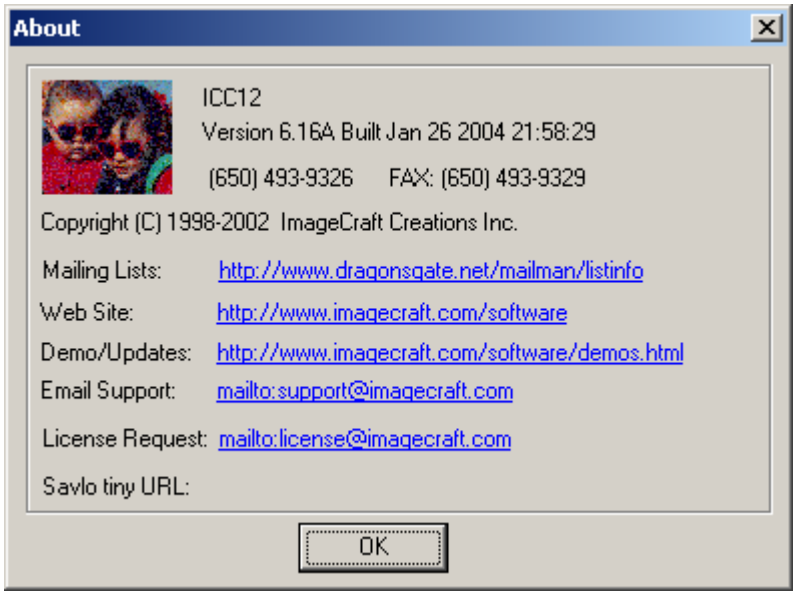
How to use ICC12 with Adapt9S12XDP512 and BDM Pod

This document will show and demonstrate the use of ImageCraft ICC12 ***Version 6*** with the Technological Arts Adapt9S12XDP512 module.

The XGATE BDM pod (MicroBDM12XG) will be used here to erase and program Flash after the compilation of a test program. While there are other methods to erase and program the Flash, this example will use the XGATE BDM pod.

This document assumes that the user is familiar with C and so will not teach how to program C here.

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://www.technologicalarts.ca/catalog/index.php
http://www.technologicalarts.ca/catalog/index.php/cPath/50_154

BDM Pod:

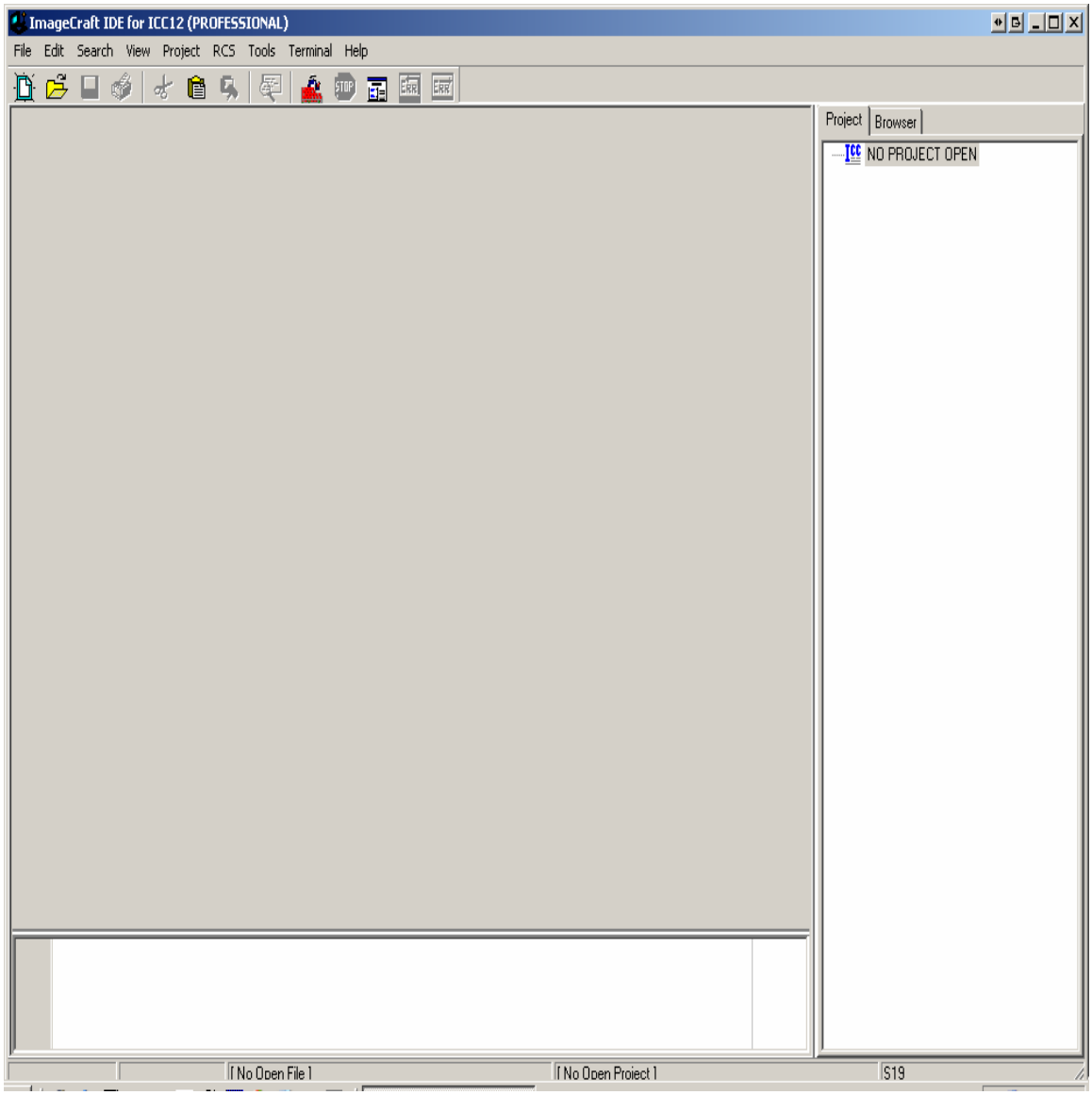
http://www.technologicalarts.ca/catalog/product_info.php/cPath/50_55/products_id/366

Getting Started:

Double click on the ICC12 icon to launch the IDE.

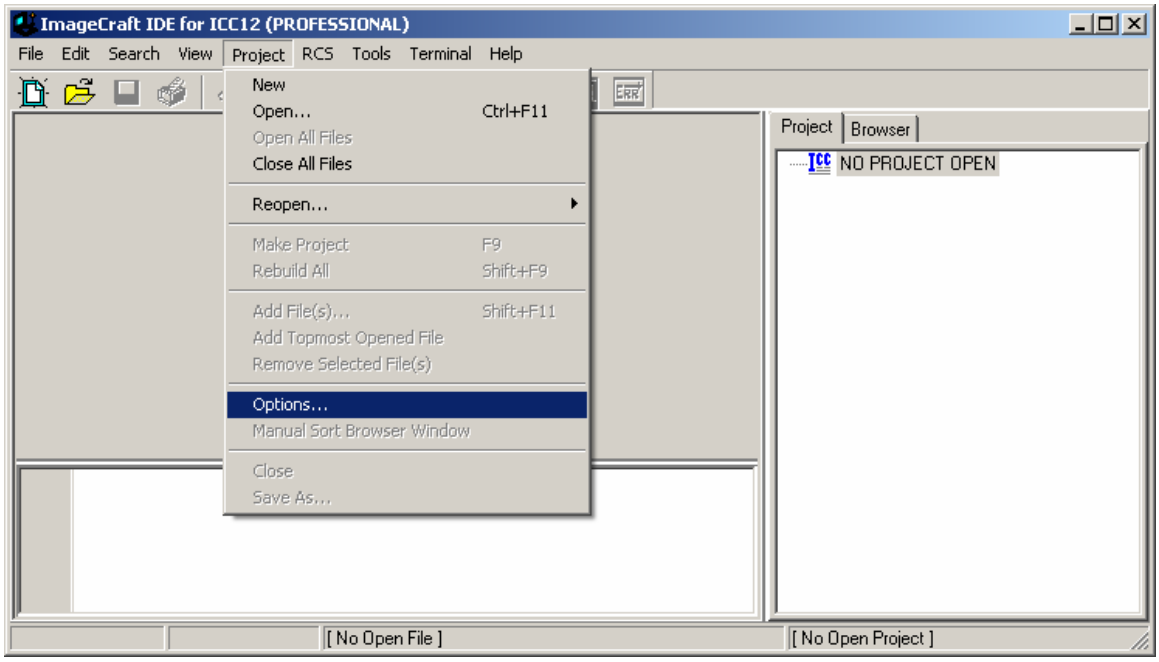
Note the three window panes. The top right pane is the project window. The lower left ane is where the status and error messages are displayed during compilation.

Before creating a new Project, the hardware target in the Compiler Options must be setup properly for the target MCU. In this example it is Adapt9S12XDP512.

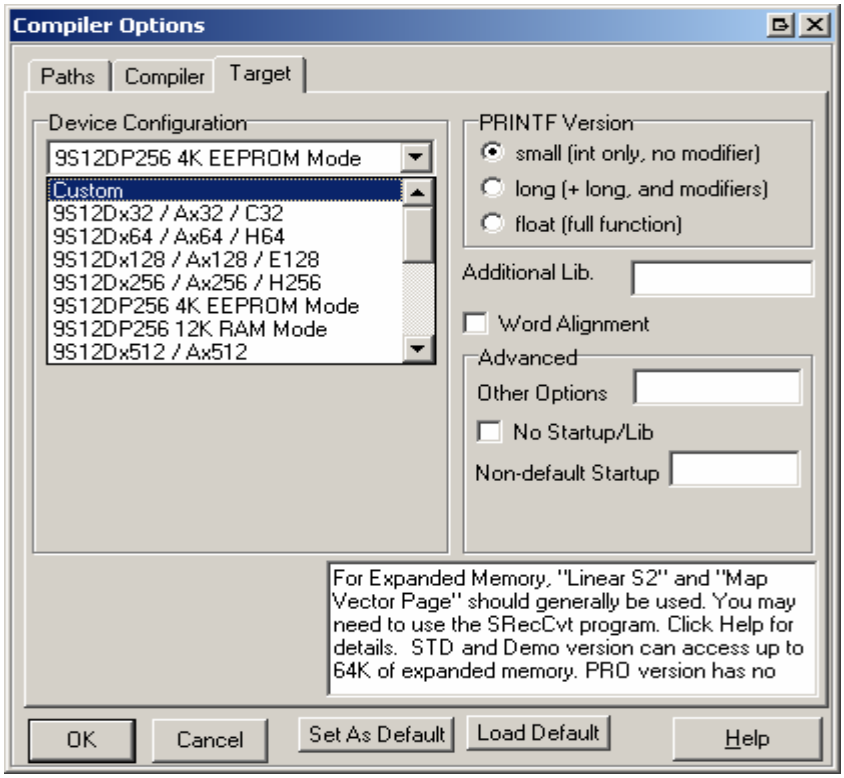


Compiler Setup:

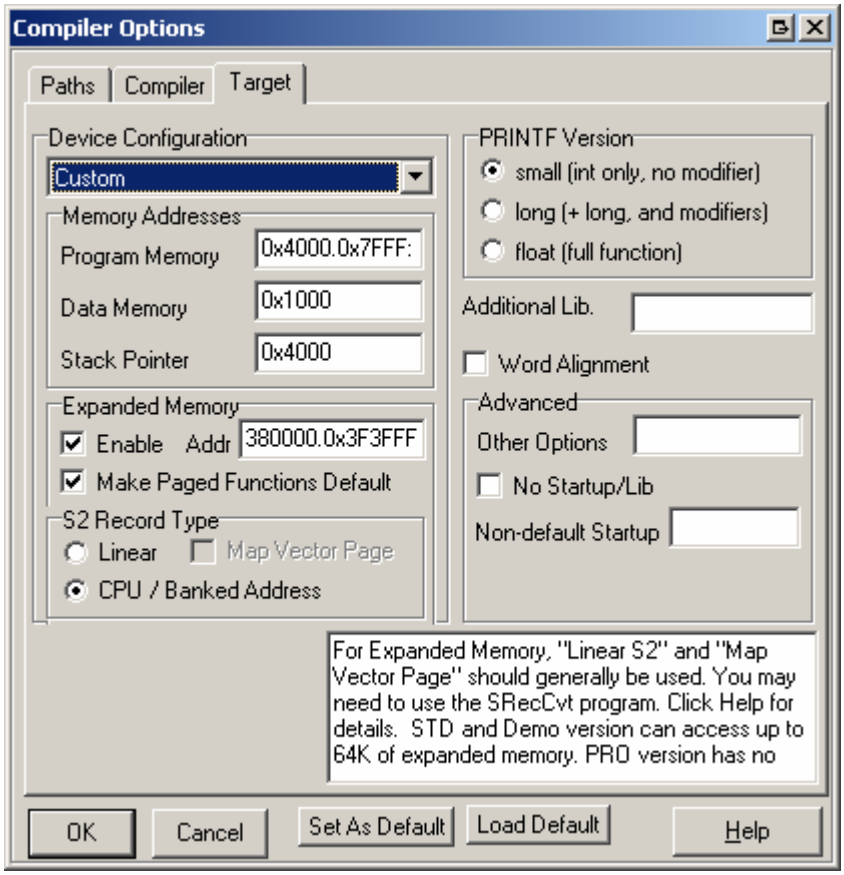
Click on Project Menu – Options – Target Tab.



Please note the Device Configuration. Click on the pull down arrow to change the device type.



Scroll up or down to select Custom as shown.



