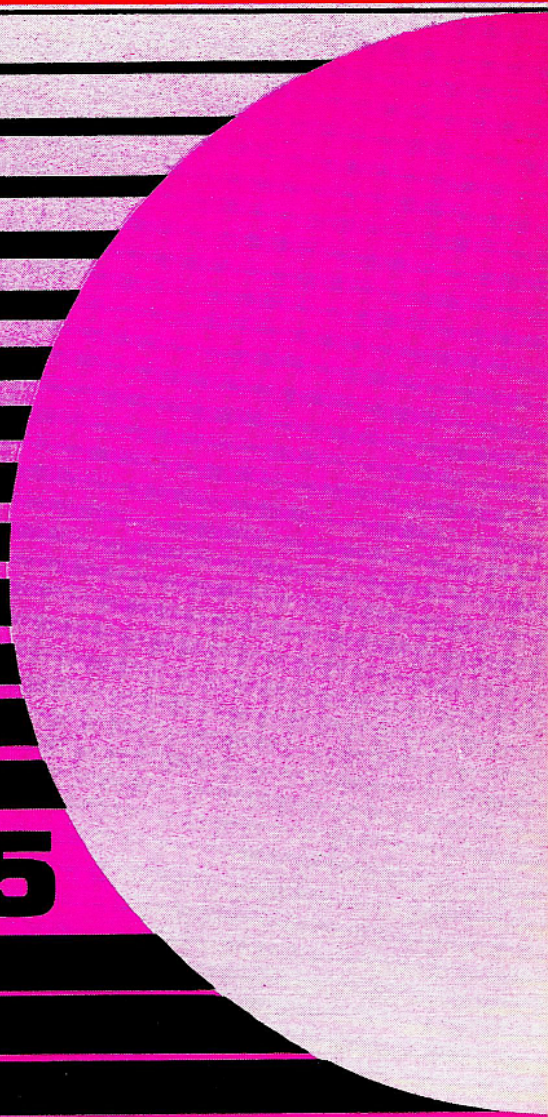


KLEINCOMPUTER



KC 85

M 030

EPROMMER

KLEINCOMPUTER

KC 85

Beschreibung zu M030 EPROMMER



veb mikroelektronik
'wilhelm pieck'
mühlhausen

KC-CLUB

René Nitzsche
Wolfgang Harwardt
Mario Leubner

09.11.2014 - 07.08.2016

Inhaltsverzeichnis

1. Einleitung.....	8
1.1. Theoretische Grundlagen zum EPROM.....	10
1.2. Aufbau und Funktionsweise.....	10
1.3. Löschen von EPROMs.....	11
1.4. EPROMs im KC85-System.....	11
2. Hinweise zum Aufbau.....	12
2.1. Layout und Bauteile.....	13
2.2. Inbetriebnahme (Abgleich Analogteil).....	15
2.3. Optionale Bestückung.....	16
2.4. Beschreibung der Steckbrücken.....	17
2.5. Umschalter 32K / 64K – Modus (Strukturbyte).....	18
3. Modulhandhabung.....	19
3.1. Stecken und Entfernen des Moduls.....	19
3.2. Stecken und Entfernen von EPROMs.....	20
3.3. Modulzuweisung und Steuerbyte.....	21
3.4. Portadressen des M030.....	22
4. Softwarebeschreibung im 32K-Modus.....	23
4.1. Hauptmenü.....	23
4.2. Brennermenü.....	24
1=READ.....	25
2=COMPARE.....	25
3=CHECKSUM.....	26
4=CHECK OF BYTECHANGE.....	26
5=BLANKTEST.....	26
6=PROGRAMMING.....	26
7=DISPLAY (RAM/ROM).....	27
8=TAPE.....	27
9=EXIT.....	27
4.3. Inbetriebnahmehilfe.....	27

5. Softwarebeschreibung im 64K-Modus.....	28
5.1. Beschreibung der speicherbezogenen Menüworte.....	30
>HELP.....	31
>MENU [n].....	31
>SWITCH [Steckplatz [Steuerbyte]].....	32
>LOAD [Speicherreferenz].....	33
>SAVE [Anfangsadresse [Endadresse+1]].....	33
>FLOAD [Speicherreferenz].....	34
>FSAVE [Anfangsadresse [Endadresse+1]].....	34
>MOVE <Anfang Quelle> <Anfang Ziel> <Länge Speicherbereich>.....	34
>FILL [<Anfangsadresse> <Endadresse+1>] [<Byte> [<Byte>...]].....	35
>MVIEW [<Anfangsadresse>].....	35
>DISP [Adresse [Zeilen [Bytes]]].....	36
>MCRC [<Anfang Speicher> [<Ende Speicher+1>]].....	37
>QUIT.....	37
5.2. Beschreibung der EPROM-bezogenen Menüworte.....	38
>EPROM [<EPROM-Typ in KByte> [Methode <A/B>] [Vpp <12/21/25>]].....	38
>TIME n1 t1 n2 t2.....	39
>EVIEW [<Anfangsadresse>].....	40
>EDISP [<Adresse> [Zeilen [Bytes]]].....	40
>ERAT [<EPROM-Bereich>] [<Speicherreferenz>].....	40
>READ [<EPROM-Bereich>] [<Speicherreferenz>].....	41
>PROG [<EPROM-Bereich>] [<Speicherreferenz>].....	42
>COMP [<EPROM-Bereich>] [<Speicherreferenz>].....	44
>ECRC [<Anfang EPROM> <Ende EPROM+1>].....	45
5.3. Versteckte Menüworte.....	46
>modify [Adresse [Bytes]].....	46
>arith <Hexzahl1> <Hexzahl2>.....	46
5.4. Inbetriebnahmehilfe.....	47
6. Schaltplan.....	50
6.1. Digitalteil.....	50
6.2. Optionale 27(C)512-Erweiterung bei Neuauflage 2015.....	51
6.3. Analogteil.....	52
7. M030 Leiterplatte und Bestückung.....	53
8. M030 Stückliste.....	54
9. Programmierung von 1K-EPROMs.....	57
9.1. 2708 – Adapter (ML).....	58
9.2. 2708 bzw. TMS2716 – Adapter (WH).....	60
9.3. Stücklisten der beiden 2708 – Adapter.....	62

10. Pinbelegung der EPROM-Typen.....	63
11. Übersicht der getesteten EPROMs.....	64
11.1. Zeitdiagramme 1K-EPROM 2708 / U555C.....	65
11.2. Zeitdiagramme 2K-EPROM 27(C)16.....	66
11.3. Zeitdiagramme 4K-EPROM 27(C)32.....	67
11.4. Zeitdiagramme 8K-EPROM 27(C)64.....	68
11.5. Zeitdiagramme 16K-EPROM 27(C)128.....	69
11.6. Zeitdiagramme 32K-EPROM 27(C)256.....	70
11.7. Zeitdiagramme 64K-EPROM 27(C)512.....	71
12. Internet-Links zum M030.....	72

1. Einleitung

Das Modul M030 EPROMMER wurde vom VEB Applikationszentrum Berlin entwickelt und vom VEB Mikroelektronik Wilhelm Pieck in Mühlhausen in nur sehr geringer Stückzahl produziert. Die Platine ist länger als eine normale Modulplatine, damit die Schwenkhebelfassung zum Einsetzen und zur Entnahme der EPROMs vorn heraus ragt. Deswegen hatte das Modul auch zwangsläufig keine normale Frontblende, sondern ein nach vorn verlängertes Modulgehäuse. Es sind fest eingestellte Programmierspannungen von 12,5V, 21,0V oder 25,0V auswählbar. Dieses M030 besitzt das Kennbyte D9h und gestattet das Programmieren von 2K- bis 32K-EPROMs vom Typ 2716 bis 27(C)256.

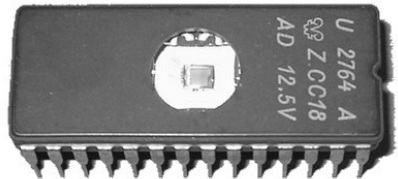
Es wurde bereits ein weiteres Kennbyte DBh vergeben, welches ein M030 zur Programmierung von EPROMs bis 64K (27512) beschreibt. Diese Variante des M030 ist jedoch nicht bekannt geworden, auch keine technischen Details wie die 64K-Erweiterung realisiert werden sollte.



1.1. Theoretische Grundlagen zum EPROM

Auszugsweise nach Wikipedia

Ein EPROM (engl. Abk. für Erasable Programmable Read-Only Memory, wörtlich: löschbarer programmierbarer Nur-Lese-Speicher) ist ein nichtflüchtiger elektronischer Speicherbaustein vor allem für die Computertechnik.



Dieser Bausteintyp ist mit Hilfe spezieller Programmiergeräte (genannt „EPROM-Brenner“) programmierbar. Er lässt sich mittels UV-Licht löschen und danach neu programmieren. Nach etwa 100 bis 200 Programmier- und Löschvorgängen hat der EPROM das Ende seiner Lebensdauer erreicht. Das zur Löschung nötige Quarzglas-Fenster (normales Glas ist nicht UV-durchlässig) macht das Schaltkreis-Gehäuse relativ teuer. Daher gibt es auch Bauformen ohne Fenster, die nominal nur einmal beschreibbar sind (One Time Programmable, OTP), sich durch Röntgenstrahlung aber ebenfalls löschen lassen.

Der EPROM wurde 1970 bei Intel entwickelt. EPROMs wurden bis etwa in die Mitte der 1990er-Jahre eingesetzt, inzwischen aber weitgehend durch EEPROMs oder Flash-EEPROMs abgelöst.

1.2. Aufbau und Funktionsweise

Ein EPROM enthält eine Matrix aus Speicherzellen, in denen jeweils ein Transistor ein Bit repräsentiert. Eine Speicherzelle besteht aus einem MOSFET-Transistor mit einer isolierten Hilfselektrode, dem Floating-Gate. Beim Programmieren wird eine erhöhte Spannung angelegt, so dass durch den Avalanche-Effekt das Floating-Gate geladen wird. Dadurch verschiebt sich die Ansteuerspannung, bei der der Transistor einschaltet. Die Daten lassen sich nun beliebig oft auslesen, wobei die Lesespannung unterhalb der Programmiervoltage liegt. Die zum Löschen verwendete Ultraviolettstrahlung bewirkt eine großflächige Ionisation des Halbleitermaterials, so dass die vorher aufgebrauchte elektrische Ladung das Floating-Gate des Transistors wieder verlassen kann. Das Bitmuster ist dadurch gelöscht und der EPROM in seinen ursprünglichen Zustand zurückversetzt. Die fensterlosen OTPs lassen sich mit Röntgenstrahlung löschen, da diese auch ohne Fenster durch das Gehäuse dringt und der Baustein selbst bis auf das Gehäuse der gleiche ist.

In der Bezeichnung eines EPROMs sagt das "C" nach der 27 aus, dass es sich um ein CMOS EPROM handelt (CMOS=Complimentary Metal Oxide Semiconductor). Diese benötigen wesentlich weniger Leistung als die alten NMOS EPROMs (NMOS=N-channel Metal Oxide Semiconductor) und kommen mit nied-

rigeren Programmierspannungen (12,5 Volt) aus. Da beide Bauteile ansonsten kompatibel sind, können die alten NMOS EPROMs durch CMOS EPROMs gleicher Größe ersetzt werden (z.B. Tausch eines 2764 durch einen 27C64).

1.3. Löschen von EPROMs

Das Löschen von EPROMs erfolgt durch das Bestrahlen mit intensivem ultraviolettem Licht im Bereich von 254 nm Wellenlänge in speziellen EPROM-Löschgeräten. Ein konventioneller Löschvorgang dauert ca. 10 bis 30 Minuten. Da die Ionisation nach dem Ausschalten der Lichtquelle nicht sofort wieder abgeklungen ist und die Bausteine je nach Bauart des Löschgerätes auch über die für das Programmieren zulässige Temperatur hinaus erhitzt werden, kann das Programmieren erst nach einer weiteren Wartezeit erfolgen. Falls die Vorgaben des Bausteinherstellers für das Löschen nicht korrekt eingehalten werden, kann eine scheinbar richtige Programmierung mit verkürzter Datenlebensdauer die Folge sein.

Das Quarzglas-Fenster sollte nach dem Programmieren mit einem lichtundurchlässigen Aufkleber geschützt werden. Ein ungeschützter EPROM kann nach ca. 90 Tagen direkter Sonneneinstrahlung gelöscht sein. Die Beleuchtung der Speicherzellen mit einem gewöhnlichen Fotoblitzzgerät kann kurzzeitige Datenverfälschung und damit Computerausfälle verursachen.

1.4. EPROMs im KC85-System

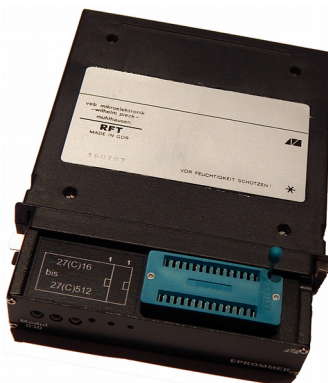
Im KC85-System sind EPROMs im Grundgerät, im D004, in der D005-Tastatur sowie in den Modulen M006, M012, M025, M026, M027, M028, M030, M033, M040, M045, M046, M047, M048 und M062 eingesetzt. Alle dort verwendeten EPROM-Typen sind mit dem Modul M030 programmierbar.

2. Hinweise zum Aufbau

Von René Nitzsche

Die Platine ist vollständig beschriftet mit Bauteilenamen und -werten. Für die Potentiometer und Relais sind verschiedene Pinouts möglich. Als Dioden sind SAY vorgesehen, es können problemlos zurecht gebogene Dioden in Axialbauform eingesetzt werden. Beim Bestückungsaufdruck der Dioden ist die Kathode markiert.

Die Platzierung des Programmiersockels (Schwenkehebbefassung) richtet sich nach dem vorgesehenen Gehäuse. Soll ein Gehäuse verwendet werden, basierend auf der entwickelten STL als 3D-Druck-Gehäuse, dann muss der Sockel so eingebaut werden, dass der Hebel nach links zeigt. Für das ebenfalls entwickelte Alu-Gehäuse muss der Hebel nach rechts zeigen. Bei Verwendung eines der beiden Gehäuse-Varianten ist eine Erhöhung des ZIF-Sockels erforderlich, da er sonst im Gehäuse verschwinden würde. Der in der Stückliste angegebene IC-Steckadapter kann direkt auf die Platine gelötet werden, darauf dann direkt den ZIF-Sockel auflöten. Es sind aber auch andere Varianten zur Erhöhung möglich. Ist kein Gehäuse vorgesehen, muss sich der Schwenkehebel rechts befinden und der Sockel kann ohne Erhöhung direkt eingelötet werden. In diesem Fall muss der Umschalter "ESW" aber durch ein flaches Modell (siehe auch Stückliste) ersetzt werden.



Sicherheitshinweis

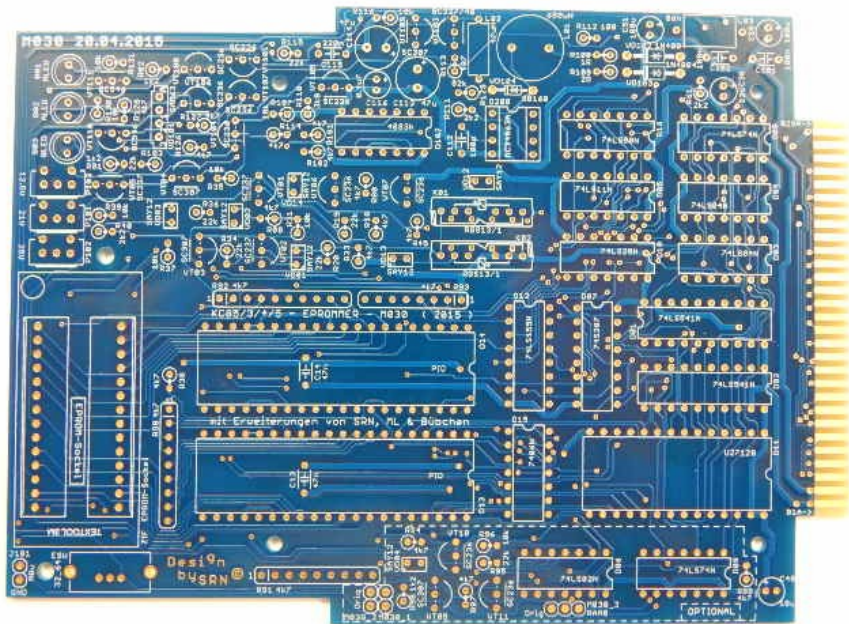
Vor Arbeiten am Computer und elektronischen Baugruppen immer einen Ladungsausgleich schaffen. Das heißt zum Beispiel einen mit Masse verbundenen Punkt am Computer oder der zu bearbeitenden Baugruppe anfassen.

Bauelemente von einer Leitschaumplatte oder Aluminiumfolie entnehmen die vorher berührt wurde, um einen Ladungsausgleich zu schaffen.

Die modernen CMOS-Bauelemente reagieren aufgrund ihrer Hochohmigkeit besonders empfindlich auf statische Ladungen. Normale TTL- und NMOS-Bauelemente sind etwas toleranter, aber auch dort ist ein Ladungsausgleich angebracht.

2.1. Layout und Bauteile

Von René Nitzsche



Das hier abgebildete Layout stellt die endgültige Fassung dar, so wie die Platine gefertigt worden ist.

So wie die Bauteile auf der Platine beschriftet sind, können sie bestückt werden. Da es sich bei den Angaben zum größten Teil um Originalbauteile aus der DDR handelt, und diese nicht bei jedem/jeder zur Verfügung stehen, können auch Äquivalenzbauteile eingebaut werden.

Welche Äquivalenzen das sind, kann der Stückliste entnommen werden, die im Anschluss an den Bestückungsplan aufgeführt ist (Spalte "Reichtelt").

Schaltkreisfassungen

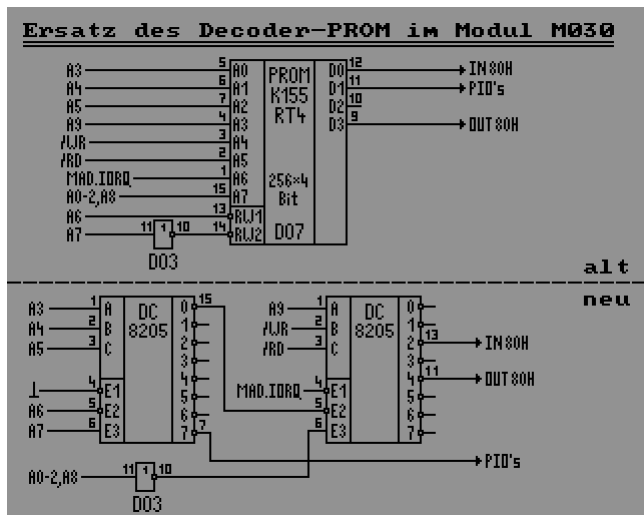
Nur für die PIO-Schaltkreise sind auf dem Layout Stützkondensatoren vorgesehen.

