

The PicBASIC - PicStic Development Package

The PicStic development package consists of two bundles. The first includes the microEngineering Labs software. The second includes hardware provided by Micromint.

The microEngineering software bundle

1. PicBASIC Compiler & PIC Macro Assembler manual

This manual describes the use of the compiler and assembler. A section is also included which lists the assembly instructions for the PIC series of microcontrollers.

2. PicStic Development System diskette

The diskette includes the microEngineering Labs PicBASIC compiler, Macro Assembler, sample PicBASIC code, and the EPIC software for Micromint's PicStic Programmer.

The Micromint hardware bundle

1. The PicStic Programmer kit (may come assembled)

This package includes the parts necessary to assemble the programmer. A few additional items which are needed to use the programmer are included in the Micromint hardware bundle. The programmer parts list, assembly instructions, and schematic are packaged with the kit.

2. PicStic controller board (if ordered with the development system)

Your package may include either a PicStic-1, -2, or -3 depending on the package you ordered.

3. DB25M-to-RJ11 adapter and 6-wire telephone cable

These two items are used to connect the EPIC programmer to the parallel port of your PC. The adapter is wired at the factory for use with the programmer and no modifications should be necessary.

4. Wall-mount power supply

The programmer requires a 15-18 VAC or DC power supply. If you are using a supply other than the one provided with the development system, it is important that you check its output level!

The PicStic Programmer

The PicStic Programmer is used to program the PicStics from the *.hex files generated by the PicBASIC Compiler, PIC Macro Assembler, and other compilers or assemblers. The PicStic Programmer connects to any standard PC compatible parallel port.

Assembling the Programmer

Included in the programmer package is a parts list and assembly instructions. Before starting to assemble the programmer, check to make sure that all of the parts necessary are included in your kit. The kits are checked at the factory, but this enables you to become familiar with the parts prior to installing them.

Software

The PicStic Development System diskette contains all the software necessary to compile PicBASIC code and assemble PIC assembly language programs. The diskette also includes EPIC software for running the PicStic programmer and some sample programs showing the use of key PicBASIC commands. Please refer to the ADC.BAS and CLOCK.BAS programs for information on using the PicStic's ADC and clock functions.

Software Installation

Create a subdirectory on your hard drive called PBC or EPIC or another name of your choosing by typing

```
md pbc
```

at the DOS prompt. Copy all of the files from the included floppy disk into the newly created subdirectory by typing

```
xcopy a:*. * pbc /s
```

PBC.EXE

PBC is the PicBASIC compiler. It is invoked by typing

```
PBC Options Filename
```

For example, typing `PBC TEST` compiles the program named `TEST.BAS` without any options. Note that the `.BAS` extension was not entered. PBC automatically assumes this extension unless another is entered. In most cases, you don't need to use any of the options. For more information on options, please refer to chapter 2 of the PicBASIC compiler manual.

PM.EXE

PM is the PIC Macro assembler provided with the development system. This program is used to assemble routines written in assembly language. PBC also uses this program to assemble the compiled output file it generates. This is done automatically when PBC is run. PM is invoked by typing

```
PM Options Filename
```

For more info on using the macro assembler, refer to the PIC Macro Assembler manual.

EPIC.EXE

EPIC is the user interface to the PicStic programmer. This program allows the user to store compiled and assembled programs in the PicStic. EPIC is invoked by typing

```
EPIC
```

When the program begins running, it searches the parallel ports of the PC for the PicStic programmer. If the programmer is not connected, or is not powered up, an error message is displayed. Once the programmer is found, a user screen is displayed. Commands may be invoked by using a mouse or the keyboard. If using a mouse, simply click on the command you wish to issue or the parameter you wish to set. To move from field to field using the keyboard, press the Tab key. Each time the key is pressed, the next field is highlighted. Quick keys are also implemented by pressing the Alt key and the highlighted letter of a command at the same time. The descriptions below explain the meaning of each command and device parameter.

<u>Command</u>	<u>Quick Key</u>	<u>Description</u>
Open File	Alt-O	Loads a compiled/assembled file into a buffer in the PC's memory.
Program	Alt-P	Takes the file in the buffer and stores it in the PicStic
Verify	Alt-V	Compares the file in the buffer to what is stored in the PicStic. Note that verify does not work if a code protected PicStic is read and then the verify command is then issued. Only the original program can verify a code protected PicStic.
About	Alt-A	Displays the version number of the EPIC program being run.
Read PIC	Alt-R	Reads the program stored in the PicStic and stores it in the buffer.
Blank?	Alt-B	Reads the PicStic to see if it is blank.
Erase	Alt-E	Erases the PicStic's memory.
Exit	Alt-X	Exits the EPIC program.

