Introduction to the IDE

1 Introduction

The IDE that we are using is called SourceBoost, and is available on all of the machines in the ELC. There is also a free version which has some limitations that you can download and run on your personal computer.

The steps in programming a microcontroller using the IDE are as follows:

1.1 Start the Software

Starting the software is done like any standard windows software.

1.2 Create or Open a Project

SourceBoost uses a “container” called a project to keep all of the details of your program together. These details include the file(s) that the program uses, specifics about the target processor, and other information.

When creating a new project, I would avoid using the wizard until you are more familiar with the IDE and C programming.

1.3 Add files to the project

All files and libraries have to be added to the project. Do not add header files to the project. They are included via the #include statements in your code.

SourceBoost includes an editor that can be used to edit and create programs, but you can use any code editor of you like.

1.4 Compile your project

With all of the files added to the project, it is time to use the ℭ compiler to compile your project. The bottom half of the screen will show the results of this process.

1.5 Link your project

Once the project has been successfully compiled, the next step is to link the project using 𝔏. This program, run from inside the IDE, will take the output of the compiler and translate it into a HEX file.

1.6 Program your microcontroller chip

The next step is to program the hex file that was generated by compiler and linker into the microcontroller. The name of the hex file will be the name of your project with the file extension .hex. When the programmer software comes up, you may have to load the

appropriate hex file before downloading. This is a separate piece of software, although it is launched from the IDE. This program sends the hex file to the hardware programmer mentioned above, which writes the program into the memory on the microcontroller.

1.7 See if the program worked

Once the hex file is downloaded, press the reset button on the development board. The program will start from the beginning and run to the end. Note that the program usually will start running without a reset, but that may not always be the case.

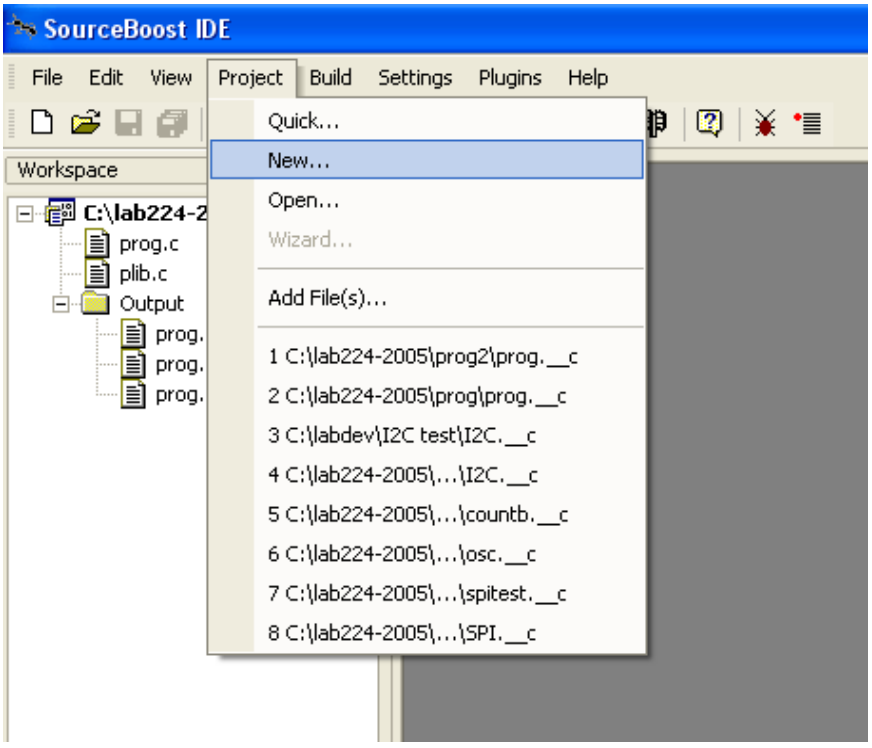
2 Detailed Instructions

The following is a cookbook approach to the steps necessary to run the IDE.

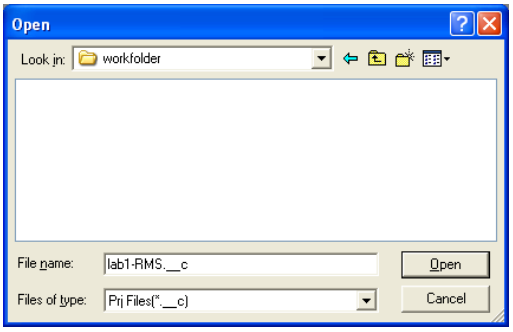
1. Determine where you are going to keep your code, libraries, etc. This needs to be a directory that you have permissions on, and can be in your AFS space, on a memory stick, etc. This document is an edited version of one used in EE 20222 and it assumes there is a directory called workfolder extant for this purpose.

