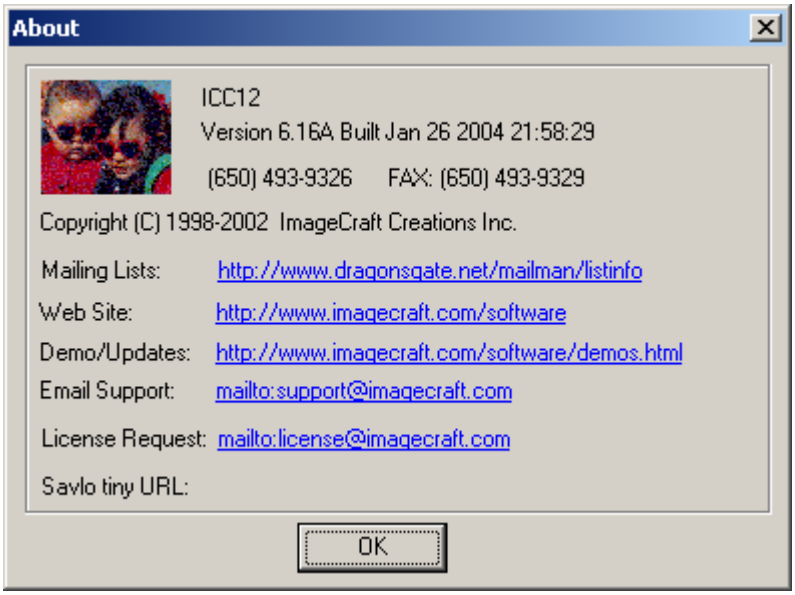
How to use ICC12 with neCore12M64 and uBUG12

This document will demonstrate the use of ImageCraft ICC12 ***Version 6*** C compiler with Technological Arts' Adapt9S12NE64 module. uBUG12 is used to erase and program flash memory after the compilation of a test program with ICC12. While other methods can be used to erase and program flash, this example will focus on uBUG12.

This document assumes that the user is already familiar with programming in C. It also assumes that the Serial Monitor has not been erased and is presently in the 9S12NE64 MCU.

ImageCraft Links:



http://www.imagecraft.com/software/
http://www.ece.utexas.edu/%7Evalvano
http://www.dragonsgate.net/FAQ/cache/20.html
http://www.imagecraft.com/software/mdevtools.html
http://www.dragonsgate.net/mailman/listinfo

Technological Arts Links:

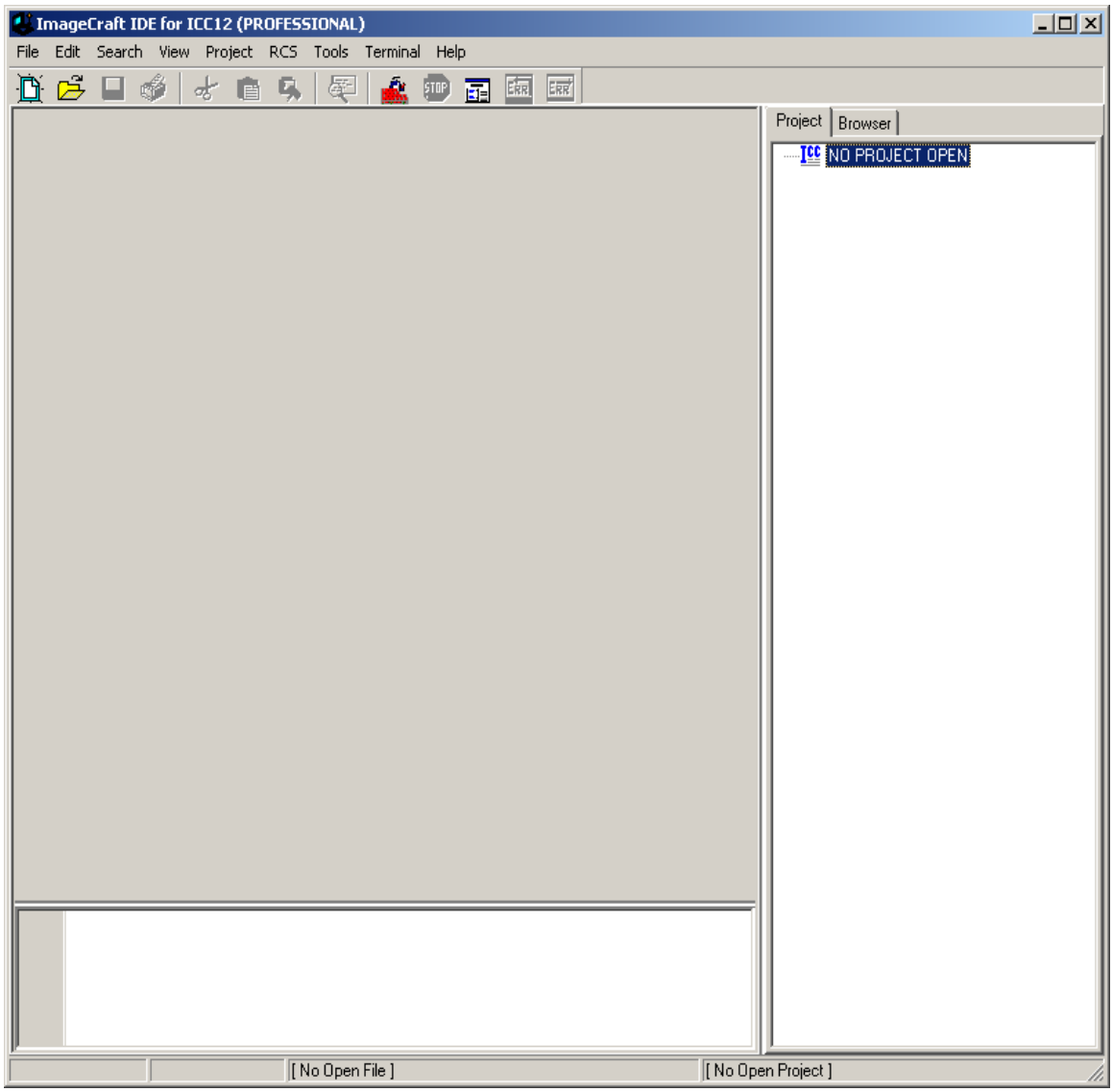
http://support.technologicalarts.ca/files/uBug12.zip

Getting Started:

Double click on the ICC12 icon. If you have not read the ICC12 manual yet, and you just opened the IDE, you will no doubt wonder what to do next. Well wonder no more.

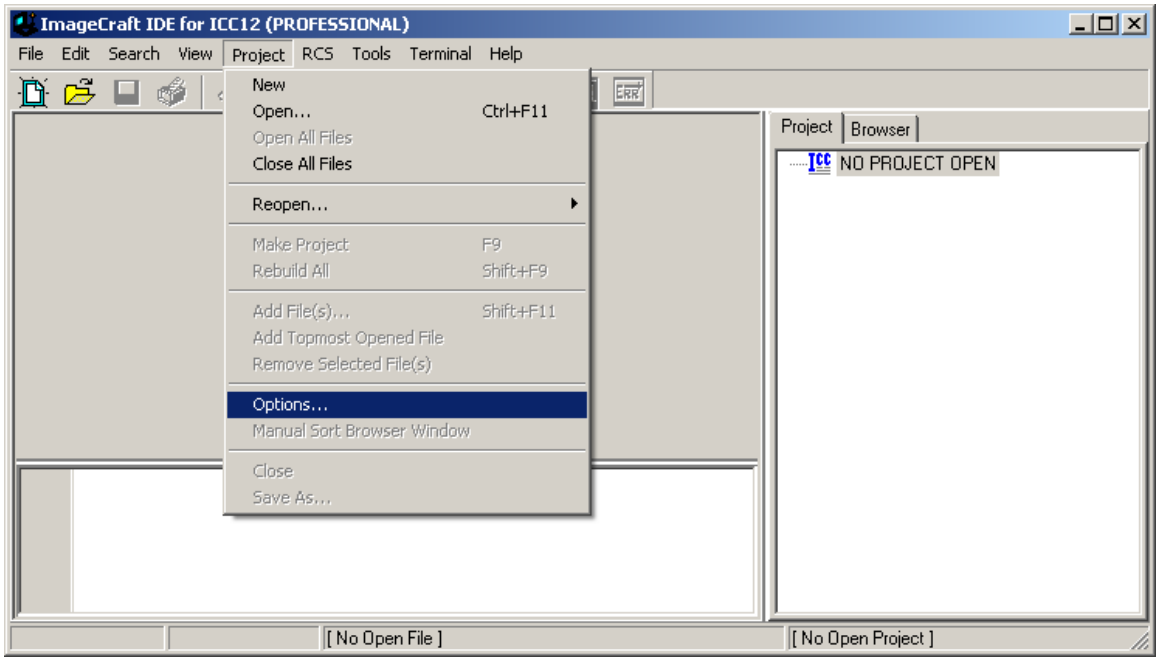
Note the three window panes. The top left pane is greyed out and the right pane displays the project window. The bottom left pane is where the error messages are displayed during compilation.

Before creating a new Project, the hardware target in the Compiler Options must be setup properly for the target MCU. This is to ensure that the compiler will know where program memory, data memory, and registers are located during the linking process. In this example it is neCore12M64.

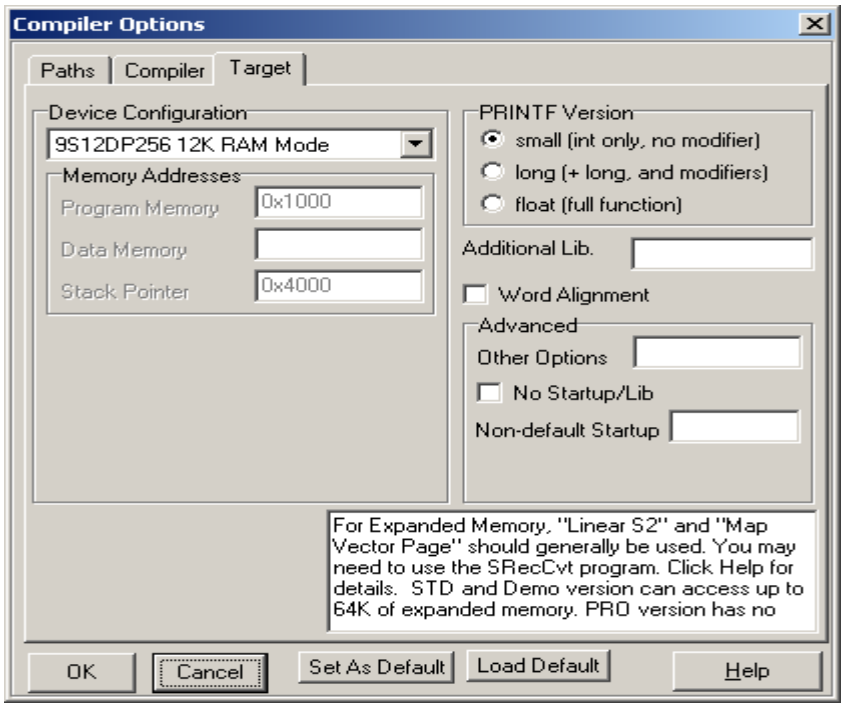


Compiler Setup:

Click on Project Menu – Options – Target Tab.

