



# Flash-Mikrocontroller-Programmer FP 51

*Single-Chip-Mikrocontroller der MCS-51-Familie schnell und einfach „brennen“, ohne zeitraubendes EPROM-Handling (brennen, löschen...) ermöglichen die neuen Flash-Mikrocontroller von ATMEL, unter Einsatz des besonders günstigen Programmers FP 51.*

## Allgemeines

Mikrocontroller der Serie MCS-51 sind weit verbreitet und finden für zahlreiche Problemlösungen in der Elektronik Anwendung. Trotz ausgefeilter Software-Simulator-Programme ist es in der Programm-entwicklungsphase jedoch oft notwendig, die Programmierung der Mikrocontroller zu ändern. Hierzu sowie zur endgültigen Programmierung benötigen Entwickler ein effektives Werkzeug zum schnellen „Brennen“ der Prozessoren.

Weit verbreitet ist dafür der Einsatz von EPROMs bzw. Prozessoren mit integriertem EPROM, die in zahlreichen Varianten sowohl als OTP-Versionen (einmal programmierbar) als auch mit Quarzfenster zur Löschung verfügbar sind. Sowohl das „Brennen“ des EPROM-Programmspeichers als auch das Löschen ist vergleichsweise zeitaufwendig. Hinzu kommt die Anschaffung entsprechender Programmier- und Löscheräte.

## Flash-Controller - blitzartig löschen

Abhilfe können hier Mikrocontroller schaffen, die mit einem Flash-Programmspeicher ausgestattet sind. Merkmal dieser Speicherbausteine ist, daß sie elektrisch

löschar und mehrfach programmierbar sind.

Diese Eigenschaft macht die Controller besonders für die Entwicklung interessant, da der lange Löschvorgang durch das UV-Löschgerät entfällt und sehr schnell eine neue Version der Software in den Controller übertragen und getestet werden kann.

ATMEL bietet solche Flash-Controller in vier verschiedenen Grundtypen an. Durch deren unterschiedliche Leistungsmerkmale ist so für jeden Anwendungsfall der richtige Controller verfügbar.

Das hier vorgestellte PC-Programmiergerät ermöglicht das effektive Programmieren, Löschen und Auslesen dieser Flash-Controller.

Auffallend ist hierbei der geringe Hardwareaufwand zur Realisierung aller Funktionen des Programmiergerätes, was zu einem sehr guten Preis-/Leistungsverhältnis des Programmers geführt hat.

Die Ablaufsteuerung aller Funktionen erfolgt durch einen PC. Dies geschieht über eine komfortable Software, die unter Windows 3.1x/95 lauffähig ist.

Lediglich ein Verbindungskabel zwischen einer freien Parallelschnittstelle des PCs und dem Programmiergerät ist erforderlich, um den Datenaustausch zu gewährleisten. Somit ist kein Eingriff in den Computer notwendig.

Die Stromversorgung des Programmiergerätes übernimmt ein Steckernetzteil.

## Die verschiedenen Mikrocontroller

Die vier am häufigsten eingesetzten Flash-Controller sollen mit ihren Leistungsmerkmalen nun näher vorgestellt werden. Die technischen Daten sind dabei in der Tabelle 1 übersichtlich dargestellt.

Die zwei Typen mit der Bezeichnung 89C51 und 89C52 sind in einem 40poligen DIP-Gehäuse untergebracht und pinkompatibel zu den bekannten OTP-Versionen 87C51 und 87C52, deren Anschlußbelegung Abbildung 1 zeigt.

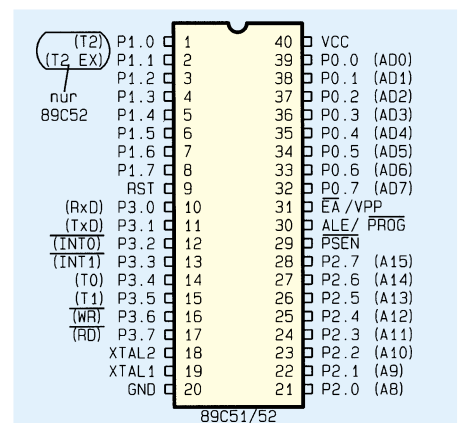
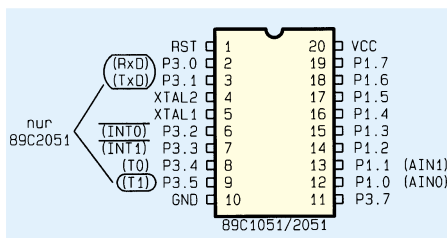


Bild 1: Anschlußbild des 89C51/52

**Tabelle 1: Technische Daten der Flash-Mikrocontroller**

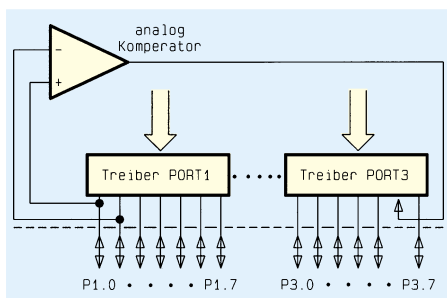
	89C1051-24PC	89C2052-24PC	89C51-20PC	89C52-20PC
Datensicherheit:	mind. 10 Jahre			
Programmierzyklen:	mind. 1000			
Betriebsspannung:	4 bis 6V			
Stromsparmodes:	Idle und Power Down			
Taktfrequenz:	0 bis 24MHz	0 bis 24MHz	0 bis 20MHz	0 bis 20MHz
Programmspeicher:	1 kByte	2 kByte	4 kByte	8 kByte
statisches RAM:	64 Byte	128 byte	128 Byte	256 Byte
Ein-/Ausgabepins:	15	15	32	32
16-Bit-Zähler:	1	2	2	3
prog. serieller Port:	0	1	1	1
Interruptquellen:	3	5	5	8
analoge Komparatoren:	1	1	0	0
Sicherheits-Bits:	2fach	2fach	3fach	3fach