Device Configuration:
Custom

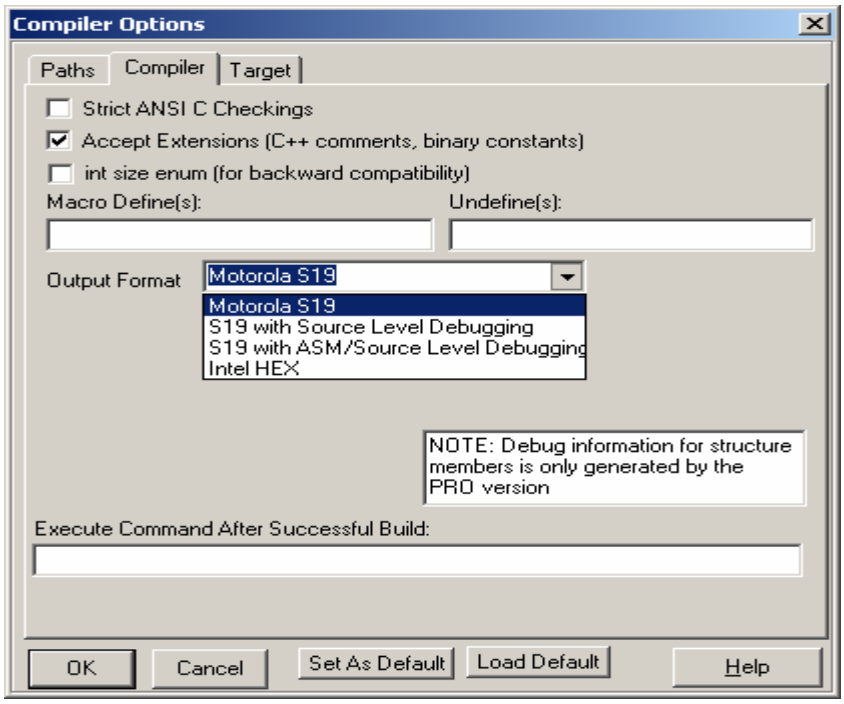
Memory Addresses:
Program Memory = ***0x4000.0x7FFF:0xC000.0xFFFF***
Data Memory = ***0x1000***
Stack Pointer = ***0x4000***

Expanded Memory:
Enable Addr = ***0x380000.0x3F3FFF***.
Make Paged Function Default

Note the address range is ***0x380000.0x3F3FFF***. It defines the available PPAGE to be from $E0 to $FC. PPAGE $FD and $FF are fixed memory areas and are allocated to ***0x4000.0x7FFF:0xC000.0xFFFF*** while PPAGE = $FE is not defined here. Version 7 of the compiler will properly accommodate the XGATE memory map when it is released.

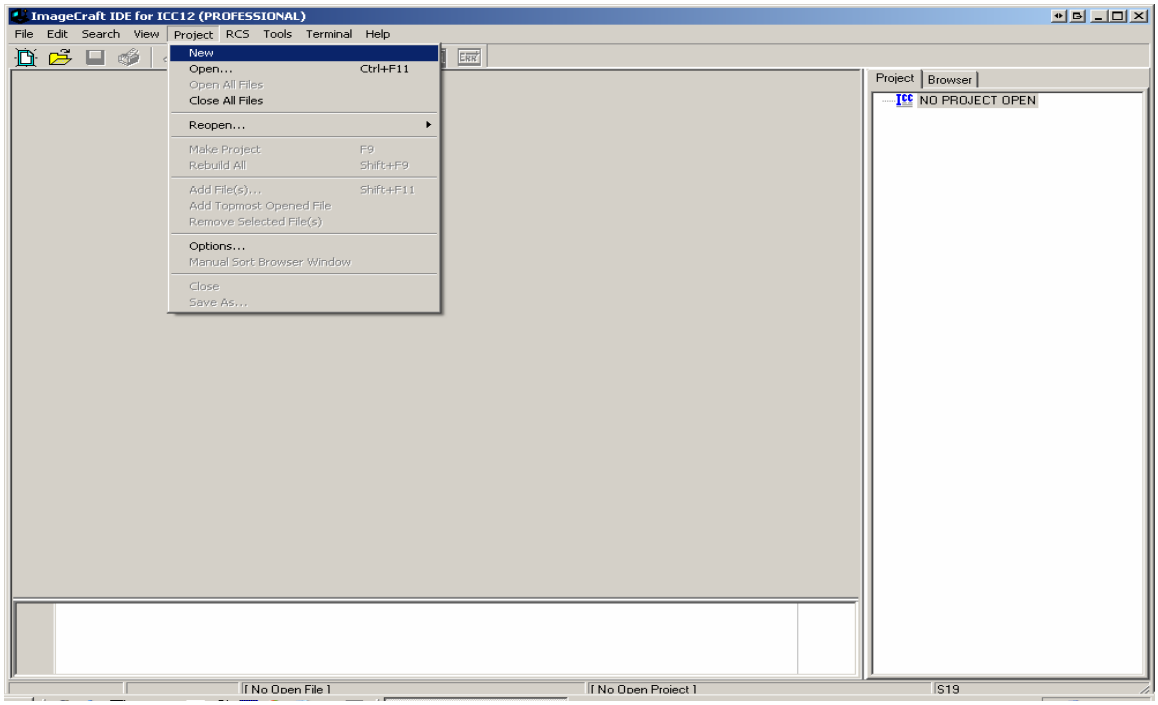
S2 Record Type:
CPU/Banked Address

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

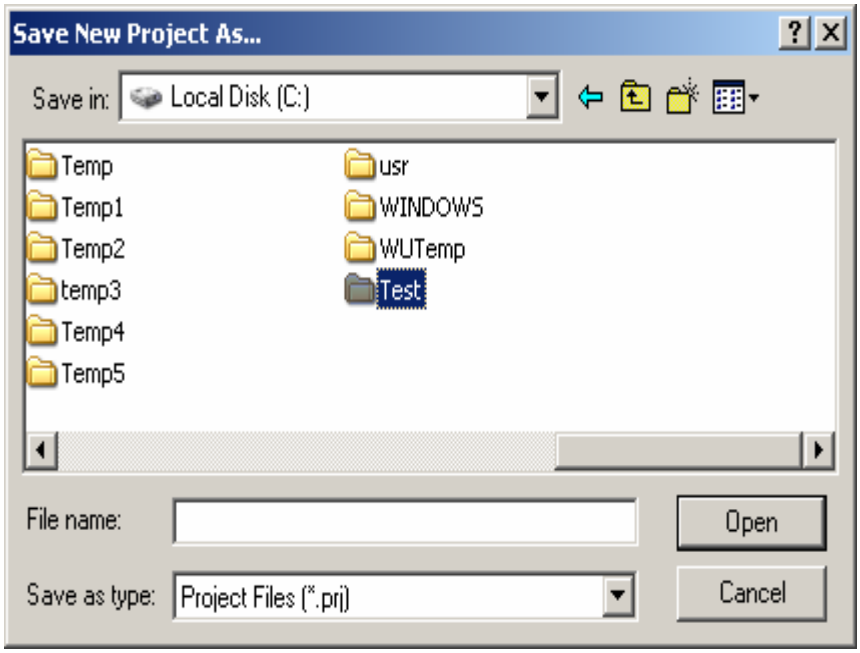


Starting a new Project:

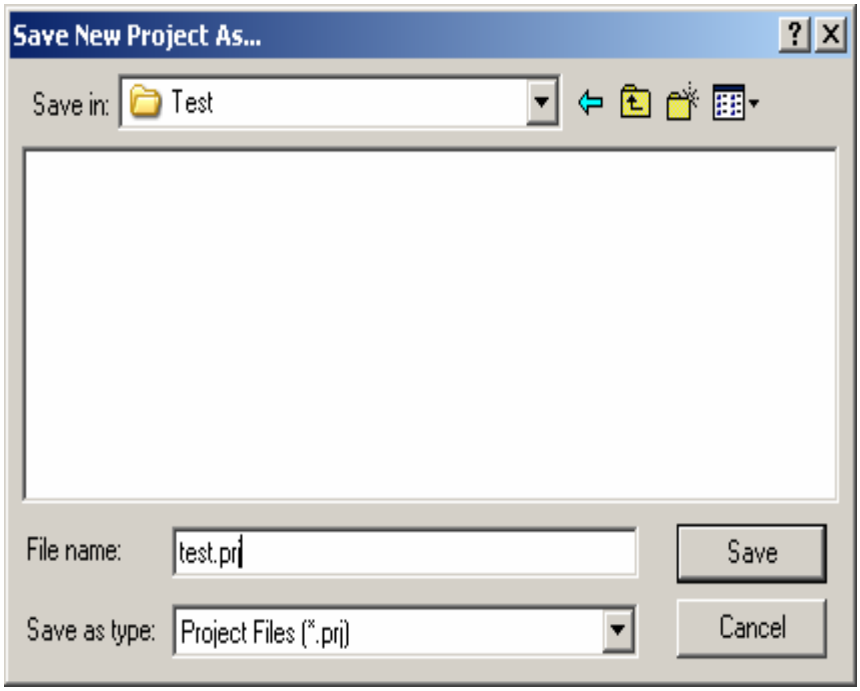
Once the compiler options are setup, a new project can be created. Click Project menu – New.



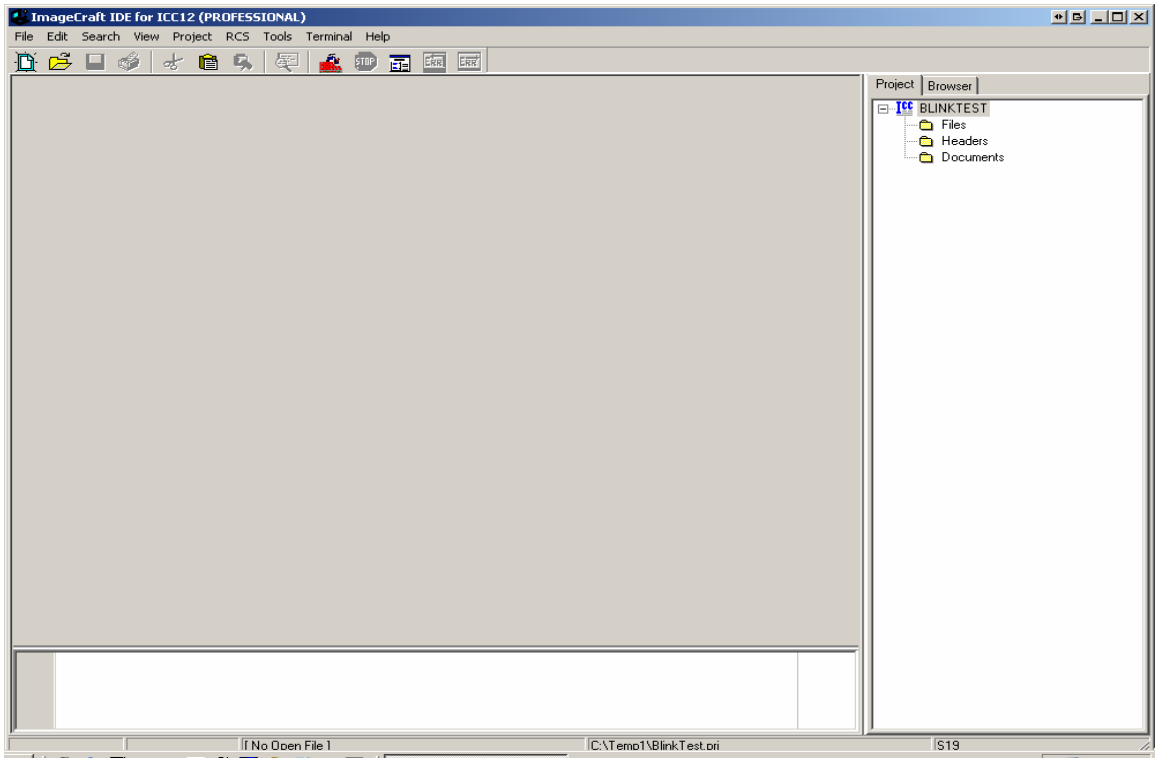
The IDE will prompt to save the new project. You may decide to create a new directory to save the new project. In this example a new directory called ***Test*** is created and the file is saved as file ***test.prj***.



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

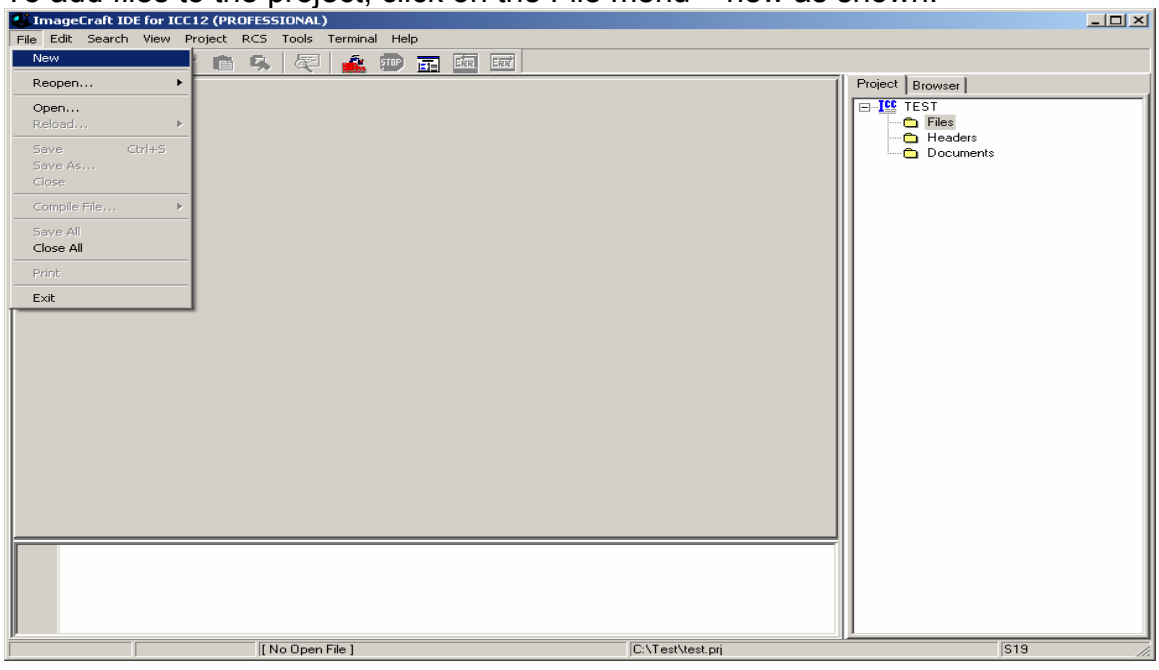


Note that the Project window has changed to add Files, Headers and Documents.