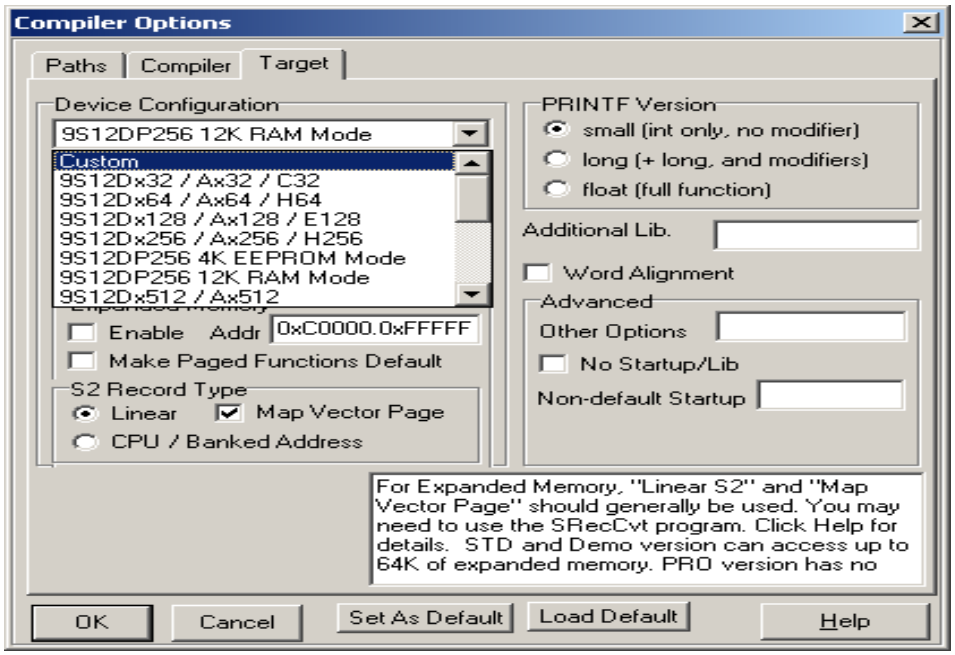


Please note that there is no 9S12NE64 listed in the Device Configuration therefore a Custom must be selected and the memory parameters set manually. Click on the pull down arrow to change the device type.



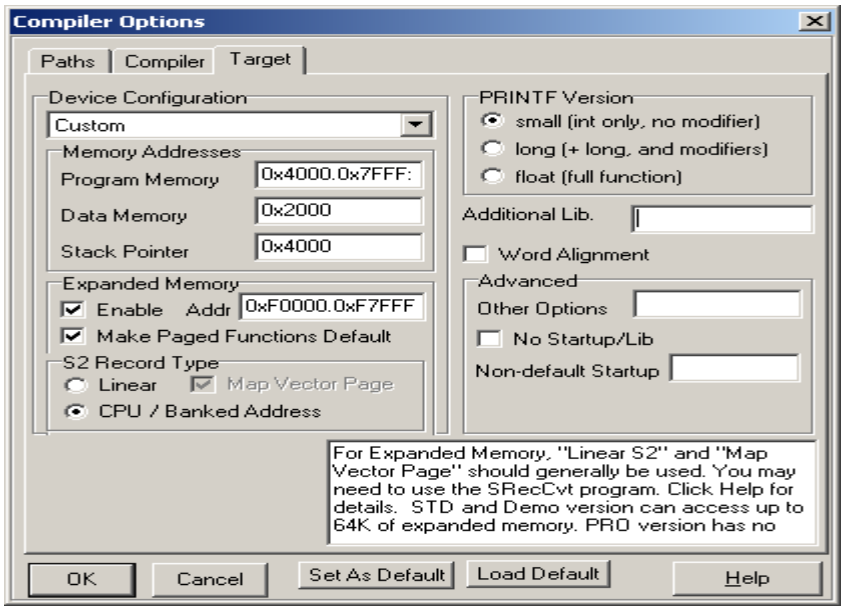
Custom Device Configuration:

Program Memory: ***0x4000.0x7FFF:0xC000.0xF7FF***
Data Memory: ***0x2000***
Stack Pointer: ***0x4000***

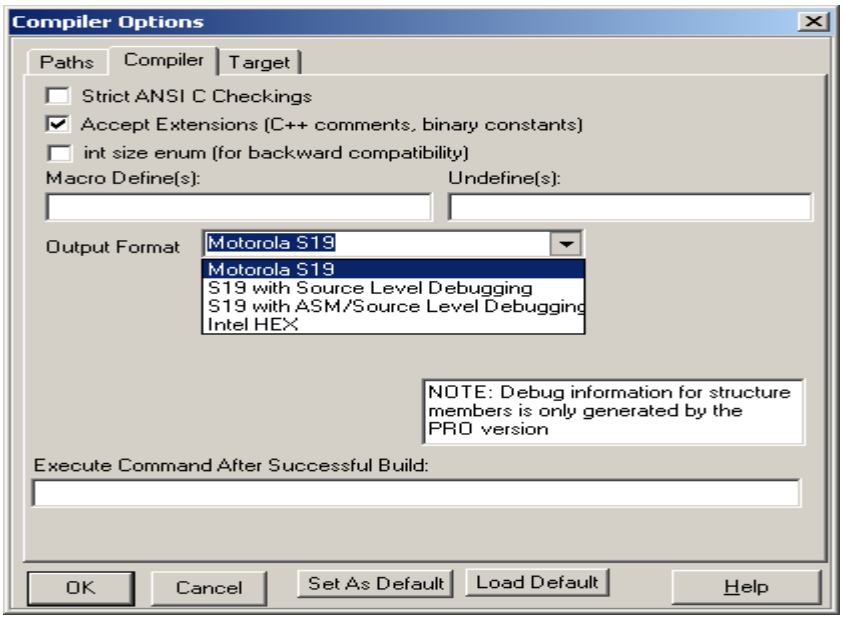


Expanded Memory:

Note the address range is ***0xF0000. 0xF7FFF***. That implies that the valid PPAGE range is from $3C to $3D. PPAGE $3E and $3F correspond to the fixed memory area, and are allocated to ***0x4000.0x7FFF:0xC000.0xF7FF***

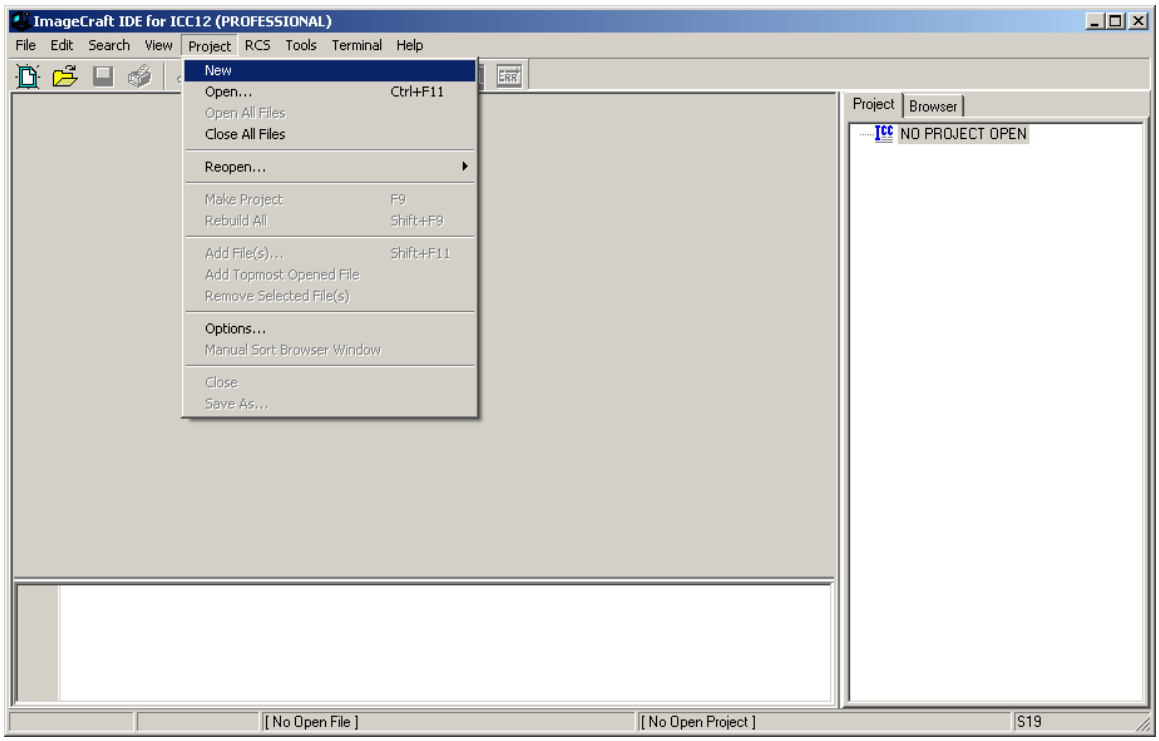


On the compiler tab there are several choices of S-record output as shown. Select one that suits you.

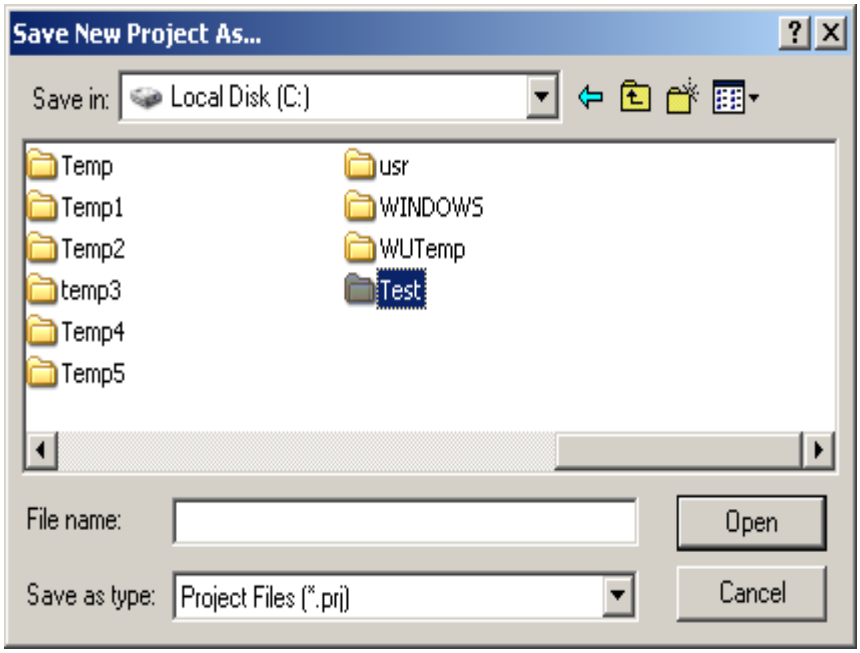


Starting a new Project:

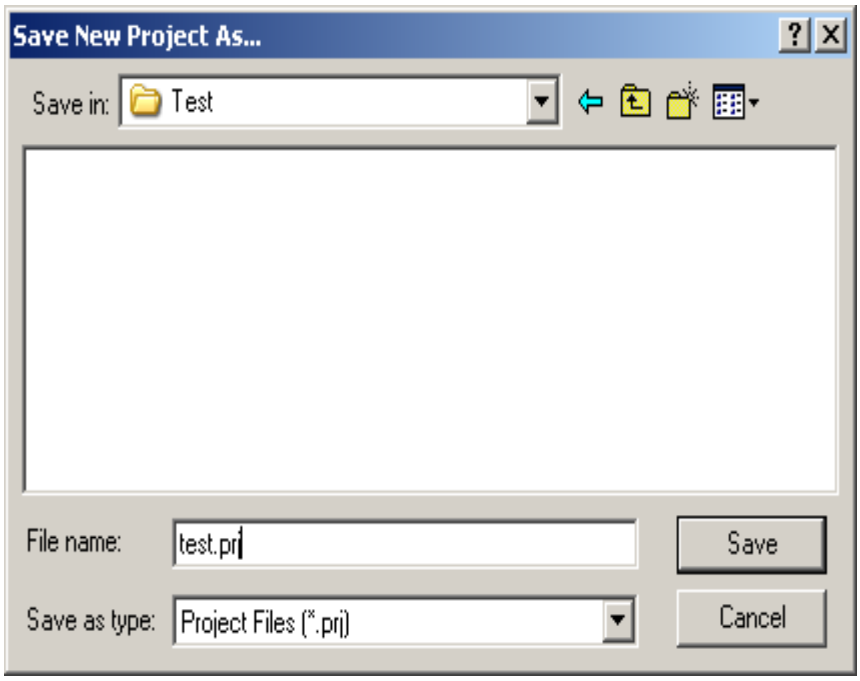
Once the compiler options are setup, a new project can be created. Click Project menu – New.