Für alle anderen Schaltkreise müssen, wenn Stützkondensatoren "gewünscht" werden, Fassungen eingebaut werden, welche mit Stützkondensatoren versehen sind. Alternativ können diese Kondensatoren auch auf der Lötseite einzeln angebracht werden.

Nachträgliche Modifikationen

Nach der Fertigstellung der Platine haben sich noch einige Änderungen herausgestellt, die die Funktionssicherheit des M030-Nachbaus erhöhen. Es wird empfohlen, diese Änderungen auch nachträglich noch vorzunehmen:

1. Es sollte eine zusätzliche Diode VD05 in Reihe mit dem Widerstand R31 zu den Kollektoren von VT02 und VT03 geschaltet werden. Anode Richtung R31, Kathode Richtung Kollektoren. Diese Diode kann problemlos auf der Bestückungsstelle des R31 ergänzt werden – beide Bauteile werden dann oben verlötet.
Sonst kann es vorkommen, dass die Programmierspannung auf die PIO gerät bei offenem Kontakt von K02.
2. Der Widerstand R94 sollte von 4,7kOhm auf 1,2kOhm reduziert werden und für die Diode VD04 eine Schottky-Diode BAT42 anstelle der SAY12 eingesetzt werden.
Sonst kann es vorkommen, dass Pin 1 beim Lesen einiger EPROMs 2764, 27128 und 27256 nicht sicher auf High-Pegel geht und fehlerhafte Daten gelesen werden.
3. Es gibt eine Ersatzschaltung für den PROM D07 (K155RT4 bzw. 74S287), welche aus zwei DS8205 bzw. 74LS138 besteht. Diese kam mit dem M030-Nachbau nicht zum Einsatz, soll der Vollständigkeit halber hier im Handbuch aber mit gezeigt werden:



2.2. Inbetriebnahme (Abgleich Analogteil)

Von Wolfgang Harwardt

Vor dem Einstellen und Test komplett durchlesen!

Nach dem Bestücken aller Bauelemente des Analogteiles, der Schaltkreisfassungen des Digitalteiles, der Kondensatoren und der SIL-Widerstände unter Berücksichtigung der richtigen Polarität und Einbaulage der Bauelemente, ist erst einmal die Programmierspannung voreinzustellen.

Nur die zum Analogteil gehörenden Schaltkreise MC34063A und 4093 sind zu stecken! Die Einstellregler P101-P103 etwa auf folgende Werte voreinstellen:

P101: 5k0 (zuständig für $V_{pp}=21V$)

P102: 3k0 (zuständig für $V_{pp}=25V$)

P103: 6k8 (zuständig für $V_{pp}=12,5V$)

Damit wird vermieden, dass Spannungswerte der Programmierspannungen auftreten, die gefährlich sind und Bauelemente zerstören können.

Die Leiterplatte ist dazu mit +12V an 12p zu versorgen. An Vcc sind +5V anzuschließen.

Zum Test und zur Einstellung an die in der Schaltung zu findenden Leitungen 74 und 75 auf der Leiterplatte provisorisch Drähte zum Beispiel in die PIO-Fassung stecken (Pins der Schaltung entnehmen). Sobald eine der beiden Leitungen mit Masse (z.B. an der Schaltkreisfassung) verbunden ist, sollte die rote LED B02 leuchten und an J101 Pin 1 die Programmierspannung anliegen. Die Spannung wird nur zu J101 Pin 1 über VT103 durchgeschaltet wenn die LED B02 leuchtet!

Die Messbuchse J101 sitzt vorn an der Leiterplatte direkt neben der Programmierfassung. Sie besitzt einen Masseanschluss (rechts) und ein Programmierspannungsanschluss (links). Ich habe dort einen Pfostenfeldstecker aufgelötet. Darauf eine Leitung aus einem ausgeschlachteten PC gesteckt, der dort von den Frontplatten-Bedienelementen zum Motherboard führt, und an diese Leitung einen Spannungsmesser angeschlossen.

Als Erstes ist die 12,5V-Programmiererspannung abzugleichen, erst danach die anderen beiden Spannungen da die Einstellung des Einstellreglers P103 auch geringfügig Einfluss auf die 21V und 25V hat!

12,5V: Dazu die Leitung 74 an Masse legen und Leitung 75 von Masse getrennt lassen. Dann mit dem Einstellregler P103 die Programmierspannung auf 12,5V einstellen.

21V: Anschließend wird die Leitung 74 von Masse getrennt und Leitung 75 an Masse gelegt. Mit dem Einstellregler P101 jetzt die Programmierspannung auf 21V einstellen.

25V: Zum Schluss beide Leitungen 74 und 75 mit Masse verbinden und mit dem Einstellregler P102 die Programmierspannung auf 25V einstellen.

Sämtliche Bauelemente vom 4093 Pin 3 bis zur Basis des PNP-Transistors SF818 (BC560) sollten nicht mehr als 5% Toleranz aufweisen!

Sind alle Spannungen eingestellt, dann die Spannungsversorgung und die provisorischen Drähte wieder entfernen. Jetzt erst alle anderen Bauelemente bestücken und die Schaltkreise aufstecken.

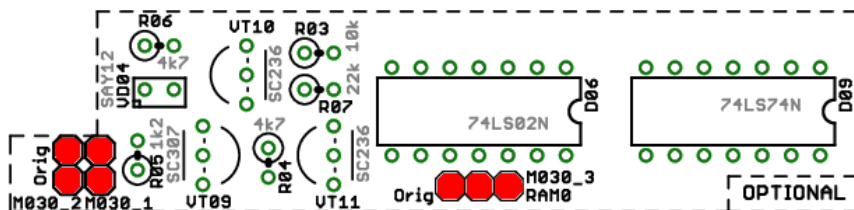
Damit steht der Erprobung im Computer mit dem auf dem EPROM befindlichen Programm nichts mehr im Wege.

2.3. Optionale Bestückung

Von René Nitzsche

Im unteren, rechten Bereich der Platine ist ein Bereich zu sehen, welcher mit einer gestrichelten Linie umrandet und als "OPTIONAL" bezeichnet ist.

Soll das Modul ausschließlich in seiner Originalfunktionalität betrieben werden, müssen diese Bauteile nicht bestückt werden. Das gilt dann auch für den Umschalter. Des Weiteren reicht dann als Steuer-EPROM ein 27(C)64 aus. Die drei Steckbrücken müssen jedoch bestückt bzw. können durch Drahtbrücken oder 0-Ohm-Widerstände ersetzt werden.



2.4. Beschreibung der Steckbrücken

Von René Nitzsche

Auf der Platine befinden sich drei Steckbrücken. Die Platine ist so hinzulegen, dass die Beschriftung ersichtlich ist und der Edge-Connector sich rechts befindet. Dann befinden sich diese Steckbrücken im optionalen Bereich, der durch eine Strich-Linie eingerahmt ist:

M030_1 und M030_2

Beide Steckbrücken gesetzt deaktivieren die optionale Schaltung, falls diese Schaltung NICHT bestückt ist. Es müssen beide Brücken gesteckt sein.

Ist die Zusatzschaltung bestückt, dann dürfen die Brücken NICHT gesteckt sein.

M030_3

Ebenfalls im Zusammenhang mit der optionalen Schaltung steht diese Steckbrücke.

Hiermit wird gesteuert ob EPROM und PIOs separat oder gemeinsam per SWITCH-Befehl geschaltet werden.

- Steckbrücke links → ohne bestückten Schaltkreis D09 muss die Brücke links gesteckt werden. Die Funktion entspricht dem Originalmodul, PIO und EPROM werden zusammen mit Bit 0 des Steuerbytes geschaltet.
SWITCH <Steckplatz> 1 schaltet das Modul ein.
- Steckbrücke rechts → mit bestücktem Schaltkreis D09 kann die Brücke rechts gesteckt werden. Dann werden PIO und EPROM separat geschaltet mit Bit 0 bzw. Bit 3 des Steuerbytes. Die neu entwickelte Software schaltet dann den EPROM automatisch ab, nachdem die Software zur Abarbeitung in den RAM0 kopiert wurde. Somit ist es der neuen Software möglich auf RAM- oder ROM-Bereiche im Adressbereich C000h-DFFFh zuzugreifen.
SWITCH <Steckplatz> 1 schaltet nur die PIOs vom Modul ein.
SWITCH <Steckplatz> 8 schaltet nur den EPROM vom Modul ein.
SWITCH <Steckplatz> 9 schaltet das Modul komplett ein.

Die Steckbrücke M030_3 kann mit bestückter Zusatzschaltung auch links gesteckt werden, wenn PIO und EPROM weiterhin zusammen geschaltet werden sollen.

2.5. Umschalter 32K / 64K – Modus (Strukturbyte)

Von Mario Leubner

Die meisten Module senden auf Anforderung ein Modulstrukturbyte. Dieses Strukturbyte widerspiegelt den Modultyp bzw. die innere Strukturierung des Moduls. Das Strukturbyte kann durch den Prozessor (auch im inaktiven Zustand des Moduls) gelesen werden und dient damit der Erkennung der Module im KC-System. In einem ausgebauten KC-System kann sich der Nutzer jederzeit einen Überblick über die kontaktierten Systemkomponenten verschaffen. Software kann in Abhängigkeit der Modulbestückung ablaufen, zum Beispiel eine Fehlermeldung ausgeben, wenn ein benötigtes Modul nicht im KC-System vorhanden ist.

Der M030-Nachbau besitzt zwei Strukturbytes, die sich je nach Stellung des Umschalters ergeben:

- Schalter in Stellung 32K = Strukturbyte D9h
- Schalter in Stellung 64K = Strukturbyte DBh



Umschalter

3. Modulhandhabung

3.1. Stecken und Entfernen des Moduls

Das EPROMMER-Modul M030 kann prinzipiell in jedem Modulsteckplatz betrieben werden, jedoch ist dabei die Modulpriorität in der gewählten Systemkonfiguration zu berücksichtigen. Der gleichzeitige Betrieb von mehreren Modulen M030 an einem KC85 ist nicht vorgesehen.

Es sind deshalb die nachfolgenden Informationen und Hinweise zu beachten.

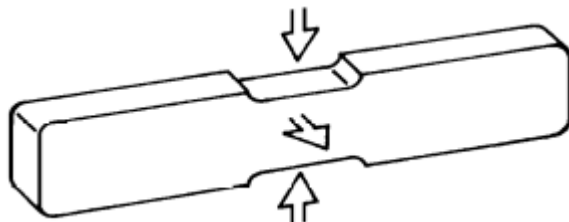
Die Modulprioritätskette muss geschlossen bleiben. Also sind zuerst im Grundgerät der Steckplatz 8 (rechts), danach der Steckplatz C (links) und anschließend weitere Steckplätze von Erweiterungsaufsätzen in aufsteigender Reihenfolge zu belegen.

Achtung!

Das Stecken und Entfernen des Moduls darf nur im ausgeschalteten Zustand des Systems vorgenommen werden!

Damit ergeben sich folgende Handgriffe für das Stecken des Moduls:

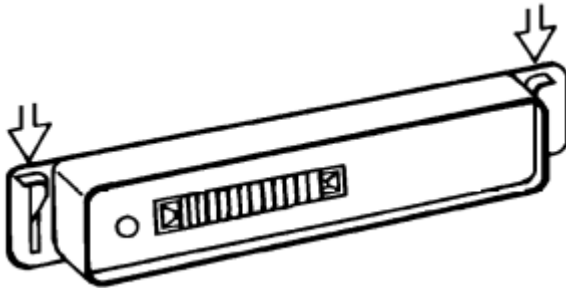
- a) Den Computer ausschalten.
- b) Die Kappe des Modulschachtes abnehmen. Hierzu muss die Kappe an den gegenüberliegenden Griffflächen leicht zusammengedrückt werden.



- c) Das Modul bis zum fühlbaren Einrasten einschieben. Das Modul ist dann richtig kontaktiert, wenn sein hervorstehender Rand unmittelbar an der Gerätewand anliegt.
- d) Nun kann der Computer eingeschaltet werden.

Bevor die weitere Bedienung beschrieben wird, soll an dieser Stelle gleich das Entfernen des Moduls aus dem System erläutert werden. Zum Entfernen des Moduls sind folgende Schritte notwendig:

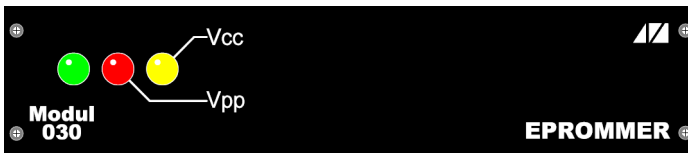
- a) Den Computer ausschalten.
- b) Den linken und den rechten Zeigefinger unter den Modulkopf legen und mit den Daumen die seitlich am Modul befindlichen Hebel gleichzeitig nach unten drücken. Dabei rastet das Modul aus und wird etwa einen Zentimeter aus dem Gerät heraus geschoben. Nun kann das Modul leicht aus dem Schacht gezogen werden.



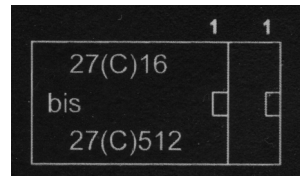
- c) Die Kappe auf die Schachttöffnung stecken.

3.2. Stecken und Entfernen von EPROMs

Das Stecken und Entfernen von EPROMs sollte grundsätzlich nur im spannungs-freien Zustand erfolgen. Zur Kontrolle besitzt das Modul an der Vorderseite drei LEDs: Die grüne LED zeigt die Aktivierung des Moduls an, die gelbe LED leuchtet, sobald die Spannung U_{cc} (5V) zugeschaltet ist, die rote LED leuchtet, wenn die Programmierspannung anliegt. Wenn sowohl die rote als auch die gelbe LED aus sind, kann der EPROM gefahrlos gezogen und gesteckt werden.



Der EPROM ist dabei so zu stecken, dass die Kerbe nach rechts zeigt und der EPROM linksbündig eingesetzt wird. Das Alu-Gehäuse enthält dazu auf der Oberseite eine entsprechende Beschriftung.



3.3. Modulzuweisung und Steuerbyte

Von Mario Leubner

Das EPROMMER-Modul M030 hat im KC System drei Betriebszustände. Der auf dem Modul befindliche EPROM kann – in Abhängigkeit von Steckbrücke M030_3 – getrennt von der restlichen Hardware zu- und abgeschaltet werden.

MODUL INAKTIV	grüne LED leuchtet nicht	Modul ist vom Prozessor getrennt
PIO AKTIV EPROM EIN	grüne LED leuchtet *1	PIOs sind vom Prozessor aus ansprechbar, das Modul kann arbeiten. Der EPROM des Moduls ist auf Adresse C000h zugeschaltet.
PIO AKTIV EPROM AUS	grüne LED leuchtet *2	PIOs sind vom Prozessor aus ansprechbar, das Modul kann arbeiten. Auf die Software im EPROM des Moduls ist kein Zugriff.

*1 in diesem Zustand arbeitet die 32K-Software

*2 in diesem Zustand arbeitet die 64K-Software

Der gewünschte Betriebszustand wird über das CAOS-Kommando

SWITCH mm kk

eingestellt. Die beiden Parameter dieses Kommandos realisieren:

- mm Mitteilung an das System, in welchem Modulschacht das zuzuweisende Modul kontaktiert ist. Dabei ist die erste Stelle von mm die Nummer des Aufsatzes (im Grundgerät ist diese Stelle Null und kann weggelassen werden). Die zweite Stelle von mm ist die Steckplatzadresse. Im Grundgerät existieren nur die Steckplatzadressen 8 (rechter Schacht) und C (linker Schacht).
- kk Betriebszustand des Moduls. Die erste Stelle von kk ist Null und kann weggelassen werden. Die zweite Stelle von kk kann sein:
 - 0 - PIOs und EPROM inaktiv
 - 1 - nur PIOs aktiv.
 - 9 - Modul und EPROM aktiv

Im Steuerbyte kk werden folgende Informationen binär verschlüsselt:

Bit 0: PIOs ein / aus

Bit 3: EPROM ein / aus

ACHTUNG !

Beim KC 85/3 unter CAOS 3.1 muss außerdem mit SWITCH 2 0 der Basic-ROM ausgeschaltet werden, damit der ROM-Bereich des Moduls auf der Adresse C000h für den Prozessor sichtbar wird.

3.4. Portadressen des M030

Von Mario Leubner

Das EPROMMER-Modul M030 enthält zwei PIO-Schaltkreise U855D bzw. Z80-PIO. Diese können über die Portadressen 0B8h bis 0BFh angesprochen werden:

PIO 1 (D13) (EPROM-Adressen 0..15)

0B8h	EPROM Adressbits 0..7
0B9h	EPROM Adressbits 8..15
0BAh	Steuerbyte Port A
0BBh	Steuerbyte Port B

PIO 2 (D14) (Port A Daten, Port B Steuersignale)

0BCh	EPROM Datenbyte
0BDh	Steuersignale:
	Bit 7 = 1 /OE = 0V
	Bit 6 = 1 /OE = Vpp
	Bit 5 = 1 A11 = Vpp
	Bit 4 = 1 K02: Ucc = 5V
	Bit 3+2 1 1 Vpp = 0V
	0 1 Vpp = 12,5V
	1 0 Vpp = 21V
	1 1 Vpp = 25V
	Bit 1 = 1 A13 = Ucc
	Bit 0 /CE
0BEh	Steuerbyte Port A
0BFh	Steuerbyte Port B

4. Softwarebeschreibung im 32K-Modus

Nach Ralf Däubner (* Software-Erweiterung gegenüber der Originalversion)

Mit dem EPROMMER-Modul M030 (siehe auch KC-News 4/98) ist es möglich, EPROMs wie sie zum Beispiel in den Modulen M025 oder M040 USER PROM eingesetzt werden, zu brennen. Es können EPROMs von Typ 2716 (2K) bis zum Typ 27256 (32K) gelesen und gebrannt werden. Es ist es nicht möglich, mehrere EPROMs "in einem Rutsch" zu brennen.

Die Software wurde für CAOS 3.1 geschrieben, ist aber auch auf dem KC85/4 lauffähig. [Leichte Anpassungen in der Bedienung sind vorgenommen worden.](#)*

Das Programm EPROMMER ist auf C000h festgelegt, so dass das Modul mit als erstes aktiviert werden muss, also nach Möglichkeit in den am höchsten priorisierten Modulschacht stecken. Die Anfangsadresse des Arbeitsspeichers ist vorzugsweise 400h und die Endadresse je nach verwendeten EPROM:

2716	0400h - 0BFFh
2732	0400h - 13FFh
2764	0400h - 23FFh
27128	0400h - 43FFh
27256	0400h - 83FFh

8K-EPROMs können so auch auf dem KC85/3 ohne Speichererweiterung gebrannt werden. Für 16K-EPROMs empfiehlt sich ein zusätzliches M022 (16K RAM) für die Lücke des RAM4 und für 32K-EPROMs ein M011 (64K RAM).