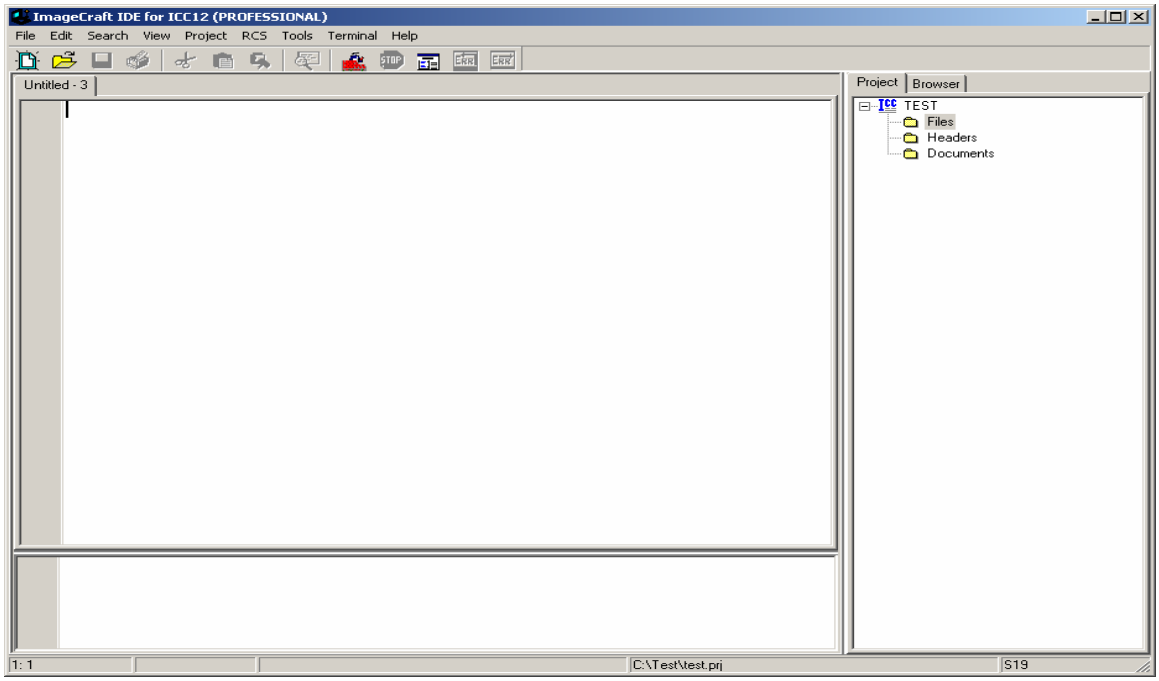


Creating a new file to the project:

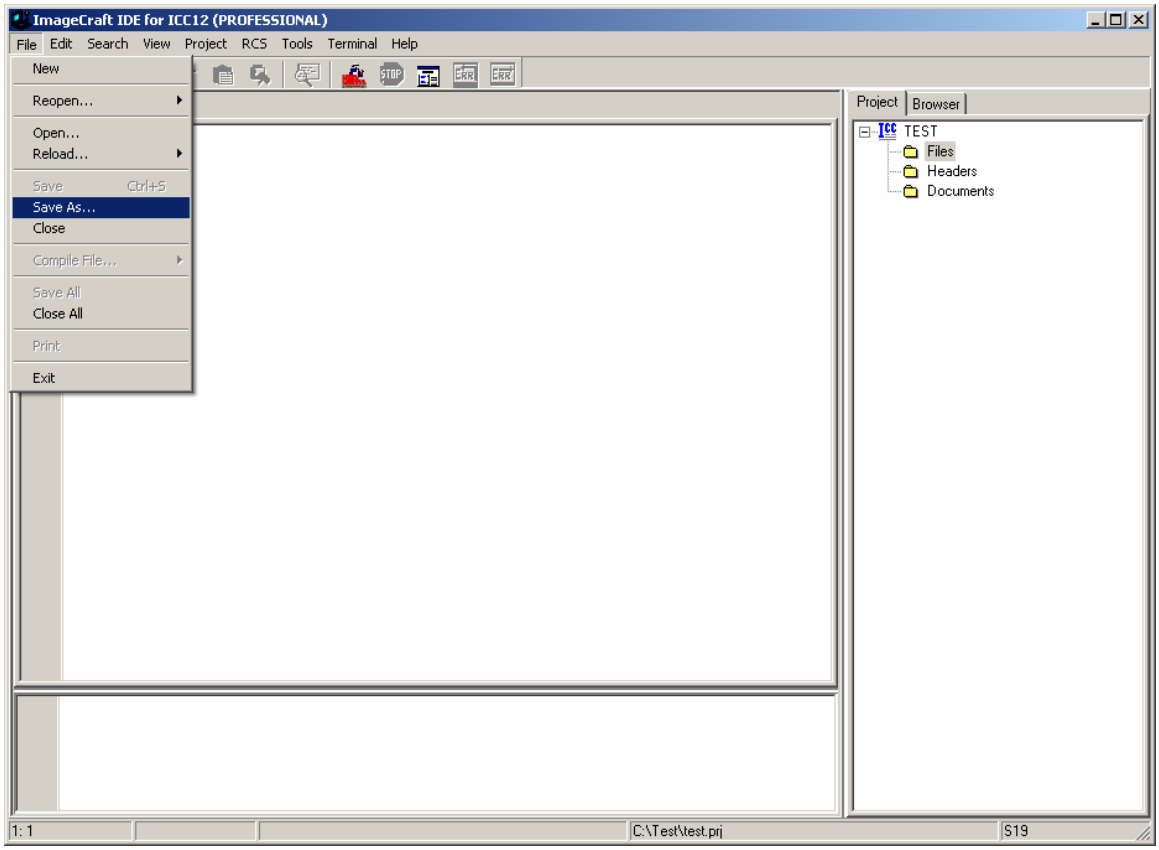
To add files to the project, click on the File menu – new as shown.



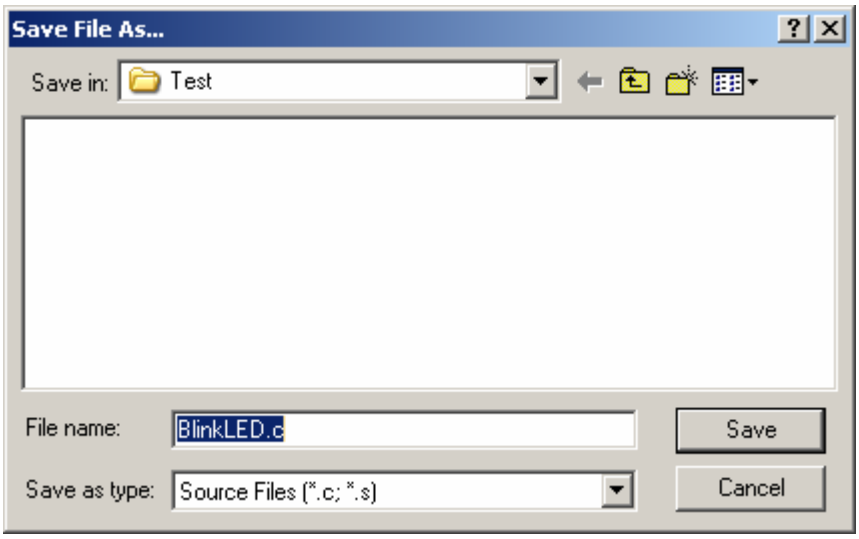
Note that ICC12 created an untitled file



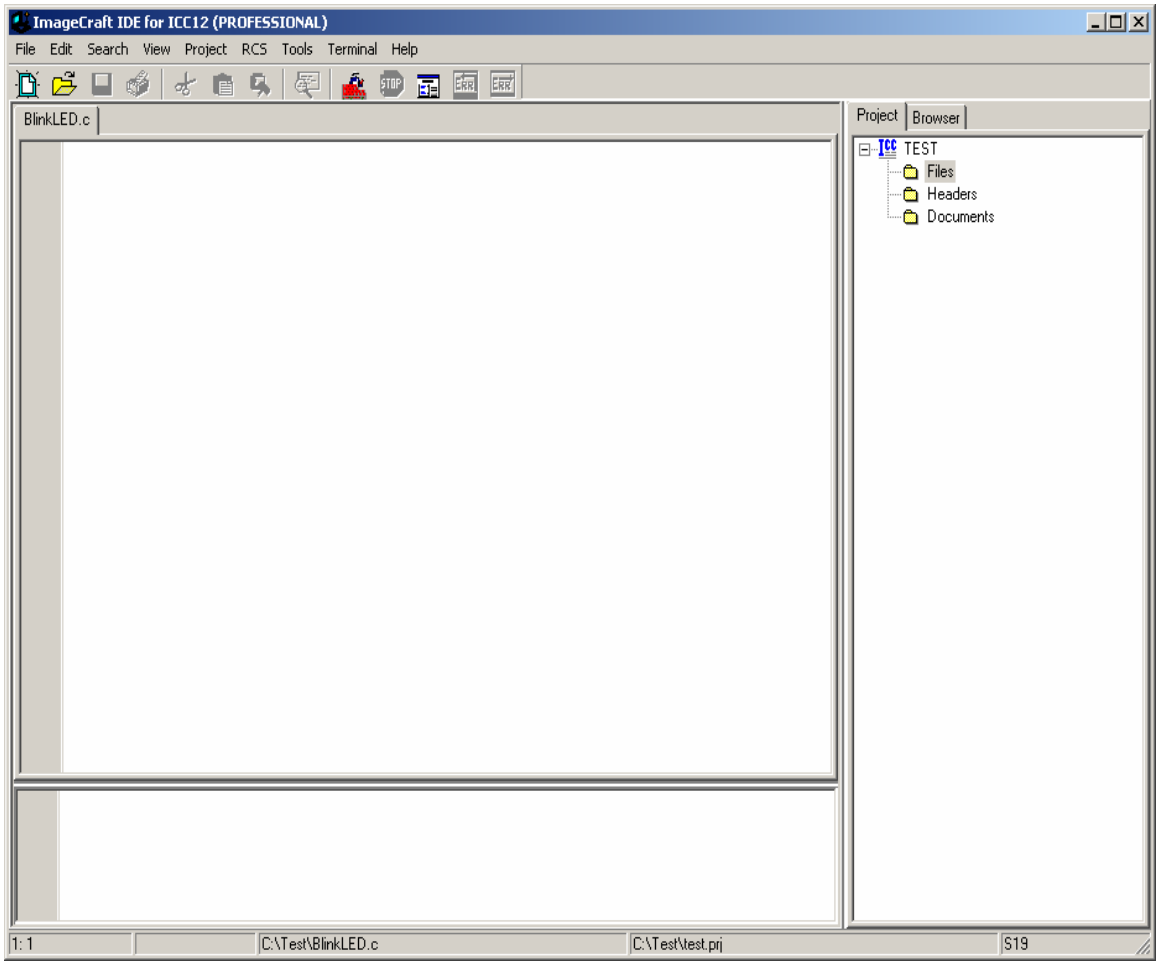
Save the file as ***BlinkLED.C***.



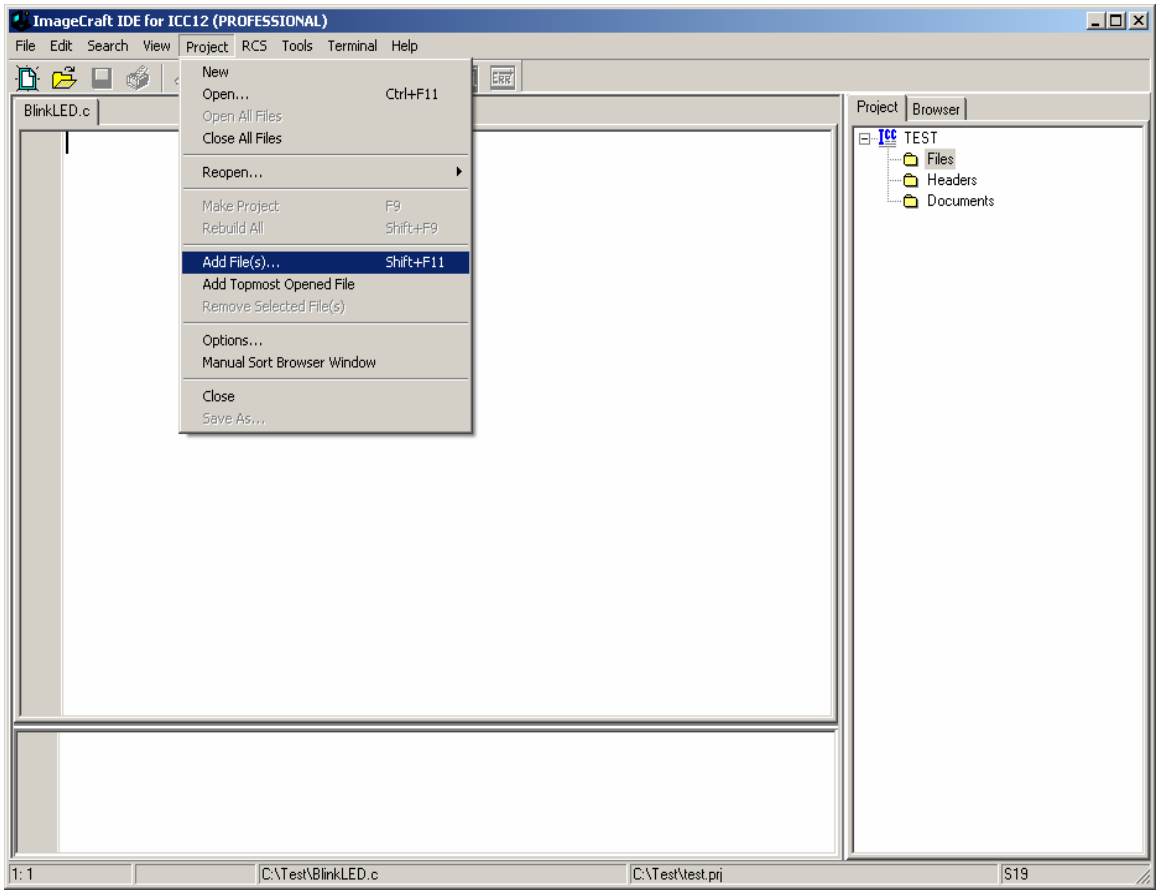
ICC12 will open an explorer window to help save the file. Type BlinkLED.c for the filename and then click the save button.



