

## **Technische Beschreibung**

### **Universelles EPROM Programmiergerät**

**Version 2.01**

**Adaptersockel Version 2.0**

**Steuerprogramm Version 2.X**

### **EPROM-Emulator**

**Version 2.1**

**Vor Inbetriebnahme des Gerätes bitte vollständig durchlesen**

## Inhaltsverzeichnis:

### Teil I

EPROM-Programmiergerät mp V2.01

### Teil II

Adaptersockel ma51 V2.0

### Teil III

Steuerprogramm mp V2.21

### Teil IV

EPROM-Emulator me V2.1

Wir hoffen, daß die hierin beschriebenen Geräte Ihnen Ihre tägliche Arbeit erleichtern und Ihnen stets gute Werkzeuge sind. Verbesserungsvorschläge nehmen wir jederzeit gerne entgegen. Nur mit Ihrer Mitwirkung ist es uns möglich die Werkzeuge Ihren Bedürfnissen bestmöglichst anzupassen. In diesem Sinne wünschen wir Ihnen und auch uns stets eine kreative Kritik.

Mit freundlichem Gruß,

SENG digitale Systeme, im November 1991

Copyright (C) 1991 by SENG digitale Systeme Göppingen. Alle Rechte vorbehalten. Nachdruck - auch auszugsweise - nur mit Genehmigung der Verfasser. Technische Änderungen vorbehalten. Eine Haftung für die Richtigkeit des technischen Inhalts ist ausgeschlossen. Wir haften nicht für Druckfehler und irrtümliche Angaben. Ebenso haften wir nicht für Schäden die unmittelbar oder mittelbar durch unsere Produkte verursacht werden.

## TEIL I

### EPROM-Programmiergerät mp V2.01

#### 1.1 Einleitung

#### 2.1 Grundlagen der Programmierung

#### 3.1 Programmierlogik

#### 3.2 Schutzschaltung

#### 3.3 Spannungsversorgung

#### 4.1 Aufbau und Abgleich

#### 5.1 Technische Daten

#### 5.2 Funktionstabellen

#### 5.3 Programmieralgorithmen

#### 5.4 Stückliste

#### 5.5 Schaltbild

#### 5.6 Bestückungsplan

## 1.1 Einleitung

Dieses Programmiergerät bietet die Möglichkeit fast alle erhältlichen EPROM-Typen (auch 27512 und 27513) zu programmieren. Es entstand aus dem Wunsch heraus die Vorteile eines Stand-Alone-Programmiergerätes (Vielseitigkeit, Verwendbarkeit mit den unterschiedlichsten Rechnertypen) mit denen eines Rechner-spezifischen Gerätes (einfacher Aufbau, da ohne eigene 'Intelligenz') zu verbinden. Das Gerät sollte ferner die Möglichkeit bieten sowohl EPROMs mit intelligenten Programmialgorithmen als auch die älteren 25V Typen programmieren zu können. Der Verzicht auf unübersichtliche Bedienelemente, sowie der Aufbau mit einer eigenen Stromversorgung und der Einbau einer Sicherung zum Schutz des Chips schien angebracht. Das Einsetzen und Herausnehmen der EPROMs in das Gerät sollte gefahrlos, bei völlig abgeschalteten Spannungen, erfolgen können.

Aus dieser Zielsetzung heraus entstand das Gerät mp V2.01 das alle angesprochenen Forderungen, und einige mehr, erfüllt. Es basiert auf dem selben Prinzip wie das in der Zeitschrift MC 8/84 vorgestellte Programmiergerät von Joachim Frank.

## 2.1 Grundlagen der Programmierung

Die Programmierung eines EPROMs (elektrisch-programmierbarer-nur-Lesespeicher) läuft im Prinzip folgendermaßen ab:

An den Chip wird die Versorgungsspannung (5V / 6V) angeschlossen, die Programmierspannung (12.5V / 21V / 25V) wird eingespeist, die Adress- und Datensignale werden angelegt. Die Programmierung wird durchgeführt indem das EPROM mit einem Puls genau definierter Länge angesteuert wird. Nach der Programmierung werden die Daten sofort wieder ausgelesen und mit den Originaldaten verglichen. Die Programmierung eines Bytes ist nun abgeschlossen.

Die Pulsdauer kann je nach Programmialgorithmus zwischen 100µs und 50ms liegen. Ältere Algorithmen laufen meist mit 50ms Zykluszeit ab, die Programmierung gestaltet sich daher äußerst zeitaufwendig (2764 ca. 7min). Neuere hingegen verwenden meist eine Pulsdauer von 1ms. Der Programmiervorgang wird hierbei so oft wiederholt bis sich der Erfolg einstellt oder aber eine maximale Anzahl von Programmierversuchen überschritten ist. Ein Überprogrammierpuls definierter Länge schließt letztendlich den Vorgang ab (Programmierdauer 2764: min.ca. 35s).

Es ist in jedem Falle darauf zu achten, daß die vom Hersteller des EPROMs vorgeschriebenen Programmialgorithmen zur Anwendung kommen, da nur so die Datensicherheit gewährleistet ist. Im Technischen Anhang finden Sie eine Darstellung der wichtigsten Algorithmen.

### 3.1 Programmierlogik

Die Schnittstelle Rechner-Programmiergerät beschränkt sich auf 5 TTL-Signale: Out, In, Clock, Strobe und Alarm. Das Signal Alarm muß nicht unbedingt zum Rechner geführt werden, kann aber zur Steigerung des Komforts hilfreich sein. Aufgrund dieser geringen Anforderungen an die Schnittstelle sollte eine Verbindung des Gerätes mit jedem Rechner möglich sein, sei es über einen freien Port, die Centronics-Schnittstelle oder....

Die auf der Leitung In vom Rechner seriell ankommenden Daten werden in einem 32-Bit langen Schieberegister abgelegt und können mit Hilfe des Strobe Signals an dessen parallelen Ausgang geschaltet werden. Das Schieberegister enthält sämtliche zur Funktion des Gerätes notwendigen Adress-, Daten- und Steuerbits. Die Ausgänge Q1 bis Q6 von IC1 dienen zur Steuerung des Gerätes. Q1 bestimmt die Höhe der Programmierspannung. Q2+Q3 und Q4+Q5 werden jeweils einem 2 Bit-Binärdekoder zugeführt der die 'Informationsbreite' etwas erweitert. Ein Lo an Q6 versetzt die Ausgänge von IC4 in den disable Zustand. Dies muß immer dann geschehen wenn Daten mit Hilfe von IC5 einem Parallel- Seriellwandler ausgelesen werden.

S1 ermöglicht die Umleitung der Programmierspannung von Pin 23 nach Pin 24 der EPROM Fassung. Bei neueren Epromtypen sind im EPROM 2 Bytes an Information gespeichert, die Auskunft über Hersteller und Typ geben. Mit Hilfe dieser Angaben kann auf den richtigen Programmieralgorithmus geschlossen werden. Zum Auslesen der Information ist eine Spannung von 12V an Pin 24 des EPROMs notwendig. Diese Spannung wird über eine Diode aus der Programmierspannung 12.5V gewonnen. Wird der Schalter in diese Stellung gebracht, so hat dies allerdings einen Nachteil; Chips vom Typ 2716 können zwar korrekt ausgelesen, aber nicht mehr programmiert werden, da Sie die Programmierspannung an Pin 23 erwarten. Ein Ausweg wäre der, auf die Programmierung von 2716 Chips zu verzichten, statt dessen das kompatible 2732 oder 2732A EPROM zu verwenden und die Daten ab Adresse \$800 in das 2732 zu kopieren. Wie auch immer, der Schalter läßt Ihnen die Wahl. Sie sollten jedoch zu jeder Schalterstellung unbedingt die richtige Software verwenden, da sonst unweigerlich die falsche Spannung an das verkehrte IC-Beinchen gelangt.

Mit den Transistoren T6 bis T8 wird die Programmierspannung an das EPROM geschaltet. T9, T10 und bei 26-poligen EPROMs T12 sorgen für die richtige Speisespannung am Chip. Diese kann an Pin 28 wahlweise 5 oder 6V betragen. Die Transistoren T11 und T13 dienen als Treiber da an den Pins 1 und 23 bei einigen Betriebszuständen ein erhöhter Strombedarf besteht. Das High-Signal an diesen Pins darf außerdem nur unwesentlich von der Versorgungsspannung des EPROMs abweichen. Ein 'normales' TTL-Signal würde diesen Anforderungen nicht genügen. Die Zenerdioden sorgen, mit ihrem, gegenüber einer 1N4148, erhöhten Spannungsabfall in Durchlaßrichtung für die richtige Spannung an den Pins.

Im Technischen Anhang finden Sie Tabellen denen Sie die Funktion des Schieberegisters im einzelnen entnehmen können.

### 3.2 Schutzschaltung

Um das EPROM bei falscher Handhabung des Gerätes vor Beschädigung zu schützen, sei es durch verkehrtes Einsetzen in den Programmiersockel oder eine falsche Programmierspannung, ist das Gerät mit einer Schutzschaltung ausgestattet.

Steigt der Programmierstrom auf über 50mA an so wird T5 leitend. Mit diesem Signal wird, zeitverzögert durch R1 und C10, T1 und hierdurch ein, aus 2 Invertern bestehendes, bistabiles Flip-Flop angesprochen, das seinerseits den Enable-Eingang von IC6/1 und den Ausgang Alarm auf einen High-Pegel setzt. Die rote LED wird eingeschaltet. Das Rücksetzen der Schutzschaltung geschieht durch das Steuerprogramm.

Da die Ansprechgeschwindigkeit der Schaltung gedrosselt sein muß damit sie nicht auch bei jedem normalen Einschalten der Programmierspannung aktiviert wird, kann der Schutz den die Schaltung bietet kein Absoluter sein.

### 3.3 Spannungsversorgung

Die Spannungsversorgung des Gerätes ist sehr konventionell aufgebaut. IC9 dient zur Stabilisierung der 6V Versorgungsspannung. Aus ihr wird die Spannung 'Vcc 5V' über den Spannungsabfall an einer Diode gewonnen. Die Programmierspannung Vpp wird von IC10 auf den jeweiligen Wert festgelegt. IC9 und IC10 bedürfen der Kühlung. Die Zenerdioden D2 und D3 sind sofern der in der Stückliste aufgeführte Netztrafo verwendet wird nicht erforderlich. Sie dienen zum Schutz des IC10 vor zu hohen Eingangsspannungen. Eine grüne LED signalisiert den Zustand 'Gerät ein'.

