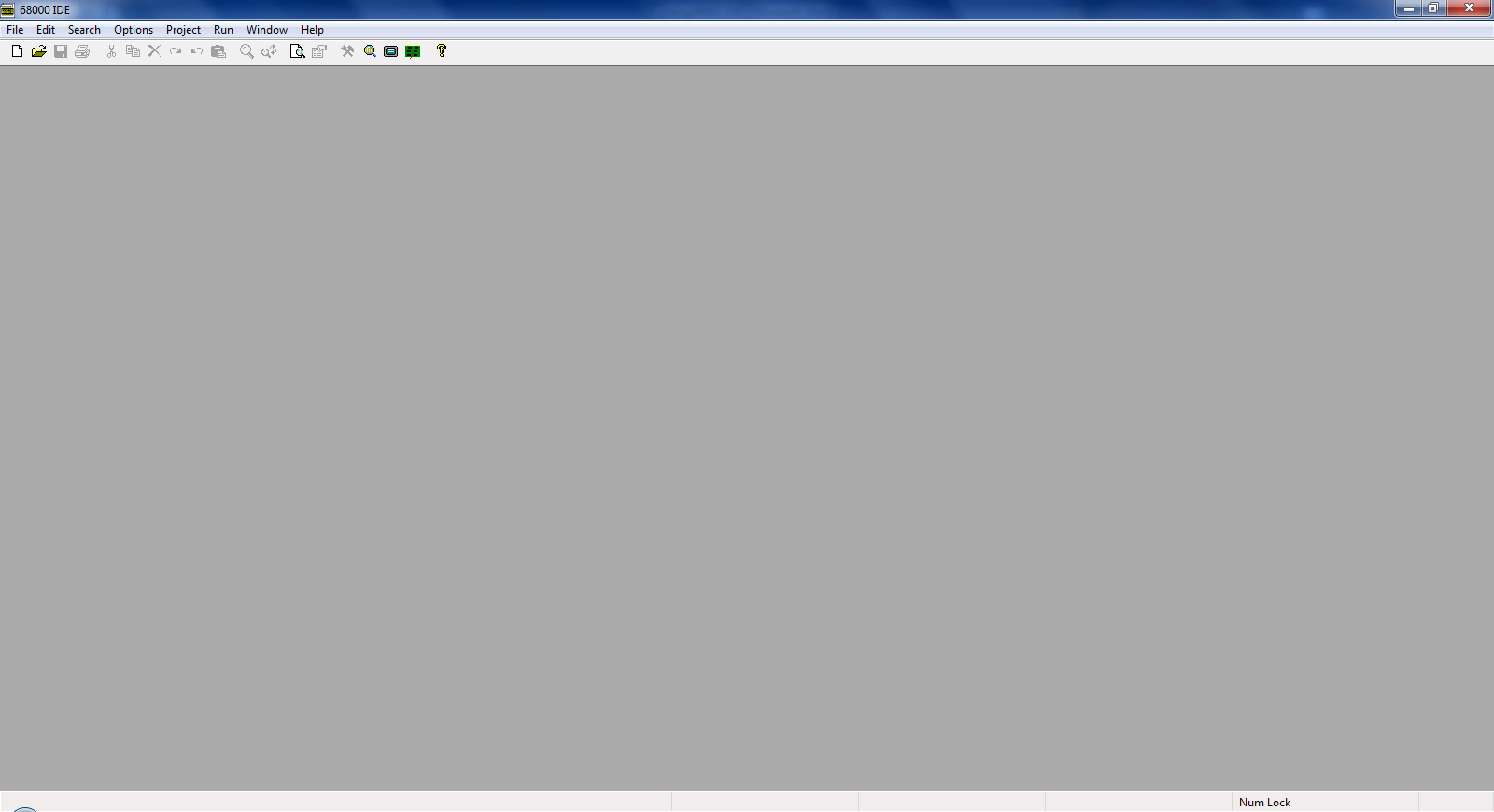
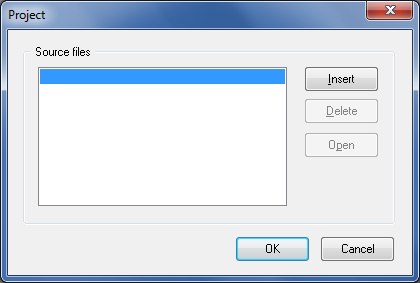
**Using IDE68K to write ‘C’ programs for the 68000   
Soft Core Processor on your Altera Board**