2. Run the SourceBoost IDE. The program can be found in Start → All Programs → SourceBoost → SourceBoost IDE.

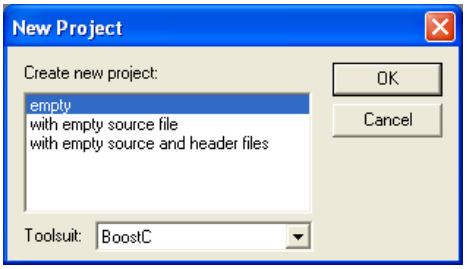
3. Create a new project in your working directory. To do this, go to the Project menu of the IDE, and select new.



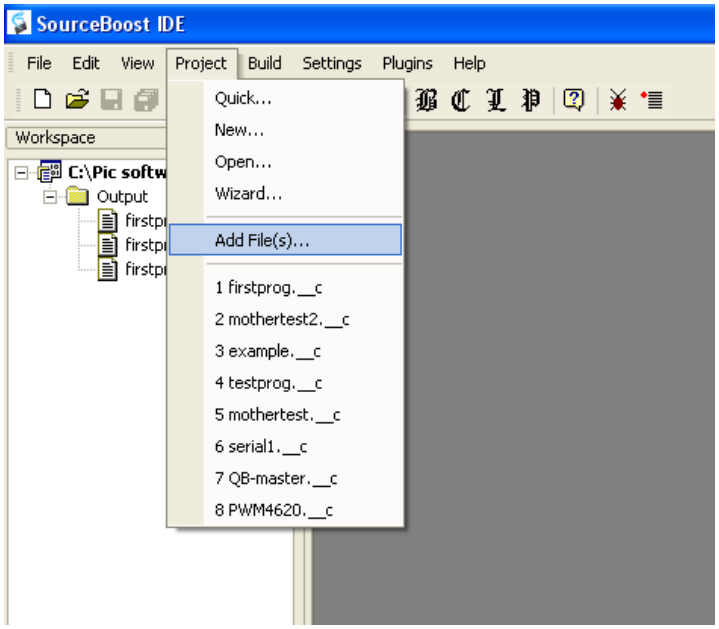
This will open up the dialog box shown below. Change the directory to point to some folder and choose a name for your project. (I chose in this example to call mine lab2rms.)



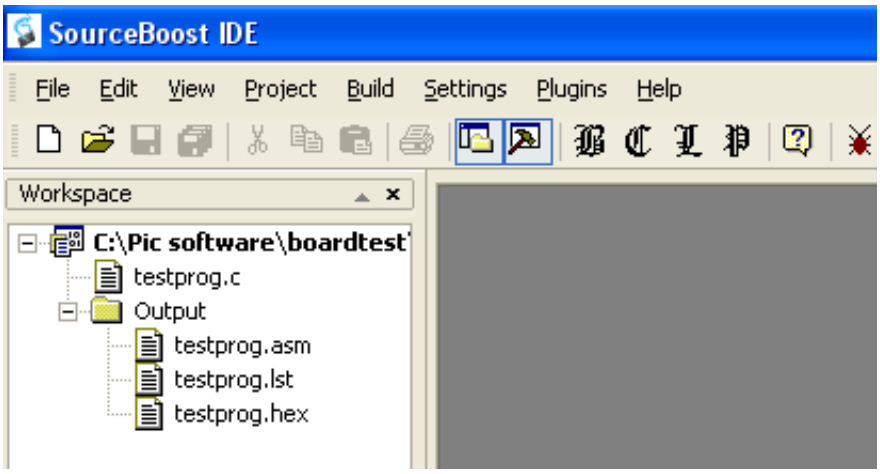
A new project dialog box will appear. You should choose to create an empty project.



