Getting started with MPLAB (PIC assembler and simulator) for PIC12F508 assembly language programming

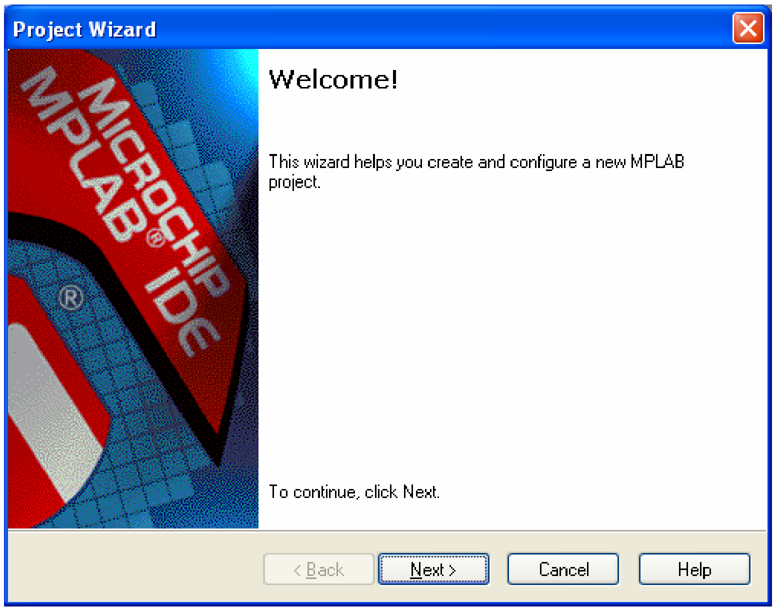
ENGR 4250 Fall 2006 Michael A. Covington

Key idea behind MPLAB: You're not just creating an .ASM file for your program. You're creating a **project** (MCP file), which tells the computer which ASM files are involved (there could be several) and other settings. MPLAB can also work with C and other programming languages.

NOTE: If MPLAB won't start *and you get the message "Access Denied," it's because you don't have permission to write in C:\Program Files\Microchip. MPLAB requires us to violate normal Windows security practice by allowing ordinary users to write in Program Files.*

Creating your program:

Go to **Project** and choose **Project Wizard**.

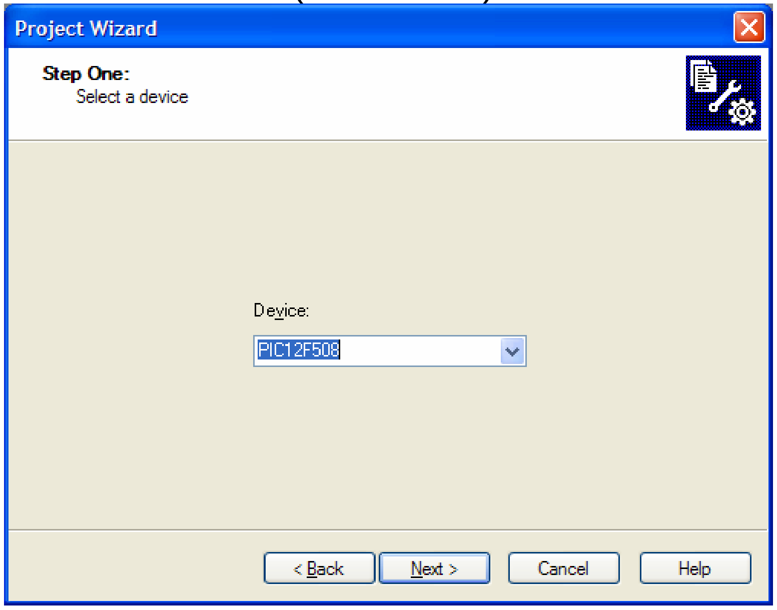


Welcome!

This wizard helps you create and configure a new MPLAB project.