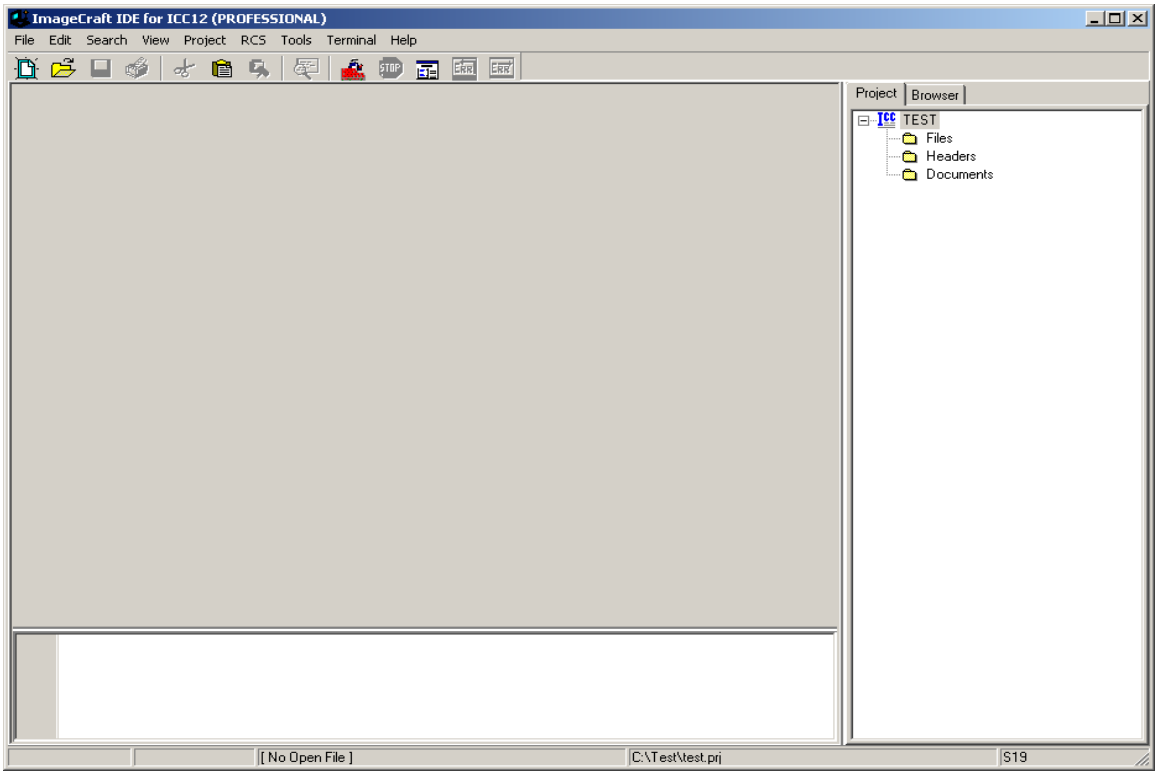
ICC12 will prompt to save the new project. You'll need to decide whether to create a new directory to save the new project. In this example a new directory called ***Test*** was created and the file was saved as ***test.prj***.



Type the filename ***test.prj*** and click on the Save button.

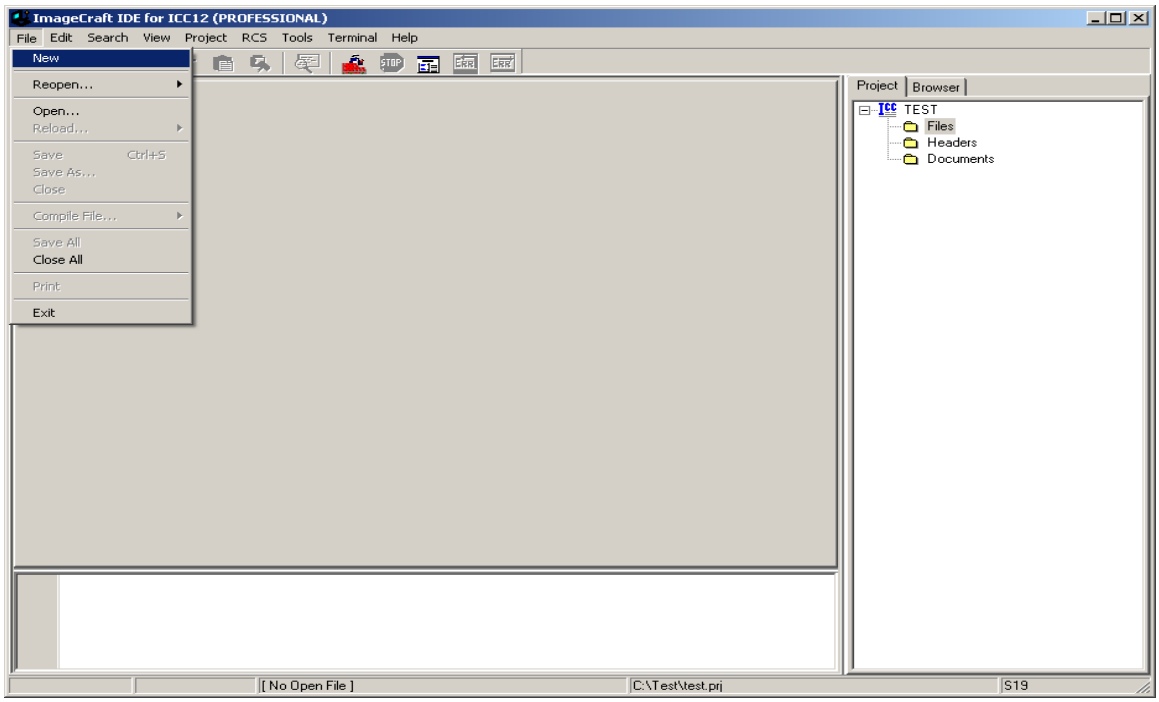


Note that the project window has changed to add Files, Headers and Documents.

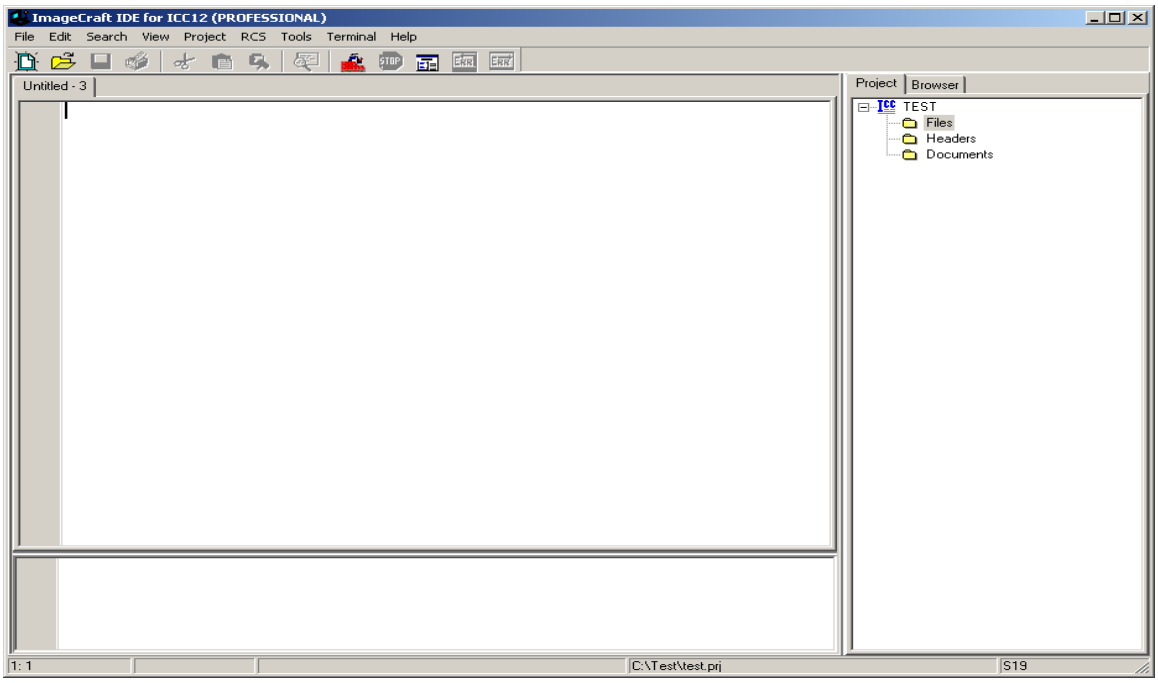


Adding new files to the project:

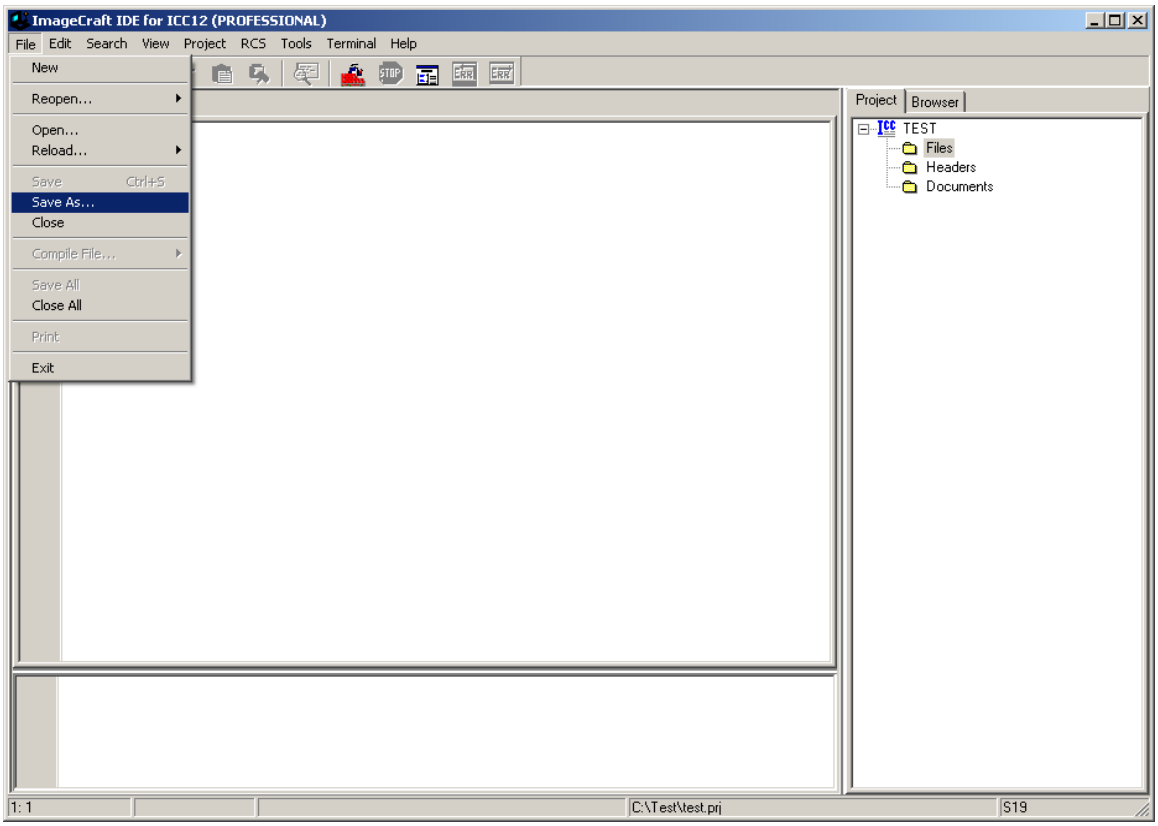
To add files to the project, click on the File menu and select New, as shown.



