How to use ICC12 with Adapt812DX and FLASH Loader

This document will show and demonstrate the use of ImageCraft ICC12 Latest ***Version 6*** with Technological Arts' Adapt812DX module.

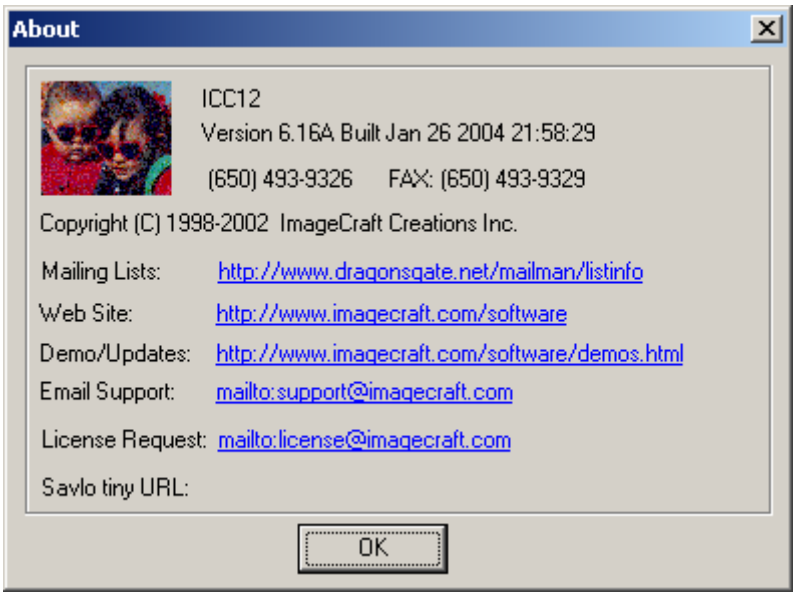
The MXFLash Loader is a utility to program the external FLASH of Adapt812DX ,DXLT or Adapt812 + MX1 combo module. The file can be downloaded at the link below.

http://www.interlog.com/~techart/myfiles/files/mxflash.zip

It will be used here to erase and program FLASH after the compilation of a test program.

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:

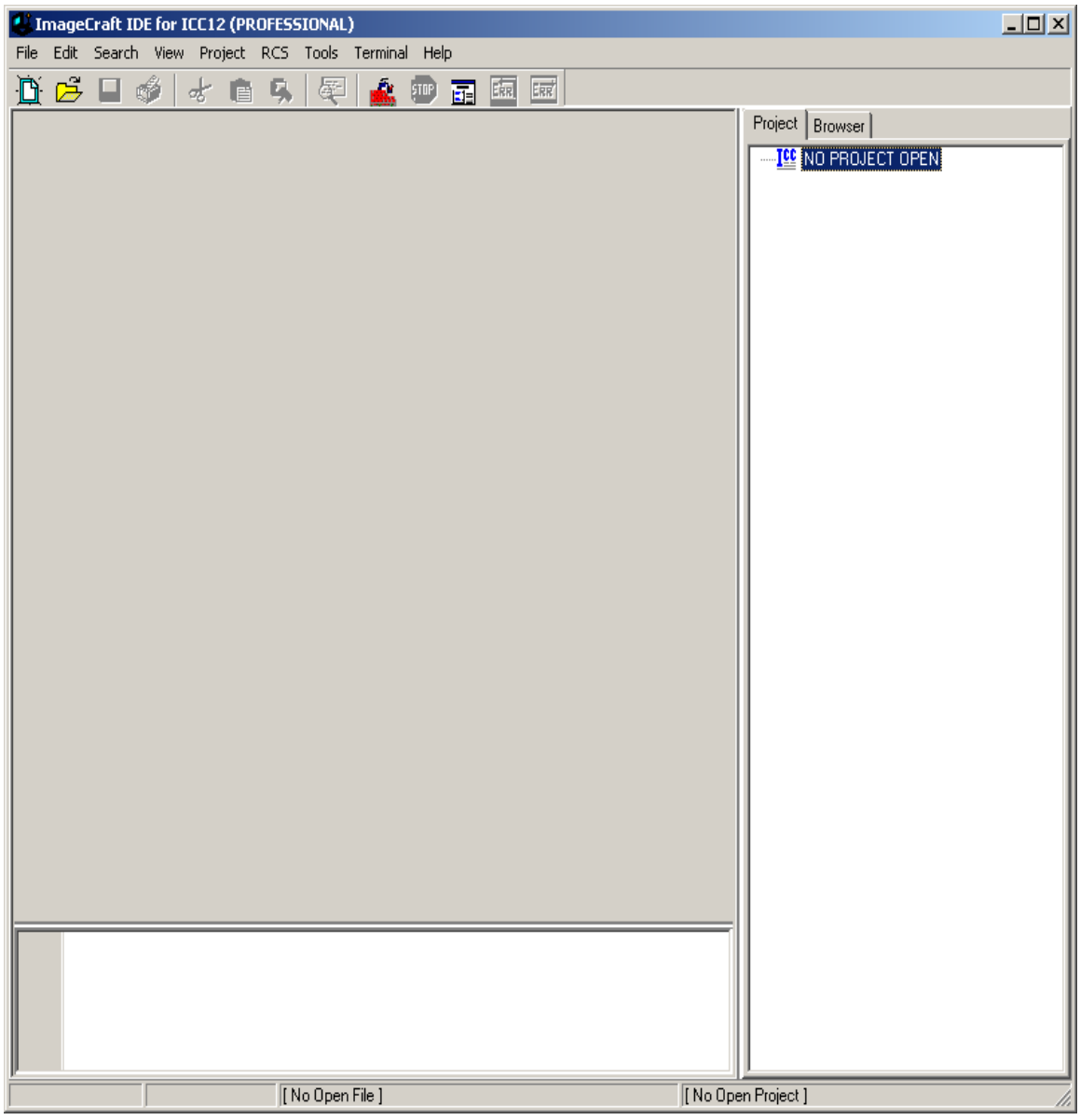
http://www.technologicalarts.com/myfiles/ad812dx.html
http://www.technologicalarts.com/myfiles/812dxlt.html