[Die benötigten RAM-Module werden beim KC85/3 automatisch zugeschaltet.](#) *

Der IRM wird beim Zugriff auf den Speicher abgeschaltet, sodass auch auf RAM-Bereiche zugegriffen werden kann, die sonst vom IRM verdeckt werden. Der Speicherinhalt des ausgelesenen EPROMs im Arbeitsspeicher bleibt auch nach EXIT erhalten.

Somit kann dessen Inhalt mit FSAVE / FLOAD auf ein D004 gespeichert werden, **allerdings nicht der Inhalt des vom IRM verdeckten RAM ab 8000h!**

4.1. Hauptmenü

Zur Aktivierung des EPROMMERs (z.B. Modul in Schacht 8) ist die Zuweisung des Moduls mit

%SWITCH 2 0	; Abschaltung des BASIC-ROM beim KC85/3
%SWITCH 8 1	; Zuschaltung der EPROMMER-Software im M030 (Steckbrücke M030_3 links)
%SWITCH 8 9	; Zuschaltung der EPROMMER-Software im M030 (Steckbrücke M030_3 rechts)

erforderlich. Damit steht im CAOS-Menü das neue Menüwort %EPROMMER, dessen Aufruf nachfolgendes Hauptmenü anzeigt:

```
      ** PROGRAMMING DEVICE **  
EPROM - TYPE:  
1 = 2716  
2 = 2732  
3 = 2764  
4 = 27128  
5 = 27(C)256  
6 = 27(C)256A  
7 = EXIT TO CAOS
```

Ist das Modul M030 zugeschaltet, erscheint im CAOS-Menü das neue Menüwort EPROMMER (ganz oben), dieses wird ohne zusätzlichen Parameter aufgerufen und es erscheint das Hauptmenü der Software.

Als Erstes wird gefragt, welcher EPROM-Typ gebrannt werden soll. Mit den Cursortasten UP/DOWN und ENTER wird der gewünschte EPROM ausgewählt. [Alternativ kann auch die entsprechende Zifferntaste gedrückt werden.](#) *
Der gewählte EPROM-Typ wird dann mit einem Stern vor der Nummer gekennzeichnet.

Grundsätzlich kann jeder Bedienungsschritt mit BREAK (BRK) abgebrochen werden. Dabei gelangt man wieder zurück in das Hauptmenü.

4.2. Brennermenü

Nach Auswahl des EPROM-Typs wird der dem EPROM-Typ betreffende Betriebsmodus eingeschaltet und mit STICK EPROM >> z.B. [27128].
zum Stecken des EPROMs aufgefordert. Im unteren Teil des Bildschirms erscheint ein weiteres Menü:

```
      STICK EPROM  >> 27128  
MENU :  
1 = READ  
2 = COMPARE  
3 = CHECKSUM  
4 = CHECK OF BYTECHANGE  
5 = BLANKTEST  
6 = PROGRAMMING  
7 = DISPLAY  
8 = TAPE  
9 = EXIT TO CAOS
```

Hier stehen nun 9 verschiedene Funktionen zur Auswahl.

1=READ

Lesen des EPROM-Inhalts in den Arbeitsspeicher. Die ursprünglich enthaltene Abfrage des EXPANDER-RAM ist nicht mehr enthalten, da bei Programmstart ein M011 bzw. zwei M022 gesucht und passend geschaltet werden. *

Meldung READ:

ROM: 0000h - 3FFFh (gewünschte Anfangs- und Endadresse)

RAM: 0400h - 43FFh (gewünschte Anfangs- und Endadresse)

Nach dem Beantworten der Fragen wird der EPROM ausgelesen und dessen Inhalt in den Speicher des Grundgerätes geschrieben. Damit kann der Inhalt bequem auf Band gesichert werden bzw. steht der weiteren Bearbeitung zur Verfügung.

Nach Beendigung des Vorgangs wird eine Checksumme ausgegeben.

```
      ** PROGRAMMING DEVICE **  
  
EPROM - TYPE:  
  
 1 = 2716  
 2 = 2732  
 3 = 2764  
 4 = 27128  
 5 = 27(C)256  
 6 = 27(C)256A  
 7 = EXIT TO CA05  
  
      STICK EPROM  >>  
  
READ :  
  
  ROM : 0000 - 3FFF  
  RAM : 0400 - 43FF  
  
  CHECKSUM * FFF0 *  
  
EXIT (Y/N) : ☐
```

Die abschließende Frage EXIT (Y/N) kann mit „N“ beantwortet werden um in der Funktion READ zu bleiben. „Y“ führt dagegen wieder in das Brennermenü.

2=COMPARE

Vergleich EPROM-Inhalt mit Speicherbereich

3=CHECKSUM

Checksumme ermitteln (im RAM oder EPROM)

4=CHECK OF BYTECHANGE

Überbrenntest

5=BLANKTEST

Test, ob EPROM leer ist

6=PROGRAMMING

Programmierung: Es sind die Adressen des zu brennenden RAM- und ROM-Bereiches anzugeben. Als nächstes wird die zu verwendende Programmierspannung abgefragt.

```

** PROGRAMMING DEVICE **

EPROM - TYPE:

  1 = 2716
  2 = 2732
  3 = 2764
*4 = 27128
  5 = 27(C)256
  6 = 27(C)256A
  7 = EXIT TO CMOS

Vpp POWER SUPPLY
  1 = 12,5V
  2 = 21V
  3 = 25V
  4 = EXIT

27128

PROGRAMMING:

  RAM : 0400 - 43FF
  ROM : 0000 - 3FFF
```

Die gewählte Programmierspannung ist zur Sicherheit 2x mit Enter zu bestätigen. Der Programmieralgorithmus brennt jede Speicherzelle mit bis zu 50x 1ms-Impulsen bis die Daten gültig sind. Falls sich damit kein Erfolg einstellt, folgen noch bis zu 5x 50ms-Impulse. Das wiederholt sich für alle zu programmierenden Speicherzellen. Zum Schluss folgt noch ein 2ms-Programmierimpuls pro Speicherzelle und danach ein Vergleich mit dem zu programmierenden RAM-Inhalt. Nach erfolgreichem Brennvorgang wird zur Kontrolle die Checksumme angezeigt.

7=DISPLAY (RAM/ROM)

Mit DISPLAY wurde eine der CAOS-Funktion sehr ähnliche HEX-ASCII-Anzeige realisiert. Zunächst wird abgefragt, ob auf den RAM oder den ROM zugegriffen werden soll. Danach ist die Adresse einzugeben. Im oberen Bildschirmfenster erfolgt dann die Anzeige.

Bei der Anzeige eines RAM-Bereiches kann mit der STOP-Taste auch in einen MODIFY-Modus gewechselt werden.

```

** PROGRAMMING DEVICE **
0400  FF FF FF FF FF FF FF FF FF  ****
0408  FF FF FF FF FF FF FF FF FF  ****
0410  FF FF FF FF FF FF FF FF FF  ****
0418  FF FF FF FF FF FF FF FF FF  ****
0420  FF FF FF FF FF FF FF FF FF  ****
0428  FF FF FF FF FF FF FF FF FF  ****
0430  FF FF FF FF FF FF FF FF FF  ****
0438  FF FF FF FF FF FF FF FF FF  ****
0440  FF FF FF FF FF FF FF FF FF  ****
0448  FF FF FF FF FF FF FF FF FF  ****
0450  FF FF FF FF FF FF FF FF FF  ****

BACKGROUND DISPLAY OF IRM

DISPLAY:      RAM/ROM:RAM

FROM ADR: 0400
```

Bei dieser DISPLAY/MODIFY-Funktion wird der IRM abgeschaltet, sodass man auf den RAM zugreifen kann, der hinter dem IRM liegt.

8=TAPE

Es wird abgefragt, ob die SAVE- oder LOAD-Funktion ausgeführt werden soll. Danach sind Adressen und Dateiname anzugeben. Auch hier wird der IRM für die Zugriffe abgeschaltet, um auf Daten im RAM8-Bereich zugreifen zu können.

9=EXIT

Mit EXIT gelangt man wieder in das CAOS-Menü zurück.

4.3. Inbetriebnahmehilfe

Die Originalsoftware wurde um das CAOS-Menüwort %IBN:M030 erweitert, welches eine Teilmenge der Inbetriebnahmehilfe der 64K-Software enthält. Erläuterungen siehe Seite 47.

5. Softwarebeschreibung im 64K-Modus

Von Mario Leubner (nach Henrik Haftmann)

Basierend auf der existierenden Software EBRZ 1.3 bzw. 1.4 von Henrik Haftmann wurde eine neue 64K-Brennsoftware für das M030 entwickelt. Das Programm EBRZ setzte jedoch eine völlig andere Hardware voraus, sodass es komplett umgeschrieben und an die Hardware des M030 angepasst werden musste. Beibehalten wurde aber das grundlegende Bedienkonzept, welches Wert auf reine Funktionalität legt!

Die für das Modul M030 angepasste und erweiterte Software hat folgende Eigenschaften (Version 1.5/1.6):

- Das Programm ist weniger als 8 KByte groß und befindet sich im ROM-Bereich des Moduls. Es wird beim ersten Aufruf in den RAM ab 200h kopiert und der ROM-Bereich des Moduls anschließend abgeschaltet.
- Arbeitszellen liegen im Adressbereich des Programms ab 200h.
- Es ist lauffähig ab CAOS 3.1, und passt sich selbstständig an die jeweiligen Bedingungen der CAOS-Versionen an.
- Einfachste Bedienung, da bei Aufruf der Menüworte ohne Parameter sinnvolle Standardwerte benutzt werden.
- Integrierte HILFE-Funktion.
- 32K-Programmierpuffer von Adresse 4000h bis BFFFh. Bei einem KC85/3 werden dafür ein M011 (64K RAM) oder zwei M022 (16K RAM) gesucht und automatisch passend eingeschaltet.
- Unter CAOS 3.4 bzw. ab CAOS 4.3 ist der Menüprompter ">", sonst "%".
- Es können die EPROM-Typen 27(C)16 bis 27(C)512 gebrannt werden, ab Version 1.6: 1K-EPROMs 2708/U555C über zwei wählbare Adapter.
- Es können, sofern möglich, Einzelbits nachgebrannt werden.
- EPROMs werden während des Brennens laufend geprüft, ein Defekt wird sofort ausgewiesen.
- Nur ein Menüwort verlängert das CAOS-Hauptmenü, es startet ein eigenes Untermenü□
- Eine Statusanzeige informiert ständig über die eingestellten Parameter wie EPROM-Typ, Programmierspannung Programmiermethode und Länge der Programmierimpulse.
- FastView-Routinen geben einen schnellen Überblick über den Inhalt von EPROMs und Speicher.
- Es können auch mehrere EPROMs in einem Zug gelesen, jedoch nicht gebrannt werden. (z.B. Prüfsumme und CRC über 4 Stück 2k-EPROMs gemeinsam.)
- 2 verschiedene Brenn-Modi: Einzelbyte-Modus (A) und blockweiser Modus (B).
- ab Version 1.6: Anzahl und Länge der Programmierimpulse einstellbar.

Um die 64K-Software des M030 zu nutzen, ist der Umschalter ESW in die entsprechende Stellung zu schalten und danach das Modul mit dem Kommando

%SWITCH <Steckplatz> 9

einzuschalten. Bei CAOS 3.1 ist außerdem der BASIC-ROM abzuschalten mit

%SWITCH 2 0

Danach steht im CAOS-Menü ein neues Menüwort: %EPROMMER. Um in das EPROMMER-Menü zu wechseln, ist folgendes Kommando einzugeben:

%EPROMMER [<EPROM-Typ in KByte> [Methode <A/B>] [Vpp <12/21/25>]]

Die Parameter können dabei auch weggelassen werden. Ohne Parameter wird beim ersten Start der Software ein 8K-EPROM 2764 mit byteweiser Programmierung und Vpp=12,5V vorgewählt. Als Erstes kopiert sich die Software in den RAM0 ab Adresse 200h und schaltet den EPROM-Bereich des Moduls ab (Voraussetzung: Steckbrücke M030_3 Steckbrücke rechts). Jeder weitere Aufruf des nun im RAM0 stehenden Menüwortes stellt einen Warmstart der Software dar, vorher eingestellte Parameter wie EPROM-Typ, Brennmethode, Programmierspannung und Anzahl/Dauer der Programmierimpulse bleiben dabei erhalten. Nach dem Start des Programms erfolgt der Wechsel in das Untermenü der Brennsoftware.

Für 1K-EPROMs mit Programmieradapter gilt folgende Kommandozeile:

%EPROMMER 1 [Adapter <A/B>]

Hier wird durch den Parameter A = Adapter (ML) oder B = Adapter (WH) der verwendete Programmieradapter ausgewählt. Die Programmiermethode und Programmierspannung lässt sich bei 1K-EPROMs nicht ändern.

Für einen Kaltstart der Software ist wieder der ROM zuzuschalten und das Menüwort %EPROMMER aus dem ROM aufzurufen. Das Menüwort im ROM-Speicherbereich C000h-DFFFh wird bei der Suche von CAOS als Erstes gefunden und gestartet.

5.1. Beschreibung der speicherbezogenen Menüworte

Die Menüworte des Brenner-Untermenüs stehen in zwei Gruppen mit einer Leerzeile getrennt. Der obere Teil ist für die speicherbezogenen Kommandos. Die im Menü nach der leeren Zwischenzeile folgenden Menüworte beziehen sich auf den EPROM.

```
EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

* M030 EPROM-Brenner 1.6 *

>HELP
>MENU
>SWITCH
>LOAD
>SAVE
>FLOAD
>FSAVE
>MOVE
>FILL
>MVIEW
>DISP
>MCRC
>QUIT
>
>EPROM
>TIME
>EVIEW
>EDISP
>ERAT
>READ
>PROG
>COMP
>ECRC
>_
```

In der Menüanzeige ist auch die Versionsnummer der Software (1.6) abzulesen.

Im Folgenden heißt <EPROM-Bereich>, dass an dieser Stelle 2 Hexzahlen anzugeben sind, nämlich <Anfang> und <ENDE+1>. Wird <EPROM-Bereich> weggelassen, wird der ganze EPROM, bei den speicherbezogenen Routinen jedoch höchstens die ersten 32 KByte bei 64K-EPROMs angenommen.

<Speicherreferenz> ist die Adresse, ab der gelesen, verglichen bzw. programmiert wird. Ohne die Angabe der <Speicherreferenz> wird 4000h angenommen. Somit kann man EPROMs bis 32 KByte einfach durch >READ und >PROG kopieren, ohne zusätzliche Hexzahleingabe. Bei 64K-EPROMs kann dies durch zwei Teilschritte erfolgen, nämlich

ausführlich:

```
>READ 0 8000 4000
>PROG 0 8000 4000
>READ 8000 0 4000
>PROG 8000 0 4000
```

oder in Kurzform:

```
>READ
>PROG
>READ 8000 0
>PROG 8000 0
```

Zu den einzelnen Menüworten in der Reihenfolge wie sie angezeigt werden:

>HELP

Kurze Hilfestellung zu den Menüworten und insbesondere zu deren Parametern.

```
EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

Kommandos:      (26.03.2016 / CAOS 4.5)

>EPROM Typ (in KByte) (Methode (A/B))
  (Programmierspannung (12/21/25))
  (Bei 1K: A=Adapter ML / B=Adapter WH)

>TIME n1 t1 n2 t2 (Anz./0,1ms-Schritte)

>MOVE Quelle Ziel Länge

>FILL Bereich Byte Byte ..

>MVIEW, EVIEW:
  Anfangsadresse/1.Hexziffer

>PROG, READ, COMP, ERAT:
  Eprom-Bereich Speicherreferenz

>MCRC, ECRC, SAVE: Bereich

>DISPLAY, EDISP: Adresse Zeilen Bytes

>LOAD: Speicherreferenz

(Bereich = 2 Zahlen: Anfang und Ende+1)
(Standardwert Speicherreferenz = 4000H)

>_
```

In dieser Anzeige kann auch das Versionsdatum (hier 25.03.2016) und die erkannte CAOS-Version (hier CAOS 4.5) abgelesen werden.

>MENU [n]

Anzeige des Brenner-Untermenüs selbst.

Ab CAOS 4.5 kann mit Hilfe des Parameters n auch die Anzeige um die versteckten Menüworte mit n=1 und/oder die Startadressen der Menüworte mit n=3 bzw. 2 ergänzt werden.

>SWITCH [Steckplatz [Steuerbyte]]

Parameter wie im CAOS-Menü. Ohne Parameter werden die Schaltzustände des internen Speichers und aller Module aufgelistet. Die Anzeige unterscheidet sich in Abhängigkeit der CAOS-Version. Unter CAOS 3.1 wird neben den Modulen nur der Schaltzustand des BASIC-ROM mit angezeigt:

```

EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

%SWITCH
02 FF 00
08 F4 C3
0C F6 43
10 7B 00
FC A7 00
%

```

Entsprechend mehr Informationen erhält man am KC85/4. Hier kommt der Schaltzustand des RAM-4 und RAM-8 hinzu, ab CAOS 4.3 auch der schaltbare CAOS-ROM-C und ab CAOS 4.6 die vollständige Modulübersicht der internen und gesteckten Module einschließlich der Modulbezeichnung:

```

EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

>SW
-- -- CAOSE ON
00 FF 03 RAM0 ON
01 FF 01 BILD 0 ON
02 FF 00 USER 0 OFF
03 FF 03 RAM8 0 ON
04 FF 03 RAM4 0 ON
05 FF 00 CAOSC OFF
-- -- JOY/CEN
0C F4 C3 16K RAM
10 7B 00 1M RAM
11 7B 00 1M RAM
12 7B 00 1M RAM
13 7B 00 1M RAM
14 FD 00 USB+NET
FC A7 04 FLOPPY
>-

```

>LOAD [Speicherreferenz]

Einlesen einer Datei in den Programmierpuffer, auch hinter den IRM. Ohne Angabe der Adresse wird unabhängig von der Adressinformation im Vorblock der Datei der Inhalt immer ab Adresse 4000h geladen. Mit der expliziten Angabe einer Ladeadresse wird diese benutzt. Im Gegensatz zur sonst üblichen Praxis unter CAOS ist der Parameter <Speicherreferenz> die direkte Ladeadresse und kein Offset! Man muss sich also nicht darum kümmern, ab welcher Adresse der Inhalt der Datei ursprünglich abgespeichert wurde. Es wird immer in den Programmierpuffer eingelesen.

