PIC PROGRAMMING PROCEDURE:

1. Open MPLAB IDE from the desktop icon or go to:

Start | Programs | Microchip MPLAB IDE | MPLAB IDE

2. If you have previously created a project, use *Project | Open*; otherwise, go to *Project | Project Wizard*.

PROJECT WIZARD

STEP 1:

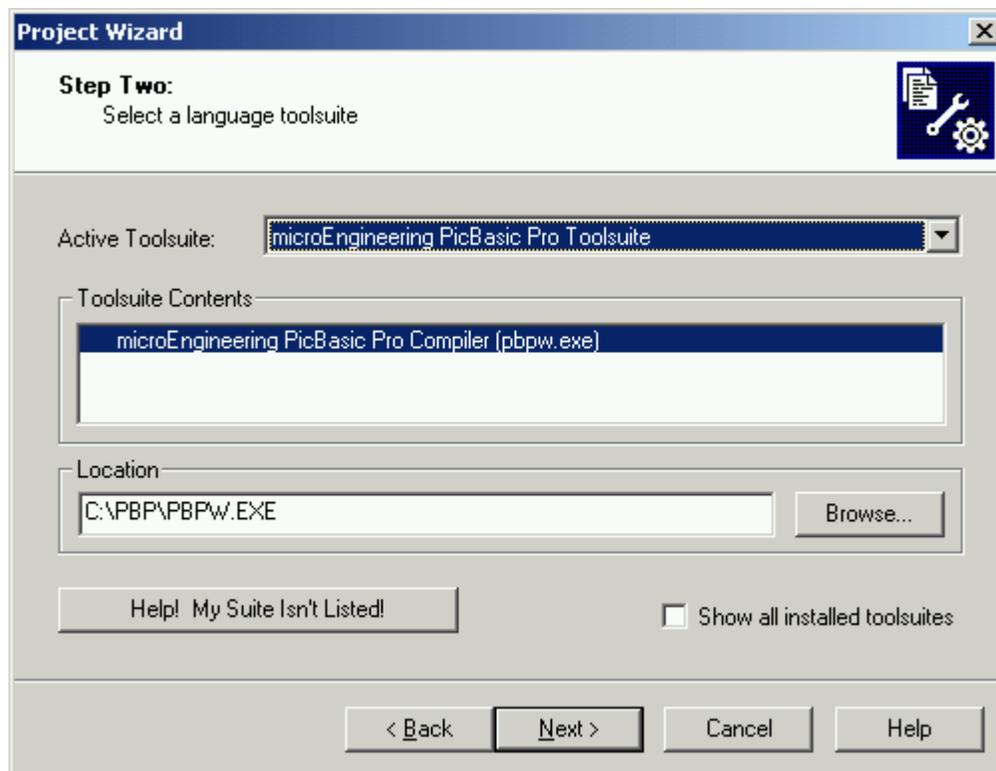Choose the device (PIC) you're using. Probably the 16F88.

STEP 2:

These fields should already be set to….

Active Toolsuite: microEngineering PicBasic Pro

Toolsuite Contents: microEngineering PicBasic Pro Compiler (pbpw.exe)
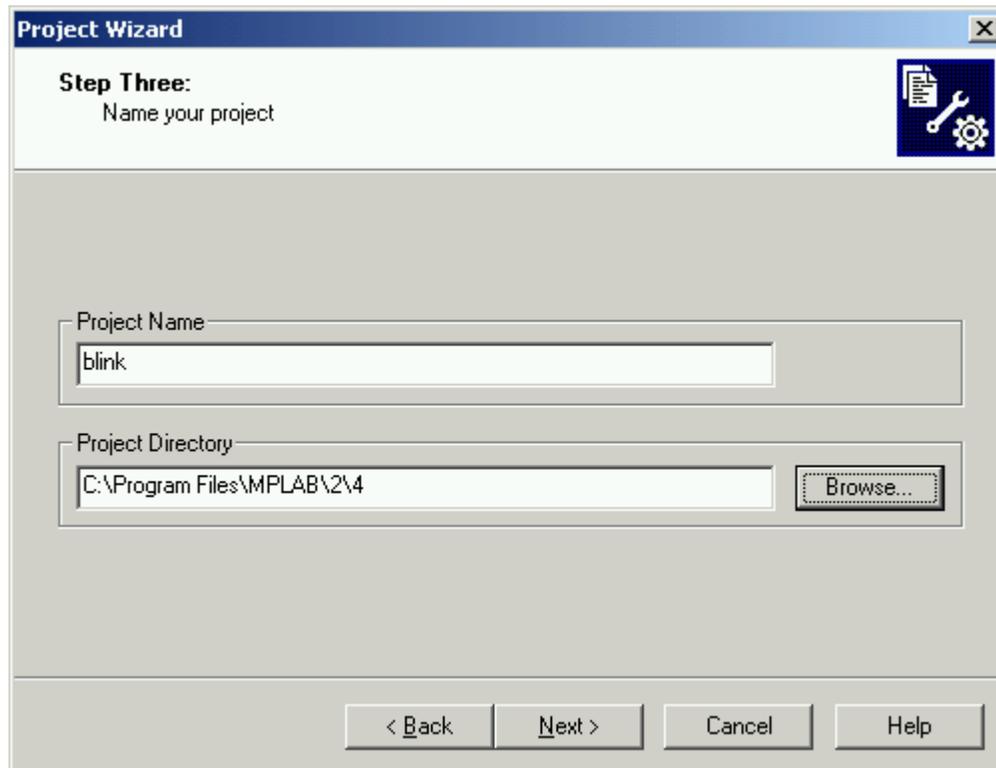
***Location*: C:\PBP\PBPW.EXE**