### 4.1 Aufbau und Abgleich

Das Bestücken der einseitigen Europakarte sollte eigentlich keine Schwierigkeiten bereiten. Der Schalter S1 ist, um Störeinstrahlungen zu verhindern, direkt auf der Platine einzulöten. Die Anschlußbeinchen der Spannungsregler werden um 90° abgewinkelt. Auf ihre Rückseite werden nun die Kühlkörper aufgeklebt (Vorsicht, die Kühlkörper der Spannungsregler dürfen sich im eingelöteten Zustand nicht berühren). Die Transistoren T9, T10, T11 und T13 werden so eingelötet, daß sie gegenüber den anderen Bauteilen etwas erhaben sind. Nun können die Spannungsregler eingelötet werden. IC's sollten gesockelt werden. Falls das Gerät an einem AT-Rechner betrieben werden soll, so müssen bei den IC's vom Typ 4094 und 4021 unbedingt schnelle Typen z.B. der Serie HEF oder HCF Verwendung finden. Der Textool-Sockel zur Aufnahme des EPROMs kann auf der Bestückungs- oder Lötseite der Platine montiert werden. Hierzu befinden sich auf der Platine jeweils zwei Lochreihen. Er sollte, da es sich hierbei um ein Verschleißteil handelt, nicht direkt auf die Platine gelötet werden. Als Steckfassung kann ein 28-poliger LC-Sockel verwendet werden.

a) Montage des Textool-Sockels auf der Lötseite der Platine: Zugehörig sind die beiden den IC's 2 und 3 nähergelegenen



## 5.1 Technische Daten

Universelles vollständig programmgesteuertes EPROM-Programmiergerät mit eigener Spannungsversorgung und Anschlußmöglichkeit an jeden Rechner.

Programmierspannungen: 12,5V  
21,0V  
25,0V

schaltbar an Pin 1, 22, 23 oder 24(-0,7V)

Versorgungsspannungen: 5,2V (Pin 28)  
6,0V (Pin 28)  
5,2V (Pin 26)

Programmiert: 2716 bis 27512, 27513 und .....  
Microcontroller 8744/8751/8753 (Adaptersockel)

Schutzschaltung: Ansprechschwelle 50mA

Bestückung: 10 ICs, 13 Transistoren, 2 LED's

Netzteil: 220V AC

Leistungsaufnahme: ca. 5VA







Größe (LxBxH): 160x100x25mm



## 5.2 Funktionstabellen



- Einlesen von Daten in das 32-Bit Schieberegister:

Wahrheitstabelle 4094

Eingänge				Parallele Ausgänge		Serieller Ausgang
Clock'	Output Enable	Strobe	In	Q1	Qn	Qs
	L	X	X	OC	OC	Q7
	L	X	X	OC	OC	NC
	H	L	X	NC	NC	Q7
	H	H	L	L	Qn-1	Q7
	H	H	H	H	Qn-1	Q7
	H	H	H	NC	NC	NC

Auslesen der Daten vom 8-Bit Schieberegister:

Wahrheitstabelle 4021

Eingänge			Serieller Ausgang
Clock'	Strobe	Pn	Out
	L		Q7
	L		Q8
X	H	Qn=Pn	Q8

Clock' =  $\overline{\text{Clock}}$

L = Low  
 H = High  
 X = Don't Care  
 NC = No Change  
 OC = Open Circuit  
 Qn = Ser. Register Data  
 Pn = Parallel Data

- Bedeutung der Steuerbytes:

Byte 1	Socket (28 Dip)		Byte 2	Socket (28 Dip)
Q1	Pin 19		Q1	Pin 03
Q2	Pin 18		Q2	Pin 04
Q3	Pin 17		Q3	Pin 05
Q4	Pin 16		Q4	Pin 06
Q5	Pin 15		Q5	Pin 07
Q6	Pin 13		Q6	Pin 08
Q7	Pin 12		Q7	Pin 09
Q8	Pin 11		Q8	Pin 10

Byte 3	Socket (28 Dip)	\$	Byte 4	Socket (28 Dip)
Q1	Pin 01	80	Q1	H:12.5V Vpp/L:21V Vpp
Q2	Pin 27	40	Q2	siehe unten
Q3	Pin 26	20	Q3	siehe unten
Q4	Pin 02	10	Q4	siehe unten
Q5	Pin 23	8	Q5	siehe unten
Q6	Pin 21	4	Q6	H:Schreiben/L:Lesen
Q7	Pin 24	2	Q7	Pin 22
Q8	Pin 25	1	Q8	Pin 20

S1 und Q1,Q2,Q3,Q4,Q5 von Byte 4 bestimmen den Betriebsmodus:

	80	40	20	10	8	\$ (Hex)
S1	Q1	Q2	Q3	Q4	Q5	Bedeutung
B	H	H	H	X	L	Chip Identifier Mode (12V Pin 24)
A	X	H	H	X	X	Vpp Pin 23
X	X	H	L	X	X	Vpp Pin 1
X	X	L	H	X	X	Vpp Pin 22
X	X	L	L	X	X	Vpp Aus
X	L	X	X	H	H	25V Vpp
X	X	X	X	H	L	5V Vcc Pin 28
X	X	X	X	L	H	6V Vcc Pin 28
X	X	X	X	L	L	Reset Schutzschaltung/0V Pin28

H = High      L = Low      X = High oder Low

- Schalter S1 bestimmt den Betriebsmodus:

Stellung A: 2716 programmierbar,kein Chip Identifier Mode

Stellung B: Chip Identifier Mode; 2716 lesbar,jedoch nicht programmierbar

### 5.3 Programmieralgorithmen

- N - Normal:
- Vcc = 5V
  - Vpp = Ein
  - ((50ms Programmierdauer/Verify) / Byte) / Gesamtbereich
  - Vcc = Vpp = 5V
  - Verify Gesamtbereich
- A - Fast A:
- Vcc = 6V
  - Vpp = Ein
  - (((1ms Programmierdauer/Verify)bis ok oder 15x durchlaufen) / Byte) / Gesamtbereich
  - Vcc = Vpp = 5V
  - ((2ms Programmierdauer) / Byte) / Gesamtbereich
  - Verify Gesamtbereich
- B - Fast B:
- Vcc = 6V
  - Vpp = Ein
  - (((1ms Programmierdauer/Verify)bis ok oder 15x durchlaufen)/(Anzahl der Durchläufe x 4)ms Programmierdauer) / Byte) / Gesamtbereich
  - Vcc = Vpp = 5V
  - Verify Gesamtbereich
- C - Fast C:
- Vcc = 6V
  - Vpp = Ein
  - (((1ms Programmierdauer/Verify)bis ok oder 25x durchlaufen)/(Anzahl der Durchläufe x 3)ms Programmierdauer) / Byte) / Gesamtbereich
  - Vcc = Vpp = 5V
  - Verify Gesamtbereich
- D - Fast D:
- Vcc = 6V
  - Vpp = Ein
  - (((1ms Programmierdauer/Verify)bis ok oder 20x durchlaufen)/(Anzahl der Durchläufe x 1)ms Programmierdauer) / Byte) / Gesamtbereich
  - Vcc = Vpp = 5V
  - Verify Gesamtbereich
- E - Fast E:
- Vcc = 6V
  - Vpp = Ein
  - (((1ms Programmierdauer/Verify)bis ok oder 10x durchlaufen) / Byte) / Gesamtbereich
  - Vcc = Vpp = 5V
  - Verify Gesamtbereich