Eine Endadresse muss nicht angegeben werden. Eingelesen wird entweder soviel, wie im Vorblock der Datei erkannt wurde, maximal aber 32 KByte wegen der Größe des Programmierpuffers von 4000h bis BFFFh. Wird kein gültiger Vorblock erkannt, dann wird soviel eingelesen wie für die eingestellte EPROM-Größe erforderlich ist. Somit können auch Binärdateien ohne CAOS-Vorblock geladen werden. Abbruch erfolgt in diesem Fall aber auch beim Einlesen der Blocknummer FF (Endeblock).

Das Einlesen kann am Bildschirm verfolgt werden. Bei korrekt eingelesenen Datenblöcken werden die Blocknummern mit einem nachfolgendem > angezeigt. Bei Lesefehlern wird zuerst die erwartete Blocknummer, nach einem Doppelpunkt dann die gelesene Blocknummer und ein Fragezeichen dargestellt. Ein Stern statt des Fragezeichens bedeutet eine nicht erwartete Blocknummer. In beiden Fällen kann die Kassette zurückgespult und der erwartete Block neu eingelesen werden.

BRK bricht das Einlesen ab.

Unter CAOS ab Version 4.6 greift LOAD auf das gerade eingestellte DEVICE zu, also auch auf Diskette oder USB. Deshalb wird bei LOAD auch der Dateiname abgefragt.

```
>SAVE C000 E000
Name : M030
40
>LOAD
Name : M030
M030 4000 6000
FF>
> _
```

>SAVE [Anfangsadresse [Endadresse+1]]

Abspeichern des Inhaltes des Programmierpuffers oder eines beliebigen anderen Speicherbereiches. Es kann wie unter CAOS üblich, die Anfangsadresse und die

Endadresse + 1 angegeben werden. Werden keine Adressen angegeben, dann wird ab Anfang des Programmierpuffers, also Adresse 4000h bis zum Ende entsprechend der EPROM-Größe abgespeichert, maximal jedoch 32K. Wird nur ein Parameter angegeben, dann ab der angegebenen Adresse in der Größe des eingestellten EPROM-Typs. Die Adressen des Speicherbereiches werden dann zur Kontrolle mit angezeigt.

```
EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

>SAVE 8000
8000 A000
Name : _
```

Ab CAOS 4.6 greift SAVE auf das gerade eingestellte DEVICE zu.

Bei 64K-EPROMs muss SAVE und LOAD in zwei Teilen durchgeführt werden, da nur 32K Programmierpuffer zur Verfügung stehen.

>FLOAD [Speicherreferenz]

>FSAVE [Anfangsadresse [Endadresse+1]]

Diese beiden Menüworte sind für das Laden bzw. Abspeichern auf Diskette vorgesehen. Sie sind nur aktiv, wenn sich im KC-System ein D004 in der CAOS-Betriebsart befindet, im D004 also das Programm DEP.COM läuft.

Ab CAOS 4.6 sind die beiden Menüworte nicht sichtbar aber trotzdem nutzbar, LOAD und SAVE mit DEVICE=1 ist gleichbedeutend verwendbar.

Die Funktion von FLOAD/FSAVE entspricht den Kommandos LOAD/SAVE.

>MOVE <Anfang Quelle> <Anfang Ziel> <Länge Speicherbereich>

Verschieben von Speicherinhalten im RAM.

Fehlermeldungen	Bedeutung
MOVE Quelle Ziel Länge	Es müssen immer alle drei Parameter angegeben werden, ansonsten erscheint diese kurze Hilfe.
Speicherschutz!	Kommando wird nicht ausgeführt, da der RAM0 von Adresse 0 bis 2000h überschrieben würde.

>FILL [<Anfangsadresse> <Endadresse+1>] [<Byte> [<Byte>]...]

Füllen des angegebenen Speicherbereiches mit einem Byte bzw. einer Folge von maximal 8 Bytes, wenn mehrere Bytes angegeben worden. Wird kein Füllbyte angegeben, so wird der Speicherbereich mit FFh (entspricht leerem EPROM) gefüllt. Ohne Angabe des Speicherbereiches wird der Bereich von 4000h bis zur eingestellten EPROM-Größe, maximal jedoch 32K gefüllt.

Beispiele:

>FILL	Füllen des aktuellen Programmierpuffers mit FFh
>FILL 0	Füllen des aktuellen Programmierpuffers mit 00h
>FILL 4000 8000	Füllen des Bereiches von 4000h bis 7FFFh mit FFh
>FILL 4000 6000 AA 55	Füllen des Bereiches von 4000h bis 5FFFh abwechselnd mit AAh und 55h

Fehlermeldungen

Bedeutung

Speicherschutz!

Kommando wird nicht ausgeführt, da der RAM0 von Adresse 0 bis 2000h überschrieben würde.

>MVIEW [<Anfangsadresse>]

Schnelles Betrachten eines Speicherbereichs bis zu 8 KByte pixelweise auf dem Bildschirm. Es erfolgt eine typgerechte Ausgabe, damit die Anzeigen auf KC85/3 und /4 gleich aussehen. Das heißt, das erste Byte wird links oben dargestellt, dann pixelweise untereinander und nach 256 Byte weiter in der nächsten Spalte. Ein komplett weißes Anzeigefeld bedeutet alle Byte sind FFh.

Die Anfangsadresse kann auch 1stellig oder 2stellig erfolgen, das erspart das Eintippen der sonst stets üblichen 2 oder 3 Nullen bei den meisten Adressen. Anzeigebeginn ist immer der Beginn der vollen Page, also auf xx00h abgerundet.

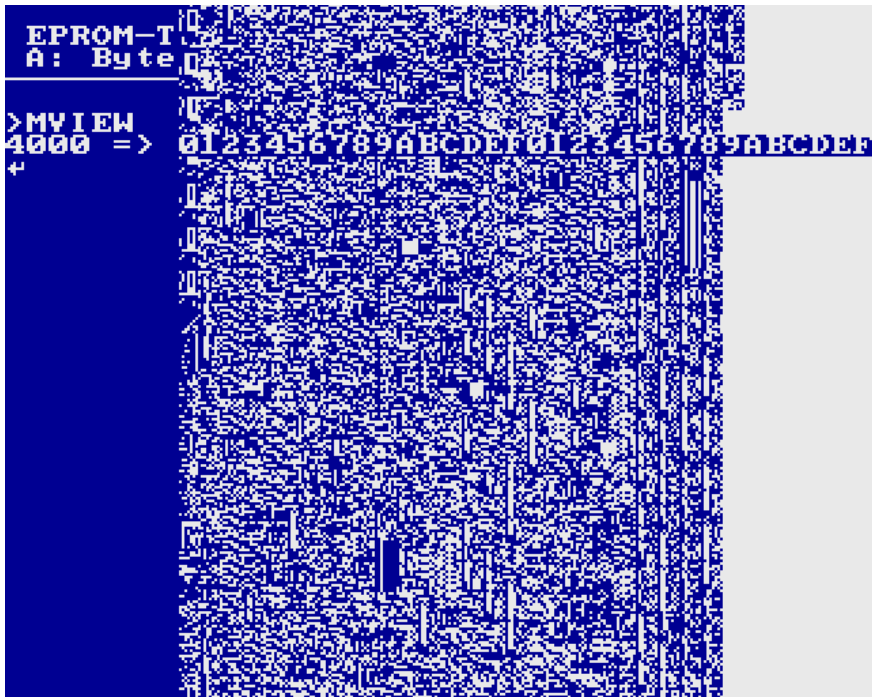
Ganz ohne Parameter wird Adresse 4000h angenommen.

Beispiele:

>MVIEW	Anzeige ab Adresse 4000h
>MVIEW 8000	Anzeige ab Adresse 8000h
>MVIEW 8	Anzeige ab Adresse 8000h
>MVIEW 56	Anzeige ab Adresse 5600h
>MVIEW 5637	Anzeige ab Adresse 5600h

Angezeigt wird soviel wie dem eingestellten EPROM-Typ entspricht. Bei eingestelltem 8K-EPROM wird der gesamte Inhalt auf einmal angezeigt. Bei 1K-, 2K- und 4K-EPROMs wird also nur ein Teil des maximal darstellbaren Speichers angezeigt, bei 16K-, 32K- und 64K-EPROMs wird in 8K-Schritten der Inhalt nach

und nach dargestellt. Mit Enter wird weiter geblättert, BRK bricht die VIEW-Funktion ab.



Zur Übersicht wird unter der Eingabezeile am Zeilenanfang die gerade dargestellte Startadresse und danach ein Lineal mit dem zweiten Halbbyte der Adresse eingeblendet, um die tatsächlichen Adressen besser abschätzen zu können.

Die überschriebene Titelzeile wird nach Abschluss von VIEW wieder hergestellt. Vorher wartet das Programm allerdings noch eine Tasteneingabe ab, um die komplette Ansicht in Ruhe betrachten zu können.

>DISP [Adresse [Zeilen [Bytes]]]

Aufruf einer CAOS-ähnlichen DISPLAY-Routine. Ohne Parameter wird standardmäßig ab Speicheradresse 4000h mit 8 Bytes pro Zeile und jeweils 4 Zeilen pro Seite angezeigt. Es kann aber auch eine andere Adresse, sowie mit einem zweiten Parameter eine abweichende Anzahl an Zeilen und mit einem dritten Parameter eine abweichende Anzahl von Bytes pro Zeile angegeben werden.

```

EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

```

```

>DISP
4000 18 0B C3 8C C0 7F 7F 42  -+C1@[]B
4008 41 53 49 43 00 21 BD C0  ASIC_!=@
4010 11 00 03 01 67 00 ED B0  7_[]ug_m0
4018 EB F9 CD 69 C6 32 AB 03  kyMifZ+[]
_

```

Im Gegensatz zu CAOS wird ab Adresse 8000h nicht der IRM, sondern der dahinterliegende RAM dargestellt.

Die Taste STOP geht zum MODIFY-Modus über, auch hier für den RAM hinter dem IRM falls die Adresse im Bereich von 8000h bis BFFFh liegt.

>MCRC [<Anfang Speicher> [<Ende Speicher+1>]

Berechnen eines Prüfpolynoms (CRC) und einer Prüfsumme über den Speicherbereich des Programmierpuffers. Wird nur die Anfangsadresse angegeben, dann wird ein Speicherbereich ab dieser Adresse in der Größe des aktuell eingestellten EPROM-Typs (maximal jedoch 32K) für die Berechnung benutzt. Werden beide Argumente weggelassen, wird der Speicherbereich ab Adresse 4000h angenommen. Während der Berechnung läuft die Speicheradresse zur Kontrolle am Bildschirm mit.

```

>MCRC
CRC16=0DE4 Prüfsumme=3AB2
>_

```

>QUIT

Verlassen des Brennprogramms und Rückkehr ins CAOS-Menü.

Die während der Arbeit mit der EPROMMER-Software im RAM gespeicherten EPROM-Inhalte stehen nach QUIT weiterhin zur Verfügung und können anderweitig verwendet werden. Hierbei ist zu beachten, dass der RAM8 vom IRM verdeckt wird. Erst ab CAOS 4.3 lässt sich der IRM mit SWITCH 1 0 ausblenden.

5.2. Beschreibung der EPROM-bezogenen Menüworte

Die nachfolgenden Kommandos setzen einen EPROM auf dem Programmiersockel und korrekt eingestellte Programmierparameter (EPROM-Typ, Programmierspannung, Methode und Impulsdauer laut Datenblatt) voraus. Das Stecken und Ziehen des EPROMs sollte vor Ausführung der jeweiligen Kommandos erfolgen. Eine zusätzliche Steckaufforderung ist in der Software nicht enthalten.

>EPROM [<EPROM-Typ in KByte> [Methode <A/B>] [Vpp <12/21/25>]]
>EPROM 1 [Adapter <A/B>]

Auswahl des EPROM-Typs sowie Vorwahl der Programmierspannung und -methode bzw. bei 1K-EPROMs Auswahl des Adapters. Die Parameter entsprechen genau denen des CAOS-Menüwortes %EPROMMER und können auch hier weggelassen werden. Ganz ohne Parameter wird hier nur die Anzeige der Statuszeile aktualisiert:

```
EPROM-Typ: 64 KByte (27512) Vpp=12,5V
A: Byte-Mode 5*1,0ms (1*2,0ms)

>EPROM 64
>_
```

Wird nur der EPROM-Typ angegeben, dann wird jeweils die letzte für den jeweiligen EPROM-Typ angegebene Einstellung aktiviert. Geladen werden auch die eingestellten Parameter für die Anzahl und Dauer der Programmierimpulse. Diese Einstellungen können bei Bedarf durch das Kommando TIME verändert werden.

Voreingestellt sind:

Speicher- kapazität	EPROM-Typ	Programmier- spannung	Methode	Pro- grammier- impulse	Sicher- heits- impulse
1K	2708 (U555C)	25V	B	100*1ms	15*1ms
2K	27(C)16	25 V	A	50*1ms	2*2ms
4K	27(C)32	21 V	A	45*1ms	2*2ms
8K	27(C)64	12,5 V	A	25*1ms	1*2ms
16K	27(C)128	12,5 V	A	5*1ms	1*2ms
32K	27(C)256	12,5 V	A	5*1ms	1*2ms
64K	27(C)512	12,5 V	A	5*1ms	1*2ms

Für 1K-EPROMs ist fest die Methode B (blockweise) und $V_{pp}=25$ vorgegeben und kann auch nicht geändert werden. Alle anderen EPROM-Typen können wahlweise mit einer der beiden Methoden programmiert werden.

Werden für EPROMs > 1KByte abweichende Werte für die Programmiervspannung und Methode gewünscht, so kann dies durch Angabe eines 2. und 3. Parameters erfolgen. Die Reihenfolge dieser Parameter ist dabei gleichgültig. Es kann also auch nur der eine Parameter angegeben werden, der von der Vorgabe abweicht.

Beispiele:

>EPROM 2 21 2K-EPROM 2716 mit $V_{pp}=21V$
 >EPROM 16 B 16K-EPROM 27128 mit $V_{pp}=12,5V$ und blockweiser
 Programmierung

>TIME n1 t1 n2 t2

Mit dem Kommando TIME werden die Anzahl und die Dauer der Programmierimpulse angepasst. Die voreingestellten Parameter sind erprobt und eignen sich für eine Vielzahl von EPROMs. Schaut man dagegen in die Datenblätter der verschiedenen EPROM-Typen unterschiedlicher Hersteller, dann werden oftmals auch andere Parameter zur Programmierung vorgegeben. Zum Beispiel reichen bei modernen EPROMs oftmals Programmierimpulse mit $100\mu s$ oder $200\mu s$ Länge aus. Dann könnten es aber auch nur maximal 10 Programmierimpulse sein. Um diese verschiedenen Vorgaben genauer einzuhalten kann hier eine Anpassung vorgenommen werden. Das Kommando TIME erfordert immer die Angabe aller 4 Parameter, und zwar in der Reihenfolge wie sie auch in der Titelseite stehen. Im Einzelnen bedeuten die 4 Parameter:

- n1 maximale Anzahl der Programmierimpulse.
Möglich sind Werte zwischen 1 und 255.
- t1 Länge der Programmierimpulse in 0,1ms-Schritten.
Möglich sind Werte zwischen 1 (entspr. $100\mu s$) und 30 (entspr. 3,0ms)
- n2 minimale Anzahl der Sicherheitsimpulse.
Möglich sind Werte zwischen 1 und 255.
- t2 Länge der Sicherheitsimpulse in 0,1ms-Schritten.
Möglich sind Werte zwischen 1 (entspr. $100\mu s$) und 30 (entspr. 3,0ms)

Fehlermeldungen

Bedeutung

Parameterfehler!

Parameter liegen außerhalb der Eingabegrenzen

TIME n1 t1 n2 t2

Es wurden nicht alle 4 Parameter angegeben, dazu der Hinweis welche Parameter benötigt werden.

>EVIEW [<Anfangsadresse>]

Schnelles Betrachten eines EPROM-Bereichs bis zu 8 KByte pixelweise auf dem Bildschirm. Ohne Parameter wird die EPROM-Adresse 0000h als Anfang angenommen. Angezeigt wird von der Anfangsadresse bis zum EPROM-Ende. Nach 8 KByte ist der Bildschirm gefüllt. Sind weitere Daten größerer EPROMs anzuzeigen, dann kann nach der Tastaturabfrage mit Enter weitergeblättert werden. BRK bricht die VIEW-Funktion ab.

Siehe dazu auch die Ausführungen zum Kommando MVIEW.

>EDISP [<Adresse> [Zeilen [Bytes]]]

Aufruf einer CAOS-ähnlichen DISPLAY-Routine direkt für die Ansicht des EPROM-Inhalts (ohne diesen vorher in den RAM einlesen zu müssen). Ohne Parameter wird standardmäßig ab EPROM-Adresse 0 angezeigt. Ein MODIFY ist bei EDISP nicht möglich, die STOP-Taste hier also unwirksam. Nach Anzeige des letzten Bytes vom EPROM wird die Anzeige automatisch abgebrochen.

```

EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

>EDISP 1FE3
1FE3 FF FF FF FF FF FF FF FF
1FEB FF FF FF FF FF FF FF FF
1FF3 FF FF FF FF FF FF FF FF
1FFB FF FF FF FF FF FF
>_

```


>ERAT [<EPROM-Bereich>] [<Speicherreferenz>]

ERAT ist die Abkürzung für "EraseTest" und ist ein bitweiser Löschtest (siehe auch >PROG). Es wird getestet, ob der (nicht gelöschte) EPROM sich für das Nachbrennen einzelner Bits eignet.

Wird das 3. Argument <Speicherreferenz> weggelassen, erfolgt ein vollständiger Löschtest des EPROMs, das heißt jede Zelle muss FFh enthalten. Ohne Angabe des EPROM-Bereiches erfolgt der Löschtest über den gesamten EPROM. Der Löschtest ist auch über mehrere EPROMs hinweg möglich!

Beispiele:

>ERAT Löschtest des gesamten EPROMs

>ERAT 2000 Test, ob sich der RAM-Inhalt ab 2000h in den EPROM brennen lässt

- >ERAT 0 800 Löschtest des EPROM-Bereiches 0 bis 800h
- >ERAT 0 400 4400 Test, ob sich der RAM-Inhalt ab 4400 in den EPROM von 0 bis 400h brennen lässt

Meldungen	Bedeutung
EPROM gelöscht	Der EPROM-Bereich ist leer, alle Bytes = FFh
EPROM brennbar	Der im Programmierpuffer stehende Inhalt kann in den EPROM gebrannt werden. Der EPROM muss dazu nicht leer sein, aber alle Bytes sind überbrennbar.
Unbrennbar: xyz EPROM löschen!	Für jede nicht brennbare Speicherzelle wird angezeigt: EPROM-Adresse, RAM-Adresse, EPROM-Inhalt, RAM-Inhalt
n Byte(s) falsch. EPROM löschen!	Falls nicht brennbare / nicht leere Speicherzeiten vorhanden sind, erscheint diese Meldung mit der Byte-Anzahl falls nicht vorher mit BRK abgebrochen wurde.