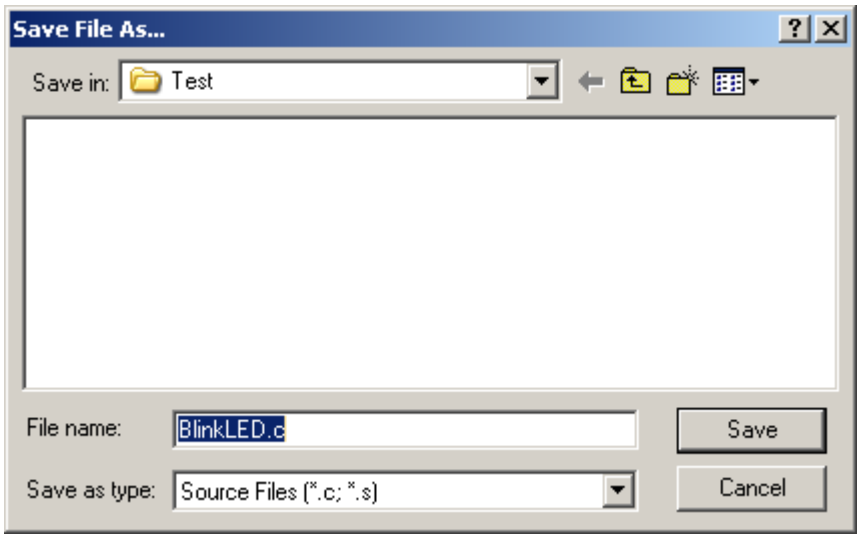
Note that ICC12 created an untitled file. Save the file as ***BlinkLED.C***.



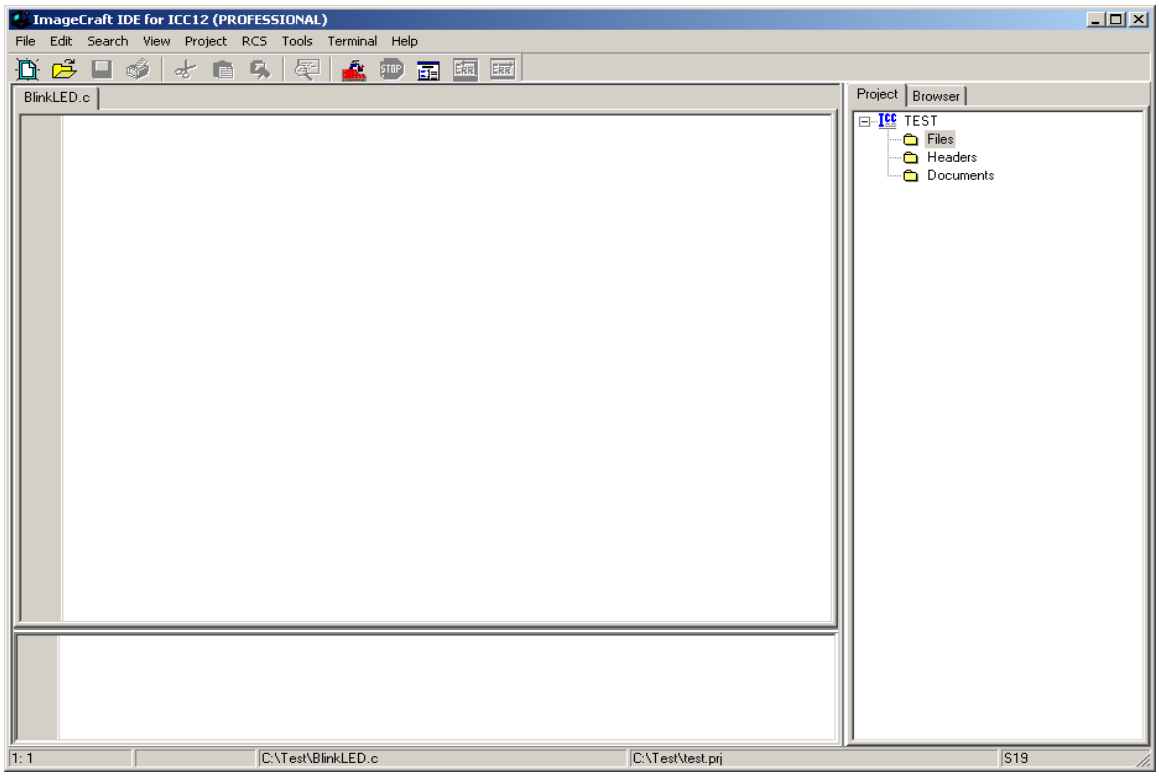
To save, click on File menu – Save As



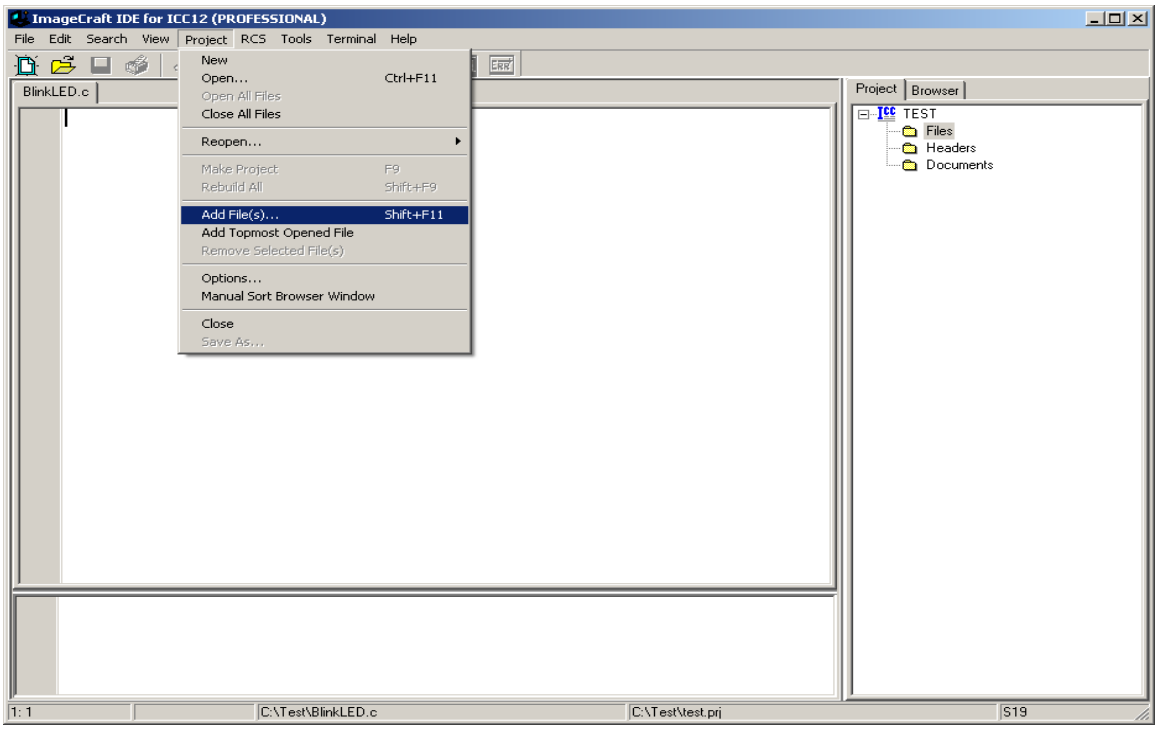
ICC12 will open an explorer window to help save the file. Type BlinkLED.c then press the save button.



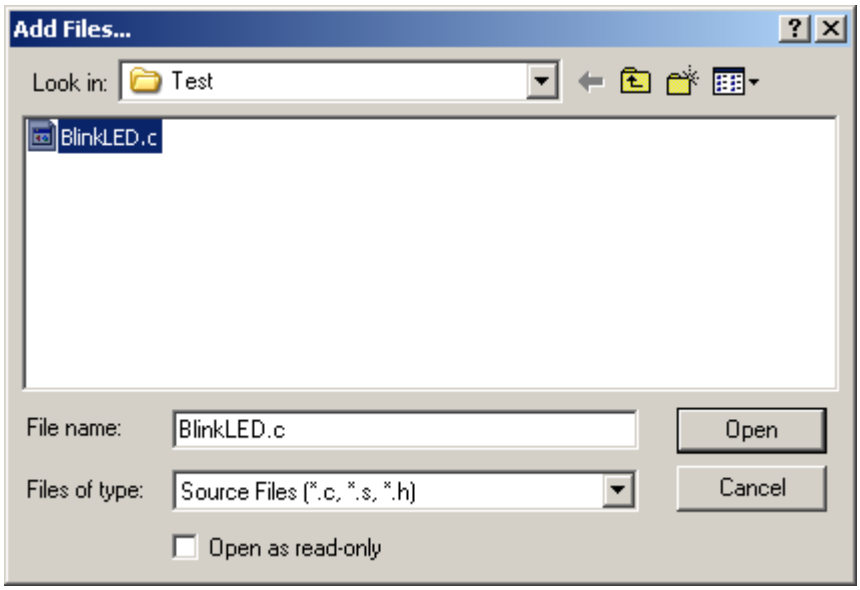
At this point ICC12 has renamed the file to BlinkLED.c.