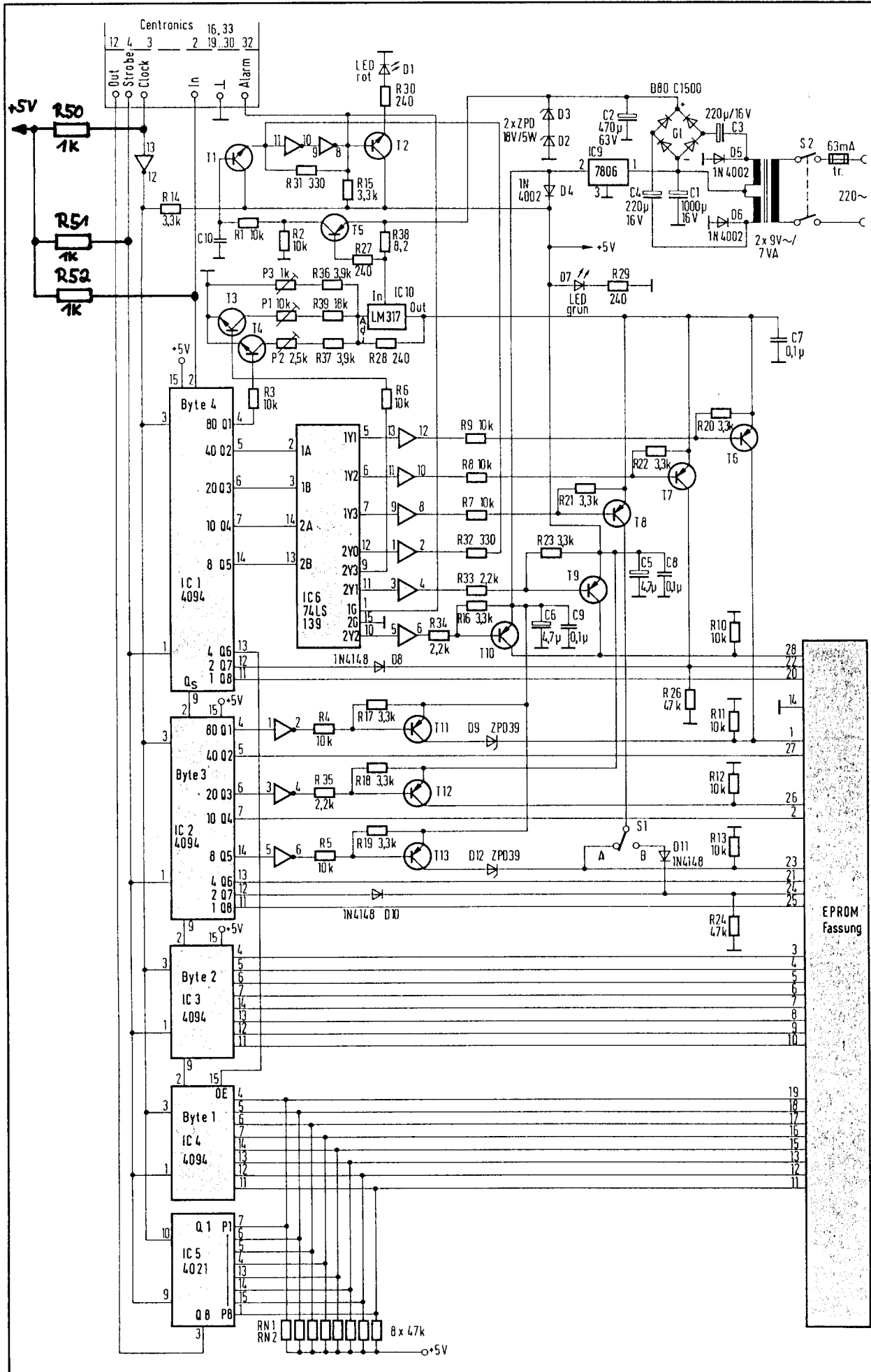
## 5.4 Stückliste

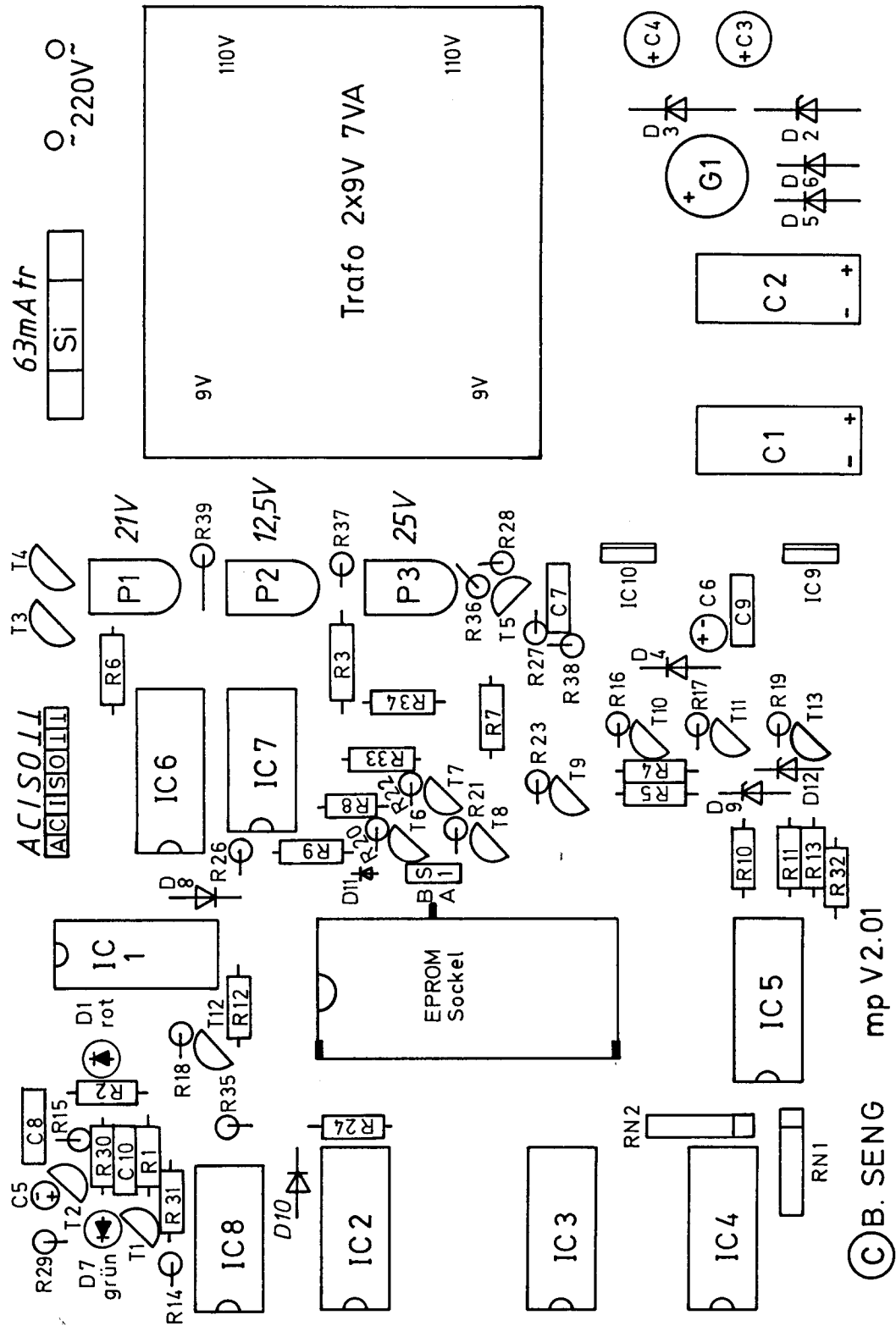
ICs:	4x HEF(HCF) 4094
	1x HEF(HCF) 4021
	1x 74LS139
	1x 7406
	1x 7407
	1x 7806
	1x LM 317T
Transistoren:	9x BC 327
	4x BC 546
Gleichrichter:	1x B80 C1500
Dioden:	3x 1N 4148
	3x 1N 4002
	2x ZPD 39V/0.3W
	1x LED rot/5mm
	1x LED grün/5mm
Kondensatoren:	1x 1000 $\mu$ F/16V/stehende Ausführung/RM 7.5
	1x 470 $\mu$ F/63V/stehende Ausführung/RM 7.5
	2x 220 $\mu$ F/16V/stehende Ausführung/RM 5
	2x 4.7 $\mu$ F/16V/stehende Ausführung/RM 2.5
	4X 100nF/keramisch/RM 5 oder 7.5
Widerstände:	1x 8.2 $\Omega$ /0.25W/5%
	4x 240 $\Omega$ /0.25W/5%
	2x 330 $\Omega$ /0.25W/5%
	3x 1k $\Omega$ /0.25W/5%
	3x 2.2k $\Omega$ /0.25W/5%
	10x 3.3k $\Omega$ /0.25W/5%
	2x 3.9k $\Omega$ /0.25W/5%
	13x 10k $\Omega$ /0.25W/5%
	1x 18k $\Omega$ /0.25W/5%
	10(2)x 47k $\Omega$ /0.25W/5%
	(2x 47k $\Omega$ /SIL4)
Potentiometer:	1x 1k $\Omega$ /RM 5x10/liegend
	1x 2.5k $\Omega$ /RM 5x10/liegend
	1x 10k $\Omega$ /RM 5x10/liegend
Sonstiges:	1x Trafo 2x110V/2x9V/7VA/Schaffer BV 222-1-01274
	1x Netzkabel/Euro
	1x Sicherung/63mAtr
	1x Sicherungshalter/2-teilig
	1x Netzschalter
	1x Jumper oder Schalter/1 x um/subminiatur
	6x Sockel/16-Dil
	2x Sockel/14-Dil
	1x Sockel/28-Dil/LC
	1x Sockel/28-Dil/3M-Textool 228-3345-00-0605
	1x Pfostensteckverbinder mit Kabel
	1x Buchse/Centronics 36-polig (oder Andere)
	2x Kühlkörper/IC-24-polig/13°C/W
	1x Knickschutztülle
	4x Blechschraube 2,9 x 6,5mm
	2x Mutter DIN 934-M3
	2x Schraube DIN 84-M3x8
	1x Gehäuse Kunststoff/Innenmaße min.160x100x25mm
	1x Platine mp V2.01

Dioden D2 und D3 im Bausatz nicht enthalten (siehe Punkt 3.3)

C1	1000 $\mu$ F/16V
C2	470 $\mu$ F/63V
C3,C4	220 $\mu$ F/16V
C5,C6	4,7 $\mu$ F/16V
C7...C10	100nF/keramisch
D1	LED rot
D2,D3	ZPD 18V/5W
D4,D5,D6	1N 4002
D7	LED grün
D8,D10,D11	1N 4148
D9,D12	ZPD 39V/0,3W
G1	B80 C1500
IC1...IC4	HEF(HCF) 4094
IC5	HEF(HCF) 4021
IC6	74LS139
IC7	7407
IC8	7406
IC9	7806
IC10	LM317
R1...R13	10k $\Omega$
R14...R23	3,3k $\Omega$
R24,R26	47k $\Omega$
R27...R30	240 $\Omega$
R31,R32	330 $\Omega$
R33...R35	2,2k $\Omega$
R36...R37	3,9k $\Omega$
R38	8,2 $\Omega$
R39	18k $\Omega$
R50...R52	1k $\Omega$
RN1,RN2	47k $\Omega$
Si	63mA/träge
S1	1 x um/subminiatur
S2	Netzschalter
T1...T4	BC546
T5...T13	BC327

# 5.5 Schaltbild





## Teil II

### Adaptersockel ma51 V2.0

- 1.1 Einleitung
- 1.2 Die Prozessoren der Familie 8051
- 1.3 Aufbau und Inbetriebnahme
- 1.4 Die Programmierung
- 1.5 Stückliste
- 1.6 Bestückungsplan
- 1.7 Schaltbild
- 1.8 Layout
- 1.9 Literaturverweise



## 1.1 Einleitung

Die Platine stellt eine Erweiterung zum Programmiergerät mp V2.xx dar. Sie ermöglicht das Programmieren des On-Chip-EPROM's der Microcontroller Intel 8744, Intel 8751, AMD 8751 und AMD 8753 im 40-poligen DIL-Gehäuse.

## 1.2 Die Prozessoren der Familie 8051

Die Microcontroller der Reihe 8051 stellen die Nachfolgenera-tion der Prozessoren der Reihe 8048 dar. Sowohl die Integra-tionsdichte als auch die Leistungsfähigkeit der Microcontroller wurde erheblich gesteigert. Aufgrund ihrer überragenden Eigen-schaften werden diese Prozessoren zur Lösung der unterschied-lichsten Aufgaben sehr häufig eingesetzt. Dies schlägt sich auch in einer breiten Produkt und Anbieterfamilie nieder. So sind Bausteine dieser Familie sowohl von Intel, AMD, Siemens, als auch Valvo erhältlich. Typen mit und ohne On-Chip-(EP)ROM, serieller Hochgeschwindigkeitsschnittstelle und AD-Wandler werden verpackt in den verschiedensten Gehäuseformen angeboten.

Die am häufigsten eingesetzten Typen mit dem breitesten Anwen-dungsspektrum dürften jedoch der 8031, 8051/8751, 8052/8752/8753 sein. Den Controllern gemeinsam sind folgende Leistungsmerkmale:

- 8 Bit CPU
- 128 Bytes RAM
- 4K Bytes ROM (8031 0K, 8052/8752/8753 8K Bytes)
- 2 interne 16-Bit Zähler
- Serieller Vollduplex-E/A-Port
- Betrieb als Boolescher Prozessor möglich
- Bit-adressierbares RAM
- On-Chip-Oszillator
- 64K Datenspeicher
- 64K Programmspeicher
- Multiplikations und Divisionsbefehle
- 2- bis 10-facher Datendurchsatz gegenüber 8048
- 5 Interruptquellen mit 2 Prioritätsstufen
- 32 E/A Leitungen

Die Prozessoren mit der Kennzeichnung 87XX besitzen ein ON-Chip-EPROM. Dieses kann durch das Programmieren eines Security-Bit gegen einen von ausen kommenden Lesezugriff geschützt werden. Somit eignen sich diese Prozessoren hervorragend für die Fer-tigung von Prototypen, Mustern und Kleinserien bei denen sich der Einsatz maskenprogrammierter Controller nicht lohnt und bei denen dennoch das Steuerprogramm nicht dem öffentlichen Zugriff überlassen werden soll.