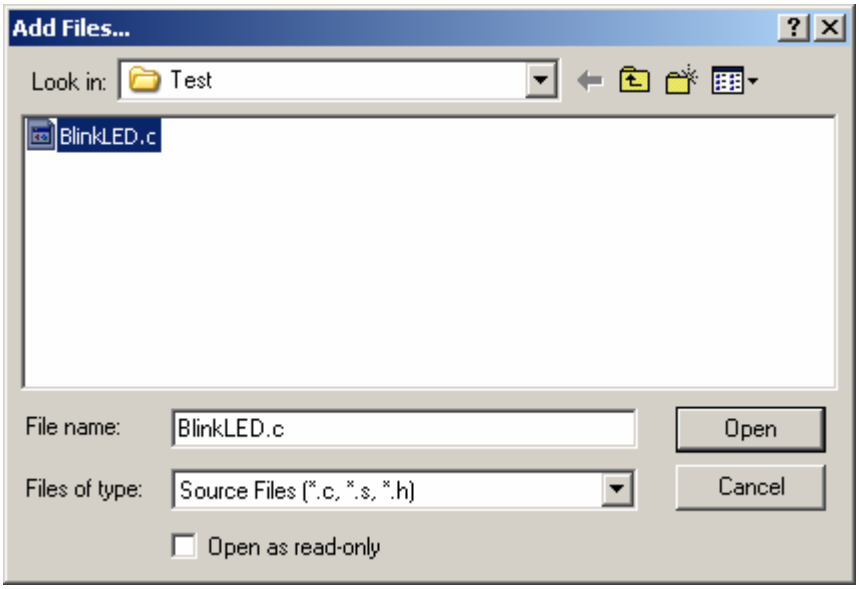
Note that ICC12 has renamed the file to BlinkLED.c.



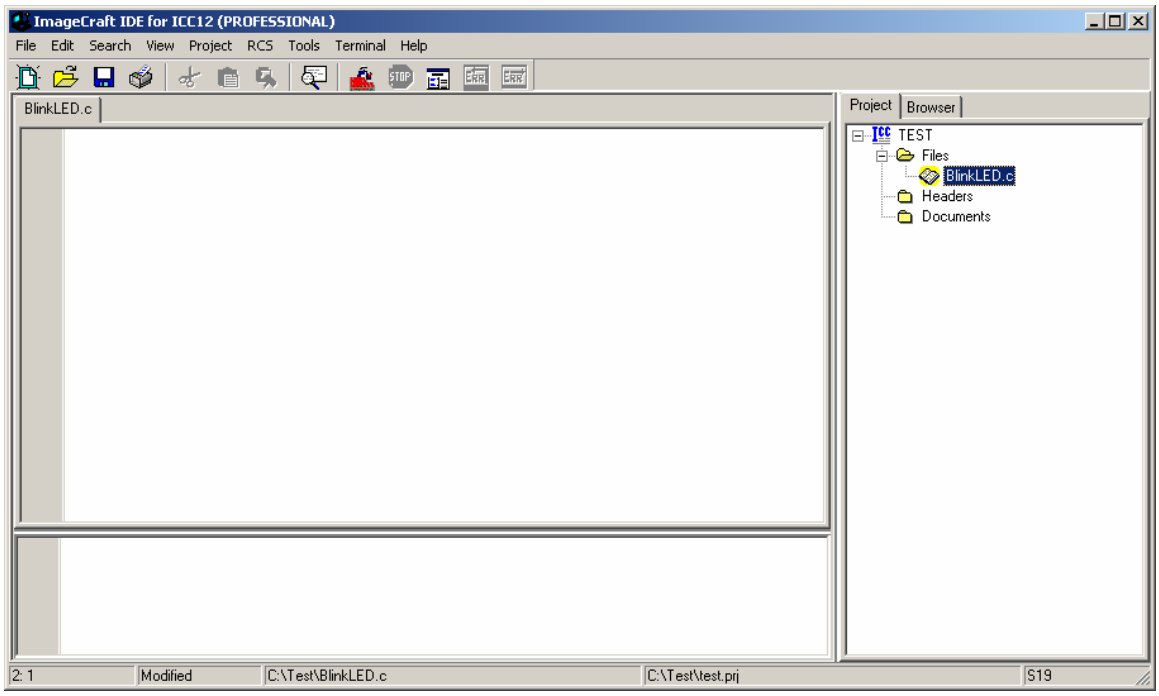
To add BlinkLED.c to the Project, click on the Project menu – Add File(s)



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

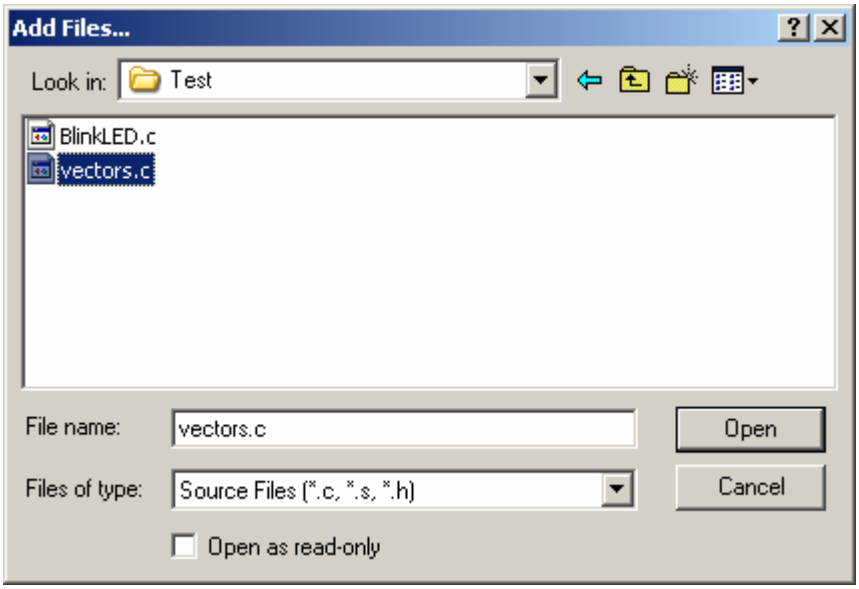


Note that the right window pane has changed to include BlinkLED.c under the Files Project.

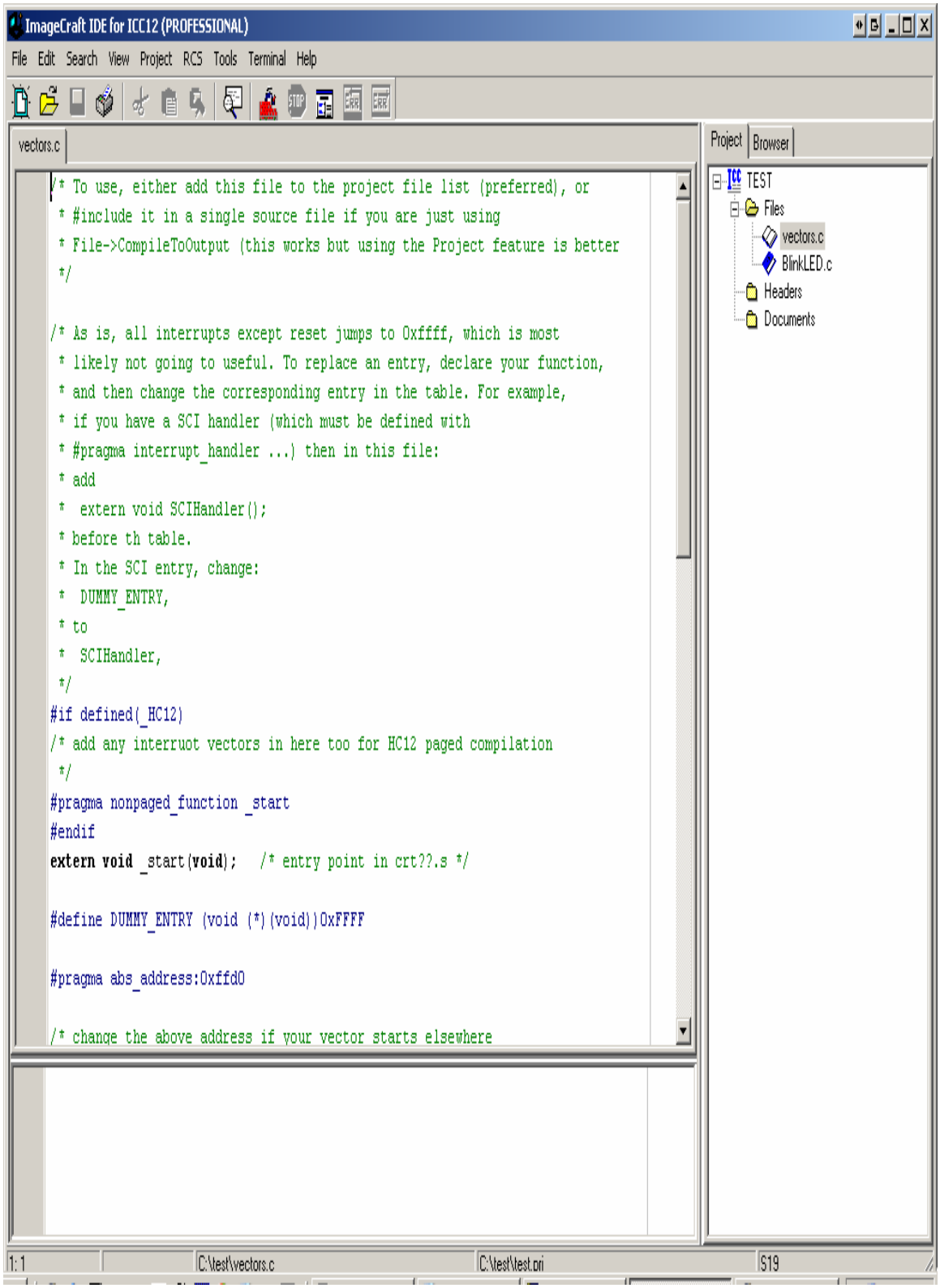


Locate ***vectors.c*** and copy it to the Test directory. This is important to do because of project dependency. It is not a good idea to keep editing a single global ***vectors.c*** file for each new project since other projects are using this file. It becomes a problem to keep track of the changes made to the different projects.

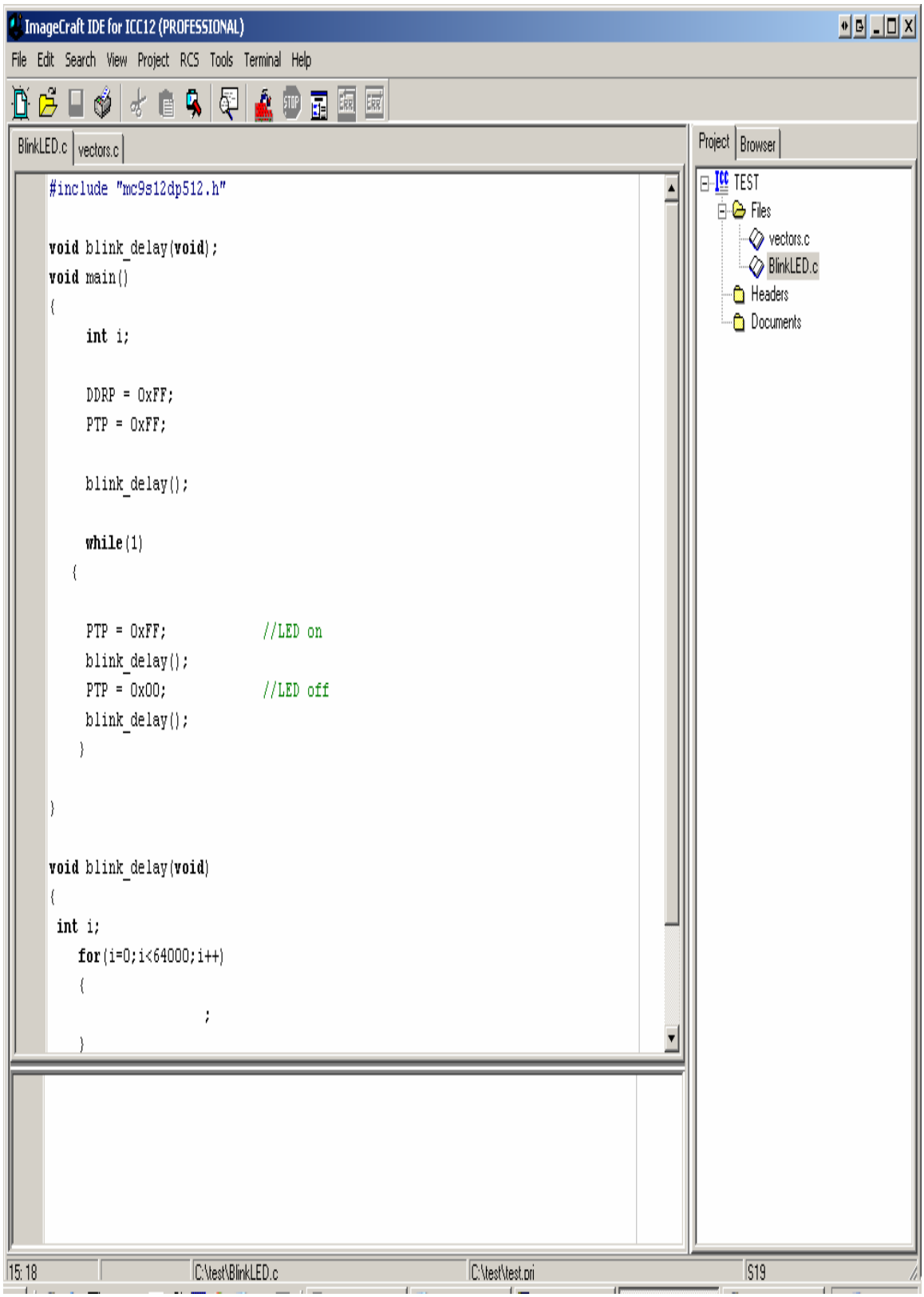
To add ***vectors.c*** to the Project, click on the Project menu – Add File(s)



Note that ICC12 has changed to include ***vectors.c*** It is important to note that the original **vectors.c** supplied by ImageCraft was written for the 68HC912B32 and 68HC812A4 MCUs. Therefore, it needs to be edited to include other ISR addresses for the 9S12XDP512.



