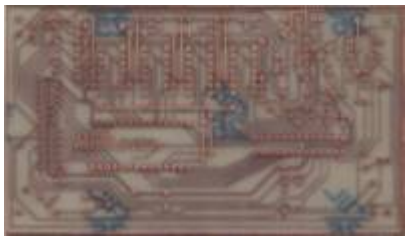
EPROM-Handler

Beschreibung zum EPROM-Programmierprogramm
EPROM-Handler V 2.9 (RAM) für Z 1013
by Rainer Brosig

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



EPROMMER 2.9/IGD, EPROMMER 2.9/CCL, Buebchen-Brenner

Hardware:

- Z 1013 / Z 9001
- EPROM-Programmiergerät der IG-HC TU Dresden, nach electronica Bd.227/228 weiterentwickelt. eprommer-227.pdf
- in Leipzig (2. Z1013-Tagung) von der IG-HC TU Dresden vorgestellt.
- zusaetzliche PIO im Z 1013; für Programmiergerät keine Interruptfähigkeit gefordert. (Zusatzbaugruppe Robotron Riesa ab I/O-Adr. 30H)
- Leiterplatte s. Eprommer der IG-HC Dresden

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-Prüfsumme
- Programmieren von beliebigem Speicherbereich, dabei automatischer Löschtest und bei negativen Ergebnis Verträglichkeitstest; während der Programmierung wird der laufende

Programmierzyklus bzw. Sicherheitszyklus durch verschiedene Grafikzeichen angezeigt ; es werden so viele Zyklen ausgeführt 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-Prüfsumme

- Vergleichen des EPROM-Inhalts mit gewähltem Speicherbereich, dabei Ausgabe ungleicher Zellen mit deren Inhalt
- Löschtest mit Ausgabe aller nichtgelöschten Zellen und deren Inhalt; diese Funktion kann auch zum schnellen Betrachten des EPROM-Inhalts genutzt werden
- Wiederholen der zuvor ausgeführten Funktion mit gleichen Parametern
- Fortsetzen der vorher ausgeführten Funktion mit gleichen Parametern außer der sich an den alten Bereich anschließenden RAM-Adressse

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

Durch BREAK kann auf der Menüebene zum Monitor zurückgekehrt werden sowie durch Eingabe von N ein anderer EPROM-Typ gewählt 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    A8
A5    A5    A5     |  5   24 | A9  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   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    D7
D0    D0    D0     | 11   18 | D6  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    D5
D2    D2    D2     | 13   18 | D4  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    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

Die Bilder befinden sich in den Unterlagen zum EPROMMER IGD eprommer_igd.pdf

Einleitung

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

Um sich für ein Programmiergerät entscheiden zu können, sind mehrere Auswahlkriterien zu beachten:

- Anzahl der zu programmierenden Typen (maximal)
- automatische Typerkennung
- Art und Umfang des Interfaces (minimal)
- Portabilität bezüglich der Rechnertypen (maximal)
- Softwareunterstützung (maximal)
- Preis-Leistungsverhältnis (optimal)
- Vorleistungen (maximal)

Bei den zu Programmiergeräten erschienenen Veröffentlichungen lassen sich zwei Kategorien unterscheiden. Die erste Kategorie umfasst Programmiergeräte, die auf einen Typ von EPROMs zugeschnitten sind /2/ /3/ /4/. Sie kommen für höhere Ansprüche 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 Programmiergerät 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 Programmiergerät

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

Der EPROM-HANDLER V 2.9 ist ein komfortables, menüorientiertes, für den Z 1013 entwickeltes Werkzeug. Er belegt den Adressraum von 100H bis 08FBH, wird mit J 100 gestartet und ist selbst nicht EPROM-fähig. Das im Anhang befindliche HEX-Listing gibt das Programm vollständig 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 für I/O-Adressen für 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.

Kernstück dieser Funktionen sind die Leseroutine (03FAH bis 0422H) und die Zellen-Programmierroutine (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 Überblick für das Signalspiel der Zellen-Programmierroutine.