Getting Started:

Double click on the ICC12 icon. If a user has not read the ICC12 manual and just open the IDE one will wonder what to do next. Well wonder no more.

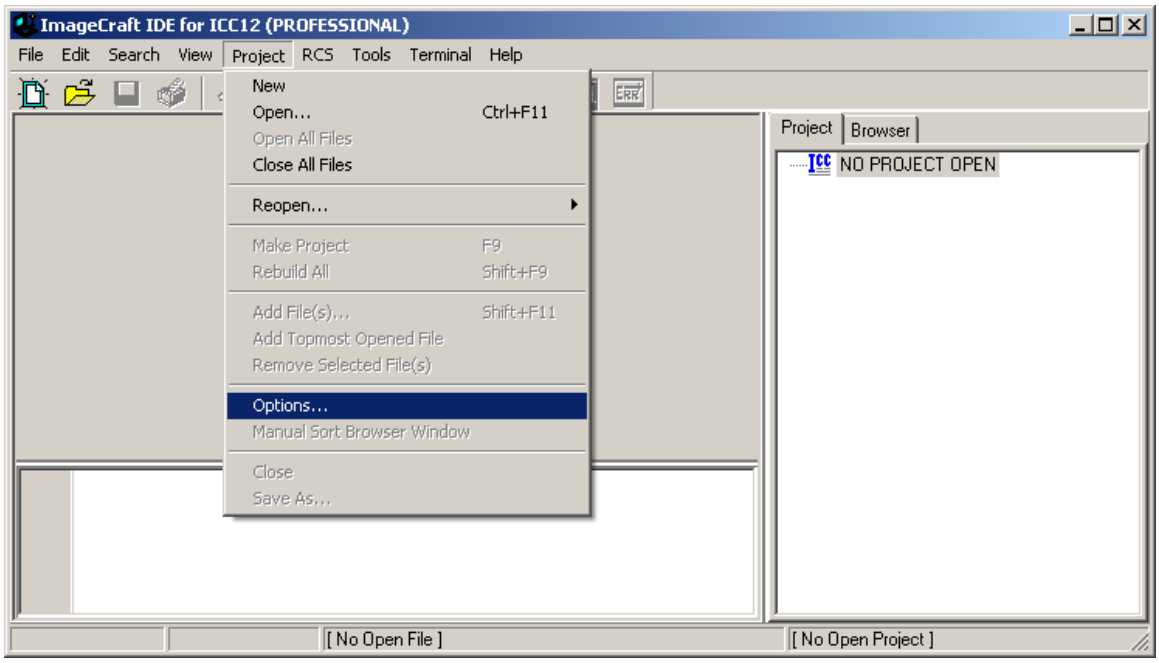
Note the 3 window panes. The top left most is greyed out and the right is the project window. The left bottom pane is where the error messages are displayed during compilation.

