangen, werden sie von Treibern, D6.2 und D6.4 fuer den COUNT-UP- und D.6.3 fuer den /CLEAR-Impuls, elektrisch gepuffert.

In /1/ werden fuer die Adresszaehlerschaltkreise die Typen D193 genannt. Sie besitzen den Nachteil, insgesamt ca. 250 mA Betriebsstrom aufzunehmen. Aus diesem Grunde wurde durch V. LUEHNE dem Typ DL093 (74LS93) der Vorzug gegeben. Er belastet das Netzteil lediglich mit 4 und 7 mA. Auf den Rueckwaertszaehleingang wurde verzichtet, da er nicht unmittelbar notwendig ist.

Insgesamt hat der Adresszaehler einen Zaehlumfang von 15 bit, so dass man ohne zusaetzliche Massnahmen nur EPROMs bis zu einem Fassungsvermoegen von 32 K-Byte bearbeiten kann. Mit Hilfe des Typsteckers nach Bild 5.1. laesst sich die Einschraenkung ueberwinden, da der Bankschalter ein Signal am A15-Pin des 27512 nachbildet.

## Read/Write-Logik

Unter Read/Write-Logik sind alle Bauelemente zusammengefasst, die unmittelbar zu VT4 gehoeren. Gemeinsam sorgen sie dafuer, dass mit Hilfe des Typsteckers eine uebliche Transistorschaltstufe (VT4, R12 (R13)) hergestellt werden kann. Wenn VT4 durchgesteuert ist, entsteht an seinem Kollektor Low-Potential. Ueber den Typstecker gelangt das Signal an die Freigabe-Eingänge CE (CS) oder OE, so dass der jeweils gesteckte EPROM gelesen werden kann.

Die VT4 ansteuernden zwei Negatoren D6.1 und D6.5 arbeiten als Leuchtdiodentreiber und sichern dem Anschluss /WRITE (A4:X1) einen TTL-gerechten Eingang. Der Widerstand R18 sorgt dafuer, dass das Highpotential den Vorrang hat, auch wenn der steuernde PIO-Ausgang, nach dem Einschalten des Z 1013, ein undefiniertes Signal fuehrt. Durch diese als Sicherheitsvorkehrung zu verstehende Massnahme sichert den Vorrang des Modus' „EPROM-Lesen“. Die LED VB2 ist in diesem Zustand dunkel.

/WRITE: low	/WRITE: high
-------------	--------------

Programmieren Lesen

VB2 hell	VB2 dunkel
----------	------------

Im Programmiermodus, wenn die Read/Write-Logik mit einem statischen Low-Signal angesteuert wird, geht die Emitter-Kollektor-Strecke von VT4 in den hochohmigen Zustand ueber. Je nach Herstellervorschrift laesst sich an den OE-Pins der EPROMs nun ueber den Typstecker ein gewuenshtes Potential herstellen. Sicherheitshalber sollte abweichend von SS 218 im Bild 1 ein Typ gewaehlt werden, der eine maximale Kollektor-Emitter-Spannung von groesser als 30 V (UPRGmax. = 30 V) besitzt.

## Programmierspannungserzeugung

Die Programmierspannungserzeugung verlaeuft in mehreren Stufen und beginnt mit der Rohspannungserzeugung ausserhalb der Programmierleiterplatte. Sie kann bis zu 0,5 V verbrummt sein und sollte mit mehr als 60 mA belastbar sein, ohne um mehr als 1 V zurueckzugehen. Diese Werte haben sich bei Untersuchungen ergeben, die davon ausgehen, dass EPROMs mit einer Programmierspannung von groesser als 26 V sehr selten sind. Im unguenstigsten Falle sollte der Spannungsstabilisator-Schaltkreis N1 noch eine Regelreserve von 3 V haben.