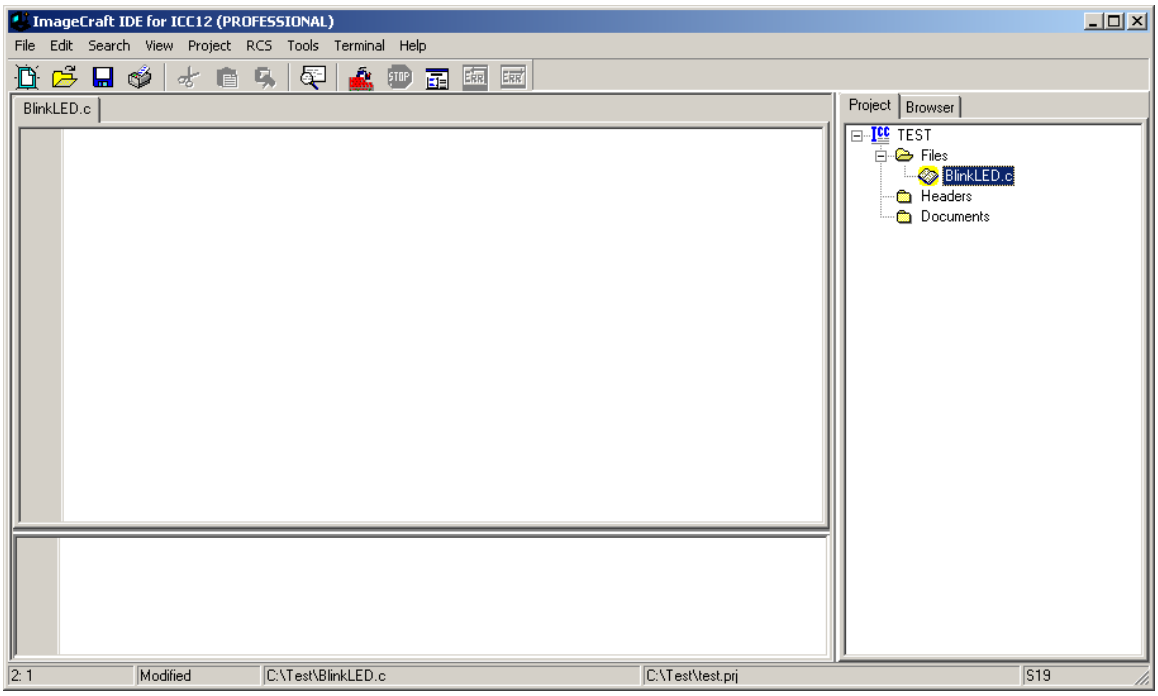
To add BlinkLED.c to the Project, click on the Project menu and select Add File(s)



ICC12 will open an explorer window to help you locate the file of interest.

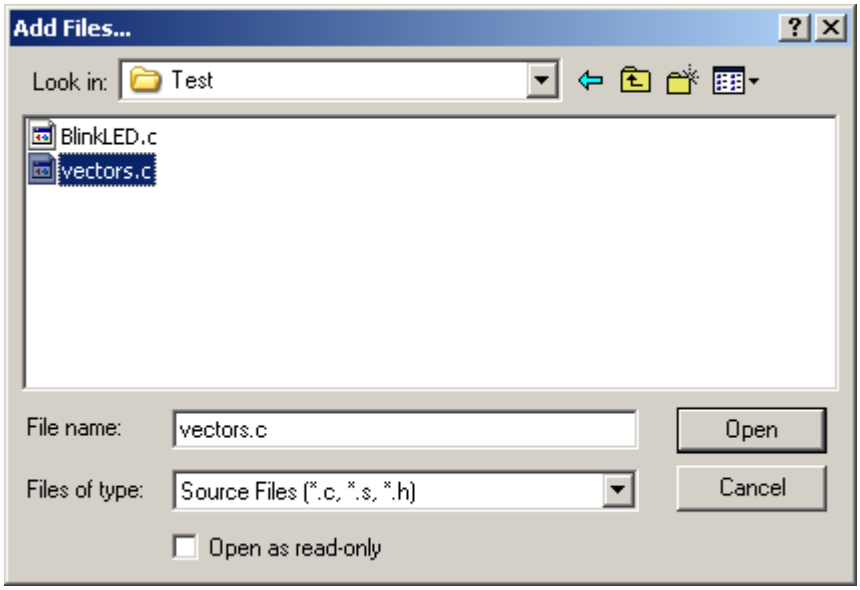


Notice that the right window pane has changed to include BlinkLED.c in the Project Files list.



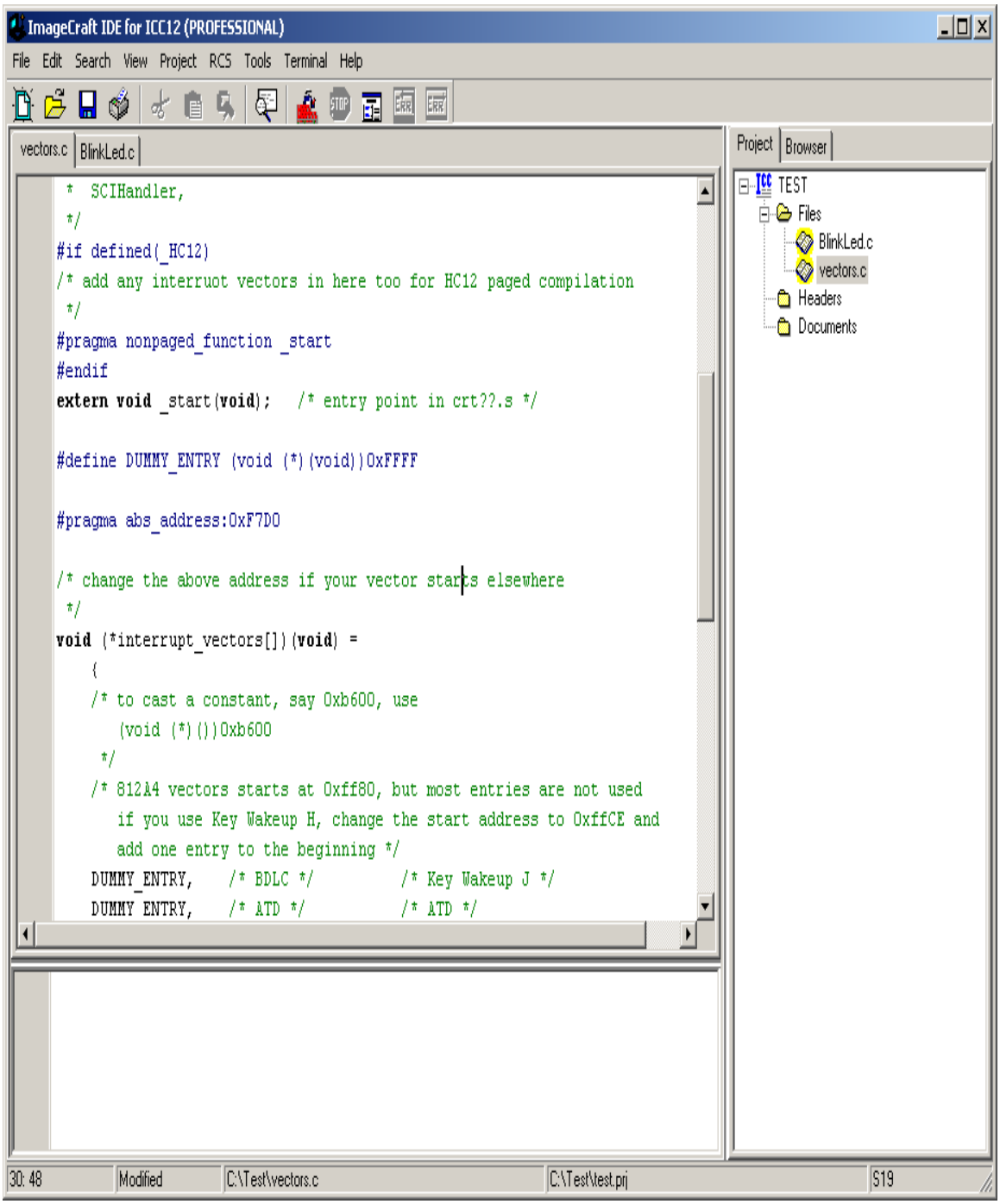
Locate ***vectors.c*** and copy it to Test directory. The main reason to do this is project dependencies. It is not good to keep editing a global ***vectors.c*** if other projects are using this same file. It becomes a problem to keep track of the changes made to the different projects.

To add ***vectors.c*** to the Project, click on the Project menu – Add File(s)



Note that the Project Files list has updated to include ***vectors.c*** It is important to note that the default **vectors.c** included with ICC12 was written for the 68HC912B32 and 812A4 MCUs. Technically, one should edit the file to include other interrupt service routine (ISR) addresses specific to the 9S12NE64. In this example the file is used “as is”.

Edit the line #pragma abs_address:0x***FFD0*** to #pragma abs_address:0x***F7D0***

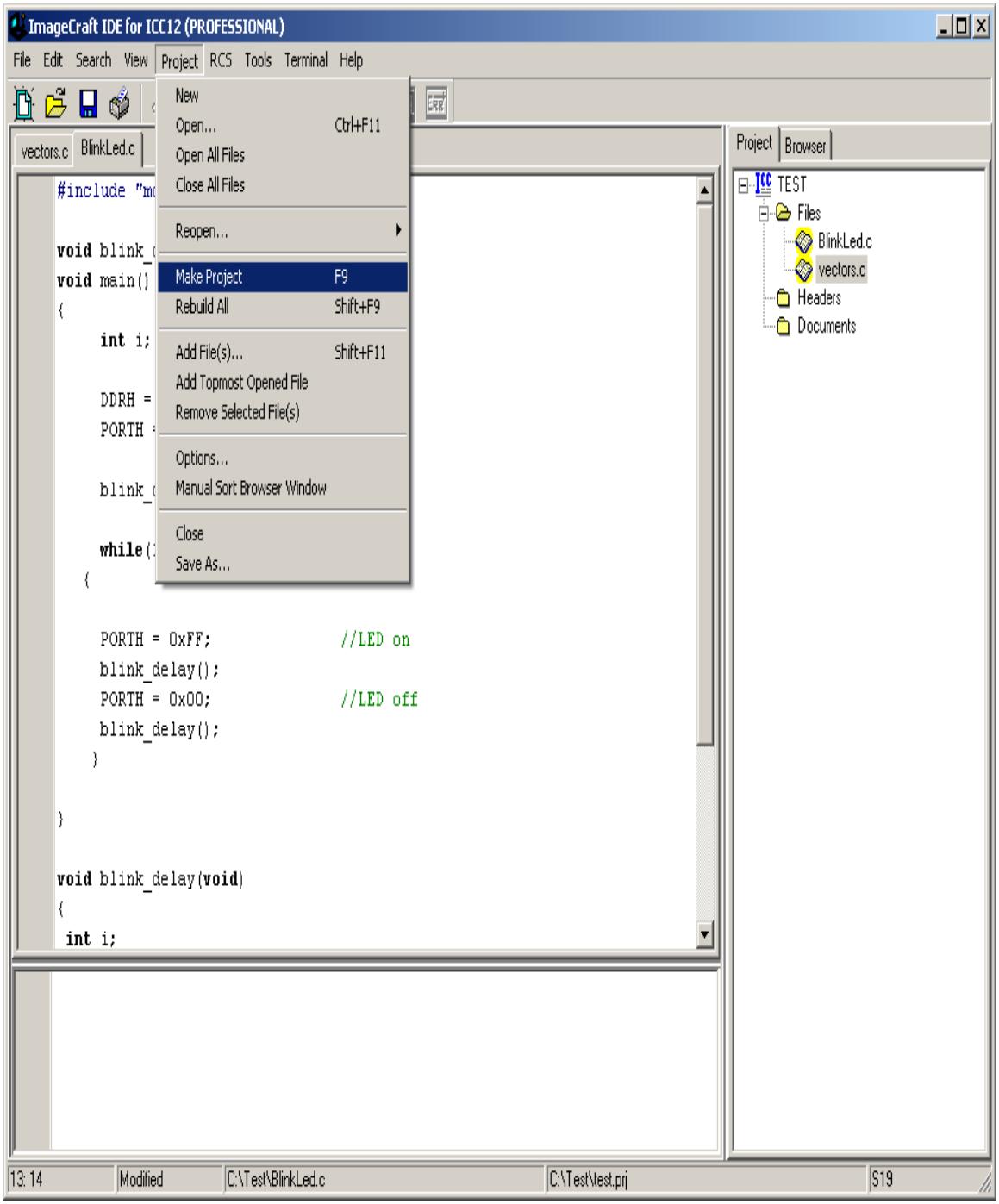


Enter the lines of program code shown below into the BlinkLED.c file. The next step is to compile/make/build the code.