Die beiden anderen Typen, mit der Bezeichnung 89C1051 und 89C2051, sind besonders für kleinere Projekte interessant. Die Controller befinden sich in einem 20poligen DIP-Gehäuse und benötigen somit sehr wenig Platz auf der Leiterplatte (siehe Abbildung 2). Sie sind befehlskompatibel zum C51-Controller-Standard und können somit ebenso einfach programmiert werden. Als Besonderheit verfügen diese Typen über einen internen Komparator, der zum Beispiel den Aufbau eines AD-Wandlers ermöglicht.



**Bild 2: Anschlußbild des 89C1051/2051**

Abbildung 3 zeigt, daß dabei die Eingänge P1.0 und P1.1 auf den Komparator geschaltet sind, dessen Ausgang auf den nicht nach außen geführten Port-Pin P3.6 gelegt ist.



**Bild 3: Komparator-Innenschaltung des 89C1051/2052**

Wie aus der C51-Controller-Serie bekannt, sind die Port-Pins mit einem internen Pull-Up-Widerstand und einem Transistor, der nach Masse schalten kann, aus-

gestattet, um diese als Ein- oder Ausgänge nutzen zu können. Bei den Pins P1.0 und P1.1 ist darauf zu achten, daß der interne Pull-Up-Widerstand nicht vorhanden ist, damit bei der Nutzung als Komparator keine Belastung der Eingangssignale auftritt. Sollen die Pins als normaler Ein- oder Ausgang dienen, so ist der Widerstand extern zu beschalten.

### Schaltung

Das Schaltbild des Flash-Programmers FP 51 ist in den Abbildungen 4 bis 6 in den einzelnen Funktionsbaugruppen Spannungsversorgung, Parallelinterface und Programmiersockelbeschaltung dargestellt.

Abbildung 4 zeigt die Spannungsstabilisierung und den Schaltungsteil zum Schalten der Programmier- und Betriebsspannung für den zu programmierenden Mikrocontroller.

Die ungestabilisierte Betriebsspannung gelangt über die Klinkenbuchse ST 1 an die Schaltung und ist über die Sicherung SI 1 gegen einen eventuellen Kurzschluß gesichert. Die Diode D1 schützt das Gerät bei versehentlicher Verpolung der Betriebsspannung. Dann leitet die Diode, was zum Ansprechen der Sicherung führt.

Der Spannungsregler IC 1 übernimmt die Stabilisierung der 12V-Spannung, die als Programmierspannung für die Controller dient. Der Spannungsregler IC 2 stabilisiert die 5V-Betriebsspannung für die Schaltung. Die Kondensatoren C 1 bis C 6 dienen zur Pufferung und Unterdrückung von Schwingneigungen und die Leuchtdiode D 2 signalisiert die Betriebsbereitschaft des Gerätes.

Die Versorgungsspannung für den zu programmierenden Controller wird über den Transistor T 2 vom Typ BC 327 geschaltet, der wiederum von IC 13 über den Transistor T 1 vom Typ BC 548 angesteuert wird. Das Anliegen der Betriebsspan-

nung signalisiert die LED D 4, die darauf hinweist, daß der Controller derzeit nicht aus der Programmierfassung entnommen werden darf.

Der Controller benötigt je nach Typ eine Programmierspannung von 5 V oder 12 V. Diese wird über IC 13 durch die Transistoren T 3 bis T 6 geschaltet.

Abbildung 5 zeigt die Schaltung des Parallel-Interface, das die Verbindung zur PC-Schnittstelle realisiert. Die BUSY-Leitung des PCs wird über das Widerstandsarray R 17 auf High-Potential gezogen, wodurch ein Drucker nachgebildet wird, der nicht bereit ist, Daten zu empfangen.

Damit die Software feststellen kann, ob der Flash-Programmer angeschlossen ist, wird die SLCT-Leitung eingesetzt. Das Signal wird über das Gatter IC 3 D invertiert und über das Gatter IC 5 D auf den ACK-Eingang des PCs zurückgeführt.

Nach dem Anlegen der Versorgungsspannung erzeugt die Kombination C 7 und R 23 einen negativen Reset-Impuls, der alle Zwischenspeicher löscht und somit das Gerät in den Grundzustand versetzt. Zusätzlich kann der Reset auch durch den PC erzeugt werden, indem dieser die RESET-Leitung nach Masse zieht. Hierdurch liegt am Ausgang des Gatters IC 3 C vom Typ 74HC32 ein High-Signal an, das den Transistor T 7 durchsteuert, der wiederum zieht die CLR-Leitung nach Masse. Trennt man das Programmiergerät vom PC, zieht der Widerstand R 18 gleichzeitig die RESET-Leitung nach Masse, wodurch das Programmiergerät ebenfalls in den Grundzustand versetzt wird.