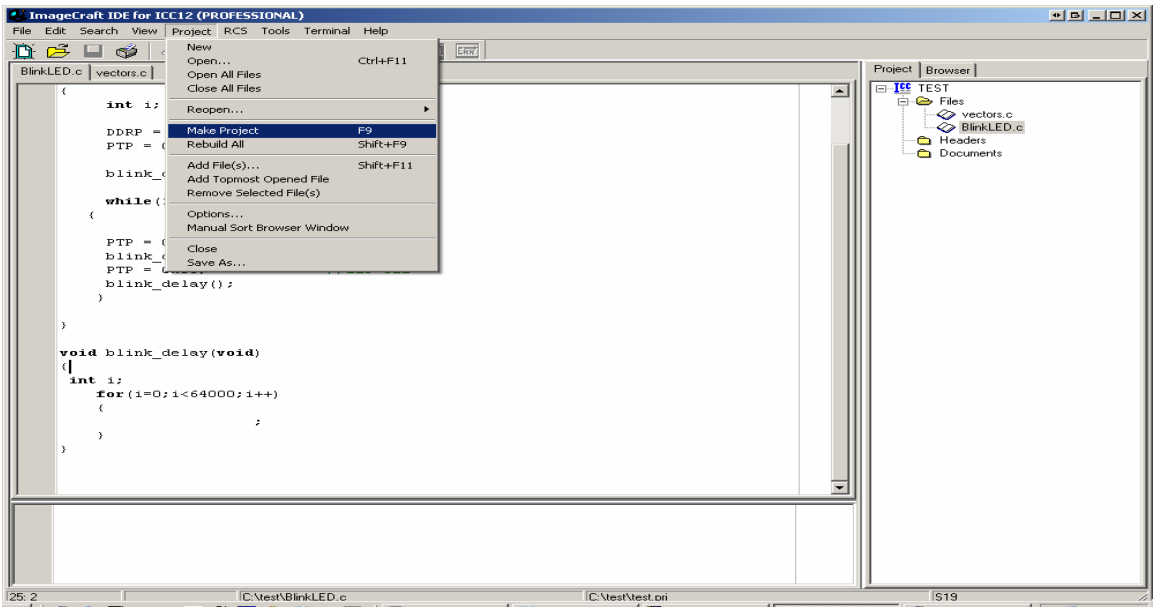
Type the lines of code below into BlinkLED.c file. Once that is done, we can compile/make/build the code.



