

EPROP+

Der PC-Megabit - EPROMmer

Bedienungsanleitung

***taskit* Rechnertechnik GmbH**

**Köpenicker Straße 145
D-10997 Berlin
Telefon (030) 611295-0
Fax (030) 611295-10
Internet <http://www.taskit.de>
Email info@taskit.de**

Alle Rechte an dieser Dokumentation und dem hierin beschriebenen Produkt verbleiben bei

***taskit* Rechnertechnik GmbH.**

Kein Teil davon darf ohne schriftliche Genehmigung in irgendeiner Form reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

Bei der Erstellung der Dokumentation wurde mit Sorgfalt vorgegangen. Selbstverständlich können Fehler trotzdem nicht vollständig ausgeschlossen werden, so daß weder das o.a. Unternehmen noch der Vertreiber für fehlerhafte Angaben, sowie daraus resultierende Fehlfunktion oder deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Waren-, Marken- und Firmennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Copyright © *taskit* Rechnertechnik GmbH, 1990-2004

Inhalt

EPROP - DER PC-MEGABIT-EPROMMER.....	4
DIE NOTWENDIGEN SCHRITTE ZUR INBETRIEBNAHME:.....	4
HARDWARE ANSCHLIESSEN.....	4
SOFTWARE INSTALLIEREN.....	4
<i>Installation unter DOS bzw. in der DOS-Eingabeaufforderung von Windows:</i>	4
<i>Installation unter Windows:</i>	5
START DER EPROP-SOFTWARE.....	5
BETRIEB DES EPROP.....	6
DIE DATEIEN EPROP.CFG UND EPROP.OPT.....	8
DIE EPROP-SOFTWARE.....	8
1. DATEI.....	8
<i>Datei - Bezugsdatei</i>	8
<i>Datei - Format</i>	9
<i>Datei - DOS</i>	9
<i>Datei - QUIT</i>	9
2. IC-AUSWAHL.....	10
3. AKTION.....	11
<i>Aktion - Leertest</i>	11
<i>Aktion - Auslesen</i>	11
<i>Aktion - Programmieren</i>	11
<i>Aktion - Verifizierung</i>	12
<i>Aktion - Löschen</i>	12
<i>Aktion - Progr. + Verify</i>	12
<i>Aktion - Zusatz</i>	12
4. OPTIONEN.....	13
<i>Optionen - Algorithmus</i>	13
<i>Optionen - Prog.Spg. (Programmierspannung)</i>	13
<i>Optionen - Busbreite</i>	13
<i>Optionen - Position</i>	14
<i>Optionen - Adressraum</i>	14
<i>Optionen - von</i>	14
<i>Optionen - bis</i>	14
<i>Optionen - Datei-Offset</i>	14
<i>Optionen - Byte-Swap</i>	15
<i>Optionen - Tastatur-Makros</i>	15
<i>Optionen - I/O-Adresse</i>	16
5. MONITOR.....	17
<i>Der Datei-Editor</i>	17
<i>Der IC-Editor</i>	17
<i>Checksumme IC</i>	17
<i>Checksumme Datei</i>	17
6. HILFE.....	18
DATEIFORMATE.....	19
1. S-RECORDS.....	19
2. INTELLEC 8/MDS FORMAT (INTEL-HEX).....	20
3. INTEL MCS-86 HEXADECIMAL OBJECT (INTEL-HEX86 FORMAT).....	20
4. TEKTRONIX-FORMAT.....	21
VARIABLEN IN EPROP.CFG.....	22
VARIABLEN IN EPROP.OPT.....	22
EPROP-ZUBEHÖR UND WEITERE PRODUKTE.....	23

EPROP - der PC-MegaBit-EPROMmer

Der EPROP+ ist ein Programmiergerät für die meisten 8- und 16-Bit EPROMs, EEPROMs und Flash-Memories. Das Gerät ist zum Anschluß an einen Personal Computer (IBM XT, AT und Kompatible) mit mindestens einer parallelen Schnittstelle bestimmt. Die Grundausstattung unterstützt ICs im DIL-/DIP-Gehäuse, es sind aber auch Adapter für PLCC- und LCC-Gehäuse erhältlich.

Die Software bietet eine komfortable Benutzerführung mit Maus- und Tastaturbedienung. Die Automatisierung von festgelegten Abläufen ist durch Tastaturmakros möglich. Eine eigene Kommandozeilen-Version der Software ist für Batch-Betrieb und die Einbindung in andere Programme vorgesehen.

Die notwendigen Schritte zur Inbetriebnahme:

Hardware anschließen

Das mitgelieferte Kabel (25-poliges Drucker-Verlängerungskabel 1:1, 1,8m) muß an eine parallele Schnittstelle des PCs angeschlossen werden. Dies kann bei eingeschaltetem Rechner geschehen. Schließen Sie bitte das Steckernetzteil (9 V Gleichspannung unstabilisiert, + Pol innen) **zuletzt** an die DC-Buchse des EPROP+ an.

Software installieren

Ab Version 4.5 der Software gibt es zwei Möglichkeiten, die Software auf Ihrem PC zu installieren: Die "klassische" Methode unter DOS und eine "Klickvariante" für Windows.

Installation unter DOS bzw. in der DOS-Eingabeaufforderung von Windows:

Auf der Festplatte kann zunächst ein Verzeichnis angelegt werden, in das die EPROP-Software installiert werden soll. Nennen Sie es z.B. „eprop“.

Die zum Lieferumfang zugehörige Diskette in das Diskettenlaufwerk (z.B. a:) einlegen. Auf der Diskette befinden sich 3 Dateien: „extract.ex_“, „install.bat“ und „setup.exe“.