>READ [<EPROM-Bereich>] [<Speicherreferenz>]

Einlesen (auch mehrerer EPROMs) vom EPROM-Programmiersockel in den Speicher. Der Verlauf kann anhand der mitlaufenden Adresse verfolgt werden. Ohne Parameter wird der gesamte EPROM, jedoch maximal 32K ab 4000h eingelesen.

Beispiele:

- >READ Einlesen des gesamten EPROMs nach 4000h (bei 64K-EPROM die ersten 32K)
- >READ 2000 Einlesen des gesamten EPROMs nach 2000h (bei 64K-EPROM die ersten 32K)
- >READ 0 800 Einlesen der ersten 2 KByte des gesteckten (auch größeren) EPROMs nach 4000h
- >READ 0 2000 Einlesen von 8 KByte nach 4000h. Ist der eingestellte EPROM-Typ kleiner als 8K wird zum EPROM-Wechsel aufgefordert bis insgesamt 8K eingelesen sind.
- >READ 100 200 2000 Einlesen des EPROM-Inhaltes Adresse 100h bis 1FFh in den RAM ab Adresse 2000h

Fehlermeldungen**Bedeutung****Speicherschutz!**

Kommando wird nicht ausgeführt, da der RAM0 von Adresse 0 bis 2000h überschrieben würde.

>PROG [<EPROM-Bereich>] [<Speicherreferenz>]

Aufruf der Brennroutine mit vorherigem bitweisen Löschtest. Der EPROM muss nicht leer, d.h. mit FF gefüllt sein, Hauptsache dort, wo 1-Bits sein sollen, sind keine 0-Bits im EPROM. Ohne angegebene Parameter wird der komplette EPROM (bei 64K-EPROMs die ersten 32K) mit dem Inhalt des Standard-Programmierpuffers ab 4000h programmiert. Durch Angabe des EPROM-Bereichs kann auch nur ein EPROM-Teilbereich neu programmiert werden. Als Speicherreferenz kann auch ein im KC befindlicher ROM verwendet werden, wenn dieser vorher mit SWITCH „sichtbar“ geschaltet wurde.

Die Software enthält zwei Brennmethode, die aktuell ausgewählte ist in der Titelzeile zu sehen. Die Methode "B" ist langsamer als "A", teilweise aber sicherer. Wenn kein Datenblatt vom zu programmierenden EPROM zur Verfügung steht, sollte man ausprobieren, welche Methode besser funktioniert. Das gleiche gilt für die Programmierspannung: Hier bei 12,5V beginnen und falls nicht erfolgreich gebrannt mit 21V bzw. 25V den Versuch wiederholen.

Eine Übersicht über erfolgreich programmierte EPROM-Typen und deren Parameter befindet sich am Ende dieses Handbuchs.

Methode A:

Hier wird jedes Byte mit maximal **n1** Programmierimpulsen der Dauer **t1** gebrannt und nach jedem Impuls mit noch anliegender Programmierspannung kontrolliert, ob das Byte bereits den Sollzustand angenommen hat. Ist das der Fall, dann schließen sich noch halb so viele Sicherheitsimpulse der Dauer **t2** an, mindestens jedoch so viele wie in **n2** festgelegt ist. Dieser Vorgang wird für alle zu programmierenden Bytes durchlaufen, Zellen mit Inhalt FFh werden nicht mit programmiert.

Zum Schluss wird bei abgeschalteter Programmierspannung eine weitere Kontrolle durchgeführt.

Während des Programmiervorgangs wird die aktuelle Adresse angezeigt, um dem Verlauf der Prozedur besser zu folgen.

Methode B:

Hier wird für alle zu programmierenden Bytes nacheinander ein Programmierimpuls der Dauer **t1** ausgelöst. Danach wird die Programmierspannung abgeschaltet und der gesamte Bereich kontrolliert. Solange noch mindestens ein Bit nicht den Sollzustand erreicht hat, werden weitere, maximal **n1** Programmierzyklen über den gesamten Bereich durchlaufen. Danach folgen noch halb so viele Sicherheitszyklen der Dauer **t2**, mindestens jedoch so viele wie in **n2** festgelegt ist. Zellen mit Inhalt FFh werden auch hier nicht mit programmiert.

Zum Schluss wird bei abgeschalteter Programmierspannung eine weitere Kontrolle (Schlussprüfung) durchgeführt.

Während des Programmiervorgangs wird vor der Adresse die Nummer des aktuellen Durchlaufs angegeben. Solange diese Zahl **rot** erscheint, laufen die Programmierimpulse. Die Sicherheitszyklen werden in **grüner** Schriftfarbe dargestellt, um das baldige Ende anzuzeigen.

Beim Programmieren von 1K-EPROMs 2708 mittels Adapter muss jede Speicherzelle mit einem 1ms-Impuls gebrannt werden, bevor der nächste Zyklus beginnen kann. Um auch bei 1K-EPROMs Teilbereiche programmieren zu können, wartet die Software in diesem Fall für $1\text{ms} \times \text{Anzahl der nicht programmierten Bytes}$ bevor der nächste Programmierzyklus gestartet wird.

Übrigens: Wer dem intelligenten Modus für den 2708 nicht traut, der gibt einfach mit dem TIME-Kommando ausreichend Programmier- und Sicherheitszyklen vor, sodass insgesamt 100 Zyklen zu 1ms erreicht werden.

Meldungen vor bzw. während des Programmiervorgangs:

Meldung	Bedeutung
Adressen für EPROM unpassend!	Die Speicherkapazität des eingestellten EPROM-Typs ist kleiner als die im Kommando angegebenen Adressen es erfordern würden.
EPROM nicht brennbar - löschen!	Vor Beginn der Programmierung wird geprüft, ob der EPROM brennbar ist. Der EPROM muss nicht leer sein, jedoch können keine 0-Bits zu 1 werden. Das ist bei dieser Meldung der Fall und der EPROM muss erst noch gelöscht werden.
Teste...	Vor dem Brennen wird getestet, ob der zu brennende Bereich des EPROMs brennbar ist. Während des Brennens im Blockmodus wird der Inhalt kontrolliert und nach dem Brennen im Byte-Modus wird noch eine Schlussprüfung durchgeführt.
Keine Daten zu brennen.	Der EPROM-Inhalt stimmt mit dem Inhalt des Programmierpuffers bereits überein. Es wird kein Brennvorgang durchgeführt.

Meldung	Bedeutung
EPROM-Adresse nnnn Soll=xy Ist=yz	Anzeige der ersten fehlerhaften EPROM-Adresse mit Ist- und Soll-Inhalt. Danach folgt noch eine der beiden nächsten Meldungen.
Zu viele erforderliche Zyklen	Das Datenbyte konnte nicht innerhalb der maximal zulässigen Anzahl Zyklen programmiert werden. Mögliche Ursachen: EPROM defekt, ungeeignet für die Programmiermethode oder Programmierspannung falsch gewählt.
0-Bits können nicht zu 1 werden	Während Programmierung sind 1-Bits zu 0 geworden, die auf 1 bleiben sollten. Meist ist der EPROM defekt.
Schlussprüfung feh- lerhaft	EPROM-Inhalt war während der Programmierung korrekt gelesen worden, jedoch bei der Kontrolle mit abgeschalteter Programmierspannung nicht mehr.
Erfolgreich pro- grammiert!	Inhalt korrekt in EPROM geschrieben und mit Schlussprüfung kontrolliert.

>COMP [<EPROM-Bereich>] [<Speicherreferenz>]

Vergleichen (auch mehrerer) EPROMs mit dem Speicherbereich auf Gleichheit. Während der Überprüfung läuft die EPROM-Adresse auf dem Bildschirm mit. Bei Gleichheit wird dies mit der abschließenden Meldung **Vergleich OK** angezeigt.

Unterschiede werden sofort auf dem Bildschirm angezeigt. Dabei steht zuerst die EPROM-Adresse mit Inhalt und danach die Speicheradresse mit Inhalt.

```

EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms      (1*2,0ms)

>COMP
0025=FF 4025=23
0201=FF 4201=00
0334=FF 4334=77
3 Byte(s) unterschiedlich.
>_

```

Unter der Liste der ungleichen Speicherzellen steht noch die Abschlussmeldung **n Byte(s) unterschiedlich**. Der Vergleich kann aber auch jederzeit mit BRK abgebrochen werden.

>ECRC [<Anfang EPROM> <Ende EPROM+1>]

Berechnen eines Prüfpolynoms (CRC16) und einer Prüfsumme über den EPROM-Bereich. Werden die beiden Argumente weggelassen, wird der gesamte EPROM angenommen. Während der Berechnung läuft die EPROM-Adresse zur Kontrolle am Bildschirm mit.

Die Berechnung kann auch über mehrere EPROMs hinweg vorgenommen werden, um z.B. die gesamte Prüfsumme über die 4 Stück 2716 eines 8K-EPROM-Moduls M025 zu kontrollieren.

Leere (gelöschte) EPROMs haben diese Prüfsummen:

Speicherkapazität	EPROM-Typ	CRC16	Prüfsumme
1K	2708 (U555C)	77EB	FC00
2K	27(C)16	33D7	F800
4K	27(C)32	0FE1	F000
8K	27(C)64	FFFC	E000
16K	27(C)128	FFF0	C000
32K	27(C)256	FF00	8000
64K	27(C)512	0000	0000

5.3. Versteckte Menüworte

Es gibt außerdem noch drei versteckte Menüworte in der M030-Software. Diese müssen bei CAOS 3.1, 4.3 und 4.4 in Kleinbuchstaben eingegeben werden. Bei CAOS 3.4, 4.2, und ab CAOS 4.5 können die Menüworte sowohl in Klein- als auch ganz normal in Großbuchstaben eingegeben werden (sie sind durch ein dem Menüwort angehängtes Leerzeichen nicht sichtbar).

Die Menüworte sind im normalen Menü deshalb nicht sichtbar, weil sie nicht direkt zum Funktionsumfang der EPROM-Software gehören und das Menü sonst zu unübersichtlich erscheinen lassen. In der Beschreibung sollen sie aber der Vollständigkeit halber mit aufgeführt werden:

>modify [Adresse [Bytes]]

Direkter Aufruf der MODIFY-Routine. Ohne Parameter wird standardmäßig bei Speicheradresse 4000h mit 1 Byte pro Zeile gestartet.

Im Gegensatz zu CAOS wird ab Adresse 8000h nicht im IRM, sondern im dahinterliegenden RAM gearbeitet.

>arith <Hexzahl1> <Hexzahl2>

Berechnen der Summe und Differenz zweier Hexzahlen:

```
EPROM-Typ: 64 KByte (27512) Vpp=12,5V
Methode A: Einzelbyte-Programmierung
```

```
>arith 6000 1800
6000 + 1800 = 7800
6000 - 1800 = 4800
>_
```

Fehlermeldungen	Bedeutung
-----------------	-----------

ARITH Zahl1 Zahl2	Beim Kommando ARITH müssen immer zwei Parameter angegeben werden. Falls diese nicht angegeben werden, erscheint dieser kurze Hilfetext.
--------------------------	---

>ibn

Dieses Menüwort unterstützt die Inbetriebnahme des Moduls M030 und die Fehlersuche. Die Parameter sind im folgenden Abschnitt beschrieben.

5.4. Inbetriebnahmehilfe

Von Mario Leubner

Das Menüwort **>IBN** gestattet es, jedes Pin des EPROM-Programmierersockels einzeln mit den möglichen Signalen zu beschalten und statisch die Funktion durchzumessen.

ACHTUNG!

Hierbei sollte kein EPROM in der Fassung stecken, bei Fehlbedienung kann eine Zerstörung des EPROMs erfolgen!

Mit den einzelnen Parametern können die verschiedenen Signale statisch angesteuert und damit an der Fassung nachgemessen werden.

Speziell die Einstellung der drei verschiedenen Programmierspannungen ist hier sehr komfortabel möglich.

Ohne Angabe von Parametern wird eine kurze Hilfeseite für dieses Kommando angezeigt:

```

EPROM-Typ: 8 KByte (2764)      Vpp=12,5V
A: Byte-Mode      25*1,0ms    (1*2,0ms)

* Parameter M030-Diagnoseprogramm *
-----
>IBN Nummer Daten/Schalter
Nr.:
0 Rücksetzen (Diagnoseabschluß)
1 Ucc=5V an Pin 28 (K2) AUS/EIN
2 Ucc=5V an Pin 26,28 (K1+K2) AUS/EIN
3 Pin 20 (/CE) 0=Low/1=High
4 Pin 22 (/OE) 0=Low/1=High/2=Vpp
5 Pin 27 (/PGM/A14) 0=Low/1=High
6 Pin 23 (Upr) 0=A11/1=Vpp
7 Pin 1 0=A15/1=Vpp
8 Adresse anlegen
9 Datenbyte lesen
A Datenbyte anlegen
B Steuersignale anzeigen
C Steuersignale anlegen
10 Vpp=0V
12 Vpp=12,5V
21 Vpp=21,0V
25 Vpp=25,0V
>_

```

Bei aktiviertem 2708-Adapter (ML) ist eine modifizierte Hilfeseite sichtbar, welche die Signale und Pins am Sockel des 2708-Adapters beschreibt. Am 2708 nicht anliegende Signale werden in der Hilfe weggelassen:

```

EPROM-Typ: 1 KByte (2708)      Vpp=25V
              100*1,0ms (15*1,0ms)

* Parameter M030-Diagnoseprogramm *
-----
>IBN Nummer Daten/Schalter
Nr.:
0 Rücksetzen (Diagnoseabschluß!)
1 Ubb=-5V Pin 21 0=AUS/1=EIN
2 Ucc=+5V Pin 24 Udd=12V Pin 19 AUS/EIN
3 Pin 20 (/CS-WE) 0=12V/1=Low
4 Pin 18 (PR) 0=Low/1=High/2=Vpp
8 Adresse anlegen
9 Datenbyte lesen
A Datenbyte anlegen
B Steuersignale anzeigen
C Steuersignale anlegen
10 Vpp=0V
25 Vpp=25,0V

(Pin-Nr. entsprechend 2708-Fassung!)
>_

```

Die Funktion der IBN-Kommandos ist aber immer identisch, egal welcher Modus eingestellt ist,

Als Erstes ist mit >IBN 0 der Grundzustand herzustellen, danach können die weiteren Diagnoseroutinen genutzt werden. Nach Abschluss der Diagnoseroutinen sollte mit >IBN 0 wieder der Grundzustand hergestellt werden, damit die EPROM-Fassung wieder spannungslos wird.

Voraussetzung für die Kommandos IBN 3 bis 25 ist eine zugeschaltete +5V-Spannung mittels IBN 1 1 oder IBN 2 1.

Beispiele:

IBN 0	Grundzustand / Initialisierung – Sockel spannungslos (Ucc und Vpp werden abgeschaltet)
IBN 1 1	Zuschalten der +5V mit Relais K02 an Pin 28
IBN 1 0	Abschalten der +5V mit Relais K02 an Pin 28
IBN 2 1	Zuschalten der +5V mit Relais K01 an Pin 26 (+5V mit Relais K02 an Pin 28 wird mit zugeschaltet)
IBN 2 0	Abschalten der +5V mit Relais K01 an Pin 26 (+5V an Pin 28 bleibt zugeschaltet)
IBN 3 0	Pin 20 /CE auf Low-Pegel
IBN 3 1	Pin 20 /CE auf High-Pegel
IBN 4 0	Pin 22 /OE auf Low-Pegel
IBN 4 1	Pin 22 /OE auf High-Pegel
IBN 4 2	Programmierspannung an Pin 22 /OE anlegen
IBN 5 1	Pin 27 (/PGM bzw. A14) auf High-Pegel
IBN 5 0	Pin 27 (/PGM bzw. A14) auf Low-Pegel
IBN 6 0	Adresse A11 auf Pin 23 schalten
IBN 6 1	Programmierspannung an Pin 23 anlegen
IBN 7 0	Adresse A15 auf Pin 1 schalten
IBN 7 1	Programmierspannung an Pin 1 anlegen
IBN 8 0	Alle 16 Adressleitungen auf Low-Pegel legen
IBN 8 FFFF	Alle 16 Adressleitungen auf High-Pegel legen
IBN 9	Datenleitungen Lesemodus, anliegendes Datenbyte anzeigen
IBN A 0	Datenleitungen Schreibmodus, alle 8 Leitungen auf Low-Pegel
IBN A FF	Datenleitungen Schreibmodus, alle 8 Leitungen auf High-Pegel
IBN B	Zustand der Steuersignale von PIO-Port 0BDh zurücklesen und anzeigen
IBN C nn	Steuerbyte nn als Steuersignal an PIO-Port 0BDh anlegen
IBN 12	Programmierspannung 12,5V einstellen an Messbuchse
IBN 21	Programmierspannung 21V einstellen an Messbuchse
IBN 25	Programmierspannung 25V einstellen an Messbuchse
IBN 10	Programmierspannung 0V (ohne die +5V abzuschalten)
IBN 0	Rücksetzen, Programmierfassung spannungslos.

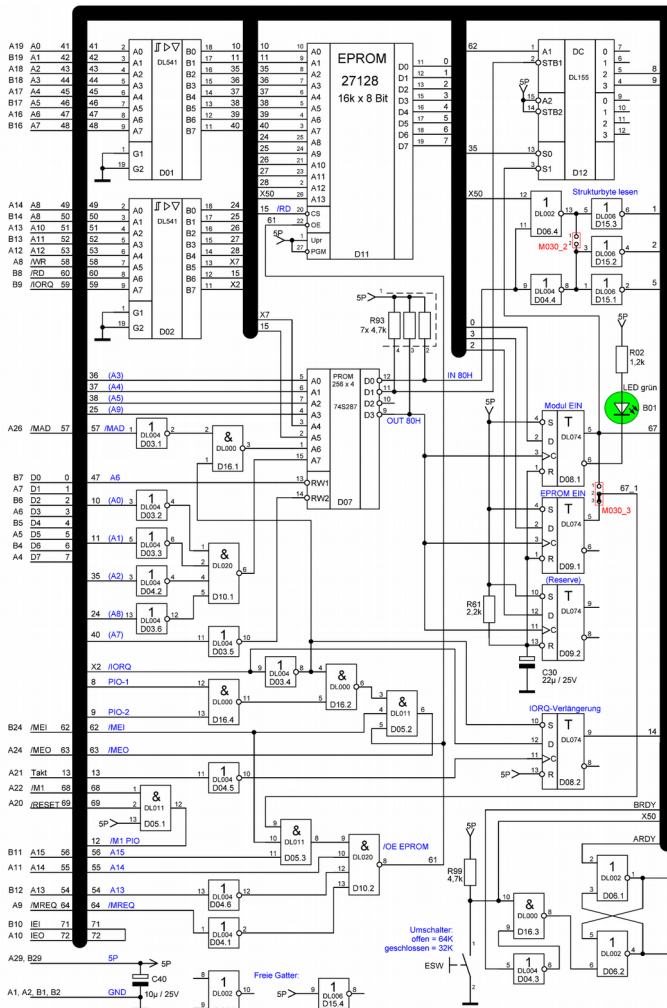
Es wird nicht geprüft, ob ein zweiter Parameter im IBN-Kommando angegeben worden ist. Fehlt der zweite Parameter, dann wird 0 angenommen.