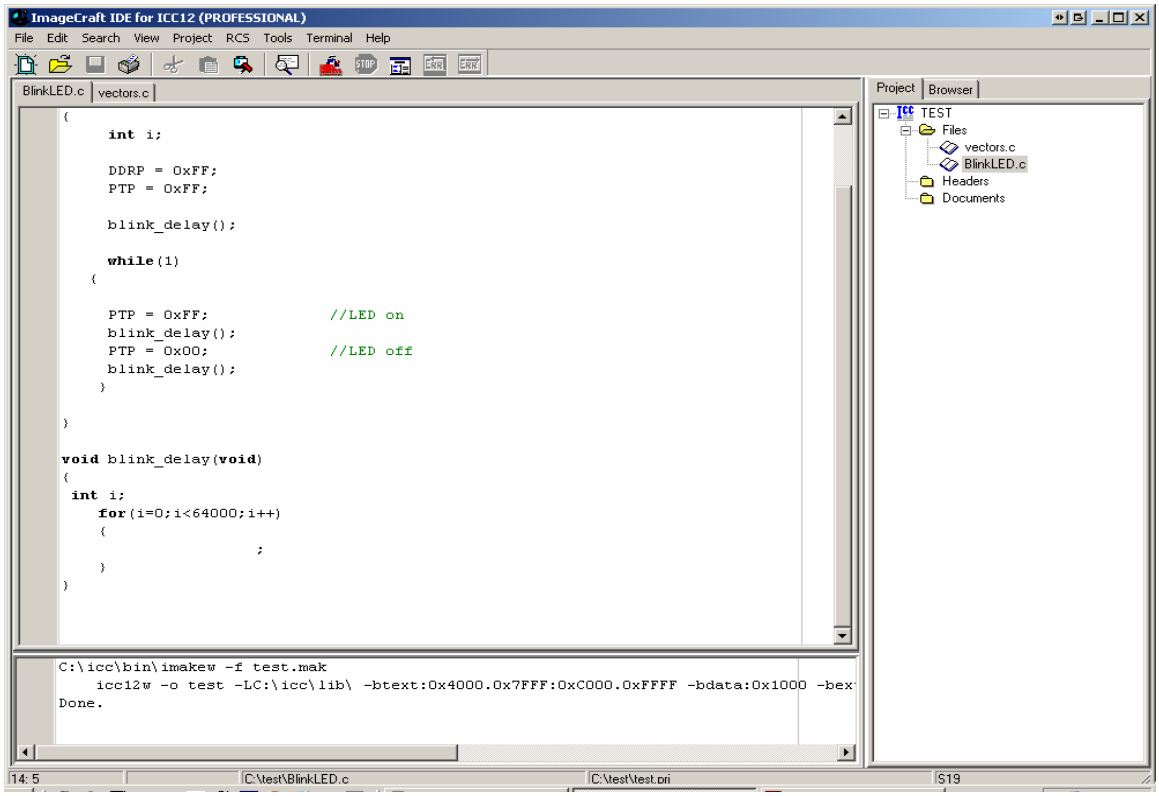
```
#include "mc9s12dp512.h"

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

        DDRP = 0xFF;
        PTP = 0xFF;

        blink_delay();

        while(1)
    {

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

}

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

Compiling/Build/Make the file:

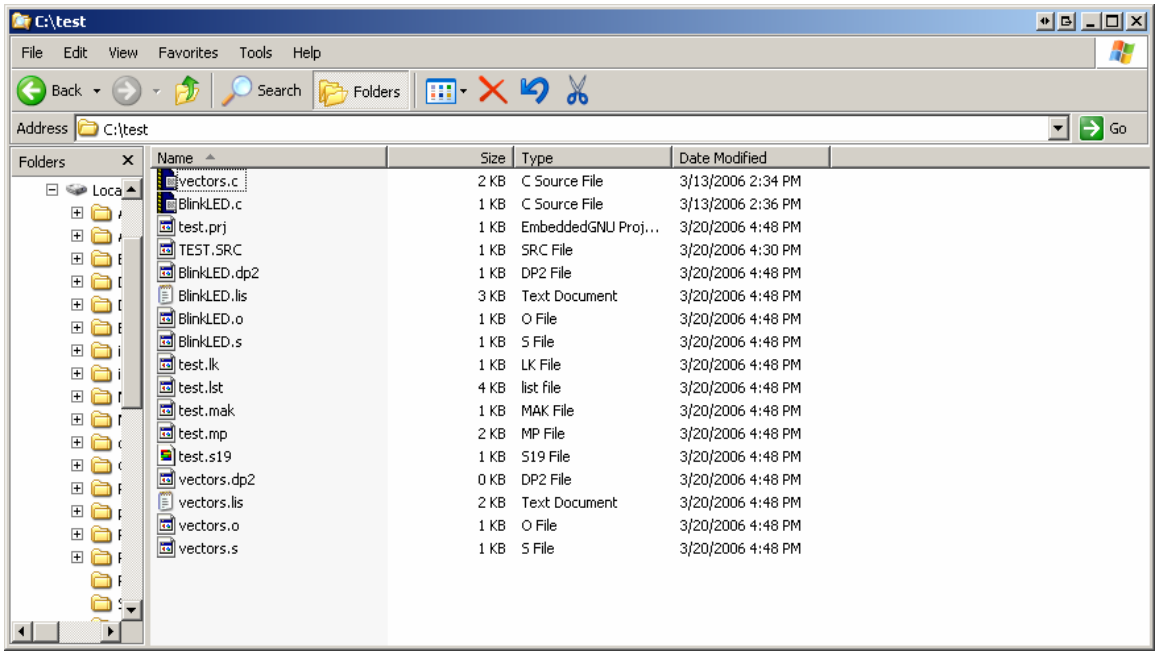
To make the file click Project menu – make project as shown.



Note the lower window pane will show messages to display how the build progressed. Any errors will be displayed in this window. The build was without error so we can progress to erasing and programming the 9S12XDP512.



Locate the c:\Test directory. Note the other extraneous files are created after a make.



Using WordPad to check the content of ***test.s19*** file. Note that the S-records are of different lengths.

S10E4000CF400016405587CE10008E04
S110400B100027056A000820F6CE405ACDAB
S111401810008E405A2706180A307020F51644
S1074026402A20FE0A
S111FFD0FFFFFFFFFFFFFFFFFFFFFFFFFFFF2D
S111FFDEFFFFFFFFFFFFFFFFFFFFFFFFFFFF1F
S111FFECFFFFFFFFFFFFFFFFFFFFFFFFFFFF11
S109FFFAFFFFFFFF4000C1
S110402A34B7751B9EC6FF7B025AC6FF7B90
S110403702584A8000E02010C6FF7B0258AA
S10D40444A8000E07902584A800027
S10A404EE020EEB757303DFE
S2121E0800034B7751B9ECC00006C1E2007EC1EED
S2121E0800EC300016C1EEC1E8CFA0025F2B7577C
S2061E0801C300A43
S10840551D0016073DEB
S9034000BC

Examining S-record:

If one looks closely at the S-record one can see a mixture of S1 and S2 lines. This is a typical S-record file generated by ICC12. S1 records are programmed in the ***$4000 - $7FFF*** and ***$C000 - $FFFF*** memory blocks. Any ISR routines are always in the fixed memory area. The ISR can call any routine inside a PPAGE when necessary. The S2 records can also be in fixed memory but are typically placed in paged memory by ICC12

Note the content of the memory address at $FFFE:$FFFF is $4000, the RESET vector.

S111FFD0FFFFFFFFFFFFFFFFFFFFFFFFFFFF2D
S111FFDEFFFFFFFFFFFFFFFFFFFFFFFFFFFF1F
S111FFECFFFFFFFFFFFFFFFFFFFFFFFFFFFF11
S109***FFFA***FFFFFFFF***4000***C1

The S-record below is the start of code. The content of address beginning at $4000 to $4050

S10E***4000***CF400016405587CE10008E04
S110400B100027056A000820F6CE405ACDAB
S111401810008E405A2706180A307020F51644
S1074026402A20FE0A

Please note the PPAGE = E0 the first available PPAGE as defined by address range ***0x380000.0x3F3FFF***.

S2121***E0***800034B7751B9ECC00006C1E2007EC1EED
S2121***E0***800EC300016C1EEC1E8CFA0025F2B7577C