* Download and Install the IDE68k software onto your PC into the folder C:\IDE68k
* Start IDE68K – the following screen appears



* IDE68K is very easy to use. Use the Project Menu to Open an existing project or create a new one.
* Chose create a new project, the following appears

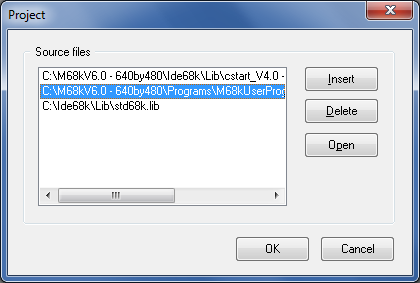


* Click insert button to add the files you want to the project.

**What files do I need to ADD to the Project?**

There are two important files that you must add to your project to allow you to build it

* A **CStart.asm** file supplied by your instructor. Download and copy this file from the course web site to the **C:\IDE68k\LIB** folder or, if you are on course CPEN 412, put in the appropriate folder where your M68k system has been downloaded e.g. the folder **C:\M68kV6.0 – 640by480\IDE68K\LIB** or **C:\M68kV6.0 – 800by480\IDE68K\LIB** (depending upon which version you have downloaded), although the exact location is not important as we can locate the file using the IDE68k compiler.  
    
  It probably has a name like “**CStart\_V4.0\_User Program.asm**”. This a 68000 assembler program that is run to “bring to life” a C program before it gets to main().   
    
  It does things like initialise the stack pointer, enable interrupts, initialise the user programs global variables etc. This kind of program always exists; it’s just that you never see it when writing programs in say Visual C++ on a windows environment.
* Add the library file “**C:\IDE68\LIB\std68k**” to the project that contains the C Standard library. This file is part of the software that comes with IDE68K so it should be found in the LIB folder where you installed IDE.
* Lastly add/create your own files to the list e.g. the C program file **M68kUserProgram.c**   
    
  Note you will be given the above file on the web site, but you can add/include other C files to the project also.
* You should see something like this, although the path names might be slightly different.

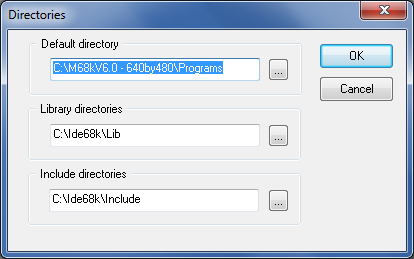


**IMPORTANT**: the “**cstart\_V4.0 xxx.asm**” file **MUST** be at the **top** of your list. If not, you will get **compile errors**. The IDE68k environment occasionally loses this fact and puts it somewhere **not** at the top, which is a pain, so you have to DRAG IT PUT IT BACK TO THE TOP OF THE LIST to get it to compile. Also make sure the std68k.lib file is at the **bottom** of the list.

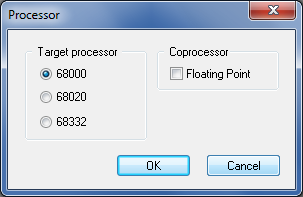
Once you have added your files to the project – click OK. You will be prompted to save the Project – give it a name and location.

**Setting the Compiler Options**

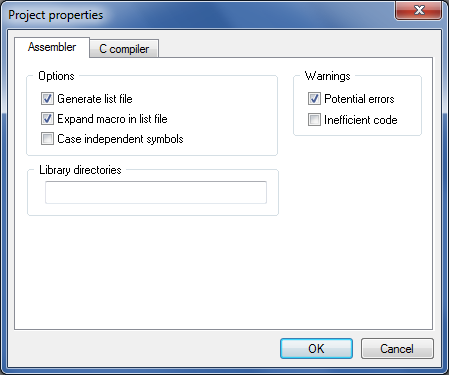
Chose Menu Options->Directories and fill in the boxes as shown below. The **default directory** name should be one of your own. This is where the compiler output will be placed.