Wenn die Software beispielsweise in das Verzeichnis 'c:\eprop' installiert werden soll, geben Sie am DOS-Prompt folgenden Text ein:

```
install c: \eprop
```

Es kann natürlich auch ein anderes Laufwerk und/oder ein anderes Verzeichnis angegeben werden. Wenn das angegebene Verzeichnis noch nicht existiert, wird es angelegt. Dabei ist jedoch zu beachten, dass der Verzeichnis-Pfad, in dem das EPROP-Verzeichnis letztlich liegen soll, bereits vorhanden sein muß.

Wichtig: zwischen : und \ muß ein Leerzeichen eingegeben werden, andernfalls schlägt die Installation fehl.

Nachdem Sie den Installationsbefehl mit „Enter“ abgeschlossen haben, wird die Software auf Ihrem ausgewählten Laufwerk, in dem von Ihnen erstellten Verzeichnis, installiert.

Nach der erfolgreichen Installation befinden Sie sich in diesem neuen Verzeichnis am DOS-Prompt.

Geben Sie folgende Zeile ein:

```
eprop.exe
```

und bestätigen Sie mit „Enter“.

Das Programm „eprop“ wird nun gestartet.

Installation unter Windows:

Öffnen Sie den Explorer oder das Arbeitsplatz-Symbol (Win 9.x/ME) bzw. den Datei-Manager (Win 3.x) und wählen Sie dort das Disketten-Laufwerk, in dem die Installations-Diskette liegt (meistens Laufwerk ‚a:‘) durch doppelklicken aus. In dem sich dann öffnenden Fenster bzw. Verzeichnis befindet sich die mit einem Eprop-Symbol versehene Datei 'SETUP.EXE'. Diese braucht nur durch einen Doppelplick gestartet werden. Soweit gewünscht, kann das voreingestellte Verzeichnis 'C:\EPROP' geändert werden. Mit einem Klick auf den 'UNZIP'-Button wird die Installation abgeschlossen.

Ab hier gelten die gleichen Bedingungen wie bei der Installation unter DOS. In beiden Fällen kann das Programm vom Dos-Prompt aus gestartet werden. Sie können aber auch auf der grafischen Oberfläche in das Verzeichnis wechseln und das Programm per Doppelklick starten. Oder sie legen sich eine Verknüpfung auf den Desktop oder in das Start-Menü.

Start der EPROP-Software

Die Software starten Sie durch den Aufruf des Hauptprogramms "EPROP.EXE". Das Programm sucht beim ersten Start an den vorhandenen parallelen Schnittstellen, ob die EPROP-Hardware zugreifbar ist. Falls diese nicht gefunden wird, gibt das Programm eine Meldung aus.

Sofern Sie diese Meldung erhalten:

```
Hardware kann nicht initialisiert werden!  
Bitte I/O-Adresse prüfen  
Weiter mit beliebiger Taste
```

handelt es sich zunächst **nicht** um eine Fehlermeldung. Bestätigen Sie die Meldung mit einer beliebigen Taste und das Menü des Programms „eprop“ erscheint.

Sie müssen nun die I/O-Adresse einstellen. Dazu wählen Sie den Menüpunkt „Optionen“ an. Es erscheint ein Auswahlmnenü. Der unterste Menüpunkt der Ihnen jetzt angeboten wird heißt „I/O-Adresse“. Wählen Sie diesen Punkt aus. Nun erscheint ein weiteres Menü, in dem Ihnen verschiedene I/O-Adressen vorgeschlagen werden. Da Ihr Gerät über einen LPT-Anschluß verfügt wählen Sie bitte nur Menü-Punkte mit „LPT-IF“ aus.

Standardmäßig sollte der erste Parallelport auf die Adresse 378h eingestellt sein und diese Adresse also auch als I/O-Adresse für den Eprop funktionieren.

Sollte es mit dieser Adresse nicht klappen (Sie erhalten nach der Anwahl dieses Menüpunktes erneut die Meldung „Hardware lässt sich nicht initialisieren“), sehen Sie bitte in den Unterlagen Ihres PC oder im BIOS nach, auf welche andere I/O-Adresse der LPT-Port eingestellt ist.

Weitere Hinweise finden Sie im Hauptmenü in der Hilfeanleitung unter dem Unterpunkt „Inhalt“.

Betrieb des EPROP

Um Beschädigungen am Baustein oder dem Gerät durch eine falsche Voreinstellung zu vermeiden, sollte der Baustein-Typ im Menüpunkt 'Baustein' ausgewählt werden, da das Gerät nicht erkennen kann, welcher Baustein-Typ eingesetzt wird.

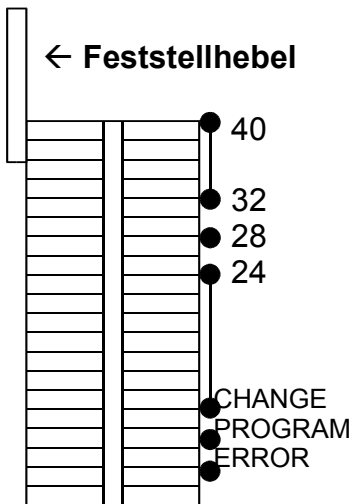
Wenn der konkret vorliegende Typ in der Liste nicht auftaucht, findet sich im Allgemeinen eine kompatible Einstellung (siehe hierzu auch Kapitel 2: IC-Auswahl).

Möglicherweise ist für die Bestimmung der Kompatibilität etwas Recherche bei den Bauteile-Herstellern notwendig (Datenblätter).

Bei UV-löschbaren Eeproms kann man fast immer davon ausgehen, dass auf den ersten Blick exotisch erscheinende Typen zu einem Standard-Typ kompatibel sind.

Bei EEproms und Flashbausteinen ist die Typenvielfalt leider sehr viel größer und beinahe monatlich kommen neue Typen auf den Markt, die sich dann in irgendeinem Detail zu den etablierten Typen unterscheiden. Trotzdem zeigt sich auch hier, daß sich im breiten Markt nur eine geringe Anzahl von Typen für lange Zeit durchsetzen, während viele andere Exoten bleiben.

