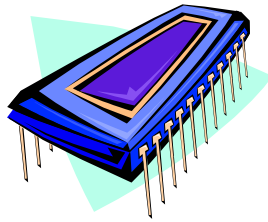
PICPRO



PIC CHIP PROGRAMMER

Ramsey Electronics Model No. PIC1

You've seen all the super neat projects using the popular PIC series of microcontrollers. Here's a really easy to use programmer to get you up and running fast!

- **Featured as the cover story in Electronics Now magazine.**
- **Programs PIC F83, F84 microcontrollers.**
- **Easy hookup – connects to computer parallel port.**
- **Even includes a simple first project to use your PICPRO programmer on.**
- **Clear, concise instructions guide you step by step.**

PARTIAL LIST OF AVAILABLE KITS

RAMSEY TRANSMITTER KITS

- **FM10, FM25B FM Stereo Transmitters**
- **FM100 Super Pro FM Transmitter**
- **MR6 Model Rocket Tracking Transmitter**
- **AM1, AM25 AM Transmitters**

RAMSEY RECEIVER KITS

- **FR1 FM Broadcast Receiver**
- **AR1 Aircraft Band Receiver**
- **SR2 Shortwave Receiver**
- **HFRC 10 MHz WWV Receiver**
- **SC1 Shortwave Converter**

RAMSEY HOBBY KITS

- **SG7 Personal Speed Radar**
- **SS70A Speech Scrambler**
- **TG1 DTMF Tone Grabber**
- **BS1 "Bullshooter" Digital Voice Storage Unit**
- **AVS10 Automatic Sequential Video Switcher**
- **WCT20 Cable Wizard Cable Tracer**
- **MD3 Microwave Motion Detector**
- **ML Music Lights Kit**
- **LC1 Inductance-Capacitance Meter**

RAMSEY AMATEUR RADIO KITS

- **HR Series HF All Mode Receivers**
- **QRP Series HF CW Transmitters**
- **CW7 CW Keyer**
- **QRP Power Amplifiers**

RAMSEY MINI-KITS

Many other kits are available for hobby, school, scouts and just plain FUN. New kits are always under development. Write or call for our free Ramsey catalog. .

Ramsey Electronics publication No. MPICPRO Revision 1.2
First printing: April 1999

Ramsey Publication No. MPIC1
Price $5.00

KIT ASSEMBLY AND
INSTRUCTION MANUAL FOR

PICPRO
PIC-CHIP MICROCONTROLLER PROGRAMMER

TABLE OF CONTENTS

RAMSEY ELECTRONICS, INC.
590 Fishers Station Drive
Victor, New York 14564
Phone (585) 924-4560
Fax (585) 924-4555

Overview and Specifications

PICPRO is Ramsey Electronics' programmer for the PIC16C84, PIC16F84, and PIC16F83 microcontrollers made by Microchip, Inc. These are a few of a large line of PIC microcontrollers; their special feature is that they use flash memory (so no UV eraser is needed) and are programmed by a serial data protocol (making the programmer circuit simple).

PICPRO requires a PC-compatible computer with one parallel port available. It runs under DOS 3.0 or higher, and in a DOS box under Windows 95, Windows 98 (not NT) or OS/2.

Required software for the Ramsey PICPRO programmer is available in the download section of the Ramsey Electronics web site at http://www.ramseyelectronics.com. This software includes the PICPRO run software, editor software, programming software, and assembler software. In addition, an enhanced Win95/98 based program development system called MPLAB, is available free of charge from the manufacturer of PIC microcontrollers at http://www.microchip.com.

PICPRO is based on NOPPP, a PIC programmer designed by Michael Covington and featured in *Electronics Now,* September 1998. For current information on NOPPP and related projects, see the NOPPP web page at http://www.mindspring.com/~covington/noppp. The NOPPP software is 100% compatible with PICPRO.

RAMSEY Parts List

CAPACITORS:

- ❑ 1 .001 uf disc capacitor (marked .001 or 102 or 1 nf) [C8]
- ❑ 3 .1 uf disc capacitor (marked .1 or 104) [C5,6,7]
- ❑ 3 10 uf electrolytic capacitor [C2,3,4]
- ❑ 1 100 uf electrolytic capacitor [C1]

RESISTORS:

- ❑ 10 330 ohm (orange-orange-brown) [R6,7,10,11,12,13,14,15,16,17]
- ❑ 3 1K ohm (brown-black-red) [R3,4,5]
- ❑ 1 2.2K ohm (red-red-red) [R1]
- ❑ 3 4.7K ohm (yellow-violet-red) [R2,8,9]
- ❑ 1 10K ohm (brown-black-orange) [R18]

SEMICONDUCTORS:

- ❑ 1 2N3904 NPN transistor (marked 2N3904) [Q1]
- ❑ 2 1N270 diode [D1,2]
- ❑ 1 1N4148 diode (marked 4148) [D4]
- ❑ 1 Jumbo red LED [D3]
- ❑ 8 Mini red LED [D5,6,7,8,9,10,11,12]
- ❑ 1 78L12 voltage regulator, TO-92 package [VR1]
- ❑ 1 78L05 voltage regulator, TO-92 package [VR2]

MISCELLANEOUS:

- ❑ 1 2.1mm power jack [J1]
- ❑ 1 DB25 connector [J2]
- ❑ 1 18 pin IC socket [U2]
- ❑ 1 18 pin IC socket [U3]
- ❑ 1 DPDT pushbutton switch [S1]
- ❑ 17 inches insulated wire
- ❑ 2 1 inch standoffs
- ❑ 4 #4 screws

Construction:

In all the following steps the word install means to:

- 1) Correctly locate and identify the part called for in the instruction step.
- 2) Carefully insert the part into the proper holes on the PC board.
- 3) Gently bend the component's leads slightly to secure it to the PC board.
- 4) Turn the PC board over and solder the component leads to the board.
- 5) Trim or nip off the excess lead length after soldering.
- 6) Check off the step when completed.

Tips on soldering:

Correct soldering is not difficult and is fun to do. Projects such as this one are easily assembled using a soldering pencil iron of 25 - 50 watts. The most important thing to remember in making good joints is to use enough heat to allow the solder to flow smoothly around the entire connection area. The component lead wire and the PC board trace area should be in contact with the iron and then the solder added. This will cause the solder to flow completely between the PC trace and lead wire. A proper solder connection should look shiny and smooth.

Now that the lessons are out of the way, let's get building!

Please pay close attention to the instruction steps and the layout diagram. The printed parts outline on the PC board is the component side, while the unprinted side is the solder side. More simply, if you are unsure of what side to put a component on, the component always goes on the side with the printed outline, and the leads are soldered on the other. We'll begin with the larger of the 3 PC boards.

- ❑ 1. Install S1, the DPDT pushbutton switch . It fits correctly only one way. Ensure that the white plastic switch extends out over the edge of the printed circuit board and the switch is placed flat against the PC board.
- ❑ 2. Install J1, the 2.1mm power jack . Solder all three points of the jack securely.
- ❑ 3. Install VR1, the 78L12 voltage regulator. Make sure the flat side of the component is placed as shown on the layout diagram.

- ❑ 4. Install D4, the 1N4148 diode (marked 4148). The diode must be placed with the black band in the position shown on the PC board layout. Be careful to orient this diode correctly.
- ❑ 5. Install C1, a 100 uf electrolytic capacitor. Electrolytic capacitors have a right and a wrong way to be installed. Usually capacitors have a wide black stripe which indicates their negative lead and the PC board or parts layout diagram will show the positive side. Observe correct polarity when installing C1.
- ❑ 6. Install R6, one of the 330 ohm resistors (orange-orange-brown).
- ❑ 7. Install VR2, the 78L05 5 volt regulator. Be sure to orient the flat side as shown on the PC board layout.
- ❑ 8. Install C2, one of the 10 uf electrolytic capacitors. Pay close attention to the proper position of the positive lead of the capacitor.
- ❑ 9. In the same manner, install C3 and C4, the remaining 10 uf electrolytic capacitors. Be sure to orient these electrolytic capacitors as shown on the parts layout diagram.
- ❑ 10. Install D3, the jumbo red LED. The long lead of the diode should be placed in the hole marked with a "+" symbol.
- ❑ 11. Install C5, .1 uf disc capacitor (marked .1 or 104).
- ❑ 12. Install Q1, the 2N3904 NPN transistor (marked 2N3904).Be sure you have properly positioned the flat side of the transistor.
- ❑ 13. Install R1, a 2.2K ohm resistor (red-red-red).
- ❑ 14. It is now time to install JMP 1. Using a scrap component lead, form this jumper and install as you would a resistor. Jumpers act like small "bridges" to route traces over obstacles (other traces).
- ❑ 15. Install C6, a .1 uf capacitor (marked .1 or 104).
- ❑ 16. Install R8, 4.7K ohms (yellow-violet-red).
- ❑ 17. Install R9, another 4.7K ohm resistor (yellow-violet-red).
- ❑ 18. Install R4, 1K ohm (brown-black-red).
- ❑ 19. Install R3, another 1K ohm resistor (brown-black-red).
- ❑ 20. Install R5, the last 1K ohm resistor (brown-black-red).
- ❑ 21. Install D2, a 1N270 diode (marked 1N270). Be sure to orient the black band as shown on the PC board silkscreen and parts layout diagram.
- ❑ 22. Install R2, 4.7K ohm resistor (yellow-violet-red).
- ❑ 23. Install D1, a 1N270 diode (marked 1N270). Again, be sure to orient the black band as shown on the PC board silkscreen and parts layout diagram.
- ❑ 24. Install R7, 330 ohms (orange-orange-brown).