## 1.3 Aufbau und Inbetriebnahme

Die Platine wird zuerst auf der Lötseite mit den beiden Pfosten-leisten ST1 und ST2 bestückt. Die Länge der einzelnen Pfosten sollte so abgestimmt sein, daß sie, wenn die Platine im Program-miergerät eingesetzt ist, nicht zu tief in den Nullkraftsockel

des Programmiergerätes hineinreichen, um eine Beschädigung des Sockels des Programmiergerätes zu vermeiden. Danach werden die anderen Teile auf der Bestückungsseite bestückt. Am Programmiergerät mp V2.xx sind keine Änderungen auszuführen.

#### 1.4 Die Programmierung

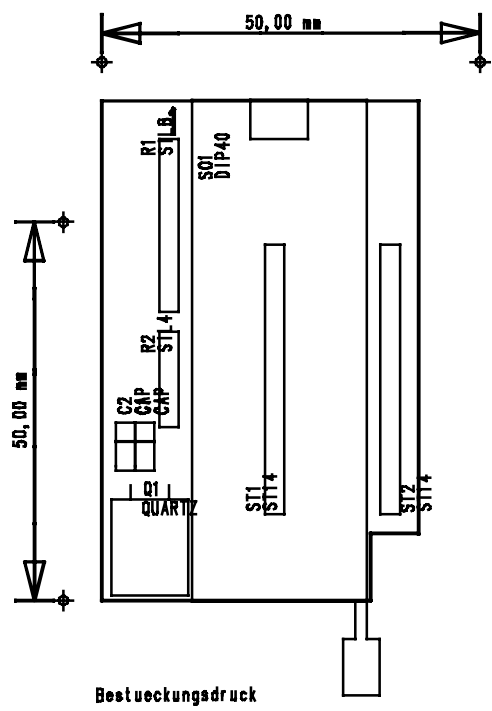
Der Ablauf entspricht dem bei der Programmierung eines normalen EPROM's nur, daß zuvor der Adaptersockel ma51 im Programmiergerät installiert werden muß. Als Steuerprogramm kann sowohl die Version mp V2.21 als auch die Version 3.0 Verwendung finden. Der Prozessortyp ist anhand der jeweiligen Typliste einzustellen. Nachdem der Baustein programmiert und verifiziert ist kann das Security-Bit gesetzt werden. Wird danach das EPROM des Microcontrollers noch einmal ausgelesen, darf nur noch der Wert FFh erscheinen. Somit kann ein gelöschter Controller nicht von einem bei dem das Security-Bit gesetzt ist unterschieden werden. Nur durch das Löschen des Controllers mit UV-Licht kann die Auslesesperre wieder zurückgesetzt werden.

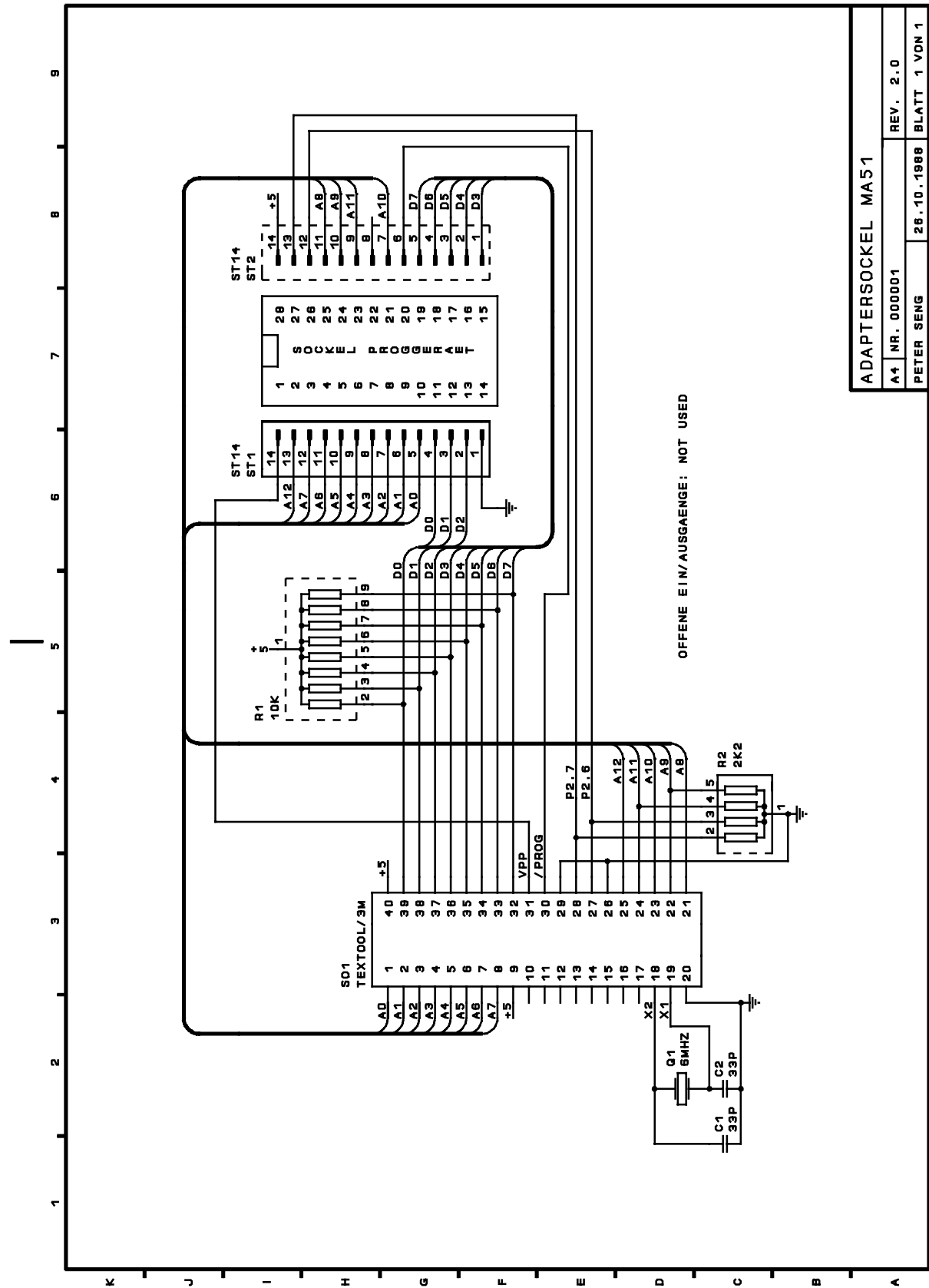
#### 1.5 Stückliste

- 2 x Kondensator/33 pF/keramischC1, C2
- 1 x Quartz/6MHz/200MILQ1
- 1 x Sockel/3M Textool 240-3346S01
- 2 x Stiftleiste/14-polig/100MILST1, ST2
- 1 x Widerstandsnetzwerk/SIL4-5-polig/2K2R2
- 1 x Widerstandsnetzwerk/SIL8-9-polig/10KR1

1.6 Bestückungsplan

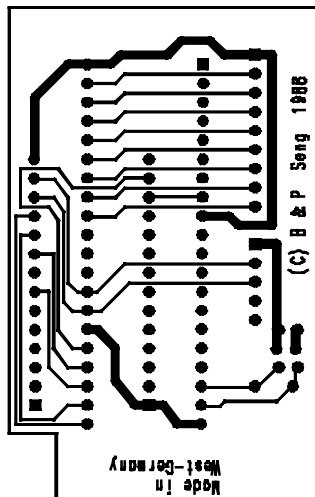
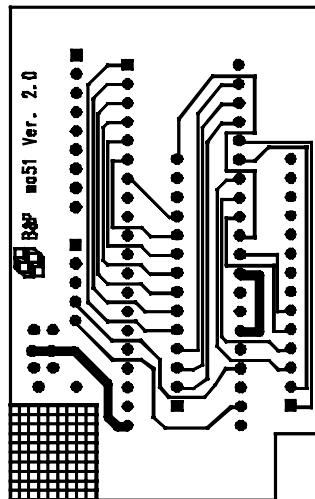
Maßstab 1:1





## 1.8 Layout

Maßstab 1:1



## 1.9 Literaturverweise

- Intel: Microcontrollers Handbook
- AMD: Microcontrollers, 1988 Data Book/Handbook
- Köhn/Schultes: 8051-Prozessoren, Franzis-Verlag, 1988

## Teil III

### Steuerprogramm mp V2.21

- 1.1 Einleitung
- 1.2 Verbindung Rechner-mp V2.01
- 1.3 Compilieren des Programmes
- 1.4 Start des Programmes
- 1.5 Das Hauptmenü
- 1.6 Allgemeine Hinweise
- 1.7 Die Kommandos
- 1.8 Makros
- 1.9 Die EPROM-Typen 27512,27513
- 2.0 EPROM-Typenverzeichnis
- 3.0 Optionen

## 1.1 Einleitung

Programm in Turbo-Pascal 3.0 für IBM PC/Kompatible.

Das hier beschriebene Steuerprogramm basiert auf dem in der Zeitschrift MC 3/86 vorgestellten Turbo-Pascal-Programm von Hans Jürgen Schmitt. Diese Überarbeitung wurde in vielen Punkten erweitert und bedienerfreundlich gestaltet.

Das Programm gestattet im Zusammenspiel mit dem Programmiergerät mp V2.01 das komfortable und schnelle Editieren/Programmieren (2764 Modus C: XT-1min 30s, AT-45s) der EPROM-Typen 2716 bis 27513 sowie der Microcontroller 8744/8751/8753. Es stehen hierbei 5 unterschiedliche Programmieralgorithmen zur Verfügung. Bei AT-Rechnern ist eine Anpassung der Software an die Rechengeschwindigkeit nicht notwendig. Das Programm kann jederzeit abgeändert und so den eigenen Bedürfnissen angepaßt werden.

Auf der Diskette befinden sich folgende Dateien:

- MP.PAS                    Quellcode
- MP\*.INC                  Quellcode
- MPA\_AT\_1.COM    für PC-AT, S1 in Stellung A, LPT1

Für andere als den genannten Rechner ist unbedingt eine Anpassung des Steuerprogrammes an den im Quellcode (MP.PAS) ersichtlichen Stellen notwendig. Aus Platzgründen sind auf der Diskette nur in eingeschränktem Maße compilierte Programmversionen enthalten. Der Quellcode enthält Routinen für XT-4,77MHz, Schneider/Amstrad-8086-PC und AT-Rechner.

## 1.2 Verbindung Rechner-mp V2.01

Die parallele Druckerschnittstelle des Rechners ist im Allgemeinen mit einer DB25S Buchse versehen. Rechner und mp V2.01 werden folgendermaßen verbunden.