Einsetzen von ICs in den EPROP-Sockel



Die Kerbe des Bausteins zeigt nach oben. (**PIN 1 links oben!**). Den Baustein **immer** von der Unterkante des Sockels aus gesehen einsetzen, d. h. je nach dem, wie groß der zu programmierende Baustein ist, reicht er bis zur 24-er, 28-er, 32er oder 40er-LED. Die Zahl an der LED muß die Pinzahl des Bausteins (ICs) übereinstimmen! Das Beispiel zeigt ein „27512“ (28polig).

Bitte beachten Sie, daß Sie beim Einsetzen stets nach unten bündig das IC einsetzen (nicht oben bündig, wo sich der Feststellhebel befindet). Die richtige Position eines ICs im Sockel ergibt sich aus der polzahlzugehörigen LED auf dem Gehäuse. Außerdem leuchtet vor jeder Aktion des EPROP eine grüne LED in Höhe der EPROM-Oberkante auf. Das IC sollte nur dann in den Sockel eingesetzt werden, wenn die Software eine entsprechende Bildschirm-Meldung ausgibt. Anschließend wird der Feststellhebel des Sockels nach unten umgelegt.

Die EPROP-Software ist weitgehend selbstdokumentierend. Zusätzliche Informationen erhalten Sie durch die **Hilfe-Funktion**, die durch Drücken der F1-Taste oder den entsprechenden Menüpunkt aufgerufen wird.

Die Dateien EPROP.CFG und EPROP.OPT

Beim ersten Start von EPROP wird im Verzeichnis von EPROP.EXE eine Datei mit dem Namen EPROP.CFG erstellt. Diese Datei enthält eine Standard-Konfiguration des Programms. In dieser Datei sind globale Programm-Variablen abgelegt, die beim Start eingelesen werden. Die Datei kann mit einem Texteditor bearbeitet werden. So können z.B. Farben und die Tonlänge des Beepers verändert werden. Dabei muß das genaue Zeilenformat gewahrt bleiben, insbesondere muß der Doppelpunkt als letztes Zeichen in Spalte 78 stehen bleiben. (Siehe auch Aufstellung "Variablen in EPROP.CFG und EPROP.OPT" am Ende dieser Anleitung.)

Die im EPROP-Menü zuletzt gewählten Einstellungen werden in der Datei EPROP.OPT im aktuellen Verzeichnis festgehalten und beim nächsten Start wieder verwendet.

Die EPROP-Software

Dieses Programm enthält eine komfortable, SAA-ähnliche Benutzer-Oberfläche für Maus- oder Tastaturbedienung. Das Grundmenü der Software enthält die Punkte:

1. Datei
2. Baustein
3. Aktion
4. Option
5. Monitor
6. Hilfe

Die Anwahl eines dieser Menüpunkte aktiviert ein Pull-Down-Menü mit weiteren Auswahlmöglichkeiten.

1. Datei

Das Datei-Menü enthält die grundlegenden Funktionen zum Laden einer Datei und zur Einstellung des Dateiformats. Es besteht aus den Unterpunkten:

- Bezugsdatei
- Format
- DOS
- Quit

Datei - Bezugsdatei

Die Bezugsdatei enthält die zu programmierenden oder zu verifizierenden Daten, bzw. speichert beim Auslesen eines Chips die ausgelesenen Daten in ihr.

Der im Parameter-Fenster angegebene Pfad wird als Suchpfad beim Zugriff auf die Bezugsdatei verwendet. Die Bezugsdatei kann in unterschiedlichen Dateiformaten vorliegen bzw. erstellt werden.

Durch Anwahl des Menüpunkts "Monitor" kann der aktuelle Inhalt der angewählten Bezugsdatei angezeigt und editiert werden.

Der Name der zuletzt eingestellten Bezugsdatei wird in der Datei "EPROP.OPT" abgelegt, so daß er beim nächsten Start voreingestellt ist.

Das Verändern der Bezugsdatei setzt die Startadresse im CPU-Adreßraum auf den Wert 0 zurück.

Datei - Format

Es werden folgende Datei-Formate unterstützt:

Binary:	Binäres Format (1:1 Kopie des Baustein-Inhalts)
Intel-HEX80:	Intellec 8/MDS (Datei-Format der Firma Intel, 8-Bit CPUs)
Intel-HEX86:	MCS-86 (Datei-Format der Firma Intel, 16-Bit-CPUs)
S1-Record:	16-Bit-Adresse (Datei-Format der Firma Motorola, 8-Bit-CPUs)
S2-Record:	24-Bit-Adresse (Datei-Format der Firma Motorola)
S3-Record:	32-Bit-Adresse (Datei-Format der Firma Motorola)
Tektronix:	Datei-Format der Firma Tektronix

(siehe auch die Beschreibung der verschiedenen Formate am Ende dieser Anleitung).

Das aktuell angewählte Datei-Format wird im Parameterfenster angezeigt. In der Datei EPROP.OPT wird das zuletzt angewählte Datei-Format gesichert, so daß es beim nächsten Start wieder voreingestellt ist.

Datei - DOS

Mit Anwahl dieses Menüpunkts kehrt man zum DOS-Prompt zurück, ohne das EPROP-Programm endgültig zu verlassen. Sie haben damit die Möglichkeit, andere Programme gleichzeitig zu starten. Durch Auslagerung des größten Teils des EPROP-Programms ist der verfügbare Speicher fast so groß wie vor dem Start von EPROP.

Beendet wird die DOS-Shell mit dem Kommando "EXIT". Sie kehren dann ins Programm EPROP zurück.

Datei - QUIT

Beenden des EPROP-Programms.

Die eingestellten Programmparameter: Bezugsdatei, Verzeichnis, Datei-Format, Speicherbaustein usw. werden in der Datei EPROP.OPT im aktuellen Verzeichnis abgelegt.

Diese Datei wird beim nächsten Start von EPROP eingelesen.