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 setup the type of MCU the C program will compile for. In this example it is the Adapt812DX, Adapt812DXLT or the Adapt812 + MX1 card.

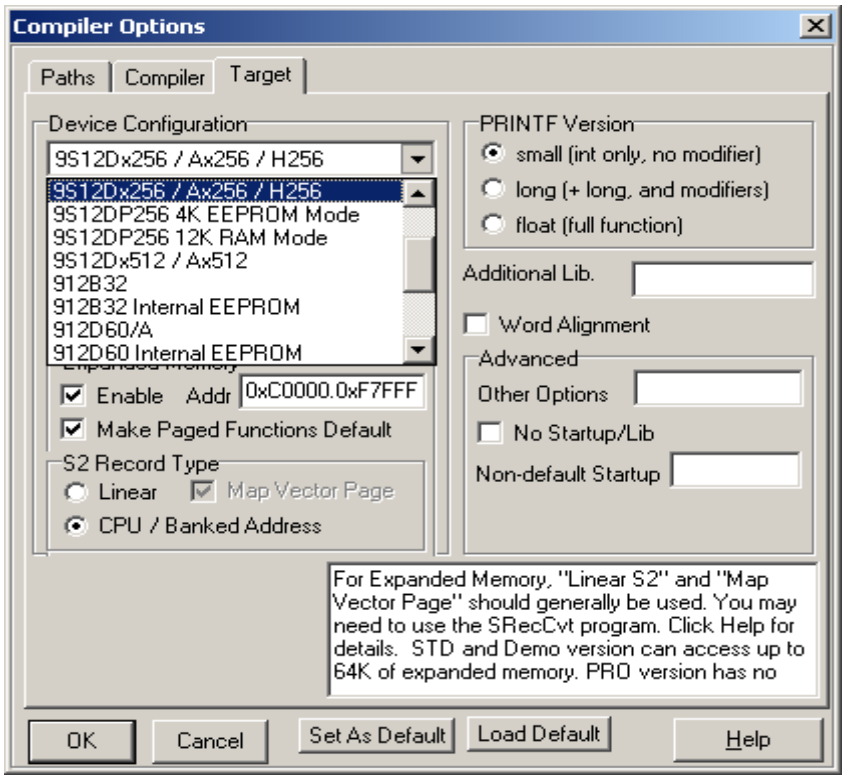


Compiler Setup:

Click on Project Menu – Options – Target Tab.

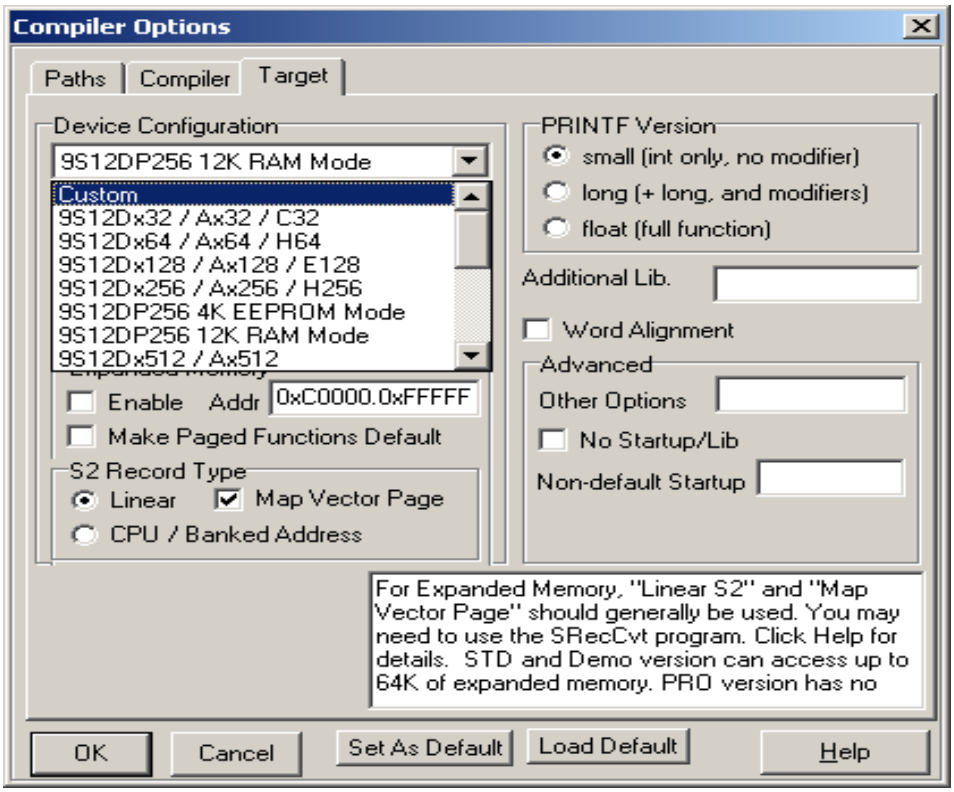


Please note the Device Configuration. Click on the pull down arrow to change the device type.