- ❑ 25. Install J2, the DB25 connector. Be sure the connector is seated flush to the PC board and all 25 pins are soldered.

The only things left to install on this board are the wires, labeled TP1A through TP3A.

- ❑ 26. Prepare the wire by cutting it into 5 pieces 2 inches long. Strip each end of each piece and tin the wire by applying solder to the stripped ends. Install one of the stripped and tinned wires in the TP1A, position.
- ❑ 27. In the same way, install the remaining four wires (TP2A through TP5A) and solder them in.

We will now move on to the next PC board and install the components. This board is labeled PICDEMO-1 on the topside silkscreen.

- ❑ 28. Install C7, .1 uf disc capacitor (marked .1 or 104).
- ❑ 29. Install R18, 10K ohms (brown-black-orange).
- ❑ 30. Install C8, .001 uf disc capacitor, marked .001, 102 or 1 nf.
- ❑ 31. Install U2, the 18 pin IC socket. While the socket can be installed either way, if there is a dot or a notch on the socket you will want to line that up with the notch on the PC board silkscreen. This will make it easier to orient your IC's when installing them in the socket. Be sure all 18 pins are fully through the PC board before soldering in the part.
- ❑ 32. It is now time to install R10 through R17, all 330 ohm (orange-orange-brown). Insert and bend the leads on each resistor then solder them all in.
- ❑ 33. Here's another group of identical parts to install; D5 through D12, mini red LEDs. Mark each of the LEDs so that they will sit about 1/4 inch above the PC board. This will make it easier to get them all installed evenly when you solder them in. The long leads are the positive and should be placed in the holes closest to the resistors R10 – R17. Insert them through the PC board and bend the leads so that the LEDs are all approximately the same height (1/4 inch) and solder them in.

We can now move on to the last PC board. This is marked PICEXTN-1 and contains the fewest parts.

- ❑ 43. Install U3, the 18 pin socket. Install the socket so that the notch is to the right when the text [RAMSEY ELECTRONICS] is right side up. Be sure to solder all 18 pins.
- ❑ 44. We will now connect the board we've been working on to the main

PC board. Start by resting the small board [PICEXTN-1] on the main [PICPRO-1] PC board so that the pads labeled TP1A – TP5A are lined up with the pads labeled TP1B – TP5B. The wires will have to be soldered on the bottom of the small board so once the wires have been matched up to their respective holes (TP1A to TP1B and so forth), the board will need to be flipped over so that the solder pads can be reached. The wires can be installed one at a time if that seems simpler. Solder each wire into its proper position then trim off the excess wire on the top side of the board.

- ❑ 45. Install the 1 inch standoffs on the main board using the #4 screws.
- ❑ 46. Flip the small PC board over and screw it to the standoffs using the remaining #4 screws.

You have completed the assembly of your PICPRO programmer! We assume that you are acquainted with PIC assembly language. If not, don't panic – you will be able to do the demonstrations here even if you don't know any assembly language yet, provided you know how to edit text files.

Installing software on your PC

You will need to download the most current PICPRO operating software from the download section of the Ramsey Electronics web site, http://www.ramseyelectronics.com.

Minimum Operating File

pic1.zip	PICPRO run software

Additional Feature Software Files

noppp.zip	NOPPP No Parts PIC Programmer Software
asmb213.zip	PIC Microcontroller Assembler
topic.zip	David Tait's freeware software package compatible with NOPPP and PICPRO

To download the file(s) go to the download page of the Ramsey web site (http://www.ramseyelectronics.com) and select the desired file, and save it to your hard drive.

Throughout this manual, we will assume you have downloaded the run files contained in pic1.zip, and have unzipped them into a directory on your PC called `c:\PICPRO`

The PICPRO run software is DOS executable file *picpro-1.exe* which would be located in your directory `c:\PICPRO` (or wherever you saved it). (Note: If you're using OS/2, be sure to enable access to hardware timers [HW_TIMER] in the properties of the DOS box.)

Connecting PICPRO to your PC

Static electricity hazard: PIC microcontrollers are sensitive to static electricity that can accumulate on your body without your knowing it. So is the parallel port of your PC. Before coming into contact with these items, ground yourself by briefly touching your desk or your PC. If humidity is low, spray anti-static compound (or, in a pinch, a fine mist of water) on the carpet and furniture where you are working.

Connect PICPRO to the parallel port of your computer using the cable supplied or any other cable that has all the appropriate pins connected. Preferably, the cable should be short (about 2 feet); some parallel ports will drive a longer cable reliably and some won't.

Serial port cables are not suitable; even if they fit the connectors, they are missing some of the necessary wires.

Make sure you know which parallel port you are using (LPT1, LPT2, or LPT3). If you don't, you can find out by trial and error when you run the software. Also, if you use a networked printer, make sure that this parallel port has not been redirected to the network by the operating system.