2. IC-Auswahl

Im Grundmenüpunkt "Auswahl" wird der zu bearbeitende Baustein-Typ ausgewählt. Hierzu ist das Pulldown-Menü in die Punkte

- 8-Bit EPROMs
- 16-Bit EPROMs
- EEPROMs
- Flash-Memories

unterteilt. Nach Auswahl eines Punktes erscheint die Liste der vom EPROP unterstützten Typen.

Bei Auswahl eines IC-Typs werden die optionalen Parameter wie "Prog.Spg", "Algorithmus", "Busbreite", "Position", "von" und "bis" auf ihre Initialwerte zurückgesetzt.

Falls der verwendete Baustein nicht in der Liste verzeichnet ist: In vielen Fällen können kompatible Typen oder der Typ "diverse" ausgewählt werden. Es sollte sichergestellt sein, daß der Ersatztyp die gleiche Pin-Belegung und die gleiche Programmierspannung verwendet. Bei EPROMs sollte im Zweifel immer zuerst die niedrigste Programmierspannung versucht werden, da bei zu hoher Programmierspannung der Baustein zerstört werden kann. In keinem Fall dürfen 5V-Flash-Speicher (29Fxxx -Typen) und EEPROMs (28Cxxx-Typen) als EPROMs oder 12V-Flash's programmiert werden.

Der Baustein-Typ wird in der Datei EPROP.OPT gesichert, so daß er beim nächsten Start voreingestellt ist.

Die Programmierspannung (V_{pp}) sowie der Algorithmus bekommen die Standard-Werte, die für den angewählten Baustein vorgegeben sind. Die Datenbusbreite der CPU wird auf die Datenbusbreite des Bausteins (8- oder 16-Bit) gesetzt. Die Anzahl der zu bearbeitenden Bytes wird auf EPROM-Größe und der Adreßoffset auf 0 gesetzt.

3. Aktion

Der Grundmenüpunkt "Aktion" enthält die zentralen Aktionen des EPROP:

- Leertest
- Auslesen
- Programmieren
- Verifikation
- Löschen
- Progr.+Verify
- Zusatz

Aktion - Leertest

Bei der Aktion "Leertest" wird geprüft, ob alle Speicherzellen den Wert FFh besitzen. Dieser Test wird nur im selektierten Adressbereich des Bausteins durchgeführt. Die Änderung dieses Adressbereichs kann mittels der Parameter "von" und "bis" vorgenommen werden.

Aktion - Auslesen

Die Aktion "Auslesen" liest die Daten aus dem angewählten Baustein und legt sie im angewählten Datei-Format in der Bezugsdatei ab.

Mit den Parametern "von" und "bis" kann der auszulesende Adreßbereich innerhalb des Bausteins eingegrenzt werden.

Die CPU-Parameter Busbreite und Position werden bei dieser Aktion nicht verwendet.

Der Datei-Offset, der nur beim "Binär-Format" angegeben werden kann, wird beim Auslesen nicht betrachtet. Im Binär-Format wird der Bereich zwischen den Parametern "von" und "bis" in die angegebene Bezugsdatei geschrieben. Eine zuvor bestehende Datei wird dabei überschrieben.

Bei Verwendung der Text-Record-Datei-Formate wird der unter dem CPU-Parameter "Adressraum" angegebene Wert bei der Adreßangabe berücksichtigt. Da eine Binärdatei keine Adreßangaben enthält, wird der Parameter "Adreßraum" in diesem Fall ignoriert.

Aktion - Programmieren

Durch Anwahl der Aktion "Programmieren" wird die Programmierung des angewählten Bausteins ausgelöst. Der Datei-Inhalt wird gemäß den Parametern "Algorithmus", "Prog.Spg.", "Busbreite", "Bus-Position", "Adresse", "von" und "bis" in den Baustein programmiert.

Durch Angabe der CPU-Busbreite und der Zuordnung des Bausteins zum CPU-Datenbus (Position des EPROMs am Datenbus) kann der Inhalt **einer** Datei auf **mehrere** Bausteine verteilt werden.

Nach der Programmierung des Bausteins ist eine anschließende Verifikation zu empfehlen.

Aktion - Verifizierung

Der Datei-Inhalt wird bei dieser Aktion mit dem Inhalt des Bausteins verglichen.

Die Parameter "Busbreite", "Bus-Position", "CPU-Adresse", "von" und "bis" werden wie bei der Aktion "Programmieren" berücksichtigt.

Aktion - Löschen

Löschen ist nur für EEPROMs und Flash-Speicher möglich.

Löschen von 12V-Flash-Bausteinen (28Fxxx): Nach den Datenblättern dieser ICs muß vor dem Löschen der gesamte Baustein mit "00" beschrieben werden. Jedoch funktioniert meistens auch der Quick-Erase Algorithmus, der darauf verzichtet (und erheblich schneller geht). Dennoch sei darauf hingewiesen, daß damit die Herstellerspezifikation umgangen wird.

EEPROMs werden beim Löschen vollständig mit "FF" programmiert.

5V-Flash-Speicher 29Fxxx können auch sektorweise gelöscht werden.

Aktion - Progr.+Verify

Dieser Befehl faßt die Befehle Programmieren und Verifizieren zusammen. Da bei einigen Bausteinen (EEPROMs und Flash-Bausteine) nicht automatisch eine echte Überprüfung der programmierten Werte stattfindet, sondern nur die Bestätigungsfunktion des ICs ausgenutzt wird, empfiehlt es sich, hier anschließend an die Programmierung eine Verifizierung durchzuführen.

Aktion - Zusatz

Unter dem Menüpunkt Aktion - Zusatz sind bausteinabhängige Sonderfunktionen zusammengefaßt. In Version 3.x steht hier nur die Set-Protection Funktion für Xicor-EEPROMs zur Verfügung.