N1 besorgt die zweite Stufe der Programmierspannungserzeugung, indem er eine typgerechte Gleichspannung bereitstellt. Ueber den Spannungsteiler R1, R2 wird sie wunschgemaess justiert. Die Widerstandswerte von R1 und R2 sind so gewaehlt, dass ohne Typstecker und ohne bestaetigten Paketschalter 12 V am Ausgang des MAA 723 (N1) erscheinen. Als besonderer Grund ist hierfuer der Umstand zu nennen, dass die niedrigste bekannte Programmierspannung bei 12,5 V liegt. Alle anderen geforderten Werte bewegen sich oberhalb dieser Grenze und koennen erzielt werden, indem man den Widerstand R2 vermindert. Diese Verminderung ist nach /1/ am leichtesten durch einen Paralellwiderstand zu erreichen, der zwischen den Anschluessen B1 und B6 des Typsteckers angebracht wird. Hieran ist nachteilig, dass nicht nur fuer n EPROM-Groessen, sondern auch fuer m verschiedene Programmierspannungen Typstecker herzustellen sind. Im Extremfall muessten n x m Typstecker angefertigt werden. Der Ausweg bietet sich hier durch einen zusaetzlichen Schalter an, durch den sich die Programmierspannung unabhængig vom Typstecker einstellen laesst. Nicht nur

der Umstand der eingesparten zusaetzlichen Typstecker, sondern auch die Moeglichkeit bei unbekannter Programmierspannung sich an einen akzeptablen Wert heranzutasten, lassen diesen Paketschalter als attraktiv erscheinen. Neben den sieben Schaltstufen fuer Festspannungen von:

12,5 V  
13,0 V  
16,0 V  
18,5 V  
21,0 V  
24,0 V  
26,0 V,

hat der Schalter eine achte Stellung, die keinen Widerstand bedient. Sie ist dafuer vorgesehen, um vorhandene Typstecker mit Widerstand einsetzen zu koennen.

Vom Ausgang des Programmierspannungserzeugers N1 wird auf zwei Stellen verzweigt. Die erste ist der Typsteckverbinderkontakt B2:X2. Von hier aus koennen die EPROM-Typen

2716  
2764  
27128 und  
27256

fuer die Programmierung versorgt werden, denn sie benoetigen eine Gleichspannung /8/, /9/ der vom Hersteller geforderten Hoehe.

Als zweite Stelle ist ein Chopper zu nennen, der als Gegentakt- Komplementaer-Stufe aufgebaut ist und aus den Transistoren VT 1, 2, 3 nebst den Widerstaenden R4...R10 besteht. Alle EPROM- Typen, die eine pulsierende Programmierspannung benoetigen (2708, 2732 und 27512), werden ueber den Anschluss A1:X2 von diesem Chopper versorgt. Er schaltet fast restspannungslos mit steiler Flanke zwischen 0 V und der durch R2 eingestellten Programmierspannung UPRG hin und her. Eventuelles Ueberschwingen soll das RC-Glied R4, C2 unterdruecken, da es empfindliche EPROM-Exemplare zerstoeren kann.

Um den Chopper zu bedienen, wird an den Basen von VT1 und VT2 als Steuerspannung eine gegenphasige Impulsfolge benoetigt. Ein monostabiler Trigger (D5) mit komplementaeren Ausgangssignalen ist hierfuer besonders geeignet. Er sichert naemlich, dass sein stabiler Zustand fuer die ausgeschaltete Programmierspannung steht und im Falle von Programmabstuerzen UPRG auf 0 V zurueck- kehrt.

Die Dauer eines Programmierspannungsimpulses wird von der R/C- Kombination C3, R14 bestimmt und liegt nach Empfehlung aus /1/ bei 1 ms. Sollte naemlich ein EPROM z. B. nach bereits zwei Programmierzyklen zu je 1 ms schon stabile Inhalte haben, dann kann auf den geforderten Rest von 48 ms Programmierzeit zumindest teilweise verzichtet werden. Eine laengere Lebensdauer dieses EPROM-Exemplares ist dann eine angenehme Folge jenes Ver- zichtes.