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-Größe bezieht. Nicht die Typen einzelner Hersteller sind gemeint, denn diese unterscheiden sich meist nach dynamischen Kennwerten, der Programmierspannung oder der Betriebsstromaufnahme. Stellvertretend für die EPROM-Größe werden deshalb die international üblichen Endnummern:

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

zur Unterscheidung herangezogen.

In den Arbeitszellen 0128H und 0129H wird die Byteanzahl, die sich aus der EPROM-Größe ergibt, abgelegt. Diese Byteanzahl ist die einzige Unterscheidungsmöglichkeit für die EPROM-Größe innerhalb der internen Abläufe des EPROM-Handlers. Sollen andere EPROM-Größen 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 berücksichtigt.

Ehe man die eingegebene Endnummer durch ENTER gültig macht, sollte man sich in jedem Falle noch einmal überzeugen, ob der richtige Typstecker gewählt wurde. Hat man irrtümlicherweise eine falsche Größe bestätigt, dann lässt sich mit dem Kommando N (NEU) zur Typauswahl zurückkehren.

Lesen

Nachdem die Typauswahl erfolgt ist, bietet der EPROM-Handler Ein Funktionswahlmenü an. Hierin steht die Kennziffer 1 für „Lesen“. Nachdem man die Kennziffer eingegeben hat, stellt der EPROM-Handler die Frage, ob man den steckenden EPROM vollständig oder teilweise lesen möchte. Mit ENTER entscheidet man sich für vollständiges Lesen und mit N (NEIN) für bereichsweises.

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

1. Anfangsadresse Größer als Endadresse und
2. Endadresse Größer als EPROM-Größe,

besteht Schutz. Soll nur eine einzelne EPROM-ZELLE gelesen werden, dann sind die Angaben für Anfangs- und Endadresse gleich. Nachdem der einzulesende EPROM-Adressbereich eindeutig und fehlerfrei feststeht, bestätigt 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 könnte 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 überschrieben werden! Nachdem die (Anfangs-) RAM-Adresse mit ENTER quittiert ist, beginnt der Lesevorgang. Hierfür wird zunächst der EPROM-Adresszähler mit Hilfe des CLEAR-Impulses rückgesetzt. Gegebenenfalls schließt sich dann ein Hochzählen auf die gewählte Anfangsadresse an, wenn man den EPROM nicht vollständig auslesen will.

Als nächstes 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 Einschwingvorgänge, welche mit der

Zuschaltung zusammenhängen, abklingen können. Unmittelbar darauf folgt das Auslesen der ersten EPROM-Zelle. Mit Hilfe eines COUNT-UP-Impulses wird der EPROM-Adresszähler inkrementiert, sodass ein neuer Lesevorgang eingeleitet werden kann. Ein Zellen-Lesezyklus beansprucht insgesamt 190 us. Für einen 1 K-EPROM entsteht so ei- ne effektive Lesezeit von 0,2 s. Die anschließende Checksummenbildung mit etwa gleicher Dauer lässt für 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 irrtümlicherweise als Ziel einen Adressbereich mit Festwertspeicherbelegung angegeben, erhält man eine falsche Angabe. Gleiches geschieht natürlich auch, wenn die Leseoperationen durch Hardwarefehler gestört sind. Nach Abschluss der Prüfsummenbildung wird das Ergebnis dargestellt und mit OK die abgearbeitete Funktion „Lesen“ fertiggemeldet. Erst mit einer beliebigen Tastenbedienung gelangt man wieder ins Funktionsmenü.