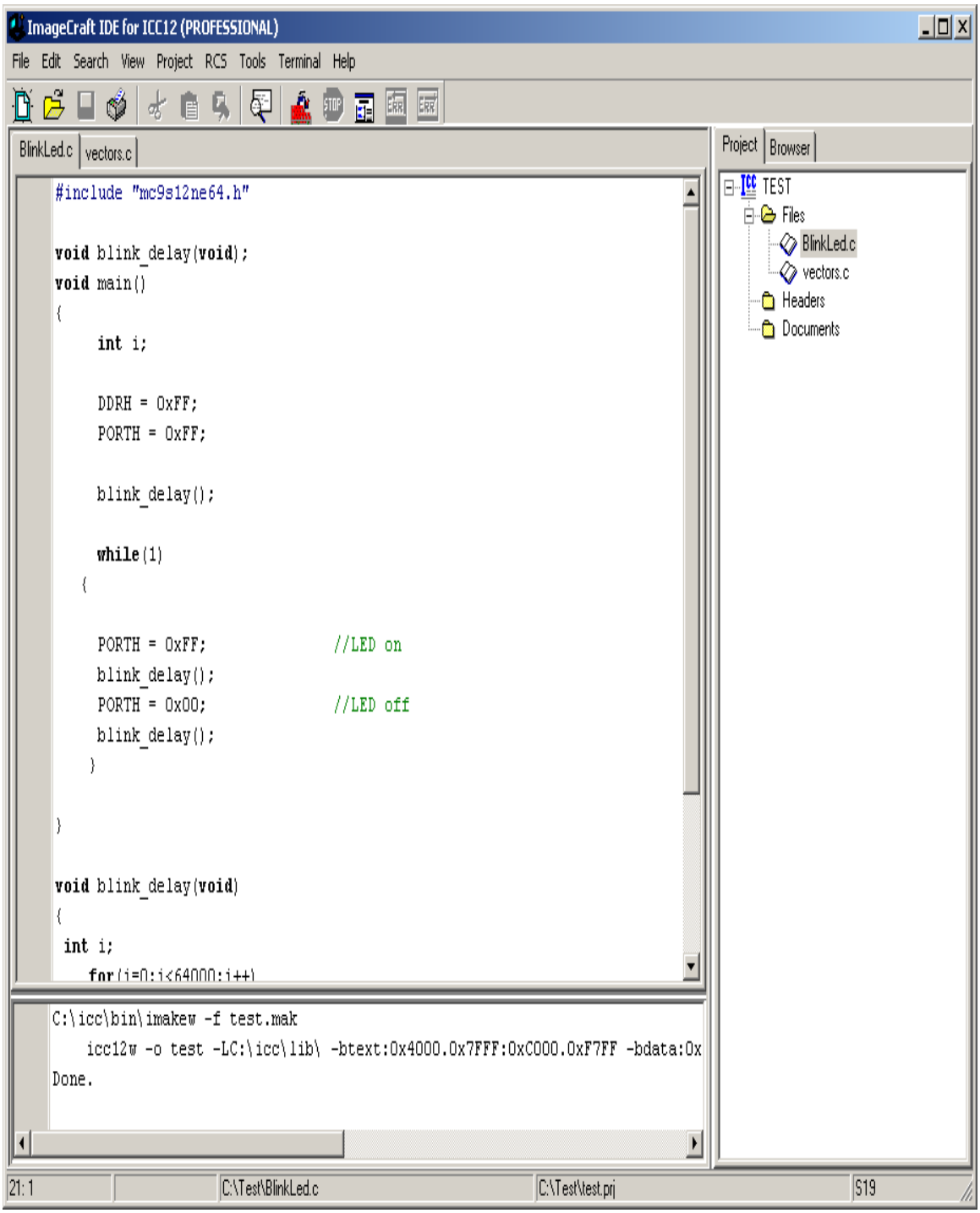
```
#include "mc9s12ne64.h"

void blink_delay(void);
void main()
{
       int i;

       DDRH = 0xFF;
       PORTH = 0xFF;

       blink_delay();

       while(1)
   {

       PORTH = 0xFF;                 //LED on
       blink_delay();
       PORTH = 0x00;                 //LED off
       blink_delay();
      }

}

void blink_delay(void)
{
 int i;
      for(i=0;i<64000;i++)
      {
                                ;
      }
}
```

Compile/Build/Make the file:

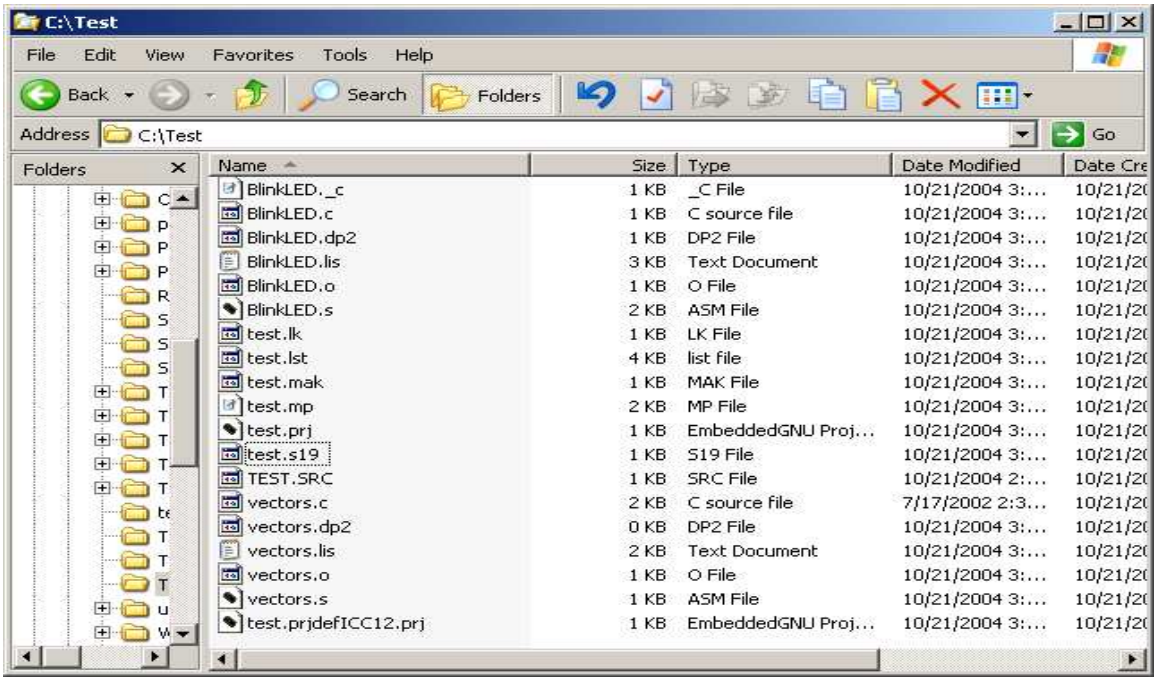
To make the file, click Project menu and select Make Project, as shown.



You'll notice that the bottom window pane shows messages to display how the build progressed. Any errors will be shown in this window. In this case, the build completed without error, so we can move on to erasing and programming the 9S12NE64.



Note in passing that some other intermediate files are created during a Make.



Use WordPad to open and inspect the content of ***test.s19***

S10E4000CF200016405587CE04008E30
S110400B040027056A000820F6CE405ACDB7
S111401804008E405A2706180A307020F51650
S1074026402A20FE0A
S110402A34B7751B9EC6FF7B025AC6FF7B90
S110403702584A80003C2010C6FF7B02584E
S10D40444A80003C7902584A8000CB
S10A404E3C20EEB757303DA2
S2123C800034B7751B9ECC00006C1E2007EC1E91
S2123C800EC300016C1EEC1E8CFA0025F2B75720
S2063C801C300AE7
S111F7D0FFFFFFFFFFFFFFFFFFFFFFFFFFFF35
S111F7DEFFFFFFFFFFFFFFFFFFFFFFFFFFFF27
S111F7ECFFFFFFFFFFFFFFFFFFFFFFFFFFFF19
S109F7FAFFFFFFFF4000C9
S10840551D0016073DEB
S9034000BC

If you look closely at the S-record you'll see a mixture of S1 and S2 lines. This is a typical file of S-records generated by ICC12. S1 records are programmed in the ***0x4000 – 0x7FFF*** and ***0xC000 – 0xF7FF*** memory blocks. ISRs are always placed in the fixed memory region. An ISR can call any routine inside a PPAGE when necessary. S2 records can also reference fixed memory regions, but are typically paged by ICC12

Below is the vector address as S1 record.

S111F7D0FFFFFFFFFFFFFFFFFFFFFFFFFFFF35
S111F7DEFFFFFFFFFFFFFFFFFFFFFFFFFFFF27
S111F7ECFFFFFFFFFFFFFFFFFFFFFFFFFFFF19
S109F7FAFFFFFFFF4000C9

Note the address at 0xFFFE|0xFFFF contains 0x4000, telling the MCU where to start executing code following power up or RESET.

S109F7FAFFFFFFFF***4000***C9

The S-record below shows the actual first few bytes of code in the program

S10E***4000***CF200016405587CE04008E30

ICC12 also generated a banked S2 record with S-record.

S212***3C***800034B7751B9ECC00006C1E2007EC1E91

PPAGE = 0x3C

S2123C***8000***34B7751B9ECC00006C1E2007EC1E91

Memory address = 0x8000.

Programming:

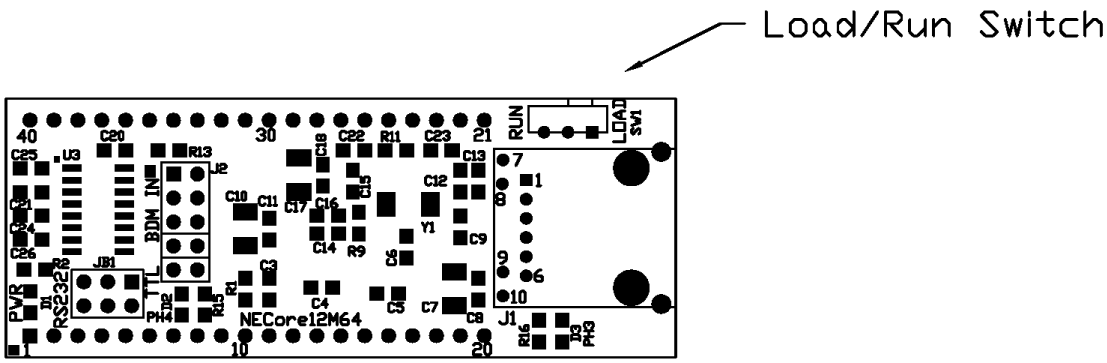
This document assumes that the Serial monitor is present on neCore12M64. It also assumes that one is using a School Board or the RX and TX line of the module is connected to PC COM port appropriately.