Vom Rechner wird der Programmierspannungs-Impulserzeuger ueber Pin 5 des Schaltkreises D5 mit einer Low-High-Flanke ange- stossen (vgl. Bild 6). Nach der Ausloeseflanke durchlaeuft das Programmierprogramm EPROM-HANDLER V 2.9 eine Warteschleife. Nach dieser Warteschleife (Pauschalzeit) erfolgt zunaechst eine Tastaturabfrage nach dem Abbruchzeichen CTRL C. Bei einer Rechner- taktfrequenz von 2 MHz dauert eine solche Tastaturabfrage ca. 10 ms, so dass das

nachfolgende Lesen des /READY-Signales einen beendeten Programmierimpuls anzeigt. Der Inhalt der Speicherzelle 061FH ist fuer die Pauschalzeit verantwortlich und liefert beim Wert von 20H 256 us.

EPROM-Typen, die keine pulsierende Programmierspannung benoetigen, haben einen TTL-gerechten Programmierimpuls. Er gibt die Programmierspannung erst im Inneren des EPROMs zur Wirkung frei. Von den gegenphasigen Ausgaengen des 74121 werden diese Programmierimpulse auf den Typsteckverbinder X2 (Bild 3) gefuehrt, um je nach Typstecker zur Anwendung zu kommen.

Die Leuchtdiode VB1 soll nach Absicht der Verfasser aus /1/ den aktivierten Impulsgeber D5 hervorheben. Jedoch ist dies nicht guenstig gewaehlt, denn die Impulsfolge an Pin 6 von D5 hat ein Tastverhaeltnis von 0,08. Die maessig glimmende LED steht daher im Schatten von VB2, zumal beide ohnehin gleichzeitig leuchten, weil der Zustand „aktive Programmierimpulserzeugung“ an den Zustand „/WRITE“, gebunden ist. Man koennte auf VB1 verzichten, wengleich die Leuchtdiode eine willkommene Testhilfe bei der Inbetriebnahme ist.

## Relaisblock

Vom Programmierprogramm EPROM-HANDLER V 2.9 werden zwei Relais (K1 und K2) bedient. K1 besitzt einen Arbeitskontakt und schaltet damit die Programmier(gleich)spannung auf den Typsteckverbinder X2. Um einen geschlossenen Kontakt anzeigen zu koennen, wird der Spulenstrom ueber die Leuchtdiode VB3 gefuehrt. Der Widerstand R24 liefert einen Nebenschluss, um einen Teil des Spulenstromes umzuleiten. Seine Groesse richtet sich nach der verwendeten Leuchtdiode und nach der gewuenschten Helligkeit.

Das Relais K2 verfuegt ueber zwei Arbeitskontakte und hat hiermit die Aufgabe, die EPROM-Betriebsspannungen UCC (+5 V) mit K1.1 und UDD (+12 V) mit K1.2 einzuschalten. Leider haben die Autoren aus /1/ hier keinen Indikator vorgesehen. Es waere durchaus ratsam, die eingeschalteten EPROM-Betriebsspannungen anzuzeigen, denn das kann ein Achtungszeichen sein, bei spannungsfuehrender Fassung, z. B. bei der Funktion 3 (Vergleichen), den EPROM nicht abzunehmen. VB4 ist nicht auf der Bestueckungszeichnung enthalten.

## Anschluss an den Z 1013

Der Anschluss des Programmiergeraetes an den Z 1013 geschieht mittels einer zusaetzlichen PIO, da das USER-Port am Steckverbinder X4 nicht ausreicht. Von den 16 Portanschlussen der PIO (A0...A7; B0...B7) werden 15 verwendet. Die ihnen zugeordneten logischen Signale mit ihren Spezifikationen sind in Tab. 3 aufgefuehrt.

Mit Hilfe eines mehradrigen Kabels wird das EPROM-Programmiergeraet an die zusaetzliche PIO angeschlossen, wobei man aus Bild 8 erkennen kann, dass die Anschlussleitung kuerzer als 60 cm sein muss, da sich ansonsten Adressversatzfehler ergeben. Die Laengenangabe gliedert sich in einen fixen Anteil von 30 cm, der sich aus den Leitungen auf dem Programmiergeraet und der PIO-Baugruppe zusammensetzt, und aus einem variablen Teil, der die Laenge des Kabels angibt. Um ohne Zusatzmassnahmen fehlerfrei zu arbeiten, muss das Anschlusskabel also kuerzer als 30 cm sein!