STEP 3:

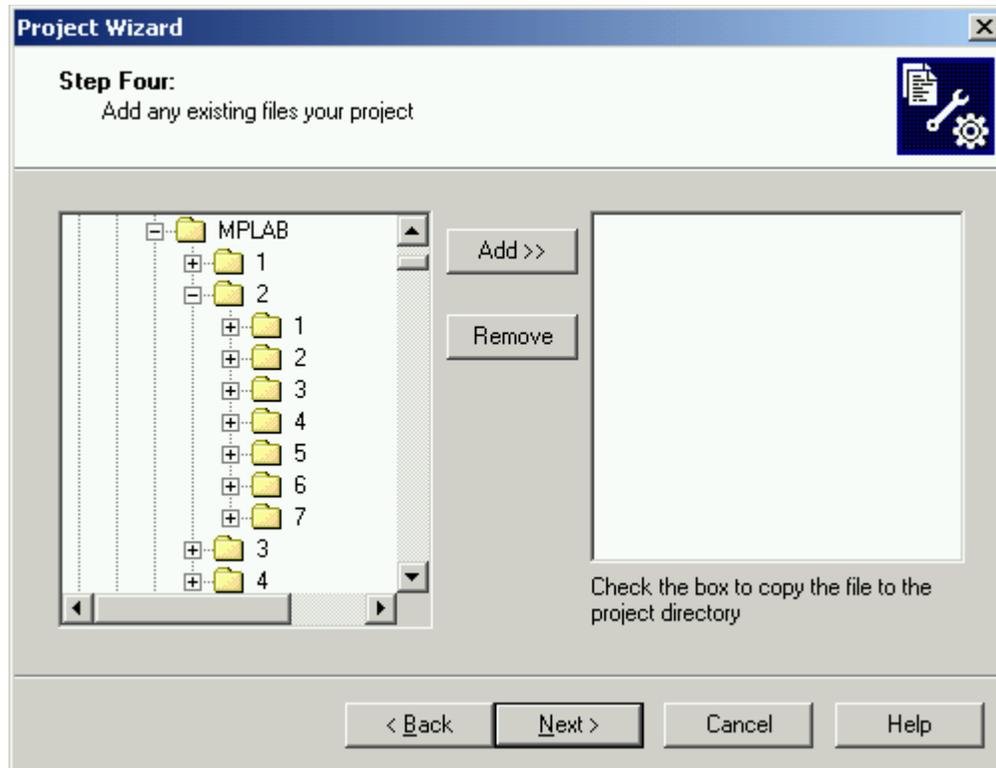*Project Name*: 8 or less characters

Project Directory: Where to save the files that go in your project. You should **save this to your U: drive** (and not the C-drive as shown below).

STEP 4:

If you had already created a ".bas" file (the type of file where your code is written), you could add it here. If you haven't written your code yet, just hit *Next* and *Finish*.

NOTE - An alternative to steps 3-5 below is to use the Microcode Studio editor to create and edit your code. Microcode provides in-line help, interactive syntax checking, and development tools that help you minimize errors.

3. Go to *File | New*

This opens a new window in which to write your code. Before writing any code, however…

4. Go to *File | Save As…*

Browse to the folder in which you created your project (Step 3 of the Project Wizard). Save your file with the same name as the project but add ".bas" to the end. For example, blink.bas. It won't add ".bas" automatically, so remember to add it yourself.