Find uBug12 on the CD that came with your evaluation or demo package. If you don't have it, you can download uBUG12 from Technological Arts, at http://support.technologicalarts.ca/files/uBug12.zip

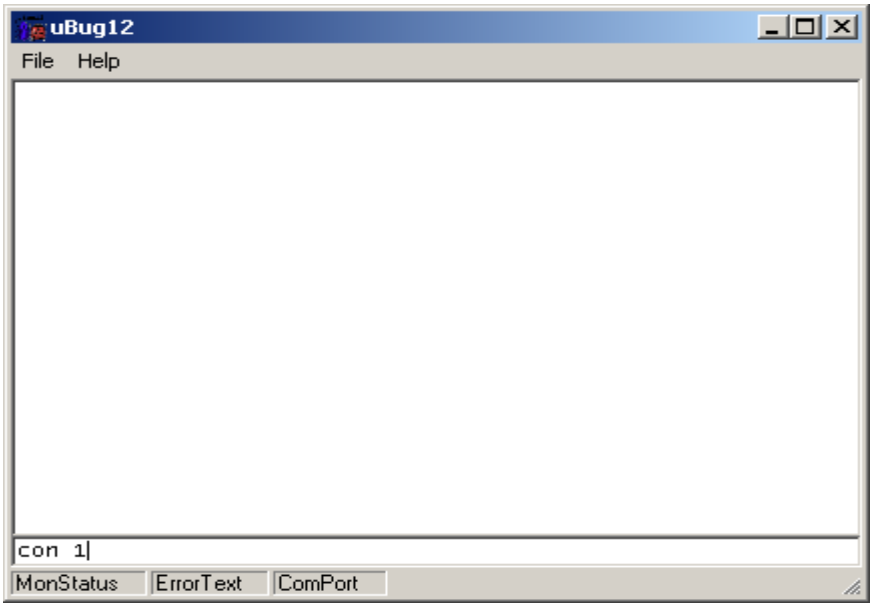
For Windows98 users the .NET framework must be installed before running uBUG12. The Microsoft site link is

http://www.microsoft.com/downloads/details.aspx?FamilyID=d7158dee-a83f-4e21-b05a-009d06457787&displaylang=en

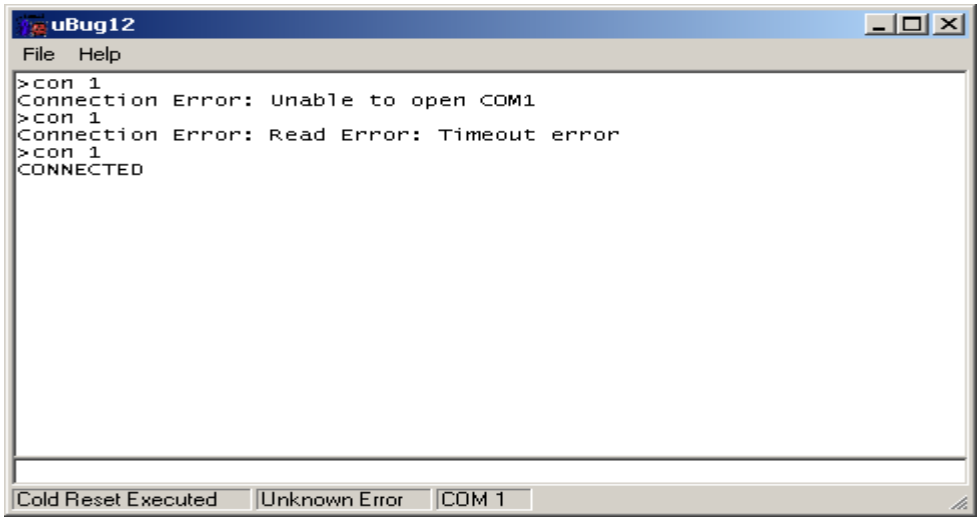
After installing uBug12, slide the Run/Load switch on neCore12M64 to the Load position, and apply power to the School Board.



Double click on the uBUG12 icon to launch it.

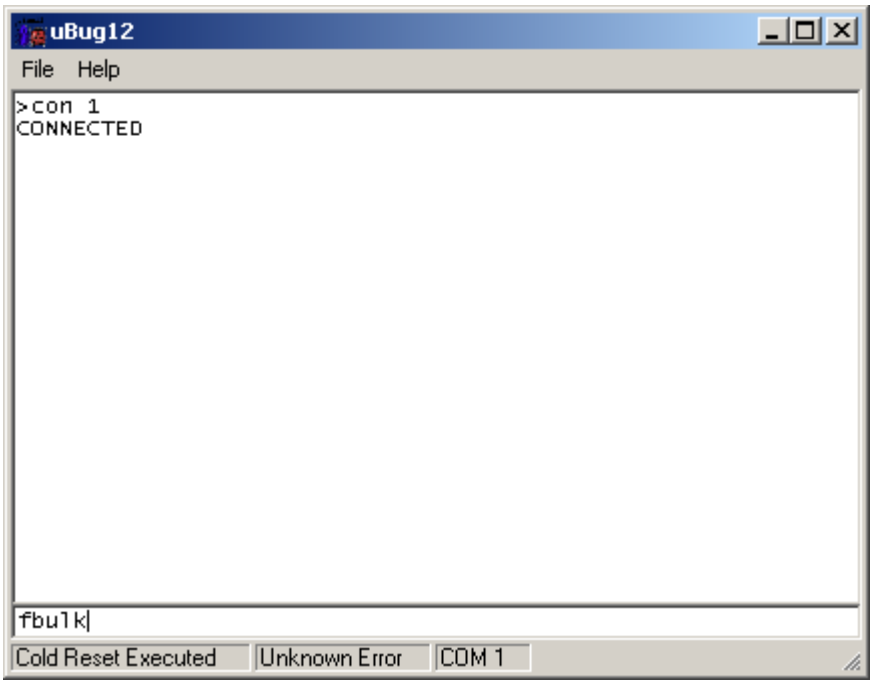


In the command bar type ***con 1*** to connect to COM 1 (or ***con 2*** if you're using COM 2). A ***CONNECTED*** message will appear to indicate that a connection between your PC and neCore12M64 has been established.

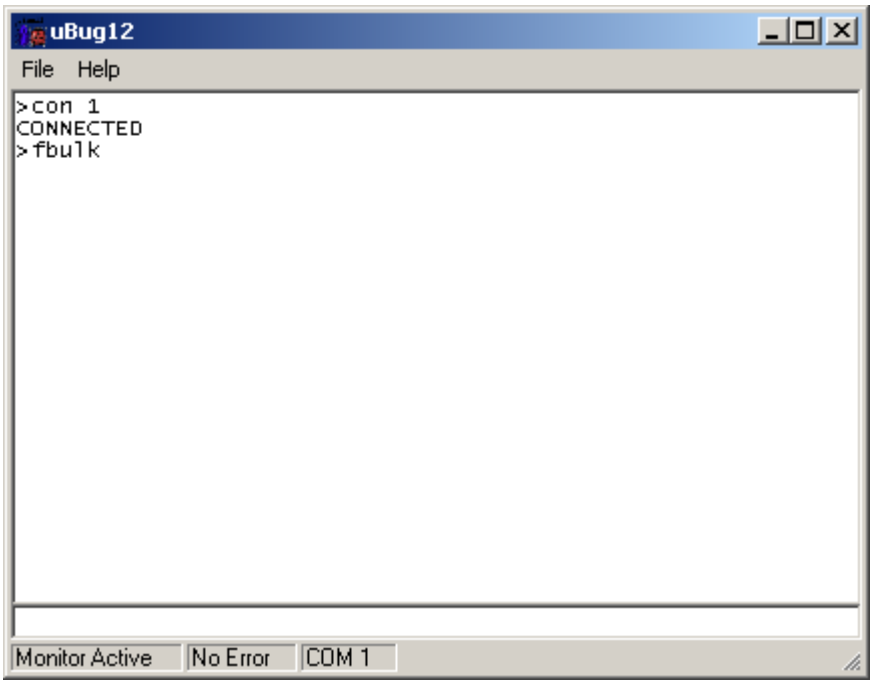


Two possible errors can occur:
Connection Error: Unable to open COM1 <- Another application is using the COM port

Connection Error: Read Error: Timeout error <- The MCU is not currently in LOAD mode, not powered up, or the cable is disconnected from either the PC or the E128 board.



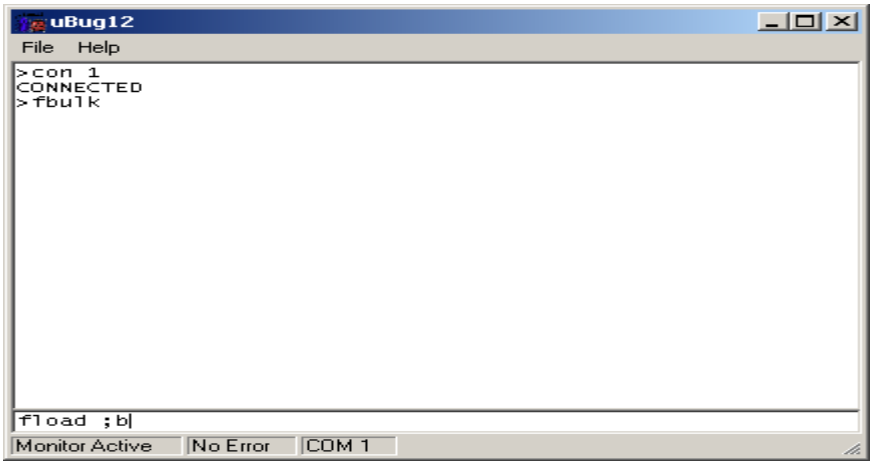
Now erase the flash memory by typing the command ***FBULK***.

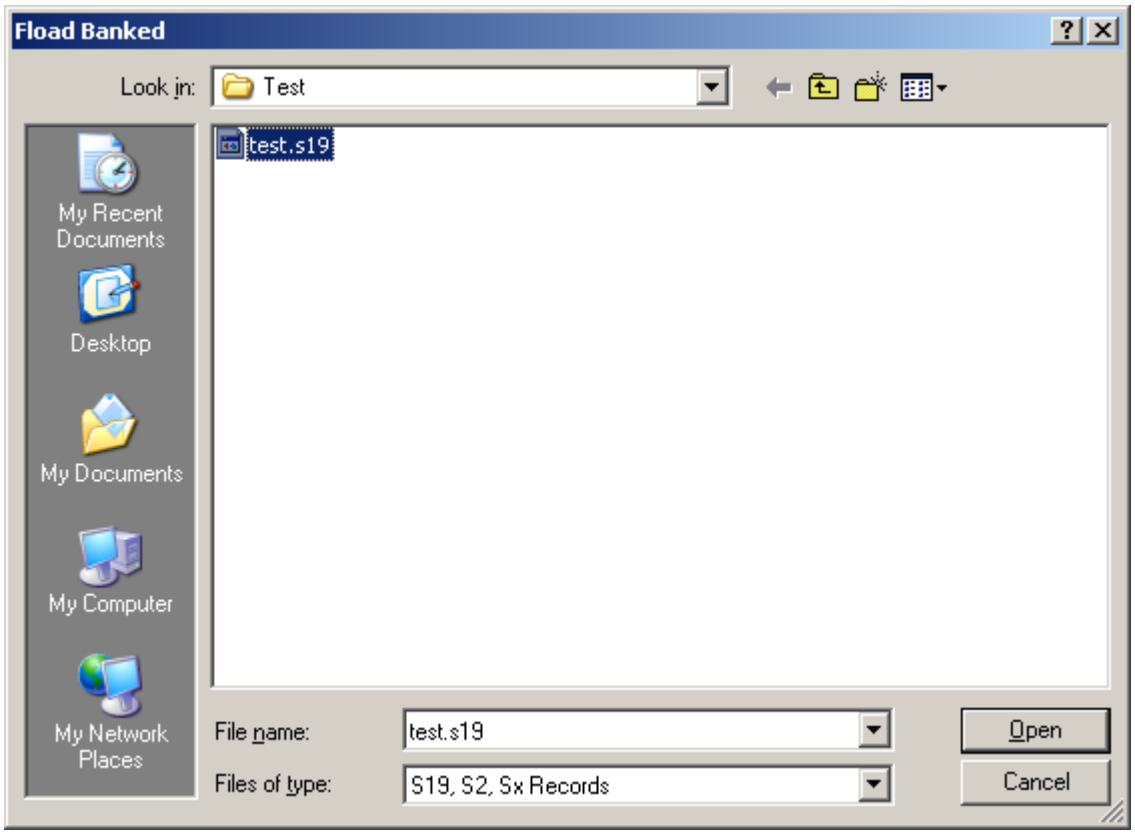


To load your program, type the command ***FLOAD ;B*** for banked S19, S2, SX and formatted S19 (i.e. went thru SrecCVT program) records. To load a file containing non-banked S2 records, the command is ***FLOAD***.