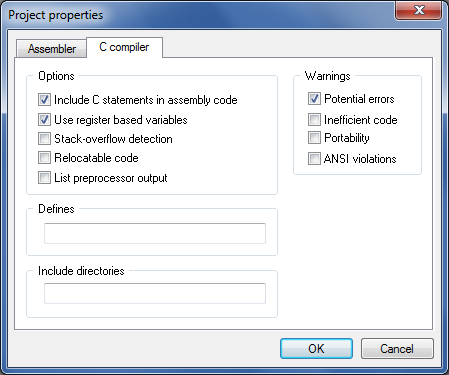


Chose menu Options->Processor and chose 68000 as the target processor (see below)



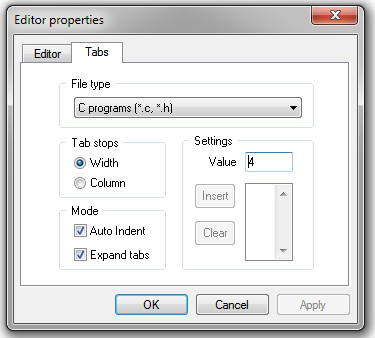
Chose menu Project->Properties and fill in the Assembler and Compiler options as shown below





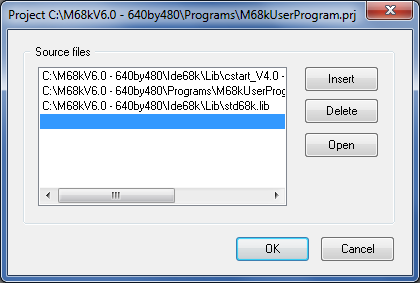
**Setting up the ‘C’ code editor**

Click on menu **Options->Editor** and the following diagram box opens, click on the Tabs tab and set the file type to C programs as shown below. Fill in the boxes as per the illustration. This sets the indentation so programs are neatly laid out.



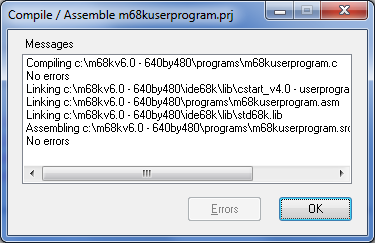
**Editing your files:**

Click menu Project->Edit Project. The following list or project files will appear. Click on one of the files e.g. M68kUserProgram.c and select Open



**Building the Project**

Press function key F7 to build. The IDE68k is very fast, it has finished almost before your key press has been released. You should see this if all compiled OK, if not, click on the errors button to locate and then correct the errors.



A number of files will have been produced

* A “.**masm**” file with the assembler output from the compiler (it puts the C in as comments so you can see how C code is translated into 68000 code)
* A “.**lst**” file. A full listing file produced by the linker
* A “.**src**” which is a source file listing
* A “.**prj**” the project file containing compiler set up/options/project file names etc.
* A “.**hex**” file which is the file we download to the DE1 board to download our program. This file is in the form of Motorola S-Records, a very popular standard. It looks something like this if you open it up with a simple editor

S004000000FB

S21480000046FC00002E7C0084000023FCFFFFFFFFE0

S214800010008013EC42B9008013F0207C008013EC43

S214800020227C00860000203C0000040E670000084A

S21480003012D8538066FA4EB90080073A60C21039EB

S21480004000400032C0BC000000FF0C8000000001B1

S214800050661A13FC00030040003210390080141228

S21480006052390080141213C000400000103900403E

S2148000700036C0BC000000FF0C8000000001661A3D

S21480008013FC0003004000361039008014145239E7

S2148000900080141413C00040000410390040003A59

S2148000A0C0BC000000FF0C8000000001661A13FC34

S2148000B000030040003A1039008014165239008040

S2 is the start of a record, followed by a 2 digit byte count (e.g. 14) followed by a 6 digit address (e.g.800000) followed by some data bytes, followed lastly by a two digit checksum at the end of the line. S3 records also exist (they have an 8 digit address suitable for a 32 bit address space)

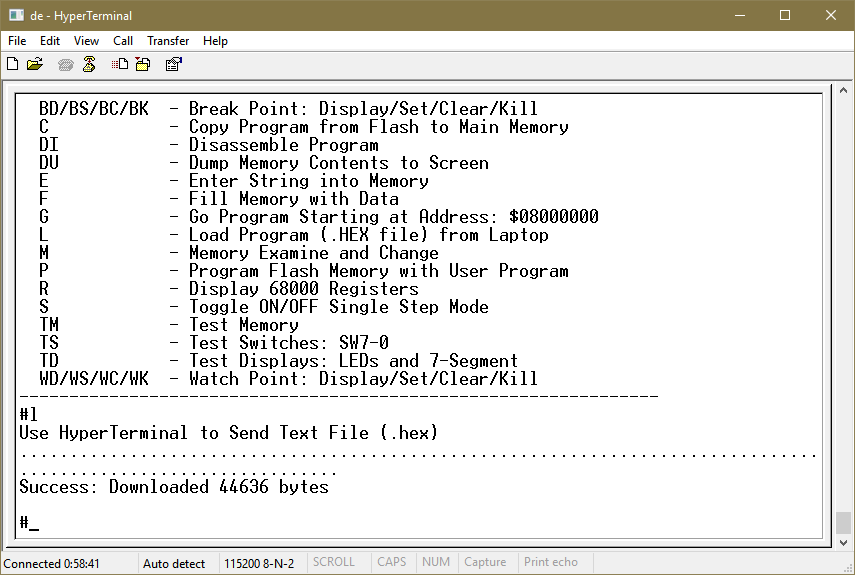
**Downloading the Program to the Altera Board**

