

PIC-Grundlagen Teil 7

Im abschließenden Teil der Artikelserie stellen wir ein Programmiergerät für den PIC 16F84 vor. Das Gerät verfügt zusätzlich über einen Adapter, der anstelle des PICs in eine Anwendungs-Schaltung eingesetzt werden kann und so die Überprüfung des Programms ermöglicht, ohne den PIC aus dem Programmiergerät entnehmen zu müssen.

Allgemeines

Jeder, der sich mit Mikrocontrollern beschäftigt, kennt die Probleme bei der Entwicklung eines Programms. Es werden Routinen erstellt, die anschließend überprüft werden müssen. Eine Hilfe ist hier der Simulator der PIC-Entwicklungsumgebung, mit dem man das Programm schrittweise ausführen lassen und den Zustand der Register beobachtet kann. Wenn es aber um die Überprüfung von Interrupt-routinen und das Erzeugen von zeitlichen Abläufen geht, ist der Simulator nicht mehr nutzbar.

Wer dann über einen Emulator verfügt, kann das Programm in diesen übertragen und direkt im Zielsystem überprüfen. Der Nachteil des Emulators ist der hohe Preis, der die Anschaffung für den privaten Gebrauch meist nicht erlaubt.

Also ist immer ein PIC zu programmieren und dann zur Überprüfung in die Schaltung zu stecken. Danach muß man den PIC erst für einige Minuten in ein UV-Löschgerät legen, bevor er neu programmierbar ist.

Zumindest der Löschvorgang kann bei dem PIC 16F84, den wir im vorangegangenen Artikel dieser Serie vorgestellt haben, eingespart werden, da er über einen Flash-Speicher verfügt und elektrisch löschtbar ist. Jedoch ist es auch hier notwendig, den PIC im Verlaufe der Programmerprobung ständig zwischen Programmiergerät und Schaltung zu wechseln.

Aufgrund der Vorzüge des PIC 16F84 ist dieser sehr beliebt, so daß zum Beispiel im Internet verschiedene Schaltungen zum Programmieren zu finden sind. Die Programmiergeräte sind meist sehr einfach ausgeführt, die Programmierung erfolgt direkt über die Port-Leitungen der parallelen oder seriellen Schnittstelle eines PCs.

Da der PIC 16F84 über ein serielles Interface programmiert wird, das recht unempfindlich gegenüber Abweichungen des Timings ist, arbeiten diese Schaltungen meist recht zuverlässig.

Das von ELV entwickelte Programmiergerät geht hier einen anderen Weg.

Das Gerät wird an eine serielle PC-Schnittstelle angeschlossen und verfügt selbst über einen kleinen Mikrocontroller,

der die Programmierung des PICs steuert.

Der Datenaustausch mit dem PC erfolgt über ein Standard-V24-Protokoll, bei dem das Programmiergerät einen Befehl vom PC erhält, diesen ausführt und anschließend eine Bestätigung zum PC zurücksendet. Durch dieses Vorgehen ist die Steuerung nicht vom PC-Typ abhängig. Hier kann man zum Beispiel ohne Probleme einen alten 386er bis hin zum schnellsten Pentium II einsetzen. Ebenso ist auch der Betrieb unter „Windows“ gewährleistet, da hier keine zeitlichen Abläufe realisiert werden müssen (unter „Windows“ nur bedingt möglich).

Der entscheidende Vorteil des ELV-Programmiergerätes ist aber die Emulatorfunktion. Dazu sind alle Pins des PICs, der sich im Programmiergerät befindet, über eine Flachbandleitung mit einem IC-Adapter verbunden, der in die Anwendungsschaltung statt des PICs einsetzbar ist.

Soll nun ein neues Programm in den PIC geladen werden, so stoppt das Programmiergerät den Programmablauf und verbindet die zur Programmierung benötigten Pins des PICs mit dem internen Mikrocon-

troller. Ist die Programmierung abgeschlossen, werden die Signale zurück auf den IC-Adapter geschaltet und das Programm wird neu gestartet.

Mit einfachen Mitteln ist so ein Emulator entstanden, der bei der Programm-Entwicklung wertvolle Hilfe leisten kann.

Bedienung

Das Programmiergerät wird über eine 9polige 1:1-Verbindungsleitung mit einer freien Schnittstelle des PCs verbunden. Soll das Gerät als Emulator zum Einsatz kommen, so ist der IC-Adapter anstelle des PICs 16F84 in das Zielsystem einzusetzen.