Es kann also beispielsweise statt IBN 8 0 auch einfach nur IBN 8 eingegeben werden, um alle Adressleitungen auf Low-Pegel zu setzen.

6. Schaltplan

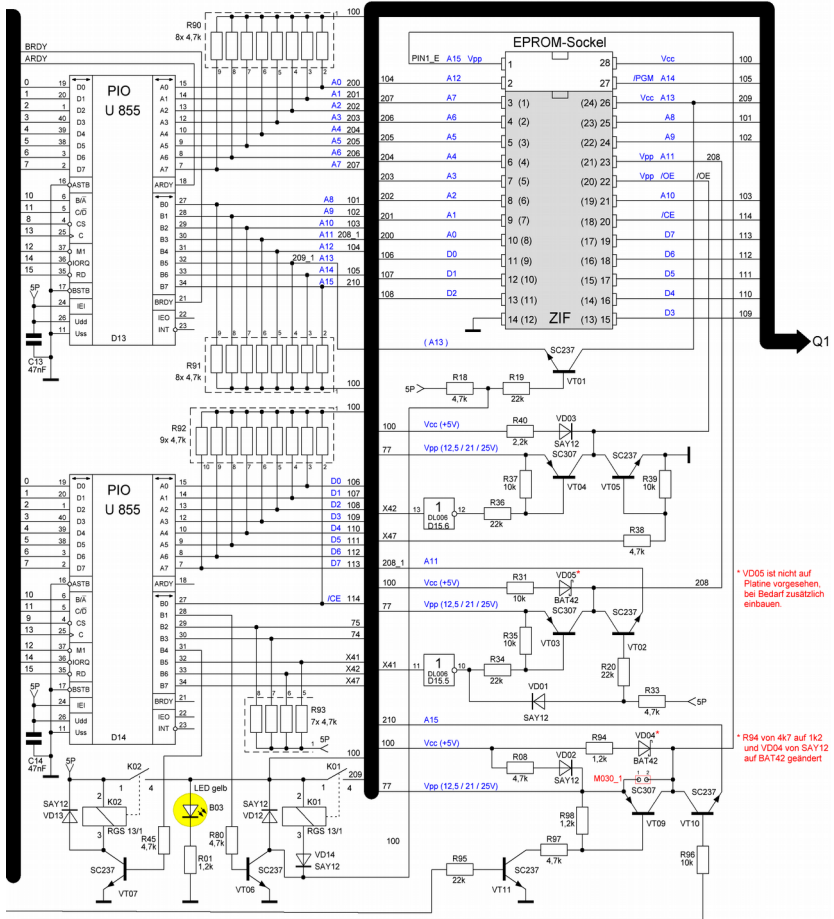
6.1. Digitalteil

Der Digitalteil des Original-M030 wurde auch beim Nachbau fast beibehalten:



6.2. Optionale 27(C)512-Erweiterung bei Neuauflage 2015

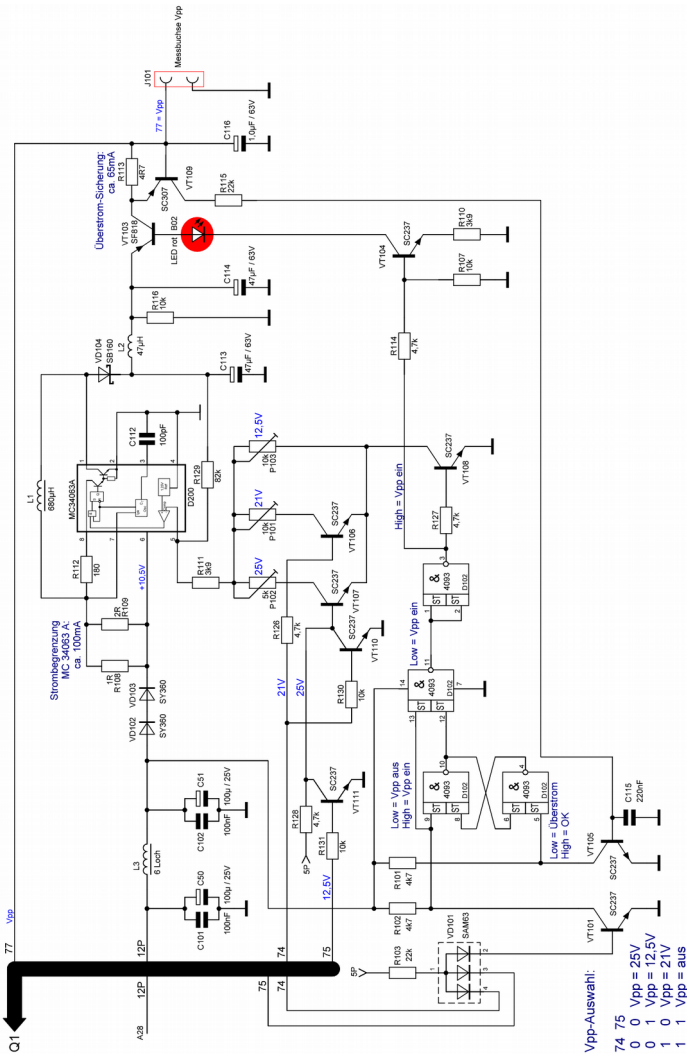
Erweitert wurde der Digitalteil mit einem Umschalter ESW für die beiden Modi 32K/64K. Dieser Umschalter steuert den EPROM-Bereich, das Strukturbyte und die zusätzliche Umschaltung von Pin 1 des EPROM-Sockels.



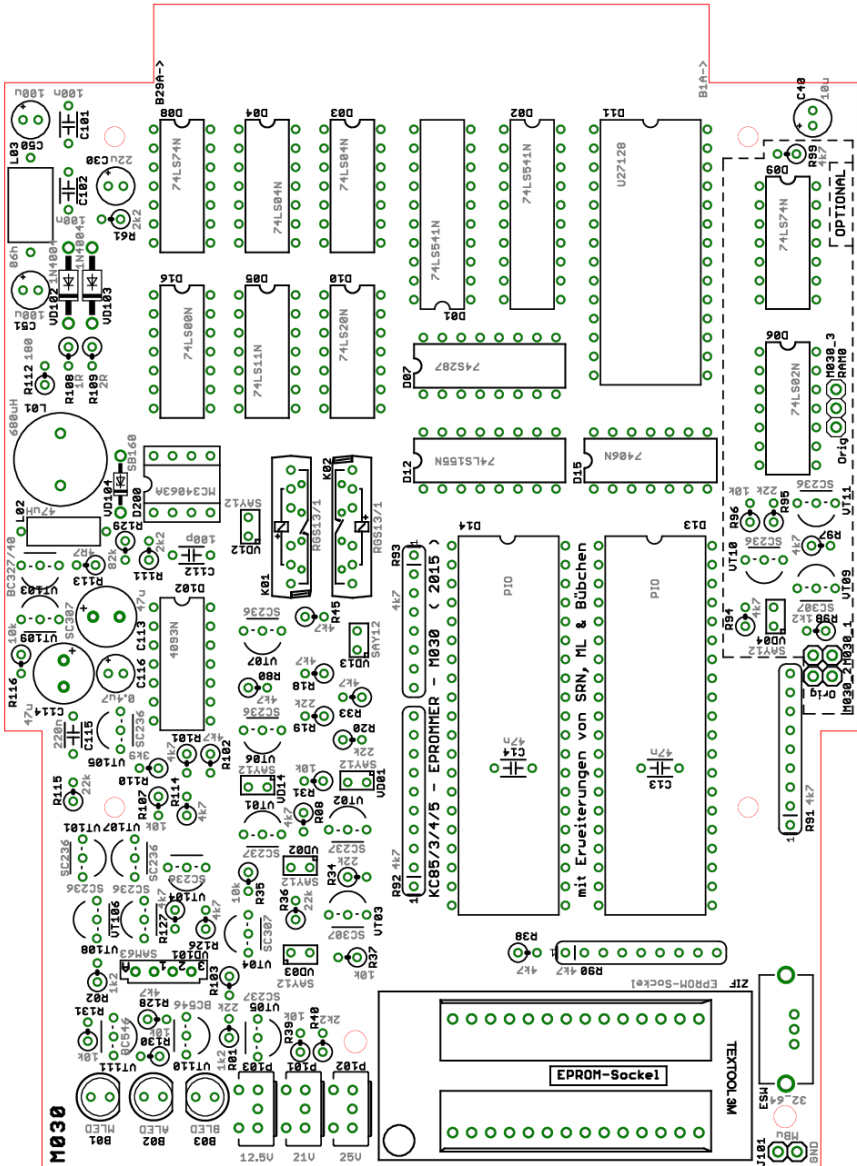
Dazu werden die Zustände der Signale ARDY und BRDY von PIO D13 in einem Flip-Flop gespeichert. So lässt sich mit Auswahl der PIO-Betriebsarten das Pin 1 steuern. Dieser Schaltungsteil ist für die Neuauflage 2015 neu entwickelt worden.

6.3. Analogteil

Komplett neu entwickelt wurde der Analogteil der Programmiervoltageserzeugung. Kern der Schaltung ist der Schaltregler MC34063A als Aufwärtswandler.



7. M030 Leiterplatte und Bestückung



8. M030 Stückliste

Schaltkreise	Typ	Reichelt Elektronik Bestellcode	Anzahl
D01, D02	74LS541N	LS 541	2
D03, D04	74LS04N	LS 04	2
D05	74LS11N	LS 11	1
D06*	74LS02N	LS 02	1 (0)
D07	74S287	Im Lieferumfang (programmiert)	1
D08*, D09	74LS74N	LS 74	2 (1)
D10	74LS20N	LS 20	1
D11	EPROM 27(C)128 EPROM 27(C)64	Im Lieferumfang (programmiert)	1 (1)
D12	74LS155N	LS 155	1
D13, D14	UB855D	(Z80 PIO)	2
D15	74LS06N	LS 06	1
D16	74LS00N	LS 00	1
D102	V4093	MOS 4093	1
D200	MC34063A	MC 34063 A	1

* ohne optionale 27512-Erweiterung

Fassungen	für	Reichelt Elektronik Bestellcode	Anzahl
8-polig	MC 34063 A	GS 8P	1
14-polig*	TTL, CMOS-IC	GS-KO 14P	11 (9)
16-polig	74LS155	GS-KO 16P	1
20-polig	74LS541	GS-KO 20P	2
28-polig	EPROM	GS-KO 28P	1
40-polig	PIOs	GS 40P	2
7polig	Relais K01, K02	SPL 20 (passend zuschneiden)	1

* ohne optionale 27512-Erweiterung

Stiftleisten	Typ	Reichelt Elektronik Bestellcode	Anzahl
M030_1 M030_2	2x2-polig RM 2.54	MPE 087-2-004	1
M030_3	3-polig RM 2.54	MPE 087-1-003	1
J101	Messbuchse	zum Beispiel: PS 25/2G WS	1
Steckbrücken		JUMPER 2,54	3

ZIF-Sockel	Typ	Reichelt Elektronik Bestellcode	Anzahl
EPROM-Sockel	Textool 28polig	Im Lieferumfang (oder TEX 28)	1
IC-Steckadapter	28polig	DILS 28 GO (Fischerelektronik)	1

Kondensatoren	Typ	Reichelt Elektronik Bestellcode	Anzahl
C13, C14	47n	X7R-5 47N	2
C30	22µ / 25V	RAD 22/25	1
C40	10µ / 25V	RAD 10/35	1
C50, C51	100µ / 25V	RAD 100/25	2
C101, C102	100n	X7R-5 100N	2
C112	100p	NPO-5 100P	1
C113, C114	47µ / 63V	RAD FC 47/63	2
C115	220n	Z5U-5 220N	1
C116	0,47µ / 63V	RAD 0,47/63	1

Widerstände	Typ	Reichelt Elektronik Bestellcode	Anzahl
R108, R109	1R2	1/4W 1,2	2
R113	4R7	1/4W 4,7	1
R112	180	1W 180	1
R01, R02, R94**, R98*	1k2	1/4W 1,2K	3 (2)
R40, R61, R111	2k2	1/4W 2,2K	3
R110	3k9	1/4W 3,9K	1
R08, R18, R33, R38, R45, R80, R101, R102, R114, R126, R127, R128, R97*, R99*	4k7	1/4W 4,7K	15 (12)
R31, R35, R37, R39, R107, R116, R130, R131, R96*	10k	1/4W 10K	9 (8)
R19, R20, R34, R36, R103, R115, R95*	22k	1/4W 22K	7 (6)
R129	82k	1/4W 82K	1
R90, R91	8 x 4k7	SIL 9-8 4,7K	2
R92	9 x 4k7	SIL 10-9 4,7K	1
R93	7 x 4k7	SIL 8-7 4,7K	1

* ohne optionale 27512-Erweiterung

** Bestückungsaufdruck = 4k7

Potentiometer	Typ	Reichelt Elektronik Bestellcode	Anzahl
P101 (21V)	10k	64Z-10K *	2
P103 (12,5V)			
P102 (25V)	5k	64Z-5,0K *	1

* Das Platinenlayout gestattet den Einsatz verschiedener Typen Einstellregler

Transistoren	Typ	Reichelt Elektronik Bestellcode	Anzahl
VT01, VT02, VT05, VT06, VT07, VT10*, VT11*, VT101, VT104, VT105, VT106, VT107, VT108, VT110, VT111	SC237	BC 547B (npn, $U_{CE}=45V$, $I_C=100mA$)	15 (13)
VT03, VT04, VT09*, VT109	SC307	BC 557B (pnp, $U_{CE}=45V$, $I_C=100mA$)	4 (3)
VT103	SF818	BC 327-40 (pnp, $U_{CE}=60V$, $I_C=500mA$)	1

* ohne optionale 27512-Erweiterung

Induktivitäten	Typ	Reichelt Elektronik Bestellcode	Anzahl
L01	680 μ H	L-09HCP 680 μ	1
L02	47 μ H	SMCC 47 μ	1
L03	06h	06H-85	1

Relais	Typ	Reichelt Elektronik Bestellcode	Anzahl
K01, K02	RGS13/1 (5V)	MS 7175-L 5V *	2

* Das Platinenlayout gestattet den Einsatz verschiedener Relais Typen

Dioden	Typ	Reichelt Elektronik Bestellcode	Anzahl
VD01, VD02, VD03, VD12, VD13, VD14, VD04*, VD05	SAY12	1N 4148	6
VD101	BAT42	BAT42	1 (2)
VD102, VD103	SAM63	Im Lieferumfang (oder 3x 1N 4004)	1
VD104	SY360	1N 4004	2
	SB160	SB 160	1

* VD04 nur bei optionaler 27512-Erweiterung

LED	Typ	Reichelt Elektronik Bestellcode	Anzahl
B01 (grün)	5mm / low current	LED 5MM 2MA GN	1
B02 (rot)	5mm / low current	LED 5MM 2MA RT	1
B03 (gelb)	5mm / low current	LED 5MM 2MA GE	1

Schalter	Typ	Bestellcode	Anzahl
ESW *	Schiebeschalter 1xUM WK	Im Lieferumfang	1

* kann durch Brücke ersetzt werden ohne optionale 27512-Erweiterung

9. Programmierung von 1K-EPROMs

Das Modul M030 (EPROMMER) ist ausgelegt, EPROMs von 2KByte (2716) bis 64KByte (27512) zu programmieren.

In vielen älteren Geräten (nicht im KC85 selbst) sind aber auch 1K-EPROMs 2708 bzw. deren DDR-Vergleichstyp U555C eingesetzt.

Um das M030 auch für 1K-EPROMs nutzen zu können, müssen einige zusätzliche Voraussetzungen geschaffen werden. Das sind von der Hardware:

- Bereitstellung einer zusätzlichen Betriebsspannung von -5V (Ubb, Pin 21)
- Bereitstellung einer zusätzlichen Betriebsspannung von +12V (Udd, Pin 19)
- Einhaltung einer definierten Einschaltreihenfolge der Spannungen: zuerst die -5V und danach die +5V und +12V
- Abschaltung in umgekehrter Reihenfolge, die -5V also zuletzt
- Programmierspannungsimpuls von +25V mit 1ms Dauer an Pin 18
- +12V an Pin 20 um den Programmiermodus des EPROMs zu erreichen

Und von der Software Einhaltung eines fest vorgeschriebenen Programmialgorithmus':

- je Adresse ein Programmierimpuls von 1ms
- 100maliger Durchlauf aller 1024 Adressen

Längere Programmierimpulse und kürzere Wiederholungen der gleichen Adresse könnten diese ältere Generation von EPROMs zerstören.

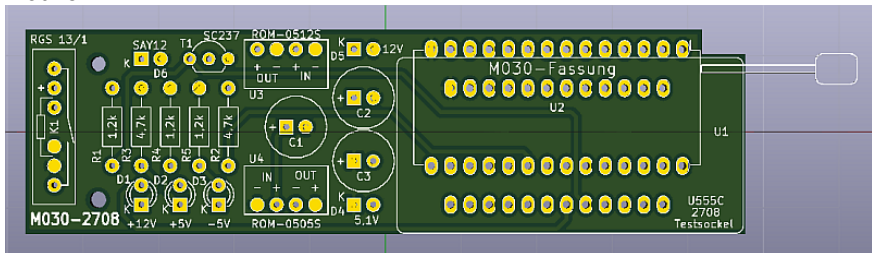
Zur Realisierung eines entsprechenden Adapters gibt es zwei unterschiedliche Ansätze. Beide setzen für die Bereitstellung der zusätzlichen Spannungen DC/DC-Converter ein, welche aus den +5V jeweils die -5V und die +12V erzeugen.

9.1. 2708 – Adapter (ML)

Der Programmieradapter von Mario Leubner ist speziell **nur** für das M030 entwickelt worden. Dabei wurden alle Hardwaremöglichkeiten des M030 ausgenutzt, um die Zusatzhardware des Adapters so einfach wie möglich zu halten. Den Rest übernimmt die Software.