4. Optionen

Der Grundmenüpunkt "Optionen" enthält folgendes Pull-Down-Menü:

- Algorithmus Änderung des voreingestellten Algorithmus.
- Prog.Spg. Änderung der voreingestellten Programmiervspannung (Vpp).
- Busbreite Angabe der CPU-Busbreite.
- Position Zuordnung von Baustein-Datenbus zum CPU-Datenbus.
- Adressraum Adresslage des ROM-Bereichs im CPU-Adressraum verändern
- von Erste zu bearbeitende Baustein-Adresse festlegen.
- bis Letzte zu bearbeitenden Baustein-Adresse festlegen.
- Makro-Record Makro aufzeichnen
- Makro-Play Makro ausführen
- Makro-beenden Makro-Aufzeichnung beenden
- Byte-Swap High- und Low-Byte der Datei vertauschen
- I/O-Adresse Portadresse des EPROP-Hostadapter (PC-Karte) ändern.

Optionen - Algorithmus

Der Programmier-Algorithmus wird im Normalfall durch den Baustein-Typ vorgegeben. Soll der Baustein jedoch mit einem anderen Algorithmus programmiert werden, kann in diesem Menüpunkt ein anderer Algorithmus eingestellt werden.

Durch Anwahl eines Bausteintyps wird dieser Parameter auf den für den Baustein üblichen Algorithmus zurückgesetzt. Es sollten möglichst die Angaben des Baustein-Herstellers beachtet werden, um eine Langzeitstabilität der programmierten Daten zu gewährleisten.

Die Liste der unterstützten Bausteine enthält hauptsächlich Standardtypen. Es kann deshalb die Eingabe eines schnelleren Algorithmus sinnvoll sein, um die Programmierzeiten zu verkürzen.

Optionen - Prog.Spg. (Programmiervspannung)

Die Programmiervspannung wird im Normalfall durch den Baustein-Typ vorgegeben. Will man den Baustein jedoch mit einer anderen Spannung programmieren, so kann über diesen Menüpunkt eine andere Spannung ausgewählt werden. Durch Anwahl eines Bausteintyps wird dieser Parameter auf die für den Baustein vorgegebene Programmiervspannung zurückgesetzt.

Da die Liste der unterstützten Bausteine in der Hauptsache Standardtypen vorsieht, ist es häufig schwierig, den zu verwendenden Baustein einem Standard-Typ zuzuordnen. Bei Zweifeln über die Programmiervspannung empfiehlt es sich, zunächst mit der kleineren Spannung zu beginnen, um einer Zerstörung des Bausteins vorzubeugen.

Optionen - Busbreite

Der CPU-Parameter "Busbreite" gibt die Datenbusbreite der verwendeten CPU an. Möglich sind die Werte 8-, 16-, 32- und 64-Bit.

Anschließend kann mit dem Parameter "Position" die Lage des Datenbusses des Speicherbausteins im CPU-Datenbus festgelegt werden. Mit den CPU-Parametern ist es möglich, eine Datei in mehrere EPROMs aufzuteilen. Wird beispielsweise ein 8-Bit EPROM in einem 16-Bit System verwendet, kann durch die Einstellung von "Busbreite" auf 16 Bit erreicht werden, daß nur jedes zweite Byte der Datei für die Programmierung des EPROMs verwendet wird, und zwar je nach Bus-"Position" die Bytes mit ungerader oder gerader Lage in der Datei. Der umgekehrte Vorgang, nämlich zwei 8-Bit Dateien

(eine mit den High- und eine mit den Low-Bytes) zu einer Datei zu verschmelzen, kann mit dem Hilfsprogramm MERGE.EXE (vom DOS-Prompt aus) durchgeführt werden.

Optionen - Position

Der CPU-Parameter "Position" gibt die Lage des Datenbus des Bausteins innerhalb des CPU-Datenbusses an. Dies ist dann von Bedeutung, wenn der CPU-Datenbus breiter ist als der Baustein-Datenbus, z.B. wenn vier 8-Bit EPROMs zur Realisierung eines 32-Bit CPU-Bus eingesetzt werden. Die Breite des CPU-Datenbusses sollte zuvor mit dem Parameter "Busbreite" eingestellt werden.

Der Parameter "Position" wird durch Neuwahl eines Baustein-Typs und durch Veränderung der CPU-Busbreite auf den niedrigsten Wert zurückgesetzt.

Durch Angabe von entsprechenden CPU-Parametern ist es möglich, eine Datei in mehrere Bausteine aufzuteilen.

Optionen - Adressraum

Der CPU-Parameter "Adressraum" gibt die Lage des Speicherbausteins (d.h. des ersten Bytes) im Adreßraum der CPU an. Der Parameter wird bei Veränderung der Bezugsdatei, des Verzeichnisses und des Formats auf seinen Initialwert 0 zurückgesetzt.

Dieser Parameter ist im Zusammenhang mit den "Text-Record"-Datei-Formaten wichtig. Bei den in Records organisierten Datei-Formaten wie Intel-Hex und S-Record kann die Adreßlage des Bausteins im CPU-Adreßraum angegeben werden. Dies ist notwendig, da diese Dateiformate auf den CPU-Adreßraum bezogene Adressen enthalten, ein IC aber im allgemeinen nicht den gesamten Adreßraum der CPU belegt und in vielen Fällen nicht bei CPU-Adresse 0 anfängt.

Optionen - von

Optionen - bis

Mit den Baustein-Parametern "von" und "bis" wird die Startadresse bzw. Endadresse im Speicherbaustein festgelegt. Auf diese Weise können Teilbereiche des Bausteins gelesen bzw. programmiert werden.

Durch Anwahl eines Baustein-Typs wird dieser Wert auf 0 zurückgesetzt.

Optionen - Datei-Offset

Durch den Parameter "Datei-Offset" ist es möglich (nur im BINARY-Format), einen bestimmten Datenbereich in einer Datei für die Programmierung oder Verifizierung auszuwählen (z.B. dann, wenn der Inhalt der Datei die Daten für mehrere EPROMs umfaßt).