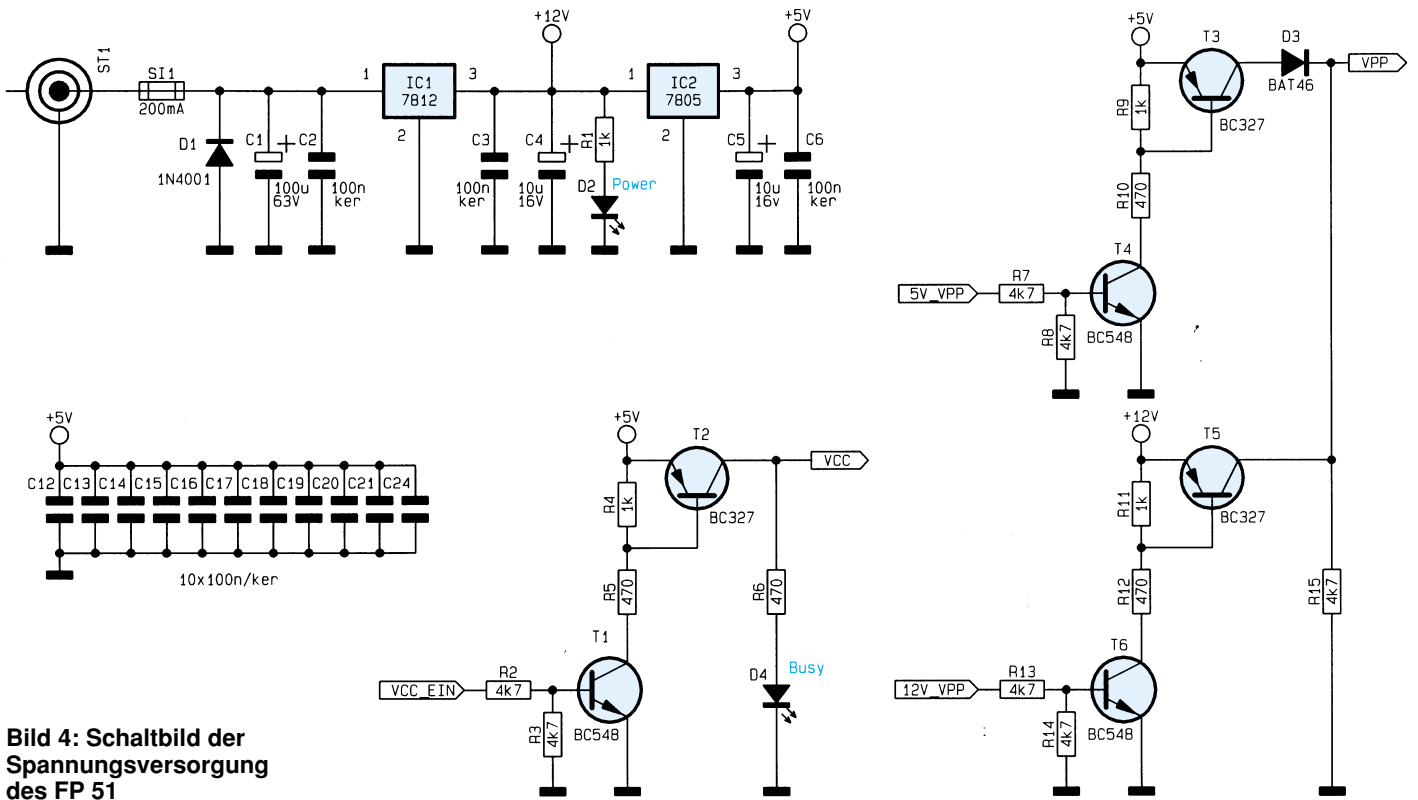
Die Datenleitungen D 0 bis D 7 des PCs sind über die Schutzwiderstände R 47 bis R 54 geführt, die den nachgeschalteten Treiber IC 4 vom Typ 74HC245 gegen statische Entladungen an den Schnittstellenpins schützen.

Über die STROBE-Leitung, die über das Gatter IC 3 A gepuffert ist, können die Signale der Datenleitungen in das D-Latch IC 7 vom Typ 74HC273, das die Steuersignale des Programmers speichert, übernommen werden.

Die Signale der Pins 2, 5 und 6 sind auf den Adreßdecoder IC 15 vom Typ 74HC138 geschaltet, der je nach anstehender Codierung einen Ausgang aktiviert, der mit den Treibern und Zwischenspeichern verbunden ist.

Durch den Wechsel der AUTOLF-Leitung des PC auf High-Potential wird der Adreßdecoder aktiviert, der den ausgewählten Ausgang nach Low-Potential schaltet. Der angeschlossene Zwischenspeicher übernimmt daraufhin die Daten vom Datenbus.

Die in den Mikrocontroller zu programmierenden Daten werden byteweise parallel an den Controller gelegt. Dazu dient der