Ist man gezwungen, laengere Anschlusskabel einzusetzen, ergeben sich Schwierigkeiten. Diese sind durch die Anpassungsverhaeltnisse an der Anschlussleitung bedingt, denn sie ist bei 200 Ohm Wellenwiderstand durch niederohmige Sender (50 Ohm) und durch hochohmige Empfaenger (6 kOhm) beschaltet. Nach einem graphischen Loesungsverfahren, das in /11/ beschrieben ist, wurde

der Signalverlauf am Leitungsende ermittelt (Bild 7). Bemerkenswert ist hierbei das positive Ueberschwingen. Es kann nach einer H/L-Flanke unerwünschte zusätzliche Schaltvorgänge auslösen, was namentlich am COUNT-UP-Eingang zu Fehlern führt, da anstelle eines Zählimpulses zwei wirken. Mitten im „Lesen“ oder „Programmieren“ scheint dann eine EPROM-Adresse ausgelassen zu sein!

Ob der Zusatzimpuls zur Wirkung kommt, hängt vorrangig von seiner Dauer ab, und diese wird von der Leitungslänge bestimmt. Bei 60 cm Leitungslänge entsteht eine Impulsbreite von 6 ns, bei 90 cm 9 ns und bei 1,4 m 14 ns.

Betrachtet man nun noch den minimalen Schaltabstand eines LS-TTL-Gatters von 9 ns, dann wird klar, dass dadurch Schaltvorgänge ausgelöst werden können. Aus Bild 8 ergibt sich, dass nach durchschnittlich 1100 COUNT-UP-Impulsen ein störender Zusatzimpuls wirkt.

Als Gegenmassnahme, die nur für die COUNT-UP-Leitung nötig ist, bietet sich als Minimalvariante ein „Entprellkondensator“ gemäss Bild 10 an. Nach Empfehlung aus [12] darf er eine Kapazität von bis zu 1 nF haben. Mit ihm lassen sich Ueberschwingimpulse bis zu einer Breite von 20 ns sicher unterdrücken. Die sich daraus ergebende Anschlusskabellänge von 1,7 m dürfte für die meisten Anwendungsfälle ausreichen.

Weniger auffällig, jedoch gefährlicher, ist das negative Ueberschwingen. Für den Fall, dass man einen EPROM programmiert, liegt an seinen Dateneingängen kurzzeitig negatives Potential an, was kleiner als der zulässige Grenzwert von -0,3 V ist. Um den EPROM vor negativen Eingangsspannungen zu schützen, empfiehlt es sich, in die Daten-Anschlussleitungen einen Leitungstreiber-IC einzufügen (siehe Bild 9). Das hat neben dem EPROM-Schutz den günstigen Nebeneffekt, dass bei der Funktion „Lesen“ die Datenausgänge des EPROMs nicht das Anschlusskabel treiben müssen.

Selbstverständlich entsteht das negative Ueberschwingen auch an den PIO-Eingängen beim „Lesen“. Hier bietet sich an, die PIO-Pins des Ports A mit Germanium-Dioden gegen Masse zu sichern.

## Netzteil

Zum Netzteil lassen sich lediglich Empfehlungen geben, denn jeder Amateur ist hierbei durch seine speziellen Gegebenheiten gebunden. Als Entscheidungshilfe sollen hierzu die Gegenüberstellung von externer und interner Stromversorgung dienen.

extern	intern
--------	--------

Stromaufnahme problemlos, übliche problematisch,

Schaltungen	drei Spannungen im Rechner nötig
-------------	-------------------------------------