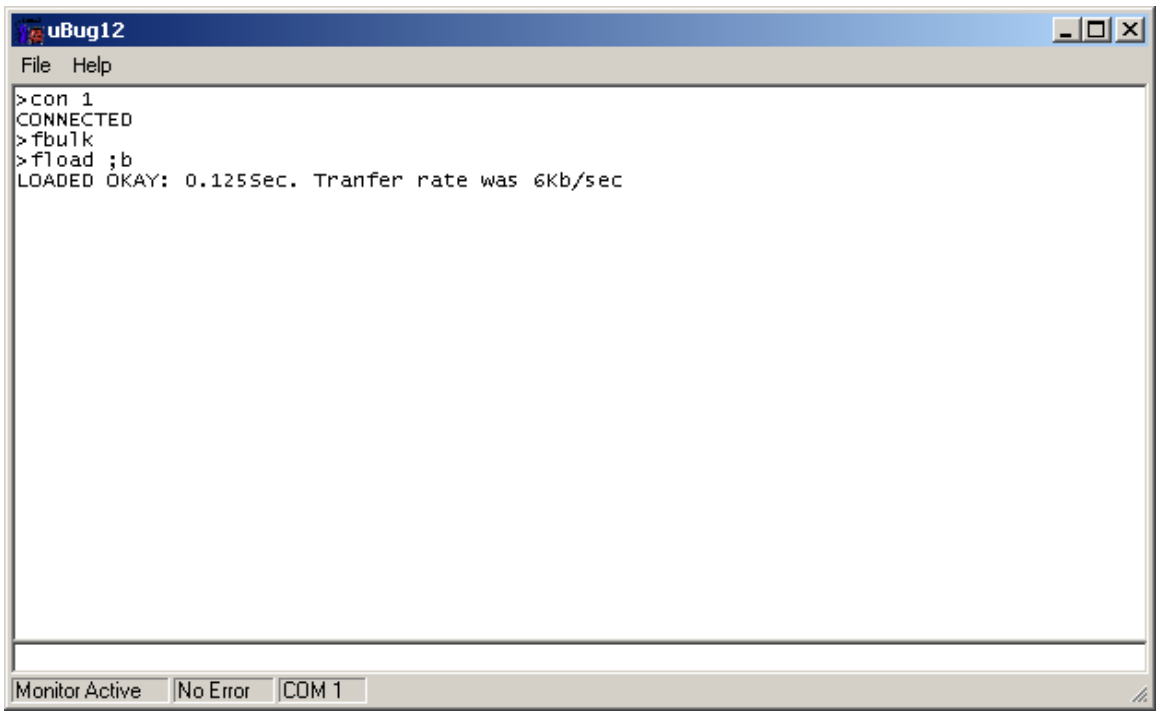
Uploading Banked S-record:

The command to upload banked S-records is ***FLOAD ;B***. It is important to include the ***;B*** option to let uBUG12 know that the S-record is banked. Make sure you become familiar with the differences between S19, SX, S2. See Appendix A for an explanation of S-records.





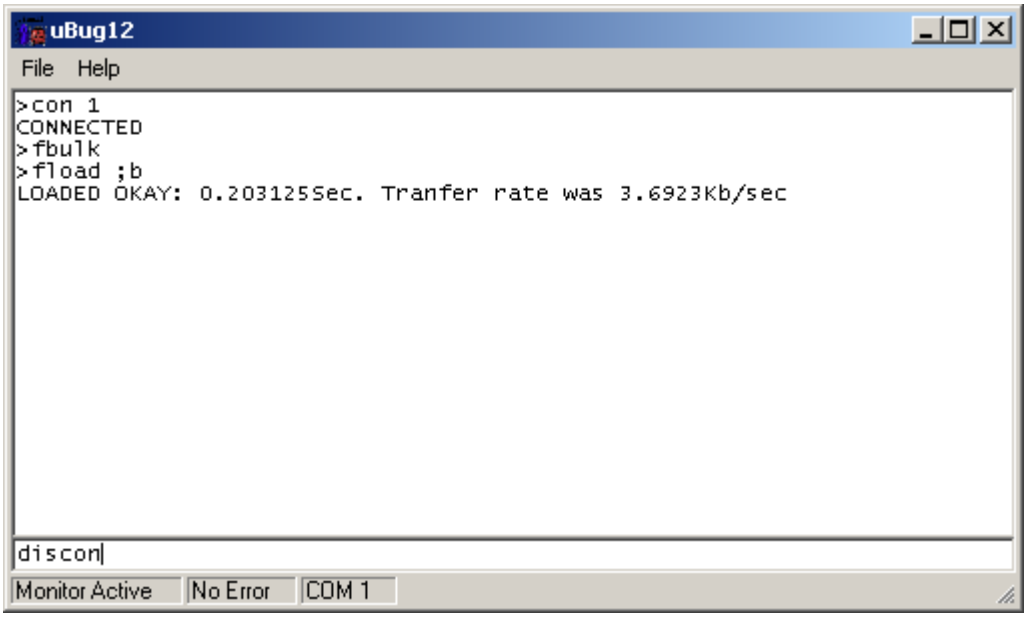
Double click on the file to initiate upload.



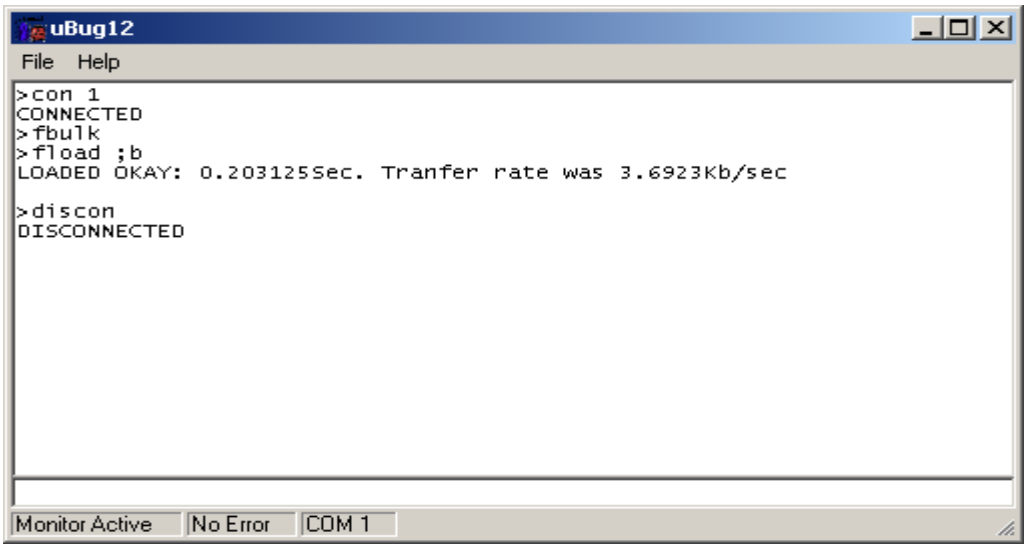
A good upload will show ***LOADED OKAY*** messages.

After successful programming, slide the Run/Load switch to Run and press the reset button. The application will begin blinking the LED connected to the port pin.

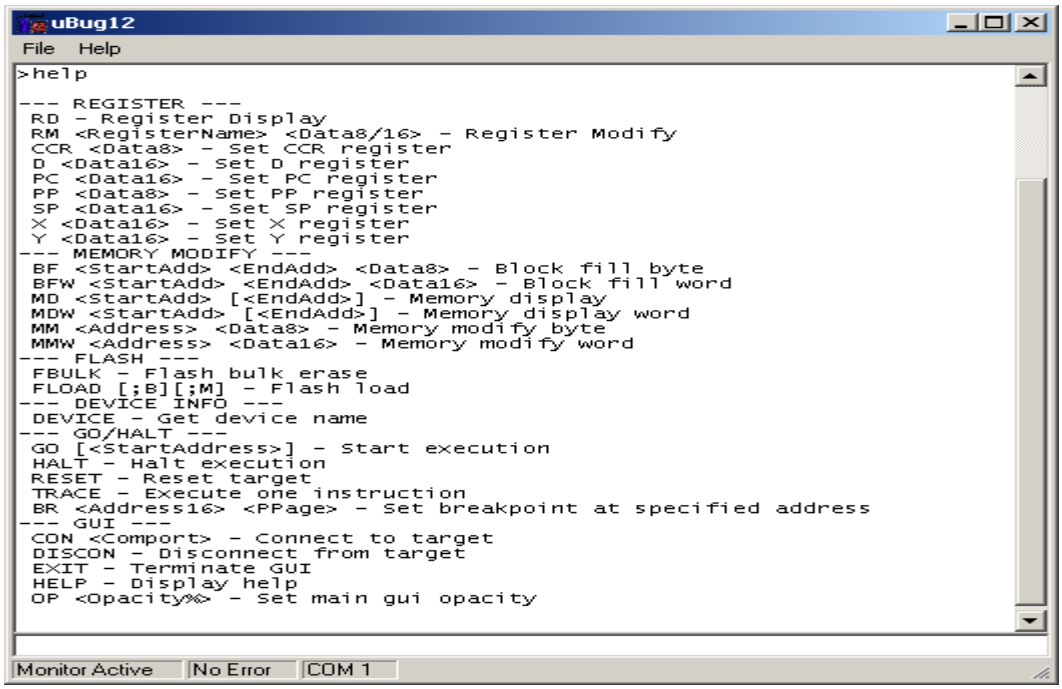
To disconnect uBUG12 from the serial port type the command ***discon***.



A Disconnected message will appear to indicate that the serial port is available for use by another application (eg. ***HyperTerm*** or ***Tera Term***).



A list of other uBUG12 commands can be viewed by typing the ***help*** command.



The commands are pretty well self explanatory but you should try them out to be familiar with their usage and capability.

This concludes the use of ICC12 from erasing and programming FLASH with using uBUG12.