Testing PICPRO

Although electronic problems are uncommon when PICPRO is built from a kit, you can test the circuit with a voltmeter to verify correct wiring. To do this, start the PICPRO software:

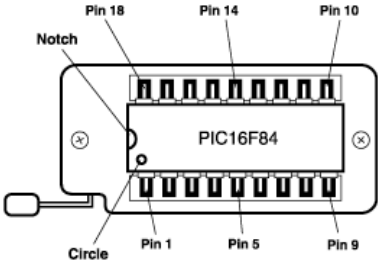
```
C:\PICPRO> picpro-1
```

Then select "T" (test) on the device menu and follow the instructions. Do not panic if some voltages are slightly out of specification; be on the lookout for large errors, such as 0 volts where you expect 5 volts.

Programming your first PIC

The software contains two already-assembled programs, DEMOF84 (for the

PIC16F84) and DEMOC84 (for the PIC16C84). Choose the one that corresponds to the PIC you're using, and you're ready to program your first PIC. Start the PICPRO software on your PC. Select the appropriate parallel port and turn on the programmer when the software tells you to do so. When asked which type of PIC you are programming, type F for PIC16F84 (or C if you are using a PIC16C84). Then insert the PIC into the socket.
Be sure to insert the PIC the right way around as shown in Figure 1. It will be ruined if you insert it backward. Note that pin 1 of the PIC is marked with a notch and/or a circle. Pin 1 must always be positioned toward the handle of the socket.



At this point you will see the main menu of PICPRO, which looks like this:

```
PICPRO-1 by Ramsey Electronics
Michael A. Covington
Version of Jan 2 1999 15:45:18
.............................
Using LPT1 on 378H
.............................
PIC16F84
.............................
L  Load HEX file
S  Select type of PIC
E  Erase PIC
P  Program PIC
V  Verify PIC

X  Exit program

Your choice (L,S,E,P,V,X):
```

Press L to load a file and type the filename `demof84.hex` and press Enter (If you're using a PIC16C84, type `democ84.hex` instead.) Then press E to erase the PIC, P to program it, and V to verify that it is programmed correctly.

You can now exit PICPRO (following instructions on the screen about

removing power and removing the PIC), then insert the programmed PIC in the LED chaser demo board. Apply power (4 to 6 volts), and the LEDs will light, one after another, in endless succession. Note that the power connections to the demo board are two large pads on the foil side of the board; you can use alligator clips to make connections to them.

```
; File DEMOF84.ASM
; Assembly code for PIC16F84 microcontroller

; Blinks LEDs on outputs in a rotating pattern.
; with 75 kHz osc, each LED stays on 1/2 second.

; CPU configuration
;       (It's a 16F84, RC oscillator,
;       watchdog timer off, power-up timer on)

        processor 16f84
        include    <p16f84.inc>
        _ _config   _RC_0SC & _WDT_OFF & _PWRTE_ON

;  Declare variables at 2 memory locations

J       equ    H'1F'    ; J = address hex 1F
K       equ    H'1E'    ; K = address hex 1E

; Program

        org    0       ; start at address 0

        ; Set port B as output and initialize it

        movlw  B'00000000'    ; w := 00000000 binary
        tris   PORTB          ; port B ctrl register := w
        movlw  B'00000001'    ; w := 00000001 binary
        movwf  PORTB          ; port B itself := w

        ; Clear the carry bit

        bcf    STATUS,C

        ; Rotate the bits of port B leftward through the carry bit

mloop:  rlf    PORTB,f

        ; Waste some time by executing nested loops

        movlw  D'50'          ; w := 50 decimal
        movwf  J              ; J := w
jloop:  movwf  K              ; K := w
kloop:  decfsz K,f            ; K = K-1, skip next if zero
        goto   kloop
        decfsz J,f            ; J = J-1, skip next if zero
        goto   jloop
```

```
        ; Do it all again

        goto    mloop

        end
```

Figure 2: Listing of DEMOF84.ASM. DEMOC84.ASM is almost identical.

Assembling a new program

Now you're ready to create a program of your own. Rather than try to teach you PIC assembly language right here, we'll guide you through making a simple modification to the LED chaser program.

File DEMOF84.ASM is shown in Figure 2. Make a copy of it (or DEMOC84. ASM if you're using the 'C84) and call the copy MYDEMO.ASM:

```
C:\PICPRO1> copy demof84.asm mydemo.asm
```

Now open mydemo.asm with your favorite editor. You can use Windows Notepad, the EDIT command in newer versions of DOS, or the editor PRED.EXE that is supplied on the diskette.

```
C:\PICPRO1> edit mydemo.asm or C:\PICPRO1> pred mydemo.asm
```

Find the line in the program that reads

```
        movlw           D'50'   ; w := 50 decimal
```

and change it to

```
        movlw           D'30'   ; w := 30 decimal
```

taking care to leave some blanks to the left of `movlw`.