5. Write your code!

> In the empty window, write your PicBasic code. **Note - If you are using the PIC16F88 you will need to include the extra header code provided earlier.**
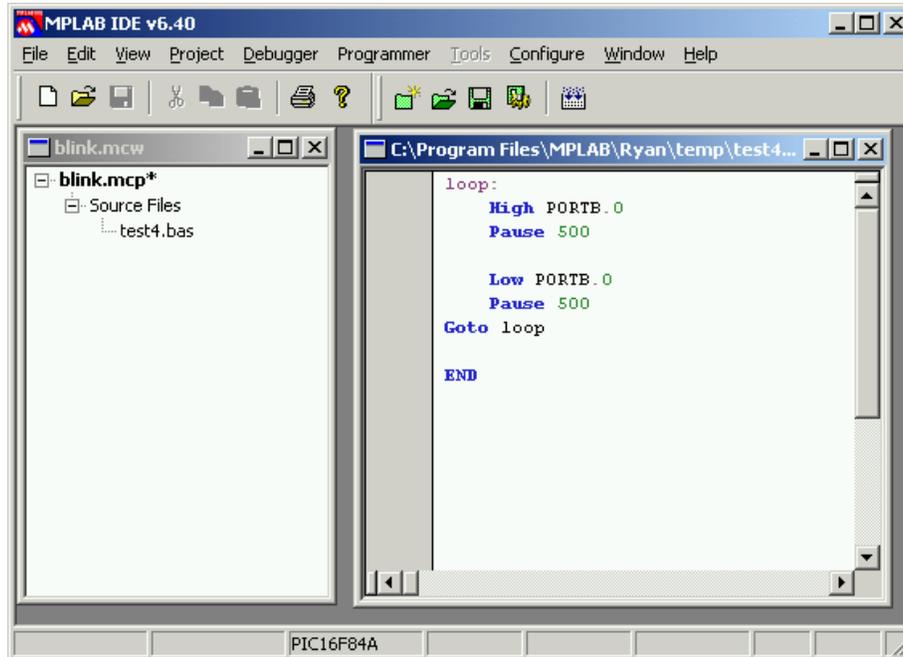
6. Go to *Project | Add Files to Project...*

> Browse to the folder where you just saved your ".bas" file and open that.

7. After writing your code, compile it.

> Either go to *Project | Compile*, hit F10, or click on the [icon] icon.

8. An Output window describing the results of the compilation is displayed. If there were no errors in your code, it will tell you that the BUILD SUCCEEDED. If you had an error, it will say BUILD FAILED and indicate which line number(s) the ERROR was on. If you had errors, go back and fix them (start with the lowest line number).

9. Go to *Configure | Configuration Bits* (if the Configuration Bits window is not visible already). Uncheck the "Set configuration bits in code" box, and **make sure the following settings are selected every time you compile your project**:

- *Oscillator (FOSC)*: INTRC (if using a PIC with an internal oscillator) or XT (if using a PIC with an external oscillator)

- *Watchdog Timer (WDTE)*: Enabled

- *Power Up Timer (PWRTE)*: Enabled

- *Code Protect (CPD)*: Off

If you are using a PIC other than the PIC16F84, make sure other settings are set properly for your application. For example, for the PIC16F88, where several pins have several possible functions, the following setting might be appropriate:

RA5/MCLR pin function select: RA5 (to use this pin as extra I/O instead of MCLR)

NOTE - These settings reset to default values every time you compile, so you must update them after every code edit.

NOTE - Depending on how multi-function pins are being used, bits in certain registers (e.g., OSCON, ANSEL, and ADCON) must be set in your code to have the pins function as desired. For example, with the PIC16F88, to use PORTA pins for digital I/O, the ANSEL bits must be set to 0. See the relevant sections in the PIC datasheet for more information.

10. Go to *Programmer | Select Programmer | PICSTART Plus*

11. Go to *Programmer | Enable Programmer*

12. Put your PIC in the PICSTART Plus programmer. Lift the handle of the ZIF socket, and put your PIC in so that pin 1 (top left) is at the top left of the socket. **Make sure the PIC is in the correct orientation.** Push the handle back down to lock the PIC in place. **Make sure the PIC is well seated when locking down so all pins will have reliable contact with the socket.**

13. Go to *Programmer | Program*

14. Click Yes on the warning window.

15. Wait until the progress bar on the bottom of the screen stops moving and the ACTIVE LED on the PICSTART Plus programmer turns off.

16. Take out your PIC and close MPLAB IDE.

Note - If you want to know how much of the PICs EEPROM memory you have used up, open the ".LST" file generated by the program and look at the bottom of the file, or use the Memory Gage feature under *File | Import | .HEX file*.