Die Spannungsversorgung erfolgt von der Testschaltung aus, zur Kontrolle leuchtet die grüne LED. Wird das Gerät nur zum Programmieren verwendet und der IC-Adapter nicht genutzt, so erfolgt die Spannungsversorgung über ein Steckernetzgerät, das eine Gleichspannung im Bereich von 9 bis 15 V liefert. Ebenso kann die Spannungsversorgung über das Steckernetzgerät erfolgen, wenn das Zielsystem nicht in der Lage ist, den benötigten Strom für das Programmiergerät zu liefern.

Die technischen Daten des Gerätes sind in Tabelle 8 zusammengefaßt.

Die Steuerung des PIC-Programmiergerätes erfolgt über eine Windows-Bediensoftware, die unter Windows 3.1x/95/98 läuft. Das Installationsprogramm richtet eine neue Programmgruppe ein, in der sich die Bediensoftware befindet, das sich durch einen Doppelklick starten läßt. Daraufhin erscheint das Programmfenster, das in Abbildung 32 dargestellt ist. Im ersten Schritt erfolgt mit dem Auswahlfeld links unten die Einstellung der seriellen Schnittstelle, an der das Programmiergerät angeschlossen ist.

Die Auswahl der Programmdatei, die in den PIC programmiert werden soll, erfolgt mit dem Button „Neu“. Daraufhin öffnet sich das Auswahlfenster, in dem die Datei, die im EX-Format vorliegen muß, auswählbar ist. Nach dem Anklicken des Fensters mit dem „OK“-Button erscheint der Dateiname im Feld „Datei“ und der Inhalt wird eingelesen. Im Feld „Optionen“ erfolgt die Einstellung, welcher Speicherteil des PICs programmiert werden soll. Standardmäßig sind hier der Programmspeicher und der

Konfigurationsspeicher ausgewählt.

Es ist möglich, daß die HEX-Datei neben dem Programmcode auch Informationen über die Konfiguration und die Daten des EEPROM-Speichers enthält. Ist der EEPROM-Inhalt im HEX-File enthalten, so erfolgt automatisch die Aktivierung der Option „Datenspeicher (EEPROM)“.

Im Feld „Konfiguration“ wird die Konfiguration des PICs vorgegeben. Sind diese Informationen im HEX-File enthalten, so wechseln die Einstellungen auf die vorgegebenen Werte.

Nachdem man die Optionen und die Konfiguration angepaßt hat, kann durch den „Programmieren“-Button die Programmierung des PICs gestartet werden. Es erfolgt das automatische Beenden des Emulatormodes, das Programmieren des PICs und anschließend das Zurückschalten in den Emulatormode.

Mit dem „Laden“-Button ist das HEX-File neu einlesbar. Diese Funktion ist besonders im Entwicklungsstadium sehr nützlich, da hier meist die PIC-Entwicklungs-umgebung parallel läuft und damit ein neues HEX-File erzeugt wird.

So kann man den neuen Programmcode einfach über den „Laden“-Button neu einlesen lassen, ohne die Datei erneut auswählen zu müssen.

Der Button „Auslesen“ ermöglicht das Auslesen eines PIC-Speicherinhalts, wobei nur die Bereiche gelesen werden, die unter „Option“ aktiviert sind. Nach dem Austausch des PICs ist dann mit dem „Programmieren“-Button der Speicherinhalt bequem in einen weiteren PIC übertragbar.

Schaltung

Die Schaltung des PIC16F84-Emulators/Programmers ist in Abbildung 33 dargestellt.

Der PIC 16F84 wird in den Testsockel IC 1 gesetzt, dessen Anschlüsse zum Teil direkt mit der Stiftleiste ST 1 verbunden sind. Hier ist wiederum die Flachbandlei-

tung mit dem IC-Adapter für das Einsetzen in die Anwendungsschaltung angeschlossen.

Die Betriebsspannung kann man dem Zielsystem entnehmen, wobei dann der Strom über die Diode D 7 vom Typ SB120 fließt. Zur Signalisierung der Betriebsbereitschaft dient die Leuchtdiode D 6.

Ebenso kann das Gerät über ein Steckernetzgerät betrieben werden, das an die 3,5mm-Klinkenbuchse BU 2 anzuschließen ist. Die Diode D 4 schützt das Gerät bei versehentlicher Verpolung. Mit dem Spannungsregler IC 6 vom Typ 7805 erfolgt die Stabilisierung der 5V-Betriebsspannung. Die Diode D 5 vermeidet einen Rückfluß des Stroms zum Spannungsregler, wenn das Gerät aus dem Zielsystem gespeist wird.