Diese Variante ist auf einer einzigen Platine realisiert worden, welche unten zwei 14polige Stiftleisten für die Aufnahme im M030-Sockel und oben eine 24polige Schwenkhebelfassung sowie die restlichen Bauteile enthält.

Platine:



Aufbau:



Zur Schaltung dieses Adapters:

Das M030 besitzt einen 28poligen Programmiersockel, bei dem sich Pin 26 (für die kleineren EPROMs 2716 und 2732) separat als Versorgungsspannung zuschalten lässt. Pin 28 ist dagegen immer an +5V, sobald irgend ein EPROM mit Ucc versorgt wird. Diese Tatsache wird ausgenutzt für die Realisierung der Einschaltreihenfolge der Spannungen. Aus den +5V an Pin 28 wird mittels DC/DC-Converter die -5V erzeugt und direkt an Pin 21 der 2708-Fassung angelegt. Außerdem wird aus diesen +5V mit einem zweiten DC/DC-Converter eine +12V-Spannung erzeugt, aber noch nicht am EPROM angelegt.

Die Software wartet jetzt eine kurze Zeit, bevor die +5V auch an Pin 26 angelegt werden. Diese +5V gehen jetzt direkt zu Pin 24 der 2708-Fassung und schalten ein Reedrelais, welches die +12V an Pin 19 der 2708-Fassung anlegt. Beim Abschalten ist ebenfalls eine kleine Verweilzeit in der Software realisiert, um sicherzustellen, dass die -5V auch als letztes noch anliegen am 2708.

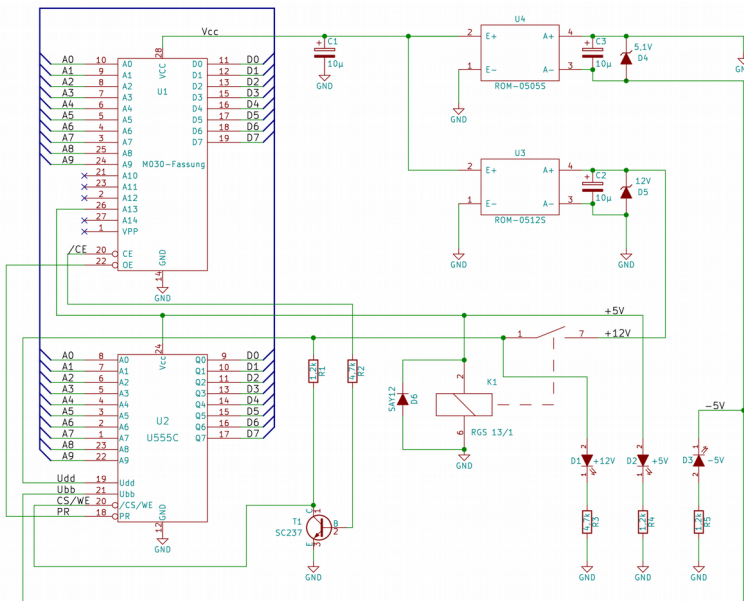
Pin 22 des M030-Programmiersockels kann einen Vpp-Impuls liefern, da dieser für die EPROM-Typen 2732 und 27512 dort benötigt wird. Dieser Vpp-Impuls wird direkt zu Pin 18 der 2708-Fassung geleitet.

Die +12V-Spannung für Pin 20 der 2708-Fassung wird aus dem Signal /CE von Pin 20 des M030 generiert. Um den nötigen 12V-Pegel für den 2708 zu bekommen, ist ein npn-Transistor dazwischen geschaltet, der nach Masse schaltet bzw. im gesperrten Zustand die 12V-Spannung von einem Widerstand von +12V an Pin 20 der 2708-Fassung freigibt.

Damit ist die Funktion des Adapters bereits realisiert.

Zur Erhöhung des Komforts sind noch drei LEDs zur Anzeige der am 2708 anliegenden drei Versorgungsspannungen vorgesehen.

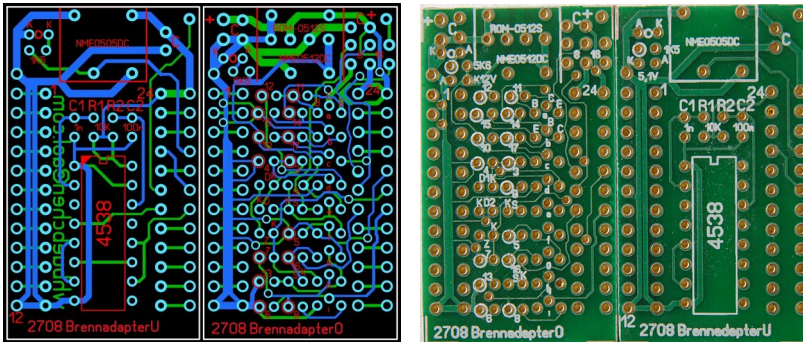
Je eine Z-Diode am Ausgang der DC/DC-Converter soll dafür sorgen, dass die Spannungen im Leerlauf nicht über die maximal zulässige Spannung ansteigen. Drei optionale Kondensatoren sind am Eingang und an den Ausgängen der DC/DC-Converter vorgesehen, um die Spannungen bei Bedarf noch etwas zu stabilisieren.



9.2. 2708 bzw. TMS2716 – Adapter (WH)

Der Programmieradapter von **Wolfgang Harwardt** verwendet die Signale wie sie beim Lesen und Brennen eines 2K-EPROMs 2716 anliegen und erzeugt daraus die nötigen Signale für den 1K-EPROM 2708. Dazu wurden zwei kleine Platinen entwickelt, die übereinander gestapelt werden und dazwischen alle zusätzlichen Bauteile aufnehmen. Vom Ansatz her könnte der Adapter auch auf anderen EPROM-Brennern funktionieren. Zur Sicherheit wird deshalb der Programmierimpuls in Hardware auf eine maximale Länge von 1ms begrenzt.

Platinen:

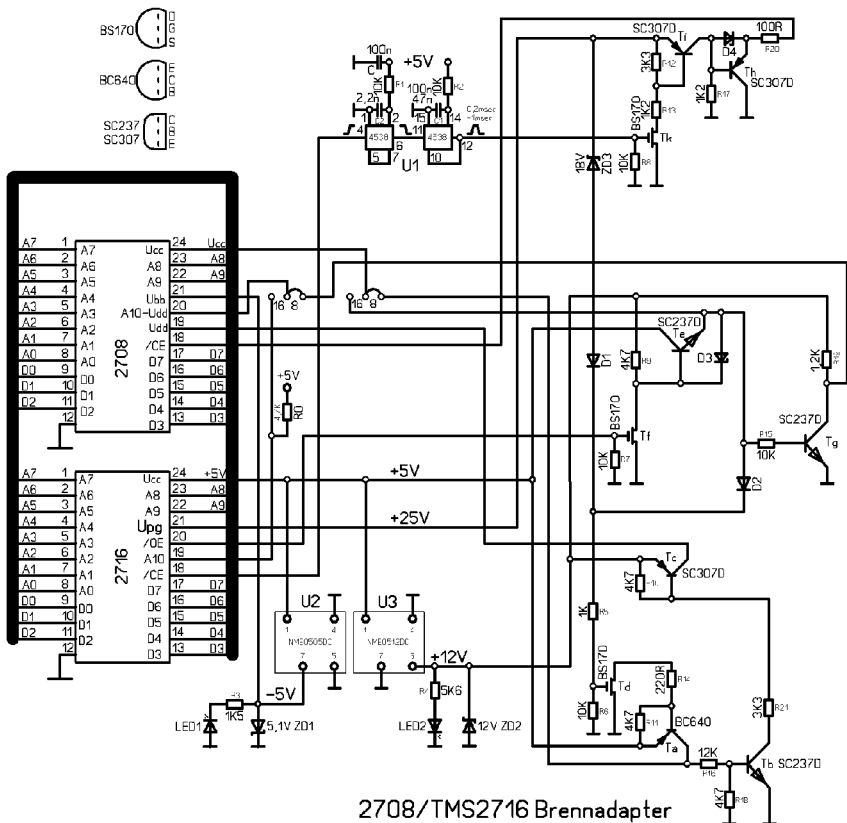


Aufbau:



Der Schaltplan dieses 2708/TMS2716-Adapters zeigt zwei DC/DC-Converter zur Erzeugung der -5V und der +12V. Die -5V werden sofort am 2708/TMS2716 angelegt, wenn Pin 24 Spannung bekommt. Die +5V und +12V werden erst dann angelegt, wenn beim Lesen /OE Low-Pegel hat oder beim Programmieren die Programmierspannung von +25V anliegt.

Die beiden Gatter des 4538 sind für die Impulsformung des Programmierimpulses zuständig. Die +12V an Pin 20 des 2708 werden bei /OE=Low angelegt. Zwei Steckbrücken bzw. Umschalter gestatten die Verwendung für den EPROM-Typ TMS2716, der wie der 2708 auch die beiden zusätzlichen Betriebsspannungen benötigt. Hier werden jedoch an Pin 20 die Adresse A10 und an Pin 24 beim Programmieren +12V oder 0V benötigt – in dieser Schaltung wird 0V angelegt.



2708/TMS2716 Brennadapter
W.Harwardt 24.03.2016

9.3. Stücklisten der beiden 2708 – Adapter

Stückliste für Adapter (ML):

Bezeichner	Typ	Reichelt Elektronik Bestellcode	Anzahl
C1, C2, C3	10µF / 35V	RAD 10/35	3
D1, D2, D3	3mm low current (rot, gelb, grün)	LED 3MM 2MA GE/GN/RT	3
D4	SZX21/5,1	ZF 5,1	1
D5	SZX21/12	ZF 12	1
D6 *	SAY12	1N 4148	1
K1	RGS 13/1 (5V)	SIL 7271-D 5V	1
R1, R4, R5	1,2k	1/4W 1,2K	3
R2, R3	4,7k	1/4W 4,7K	2
T1	SC237	BC 547B (npn, U _{CE} =45V, I _C =100mA)	1
U1	14polige Stiftleiste	MPE 087-1-016 (kürzen!)	2
U2	Testsockel 24pin	1366937 (Conrad Electronic)	1
U3	ROM-0512S	154783 (Conrad Electronic)	1
U4	ROM-0505S	154777 (Conrad Electronic)	1

* D6 kann entfallen, wenn ein Relais mit integrierter Diode eingesetzt wird.

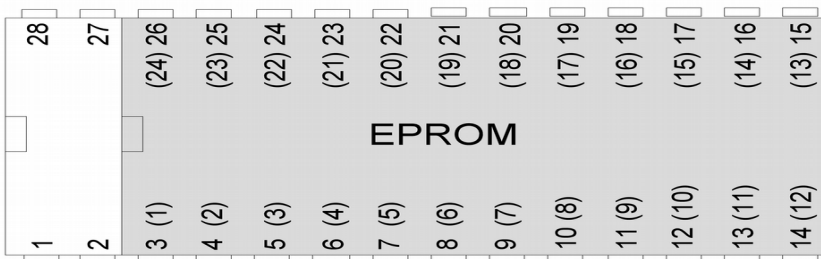
Stückliste für Adapter (WH):

Bezeichner	Reichelt Elektronik Bestellcode	Anzahl
R0, R9, R10, R11, R18	METALL 4,70K – für R0 SMD empfehlenswert	5
R1, R2, R6, R7, R8, R15	METALL 10,0K – für R1, R2 kleine Bauform 0204 oder SMD	6
R3	METALL 1,50K – für R3 SMD empfehlenswert	1
R4	METALL 5,60K	1
R5	METALL 1,00K	1
R12, R21	METALL 3,30K	2
R13, R17, R19	METALL 1,20K	3
R14	METALL 220	1
R16	METALL 12,0K	1
R20	METALL 100	1
C1	MKS-2 100N	1
C2	KERKO 2,2N	1
C	X7R-2,5 100N	3
ZD1	ZF 5,1, Z-Diode 5,1V	1
ZD2	ZF 12, Z-Diode 12V	1
ZD3	ZF 18, Z-Diode 18V	1
LED1, LED2	LED 3MM 2MA RT	2
D1, D2	Si Schaltodiode 1N4148	2
D3, D4	Schotky Diode BAT42	2
Ta	BC 640 (npn-Transistor TO-92 U _{CE} =80V I _C =1A 0,8W)	1
Tb, Te, Tg	SC237D (DDR npn-Transistor U _{CE} =45V, I _C =100mA)	3
Tc, Th, Ti	SC307D (DDR pnp-Transistor U _{CE} =45V, I _C =100mA)	3
Td, Tf, Tk	BS 170 (N-FET-Transistor TO-92 60V 0,5A 0,35W)	3
U1	V4538 oder 74HCT4538	1
U2	NME0505DC oder ROM-0505S	1
U3	NME0512DC oder ROM-0512S	1
2716	12poliger Pfostenfeldstecker mit ca. 12mm langen Stiften alternativ: AW122/32 Adapterleiste + GS24P	2
2708	Testsockel 24pin, MPE 115-1-036 Buchsenleiste, AW 122/32 Adapterleiste	je 1
Leiterplattenverbinder	Buchsenleiste MPE 094-1-016 und Stiftleiste MPE 087-1-016	je 2
Auswahl 2708/2716	SL 2X10W 2,54, Pfostenfeldstecker 2-reihig, gewinkelt mit JUMPER 2,54 Alternative: 2 Stück Subminiaturschalter NK236	1

10. Pinbelegung der EPROM-Typen

Belegung der unterstützten EPROM-Typen (*2708 nur mit Adapter):

2708*	2716	2732	2764	27128	27256	27512
			U _{CC}	U _{CC}	U _{CC}	U _{CC}
			PGM	PGM	A14	A14
	U _{CC}	U _{CC}	n.c.	A13	A13	A13
A8	A8	A8	A8	A8	A8	A8
A9	A9	A9	A9	A9	A9	A9
U _{DS} -5V	U _{PR}	A11	A11	A11	A11	A11
CS/WE	OE	OE/U _{PR}	OE	OE	OE	OE/U _{PR}
U _{DD} +12V	A10	A10	A10	A10	A10	A10
PR	OE	OE	OE	OE	OE	OE
D7	D7	D7	D7	D7	D7	D7
D6	D6	D6	D6	D6	D6	D6
D5	D5	D5	D5	D5	D5	D5
D4	D4	D4	D4	D4	D4	D4
D3	D3	D3	D3	D3	D3	D3



27512	27256	27128	2764	2732	2716	2708*
A15	U _{PR}	U _{PR}	U _{PR}			
A12	A12	A12	A12			
A7	A7	A7	A7	A7	A7	A7
A6	A6	A6	A6	A6	A6	A6
A5	A5	A5	A5	A5	A5	A5
A4	A4	A4	A4	A4	A4	A4
A3	A3	A3	A3	A3	A3	A3
A2	A2	A2	A2	A2	A2	A2
A1	A1	A1	A1	A1	A1	A1
A0	A0	A0	A0	A0	A0	A0
D0	D0	D0	D0	D0	D0	D0
D1	D1	D1	D1	D1	D1	D1
D2	D2	D2	D2	D2	D2	D2
U _{DS}	U _{SS}	U _{SS}	U _{SS}	U _{SS}	U _{SS}	U _{SS}

11. Übersicht der getesteten EPROMs

Die auf den nächsten Seiten gezeigten Impulsdiagramme zeigen die Signalverläufe wie sie am M030 mit der 64K-Software realisiert wurden. Für jede EPROM-Größe gibt es genau eine Signalfolge, die für alle EPROMs der gleichen Größe zur Anwendung kommt.

Das gezeigte Diagramm für den 2708 gilt nur für den ML-Adapter. Für den WH-Adapter gilt der Signalverlauf wie beim 2716.

Bei Problemen können diese Impulsdiagramme mit denen der EPROM-Datenblätter verglichen werden.

In den Tabellen unter den Diagrammen ist eine Auswahl von EPROM-Typen aufgeführt, die erfolgreich mit dem M030-Nachbau programmiert worden sind. Angegeben sind dazu auch die eingestellten Parameter.

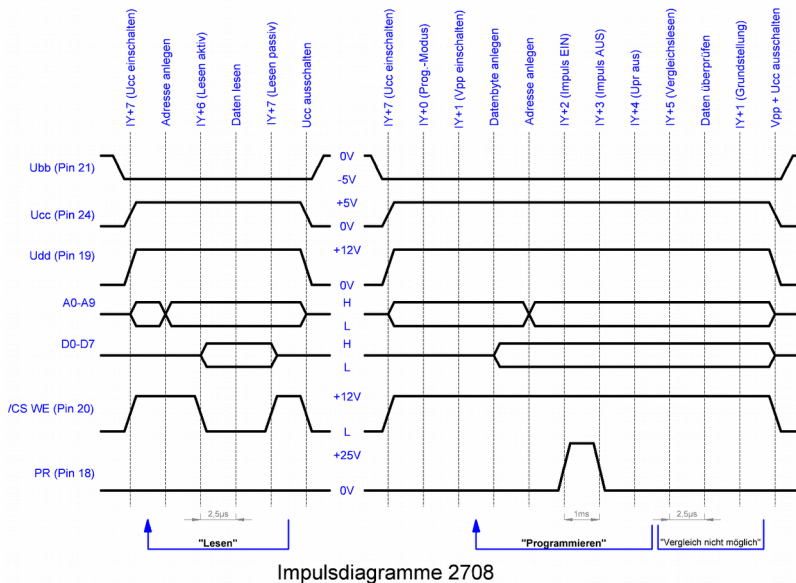
Dabei bedeuten Ziffern in der Spalte „Methode“ den Menüpunkt der 32K-Software und Buchstaben die Methode der 64K-Software, mit denen die EPROMs programmiert wurden.

Der farbige Punkt in der letzten Spalte der Tabelle (gekennzeichnet mit „?“) gibt an, wer den EPROM-Test durchgeführt hat:

-  Mario Leubner
-  Manfred Bagola
-  Wolfgang Harwardt
-  René Nitzsche

Die EPROMs wurden nach Möglichkeit mit den Einstellungen laut Datenblatt programmiert. Ein * hinter dem EPROM-Typ bedeutet, dass kein Datenblatt vorhanden ist und der EPROM deshalb mit den Standard-Einstellungen programmiert worden ist.