Programmieren

Die Funktion „Programmieren“ ruft man mit der Kennziffer 2 auf. In der sich unmittelbar anschließenden Frage, ob der EPROM vollständig oder teilweise zu programmieren ist, muss man sich mit N (NEIN) für 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 Funktionsmenü, wird durch ENTER quittiert. Ehe das Programm aber nach der Quelladresse des zukünftigen EPROM-Inhaltes fragt, wird der genannte EPROM-Adressbereich nach gelöschten 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 Verträglichkeitstest. Während der Löschtest nach dem EPROM-Inhalt FFH fragt, sucht der Verträglichkeitstest 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 Löschung und Bedienfehler zeigt dieser Verträglichkeitstest mit großer Sicherheit an. Zusätzlich wird nicht nur die Unverträglichkeit konstatiert, sondern auch alle unverträglichen EPROM-Zellen im Tabellenvergleich zu den RAM-Zellen angezeigt. Mit CTRL C kann man anschließend wieder ins Funktionswahlmenü zurückkehren.

Erst nach erfolgreichem Verträglichkeitstest 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 für die erste EPROM-Zelle, passiver Programmierimpuls, Umschaltung in den Schreibmodus (LED VB 2 leuchtet, Programmierimpuls IMP noch passiv), erfüllt. 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. Unabhängig vom EPROM-Handler verläuft die restliche Zeit des Programmierimpulses bis zum Zeitpunkt 3. Länger 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

nächste EPROM-Adresse zum Zeitpunkt 5 erkennbar ist. Die Zeitspanne zwischen 5 und 6 benötigt der EPROM-Handler, um den Inhalt der nächsten Zelle aus dem RAM zu holen, die internen Adressregister zu erhöhen, auf „Ende“ zu testen und zu fragen, ob die Zelle überhaupt programmiert werden muss. Ist eine Programmierung nicht nötig (RAM: FFH, EPROM: FFH), dann fällt der eigentliche Programmiervorgang aus und der Übergang zur nächsten Zelle dauert nur 210 us. Soll keine Zelle übersprungen werden, dann sind ab Zeitpunkt 7 wieder alle Vorbedingungen erfüllt und der nächste Zellen-Progammierzyklus kann mit der L/H-Flanke von IMP beginnen. 2)

Für die Programmierung einer EPROM-Zelle stehen, wie im Bild 6 zu erkennen ist, ca. 13 ms zur Verfügung. Für 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 für eine EPROM-Zelle Ungleichheit fest, dann unterscheidet er zusätzlich, ob Verträglichkeit vorliegt, also weiterprogrammiert werden kann, oder wegen Unverträglichkeit unterbrochen werden muss. Unverträglichkeit kann trotz vorheriger ausreichender Loeschung dann auftreten, wenn die Programmierspannung UPRG zu groß gewählt wurde und dadurch benachbarte Bitpositionen auf dem EPROM-Chip unbeabsichtigt auf „0“ gebracht werden. Die erste unverträgliche EPROM-Zelle wird mit Angabe von RAM- und EPROM-Adresse angezeigt, so dass man danach mit CTRL C zum Funktionswahlmenü zurückkehren oder mit einer beliebigen anderen Taste fortsetzen kann.

Verläuft der Erfolgstest so, dass Ungleichheit verbunden mit Verträglichkeit festgestellt wird, dann bestätigt der EPROM-Handler einen vollständig abgearbeiteten Programmierzyklus mit einem ungefüllten Kreis als Quittungssymbol. Danach folgen bis maximal 100 Programmierzyklen, ehe gänzlich abgebrochen wird. Wenn vorher bei einem Erfolgstest Übereinstimmung aller Zellen festgestellt wird, schließt der EPROM-Handler Sicherheits(programmier)zyklen an. Insgesamt die dreifache Anzahl (jedoch mindestens zwölf) der benötigten Programmierzyklen folgen, bis ein abschließender Erfolgstest und eine Checksummenberechnung das „Programmieren“ abschließt. Als zugehörige Fertigmeldung schließt sich ein OK an.

Vergleichen

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

So wie bei allen vorher beschriebenen Funktionen kann man den gesamten EPROM-Inhalt oder Teile davon auswählen. Mit N (NEIN) entscheidet man sich für Teilbereiche des EPROM. Anschließend werden die Bereichsadressen, als „rel. EPROM-Adresse“ bezeichnet, abgefordert (rel.= relative). Nach abgeschlossener Eingabe derselben fragt der EPROM-Handler nach der (Anfangs-)RAM-Adresse, von der an verglichen werden soll. Fällt der Vergleich positiv aus, dann meldet sich der EPROM-Handler mit einem OK, sodass man mit einem beliebigen Tastendruck zum Funktionswahlmenü zurückkehren kann.