Beim Programmieren und Verifizieren im BINARY-Format bezeichnet der Datei-Offset das erste Byte der Datei, das programmiert bzw. verifiziert werden soll.

Da bei den "Text-Record"-Datei-Formaten kein Zusammenhang von Datei-Position und EPROM-Adresse besteht, hat dort dieser Parameter keine Bedeutung. Er kann daher auch nicht angewählt werden.

Optionen - Byte-Swap

Für bestimmte Anwendungen ist es notwendig, High- und Low-Bytes einer Datei zu vertauschen. Dies läßt sich mit Byte-Swap durchführen. Vorsicht ist geboten bei Datenblöcken, die an ungeraden Adressen anfangen oder an geraden Adressen aufhören, da Byte-Swap an den Grenzen dann undefiniert ist. Insbesondere bei S-Record oder Intel-Hex Dateien sollte sichergestellt sein, daß jeder einzelne Datensatz mit gerader Adresse anfängt und mit ungerader Adresse aufhört.

Optionen - Tastatur-Makros

Durch die Aufzeichnung eines Tastatur-Makros kann ein bestimmter Ablauf automatisiert werden, z.B. für den Batch-Betrieb. Beim Abspielen des Makros mit "Makro-Play" dient die aufgezeichnete Tastenfolge als Eingabe, die Aktionen werden genauso ausgeführt wie während der Aufzeichnung. Nur bei Aufforderungen (z.B. zum Einlegen eines Bausteins in den Programmiersockel) und Mitteilungen an den Benutzer wird zur Bestätigung eine Tastatur-Eingabe erwartet.

Folgende Befehle stehen für Makros zur Verfügung:

Aufzeichnen eines Makros durch

Optionen - Makro-Record;

Beenden der Makro-Aufzeichnung durch

Optionen - Makro beenden;

Ausführen eines Makros durch

Optionen - Makro-Play.

Das Makro wird in der Datei EPROP.KB im aktuellen Verzeichnis gespeichert. Diese Datei kann als Textdatei auch mit einem Texteditor bearbeitet werden.

So führt das Löschen einer {PROMPT}-Anweisung dazu, das das Makro an dieser Stelle nicht mehr auf eine Bestätigung wartet. Durch den Befehl {PAUSE} können kurze Wartezeiten hinzugefügt werden.

Abbrechen eines Makros: Bei jedem Haltepunkt des Makros (Eingabe-Aufforderung) kann die Ausführung mit der ESC-Taste beendet werden.

Ein Makro kann auch in einer Schleife (rekursiv) aufgerufen werden. Hierzu muß der "Makro-Play" Befehl nach der Eingabe der übrigen Makro-Befehle aufgerufen werden; die Aufzeichnung wird danach automatisch beendet.

Batch-Betrieb

Durch den Aufruf des EPROP-Programms in der Form

EPROP -B

auf der DOS-Kommandozeile wird das in der Datei EPROP.KB gespeicherte Makro sofort nach dem Laden des Programms ausgeführt. Das Eprop-Programm kann daher auch von einer Batch-Datei aus gestartet werden, um z.B. unmittelbar nach dem Kompilieren einer EPROM-Datei ein EPROM zu brennen. Die eventuell notwendigen Parameter-Einstellungen (wie "von", "bis", Adressbereich) können dabei ebenfalls automatisch geladen werden.

Optionen - I/O-Adresse

Siehe Beschreibung unter "Die notwendige Schritte zur Inbetriebnahme" am Anfang dieser Bedienungshinweise.

Die I/O-Basisadresse wird in der Datei EPROP.CFG gesichert und wird daher bei allen folgenden Aufrufen des Programms verwendet.

5. Monitor

Zum Grundmenüpunkt "Monitor" gehört ein Pull-Down Menü mit den Unteroptionen

- Datei-Editor
- IC-Editor
- Checksumme Datei
- Checksumme IC

Der Datei-Editor

startet ein eigenständiges Programm zur Ansicht der Bezugsdatei. Dieses Programm mit dem Namen HEXEDIT.EXE sollte immer im gleichen Verzeichnis zu finden sein, in dem sich auch EPROP.EXE befindet.

Die Darstellung der Datei erfolgt jeweils mit 16 Bytes in einer Zeile. Diese 16 Bytes werden mit vorangehender Adresse in HEX-Darstellung angezeigt. Am Ende jeder Zeile werden die 16 Bytes als Zeichen (ASCII) dargestellt. Das Datei-Format wird nicht berücksichtigt, d.h. jede Datei wird als Binärdatei interpretiert.

Der Aufruf der Datei Hexedit und die Übergabe der nötigen Parameter erfolgt in der Datei eprop.cfg. Die Datei eprop.cfg wird von der Eprop-Software beim Starten eingelesen und beim regulären Beenden neu geschrieben.

Im Monitor-Aufruf steht „-o%lx“ für den Offset und „%s“ für die angewählte Bezugsdatei. Der Aufruf lautet beispielsweise:

```
Monitor_Call :C:\EPROP\HEXEDIT -o%lx %s :
```

Diese Zeile sollte nicht verändert werden, damit der Aufruf von hexedit.exe und die Übergabe der nötigen Parameter funktionieren.

Der IC-Editor

startet das gleiche Monitor-Programm wie der Datei-Editor, um den Inhalt des Bausteins anzuzeigen. Dazu wird der Inhalt des Bausteins bzw. der durch Option-"von" und Option-"bis" festgelegte Bereich in eine temporäre Datei ausgelesen. Der Pfad der temporären Datei ist in EPROP.CFG festgelegt.

Checksumme IC

Bei Anwahl dieses Punktes wird die Checksumme eines Bausteins berechnet. Dazu wird der Inhalt des Bausteins in eine temporäre Datei eingelesen.