<u>Parameter</u>	<u>Options</u>	<u>Description</u>
Device	61/71 84 6x/7x	Type of processor. Select 84 for the PicStic
ROM Size	.5K 1K 2K 4K	This display changes depending upon the type of processor selected. When Device is set to 84 only 1K displays. This is correct for use with the PicStic.
OSC	LP XT HS RC	Oscillator type. The PicStic uses a ceramic resonator so XT should be selected.
Watchdog Timer	ON OFF	The watchdog timer is used by some of the PicBASIC commands such as Sleep and Nap. The timer should be set to ON when using the PicStic.
Power-up Timer	ON OFF	When the PIC processor uses a crystal or ceramic resonator, a delay is needed after power-up to allow the clock to stabilize. Turning the power-up timer on holds the processor in reset for 18 ms after a valid high is seen at the *MCLR pin. The timer should be set to ON when using the PicStic.
Skip blank check before programming	<input type="checkbox"/>	Normally when the Program command is issued EPIC tests PicStic to make sure it is blank. Selecting this option allows this test to be skipped.

Connecting the Programmer to your PC

The PicStic Programmer connects to your PC's parallel printer port. A cable and DB25M-to-RJ11 adapter have been included for this purpose. *Please note that the only time a PicStic may be safely installed in or removed from the programming socket is when the programmer is connected to the PC, the programmer is powered up, the EPIC control program is running, and the LEDs on the programmer is off. If all four of these conditions are not met, you can assume it is not safe to install or remove the PicStic.* Also be sure that the PicStic Programmer is placed on an insulated surface to prevent the shorting out of traces on the bottom of it.

Follow these steps for connecting the programmer to your PC:

- 1) Make sure that the programming socket strip (J2) of the PicStic programmer is empty and that the power supply is not connected to the programmer.

- 2) Connect the programmer to the PC's parallel port using the RJ11 to DB25M adapter cable. Make sure to connect the programmer to a parallel printer port. Connection to a serial or SCSI port, which have similar connectors, may result in damage to the port or to the programmer.
- 3) Plug the power supply into the power connector on the programmer and then into a wall outlet. The power supply should output between 15 and 18 VAC or DC at about 250 ma.

Note: The LEDs on the programmer may be lit at this point. It is no cause for alarm. The LEDs should go out when the EPIC programming software is run. Do not insert or remove a PicStic when the light is on.

- 4) Start the EPIC control software on the PC. The LEDs should now be off. At this point, it is safe to install a PicStic in the programming socket.

Quick Start Example: Writing, Compiling, and Programming a PicBASIC Program

To better understand the process of writing, compiling, and programming a program into a PicStic, let's use a short example.

To write your program, you need a text editor. The editor must be capable of storing the program as an ASCII text file. This means that the editor should not add any special formatting characters to the file. Most text editors and word processors allow you to do this by using the "Save As" command. When saving the file, the editor asks you what "type" of file it should be saved as. Select "Text," "Text(DOS)," or "ASCII." You should also give your file an extension of ".BAS" In this example we'll call our program "TEST.BAS"

Using your text editor or word processor type in the following program:

```
'Filename: TEST.BAS
'This is a sample program to show how a program is compiled and programmed into
'a PicStic. It flashes two LEDs on and off.
'
Symbol LED0 = 0      'Call pin PB0 LED1
Symbol LED1 = 1      'Call pin PB1 LED2

Bgn:   Low LED0      'Make the pins outputs, turn off LED0 and turn on LED1
       High LED1

Loop:  Pause 500     'Pause for 0.5 seconds

       Toggle LED0   'Change the state of the LEDs. If low make it high, if high
       Toggle LED1   'make it low

       Goto Loop     'Repeat the loop continuously
```

Once you have the program typed in, save it as a text file named TEST.BAS. After the program has been saved it needs to be compiled. What we are actually doing is compiling the program to a source file (TEST.SRC) containing assembly code and then assembling it to a machine code file (TEST.HEX). Type the following to begin the process:

```
PBC TEST
```

Press enter and the following should appear on the screen:

```
PicBASIC Compiler Version x.xx      Copyright xxxx, microEngineering Labs  
PIC Macro Assembler Version x.xx   Copyright xxxx, microEngineering Labs
```

where x.xx are version numbers and xxxx is a year

If this is all that appears then the process has completed without any errors. If there are errors, a message displays which identifies the error and its line number. If any errors occur when you compile TEST.BAS, go back and make any necessary corrections.