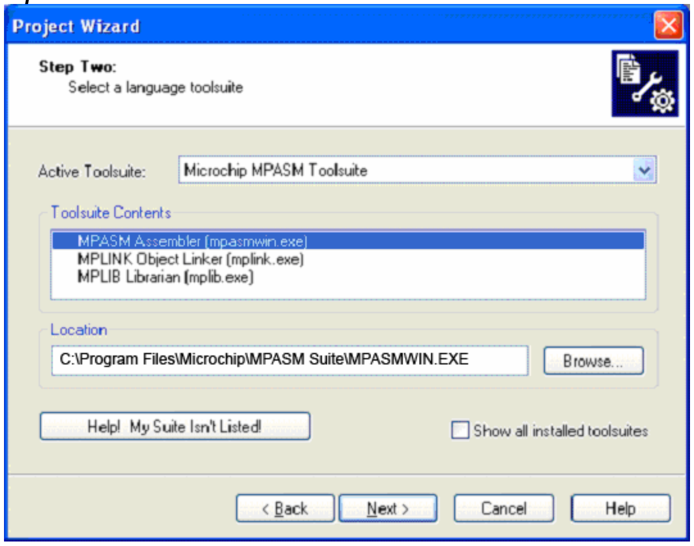
To continue, click Next.

Select the device (PIC12F508).



Device: PIC12F508

Select **Microchip MPASM Toolsuite.**
If there's a red X showing anywhere, MPLAB needs to know where the tools are, namely C:\Program Files\Microchip\MPASM Suite. Don't mix up MPASMWIN with MPASM.



Active Toolsuite: Microchip MPASM Toolsuite

Toolsuite Contents

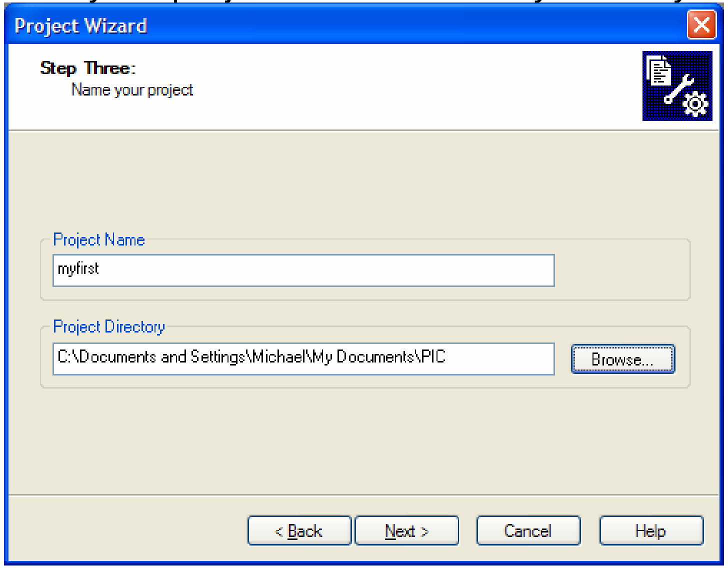
MPASM Assembler (mpasmwin.exe)
MPLINK Object Linker (mplink.exe)
MPLIB Librarian (mplib.exe)

Location

C:\Program Files\Microchip\MPASM Suite\MPASMWIN.EXE

☐ Show all installed toolsuites

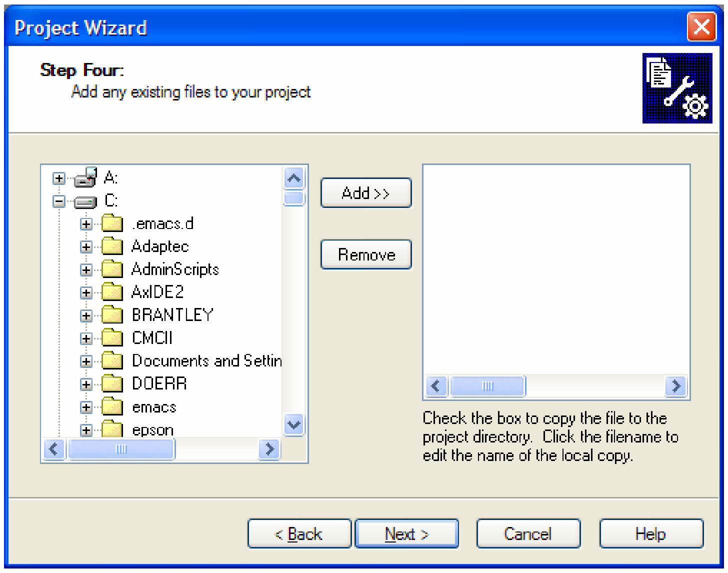
Give your project a name and say where you're going to put it.