4. Add your file to your project.

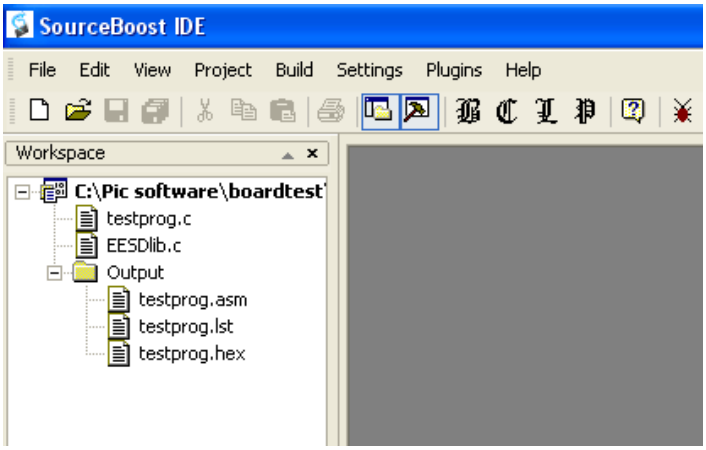


After adding the file, it should show up in the left column.

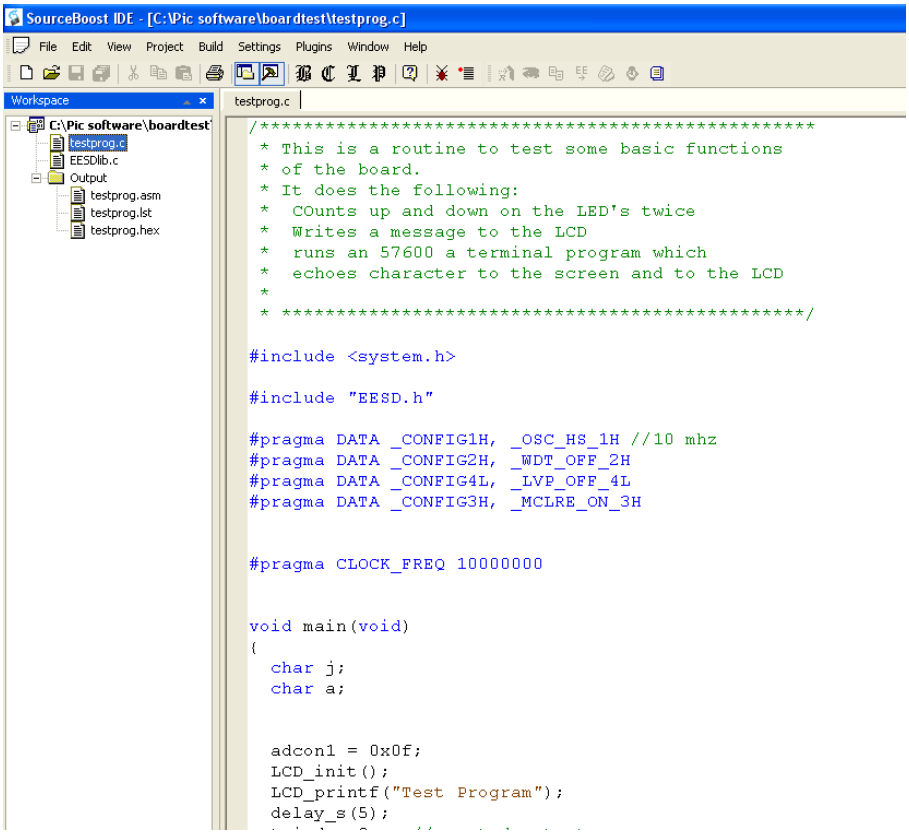


Notice the workspace display. The project name is listed in bold on the first line. Below it you can see the file that I added to the project.

5. Add the EESDlib.c to the project.

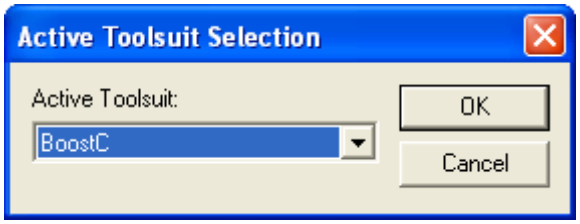


6. Double click on the name of your C file, and it will show up on the right hand part of the screen, where it can be viewed and edited. Note that the editor uses colors to indicate various different types of information in a C program.