S2061***E0***801C300A43

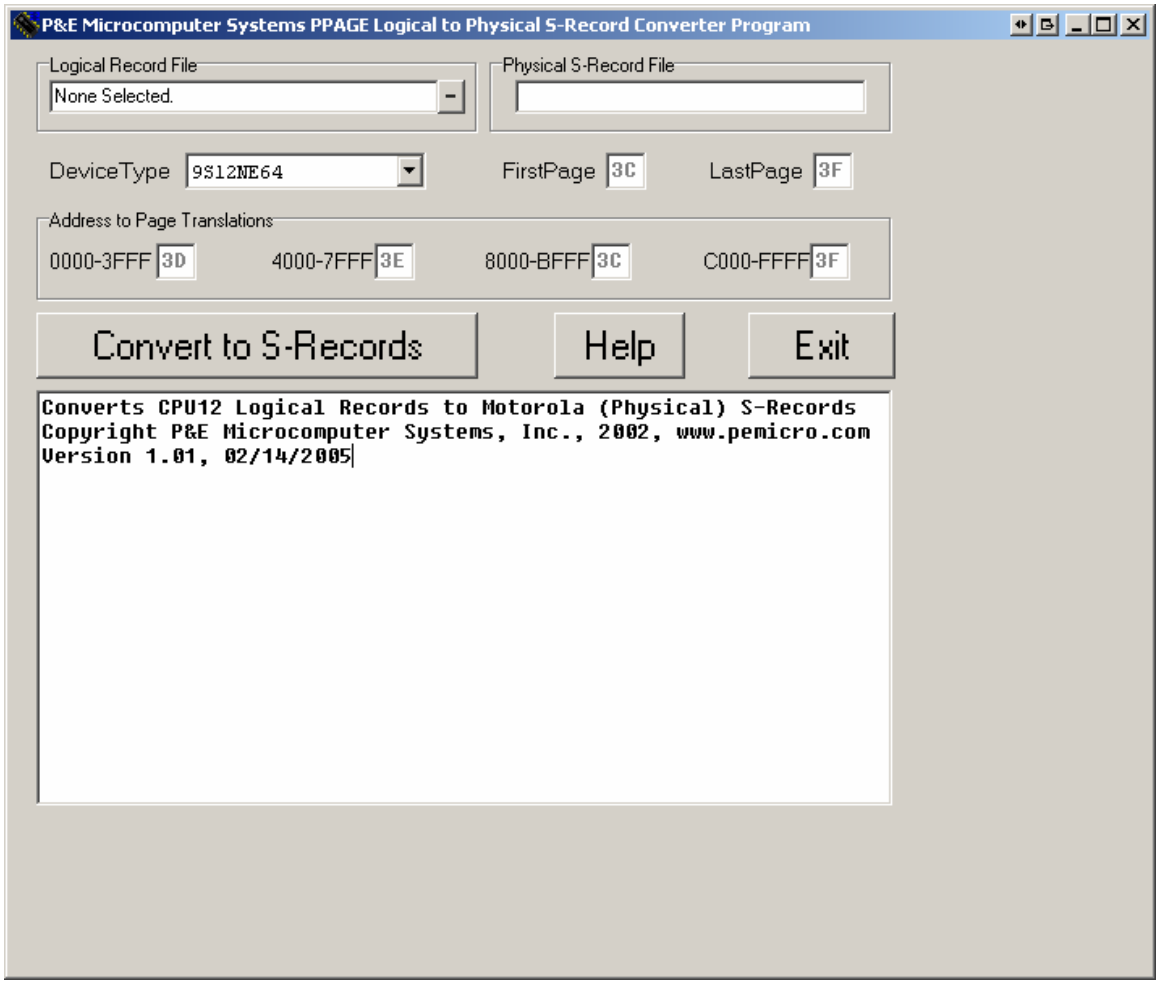
Using P&E LogPhy program:

The LogPhy is used to reformat the ICC12 generated S-record.

Download the LogPhy program from the P&E website and install it on the PC. While the utility is free, you'll need to register to download program

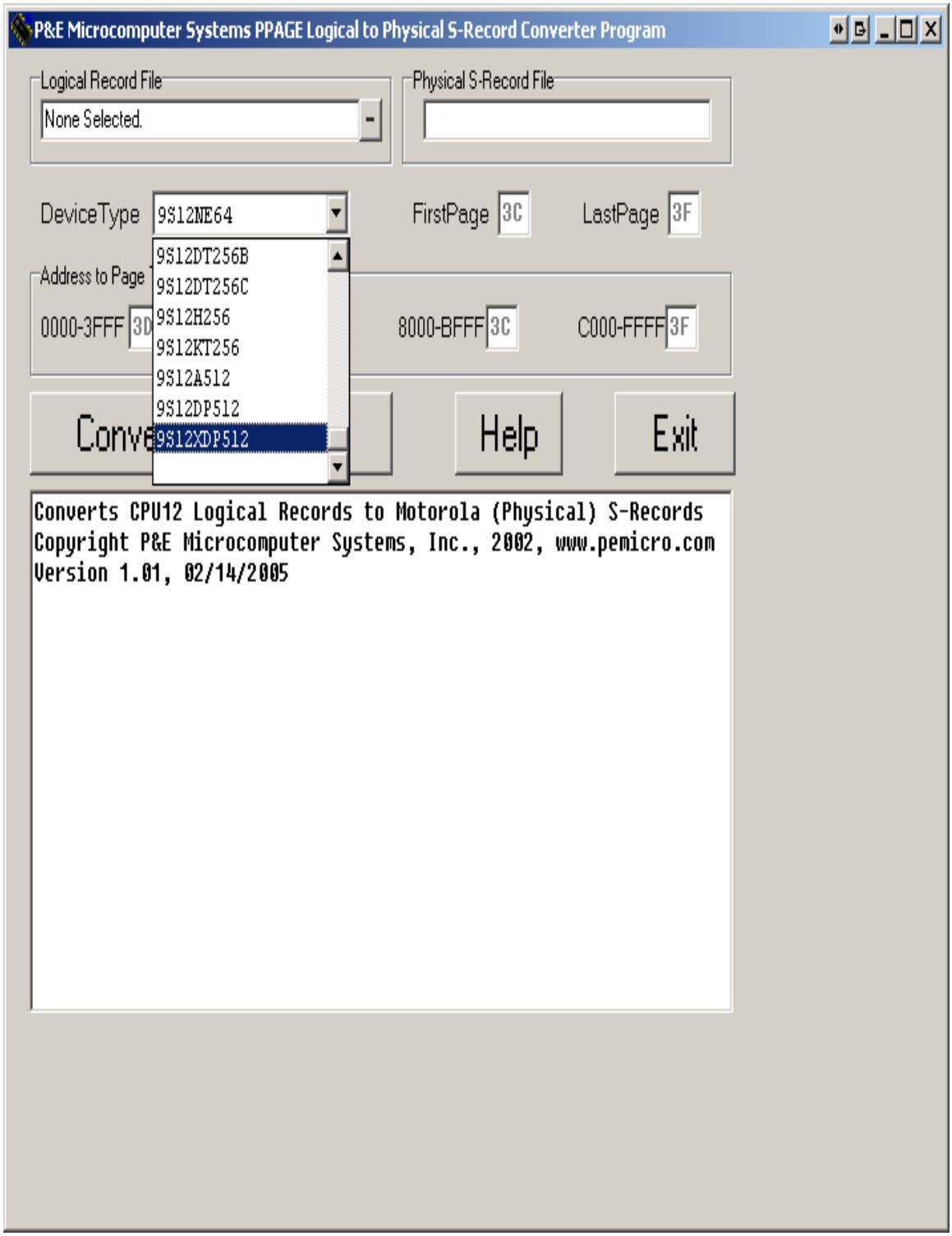
http://www.pemicro.com/
http://www.pemicro.com/support/download_processor.cfm?family=4

Run LogPhy as shown.



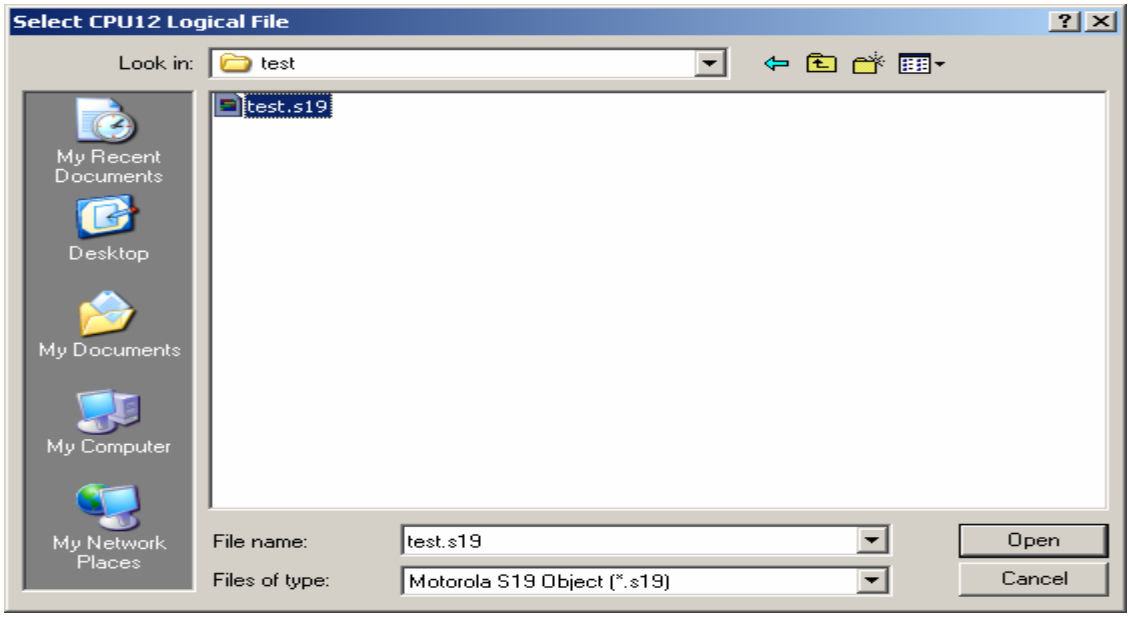
Selecting Device:

Click on the Device Type dropdown list and select 9S12XDP512 as shown.

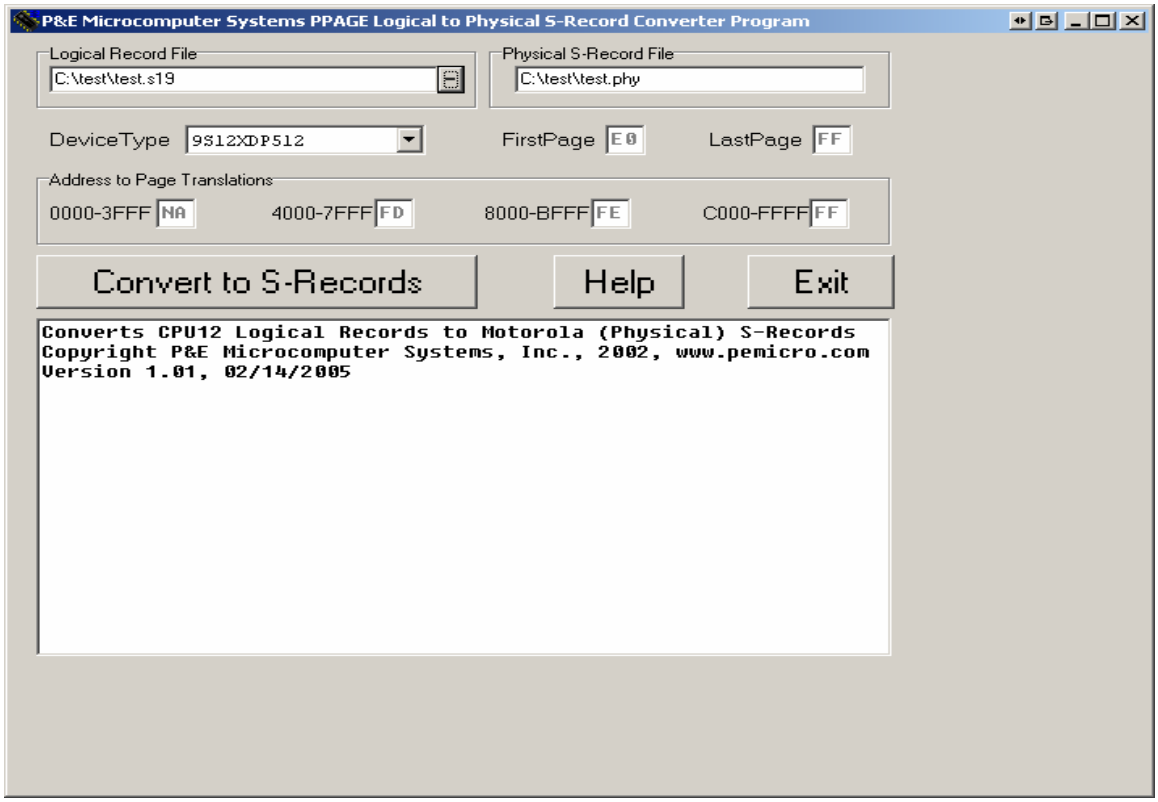


Logical Record File:

Click on the Logical Record File to help locate the ICC12 generated ***test.s19*** s-record.

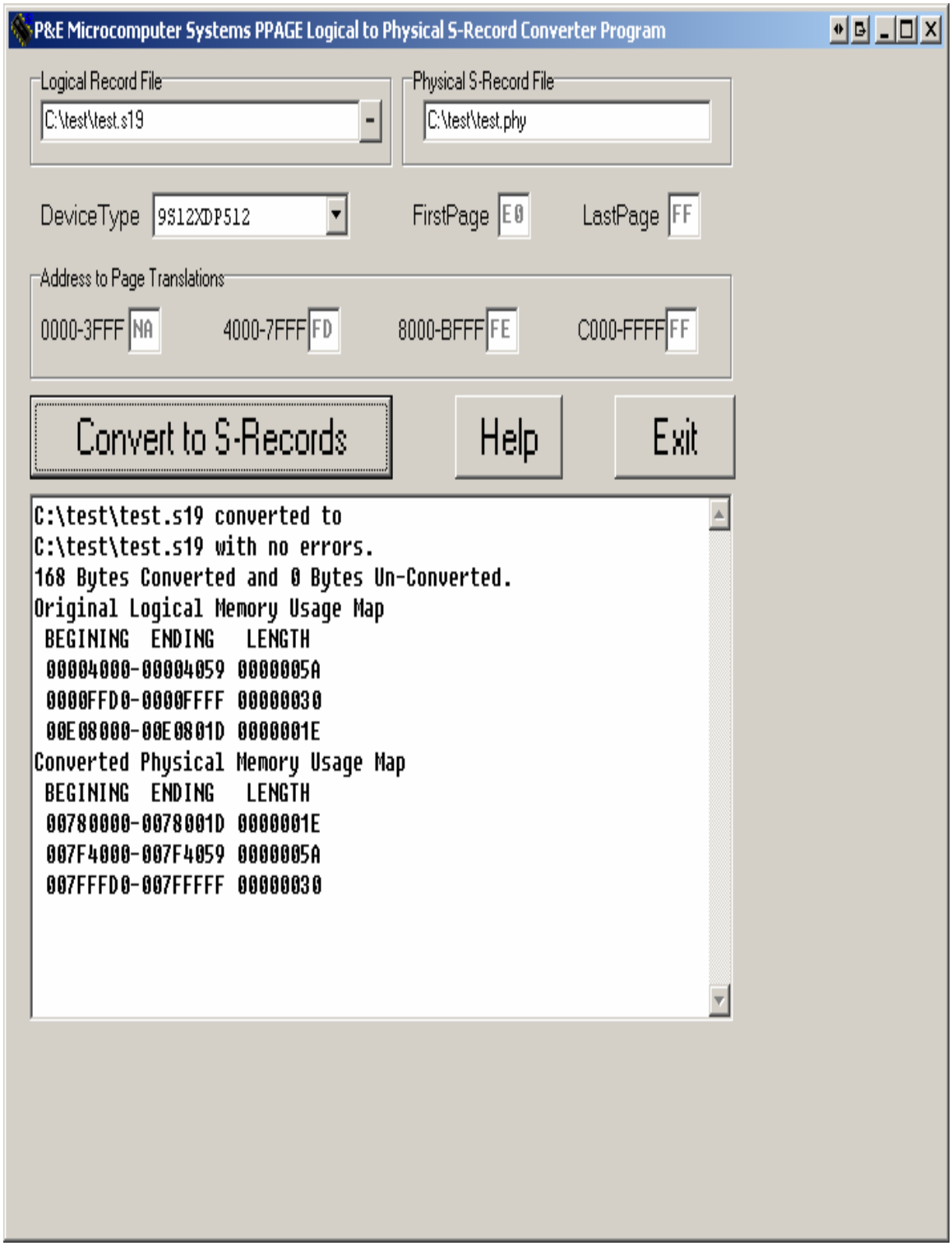


Click the open button. Note the formatted file will be called ***test.phy.***



Converting:

Press the Convert to S-Record.



Note the various messages generated by LogPhy. Open the ***test.prj*** to examine the resulting file.

S2147F4000CF400016405587CE10008E100027056AD9
S2147F4010000820F6CE405ACD10008E405A2706184C
S20E7F40200A307020F516402A20FEB5
S2147FFFD0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAD
S2147FFFE0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9D
S2147FFFF0FFFFFFFFFFFFFFFFFFFFFFFFFF40004B
S2147F402A34B7751B9EC6FF7B025AC6FF7B02584A69
S2147F403A8000E02010C6FF7B02584A8000E07902A3
S20F7F404A584A8000E020EEB757303D5C
S21478000034B7751B9ECC00006C1E2007EC1EC30010
S212780010016C1EEC1E8CFA0025F2B757300AEB
S2097F40551D0016073D6B

Note the bold s-record.

S2 – S-record type
09 – Byte count
7F - PPAGE
4055 – Address of DATA to be written to FLASH
1D0016073D - DATA
6B – CRC

The address ***4055*** is a misaligned word address. The DATA needs to be started at an even address for use by the Flash programming routine of the BDM pod, so it will need to be manually corrected.

Open the Original S-record ***test.s19*** to fix this problem

S10E4000CF400016405587CE10008E04
S110400B100027056A000820F6CE405ACDAB
S111401810008E405A2706180A307020F51644
S1074026402A20FE0A
S111FFD0FFFFFFFFFFFFFFFFFFFFFFFF2D
S111FFDEFFFFFFFFFFFFFFFFFFFFFFFFFF1F
S111FFECFFFFFFFFFFFFFFFFFFFFFFFFFF11
S109FFFAFFFFFFFF4000C1
S110402A34B7751B9EC6FF7B025AC6FF7B90
S110403702584A8000E02010C6FF7B0258AA
S10D40444A8000E07902584A800027
S10A404EE020EEB757303DFE
S212E0800034B7751B9ECC00006C1E2007EC1EED
S212E0800EC300016C1EEC1E8CFA0025F2B7577C
S206E0801C300A43
S10840551D0016073DEB

Move the Bold S-record around to create contiguous S-record addresses.