Die Steuerung und die Programmierung des IC 1 erfolgt durch den Mikrocontroller IC 3 vom Typ ELV9888. Hierbei handelt es sich um einen bereits programmierten Mikrocontroller vom Typ AT89C2051.

Dessen Oszillator ist mit den externen Komponenten Q 1, C 3 und C 4 aufgebaut, die Bauteile C 1, D 2 und R 2 bilden die RESET-Schaltung.

Der Pegelwandler IC 4 vom Typ MAX232 dient zur Anpassung der Pegel des Mikrocontrollers IC 3 an die Pegel der seriellen PC-Schnittstelle.

Da der PIC 16F84 während der Programmierung mit einer 12V-Programmier-spannung am Reset-Pin (Pin 4) beschaltet werden muß, ist diese Spannung intern zu erzeugen. Dazu bildet das IC 5 vom Typ MAX662 gemeinsam mit den externen Kondensatoren C 9 und C 10 eine Spannungspumpe, die am Ausgang (Pin 6) eine 12V-Spannung generiert.

Diese versorgt den Analogschalter IC 2 vom Typ CD4053, der die zur Programmierung benötigten Pins des IC 1 umschaltet. Im Emulatormode steuert IC 3 die Transistoren T 3 und T 2 vom Typ BC548 durch, so daß sich die Umschalter des IC 2 im Ruhezustand befinden. Dabei sind die

Tabelle 8
Technische Daten:

Versorgungsspannung:
aus dem Zielsystem: 5 V DC
über Steckernetzgerät: 9 V bis 15 V DC
Stromaufnahme: ca. 45 mA
Datenübertragung
zum PC: seriell, 9600 Baud
gerade Parität, 8 Datenbits und 1 Stopbit

Bild 32:
Ansicht der
Bedien-
software

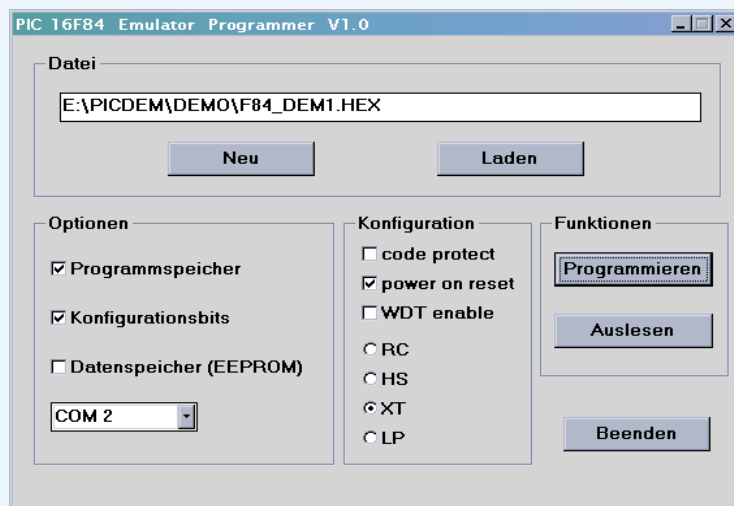
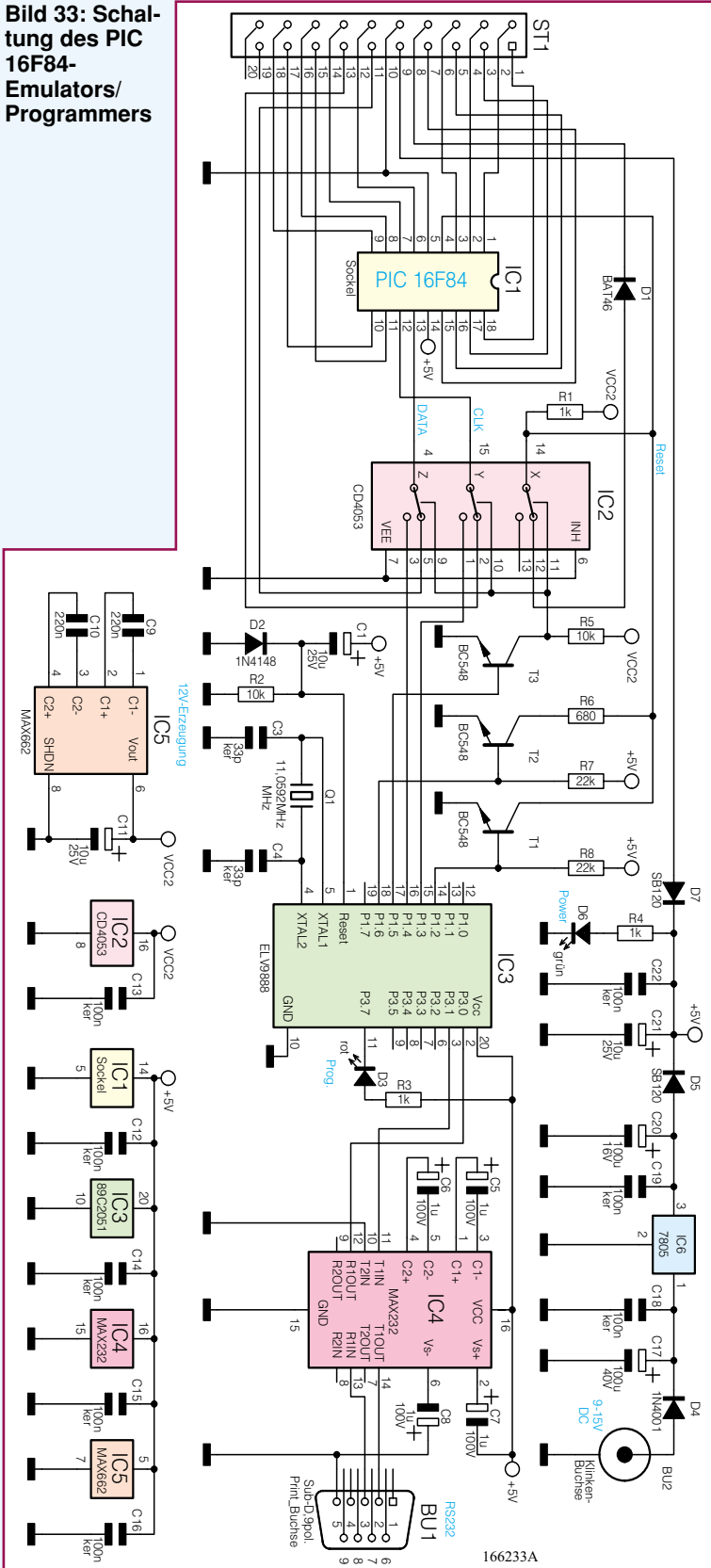