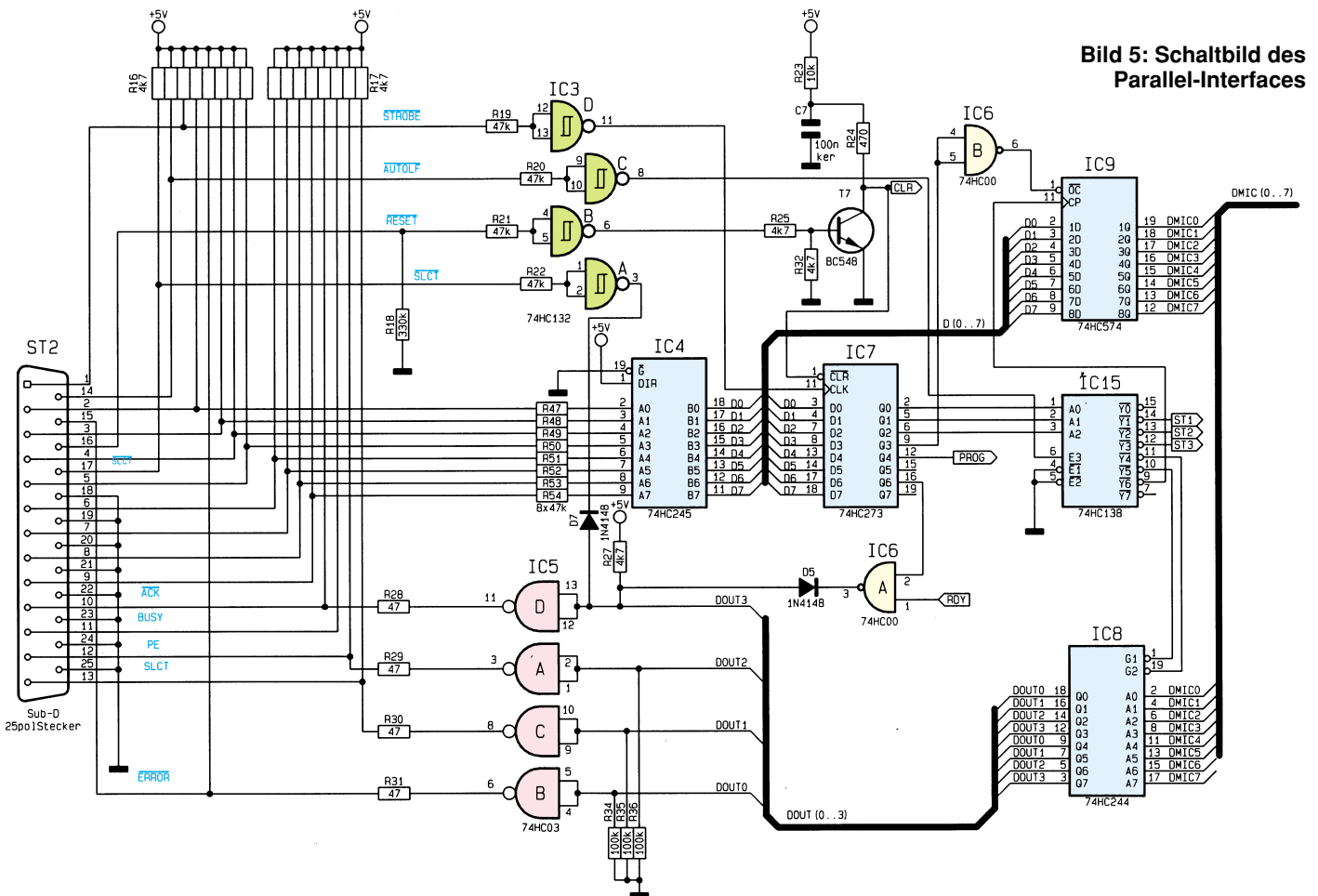


**Bild 4: Schaltbild der Spannungsversorgung des FP 51**

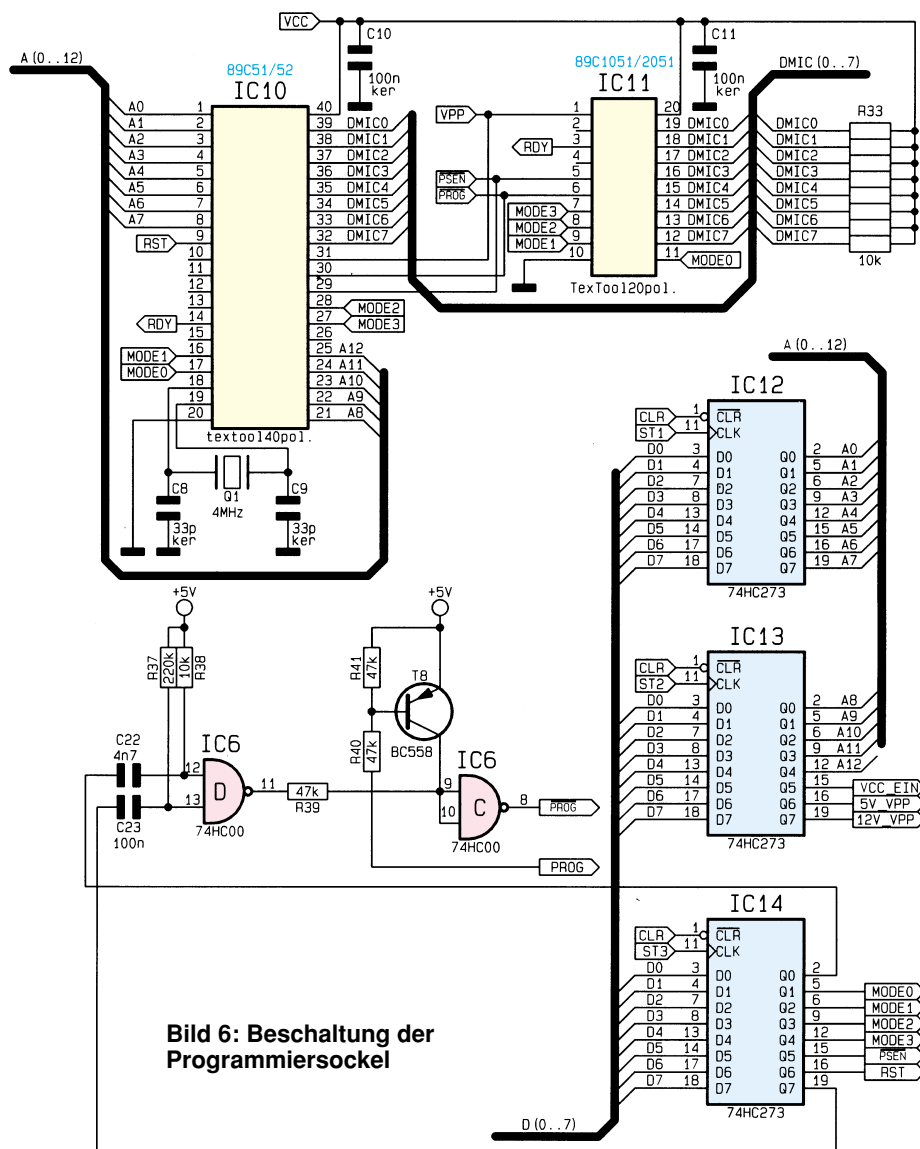
Zwischenspeicher IC 9 des Typs 74HC574, der über IC 15 getaktet wird und daraufhin das Datenwort übernimmt. Die Freigabe der Treiberausgänge erfolgt über Pin 9 des