Rechner	DB25S	Centronics	mp V2.01
PE -----	Pin 12 -----	Pin 12 -----	Out
D2 -----	Pin 04 -----	Pin 04 -----	Strobe
D1 -----	Pin 03 -----	Pin 03 -----	Clock
D0 -----	Pin 02 -----	Pin 02 -----	In
Error *****	Pin 15 *****	Pin 32 *****	Alarm
GND -----	Pin 18-25 -----	Pin 16,19-30,33 ---	GND

\* kann entfallen

Bei Verwendung einer Centronics-Buchse am Programmiergerät können Sie hierzu das Druckerkabel benutzen. Prüfen Sie Ihre Verbindungsleitung daraufhin ob das Kabel, besonders die Masseleitungen entsprechend belegt sind.

Die Leitung 'Alarm' wird vom Steuerprogramm nicht ausgewertet.

### 1.3 Compilieren des Programmes

Falls Sie das Programm neu compilieren, sei es um die Schnittstellenadresse zu ändern oder am Programm Abänderungen vorzunehmen, so beachten Sie bitte unbedingt die im Quelltext eingefügten Bemerkungen. Wird das Programm auf anderen Rechnern als den im Quelltext aufgeführten eingesetzt so sind sämtliche Zeitabläufe neu einzustellen.

### 1.4 Start des Programmes

Verbinden Sie den Rechner mit dem Programmiergerät. Booten sie wie gewohnt. Legen Sie die Programmdiskette in das angemeldete Laufwerk und starten sie das Programm mit der Eingabe von:

```
mp*_**_* <Return>
```

(Siehe Punkt 1.1, je nach Schalterstellung und Schnittstelle). Nach kurzer Zeit erscheinen abwechselungsweise folgende Meldungen:

```
* Bitte Gerät anschließen *
```

```
* <Return> Keine Hardware *
```

Schalten Sie nun das Gerät ein. Es erscheint die Meldung:

```
* Funktion ok *
```

Danach sehen Sie das Hauptmenü.

Falls dies nicht geschieht so sind die Port-Konstanten falsch eingestellt, die Verbindung vom Programmiergerät zum Rechner ist nicht in Ordnung, oder aber das Gerät ist defekt.

Wollen Sie das Programm benutzen ohne das Programmiergerät anzuschließen, so können Sie die, am Anfang des Programmes durchgeführte, Funktionsüberprüfung der Hardware durch Drücken der Taste <Return> umgehen. Ein akustisches Zeichen wird ausgegeben (Fehlermeldung). Es erscheint der Text:

```
* Kein Zugriff auf Programmiergerät möglich *
```

Hierauf sehen Sie das Hauptmenü.

### 1.5 Das Hauptmenü

Das Hauptmenü bietet folgende Informationen:

- Die Version und den Typ des laufenden Programmes
- Die Liste der möglichen Kommandos, gefolgt von jeweils einem Beispiel.
- Den aktuellen EPROM-Typ mit den wichtigsten Parametern.



## 1.6 Allgemeine Hinweise

- Zahleneingaben erfolgen hexadezimal.
- 'Speicherbereich' bezieht sich jeweils auf den RAM-Speicherbereich den der Rechner zur Verfügung stellt, nicht auf den jeweiligen EPROM-Speicherbereich.
- Der RAM-Speicherbereich erstreckt sich von \$0 bis \$7FFF.
- EPROM Routinen starten jeweils bei EPROM-Adresse \$00.

Beispiel: V 100 b 500

Der RAM-Speicherbereich \$100 bis \$500 wird mit dem EPROM-Speicherbereich \$0 bis \$400 verglichen.

- Nach Eingabe des Kommandos ist <Return> zu drücken.
- Kommandos können vor Eingabe von <Return> mit <Backspace> korrigiert werden.
- Die Kommandos Zeigen, Vergleichen, Programmieren und Auslesen können mit einer Standardvorgabe ausgeführt werden. Besteht das Kommando nur aus 2 Parametern (z.B. p 100) so wird die Standardvorgabe verwendet.

RAM-Speicherbereich = Anfangsadresse+EPROMgröße-1

- Falls die Überstromsicherung angesprochen hat (rote LED brennt), so ist entweder das EPROM defekt, falsch im Sockel platziert oder Sie haben den falschen Typ angewählt. Gehen Sie zurück ins Hauptmenü, berichtigen Sie den Fehler und unternehmen Sie einen erneuten Versuch.
- Kommandos können in Groß- und Kleinschrift eingegeben werden.
- Zwischen den einzelnen Parametern eines Kommandos muß ein Leerzeichen stehen.
- Auftretende Fehler werden durch eine Fehlermeldung angezeigt. Nach korrekt ausgeführten Kommandos erfolgt keine Meldung, es wird direkt ins Hauptmenü zurückgesprungen.

## 1.7 Die Kommandos

Das Kommando <I> ist nur in Programmversionen enthalten die vor dem Compilieren entsprechend den im Quelltext eingefügten Bemerkungen abgeändert wurden.

### <Z> Zeige/Ändere Speicherbereich

Der Speicherbereich wird in Blöcken zu jeweils 256 Bytes dargestellt. Das Editieren ist im ASC- und Hex-Code möglich.

Beispiel: z 2f B 200

Der Speicherbereich von \$2F bis \$200 kann betrachtet/editiert werden.

### <F> Fülle Speicherbereich mit Hex-Wert

Beispiel: F 0D0a b 300A m fA

Der Speicherbereich von \$0D0A bis 300A wird mit dem Hex-Wert FA gefüllt.

### <K> Löschkontrolle EPROM

Das gesamte EPROM wird auf vollständige Löschung untersucht

### <P> Programmiere EPROM mit Speicherbereich

Beispiel: P 200 b 3Ff

Das EPROM wird von Adresse \$0 bis \$1FF mit dem Inhalt des Speicherbereichs von \$200 bis \$3FF programmiert.

### <V> Vergleiche EPROM mit Speicherbereich

Beispiel: v 100 B 5eA

Der EPROM Inhalt von \$0 bis \$4EA wird mit dem Speicherbereich \$100 bis \$5EA verglichen.

### <B> Brenne MC Security-Bit

Setzen der Auslesesperre bei Microcontrollern Reihe 8051

### <A> Lese EPROM aus nach Speicherbereich

Beispiel: a 10 b 6aF

Der EPROM Inhalt von \$0 bis \$69F wird in den Speicherbereich \$10 bis \$6AF eingelesen.

### <L> Lade Dateibereich nach Speicheradresse

Beispiel: L b:eva 100 b 300 n 2A

Der Dateibereich \$100 bis \$300 der Diskettendatei EVA auf Laufwerk B: wird in den Speicherbereich \$2A bis \$22A kopiert.

- <S> Schreibe Datei von Speicherbereich  
Beispiel: s A:AdAM 100 b 600  
Der Speicherbereich \$100 bis \$600 wird auf Laufwerk A: als eine Diskettendatei mit Namen ADAM abgespeichert.
- <T> Typwahl  
Beispiel: T 40 Das EPROM mit der Typnummer 40 wird aktuelles EPROM.  
Beispiel: T 70 2 Das EPROM mit der Typnummer 70 wird aktuelles EPROM. Aktuelle Seite ist Seite 2.
- <H> Hilfe Typnummer  
Das Verzeichnis ermöglicht die Wahl der richtigen Typnummer aufgrund der Kenndaten des Eproms (Beschreibung der Programmiermodi im Teil A - Hardware 5.3).
- <I> Identifiziere EPROM  
Ermöglicht das Auslesen des Hersteller- und Typencodes. Darf nur bei dafür geeigneten EPROMs Anwendung finden. Wir empfehlen den Betrieb des Gerätes in Schalterstellung S1=A, da auf den Bausteinen alle erforderlichen Daten aufgedruckt sind und ältere für das automatische Identifizieren nicht geeignete Bausteine durch die angelegte hohe Spannung zerstört werden können.
- <D> DOS-Aufruf  
Ermöglicht die Ausführung von DOS-Befehlen im laufenden Programm. Auf DOS-Ebene erlaubte Befehle sind zulässig (z.B. DIR , DEL , COPY).
- <O> Optionen  
Aufruf des Menüs 'Optionen'.
- <E> Ende  
Ermöglicht das Verlassen des Programmes.

## 1.8 Makros

Unter Makros sind Kommandofolgen zu verstehen die automatisch, entsprechend ihrer Reihenfolge abgearbeitet werden. Das Programm bietet die Möglichkeit Makros zu erstellen, auszuführen, zu laden und zu speichern. Diese Programmpunkte sind über den Programmpunkt Optionen zu erreichen. Die Form eines Makros ist kompatibel zu der einer Batch-Datei bzw. einer normalen Textdatei ohne Steuerzeichen. Somit können Makros sowohl im Programm selbst, als auch unter Zuhilfenahme eines Texteditors oder im DOS unter Zuhilfenahme des Befehles 'COPY CON:\*.MAK' erstellt werden. Die Namen der Makros besitzen die Form: '\*.MAK'. Ein Makro kann maximal 48 Zeilen umfassen, pro Zeile ist ein Befehl zulässig. Da

das Programm nicht in der Lage ist größere Datenmengen als 32 KByte gleichzeitig zu handhaben ist es praktisch sich für die Handhabung großer EPROM-Speicher Makros anzulegen.

#### 1.9 Die EPROM-Typen 27512,27513

27512: Dieser Typ ist organisiert zu 64K \* 8 Bit. Unter Typ Nummer 60 sind die Adressen \$0 bis \$7fff ansprechbar. Typ Nummer 61 ermöglicht den Zugriff auf die Adressen \$8000 bis \$FFFF. Gleiches gilt für die Typen 62/63, 64/65.

27513: Dieser Typ ist organisiert zu 4 Seiten à 16K \* 8 Bit. Gültige Seiten sind die Seitennummern 0 bis 3. Bei der Typeingabe sind 2 Parameter anzugeben: 'TYPNUMMER SEITE'. Beispiel: T 70 2 Typ 70 Seite 2 wird aktuelles EPROM.

Die Eprom-Typen 27512 und 27513 können über den Menüteil Optionen mit dem Makro 'Copy512' bzw. 'Copy513' auf einmal kopiert werden.