Bild 33: Schaltung des PIC 16F84-Emulators/ Programmers



CLK- und DATA-Leitungen mit ST 1 verbunden sind und als normale Portleitungen nutzbar. Die Reset-Leitung führt über den Spannungsteiler, bestehend aus R 1 und R 6, genau 5 V. Durch kurzzeitiges Durchsteuern des Transistors T 1 wird die Reset-Leitung des PICs nach Masse gezogen, woraufhin dieser einen Reset ausführt und das Programm neu startet. Der Reset ist

auch extern auslösbar, indem die Resetleitung vom Zielsystem aus über die Diode D 1 nach Masse gezogen wird.

Um nun in den Programmiermode zu wechseln, schaltet der Mikrocontroller zuerst den Transistor T 1 durch, woraufhin der PIC in den Reset-Zustand geht und alle Ports als Eingänge geschaltet sind. Daraufhin wird der Transistor T 3 gesperrt, und es

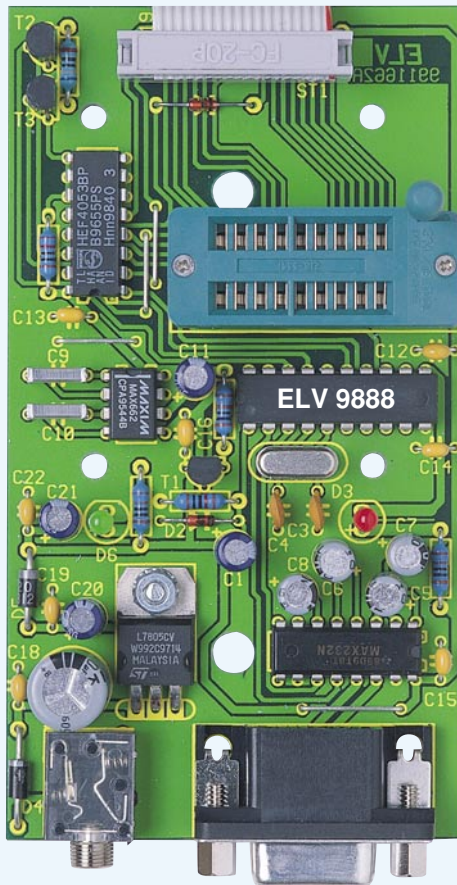
Stückliste: PIC16F84-Emulator-Programmer