IC 7 und das Gatter IC 6 B. Um den Speicherinhalt eines Controllers auslesen zu können, muß das Datenwort vom Mikrocontroller gelesen werden. Dazu sind die Ausgangstreiber des IC 9 zu deaktivieren und über den Treiber IC 8 jeweils 4 Bits auf die Gatter des IC 5 zu schalten. Letztere sind mit den

den. Dazu sind die Ausgangstreiber des IC 9 zu deaktivieren und über den Treiber IC 8 jeweils 4 Bits auf die Gatter des IC 5 zu schalten. Letztere sind mit den



**Bild 5: Schaltbild des Parallel-Interfaces**



**Bild 6: Beschaltung der Programmiersockel**

Eingangsleitungen  $\overline{ACK}$ , PE, SLCT und ERROR des PCs verbunden. Der PC kann somit durch zwei Lesevorgänge die oberen und unteren 4 Bit jedes Datenwortes einlesen.

Wird ein Byte in einen Mikrocontroller programmiert, so benötigt dieser ca. 2 ms für das Abspeichern. Während dieser Zeit kann der Controller keine weiteren Befehle verarbeiten, und das Programm muß warten.

Dazu legt der zu programmierende Mikrocontroller seine RDY-Leitung auf Low. Diese Information gelangt an das Gatter IC 6 A, das wiederum über den Pin 16 des IC 7 freigeschaltet wird. Über die  $\overline{ACK}$ -Leitung kann der PC dann den Zustand der Programmierung überwachen und warten, bis das Byte in den Controller geschrieben ist.

Abbildung 6 zeigt die Beschaltung der Programmiersockel. Bei den 40poligen Mikrocontrollern sind die Adresse der zu programmierenden Speicherstelle und das zu schreibende Byte an den Controller zu legen und dann ein Programmierimpuls zu

generieren. Im Gegensatz dazu werden die 20poligen Controller seriell beschrieben, indem die zu schreibenden Bytes hintereinander an den Controller gelangen und nach dem Programmierimpuls der interne Adreßzähler jeweils um eine Stelle weitergesetzt wird.

Die Zwischenspeicher IC 12 bis IC 14 puffern die Steuersignale und Adreßleitungen.

Die Signale der Leitungen MODE 0 bis MODE 3 bestimmen, ob der Controller gelöscht, ausgelesen, programmiert oder ob die Sicherheitsbits gesetzt werden sollen.

Für das Löschen der Controller ist ein negativer Programmierimpuls von 10 ms erforderlich, und für das Programmieren einer Speicherstelle wird ein Programmierimpuls von 1 bis 100  $\mu$ s benötigt.

Da diese Zeiten nicht weit von den Vorgaben abweichen dürfen, werden diese Impulse durch die Hardware erzeugt. Zudem ist es unter Windows kaum möglich, solche Zeiten exakt zu generieren, da aufgrund des Multitaskings Verzögerungen auftreten können.

Die Impulse werden über das Gatter IC 6 D erzeugt, dessen Eingänge mit zwei RC-Gliedern beschaltet sind. Im Ruhezustand führen die Ausgänge Pin 2 und Pin 19 des IC 14 High-Potential. Die Eingänge des IC 6 D liegen ebenfalls auf High-Potential. Durch den nachgeschalteten Inverter IC 6 C gelangt dann das High-Potential an den Controller. Über den Transistor T 8 kann der Eingang des IC 6 C auf „high“ gezogen werden, um so den Ausgang des Gatters auf Low-Potential zu schalten.

Wechselt nun Pin 2 oder Pin 19 des IC 14 auf Low-Potential, so entsteht am Ausgang IC 6 D ein negativer Programmierimpuls mit ca. 10 ms bzw. 47  $\mu$ s Dauer, unabhängig davon, wie lange die Leitung des IC 14 auf Low-Potential bleibt.

## Software

Die Installation der Windows-Bediensoftware erfolgt menügesteuert, indem das Programm „Install“ auf der Programmdiskette ausgeführt wird. Hierbei ist die Programmgruppe und das Installationsverzeichnis frei wählbar.

Vor dem Start des Programms ist das Programmiergerät zunächst an den PC anzuschließen. Dazu wird das Gerät über eine 25polige 1 : 1-Verbindungsleitung mit einer freien parallelen Schnittstelle des PCs verbunden. Über ein Steckernetzgerät (15 V/200 mA) erfolgt die Spannungsversorgung an ST 1.