Now you must assemble MYDEMO.ASM to create MYDEMO.HEX. This is done with the program MPASM. Type the command:

```
C: \PICPRO1> mpasm mydemo.asm
```

to do this. If you prefer, you can type `mpasm` without the filename and you will be prompted to enter the name of the file and other information.

The assembly process will create several files: MYDEMO.HEX (which contains the new data to be programmed into the PIC), MYDEMO.LST (a listing that contains error messages, if any), and a couple of files you can delete called MYDEMO.COD and MYDEMO.ERR.

If there are fatal errors, MYDEMO.HEX is not created and you have to look at MYDEMO.LST to find out what went wrong.

Although your program should assemble without any errors, take a look at MYDEMO.LST anyhow (using EDIT, PRED, or Windows Notepad). If you programmed computers before 1980 or so, you've seen a lot of files like this - it's a traditional compiler listing, designed to be printed, containing every line of the program together with error messages and other commentary from the compiler.

Don't get mixed up and edit the wrong file - remember that although the .LST file contains the error messages, the .ASM file is the one you need to edit in order to correct them!

You'll note that the TRIS PORTB instruction in the program provoked a warning. The reason is that some higher-end PICs do not have this instruction, and although it works fine on the PIC16F84, Microchip is discouraging people from using it. Pay no attention to the warnings; TRISB (along with TRISA) is a perfectly legitimate instruction on the chips that PICPRO uses.

Now start the PICPRO1 program and erase the PIC, then program MYDEMO.HEX into it. The procedure is the same as before. This time, when you put the PIC into the LED chaser demonstration circuit, the action will be significantly faster because you've reduced the value of a loop counter.

Frequently-asked questions

Does PICPRO work with all PCs?

It works with all the PCs we have tried, ranging from a 4.77-MHz 8088 to a 300-MHz Pentium II. There is a very small chance that your parallel port will not drive PICPRO properly. In that case, try a different parallel port or a different PC.

What kind of power supply is needed?

The power supply that comes with PICPRO may be labeled "12 volts," but

under light load it delivers more than 14 volts, and the circuit relies on this. Input to PICPRO should always be at least 14 volts DC.

Where do I get the PIC assembler and simulator?

An assembler for DOS is included in the downloadable PICPRO run files; a much better development system called MPLAB is available free of charge from Microchip, Inc., at http://www.microchip.com. You can download it from the web site or request a copy on CD-ROM.
The complete MPLAB package for Windows, including assembler and simulator/debugger, is about 7 megabytes in size. That's less than an hour of downloading at 28.8 kbaud.

It doesn't matter which version of MPLAB you get; all reasonably recent versions work fine. Recent revisions fix bugs affecting other PICs, not the 16C84/F84/F83.

Which kind of PIC should I buy, and where can I get it?

PICPRO will program any PIC16C84, PIC16F84, or PIC16F83 chip that you can plug into it. The cheapest and most readily available of these is the PIC16F84-04P. Here 04 means the maximum clock speed is 4 MHz (as opposed to 10) and P means it's in a plastic DIP package (not surface mount).

PICPRO is probably also suitable for the low-voltage PICs, PIC16LC84 and PIC16LF84, although it has not been tested extensively with them.

You can buy PICs and PIC-related items from:

- Digi-Key (1-800-DIGIKEY, http://www.digikey.com)
- Jameco (http://www.jameco.com)
- Dontronics (http://www.dontronics.com)

The last of these has a particularly interesting selection of resources for experimenters and hobbyists.

Where can I learn more about PICs?

You can download PIC manuals in PDF form, or request printed manuals, from Microchip, Inc. (http://www.microchip.com).

Beyond that, I particularly recommend the book *Easy PIC'n,* by David Benson, published by Square One (squareone@zapcom.net). Advanced PIC

programming is covered in *Design with PIC Microcontrollers,* by John B. Peatman, published by Prentice-Hall. Intermediate between these is Myke Predko's *Programming and Customizing the PIC Microcontroller,* published by TAB.

You can special-order these through any good bookstore or buy them online at http://www.amazon.com.

Why do you use the TRIS instruction when Microchip says not to?

TRIS is a fully supported instruction in the PIC16C84 and related chips. Microchip discourages people from using it for only one reason: some of the higher-end PICs lack it, and they want you to be able to move up.

I find the TRIS instruction much more convenient than the alternative, and I see no reason not to use it. If I ever port my programs to a higher-end PIC, I'll be able to remove the TRIS instruction, or change it to a macro, at the same time that I make other changes.

Isn't it dangerous to insert the PIC into the socket with power already applied?

After careful thought, I decided it wasn't. Here's why: The usual hazard from inserting an IC into a socket with power already on is that CMOS chips will go into "SCR latchup." This happens when an input is high but the V+ pin is not yet powered up.