Die Checksumme errechnet sich aus der Summe aller Bytes über den Adreßraum des Bausteins zwischen den Baustein-Parametern "von" und "bis".

Checksumme Datei

berechnet die Checksumme der Bezugsdatei. Das Dateiformat wird dabei nicht berücksichtigt, d.h. die Datei wird als Binärdatei interpretiert. Als Anfangs- und Endposition werden die Baustein-Parameter "von" und "bis" verwendet. Daher kann die Checksumme auch über einen eingeschränkten Datenbereich errechnet werden.

6. Hilfe

Der Grundmenüpunkt "Hilfe" enthält das Pull-Down-Menü:

- Inhalt
- Info

Mit **Inhalt**

gelangt man zum Hauptverzeichnis der zur Verfügung stehenden Hilfe-Themen. Die einzelnen Themen können mit der Maus oder der Return-Taste angewählt werden. Querverweise im Hilfetext sind hervorgehoben und können ebenfalls mit der Maus oder Return-Taste aufgerufen werden.

Die kontextsensitive Hilfe kann auch über die F1-Taste aufgerufen werden. Je nach dem gerade angewählten Menüpunkt erscheint der zugehörige Hilfe-Text.

Info

zeigt die Versionsnummer der EPROP-Software sowie einen Copyright-Vermerk an.

Dateiformate

Neben reinen Binärdateien geben viele Compiler auch sogenannte Textrecord-Dateien aus. In diesen werden alle Bytes als zwei Hexziffern dargestellt, die wiederum im (druck- und editierbaren) ASCII-Code abgespeichert werden, es wird also ein Byte pro Hexziffer gespeichert. Außerdem wird für jede Zeile ("Record") eine Adresse und eine Prüfsumme angegeben. Zwei verschiedene Records können daher Daten in ganz unterschiedlichen Adreßbereichen enthalten. Die Adreßbereiche aufeinanderfolgender Records müssen weder zusammenhängend noch geordnet sein.

Die gängigen Textrecord-Dateiformate sind Intel-HEX80, Intel-HEX86, S1-, S2 und S3-Records sowie Tektronix.

1. S-Records

Aufbau eines Records einer S-Record Datei:

:<Sn> <Länge> <Adresse> <Datenbytes> <Checksumme>

Sn = S1, S2 oder S3 je nach Typ

Länge (1 Byte) Anzahl Datenbytes + n + 2 (=Anzahl der nachfolgenden Bytes)

Adresse (n+1 Bytes) Adresse des ersten Datenbytes des Records im CPU-Adreßraum

Checksumme (1 Byte) Einerkomplement der Summe modulo 256 von Länge, Adresse und Datenbytes (jeweils byteweise addiert)

S0-Record

(am Anfang der Datei): kein Datenrecord, sondern Kommentar.

Beispiel:

S1104000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDF S1-Datenrecord mit 10h
Datenbytes (alle FF), Beginn bei CPU-Adresse 4000h, Checksumme DF.

Schluß-Record (EOF-Record):

S9030000FC bei S1-Records

S803000000FC bei S2-Records

S70300000000FC bei S3-Records

Die Datei kann beliebig viele Records haben. Alle Bytes werden als zwei Hexzeichen im ASCII-Code dargestellt.

2. Intellec 8/MDS Format (Intel-Hex)

Aufbau eines Records im Intel-Hex Format:

:<Anzahl Datenbytes><Adresse><Typ><Datenbytes><Prüfsumme>

Anzahl Datenbytes:	1 Byte
Adresse:	2 Bytes
Typ:	1 Byte; 00 = Datenrecord, 01= EOF-Record
Datenbytes:	bis zu 255
Prüfsumme	1 Byte: Zweierkomplement der Summe (modulo 256) über alle vorangehenden Bytes im Record.

Beispiel:

:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFC0	Datenrecord
:00000001FF	EOF- (End of File-) Record

Der Datenrecord enthält 10h Datenbytes (alle "FF"), die ab Adresse 0040h stehen. Die "00" zwischen Adresse und Daten kennzeichnet den Record als Datenrecord. Die Prüfsumme ist "C0".

Eine Intel-Hex Datei kann beliebig viele Records haben. Alle Bytes werden als zwei Hexzeichen im ASCII-Code dargestellt.

3. Intel MCS-86 Hexadecimal Object (Intel-Hex86 Format)

Die Datenrecords sind genauso wie beim Intel-Hex-Format aufgebaut. Hinzu kommen noch Records vom Typ 2 für die Segment-Adressen.

Allgemeine Form einen Intel-Hex86-Records:

:<Anzahl Datenbytes><Adresse><Typ><Datenbytes><Prüfsumme>

Anzahl Datenbytes:	1 Byte
Adresse:	2 Bytes
Typ:	1 Byte; 00 = Datenrecord, 01= EOF-Record
Datenbytes:	bis zu 255
Prüfsumme	1 Byte: Zweierkomplement der Summe (modulo 256) über alle vorangehenden Bytes im Record

Typ-2 Record: z.B.

:02000002FFFFFFE Segment-Adresse = FFFFh Prüfsumme = FEh

Die Offset-Adresse ist beim Typ-2-Record immer 0. Die Segment-Adresse ist für alle nachfolgenden Daten-Records bis zum nächsten Typ-2-Record gültig. Eine Intel-Hex86 Datei kann beliebig viele Records haben. Alle Bytes werden als zwei Hexzeichen im ASCII-Code dargestellt.

4. Tektronix-Format

Der Aufbau ist wiederum ähnlich wie bei Intel-Hex, es gibt jedoch zwei Prüfsummen und keine Typ-Angabe. Die Prüfsummen werden hier nicht über ganze Bytes, sondern über Hexziffern (Nibbles) gebildet.