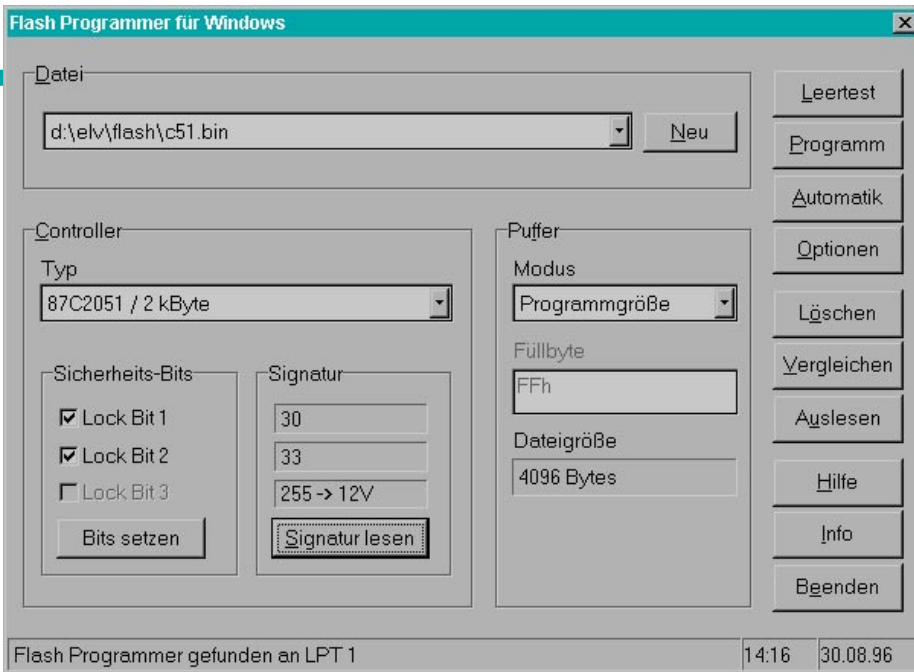
Nun startet man das Programm unter Windows 3.x durch einen Doppelklick auf das Programmsymbol oder unter Windows 95 auch über das Startmenü. Es erscheint das Hauptmenü der Bediensoftware (Abbildung 7). Gleichzeitig sucht das Programm an den vorhandenen Schnittstellen automatisch nach dem Flash-Programmer. In der untersten Zeile des Menü-Fensters wird dabei angezeigt, an welchem Port die Software das Programmiergerät gefunden hat.

Die Bedienung des Programms ist durch die übersichtliche Benutzeroberfläche sehr einfach.

Im Bereich „Datei“ läßt sich diejenige Datei auswählen, die in den Mikrocontroller programmiert werden soll. Das Programm kann Binär-Dateien verarbeiten, bietet aber auch die Möglichkeit, eine HEX-Datei auszuwählen, die dann zuvor in eine Binär-Datei umgewandelt und auf der Festplatte abgelegt wird.

## Schnelles Vielfach-Programmieren mit History-Funktion

Als Besonderheit bietet das Programm eine History-Funktion, die es ermöglicht, die letzten 16 Dateien wieder auszuwählen. Dies ist besonders zweckmäßig beim



**Bild 7: Ansicht des Programm-Fensters**

Programmieren mehrerer Controller sowie beim wiederholten Programmieren des gleichen Controllers. Dazu wird der Button rechts im Anzeigenfeld für den Dateinamen mit der Maus betätigt, woraufhin eine Liste mit den zuletzt bearbeiteten Dateien erscheint. Trifft man eine Auswahl, so erfolgt nicht nur das Laden der Datei, sondern auch das Wiederherstellen der dazugehörigen Einstellungen für den Controllertyp usw. Das Programm legt hierzu für jede geöffnete Datei eine Projektdatei an, in der alle Einstellungen gespeichert sind.

Im Feld „Controller“ läßt sich der zu programmierende Controllertyp auswählen, und im Feld „Signature“ kann mit der Funktion „Signature lesen“ die Bauteilkennzeichnung des in der Programmierfassung befindlichen Mikrocontrollers gelesen und angezeigt werden.

Im Bereich „Sicherheits-Bits“ kann man vorgeben, welche Sicherheits-Bits im Controller gesetzt werden sollen, wobei die 20poligen Controller nur über zwei Sicherheits-Bits verfügen. Zu beachten ist hierbei, daß das erste Lock-Bit aktiv sein muß, wenn das zweite Lock-Bit programmiert werden soll. Ebenso müssen die Lock-Bits 1 und 2 aktiv sein, wenn das dritte Lock-Bit gesetzt werden soll.

Die einzelnen Bits haben dabei folgende Bedeutung:

Bei Aktivierung des ersten Sicherheitsbits ist keine weitere Programmierung des Controllers möglich.

Das zweite Sicherheitsbit verhindert das Auslesen des Programmspeichers.

Das Setzen des dritten Sicherheitsbits ist nur bei den 40poligen Versionen möglich und verhindert, daß der Controller mit einem externen Programmspeicher betrieben werden kann.

Durch Ausführen der Funktion „Bits setzen“ werden die angegebenen Bits des

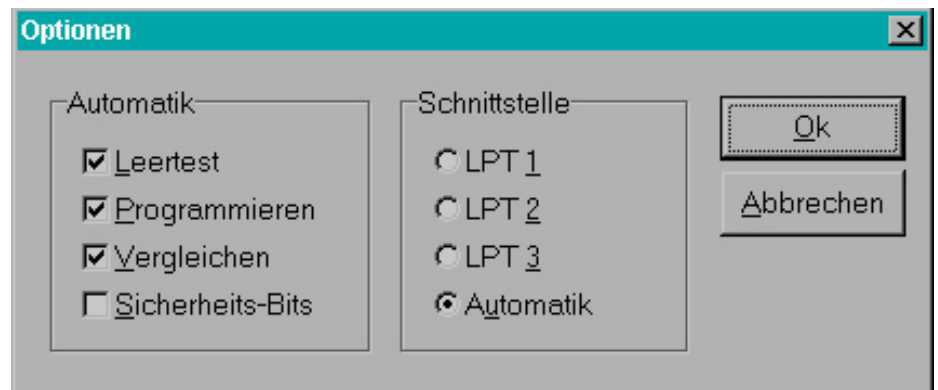
in der Programmierfassung befindlichen Mikrocontrollers programmiert. Die Controller sind jedoch jederzeit wieder löscht und neu programmierbar, gleichgültig, welche Sicherheits-Bits gesetzt sind.