Project Name: myfirst

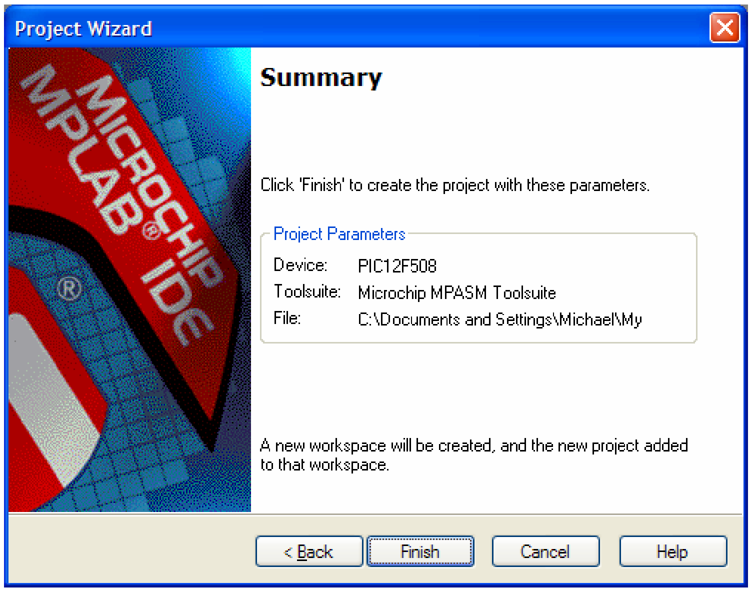
Project Directory: C:\Documents and Settings\Michael\My Documents\PIC

Next, Project Wizard offers to add some files to your project. We're going to skip this window and use a much better user interface to add files at the next step. So just click Next at this screen:



Check the box to copy the file to the project directory. Click the filename to edit the name of the local copy.

Give final approval, and your project will be created:



Summary

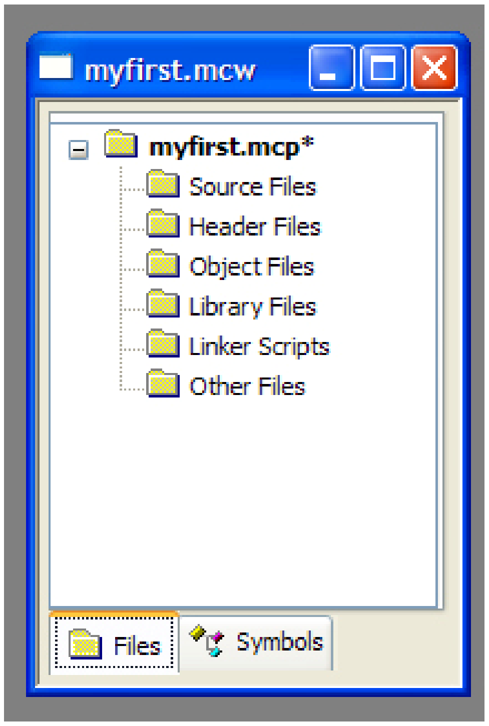
Click 'Finish' to create the project with these parameters.

Project Parameters

Device:	PIC12F508
Toolsuite:	Microchip MPASM Toolsuite
File:	C:\Documents and Settings\Michael\My

A new workspace will be created, and the new project added to that workspace.

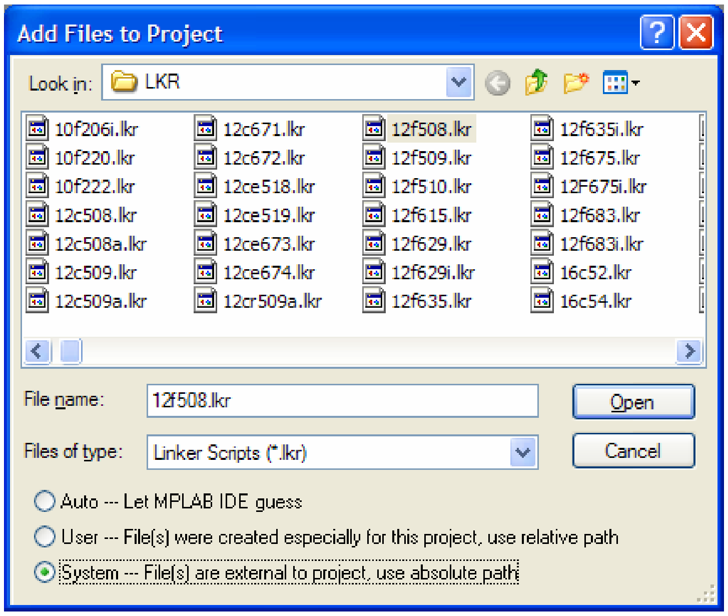
Now you have an empty project with no files in it. You must create or add an .ASM file in order to have something assemble. We'll get to that. Here's what an empty project looks like:



As a first step, adding a **linker script** to your project is strongly recommended. This tells the MPASM suite to use a more modern method of generating your .HEX file which will get around an old 62-

character limit on path lengths. *If you skip this step,* you can still assemble your file provided the full path to it (C:\Documents...\...\etc.) isn't too long.

Let's add the linker script now. Right-click on **Linker Scripts**, navigate to C:\Program Files\Microchip\MPASM Suite\LKR, and pick the file that matches your processor. Check "System" so MPLAB will know you want to use the "canned" linker script without editing it.



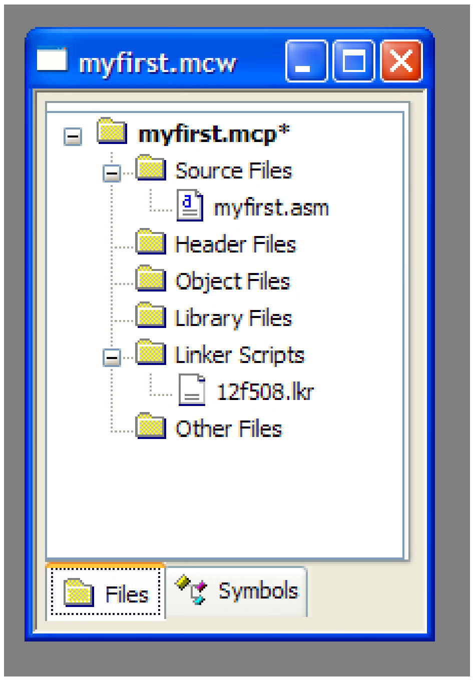
File name: 12f508.lkr

Files of type: Linker Scripts (*.lkr)

( ) Auto --- Let MPLAB IDE guess
( ) User --- File(s) were created especially for this project, use relative path
[✓ System --- File(s) are external to project, use absolute path]

(An alternative is to use Windows to make a copy of 12f508.lkr into your project directory, then add the copy. That may be wiser in the long run.)

Now it's time to create or add an .ASM file. Right-click on **Source Files**, navigate to your .ASM file, and add it. Or choose File, New, type in at least part of the file, save it, and then add it. When you're done, your project will look like this:

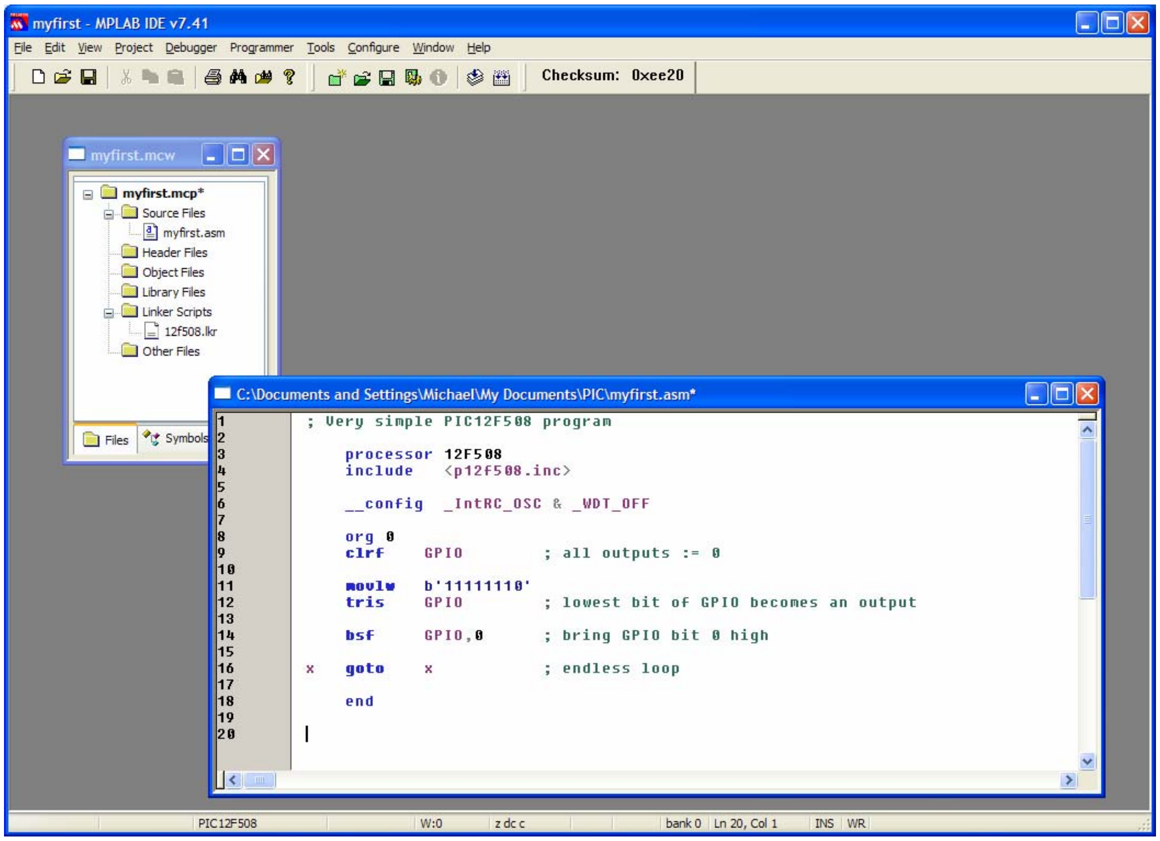


Now you have a working environment with an editor for your file, and the file is shown in the contents of the project. To open your .asm file if it's not already open in the editor, just double-click on it.

Here is a very simple program you can type in:

```
; Very simple PIC12F508 program

      processor 12F508
      include <p12f508.inc>

      __config  _IntRC_OSC & _WDT_OFF

      org   0
      clrf  GPIO         ; all outputs := 0

      movlw b'11111110'
      tris  GPIO         ; lowest bit of GPIO becomes an output

      bsf   GPIO,0       ; bring GPIO bit 0 high

x     goto  x            ; endless loop

      end
```

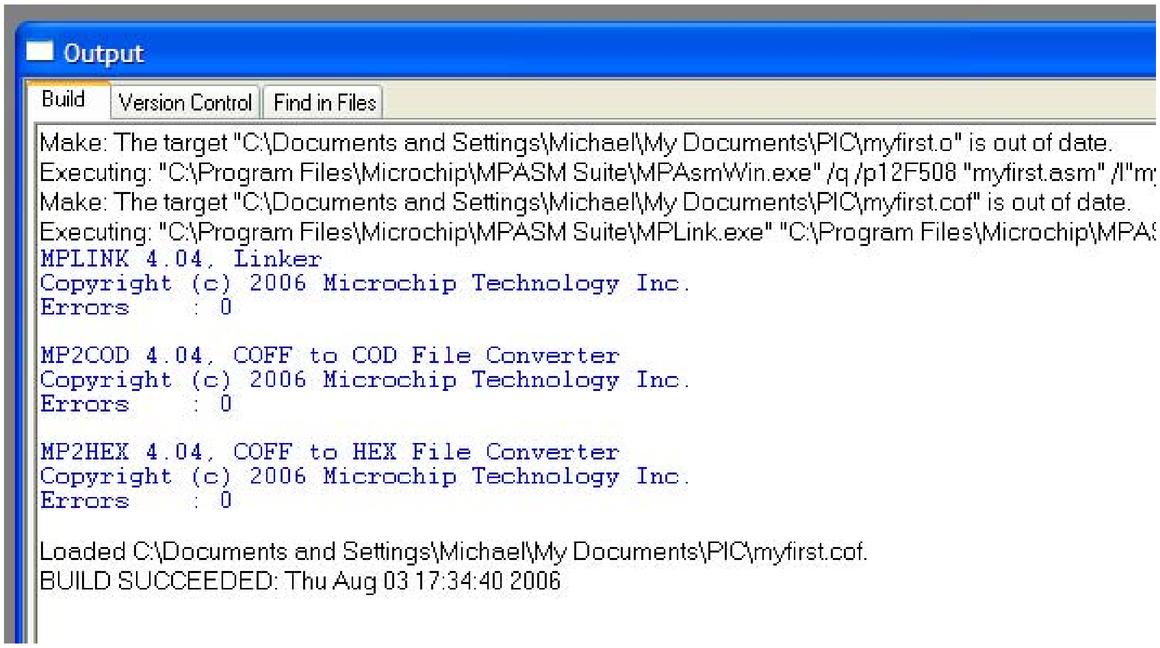
With this open in the editor, here's your project:



```
; Very simple PIC12F508 program

    processor 12F508
    include   <p12f508.inc>

    __config  _IntRC_OSC & _WDT_OFF

    org 0
    clrf    GPIO        ; all outputs := 0

    movlw   b'11111110'
    tris    GPIO        ; lowest bit of GPIO becomes an output

    bsf     GPIO,0      ; bring GPIO bit 0 high

x   goto    x           ; endless loop

    end
```