In PICPRO, that doesn't happen. When you're told to insert the chip, V+ is powered up (+5V) but all the other pins are near 0 volts.

What do the error messages mean?

"Caution: programmer hardware not found"

This means the PICPRO software is unable to detect the PICPRO hardware. Most likely, you've chosen the wrong parallel port or you have not yet applied power to PICPRO.

"Caution: Configuration word appears to contain invalid bits."

Your .HEX file was assembled for a different PIC (such as 16F84 rather than 16C84). It may not work if programmed into the PIC that you have currently selected.

“Failed at 0000: Expecting (something), found 3FFF.”

You are trying to verify a blank PIC without having programmed it, or . . .

You have a defective PIC, or . . .

You have some kind of data communication problem; the PIC is not receiving commands correctly. Try a shorter cable; check the circuit carefully; run the voltage checks.

Troubleshooting

My PICPRO doesn't work. What should I check?

1. Are all connections correct and properly soldered? (Check with an ohmmeter, or better yet, run the voltage checks in the test mode of the software.)
2. Are you using a fresh or tested PIC and protecting it from static electricity?
3. Have you tried different BIOS settings for the parallel port that you are using? PICPRO should work with any settings, but in difficult cases it's good to try alternatives.
4. Does the cable from your computer to PICPRO actually connect all the lines that are used? *Serial-port cables do not!* Also, my own experience is that *cables that make bad connections are extremely common.* A cable can test out fine with an ohmmeter and still fail to make good connections to some sockets due to slight variation in the length of the pins.
5. Is your cable too long? Try using the shortest cable possible (under 2 feet). Some parallel ports work fine driving PICPRO through much longer cables; some don't.
6. Finally, does your parallel port work? Can you use it to connect your computer to a printer? Some people have had trouble with PICPRO only to discover they were using a dead parallel port (parallel output had been redirected to a network).

MPLAB: Notes for the perplexed

MPLAB is the full-featured PIC development environment that you can download from http://www.microchip.com or obtain on CD from Microchip, Inc., the maker of PIC microcontrollers.

If you are not familiar with advanced software development tools, you may find MPLAB a bit perplexing. These are some brief notes to help you get started. Please note that MPLAB is not a product of Ramsey Electronics or Covington Innovations and any further questions about it should be addressed to its manufacturer.

How to create a project in MPLAB

Everything in MPLAB revolves around the notion of a "project," which is a set of files that need to be processed to assemble (compile) your program.

Note that you must use the __config statement (with two _ marks) to specify whether you are using the RC or crystal oscillator and whether you want the watchdog timer on (usually you don't). There are menus in MPLAB for specifying this configuration information when you are using one of Microchip's programmers that interface directly with MPLAB; they don't apply when you're assembling a .HEX file and then programming the PIC through the PICPRO-1 software.

We'll assume you have just one file, MYPROG.ASM, and from it you want to generate MYPROG.HEX.

Start MPLAB.

Choose Project, New Project.

Go to the directory where MYPROG.ASM resides, and create MYPROG.PJT there. The dialog box will say `*.pjt` and you type "`myprog`" in place of *.

A diagram of the structure of the project will appear. It contains only one item, `myprog[.hex].` This is a "node," i.e., a file to be created by MPLAB.

Click *once* on `myprog[.hex]` and add the node `myprog.asm` under it. Then click OK a couple of times to get back to the main menu.

Choose Project, Save Project.

Then go to File, Open, and open the file MYPROG.ASM.

How to assemble (compile) your program

Now it's time to tell MPLAB a bit more about what you're doing.

Go to Options, Development Mode, and choose MPSIM Simulator and tell it what kind of PIC you're using. Also go to Options, Processor Setup, and make sure everything is right.

Edit your file ad libitum, save it, and then go to Project, Build All. That means "do all the assembling and other things needed to get the .hex file into final form."

The assembler will run and some messages will appear on the screen:

- "Use of this instruction is not recommended" is normal for TRIS instructions.
- "Label found..." is serious: it means you misspelled an instruction. (The assembler may not *think* it's serious, but it is.)
- "MPLAB is unable to find the output file MYPROG.HEX" means the assembly failed due to errors. Correct the errors and try again.

You can double-click on any error message to go to the line it corresponds to.

You can also view the entire source listing (LST file), with messages in context, by choosing Window, Absolute Listing. Get into the habit of doing this. Close the "Build Results" window when you are through with it.