Im Bereich „Puffer“ des Hauptfensters läßt sich auswählen, ob der Programmspeicher des Mikrocontrollers nur bis zur Größe des eigentlichen Programms oder über den gesamten Speicherbereich programmiert werden soll, wobei dann ein Füllbyte für den restlichen Speicherbereich zu definieren ist.

Auf der rechten Seite des Hauptfensters befinden sich die Buttons für die Funktionen, zum Löschen, Programmieren, Vergleichen und Auslesen.

Für die Programmierung von Serien bietet das Programm die Möglichkeit der automatischen Programmierung durch die

**Bild 8: Ansicht des Fensters zur Konfiguration der Automatik-Funktion**



Funktion „Automatik“. Hierbei lassen sich Leertest, Löschen, Programmieren, Vergleichen und das Setzen von Sicherheits-Bits automatisch ausführen. Welche Funktionen dabei im einzelnen einstellbar sind, zeigt das zugehörige Auswahlfenster, das

## PC-Hardware

über die Funktion „Optionen“ zu erreichen ist (Abbildung 8).

### Nachbau

Die Schaltung des ELV-Flash-Programmers ist auf einer 82 x 163 mm messenden, doppelseitigen Leiterplatte untergebracht. Die Bestückung erfolgt in gewohnter Weise anhand der Stückliste und des Bestückungsplanes, wobei zunächst die niedrigen, gefolgt von den höheren Bauteilen bestückt, verlötet und die überstehenden Drahtenden abgeschnitten werden. Die LEDs D 2 und D 4 sowie die Programmiersockel IC 10 und IC 11 sind vorerst nicht zu bestücken.

Die Montage des Spannungsreglers IC 1 erfolgt liegend. Er ist mit einer M 3 x 8 mm-Zylinderkopfschraube und einer M 3-Mutter auf der Leiterplatte zu verschrauben.

Sind die Bestückungsarbeiten soweit abgeschlossen, werden die LEDs von der Lötseite aus, mit einem Abstand von 4 mm zur Leiterplatte, bestückt und verlötet.

Nun sind die Programmiersockel ebenfalls von der Lötseite aus einzusetzen und von der Bestückungsseite her zu verlöten. Dazu verwendet man einen 20- und einen 40poligen IC-Sockel und steckt anschließend einen zweiten Sockel zur Vergrößerung der Bauhöhe in den ersten. Soll das Programmiergerät mit TEXTTOOL-Sockel aufgebaut werden, so sind diese direkt auf die Leiterplatte zu setzen und zu verlöten.

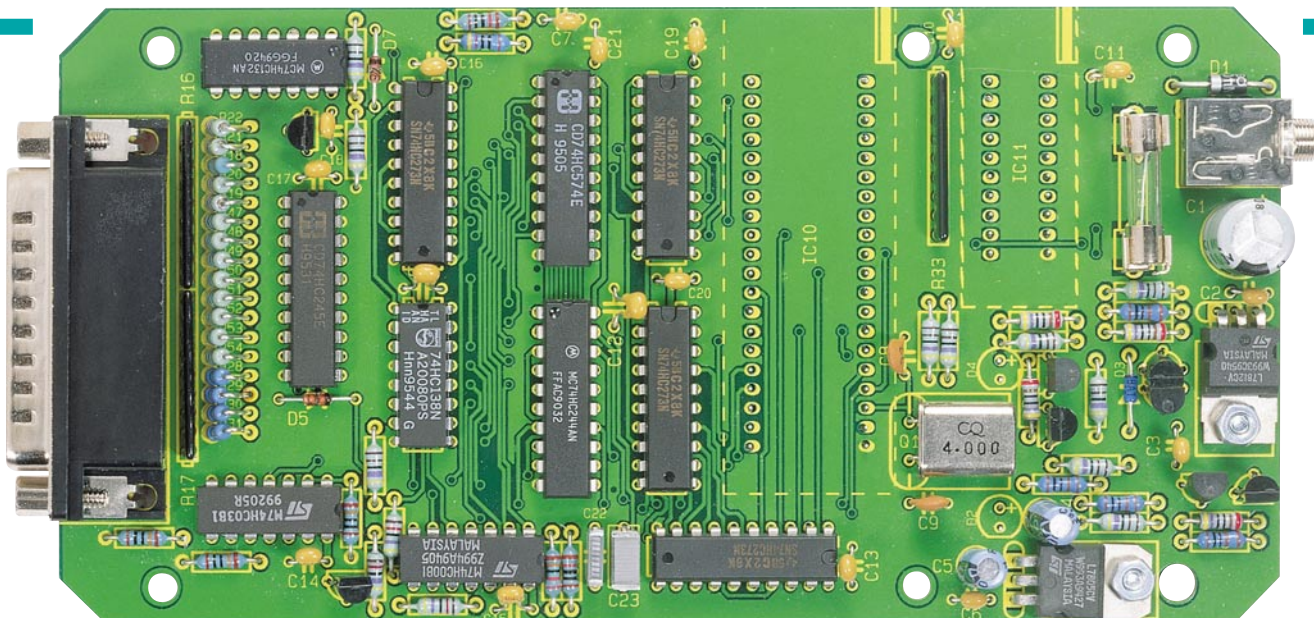
Damit ist der Aufbau bereits abgeschlossen, und es folgt der Einbau in das bearbeitete und bedruckte Gehäuse.