Widerstände:
 680Ω R6
 1kΩ R1,R3,R4
 10kΩ R2,R5
 22kΩ R7,R8

Kondensatoren:
 33pF/ker C3,C4
 100nF/ker C12-C16,C18, C19,C22
 220nF C9,C10
 1µF/100V C5-C8
 10µF/25V C1,C11,C21
 100µF/40V C17
 100µF/16V C20

Halbleiter:
 PIC16F84-04 IC1
 CD4053 IC2
 ELV9888 IC3
 MAX232 IC4
 MAX662 IC5
 7805 IC6
 BC548 T1,T2,T3
 BAT46 D1
 1N4148 D2
 1N4001 D4
 SB120 D5,D7
 LED, 3 mm, rot D3
 LED, 3 mm, grün D6

Sonstiges:
 Quarz, 11,0592 MHz Q1
 Stiftleiste, 2 x 10polig ST1
 SUB-D-Buchsenleiste, 9polig, print BU1
 Klinkenbuchse, 3,5 mm, stereo BU2
 2 Buchsenleisten, 15polig
 1 Textool-Sockel, 18polig
 1 IC-Adapter, 18polig
 1 Zylinderkopfschraube, M3 x 6mm
 1 Mutter, M3
 1 Fächerscheibe, M3
 4 Knippingschrauben, 2,9 x 6,5mm
 1 Gehäuse, bearbeitet und bedruckt
 1 3,5" Diskette, PIC16F84-Emulator-Programmer-Software
 10 cm Flachbandleitung, 20polig
 15 cm Schaltdraht, blank, versilbert
 1 Pfostenverbinder, 20polig

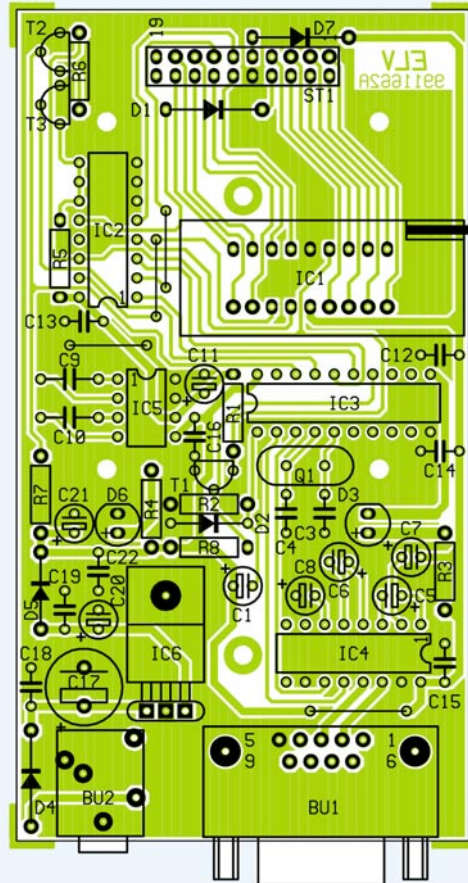


DATA-Leitungen, deren Steuerung der Mikrocontroller IC 3 übernimmt.

Nachbau

Der Nachbau erfolgt in gewohnter Weise anhand des Bestückungsplans und der Stückliste, wobei die Bauteile bestückt, verlötet und die überstehenden Drahtenden abgeschnitten werden. Das Bestücken beginnt mit den flachen Bauelementen wie Brücken, Widerständen, Dioden (Polung beachten) und Kondensatoren, gefolgt vom polrichtigen Einsetzen der Elkos und der lagerichtigen Bestückung der ICs. Vor dem Anlöten des Spannungsreglers IC 6 sind dessen Anschlußdrähte um 90° nach hinten abzuwinkeln und das Gehäuse mit einer M3x6 mm-Zylinderkopfschraube, einer Zahnscheibe und einer M3-Mutter zu verschrauben. Am Bestückungsplatz des Testsockels IC 1 sind zuerst Buchsenleisten zu montieren (zuvor auf 9 Buchsen kürzen). In die Buchsenleisten wird dann nach der Gehäusemontage der Testsockel eingesetzt. So wird erreicht, daß der Testsockel später nach der Gehäusemontage bequem von außen erreicht und bedient werden kann. Die LEDs sind in einem Abstand von 13 mm, gemessen von der Leiterplattenoberfläche bis zur Unterseite des Gehäusekörpers, einzulöten (abgeflachte Gehäusesseite ist die Katode).

