

Chapter 4: Coding Environment

In the previous chapter, we studied the instruction sets of PIC 16F877 and some tips and tricks of using the instructions are discussed. Even though we met a few example codes, we are not sure where and we physically type the code and not sure if the codes are correct. And we do not know how we if one code is correct or wrong. In this chapter, we introduce a coding environment which allows us to edit an Assembly code, assemble (or compile) the code, simulate the code, and finally download the compiled code to the internal program memory of the PIC chip. The first three involves Microchip's MPLAB (Microchip Assembler) and MPSIM (Microchip Simulator) Integrated Development Environment (IDE), and the last one involves a physical programmer or a burner. Note that the MPLAB explained and demonstrated thoroughly here is MPLAB Version 5.20. Currently, higher version of the IDE is available with version 6.40, therefore, here I add additional section (see section 3) to briefly explain how quickly assemble an Assembly language code under MPLAB v.6.40. The version 6.40 would be more reliable in Windows XP operating condition. I am not 100 percent positive on this but I had experienced some problem of using MPLAB 5.20, and I heard of the problem too, when you try to install it on a tablet PC with Windows XP (or Tablet version of Windows XP). Anyway, one thing you may have to be careful with the version 6.40 is that it is case-sensitive, and nullifying that is not apparently easy. To be in safe side, under MPLAB v.6.40, I suggest you write your code as if the case-sensitive rule is universal in PIC programming.

The last subject of the chapter covers a bootloader code which allows the PIC chip to be ready to communicate via serial communication to your PC, and a PIC code downloader which is a Windows based software that actually downloads your hex code to the PIC chip. One problem with this is that you first have to have the bootloader inside your PIC chip. Without this there is no serial communication for the PIC code download software to download your code. An actual PIC programmer, i.e., a burner in a sense, is needed to burn the bootloader to the chip. The PIC burner is also covered in the chapter.

1. MPLAB v5.2

MPLAB is a Windows-based Integrated Development Environment (IDE) for the Microchip Technology's PICmicro microcontroller families. MPLAB allows you to write, debug, and optimize PICmicro applications for embedded computing product designs. MPLAB includes a text editor, simulator, and project manager. You use the MPLAB Editor to create and edit text files such as source files, code, and linker script files. The MPLAB-SIM Simulator models the instruction execution and I/O of the PICmicro Microcontrollers (MCUs). The project manager is out of scope of this book since our approach is not to make a project file from a code but to quickly build a code to a hex code. The MPASM Assembler allows source code to be assembled without leaving MPLAB.

The following minimum configuration is required to run MPLAB:

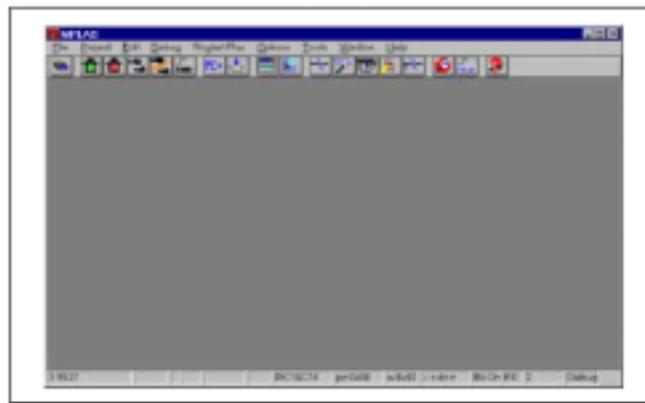
- PC compatible 486 or better class system (Pentium recommended)
- Microsoft Windows Operating System
- VGA display (Super VGA recommended)
- 8 MB memory (32 MB recommended)
- 20 MB of hard disk space

- Mouse or other pointing device

MPLAB may be obtained by contacting any Microchip sales office and requesting the Technical Library CD-ROM or by downloading the files from the Microchip website (www.microchip.com). Version 5.20 of MPLAB can be installed by the file of `Mp520ful.exe`. The executable file `Mp520ful.exe` installs the Microchip MPLAB Integrated Development Environment version 5.20 in your computer. Once you downloaded the file, execute the install program. When the .exe file is run, it will start installing MPLAB onto your system.

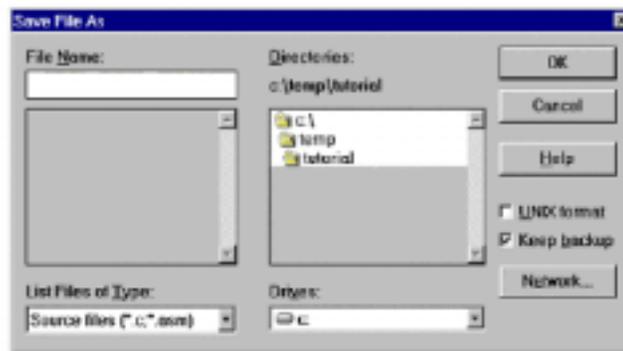
Starting MPLAB

Execute **MPLAB.EXE** or click on the MPLAB icon to start up the system. You will see MPLAB's desktop as shown below.



Creating a New Source File

Click once in the blank space of the empty file window that has been created for you. It is probably titled "Untitled." Or you open a new file. Use the **File>Save As...** menu option and save the empty file as `ex01.asm`. When the standard browse dialog opens, you will find that it is located in the current working directory of the project. If you want to change it to your USB stick, select proper place for your file. Then, enter the file name and press **OK**.



You will now be presented with the MPLAB desktop and the empty file window, but the name of the file window will reflect its new name. MPASM will always create an output hex file with the same name as the source file, and this cannot be changed.