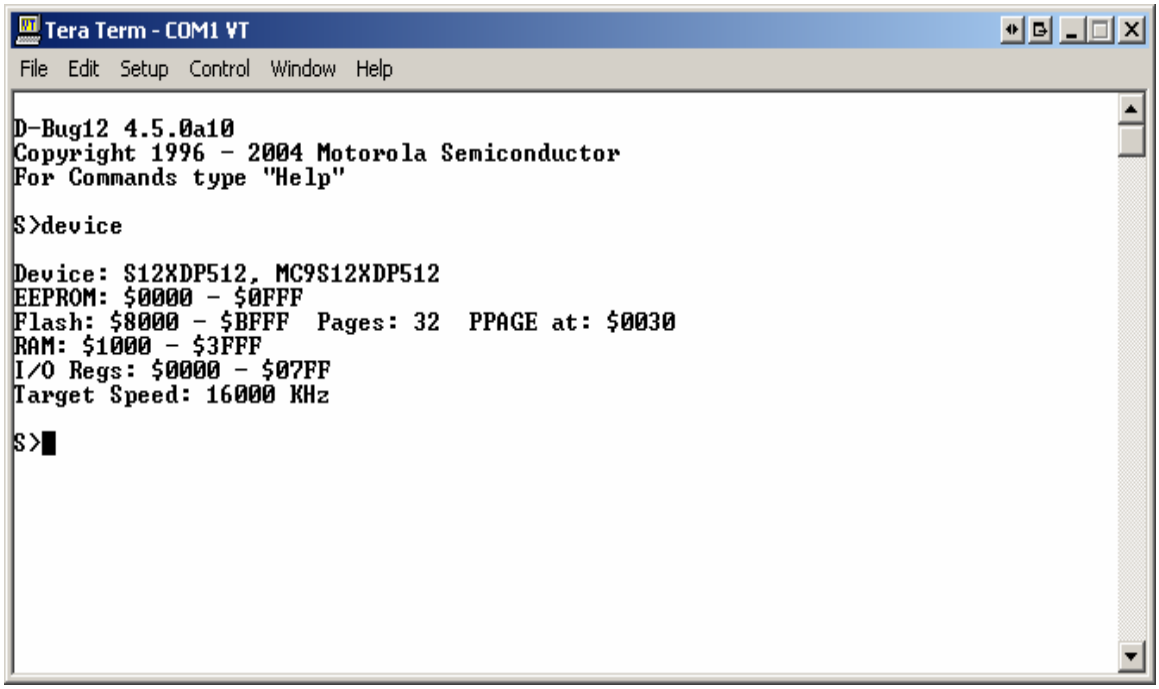
S10E4000CF400016405587CE10008E04
S110400B100027056A000820F6CE405ACDAB
S111401810008E405A2706180A307020F51644
S1074026402A20FE0A
S110402A34B7751B9EC6FF7B025AC6FF7B90
S110403702584A8000E02010C6FF7B0258AA
S10D40444A8000E07902584A800027
S10A404EE020EEB757303DFE
S10840551D0016073DEB
S111FFD0FFFFFFFFFFFFFFFFFFFFFFFFFF2D
S111FFDEFFFFFFFFFFFFFFFFFFFFFFFFFF1F
S111FFECFFFFFFFFFFFFFFFFFFFFFFFFFF11
S109FFFAFFFFFFFF4000C1
S212E0800034B7751B9ECC00006C1E2007EC1EED
S212E0800EC300016C1EEC1E8CFA0025F2B7577C
S206E0801C300A43

Save the edited file then run LogPhy once more. Below is the newly formatted ***test.phy*** file. Note that it no longer has the misaligned address problem.

S2147F4000CF400016405587CE10008E100027056AD9
S2147F4010000820F6CE405ACD10008E405A2706184C
S2147F40200A307020F516402A20FE34B7751B9EC6D0
S2147F4030FF7B025AC6FF7B02584A8000E02010C6EC
S2147F4040FF7B02584A8000E07902584A8000E020D1
S20E7F4050EEB757303D1D0016073D02
S2147FFD0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFAD
S2147FFE0FFFFFFFFFFFFFFFFFFFFFFFFFFFFF9D
S2147FFF0FFFFFFFFFFFFFFFFFFFFFFFFFF40004B
S21478000034B7751B9ECC00006C1E2007EC1EC30010
S212780010016C1EEC1E8CFA0025F2B757300AEB

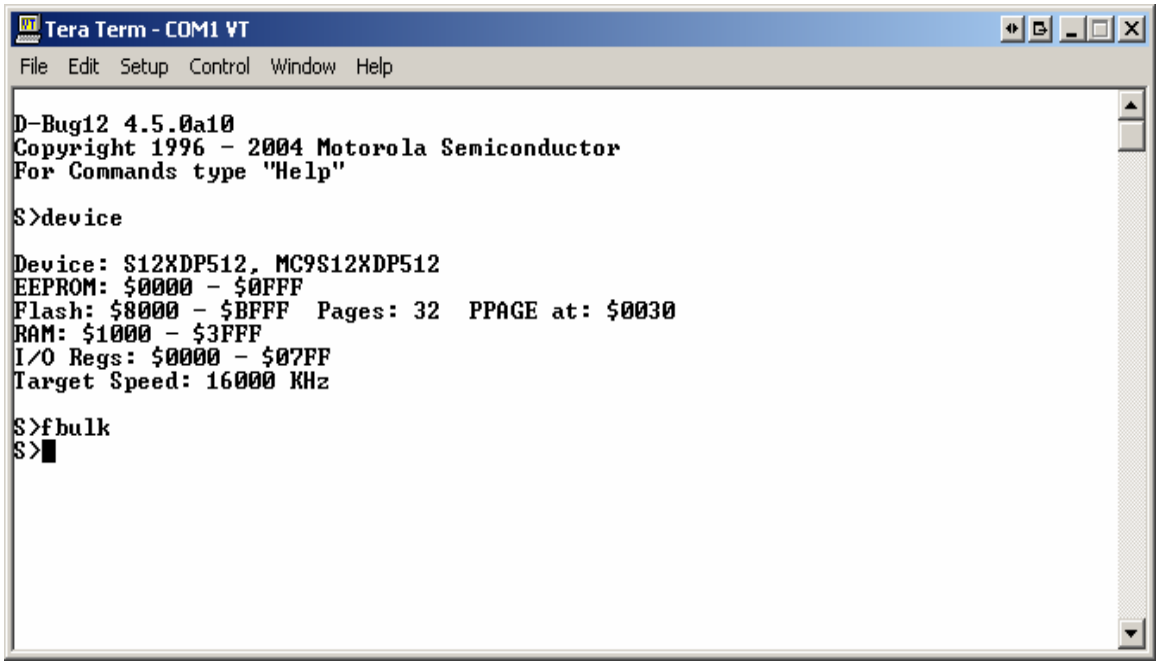
Programming the Adapt9S12XDP512:

Connect the BDM pod to the target and open a terminal program. In this example the terminal program is Tera Term.



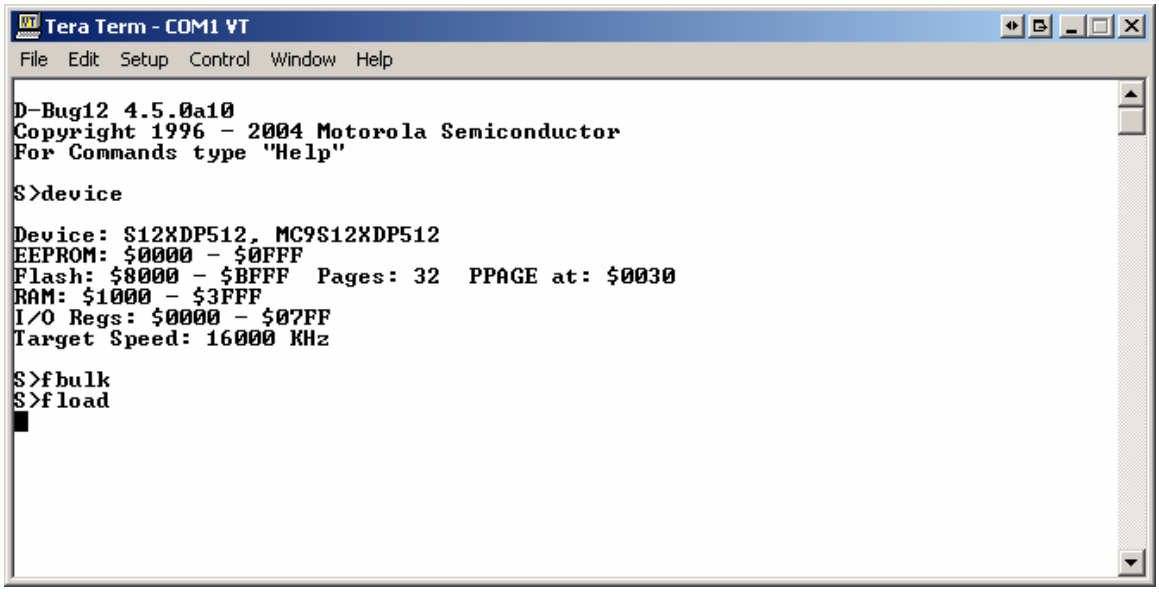
Erasing:

To erase type ***fbulk*** followed by ***Enter***, as shown.

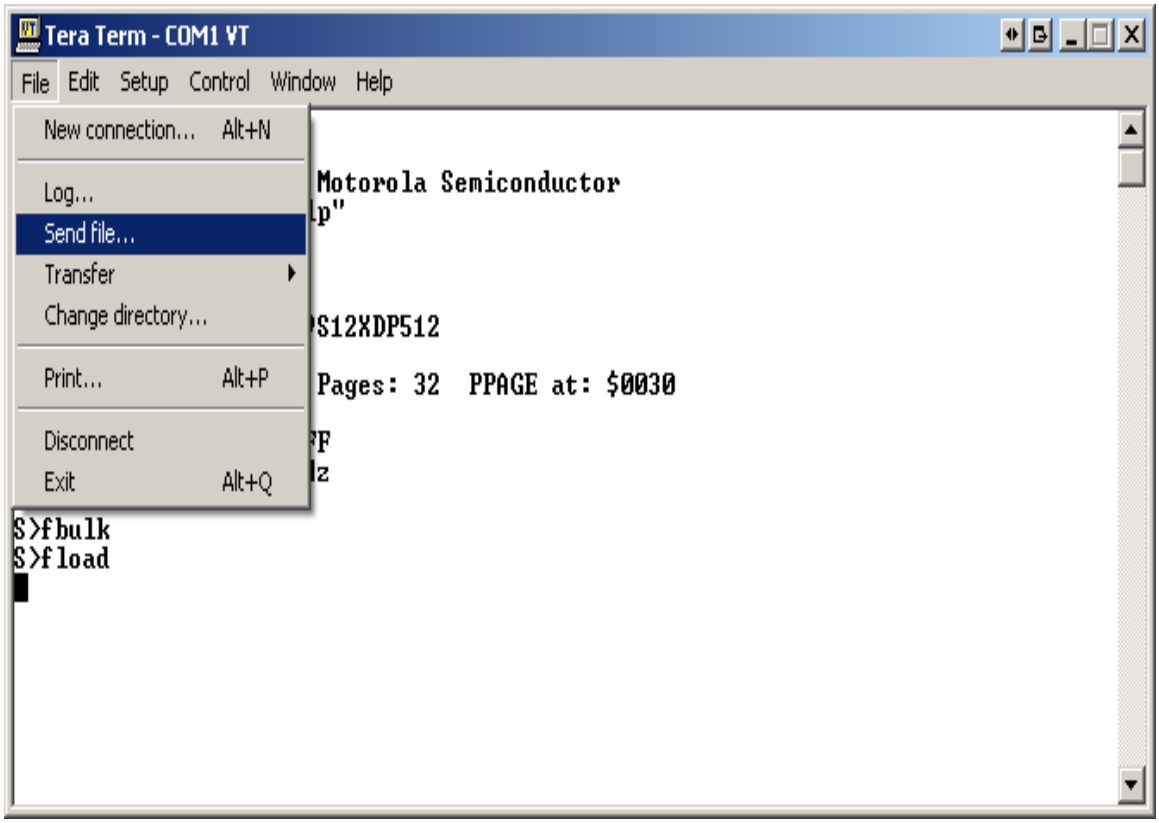


Programming:

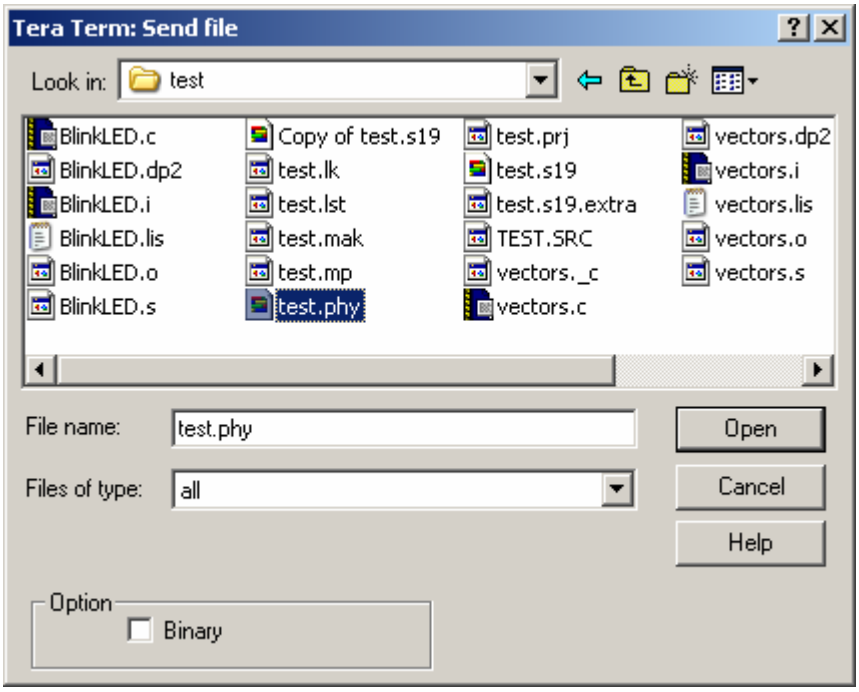
To program, type ***fload*** followed by ***Enter***, as shown.



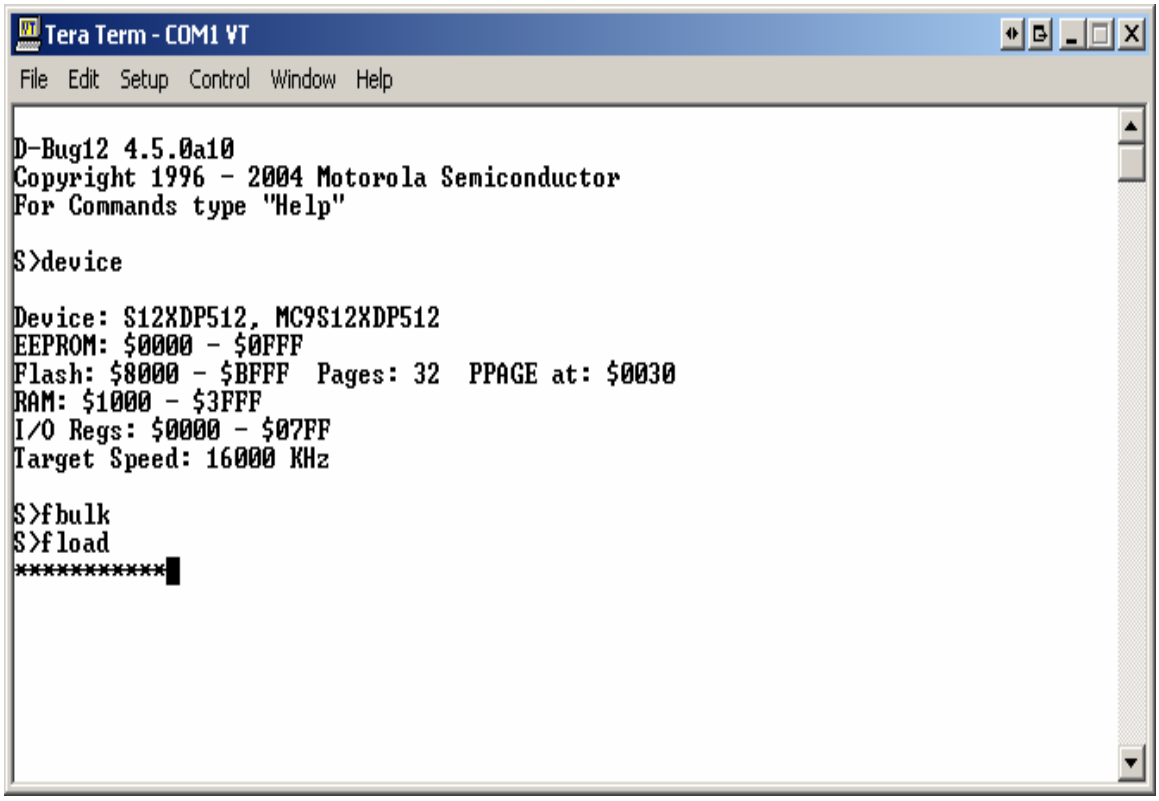
Then locate the file **test.phy** to be sent to the target.



An explorer window will pop up to help, as shown.

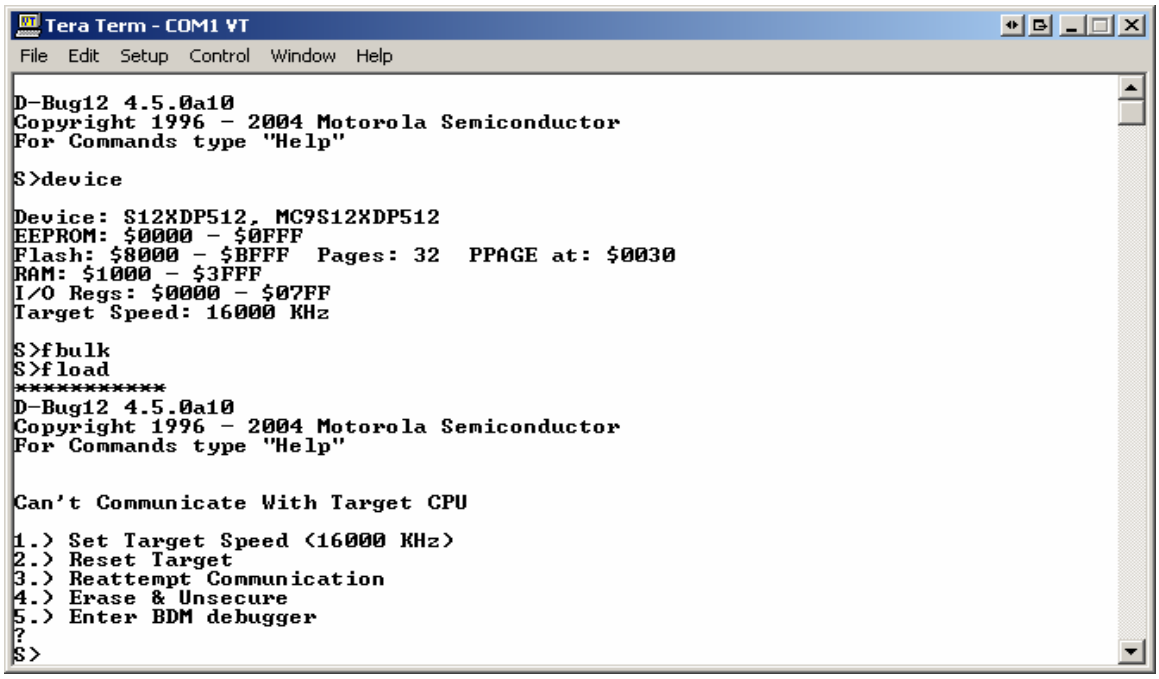


Click ***Open*** to initiate the file transfer.



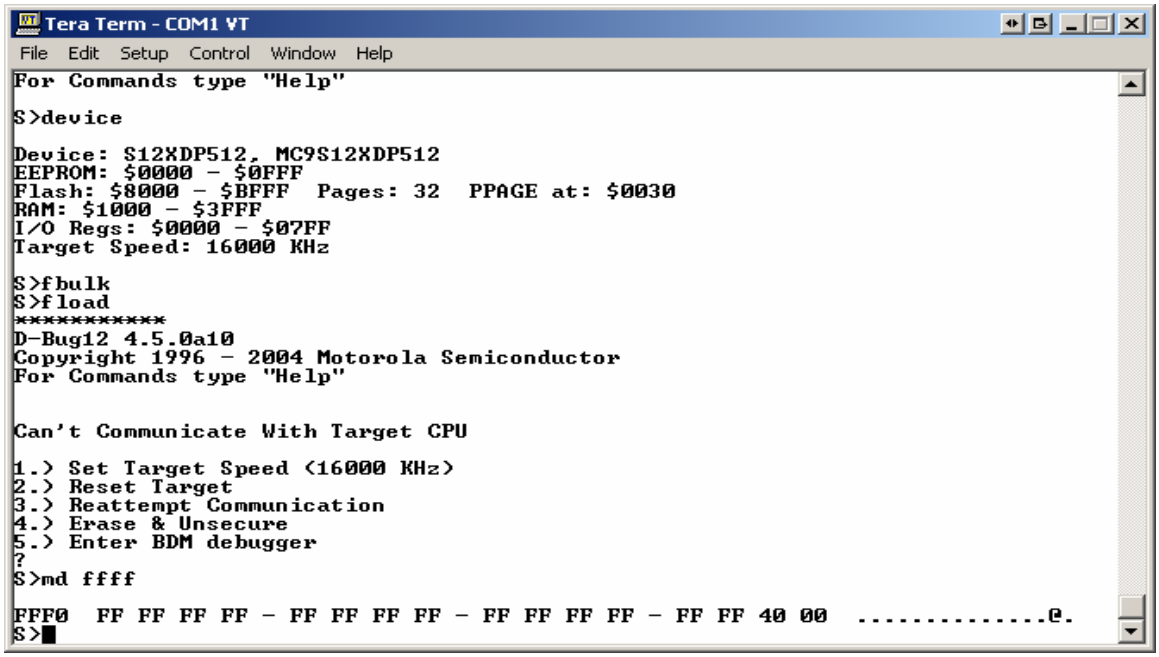
Reset BDM pod:

Note that the BDM pod seems to “hang” and is unresponsive. Press the BDM pod reset button to recover. The reason this happens is due to LogPhy not including an S9 termination S-record.



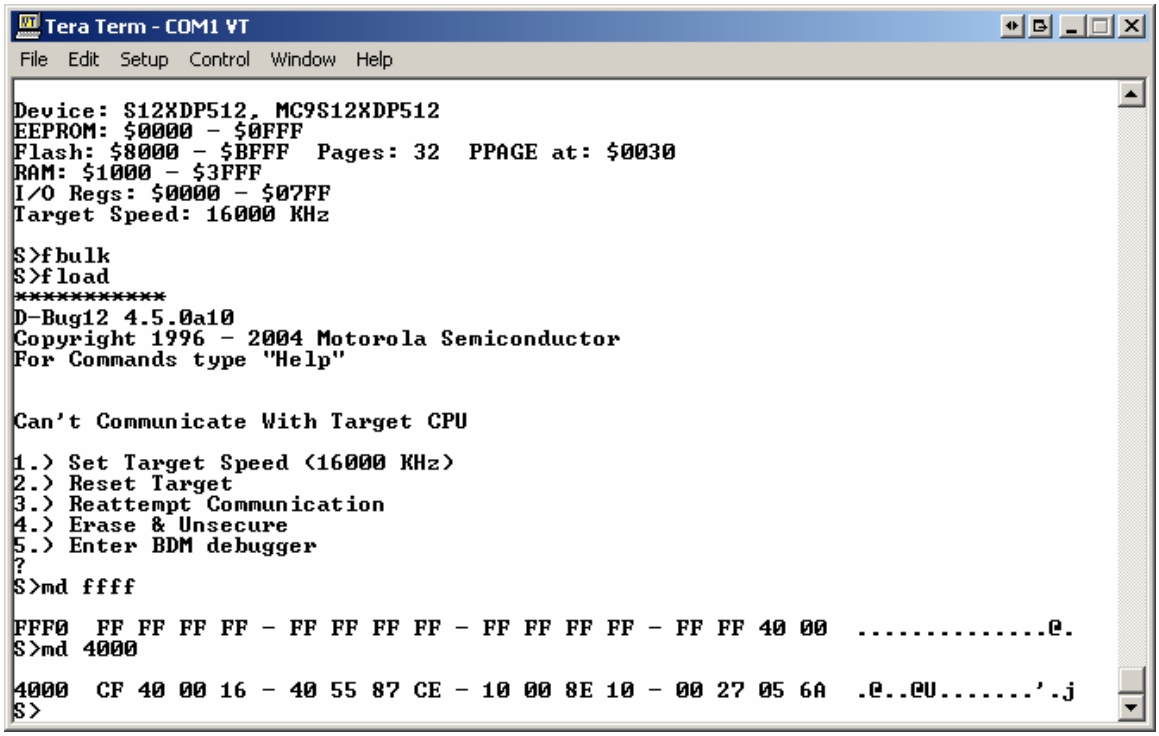
Checking:

Use the BDM pod to check the vector address in Flash.



Check start of Program:

Use the BDM pod to check address $4000, as shown.



Press the Adapt9S12XDP512 RESET button.

Once the RESET button is pressed, the Red LED will begin to blink. This concludes how to use ICC12 with Adapt9S12XDP512 and MicroBDMXG BDM pod.