## 2.0 EPROM-Typenverzeichnis

AMD	Fujitsu	General Instruments
Am 2716 T 10	MBM 2716 T 10	27HC64 T 33
Am 2716B T 11	-----	27c64 T 33
-----	MBM 2732 T 20	-----
Am 2732 T 20	MBM 2732A T 21	27c128 T 43
Am 2732A T 21	MBM 27c32A T 21	-----
Am 2732B T 22	-----	27256 T 53
-----	MBM 2764 T 35	27c256 T 53
Am 2764 T 32	MBM 27c64 T 35	-----
Am 2764A T 31	-----	27c512 T60/T61
-----	MBM 27128	
Am 27128 T 42	MBM 27c128	
Am 27128A T 41	-----	
-----	MBM 27256 T 53	
Am 27256 T 51	MBM 27c256 T 56	
Am 27c256 T 51	MBM 27c256A T 53	
-----	-----	
Am 27512 T62/63	MBM 27c512 T60/61	
Am 27c512 T62/63		
Hitachi	Hyundai	Intel
HN 462716 T 10	HN 27c64 T 31	i 2716 T 10
-----		-----
HN 462732 T 20		i 2732A T 21
HN 482732A T 21		-----
HN4627c32 T 20		i 2764 T 32
HN4827c32A T 21		i 2764A T 33
-----		i 27c64 T 33
HN 482764G T 32		-----
HN 482764P T 32		i 27128 T 42
HN 27c64G T 32		i 27128A T 43
HN4827c64 T 32		i 27128B T 43
-----		-----
HN 27128A T 43		i 27256 T 53
HN4827128 T 42		i 27c256 T 53
-----		-----
HN 27256 T 53		
HN 27c256 T 53		
-----		
HN 27512 T60/61		i 27512 T60/61
		-----
		i 27513 T 70

Mitsubishi	National	NEC
2716 T 10 ----- 2732 T 20 2732A T 21 ----- M5L 2764 T 32 ----- M5L 27128 T 42 M5M27c128 ----- M5L 27256 T 53 M5M27c256 T 53 ----- M5L 27512 T60/61	MM 2716 T 10 NMC 27c16 T 10 ----- MM 2732 T 20 NMC 27c32 T 20 ----- MM 2764 T 32 NMC 27c64 T 36 ----- NMC 27c128 ----- NMC 27c256 T 57 ----- NMC 27c512 T64/65	μPD 2716 T 10 ----- μPD 2732 T 20 μPD 2732A T 21 ----- μPD 2764 T 32 μPD 27c64 T 32 ----- μPD 27128D T 42 μPD 27128C T 42 ----- μPD 27256D T 56 μPD27c256D T 56 μPD27c256C T 56 μPD 27256AD T 55 μPD27c256AD T 53 μPD27c256AC T 53 ----- μPD27c512C T60/61 μPD27c512D T60/61
O K I	S E E Q	S G S
MSM 2716 T 10 ----- MSM 2732 T 20 MSM 2732A T 21 ----- MSM 2764 T 32 ----- MSM 27128 T 42 MSM 27128A ----- MSM 27256 T 53 MSM27c256 T 53 MSM27c256A T 53	DE 2764 T 32 DM 2764 T 32 DQ 2764 T 33 DQ 5133 T 33 ----- DE 27128 T 42 DM 27128 T 42 DQ 27128 T 43 DQ 5143 T 43 ----- DM 27256 T 53 DQ 27256 T 53 DM 27c256 T 53 DQ 27c256 T 53	M 2716 T 10 ----- M 2732 T 20 M 2732A T 21 ----- M 2764 T 32 M 2764A T 33 ----- M 27128A T 43 ----- M 27256 T 53 ----- M 27512 T60/61

T I	Thomson	Toshiba
TMS 2516 T 10 -----	ET 2716 T 10 ETC 2716 T 10 -----	TMM 323 T 10 -----
TMS 2732 T 20	ET 2732 T 20	TMM 2732 T 20
TMS 2732A T 21 -----	ETC 2732 T 20 -----	TMM 2732A T 21 -----
TMS 2764 T 32 -----	TS 27c64 T 33 -----	TMM 2764 T 32 TMM 2764D T 32 TMM 2764A T 33 -----
TMS 27128 T 42	TS 27c256 T 53	TMM 27128 T 42
TMS 27128A T 43		TMM 27128A T 43 -----
TMS 27c128 T 43		TC 57256
TMS 27c128A T 43 -----		TMM 27256 T 53 TMM 27256A T 53 TMM27c256 -----
TMS 27256 T 53		TMM 27512A T60/61
TMS 27c256 T 53		
TMS 27C256A T 53 -----		
TMS 27c512 T60/61		
TMS 27c512AT60/61		
Valvo		
27C64 T 33		

Da die Daten der Vergleichstabelle Herstellerdatenbüchern entstammen, können wir für die Vollständigkeit und Richtigkeit keine Gewähr übernehmen.

### 3.0 Optionen

- Steuerprogramm mp V3.x (Auto-Setup, 16 Bit, diverse Datenformate, Window-Technik)

## Teil IV

### EPROM-Emulator me V2.1

- I Kurzbeschreibung
- 1. Allgemeines
  - 1.1 Hinweise
- 2. Gerätebeschreibung
  - 2.1 Bedienungselemente
  - 2.2 Bedienungshinweise
  - 2.3 Service-Wartung-Abgleich
  - 2.4 Hinweise für den Geräteaufbau
- 3. Steuerprogramme
  - 3.1 Download
  - 3.2 Downhex
  - 3.3 mp V3.x
  - 3.4 Split
  - 3.5 RI
  - 3.6 Exchange
- 4. Technische Daten
  - 4.1 Lieferumfang
  - 4.2 Stückliste
- 5. Technischer Anhang
  - Schaltbild
  - Kabelverbindungen
  - Timing-Diagramme
  - Tabellen
  - Funktionsgleichungen PLD GAL16V8 Typ 002B
  - Bestückungsplan Bestückungsseite
  - Bestückungsplan Lötseite



## I Kurzbeschreibung

- a). Verkabelung (Betrieb an mp Vx.x, Markierung = Pin 1):
- Centronics Buchse PC-80x86 <--> Druckerlabel <--> mp Vx.x
  - mp Vx.x <--> Flachbandkabel A <--> me V2.1 'Programmer'
  - me V2.1 'Probe' --> Flachbandkabel B --> Zielschaltung

- b). Schalterstellungen:

		S1	S2/1	S2/2	S2/3	S2/4	S2/5	S2/6	S2/7	S3	S4
27513	R	ON	ON	ON	OFF	OFF	ON	OFF	R	A	A
27512	d	ON	ON	ON	ON	ON	OFF	OFF	S	R	R
27256	W	ON	ON	ON	ON	OFF	OFF	OFF	I	T	I
27128	O	ON	ON	ON	OFF	OFF	OFF	OFF	N	I	N
2704	V	ON	ON	OFF	OFF	OFF	OFF	OFF	I	F	I
2732	Y	ON	OFF	OFF	OFF	OFF	OFF	ON	O	N	O
2716	N	OFF	OFF	OFF	OFF	OFF	OFF	ON	N	I	I

RST: Reset aktiv high  
 /RST: Reset aktiv lo  
 ADR A: Geräteadresse = A  
 ADR B: Geräteadresse = B  
 S1: Mode-Schalter

Dip-Schalter S1, S2/x, S3, S4

Mode-Schalter:	Read Write Verify	Read-Only
	Normalmodus	Akkupufferung ein, Stromversorgung aus der Zielschaltung

- c). Steuerprogramme

typ = 16, 32, 64, 128, 256, 512, 5130, 5131, 5132, 5133  
 muß dem an S2/x eingestellten EPROMtyp entsprechen  
 p = Programmieren (Datei in Emulator kopieren)  
 v = Verify (Emulatorinhalt mit Datei vergleichen)  
 a = Emulator A  
 b = Emulator B  
 r = Reset  
 o = Offsetadresse EPROM - Emulator (Hex)  
 s = Startadresse Datei (Hex)  
 e = Endadresse Datei (Hex)

Download: ---> Kopieren einer Binärdatei in den Emulator  
 DOWNLOAD filename /typ [/p][v][a|b][r][sxxxx][exxxx][oxxxx]  
 Beispiel: DOWNLOAD neubios.dat /512 /p /v /a

Downhex: ---> Kopieren einer Intel-Hex-Datei in den Emulator  
 DOWNHEX filename /typ [/p][v][a|b][r]  
 Beispiel: DOWNHEX neubios.hex /256 /p /v /b

mp V3.x: ---> Kopieren einer Binärdatei in den Emulator.  
 Wahl der Geräteadresse im Typwahl-Menü  
 me - Adr A ==> Emulatoradresse A (Schalter S4)  
 me - Adr B ==> Emulatoradresse B (Schalter S4)  
 Sonstige Handhabung wie EPROM

- d). Sonstiges

Das Vergleichen des Emulatorinhalts mit einer Datei ist bei Verwendung des Programmiergerätes mp V2.01 nur bei der oben angegebenen Verkabelung möglich.

## 1. Allgemeines

Das Gerät gibt dem Entwickler die Möglichkeit Änderungen in EPROM's (Elektrisch-Programmierbaren-Nur-Lese-Speichern) schnell und effizient zu realisieren. Oftmals ist bei der Änderung des Inhalts eines EPROM's dieses mehrmals aus dem Sockel zu nehmen, zu Löschen erneut zu Programmieren und wieder in die Schaltung einzusetzen. Dieser Vorgang ist nicht nur äußerst zeitaufwendig, er führt auch sehr schnell zu Abnützungerscheinungen am Verbindungselement Platine-EPROM, dem Sockel.

Mit der Verwendung eines EPROM Emulator's lassen sich diese Arbeiten wesentlich komfortabler und wirtschaftlicher erledigen. Es handelt sich hierbei im Prinzip um ein zeitweises Ersetzen des EPROM's durch ein RAM (Schreib-Lese-Speicher) während der Entwicklungsphase. Der EPROM-Emulator wird über ein Flachbandkabel mit der Zielschaltung verbunden. Der Stecker am Flachbandkabel wird in den Sockel des zu ersetzenden EPROM's gesteckt. Das RAM des Emulators wird nun vom einem Rechner über zweites Kabel mit dem Inhalt des EPROM's geladen. Der Emulator verhält sich von der Zielschaltung aus gesehen wie ein EPROM.

### 1.1 Hinweise

Das Gerät sollte nach Entfernen der Verpackung auf Beschädigung hin überprüft werden. Falls ein Transportschaden vorliegt so darf das Gerät nicht in Betrieb genommen werden. Falls anzunehmen ist, daß ein gefahrloser Betrieb des Gerätes nicht mehr gewährleistet ist (nach Lagerung unter ungünstigen Verhältnissen z.B. in feuchten Räumen, bei sichtbarer Beschädigung, losen Teilen im Gerät, Funktionsstörungen usw.) so ist das Gerät außer Betrieb zu setzen und gegen Betrieb zu sichern. Das Gerät wurde nach der Herstellung einem ausgiebigen Funktionstest unterzogen. Die Garantieleistungen des Herstellers erlöschen falls am Gerät Änderungen vorgenommen werden oder das Gerät geöffnet wird. Transportschäden sind vom Garantieanspruch ausgenommen. Bei Beanstandungen ist das Gerät mit einer stichwortartigen Beschreibung des Fehlers an den Hersteller einzusenden. Die angegebenen Geräte-Spezifikationen sind unbedingt einzuhalten. Beim Öffnen oder Schließen des Gerätes muß das Gerät von allen Spannungsquellen getrennt sein. Falls eine Messung, ein Abgleich oder eine sonstige Arbeit am geöffneten Gerät unter Spannung unvermeidlich ist, so darf dies nur durch eine Fachkraft geschehen, die mit den damit verbundenen Gefahren vertraut ist.