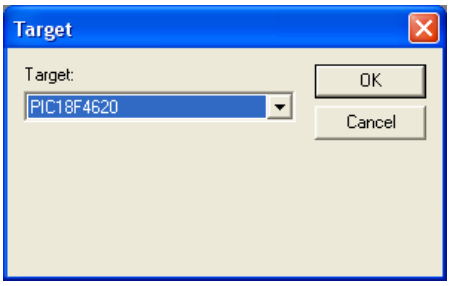


7. Note that there are a number of settings under options that should be correctly pre-set for you but with multiple students using each setup, there are no guarantees. . The correct setting are as follows:

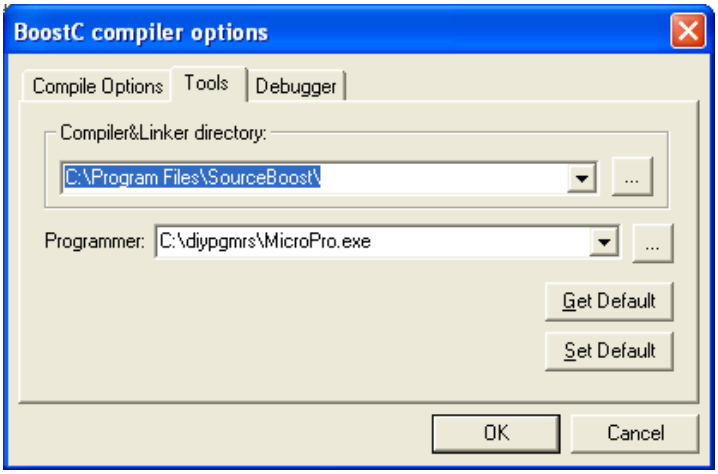
Settings -> Toolsuit



Settings -> Target



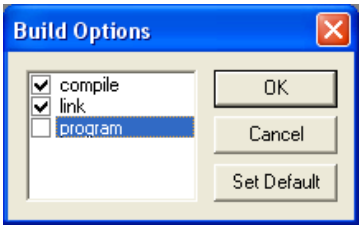
Settings -> Options -> tools



Note that the specific location of the programmer software may be different than what is shown because of the installation setup in the IDE.

Settings -> Build

Set the options you want to happen when you click on build 𝔅. If you include program, it will automatically launch the programmer software.

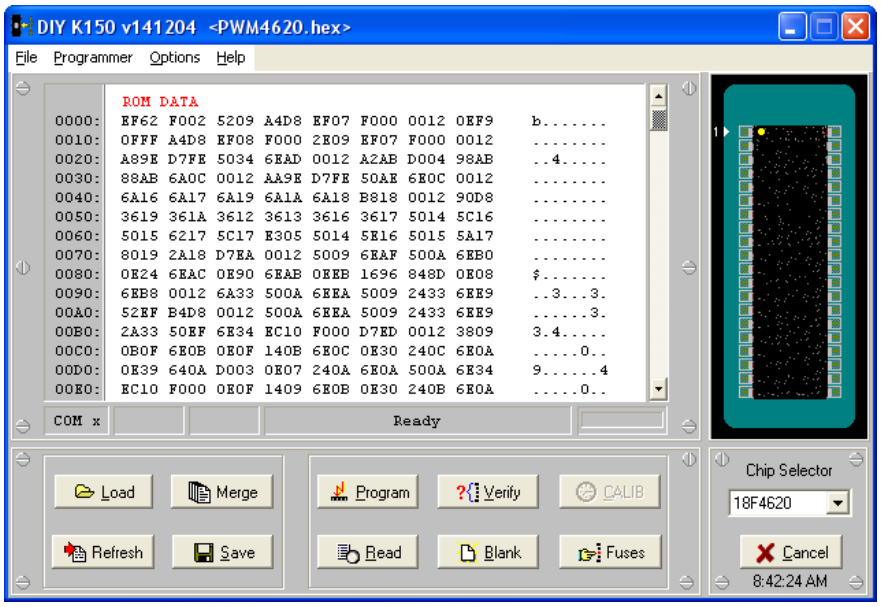


8. Compile your project. Simply click on the gothic ℭ in the top toolbar. In the bottom window, you will see the output from the compile, and if you entered the program correctly you should see no errors. (Note that there will be a number of warnings! These can be safely ignored.) If there are errors, you can double click on the error and it should take you to the offending line. (Note that it is not always possible for the compiler to figure out exactly where the error is, but it should be close to the line given.)