Save your program.
Assemble it by pressing F10.

When your program assembles successfully, you'll get output like this:



```
Make: The target "C:\Documents and Settings\Michael\My Documents\PIC\myfirst.o" is out of date.
Executing: "C:\Program Files\Microchip\MPASM Suite\MPAsmWin.exe" /q /p12F508 "myfirst.asm" /l"m
Make: The target "C:\Documents and Settings\Michael\My Documents\PIC\myfirst.cof" is out of date.
Executing: "C:\Program Files\Microchip\MPASM Suite\MPLink.exe" "C:\Program Files\Microchip\MPAS
MPLINK 4.04, Linker
Copyright (c) 2006 Microchip Technology Inc.
Errors    : 0

MP2COD 4.04, COFF to COD File Converter
Copyright (c) 2006 Microchip Technology Inc.
Errors    : 0

MP2HEX 4.04, COFF to HEX File Converter
Copyright (c) 2006 Microchip Technology Inc.
Errors    : 0

Loaded C:\Documents and Settings\Michael\My Documents\PIC\myfirst.cof.
BUILD SUCCEEDED: Thu Aug 03 17:34:40 2006
```

Here’s what the messages mean:

“myfirst.o is out of date” – Your program has changed since the last time it was assembled, so it’s going to be assembled now.

“myfirst.cof is out of date” – Same thing. The .o file and then the .cof file and finally the .hex file are the products of assembly and linkage. (Linkage means putting together subroutines that were in separate .ASM files.)