Treten jedoch Abweichungen auf, dann wird die erste festgestellte Zelle mit RAM-Adressangabe, dazugehörige EPROM-Adresse und deren nicht übereinstimmenden Inhalten in einer Bildschirmzeile angezeigt und der Vergleichsvorgang danach unterbrochen. Mit jeder beliebigen Taste fährt das Programm mit der Arbeit fort, bis die nächste Abweichung auftritt. Nachdem alle Zellen überprüft sind, meldet sich der EPROM-Handler mit OK zurück. Durch Druck einer beliebigen Taste erreicht man, wie oben gesagt, das Funktionswahlmenü. Vorher lässt sich mit CTRL C (S4-K) ein Zwangsabbruch der Funktion „Vergleichen“ herbeiführen.

Löschtest

Die Funktion „Löschtest“ wird über die Kennziffer 5 erreicht. Ebenso wie beim „Lesen“, „Programmieren“ und „Vergleichen“ sind Bereichsangaben möglich. Bei einem erfolgreichen Test, der Löschtest ist ein abgewandeltes „Vergleichen“, wobei alle EPROM -Zellen den Inhalt FFH haben sollen, folgt das übliche OK. Andernfalls wird die erste ungelöschte Zelle mit EPROM-Adresse und dem Hinweis „no blank“ angezeigt. Nach jedem Tastendruck wird der Löschtest fortgeführt, bis die nächste ungelöschte Zelle ermittelt ist oder bis alle Zellen überprüft worden sind. Nach OK kehrt man mit beliebigem Tastendruck zum Funktionswahlmenü zurück.

Wiederholen

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

```
gewaehlte Kennziffer,
Typ (EPROM-Größe),
EPROM-Anfangsadresse,
EPROM-Endadresse und
RAM-Adresse
```

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

Fortsetzen

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

```
RAM-Adresse
```

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

Hardware des EPROM-Programmiergerätes

Allgemeiner Aufbau

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

1. Fassung und Typsteckverbinder
2. Adresszähler

3. Read/Write-Logik
4. Programmierspannungserzeugung und
5. Relaisblock

finden auf der Leiterkarte Platz.

Im Bild 1 ist der Stromlaufplan des EPROM-Programmiergerätes ausgewiesen. Er stellt außer Netzteil und Verbindungselemente zum Z 1013 alle oben genannten Baugruppen dar. Die Bestückungszeichnung ist unter Bild 2 angegeben.

So wie in beiden Abbildungen dargelegt, ist das EPROM-Programmiergerät bereits funktionsfähig. Über einen 26poligen EFS-Steckverbinder (X1) kann es an eine PIO gemäß der Tabellen 3 und 4 angeschlossen werden, wobei es nicht zwingend ist, einen Z 1013 als Master zu benutzen. Lediglich alle betriebssystembedingten Besonderheiten des unter Punkt 1 beschriebenen EPROM-Handlers sind zu berücksichtigen.

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 zugänglich bleibt. Besonders empfehlenswert ist es, die Fassung nicht direkt auf die Leiterplatte zu löten, sondern sie etwas erhöht anzubringen, damit sie von keinem anderen Bauelement auf der Leiterplatte überragt werden kann. Wenn man auf eine gesonderte Befestigung der Fassung verzichten möchte, ist es unbedingt notwendig, die verlängerten Zuleitungen auch auf der Leiterseite anzulöten. Auf die Fassung werden acht festverdrahtete Adressleitungen und acht festverdrahtete Datenleitungen geführt. 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-Programmiergeräte zu lösen, benutzen die Autoren aus /1/ einen Typstecker als variablen Leitungsverteiler für Adress-Signale, Betriebsspannungen, Programmierimpulse, Programmierspannung und Schaltkreissteuerung.