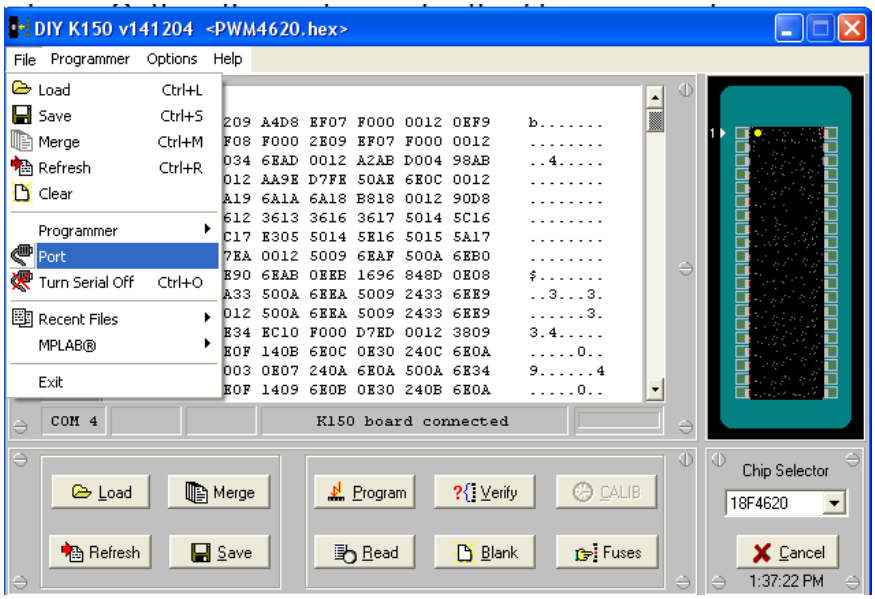
9. Link your project. Click on the gothic 𝔏, and the linker will link the project. (Note that if you had the build options as shown above, clicking on the gothic 𝔅 would have done both these steps.)

10. Compiling and linking produces three output files. An assembly file (.asm), a list file (.lst) and a file that can be downloaded to the microcontroller (.hex).

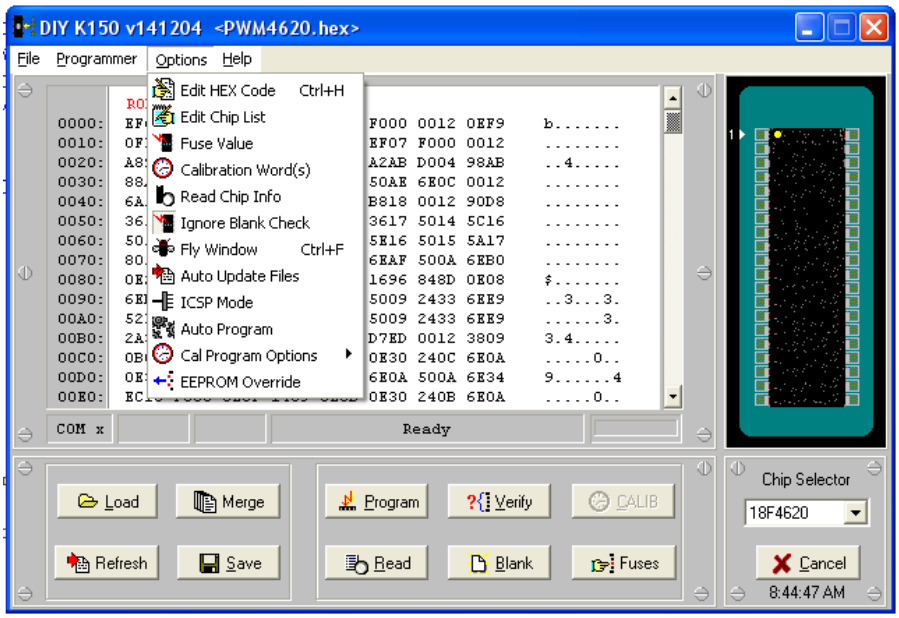
Download the hex file to the microcontroller board. Click on the gothic ℜ and the programmer will start and will appear as below. Note the window that says target. This has to be set as shown. Once it is set, it should remain so.