/<Adresse><Anzahl Datenbytes><Prüfsumme 1><Datenbytes><Prüfsumme 2>

Adresse:	2 Bytes
Anzahl Datenbytes:	1 Byte, 0 = EOF-Record
Prüfsumme 1	1 Byte: Summe (modulo 256) über alle vorangehenden Hexziffern im Record
Typ:	1 Byte; 00 = Datenrecord, 01= EOF-Record
Datenbytes:	bis zu 255
Prüfsumme	1 Byte: Summe (modulo 256) über alle Hexziffern der Daten im Record

Beispiel:

/ABDE1031FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE

Adresse: ABCEh, Zahl der Datenbytes 10h, Prüfsumme 31h, Datenbytes (alle FFh), Prüfsumme E0h

/00000000 EOF-Record

Variablen in EPROP.CFG

In der Datei "EPROP.CFG" werden mehrere Variablen gesichert, die das Verhalten des Programms bestimmen. Die Datei wird in dem Verzeichnis erstellt, in dem sich auch EPROP.EXE befindet, und bei jedem Start eingelesen.

Die Dateien "EPROP.CFG" und "EPROP.OPT" haben folgendes Zeilenformat:

Variablenname: Wert :

Dieses Format sollte nicht geändert werden. Jede Zeile muss 78 Zeichen lang sein und mit einem Doppelpunkt abgeschlossen werden.

Variablen in EPROP.CFG:

PORTBASE: I/O-Basisadresse

_Color Normal_Color, Back_Color, Title_Color, Select_Color, Border_Color, und Alarm_Color sind Variablen, die als Wert das jeweils benutzte Bildschirmattribut enthalten. Bei Monochrom-Bildschirmen ist es häufig empfehlenswert, Select_Color auf 70h (statt 6Fh) zu setzen.

SoundDelay: Gibt die Tonlänge des Beepers an. Mit dem Wert 0 kann der Ton abgeschaltet werden.

Display_Mode: Gibt an, ob die Darstellung farbig oder monochrom erfolgen soll. "COLOR" bedeutet Farbe, "MONO" bedeutet Monochrom.

Variablen in EPROP.OPT

In der Datei EPROP.OPT werden mehrere Variablen des Programms gesichert, damit sie beim nächsten Start voreingestellt sind.

Die Datei wird bei Verlassen von EPROP im aktuellen Verzeichnis abgelegt. Daher sind für unterschiedliche Start-Verzeichnisse (Projekte) auch unterschiedliche Voreinstellungen möglich.

Bezüglich Zeilenformat gilt auch hier wieder das oben Gesagte zu Länge und Doppelpunkt: jede Zeile ist 78 Zeichen lang und wird mit einem Doppelpunkt abgeschlossen.

Variablenname : Wert :

Variablen in "EPROP.OPT":

WorkDir : Arbeits-Verzeichnis mit Laufwerksangabe
 Filename : Bezugsdatei
 FileFormat : Datei-Format Kodiert
 EpropType : Baustein-Typ

EPROP-Zubehör und weitere Produkte

Löschgeräte

EPROM-Löschgerät 1 - 4 W, max. 5 EPROMs, inkl. Netzteil

EPROM-Löschgerät 2 - 32 W, max. 48 EPROMs

(P)LCC-Adapter für den EPROP

EPROM-Adapter AL32D28E - 32-poliger LCC-Nullkraft-Sockel auf 28-poligen DIL-Sockel für LCC-EPROMs bis 512 kBit

EPROM-Adapter AP32D28E - 32-poliger PLCC-Nullkraft-Sockel auf 28-poligen DIL-Sockel für PLCC-EPROMs bis 512 kBit

Universal-Adapter AL32D32U - 32-poliger LCC-Nullkraft-Sockel auf 32-poligen DIL-Sockel für EPROMs ab 1 Mbit und Flash-EPROMs

Universal-Adapter AP32D32U - 32-poliger PLCC-Nullkraft-Sockel auf 32-poligen DIL-Sockel für EPROMs ab 1 Mbit und Flash-EPROMs

EPROM-Adapter AL44D40E - 44-poliger LCC-Nullkraft-Sockel auf 40-poligen DIL-Sockel

EPROM-Adapter AP44D40E - 44-poliger PLCC-Nullkraft-Sockel auf 40-poligen DIL-Sockel

Emulatoren

Simulant - der EPROM- und SRAM-Emulator

RAM/ROM-Emulator für alle gängigen 8- und 16-Bit EPROMs vom 2764 (8 kByte) bis zum 27080 (1MByte) und 27240 (512 kByte). In Systemen mit 16-Bit-Datenbus können zwei 8-Bit EPROMs gleichzeitig emuliert werden (in diesem Fall bis zum 27040). Die SRAM-Emulation erlaubt das Auslesen von Daten in Mikroprozessor-Schaltungen.

Design-51 - In-Circuit-Emulator für die 8051-Familie

Mit DESIGN-51 steht ein vollständiges Entwicklungssystem für einen großen Teil der 8051 Familie zur Verfügung. Das DESIGN-51 System besteht aus dem DESIGN-51 In-Circuit-Emulator und dem Symbolischen Debugger MBUG mit integriertem Cross-Assembler. Programme können komfortabel in der Ziel-Hardware getestet werden.

Der Symbolische Debugger MBUG ist ein PC-basiertes Programm, das die Steuerung des DESIGN-51 In-Circuit-Emulators über die serielle Schnittstelle übernimmt. Mit MBUG können Sie Programme übersetzen, auf den Emulator laden, debuggen und ausführen. Durch die ausgiebige Verwendung von Fenstern, Menus und Funktionstasten wird eine komfortable Benutzeroberfläche zum effektiven Arbeiten mit MBUG zur Verfügung gestellt. Der integrierte Cross-Assembler erlaubt das Übersetzen von Programmen für den Zielprozessor. Programme im Intel-Hex oder Intel OMF-51 Format werden in den Emulationsspeicher des DESIGN-51 geladen und anschließend getestet. Hochsprachen-Unterstützung gibt es für Compiler, die das Intel OMF-51 Format unterstützen.

Notizen