How to step through your code

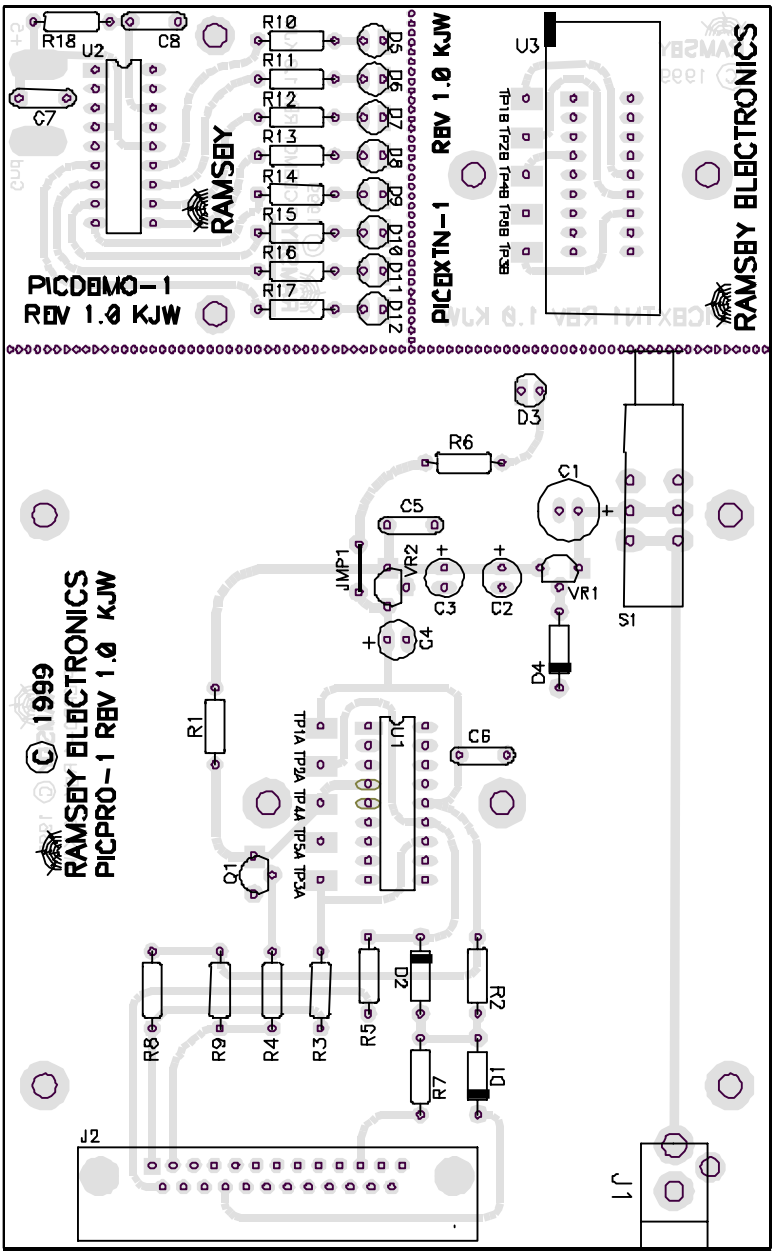
Once you have the project successfully assembled, you can simulate execution of it. Be sure you're in MPSIM simulator mode (see previous page). You will be using the Debug menu to run the program. But you must also set up some memory locations to be displayed, or you won't see anything. Here is one of *many* things you can do in the simulator:

- Choose Window, File Registers, to display a map of RAM.
- Choose Window, Special Function Registers, to display the registers.
- Go to any assembly instruction in the absolute listing window, right-click on it, and choose Run To Here.

The program will execute to the line you chose and then stop, displaying the state of memory.

From there, you can do another Run To Here, or step through the code, or reset the program, using the Debug, Run menu.

PICPRO PARTS LAYOUT DIAGRAM



How to program your code into a PIC

For PICPRO users, that's the easy part, at least in a backhanded sense of "easy."

If you had a Microchip programmer such as a PICstart Plus, you could turn it on from within MPLAB and do your programming. But PICPRO doesn't work that way. Exit MPLAB, find your .HEX file, and use the PICPRO software (at a DOS prompt) to do the programming.

Common errors in assembly language

Assembly-language programming is not easy, and careful attention to the CPU architecture is required. Sometimes, though, a program looks perfectly correct and yet doesn't work. Here are some common errors:

- Starting a statement at the beginning of the line. The only things that can appear in the very first column are a label (on a statement or `EQU`) or a comment. The actual instruction always starts in column 2 or later.
- Failing to mark numbers as decimal (`D'199'`), hex (`H'C7'`), or binary (`B'11000111'`). Unmarked numbers are usually taken as hex, but it's best to mark the base (radix) of all numbers in programs.
- Trying to count higher than 255 in a 1-byte counter. (If you're accustomed to 80386 assembly language, it's easy to forget that the PIC is an 8-bit processor.)
- Failing to look at the LST file to see your error messages - or mixing up the LST and ASM files and trying to make corrections in the LST.
- Writing only one underscore mark in `__config`. It requires two.
- Leaving out the `__config` command. PICPRO-1 does not give you any other way to specify the configuration, and the default values may not be what you want.