Der fertige Aufbau wird hierzu so in das Gehäuseunterteil eingesetzt, daß die Programmiersockel durch die entsprechenden Öffnungen im Gehäuse ragen. Daraufhin wird die Leiterplatte mit 6 Knippingschrauben (2,2 x 6,5mm) im Gehäuse verschraubt.

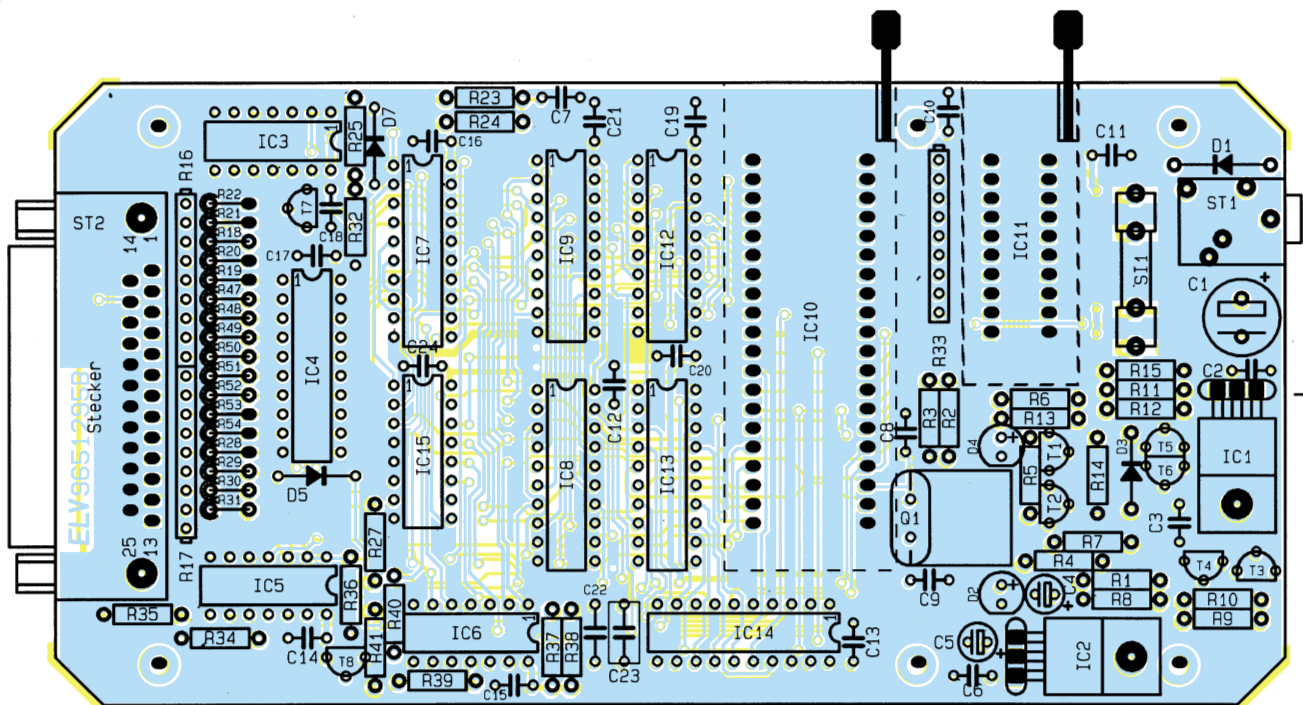
Zum Abschluß ist die zweite Gehäuse-

hälfte aufzusetzen und mit den dazugehörigen Gehäuseschrauben zu verschließen.

Somit ist das Programmiergerät einsatzbereit für das Programmieren, Löschen und Lesen von Flash-Mikrocontrollern.



Ansicht der fertig aufgebauten Leiterplatte mit zugehörigem Bestückungsplan



### Stückliste: Flash-Mikrocontroller-Programmer

#### Widerstände:

150Ω .....	R28-R31
470Ω .....	R5, R6, R10, R12, R24
1kΩ .....	R1, R4, R9, R11
4,7kΩ .....	R2, R3, R7, R8, R13, R14, R15, R25, R27, R32
10kΩ .....	R23, R38
47kΩ .....	R19-R22, R39-R41, R47-R54
100kΩ .....	R34-R36
220kΩ .....	R37
330kΩ .....	R18
Array, 4,7kΩ .....	R16, R17
Array, 10kΩ .....	R33

#### Kondensatoren:

33pF/ker .....	C8, C9
4,7nF .....	C22
100nF .....	C23

100nF/ker .....	C2, C3, C6, C7, C10-C21
10µF/25V .....	C4, C5
100µF/63V .....	C1

#### Halbleiter:

7812 .....	IC1
7805 .....	IC2
74HC132 .....	IC3
74HC245 .....	IC4
74HC03 .....	IC5
74HC00 .....	IC6
74HC273 .....	IC7, IC12-IC14
74HC244 .....	IC8
74HC574 .....	IC9
74HC138 .....	IC15
BC548 .....	T1, T4, T6, T7
BC327 .....	T2, T3, T5
BC558 .....	T8

1N4001 .....	D1
BAT46 .....	D3
1N4148 .....	D5, D7
LED, 3mm, rot .....	D2, D4

#### Sonstiges:

Quarz, 4MHz .....	Q1
Klinkenbuchse, 3,5mm, stereo .....	ST1
SUB-D-Stiftleiste, 25polig .....	ST2
Sicherung, 200mA, träge .....	SI1
Platinensicherungshalter (2 Hälften)	
2 Präzisions-IC-Fassung, 20polig	
2 Präzisions-IC-Fassung, 40polig	
2 Zylinderkopfschrauben, M3 x 6mm	
6 Knippingschrauben, 2,2 x 6,5mm	
2 Muttern, M3	
1 Gehäuse, gebohrt und bedruckt	