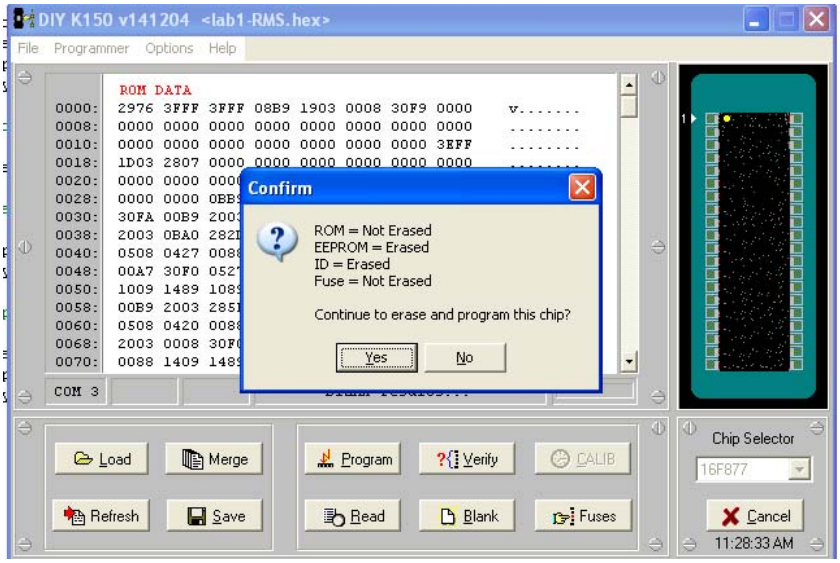
Depending on the USB port that you connect the programmer to, the port setting in the programmer may change. You may have to do some searching through port numbers. Setting the port is under the file command.



There are a number of options for the programmer. I usually have my set as shown below which tells the device to ignore the “blank check” (don’t bother to check if there is a program already in the microcontroller.

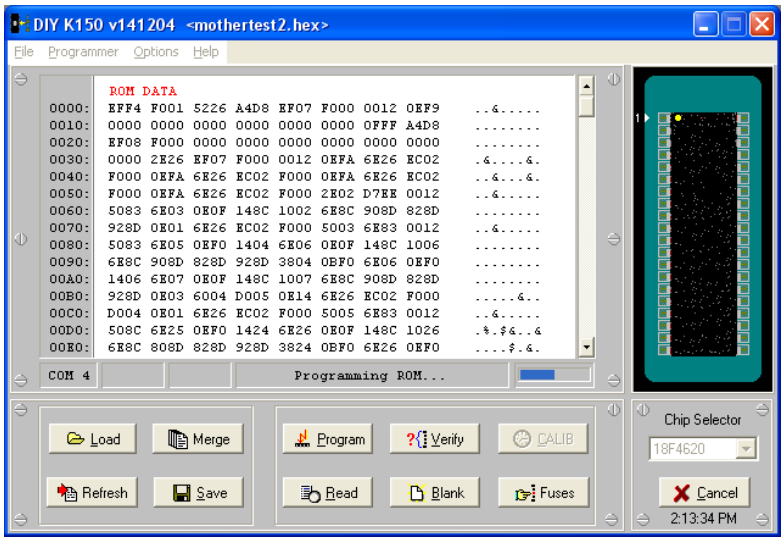


If you haven't told the programmer to ignore the blank check, is will ask you if you really want to program over what is in the chip.

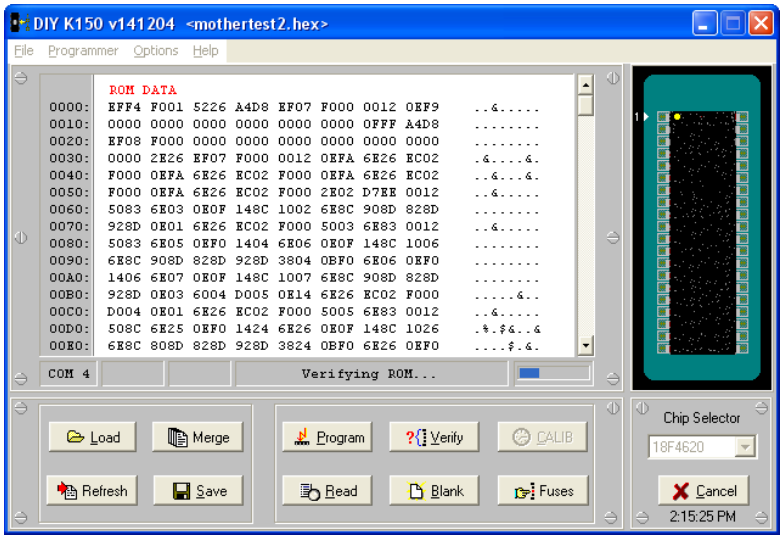


The programmer will download the hex file to the microcontroller board and it will usually begin executing.

The programmer will show progress of programming



and then will show that it is verifying as shown below.



Verifying takes a long time, but you can cancel once it gets to this point, rather than wait for verification to complete. I also make a habit of closing the programmer at this point.

11. If the program doesn't start running, press the reset button on the microcontroller board.