Bild 3 zeigt die Belegung der Signale, die bei geeigneter Verbindung für die jeweilige EPROM-Größe die geforderten Anschlussbedingungen ergeben. In den Bildern 4.1 bis 4.6 sind die Typstecker aufgeführt. 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. Für die Typen 2764 und 27128 stammt die Schaltung aus /7/ und musste ebenfalls berichtigt werden. Die übrigen zwei Typstecker wurden vom Autor entworfen, wobei jedoch nur der für 32 K-EPROMs getestet werden konnte. Zum Typstecker nach Bild 4.6. sei vermerkt, dass er vom EPROM-HANDLER V 2.9 nur indirekt gestützt wird (siehe 2.3). Als Typsteckverbindertyp wurde ein 26poliger direkter Steckverbinder (TGL 29331/01) gewählt, wie ihn auch die Autoren in /1/ vorschlagen. Dies gestattet es, sich die Typstecker relativ leicht selbst herzustellen. Solche Abweichungen können Potentiometer sein, die anstelle des Festwiderstandes stehen und so die Programmierspannung kontinuierlich zu verändern gestatten. Auch zusätzliche Leuchtdioden, die die anliegende Betriebsspannung signalisieren, sind denkbar, wenn man VB 4 in Bild 1 nicht auf der Leiterplatte unterbringen will. Selbstverständlich kann auf der Typsteckerleiterplatte eine Hilfsspannungsquelle (Batterie, Transverter) untergebracht werden.

Adresszaehler

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

In /1/ werden für die Adresszählerschaltkreise 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 Rückwärtszähleingang wurde verzichtet, da er nicht unmittelbar notwendig ist.

Insgesamt hat der Adresszähler einen Zählumfang von 15 bit, so dass man ohne zusätzliche Maßnahmen nur EPROMs bis zu einem Fassungsvermögen von 32 K-Byte bearbeiten kann. Mit Hilfe des Typsteckers nach Bild 5.1. lässt sich die Einschränkung überwinden, 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 gehören. Gemeinsam sorgen sie dafür, dass mit Hilfe des Typsteckers eine übliche Transistorschaltstufe (VT4, R12 (R13)) hergestellt werden kann. Wenn VT4 durchgesteuert ist, entsteht an seinem Kollektor Low-Potential. Über 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 dafür, dass das Highpotential den Vorrang hat, auch wenn der steuernde PIO-Ausgang, nach dem Einschalten des Z 1013, ein undefiniertes Signal führt. 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 über. Je nach Herstellervorschrift lässt sich an den OE-Pins der EPROMs nun vermittels der Typstecker ein gewünschtes Potential herstellen. Sicherheitshalber sollte abweichend von SS 218 im Bild 1 ein Typ gewählt werden, der eine maximale Kollektor-Emitter-Spannung von Größer als 30 V (UPRGmax. = 30 V) besitzt.

Programmierspannungserzeugung

Die Programmierspannungserzeugung verläuft in mehreren Stufen und beginnt mit der Rohspannungserzeugung außerhalb der Programmiererleiterplatte. Sie kann bis zu 0,5 V verbrummt sein und sollte mit mehr als 60 mA belastbar sein, ohne um mehr als 1 V zurückzugehen. Diese Werte haben sich bei Untersuchungen ergeben, die davon ausgehen, dass EPROMs mit einer Programmierspannung von größer als 26 V sehr selten sind. Im ungünstigsten 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. Über den Spannungsteiler R1, R2 wird sie wunschgemäß justiert. Die Widerstandswerte von R1 und R2 sind so gewählt, dass ohne Typstecker und ohne bestätigten Paketschalter 12 V am Ausgang des MAA 723 (N1) erscheinen. Als besonderer Grund ist hierfür der Umstand zu nennen, dass die niedrigste bekannte Programmierspannung bei 12,5 V liegt. Alle anderen geforderten Werte bewegen sich oberhalb dieser Grenze und können erzielt werden, indem man den Widerstand R2 vermindert. Diese Verminderung ist nach /1/ am leichtesten durch einen Paralellwiderstand zu erreichen, der zwischen den Anschlüssen B1 und B6 des Typsteckers angebracht wird. Hieran ist nachteilig, dass nicht nur für n EPROM-Größen, sondern auch für m verschiedene Programmierspannungen Typstecker herzustellen sind. Im Extremfall müssten n x m Typstecker angefertigt werden. Der Ausweg bietet sich hier durch einen zusätzlichen Schalter an, durch den sich die Programmierspannung unabhängig vom Typstecker einstellen lässt. Nicht nur der Umstand der eingesparten zusätzlichen Typstecker, sondern auch die Möglichkeit bei unbekannter Programmierspannung sich an einen akzeptablen Wert heranzutasten, lassen diesen Paketschalter als attraktiv erscheinen. Neben den sieben Schaltstufen für 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 dafür vorgesehen, um vorhandene Typstecker mit Widerstand einsetzen zu können.