Programmierspannung problemlos, übliche problematisch,

Schaltungen	Verwendung des Koppelbusses bei K 1520-kompatib-
-------------	--

	len Baugruppen, Abweichungen von der Systembusbe- legung (auf alle Faele vermeiden), Aufbau eines Transver- ters auf dem PIO- Modul, separate Zufuehrung
--	---

Schaltungsaufwand mittlerer Aufwand mittlerer Aufwand

mechanischer Aufwand sehr hoch gering

Anschluss lange Leitungen kurze Leitungen

Handhabbarkeit gut, an anderen Orten eingeschraenkt,

Rechnertypen einsetz- bar	da an Ort und Aufbau gebunden
------------------------------	----------------------------------

In Bild 4 ist der Vorschlag eines externen Netzteil abgebil- det. Die Stromergiebigkeit der 17 V- Gleichspannungsquelle soll- te mindestens 500 mA betragen. Nachstehend aufgefuehrte Tabel- le gibt ueber die Stromaufnahme des EPROM-Programmiers Aus- kunft:

passiv	aktiv
--------	-------

+ 5 V 65 mA	+ ICC-EPROM 140 mA	+12 V 0 mA 40 mA
kleiner 2 mA	kleiner 50 mA	+ IDD-EPROM - 5 V 0 mA
		IBB-EPROM +30 V

### Anmerkungen

1) Fuer 1 KByte-EPROMs kann auch die Kennziffer 55

angegeben werden, da der der internationale Typ 2708 in der DDR die Bezeichnung U 555 C tr{gt.

2) Das Verlassen des EPROM-Handlers mit RESET ist

w{hrend des aktiven Progammierens zu vermeiden, da ansonsten die EPROM-Zelle "0" einen falschen Inhalt annehmen kann.

### Literaturverzeichnis

/ 1/ Huebler, H.; Evert, K.-P.: Amateurreihe „Elektronika“ Nr. 227 u. 228/ Huebler, H.;

Evert, K.-P. - Berlin, 1985  
S. 76

/ 2/ LC-80 Systemunterlagen

/ 3/ Buerger, B.: Reihe „Information Applika-

tion" Nr. 41 Halbleiterspei-  
cher Teil 3 / Buerger, B. -  
KdT Frankfurt/0., 1987 S. 42

/ 4/ Buerger, B.: Reihe „Information Applika-

tion" Nr. 41 Halbleiterspei-  
cher Teil 3 / Buerger, B. -  
KdT Frankfurt/0., 1987 S. 40

/ 5/ Mueller, M.; Krueger, S.: EPROM-Programmiergeraet /

Mueller, M.; Krueger, S.  
Radio Fernsehen Elektronik. -  
Berlin 31 (1982) 10. S. 621

/ 6/ Kuehnel, Th.: Einfacher EPROM-Programmier-

modul / Kuehnel, T. Radio  
Fernsehen Elektronik. -  
Berlin 35 (1986) 10. S. 633

/ 7/ Brosig, R.: CODIERSTECKER 4. Textfile

/ 8/ Mueller, M.: EPROM-Programmiergeraet /

Mueller, M. Radio Fernsehen  
Elektronik. - Berlin 37 (1988)  
7. S. 429

/ 9/ Buerger, B.: Reihe „Information Applikation“

Nr. 41 Halbleiterspeicher  
Teil 3 / Buerger, B. - KdT  
Frankfurt/0., 1987

/10/ Brosig, R.: Z-1013-Tastatur mit Raffines-

sen. Mikroprozessortechnik  
7/88, S. 215

/11/ Turinsky, G.: Leitungsabschlusschaltungen /

Turinsky, G. Radio Fernsehen  
Elektronik. - Berlin 32 (1983)  
8. S. 182

/12/ Kuehn, E.; Schmied, H.: Handbuch Integrierter Schalt-

kreise / Kuehn, E.; Schmied, H.  
- Berlin, 1979 S. 113

Autor: U. Hinz

Dresden, 06. 03. 1988

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

Permanent link:  
<https://hc-ddr.hucki.net/wiki/doku.php/z1013/software/eprommer?rev=1316098263>

Last update: **2011/09/15 14:51**