## 2. Gerätebeschreibung

Der EPROM-Emulator me V2.1 stellt eine funktionale Ergänzung zu einem EPROM-Programmiergerät dar. Der Emulator ist an jedes Standard - Programmiergerät anschließbar, dessen Steuerprogramm den EPROM-Emulator mp V2.1 unterstützt. Der Emulator wird von allen Programmiergeräten der Serie mp Vx.x unterstützt. In dieser Beschreibung ist ferner sämtliche Information enthalten welche

zur Programmerstellung für Programmiergeräte anderer Hersteller benötigt wird.

Der Typumfang der emulierbaren EPROM-Typen umfasst die Typen 2716, 2732, 2764, 27128, 27256, 27512, 27513. Die Kombination EPROM-Programmiergerät mp Vx.x und EPROM-Emulator me V2.1 bildet ein für den Anwender und Programmentwickler einfach handhabbares, preiswertes und äußerst effizientes Entwicklungswerkzeug.

- Emulation der EPROM-Typen 2716-27512/513 mit nur einer Standard-Probe. Die Probe ist ca. 20cm lang und besitzt auf der einen Seite eine Buchsenstecker DIN41651/34-polig auf der anderen Seite einen 28-poligen Dil-Stecker sowie eine Reset-Probe (KLEPS). Die KLEPS kann im Zielsystem dessen EPROM emuliert werden soll an das Reset-System angeschlossen werden. Während Daten in den Emulator kopiert oder von diesem verifiziert werden ist diese Leitung der Probe aktiv. Der aktive Pegel der KLEPS ist mit Schalter S3 einstellbar. Ein automatischer Reset des Zielsystems ist somit ohne Aufwand machbar. Die Reset-Probe (Kleps) ist kurzschlussicher (max. 5V). Die Länge der Probe ist auf das Laufzeitverhalten optimiert. Eine Verlängerung der Probe wirkt sich bei Verwendung normaler Flachbandleitung durch eine Verschlechterung der 'Access-Time' aus. Die Länge der Probe ist für die praktische Handhabbarkeit des Gerätes jedoch von untergeordneter Bedeutung. Sollte Ihre Länge nicht ausreichen so kann der Emulator bedingt durch seine Kunststoffkapselung auf die Zielschaltung bzw. in das Zielgerät gelegt werden. Eine längere Probe ist mit Standardteilen leicht herstellbar, jedoch ist die Vergrößerung der 'Access-Time' zu beachten. Spezial-Probes sind auf Anfrage erhältlich. Die Bestimmung der 'Access-Time' in Abhängigkeit von der Probe bedarf spezieller Meßgeräte. Sollten bei der Emulation 24-poliger EPROM's Platzprobleme am 28-poligen Dil-Stecker der Probe auftreten so können auf den Stecker 24-polige Präzisions-Dil-Sockel zur Distanzierung aufgesteckt werden. Die Markierung der Probe kennzeichnet die Lage von Pin 1 bei 28-poligen EPROM's. Bei der Emulation 24-poliger Typen ist zu beachten, daß die Probe so eingesteckt wird, daß die der Kennzeichnung gegenüberliegende Seite bündig mit der EPROM-Fassung ist.
- Die Verbindung Programmiergerät <---> me V2.1 geschieht über ein ca 70cm langes Flachbandkabel. Die Länge dieses Kabels ist relativ unkritisch. Es besitzt auf der einen Seite einen 28-poligen Dil-Stecker und auf der Anderen einen Buchsenstecker DIN 41651/34-polig (2. Buchsenstecker ST5 optional). Die Markierung am Flachbandkabel kennzeichnet Pin 1. Der 28-polige Dil-Stecker am Flachbandkabel wird in den Dil-Sockel am Programmiergerät gesteckt. Beachten Sie hierbei die Kennzeichnung am Programmiergerät. An einem Programmiergerät können gleichzeitig zwei Emulatoren me V2.1 betrieben werden. Somit ist mit zwei Emulatoren und einem Programmiergerät eine Emulation 2 \* 8-Bit breit (Higher-Lower-Byte) möglich. Bei Verwendung zweier Emulatoren an einem Programmiergerät ist mit Schalter S4 die Adresse des ersten Emulators auf A, die des zweiten auf B einzustellen. Wird nur ein Emulator verwendet so kann der Schalter S4 sich in beliebiger Stellung befinden. Die Schalterstellung muß mit den Angaben bei der Handhabung der

Steuerprogramme übereinstimmen, da sonst der Emulator unter der falschen Geräteadresse angesprochen und somit nicht korrekt bedient wird.

- Die Betriebsarten Read/Write/Verify und Read-Only werden über den Schalter 'Mode' eingestellt.  
Die Betriebsart 'Read/Write/Verify' ist die normale Standard-Betriebsart. Die Stromversorgung des Emulators erfolgt über das Programmiergerät. Der Emulator wird über das Programmiergerät vom Rechner ferngesteuert, er verhält sich als 'Slave' des Rechners. Während eine Datei in den Emulator kopiert oder von diesem verifiziert wird ist die Reset-Probe des Emulators aktiv (Pegel mit Schalter S3 einstellbar).  
Die Betriebsart 'Read-Only' ist eine Betriebsart welche bei konventionellen Emulatoren nicht verfügbar ist. Sie darf erst aktiviert werden nachdem eine Datei in den Emulator kopiert worden ist (durch Umstellung des Schalters 'Mode'), da der Emulator in dieser Betriebsart nicht fernsteuerbar ist. Der Emulator verhält sich danach vollständig wie ein Eprom. Die Stromversorgung erfolgt nun aus der Zielschaltung. Der Emulator kann vom Programmiergerät getrennt werden; hierbei bleibt der momentane Inhalt des Emulators aufgrund einer in dieser Betriebsart bestehenden Akkupufferung des internen RAM's erhalten. Der Akku wird während des aktiven Betriebes (unabhängig von Schalter 'Mode') nachgeladen. Der Inhalt des Emulator's ist in dieser Betriebsart über das Programmiergerät nicht verifizierbar. Beim Zurückstellen des Schalters 'Mode' von 'Read-Only' auf 'Read/Write/Verify' kann der Inhalt des Emulators teilweise unkontrolliert überschrieben werden. Es ist deshalb nach einem Rückstellen des Schalters kein sicherer Betrieb des Emulators möglich. Sie sollten in diesem Falle die Datei auf jeden Fall erneut in den Emulator kopieren. Somit ist es nicht möglich den Inhalt des Emulators nach einem Rückstellen des Schalters mit Sicherheit zu verifizieren; dies sollte falls gewünscht vor dem Umstellen des Schalters 'Mode' in die Stellung 'Read-Only' geschehen. Beim Umstellen des Schalters 'Mode' aus der Stellung 'Read/Write/Verify' in die Stellung 'Read-Only' sind diese unkontrollierbaren Zustandsänderungen mit Sicherheit ausgeschlossen.

## 2.1 Bedienungselemente

- a). Stiftleiste 'Programmer'  
Nimmt das Flachbandkabel auf, welches die Verbindung zum Programmiergerät herstellt. Die Verbindung ist verpolungssicher. Die Markierung am Flachbandkabel kennzeichnet Pin 1. Der 28-polige Dil-Stecker am Flachbandkabel wird in den Dil-Sockel am Programmiergerät gesteckt. Beachten Sie hierbei die Kennzeichnung am Programmiergerät.
- b). Stiftleiste 'Probe'  
Nimmt die Probe auf, welche die Verbindung zum Zielsystem herstellt. Die Verbindung ist verpolungssicher. Die Markierung am Flachbandkabel kennzeichnet Pin 1. Der 28-polige Dil-Stecker am Flachbandkabel wird mit dem Zielsystem verbunden. Die Reset-Probe (Kleps) ist kurzschlussicher (max. 5V), Ihr

Pegel über S3 einstellbar. Muß die Probe in einen 24-poligen Sockel adaptiert werden, so können falls Platzprobleme bestehen 24-polige Präzisions-Dil-Sockel zur Distanzierung verwendet werden.

c). Dil-Schalter S2/x

Mit den Schaltern wird der zu emulierende EPROM-Typ eingestellt.

S2/x	1	2	3	4	5	6	7
2716	off	off	off	off	off	off	on
2732	on	off	off	off	off	off	on
2764	on	on	off	off	off	off	off
27128	on	on	on	off	off	off	off
27256	on	on	on	on	off	off	off
27512	on	on	on	on	on	off	off
27513	on	on	on	off	off	on	off

d). Schalter S3

Mit dem Schalter wird der aktive Pegel der Reset-Probe eingestellt.

Pegel	S3
Aktiv Lo	off
Aktiv High	on

e). Schalter S4

Mit dem Schalter wird die Emulator-Adresse eingestellt.

Adresse	S4
A	off
B	on

f). Schalter S1, 'Mode'

Mit dem Schalter wird der Betriebsmodus eingestellt.

Mode	S1
Read, Write, Verify	off
Read-Only	on

Bedeutung der Schalterstellungen siehe Abschnitt 2.

## 2.2 Bedienungshinweise

- Verbinden Sie das Programmiergerät mit dem Rechner laut Anleitung zum Programmiergerät.
- Verbinden Sie nun Programmiergerät, Emulator und Zielsystem entsprechend den oben gemachten Angaben.
- Wählen Sie mit Schalter S4 die Emulatoradresse unter der das Gerät angesprochen werden soll. Falls sie nur einen Emulator verwenden, so setzen Sie dieses Gerät auf die Adresse A, verwenden Sie zwei Emulatoren an einem Programmiergerät so stellen Sie ein Gerät auf Adresse A, das andere auf Adresse B.
- Stellen Sie am Schalter S2 den zu emulierenden EPROM-Typ ein.
- Am Schalter S3 wird der Reset-Pegel eingestellt. Dieser ist abhängig von der Art Ihres Zielsystems.
- Bringen Sie den Schalter 'Mode' in Position 'Read/Write/Verify'.
- Starten Sie jetzt das für Ihre Zwecke am besten geeignete Steuerprogramm, sämtliche Voreinstellungen sind nun getroffen.
- Handhabung der Steuerprogramme siehe Abschnitt 3.x.

## 2.3 Service-Wartung

Das Gerät besitzt keine zu wartenden oder abzugleichenden Elemente. Das Gehäuse besteht aus zwei mit 4 Schrauben verbundenen Halbschalen. Die Schrauben befinden sich auf der Unterseite des Gehäuses.

Wurde das Gerät sehr lange nicht benutzt, oder ist absehbar, daß das Gerät in der Betriebsart 'Read-Only' in einem Zielsystem für lange Zeit eingesetzt wird, das die meiste Zeit stromlos ist, d.h., daß die Akkupufferung des Emulators aktiviert wird ohne, daß der Akku in ausreichendem Maße während des Betriebes nachgeladen wird, so ist der Akku vorher auf seine maximale Kapazität aufzuladen. Hierzu ist es ausreichend das Gerät im Modus 'Read/Write/Verify' ca. 14 Stunden zu betreiben. Bei normalem Betrieb ist ein Nachladen des Akkus jedoch nicht notwendig. Der Akku unterliegt einer gewissen Alterung. Sollte ein Austausch erforderlich werden, so ist der alte Akku der Sondermüllverwertung zuzuführen.