Michael A. Covington
Microcontroller consultant
Covington Innovations
285 Saint George Drive
Athens, Georgia 30606
http://www. mindspring.com/~covington

Notes:

Notes:

Notes:

The Ramsey Kit Warranty

Please read carefully BEFORE calling or writing in about your kit. Most problems can be solved without contacting the factory.

Notice that this is not a "fine print" warranty. We want you to understand your rights and ours too! All Ramsey kits will work if assembled properly. The very fact that your kit includes this new manual is your assurance that a team of knowledgeable people have field-tested several "copies" of this kit straight from the Ramsey Inventory. If you need help, please read through your manual carefully, all information required to properly build and test your kit is contained within the pages!

1. DEFECTIVE PARTS: It's always easy to blame a part for a problem in your kit, Before you conclude that a part may be bad, thoroughly check your work. Today's semiconductors and passive components have reached incredibly high reliability levels, and it's sad to say that our human construction skills have not! But on rare occasions a sour component can slip through. All our kit parts carry the Ramsey Electronics Warranty that they are free from defects for a full ninety (90) days from the date of purchase. Defective parts will be replaced promptly at our expense. If you suspect any part to be defective, please mail it to our factory for testing and replacement. Please send only the defective part (s), not the entire kit. The part(s) MUST be returned to us in suitable condition for testing. Please be aware that testing can usually determine if the part was truly defective or damaged by assembly or usage. Don't be afraid of telling us that you 'blew-it', we're all human and in most cases, replacement parts are very reasonably priced.

2. MISSING PARTS: Before assuming a part value is incorrect, check the parts listing carefully to see if it is a critical value such as a specific coil or IC, or whether a RANGE of values is suitable (such as "100 to 500 uF"). Often times, common sense will solve a mysterious missing part problem. If you're missing five 10K ohm resistors and received five extra 1K resistors, you can pretty much be assured that the '1K ohm' resistors are actually the 'missing' 10 K parts ("Hum-m-m, I guess the 'red' band really does look orange!") Ramsey Electronics project kits are packed with pride in the USA. If you believe we packed an incorrect part or omitted a part clearly indicated in your assembly manual as supplied with the basic kit by Ramsey, please write or call us with information on the part you need and proof of kit purchase

3. FACTORY REPAIR OF ASSEMBLED KITS:

To qualify for Ramsey Electronics factory repair, kits MUST:

1. NOT be assembled with acid core solder or flux.
2. NOT be modified in any manner.
3. BE returned in fully-assembled form, not partially assembled.
4. BE accompanied by the proper repair fee. No repair will be undertaken until we have received the MINIMUM repair fee (1/2 hour labor) of $25.00**,** or authorization to charge it to your credit card account.
5. INCLUDE a description of the problem and legible return address. DO NOT send a separate letter; include all correspondence with the unit. Please do not include your own hardware such as non-Ramsey cabinets, knobs, cables, external battery packs and the like. Ramsey Electronics, Inc., reserves the right to refuse repair on ANY item in which we find excessive problems or damage due to construction methods. To assist customers in such situations, Ramsey Electronics, Inc., reserves the right to solve their needs on a case-by-case basis.

The repair is $50.00 per hour, regardless of the cost of the kit. Please understand that our technicians are not volunteers and that set-up, testing, diagnosis, repair and repacking and paperwork can take nearly an hour of paid employee time on even a simple kit. Of course, if we find that a part was defective in manufacture, there will be no charge to repair your kit (But please realize that our technicians know the difference between a defective part and parts burned out or damaged through improper use or assembly).

4. REFUNDS: You are given ten (10) days to examine our products. If you are not satisfied, you may return your unassembled kit with all the parts and instructions and proof of purchase to the factory for a full refund. The return package should be packed securely. Insurance is recommended. Please do not cause needless delays, read all information carefully.

PICPRO PIC Chip Programmer
Quick Reference Page Guide

REQUIRED TOOLS

- Soldering Iron Ramsey WLC100
- Thin Rosin Core Solder Ramsey RTS12
- Needle Nose Pliers Ramsey MPP4 or RTS05
- Small Diagonal Cutters Ramsey RTS04

<OR> Technician's Tool Kit TK405

ADDITIONAL SUGGESTED ITEMS

- Holder for PC Board/Parts Ramsey HH3
- Desoldering Braid Ramsey RTS08
- Digital Multimeter Ramsey M133

TOTAL SOLDER POINTS
70

ESTIMATED ASSEMBLY TIME

Beginner ..... 3 hrs
Intermediate ..... 2 hrs
Advanced..... 1 hrs

Price: $5.00
Ramsey Publication No. MPIC1
Assembly and Instruction manual for:
RAMSEY MODEL NO. PIC1

RAMSEY ELECTRONICS, INC.
590 Fishers Station Drive
Victor, New York 14564
Phone (585) 924-4560
Fax (585) 924-4555
www.ramseykits.com