11.1. Zeitdiagramme 1K-EPROM 2708 / U555C



Impulsdigramme 2708

Mit Adapter (ML) programmierte EPROM-Typen:

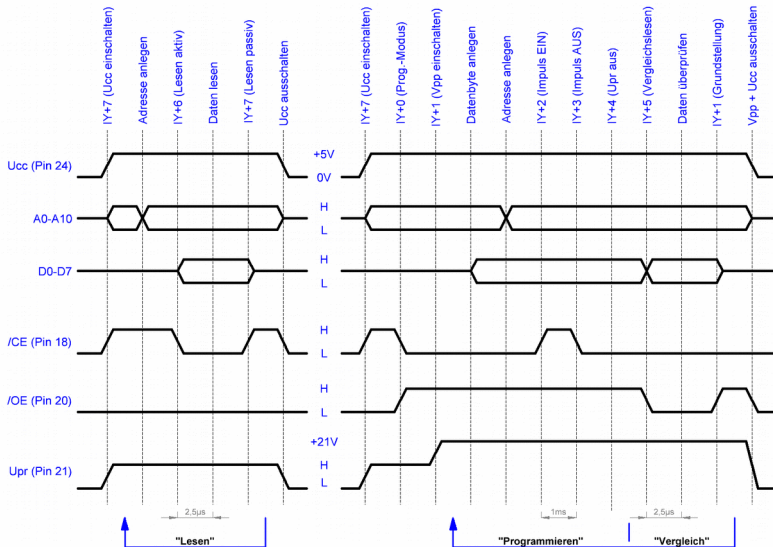
Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
MME (DDR)	U555C	B	25V	100x1,0ms / 15x1,0ms	●
MME (DDR)	U505C	-	-	nur lesen (ROM)	●
Texas Instruments	TMS 2708JL	B	25V	100x1,0ms / 15x1,0ms	●
UdSSR	K573PΦ1	B	25V	100x1,0ms / 15x1,0ms	●

Mit Adapter (WH) programmierte EPROM-Typen:

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
DDR	2708 S9	B	25V	100x1,0ms / 15x1,0ms	●
Fairchild	F2708	B	25V	100x1,0ms / 15x1,0ms	●
Mitsubishi	MCM2708	B	25V	100x1,0ms / 15x1,0ms	●
Mostek	SFF71708	B	25V	100x1,0ms / 15x1,0ms	●
MME (DDR)	U555C	B	25V	100x1,0ms / 15x1,0ms	●
Texas Instruments	TMS 2708JL	B	25V	100x1,0ms / 15x1,0ms	●
UdSSR	K573PΦ1	B	25V	100x1,0ms / 15x1,0ms	●
unbekannt	M2708F	B	25V	100x1,0ms / 15x1,0ms	●
Texas Instruments	TMS 2716	B	25V	100x1,0ms / 45x1,0ms	●

Der TMS 2716 benötigt -5V und +12V, deshalb nur mit WH-Adapter möglich!

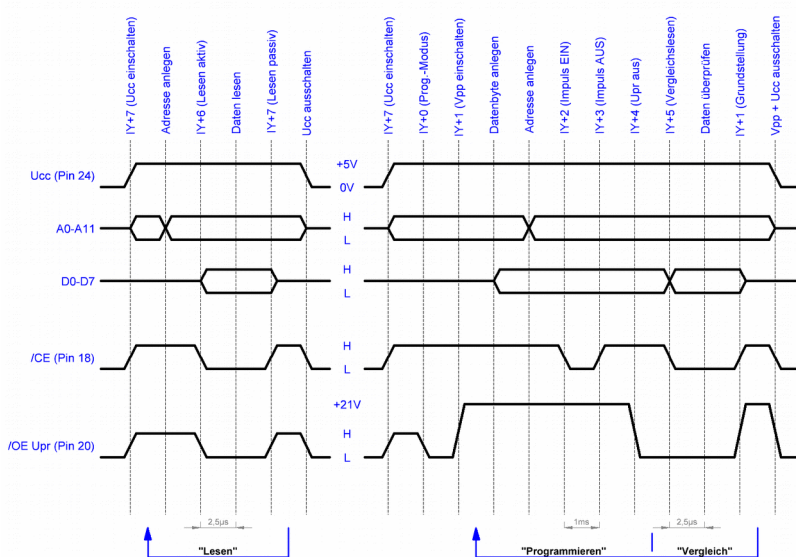
11.2. Zeitdiagramme 2K-EPROM 27(C)16



Impulssdiagramme 2716

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
AMD	AM2716	1, A	25V	50x1,0ms / 2x2,0ms	●
Fujitsu	MB8516	A und B	25V	50x1,0ms / 2x2,0ms	●
Fujitsu	MBM2716H	A und B	25V	50x1,0ms / 2x2,0ms	●
Intel	C2716 *	A und B	25V	50x1,0ms / 2x2,0ms	●
MME (DDR)	U2716C35	A und B	25V	50x1,0ms / 2x2,0ms	●
Mostek	MK2716T	A und B	25V	50x1,0ms / 2x2,0ms	●
National Semiconductor	NM27C16Q	A und B	25V	50x1,0ms / 2x2,0ms	●
NEC	D2716D-2	A und B	25V	50x1,0ms / 2x2,0ms	●
OKI	MSM2716AS	A und B	25V	50x1,0ms / 2x2,0ms	●
ST (SGS)	M2716	1, A und B	25V	50x1,0ms / 2x2,0ms	●
Tesla	MHB2716 *	A und B	25V	50x1,0ms / 2x2,0ms	●
Texas Instruments	TMS 2516	A und B	25V	50x1,0ms / 2x2,0ms	●
Thomson	ET 2716Q-I	1	25V	unbekannt	●
UdSSR	K573PΦ2	1, A und B	25V	50x1,0ms / 2x2,0ms	●
UdSSR	K573PΦ5	A	25V	50x1,0ms / 2x2,0ms	●
UdSSR	KC573PΦ2 *	A und B	25V	50x1,0ms / 2x2,0ms	●

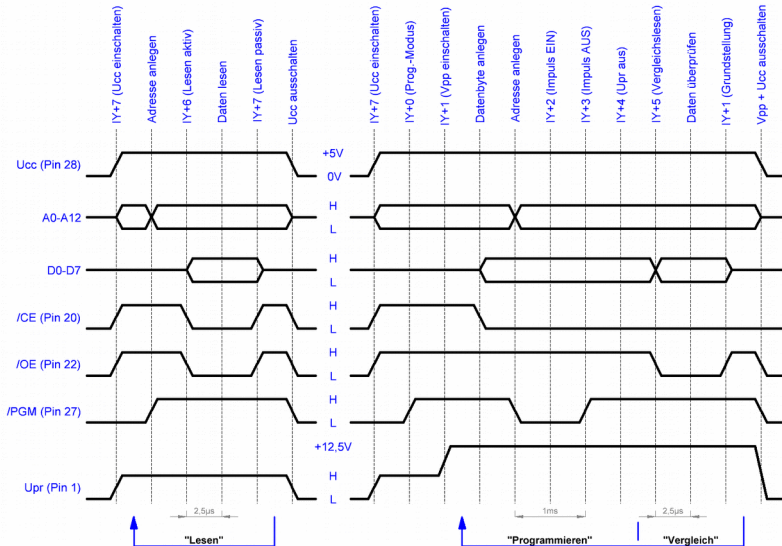
11.3. Zeitdiagramme 4K-EPROM 27(C)32



Impulsdiagramme 2732

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
Hitachi	HN462732G	A und B	25V	45x1,0ms / 2x2,0ms	●
Hitachi	HN482732AG	A und B	21V	45x1,0ms / 2x2,0ms	●
Intel	D2732A	A und B	21V	45x1,0ms / 2x2,0ms	●
Mitsubishi	M5L2732K *	A und B	25V	45x1,0ms / 2x2,0ms	●
MME (DDR)	U2732CC35	2, A und B	21V	45x1,0ms / 2x2,0ms	●
National Semiconductor	NMC27C32BQ	A und B	12,5V	45x1,0ms / 2x2,0ms	●
unbekannt	2732AJL-45 *	A und B	21V	45x1,0ms / 2x2,0ms	●

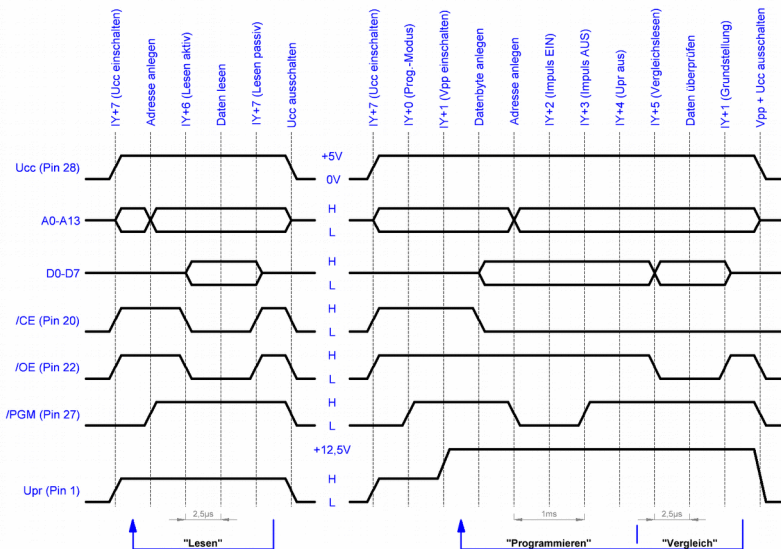
11.4. Zeitdiagramme 8K-EPROM 27(C)64



Impulssdiagramme 2764

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
AMD	AM27C64-150	A	12,5V	25x0,1ms / 1x0,2ms	●
AMD	AM2764-2DC *	A und B	21V	25x1,0ms / 1x2,0ms	●
AMD	AM2764ADC	A und B	12,5V	25x1,0ms / 1x2,0ms	●
Fujitsu	MBM2764-30	3, A und B	21V	20x1,0ms / 1x1,0ms	●
Hitachi	HN27C64	A	21V	1ms	
Hitachi	HN482764G-4	3	21V	unbekannt	●
Intel	D2764A	A	12,5V	1ms	
Mitsubishi	M5L2764K	A	21V	15x1,0ms / 1x1,0ms	●
National Semiconductor	NMC27C64Q200	A und B	12,5V	20x0,5ms / 1x0,5ms	●
NEC	D27C64D-25 *	A	21V	25x1,0ms / 1x2,0ms	●
SEEQ	2764	A	21V	1ms	
ST (SGS)	M2764A	3, A und B	12,5V	25x1,0ms / 1x2,0ms	●
ST (SGS)	M27C64A	A und B	12,5V	25x1,0ms / 1x2,0ms	●
Texas Instruments	TMS27C64	A	12,5V	1ms	

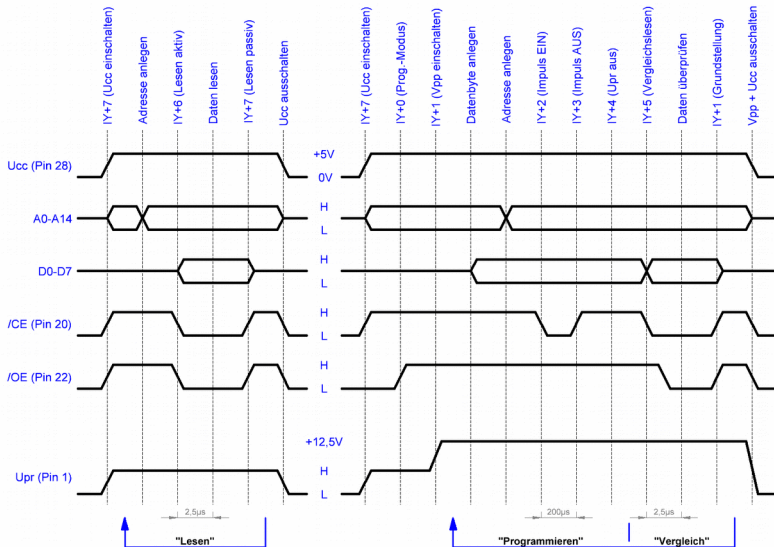
11.5. Zeitdiagramme 16K-EPROM 27(C)128



Impulsdiagramme 27128

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
AMD	AM27128A	A und B	12,5V	25x0,1ms / 1x0,1ms	●
Fujitsu	MBM27128	4, A und B	21V	20x1,0ms / 1x1,0ms	●
General Instrument	27C128-25	A und B	12,5V	25x1,0ms / 1x3,0ms	●
Intel	D27128	A und B	21V	25x1,0ms / 1x3,0ms	●
Intel	D27128A	A und B	12,5V	25x1,0ms / 1x3,0ms	●
Intel	D27C128 *	A und B	12,5V	25x0,1ms / 1x0,1ms	●
Microchip	27C128	A	12,5V	5x1,0ms / 1x2,0ms	●
Mitsubishi	M5L27128K	4, B	21V	5x1,0ms / 1x2,0ms	●
NEC	D27128D	A	21V	5x1,0ms / 1x2,0ms	●
SEEQ	27C128	A	12,5V	5x1,0ms / 1x2,0ms	●
SEEQ	DQ27128K	A	21V	5x1,0ms / 1x2,0ms	●
ST (SGS)	M27128A	4, A und B	12,5V	25x1,0ms / 1x3,0ms	●
Taiwan	27C128-25	A und B	12,5V	1ms	●
Texas Instruments	TMS27C128	A und B	12,5V	10x0,1ms / 1x0,1ms	●
Toshiba	TMM27128AD-15	A und B	12,5V	5x1,0ms / 1x2,0ms	●
Toshiba	TMM27128DI-20 *	A und B	21V	5x1,0ms / 1x2,0ms	●

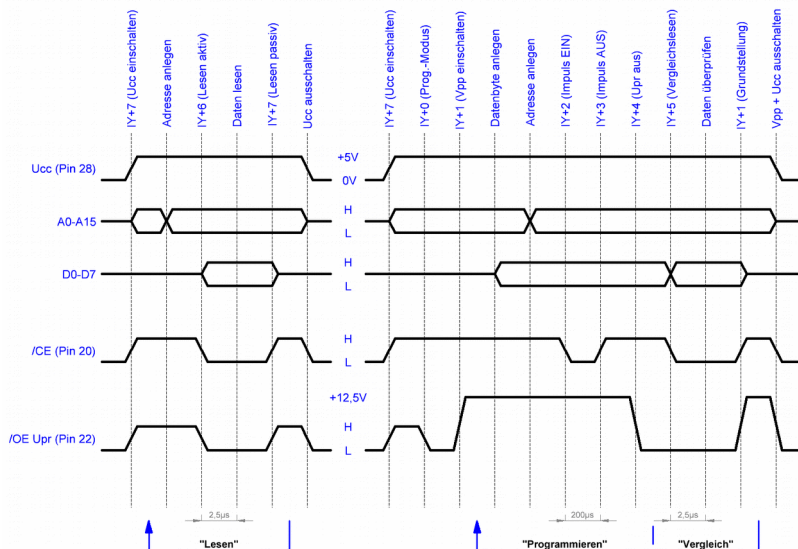
11.6. Zeitdiagramme 32K-EPROM 27(C)256



Impulsdiagramme 27256

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
AMD	AM27C256-205DC	5	12,5V	unbekannt	●
Atmel	AT27C256R	A und B	12,5V	10x0,1ms / 1x0,1ms	●
Hitachi	HN27C256G-20	A und B	12,5V	5x1,0ms / 1x2,0ms	●
Intel	D27256	A und B	12,5V	25x0,1ms / 1x0,1ms	●
Microchip	27HC256-70	A und B	12,5V	10x0,1ms / 1x0,1ms	●
National Semiconductor	NM27C256Q	A und B	12,5V	25x1,0ms / 1x1,0ms	●
ST (SGS)	M27256-2FI	5	12,5V	unbekannt	●
ST (SGS)	M27256F1	A und B	12,5V	5x1,0ms / 1x2,0ms	●
ST (SGS)	M27C256B	A und B	12,5V	25x0,1ms / 1x0,1ms	●
Texas Instruments	TMS 27C256-10	A und B	12,5V	10x0,1ms / 1x0,1ms	●
Toshiba	TC57HC256D	A	12,5V	5x1,0ms / 1x2,0ms	●

11.7. Zeitdiagramme 64K-EPROM 27(C)512



Impulsdiagramme 27512

Hersteller	Bezeichnung	Methode	Vpp	Impulsdauer	?
AMD	Am27C512-200DC	A und B	12,5V	25x0,1ms / 1x0,1ms	●
Hitachi	HN27512G-25	A und B	12,5V	25x1,0ms / 1x3,0ms	●
Intel	D27C512	A und B	12,5V	25x0,1ms / 1x0,1ms	●
National Semiconductor	NM27C512Q150	A und B	12,5V	25x,01ms / 1x0,1ms	●
S (Signetics?)	27C512	A und B	12,5V	25x0,1ms / 1x0,1ms	●
ST (SGS)	M27C512	A und B	12,5V	25x0,1ms / 1x0,1ms	●
Texas Instruments	TMS 27C512-15	A und B	12,5V	10x0,1ms / 1x0,1ms	●
Texas Instruments	TMS JL 27512-12	A und B	12,5V	10x0,1ms / 1x0,1ms	●
Toshiba	TMM27512D	A und B	12,5V	25x1,0ms / 1x3,0ms	●

12. Internet-Links zum M030

Weitergehende Informationen zum Modul M030, sowie aktuelle Software finden sich auf den folgenden Websites:

http://www.mpm-kc85.de/html/m030_epromer.htm

oder

http://www.kcclub.de/index.php?option=com_content&view=article&id=601&Itemid=632

Beschreibung des Original-M030

http://www.kcclub.de/index.php?option=com_content&view=article&id=427&Itemid=462

oder

<http://www.iee.et.tu-dresden.de/~kc-club/02/KCN99-02/KCN99-02-07.HTML>

Hier ist die Fertigstellung eines unvollständigen Original-M030 beschrieben. Der Artikel ist in den KC-News 02/1999 erschienen.

http://kc-und-atari.idylle-am-waldrand.com/kc85/m030_g.html

Nachbau des M030 auf der Website von René Nitzsche

<http://buebchen.jimdo.com/selbst-gebaut-diy-homebrew-feito-por-mim/kc85-2-3/m030/>

Nachbau des M030 auf der Website von Wolfgang Harwardt

<http://buebchen.jimdo.com/selbst-gebaut-diy-homebrew-feito-por-mim/8-bit-selbstbau-8-bit-diy/auslese-und-programmieradapter-2708-progadapter-2708/>

Beschreibung des 2708-Adapters auf der Website von Wolfgang Harwardt

<http://susowa.homeftp.net>

Beschreibung des 2708-Adapters von Mario Leubner.

Im Downloadbereich befinden sich außerdem die aktuelle Software und das Handbuch als PDF-Datei.

<http://www.robotrontechnik.de/html/forum/thwb/showtopic.php?threadid=11704>

Diskussion zum Nachbau des M030

<http://www.robotrontechnik.de/html/forum/thwb/showtopic.php?threadid=12980>

Diskussion zum 2708-Adapter für das M030

Notizen

Notizen

mikroelektronik

RFT



© KC-Club: SRN, Bübchen & ML