Using the Hyper terminal software, make sure you have communicated with the Altera board, i.e. have the “#” prompt and can communicate with the debug monitor (if you haven’t downloaded a working 68000 soft core processor design/system, i.e. the ‘MC68k.sof’ file to the Altera board you **won’t** get this prompt).

At the debug monitor prompt ‘#’, enter the command ‘**L**’ for load. The board will then wait for you to download (see below).

Click on the HyperTerminal “**Transfer**” menu and select “**Send Text File**”. Locate the file in the dialog box. You are looking for a file with a “**.HEX**” file name extension.

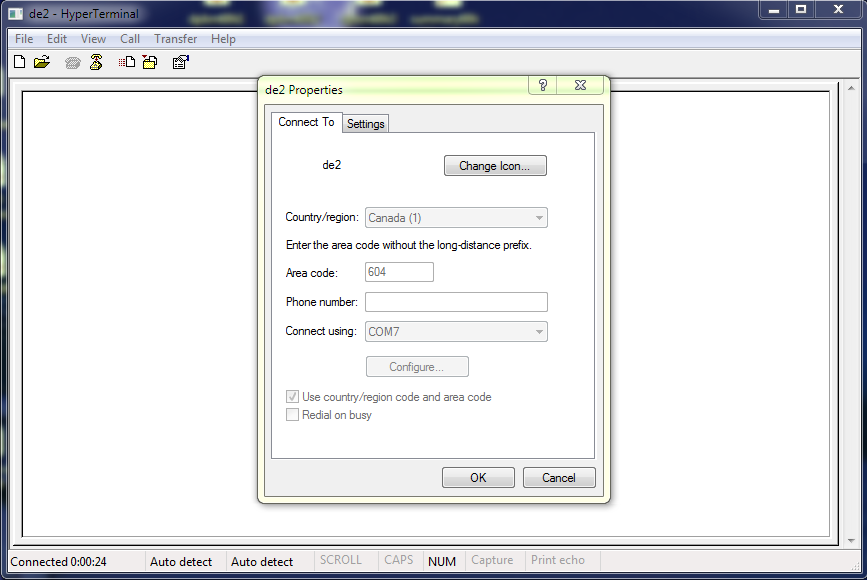
Once you have found it, click OK and you should see a screen similar to that below. The dots indicate progress as the S-Records are downloaded.



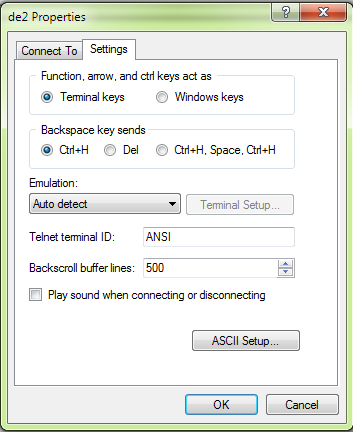
**What if the Download Fails?**

The vast majority of laptops will work fine, **BUT**, if you experience consistent problems downloading your program, experiment by making some changes within the **HyperTerminal** program as follows.

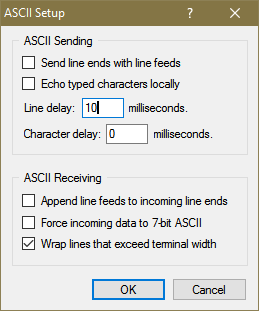
In **HyperTerminal**, click File->Properties, the following screen appears



Click on the Settings tab, the following appears



Click the ASCII Setup button, the following windows appears,



Experiment with the value of the Line Delay (e.g. 10) to pause transmission after each line of text is sent – try different values until you find one that works. You unfortunately will have to enter this again **each time** you start HyperTerminal (unless you buy the HyperTerminal program which allows you to save the settings).

**Running the Program**

At the “#” prompt, type ‘G’ for go. Press the Blue reset button the Altera board at any time to return you to the Debug prompt ‘#’.