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

```
2716
2764
27128 und
27256
```

für die Programmierung versorgt werden, denn sie benötigen eine Gleichspannung /8/, /9/ der vom Hersteller geforderten Höhe.

Als zweite Stelle ist ein Chopper zu nennen, der als Gegentakt-Komplementär-Stufe aufgebaut ist und aus den Transistoren VT 1, 2, 3 nebst den Widerständen R4…R10 besteht. Alle EPROM-Typen, die eine pulsierende Programmierspannung benötigen (2708, 2732 und 27512), werden über 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 Überschwingen soll das RC-Glied R4, C2 unterdrücken, da es empfindliche EPROM-Exemplare zerstören kann.

Um den Chopper zu bedienen, wird an den Basen von VT1 und VT2 als Steuerspannung eine gegenphasige Impulsfolge benötigt. Ein monostabiler Trigger (D5) mit komplementaeren Ausgangssignalen ist hierfür besonders geeignet. Er sichert nämlich, dass sein stabiler Zustand für die ausgeschaltete Programmierspannung steht und im Falle von Programmabstürzen UPRG auf 0 V zurückkehrt.

Die Dauer eines Programmierspannungsimpulses wird von der R/C-Kombination C3, R14 bestimmt und liegt nach Empfehlung aus /1/ bei 1 ms. Sollte nämlich 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 längere Lebensdauer dieses EPROM-Exemplares ist dann eine angenehme Folge jenes Verzichtes.

Vom Rechner wird der Programmierspannungs-Impulserzeuger über Pin 5 des Schaltkreises D5 mit einer Low-High-Flanke angestoßen (vgl. Bild 6). Nach der Auslöseflanke durchläuft das Programmierprogramm EPROM-HANDLER V 2.9 eine Warteschleife. Nach dieser Warteschleife (Pauschalzeit) erfolgt zunächst eine Tastaturabfrage nach dem Abbruchzeichen CTRL C. Bei einer Rechnertaktfrequenz 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 für die Pauschalzeit verantwortlich und liefert beim Wert von 20H 256 us.

EPROM-Typen, die keine pulsierende Programmierspannung benötigen, haben einen TTL-gerechten Programmierimpuls. Er gibt die Programmierspannung erst im Inneren des EPROMs zur Wirkung frei. Von den gegenphasigen Ausgängen des 74121 werden diese Programmierimpulse auf den Typsteckverbinder X2 (Bild 3) geführt, 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 günstig gewählt, denn die Impulsfolge an Pin 6 von D5 hat ein Tastverhältnis von 0,08. Die mäßig 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 könnte auf VB1 verzichten, wenngleich 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 können, wird der Spulenstrom über die Leuchtdiode VB3 geführt. Der Widerstand R24 liefert einen Nebenschluss, um einen Teil des Spulenstromes umzuleiten. Seine Größe richtet sich nach der verwendeten Leuchtdiode und nach der gewünschten Helligkeit.