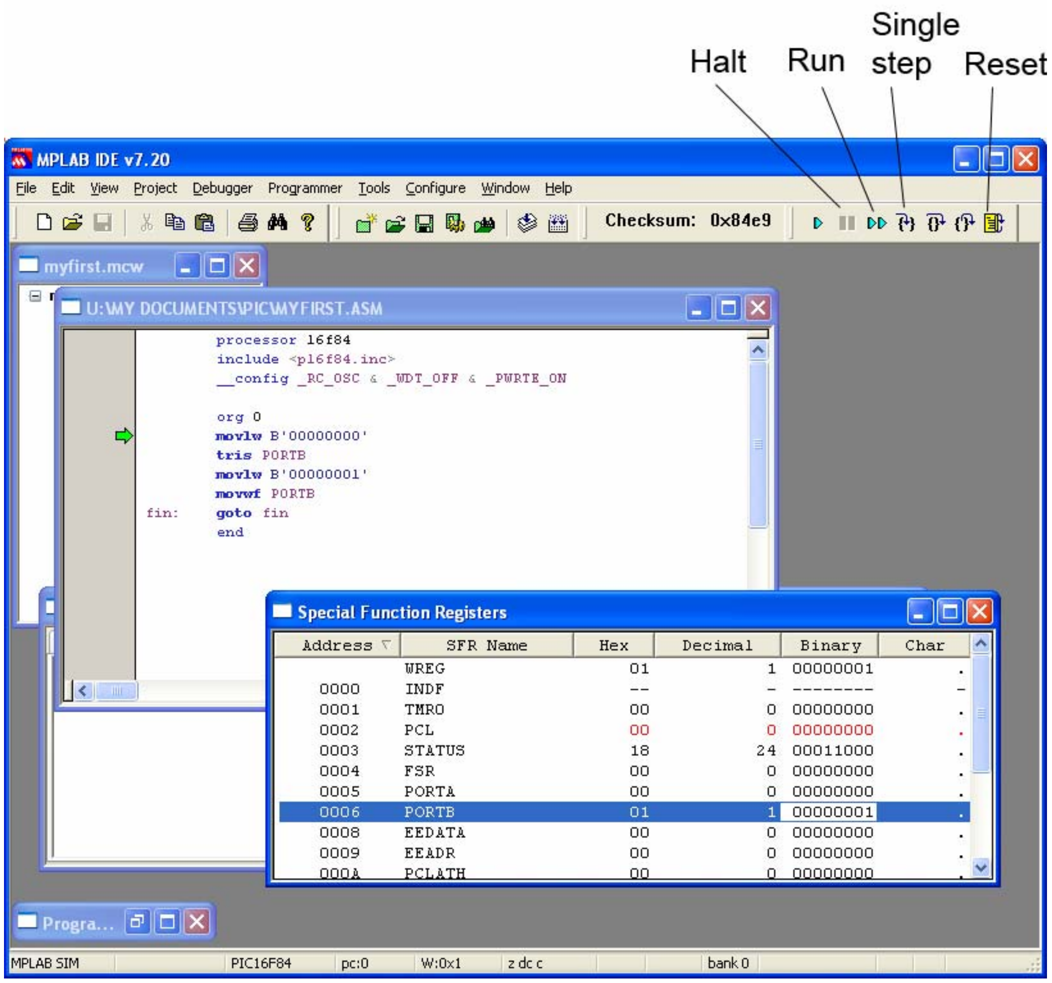
If there is an error message about the COD file, ignore it. COD files can’t contain long paths (“C:\Documents and Settings\...\...\...”) above a certain length. If you are fastidious, you can go to Project, Build Options, MPLINK, and check Suppress COD File Generation.

Simulate your program

Go to Debugger, Select Tool, MPLAB SIM.

Go to View, Special Function Registers, so you’ll be able to see the output port. (You can also view several other things.)

Then run the program using the buttons at the top right:


Halt
Run
Single step
Reset
MPLAB IDE v7.20
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x84e9
myfirst.mcw
U:\MY DOCUMENTS\PIC\MYFIRST.ASM
Special Function Registers
Progra...
MPLAB SIM
PIC16F84
pc:0
W:0x1
z dc c
bank 0

```
        processor 16f84
        include <p16f84.inc>
        __config _RC_OSC & _WDT_OFF & _PWRTE_ON

        org 0
        movlw B'00000000'
        tris PORTB
        movlw B'00000001'
        movwf PORTB
fin:    goto fin
        end
```

Address	SFR Name	Hex	Decimal	Binary	Char
	WREG	01	1	00000001	.
0000	INDF	--	-	--------	-
0001	TMR0	00	0	00000000	.
0002	PCL	00	0	00000000	.
0003	STATUS	18	24	00011000	.
0004	FSR	00	0	00000000	.
0005	PORTA	00	0	00000000	.
0006	PORTB	01	1	00000001	.
0008	EEDATA	00	0	00000000	.
0009	EEADR	00	0	00000000	.
000A	PCLATH	00	0	00000000	.

(This picture is from simulation of a different program on a different CPU than the preceding example.)

Program your PIC

Connect the PICSTART Plus to the serial port and power it up.

Go to Programmer, Select Programmer, PICSTART Plus.
Go to Programmer, Enable Programmer.
(Ignore an error about the PICSTART Plus needing to be updated.)

Go to Configuration and make sure
Select Device and Configuration Bits are correct.
(They should agree with the __config statement in the program and the selections you made earlier, but please double-check.)

Insert the PIC in the PICSTART Plus.
Go to Programmer and program or verify your device.

Insert the PIC in the breadboard and see if it works!

When you finish

Be sure to “Save Workspace” (under File) as well as saving your assembly language program.