The compiler and assembler create a file called TEST.HEX. This is the file which the EPIC programmer stores in the PicStic. By this point, you should have assembled and tested the programmer board. Make sure it is connected to the parallel port on your PC and that the power supply is plugged in. Type the following and press enter to start the programmer software:

```
EPIC
```

The EPIC screen should now appear and the LED on the programmer should be off. Load the program by selecting the Open File command and selecting the TEST.HEX file. Make sure that the programming parameters are set to:

Device	84
ROM Size	1K
OSC	XT
Watchdog Timer	OFF ON
Power-up Timer	ON or OFF
Code Protect	ON or OFF (It doesn't matter for this test.)
Skip blank check	OFF (not selected)

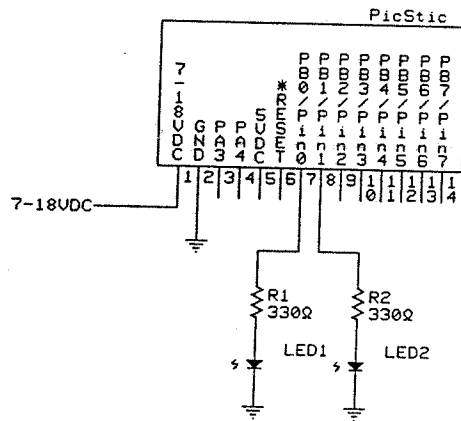
Now insert the PicStic into the programming socket strip (J2) on the programmer. Make sure that pin 1 of the PicStic (on the same end as C1) is installed in the programming socket strip where the silkscreen is labeled Pin1 (PicStic parts toward the center of the PicStic programmer.) Next select Erase to clear the PicStic's memory.

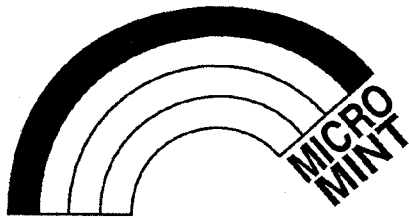
Once this is done, select Program and the programmer begins testing the PicStic's memory to make sure it is blank. When the blank test is completed, the program begins to store the file into the PicStic. As it is storing the file, the programmer verifies that data has been stored correctly in the PicStic. If a memory location is not programmed correctly, an error message appears. If this happens, erase the PicStic and try again.

This problem can occur when EPIC is running under Windows, even though the PicStic is being programmed correctly. It is best to exit Windows before running the EPIC program so that this error does not happen unless a real programming error occurs.

Once programming is completed, you can remove the PicStic from the programming socket. At this point, the PicStic is programmed and will begin running the TEST.BAS program when it is powered up. Connect your PicStic as shown in the schematic below. The two LEDs should start alternating on and off. If the LEDs do not flash, go back and check your connections. If you are sure the circuit is wired correctly, you can verify that the program is stored in the PicStic by using the Verify command in the EPIC program.

If everything is working correctly, then congratulations. You have successfully written, compiled, and stored a PicBASIC program in your PicStic. Feel free to modify the TEST.BAS program to try out other commands and functions.





Micromint, Inc.
4 Park St.
Vernon, CT 06066
Tel: (860)871-6170
Fax: (860)872-2204

PicStic Programmer-2

The PicStic Programmer-2 is designed to work with all current PicStic devices. This programmer allows you to program PicStic-1, -2, -3, -4, PC, and HD devices. Users who currently have the original PicStic programmer and wish to use PicStic-4 or HD devices should order the PicStic Programmer-2 upgrade. The PicStic Programmer-2 features a 32 pin socket (ZIF socket available optionally). This programmer also allows for future expansion should additional versions of the PicStic become available. Please note that you must use a ZIF socket when programming PicStic-HD devices.

Configuring the Programmer for DIP or SIP devices

The PicStic Programmer-2 interfaces to the PC, and uses the same EPIC control program, as the original programmer. The programmer is jumper selectable for SIP or DIP style PicStics. JP2 should be set as follows.

JP2 Setting	Device
DIP PicStic	PicStic-4
SIP PicStic	PicStic-1, -2, -3, -PC, -HD*

* While the PicStic-HD is a "DIP" package it actually contains a SIP device

Inserting devices in the programming socket

Pin 1 of the device you wish to program must be inserted into pin one of the programming socket. This applies to all versions of the PicStic. Pin 1 of the socket is marked on the programmer board as "Pin 1 all devices" with an arrow pointing toward pin 1 of the socket.

A special note about PicStic-3 devices

When a PicStic-HD-3 is inserted into the programming socket the two analog input pins occupy the same ZIF socket locations as pins 13 and 14. This is acceptable and will not cause any damage to the device when it is programmed.