Die Flachbandkabel sowie die 28-poligen DIL-Steckverbinder unterliegen einem natürlichen Verschleiß. Ersatz ist auf Anfrage erhältlich, kann jedoch auch leicht selbst hergestellt werden, da bei der Entwicklung großer Wert darauf gelegt wurde Verschleißteile mit handelsüblichen Elementen zu realisieren.

## 2.4 Hinweise für den Geräteaufbau

Falls Sie Ihr Gerät als Bausatz erworben haben beachten Sie bitte die folgenden Hinweise.

Da die Platine doppelseitig mit Bauelementen bestückt wird ist hierbei eine bestimmte Reihenfolge einzuhalten.

- C14-C17
- C1-C7, falls diese nicht in die IC-Sockel integriert sind
- C8

- Fassungsleisten Z6, Z7 als Sockel für IC5
- C9, falls dieser nicht in den IC-Sockel integriert ist
- Sockel für IC6  
Beachten Sie, daß die IC-Sockel gegebenenfalls über die Kondensatoren bestückt werden. Bestücken Sie die Halbleiter am Schluß, vermeiden Sie unnötige Hitzezufuhr.
- alle anderen Teile, außer G1
- G1

### 3. Steuerprogramme

Im Lieferumfang sind mehrere Steuer- und Hilfsprogramme enthalten welche für verschiedene Anwendungszwecke optimiert sind. Die Programme sind auf Rechnern der Serie PC/XT/AT/PS2 lauffähig und mit Auto-Setup Prozeduren ausgerüstet; sie bedürfen deshalb keinerlei Anpassung an den verwendeten Rechner. Der Quellcode ist teilweise auf Anfrage verfügbar. Die Emulatoren werden entsprechend den oben gemachten Angaben mit Programmiergerät, Rechner und Zielsystem verkabelt.

#### 3.1 Download

Kopiert eine Binärdatei von Platte/Diskette in den Emulator.

```

typ = 16, 32, 64, 128, 256, 512, 5130, 5131, 5132, 5133
      muß dem an S2/x eingestellten EPROMtyp entsprechen
p    = Programmieren (Datei in Emulator kopieren)
v    = Verify (Emulatorinhalt mit Datei vergleichen)
a    = Emulator A
b    = Emulator B
r    = Reset (Emulator wird zurückgesetzt)
o    = Offsetadresse EPROM - Emulator (Hex)
s    = Startadresse Datei (Hex)
e    = Endadresse Datei (Hex)

```

Syntax: DOWNLOAD filename /typ [/p][/v][a|b][r][sxxxx][exxxx][oxxxx]

Beispiel: DOWNLOAD neubios.dat /512 /p /v /a  
Emuliert wir ein EPROM vom Typ 27512, der Emulator befindet sich auf Adresse A. Die Datei wird zuerst in den Emulator kopiert und anschließend verifiziert.

#### 3.2 Downhex

Kopiert eine Intel-Hex-Datei von Platte/Diskette in den Emulator. Die Adress-Umsetzung von Extended-Adressen erfolgt mod 64K, d.h. die obersten 4 Adressbits (MSB) der vollen 20-Bit-Adresse werden vernachlässigt (12345h --> 2345h).

```

typ = 16, 32, 64, 128, 256, 512, 5130, 5131, 5132, 5133
      muß dem an S2/x eingestellten EPROMtyp entsprechen
p    = Programmieren (Datei in Emulator kopieren)
v    = Verify (Emulatorinhalt mit Datei vergleichen)

```

a = Emulator A  
b = Emulator B  
r = Reset (Emulator wird zurückgesetzt)  
Syntax: DOWNHEX filename /typ [/p[/v[/a|b]]/r]  
Beispiel: DOWNHEX bios.hex /256 /p /r /b  
Emuliert wird ein EPROM vom Typ 27256, der Emulator befindet sich auf Adresse B. Der Emulator wird rückgesetzt, dann wird die Datei bios.hex in den Emulator kopiert.

### 3.3 mp V3.x

Ermöglicht die komfortable Handhabung des Emulators sowie des Programmiergerätes (kopieren von Dateien oder einzelnen Bytes in den Emulator, integrierter Editor, Window-Technik, Makroverarbeitung).

Wahl der Geräteadresse im Typwahl-Menü (mit den Cursortasten):  
me - Adr A ==> Emulatoradresse A (Schalter S4)  
me - Adr B ==> Emulatoradresse B (Schalter S4)  
Sonstige Handhabung wie EPROM.

### 3.4 Split

16-Bit Binärdateien werden nach Lower- und Higherbyte aufgetrennt. Die beiden outfile's werden mit der Extension .lo bzw. .hi gekennzeichnet.

Syntax: split infile outfile  
Beispiel: SPLIT test.bin test

### 3.5 RI

Wandelt eine Intel-Hex-Datei in das Binärformat um. Verarbeitet im Gegensatz zu 'Exchange' auch Extended-Adress-Record's welche oftmals zur Erzeugung von Code für 80x86 Prozessoren verwendet werden.

Syntax: ri infile.xxx outfile [/64k]  
Beispiel: ri muster.hex muster /64k

Outfile erhält als Extension die Segmentnummer. Zusätzlich wird ein DOC-File erstellt, welches die exakten Anfangs- und Endadressen der jeweiligen Dateien dokumentiert. Falls die Daten anschließend mit SPLIT aufgetrennt werden sollen, so ist darauf zu achten, daß die Startadresse gerade ist. Ist dies nicht der Fall so werden die Datenfiles .HI und .LO gerade vertauscht. Es empfiehlt sich deshalb RI mit dem Switch /64k zu starten; hierbei werden jeweils komplette 64KB Segmente geschrieben, das Problem mit der Startadresse entfällt somit.



### 3.6 Exchange

Konvertierungsprogramm zwischen diversen Datenformaten. Das gewünschte/vorhandene Format wird bei infile bzw. outfile mit der File-Extension angegeben.

```
*.hex  Intel-Hex-Format (kein Extended-Intel-Hex)
*.ms1  Motorola S1
*.ms2  Motorola S2
*.ms3  Motorola S3
*.tek  Tektronix
*.bin  Binärformat
```

Syntax: exchange infile.xxx outfile.xxx

Beispiel: exchange motor.bin motor.hex

## 4. Technische Daten

EPROM-Emulator zum Anschluß an Programmiergeräte.

Emulierbarer Typumfang:	2716-27512/27513
Zugriffszeit:	typ. 120ns
Maße (L*B*H):	128 * 70 * 30 mm
Länge der Probeleitung:	20cm
Reset-Probe:	Schaltbar aktiv high/lo, kurzschlussfest (5V)
Emulationsspeicher:	64KB S-RAM
Betriebsstrom:	max. ca. 70mA
Ruhestrom:	ca. 35mA
Bufferstrom:	ca. 5µA
Akku-Kapazität:	60mAh
Akku-Ladedauer:	max. 14 Stunden
Datenerhalt:	ca. 1 Jahr
Dateiformate:	Binär und Intel-Hex direkt, Motorola S1/S2/S3 und Tektronix über Konver- tierungsprogramme
Umgebungstemperatur:	0 bis 40°C (Betrieb) -10 bis 60°C (Lagerung)

### 4.1 Lieferumfang

```
1 * me V2.1
1 * Probe 20cm
1 * Verbindungskabel 70cm
1 * Handbuch me V2.1
1 * Diskette 5¼" oder 3½"
```

## 4.2 Stückliste

POS.	MENGE	ARTIKEL	REFERENZ
1	1	16V8 - GAL,GENERIC ARRAY LOGIC,25ns,002A.JED	IC7
2	2	43256 - STATIC RAM,32K*8,100ns,Lo-Standby-Current	IC5,IC6
3	1	74HCT04 - HCT HEX-INVERTER	IC1
4	1	74HCT245 - HCT OCTAL BUS TRANSCEIVER	IC2
5	5	74HCT541 - HCT OCTAL-BUS-LEITUNGSTREIBER	IC10,IC3,IC4,IC8,IC9
6	1	AKKU,2.4V,60mA,VARTA FL 2/60DK 2.4V	G1
7	1	BC327 - PNP Transistor	T1
8	2	BUCHSENSTECKER 34-POLIG DIN41651	ST4,(ST5 optional),ST6
9	2	DIL-STECKER f.FLACHBANDKABEL,28-POLIG	ST3,ST7
10	2	DIODE,1N5818,SCHOTTKY	D3,D7
11	4	DIODE,ZENER,5V1,300mW	D2,D4,D5,D6
12	1	DIODE,ZENER,7V5,300mW	D1
13	2	FASSUNGSLEISTE 14-polig 100mil	Z6,Z7
14	1	FLACHBANDKABEL 34-POLIG 100cm	Z2
15	1	FLACHBANDKABEL 34-POLIG 20cm	Z1
16	1	GEHAEUSE PACTEC HML COMPUTERGRAU 002A	Z4
17	1	IC-FASSUNG DIP 14	Z5
18	7	IC-FASSUNG MIT KONDENSATOR DIP 20,100n,10V,Z5U	C1,C2,C3,C4,C5,C6,C7
19	1	IC-FASSUNG MIT KONDENSATOR DIP 28,100n,10V,Z5U	C9
20	1	KLEMPRUEFSPITZE,ROT,ITT Pomona 4233-2	ST8
21	1	KONDENSATOR,100n,10V,Z5U	C8
22	5	KONDENSATOR,1n,10V,200mil,KERAMISCH	C12,C14,C15,C16,C17
23	1	KONDENSATOR,2n7,10V,KERAMISCH	C13
24	2	KONDENSATOR,ELKO,220µ,6V3,6.3x7mm,PANASONIC ECE-AOJK-221	C10,C11
25	1	LEITERPLATTE EPOXY 1.0mm ME V2.1 002A	Z3
26	2	SCHALTER,1*UM,SECME 1K2	S3,S4
27	1	SCHALTER,2*UM,KNITTER MFP 220	S1
28	1	SCHALTER,KODIER,7 FACH EIN,AMP 435640	S2
29	1	STIFTLEISTE,34-POLIG,DIN41651	ST2
30	1	STIFTLEISTE,34-POLIG,DIN41651,90 GRAD	ST1
31	3	WIDERSTAND,100R	R12,R13,R14
32	9	WIDERSTAND,10K	R10,R11,R16,R2,R4,R5,R6,R7,R8
33	7	WIDERSTAND,1K	R1,R15,R17,R18,R19,R3,R9
34	8	WIDERSTANDSNETZWERK,SIL8,9POLIG,10K	RN1,RN2,RN3,RN4,RN5,RN6,RN7,RN8