Im nächsten Arbeitsschritt ist die Flachbandleitung gemäß Abbildung 34 anzufertigen.



tigen. Dazu müssen zuerst von der 20poligen Flachbandleitung zwei graue Adern abgetrennt werden.

Auf der einen Seite ist dann ein 20poliger Pfostenverbinder aufzupressen, wobei die rote Leitung an der Seite, die mit einem Pfeil gekennzeichnet ist, angelegt wird. Die unteren zwei Schneidklemmen der Buchsenleiste bleiben dabei frei. Auf das andere Ende der Flachbandleitung ist dann der IC-Adapter aufzuquetschen. Hier muß das Einsetzen der roten Leitung auf der mit „1“ gekennzeichneten Seite des Adapters erfolgen. Nach der Fertigstellung der kompletten Flachbandleitung ist der Pfostenverbinder auf die entsprechende Stiftleiste der Leiterplatte zu stecken (rote Ader an Pin 1 der Stiftleiste, siehe Bestückungsdruck).

Damit ist der Aufbau der Leiterplatte abgeschlossen, und es erfolgt der Einbau in das fertig bearbeitete Gehäuse. Dazu wird die Leiterplatte in das Unterteil des Gehäuses eingesetzt und mit vier 2,9x6,5mm-Knippingschrauben befestigt. Abschließend ist das Gehäuseoberteil aufzusetzen (LEDs vorsichtig einfädeln), mit den dazugehörigen Schrauben zu befestigen und die Testfassung für den PIC einzusetzen.

Ansicht der fertig bestückten Platine mit zugehörigen Bestückungsplan

Schlußwort

Damit endet die Artikelserie, die die Grundlagen zur PIC-Programmierung vermittelt hat. Dabei wurden einige ausgewählte Mikrocontroller ausführlich beschrieben und anhand von kleinen Beispielen der Umgang mit der PIC-Entwicklungs-umgebung erläutert. Somit ist es auch einem Anfänger, der sich zum ersten Mal mit der Programmierung von Mikrocontrollern beschäftigt, möglich, selbst ein Programm zu erstellen.

Die PIC-Mikrocontroller-Reihen bieten jedoch noch eine Vielzahl von Derivaten - für fast jede Anwendung den passenden Typ. So gibt es zum Beispiel kleine PICs in einem 8poligen Gehäuse, mit denen Kleinanwendungen realisierbar sind. Aber auch für Anwendungen mit hohen Anforderungen enthält die PIC-Serie Controller mit größerem Pro-

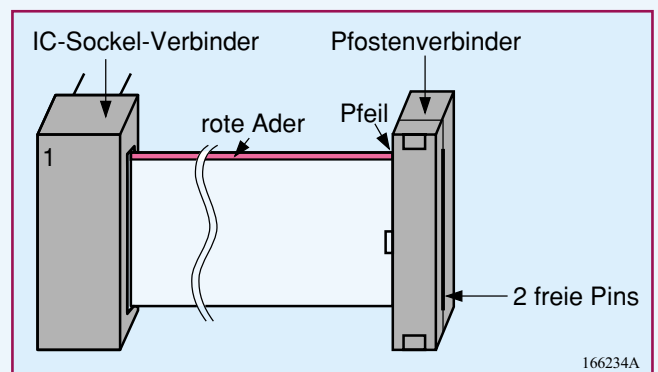


Bild 34: Flachbandleitung

gramm- und Datenspeicher.

Wer sich mit den Grundlagen der PICs auseinandergesetzt hat, wird nach Durchsicht der entsprechenden Datenblätter in der Lage sein, die Besonderheiten der einzelnen Typen zu berücksichtigen und diese zu programmieren.

Ebenso lohnt es sich, zum Beispiel im Internet nach fertigen Applikationen oder ProgrammROUTINEN Ausschau zu halten, von denen man Teile in eigene Anwendungen übernehmen kann. Als erste Adresse ist hier die Homepage vom Microchip (<http://www.microchip.com>) zu nennen, auf der aktuelle Informationen zu den PICs sowie Beispiele und fertige Applikationen zu finden sind. Auf diesem Weg ist auch die aktuellste Version der PIC-Entwicklungssoftware herunterladbar. **ELV**