Das Relais K2 verfügt über 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 wäre durchaus ratsam, die eingeschalteten EPROM-Betriebsspannungen anzuzeigen, denn das kann ein Achtungszeichen sein, bei spannungsführender Fassung, z. B. bei der Funktion 3 (Vergleichen), den EPROM nicht abzunehmen. VB4 ist nicht auf der Bestückungszeichnung enthalten.

Anschluss an den Z 1013

Der Anschluss des Programmiergerätes an den Z 1013 geschieht vermittels einer zusätzlichen PIO, da das USER-Port am Steckverbinder X4 nicht ausreicht. Von den 16 Portanschlüssen der PIO (A0…A7; B0…B7) werden 15 verwendet. Die ihnen zugeteilten logischen Signale mit ihren Spezifikationen sind in Tab. 3 aufgeführt.

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

Ist man gezwungen, längere Anschlusskabel einzusetzen, ergeben sich Schwierigkeiten. Diese sind durch die Anpassungsverhältnisse an der Anschlussleitung bedingt, denn sie ist bei 200 Ohm Wellenwiderstand durch niederohmige Sender (50 Ohm) und durch hochohmige Empfänger (6 kOhm) beschaltet. Nach einem graphischen Lösungsverfahren, das in /11/ beschrieben ist, wurde der Signalverlauf am Leitungsende ermittelt (Bild 7). Bemerkenswert ist hierbei das positive Überschwingen. 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 ausgeloest 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 Überschwingimpulse 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 Überschwingen. 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 Überschwingen 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 Schaltungen	problematisch, drei Spannungen im Rechner nötig
Programmierspannung	problemlos, übliche Schaltungen	problematisch, Verwendung des Koppelbusses bei K 1520-kompatiblen Baugruppen, Abweichungen von der Systembusbelegung auf alle Fälle vermeiden), Aufbau eines Transverters auf dem PIO-Modul, separate Zuführung
Schaltungsaufwand	mittlerer Aufwand	mittlerer Aufwand
mechanischer Aufwand	sehr hoch	gering
Anschluss	lange Leitungen	kurze Leitungen
Handhabbarkeit	gut, an anderen Orten Rechnertypen einsetzbar	eingeschränkt, da an Ort und Aufbau gebunden

In Bild 4 ist der Vorschlag eines externen Netzteiles abgebildet. Die Stromergiebigkeit der 17 V-Gleichspannungsquelle sollte mindestens 500 mA betragen. Nachstehend aufgeführte Tabelle gibt über die Stromaufnahme des EPROM-Programmierers Auskunft:

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

Anmerkungen

1) Für 1 KByte-EPROMs kann auch die Kennziffer 55 angegeben werden, da der der internationale Typ 2708 in der DDR die Bezeichnung U 555 C trägt.

2) Das Verlassen des EPROM-Handlers mit RESET ist während des aktiven Programmierens 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 Applikation“ Nr. 41 Halbleiterspeicher Teil 3 / Buerger, B. - KdT Frankfurt/O., 1987 S. 42

/ 4/ Buerger, B.: Reihe „Information Applikation“ Nr. 41 Halbleiterspeicher Teil 3 / Buerger, B. - KdT Frankfurt/O., 1987 S. 40

/ 5/ Mueller, M.; Krueger, S.: EPROM-Programmiergerät / Mueller, M.; Krueger, S. Radio Fernsehen Elektronik. - Berlin 31 (1982) 10. S. 621

/ 6/ Kuehnel, Th.: Einfacher EPROM-Programmiermodul / Kuehnel, T. Radio Fernsehen Elektronik. - Berlin 35 (1986) 10. S. 633

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

/ 8/ Mueller, M.: EPROM-Programmiergerät / 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/O., 1987

/10/ Brosig, R.: Z-1013-Tastatur mit Raffinessen. Mikroprozessortechnik 7/88, S. 215

/11/ Turinsky, G.: Leitungsabschlussschaltungen / Turinsky, G. Radio Fernsehen Elektronik. - Berlin 32 (1983) 8. S. 182

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

Last update: **2012/10/31 08:53**