Scroll up or down to select Custom as shown. Note that an Adapt812DX device Configuration does already exist. Unfortunately, the addresses are not setup properly, therefore the Custom configuration must be selected and the memory parameters are changed for proper operations.

In this example an ***Adapt812DX with 128Kbyte FLASH*** is used. Process are the same for different FLASH sizes.



Device Configuration:

Program Memory: ***0xC000***
Data Memory: ***0x7000***
Stack Pointer: ***0x0C00***

The external memory area from 0xC000 to $FFFF is where the start of code and the ISR must reside. Any interrupt routine must be in this area. It is possible to call routine in another PAGE while servicing the ISR.

The external memory area 0x7000 to 0x7FFF is RAM. The DPAGE is enabled and is used to access the RAM memory in paged window.

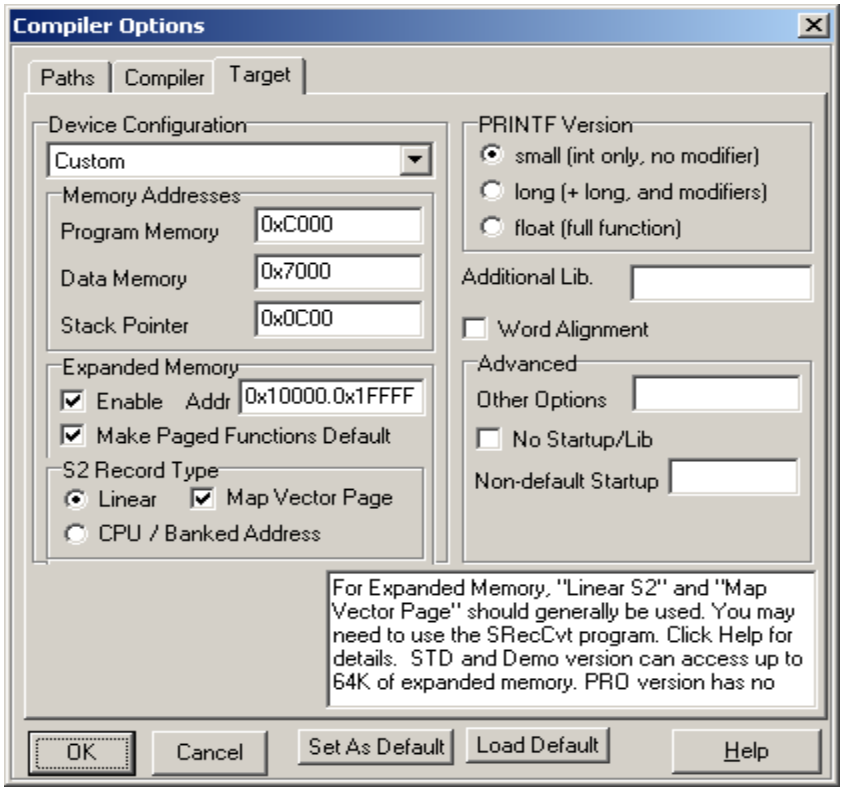
The RAM memory area 0x0800 to 0x0C00 is internal to the MCU. The stack pointer will use this area.

Expanded Memory:

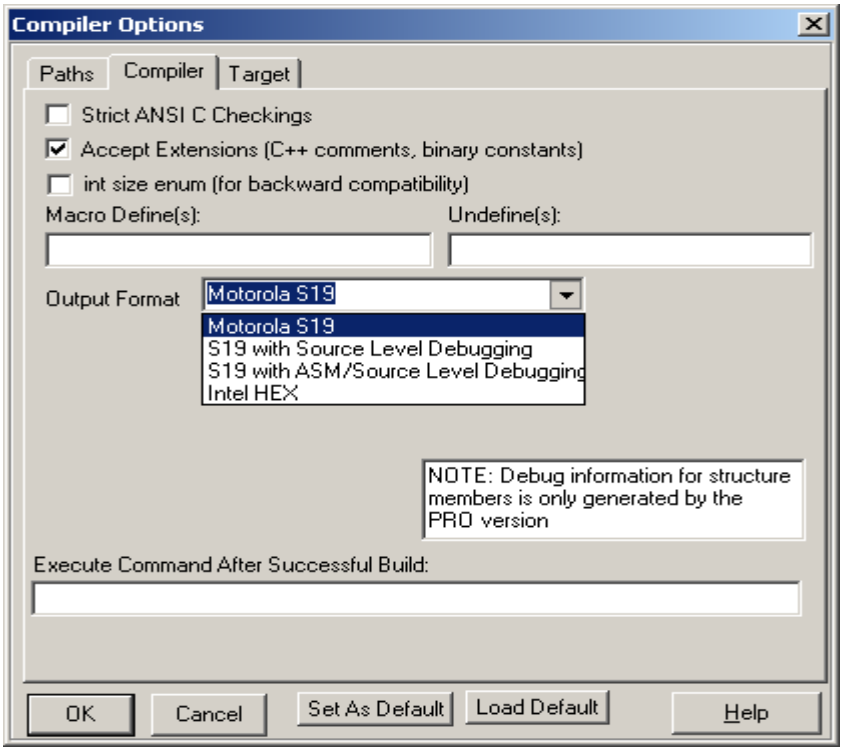
Note the address range is ***0x10000. 0X1FFFF*** for a 128Kbyte FLASH.
For 512Kbyte Flash the address range is ***0x10000. 0X7FFFF.***

S2 Record Type:

Select Linear and check marked the Map Vector Page. ICC12 will output Linear S2 record.

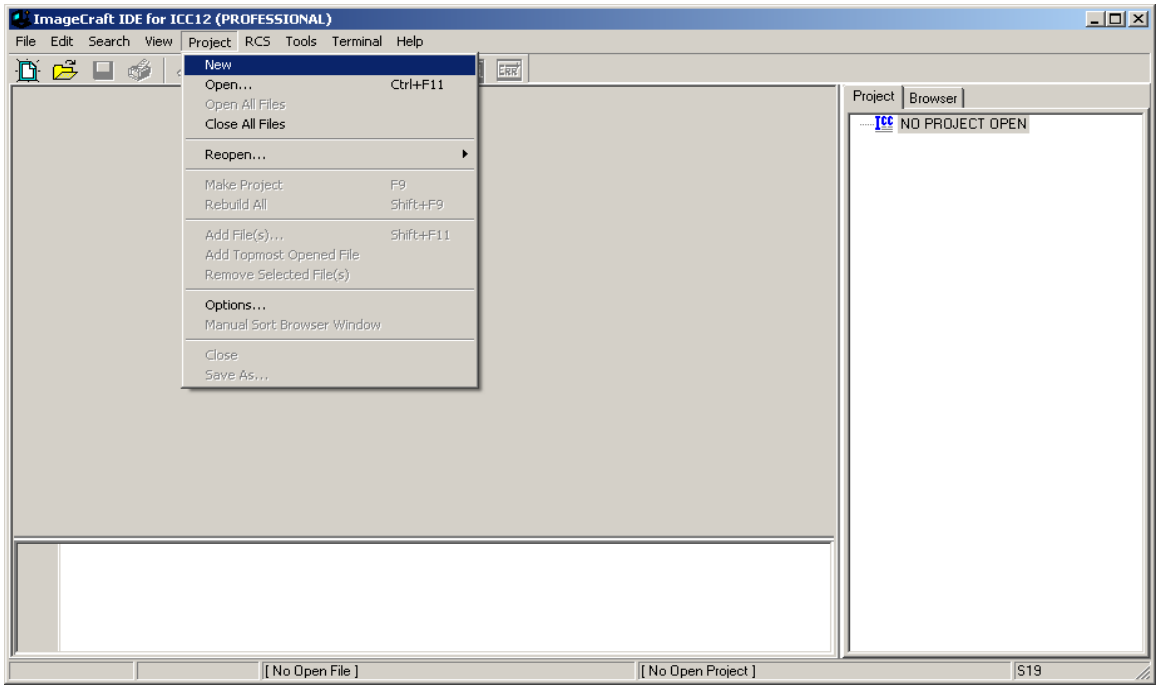


On the compiler tab there are several choices of S-record output as shown. Select which one that suits you.

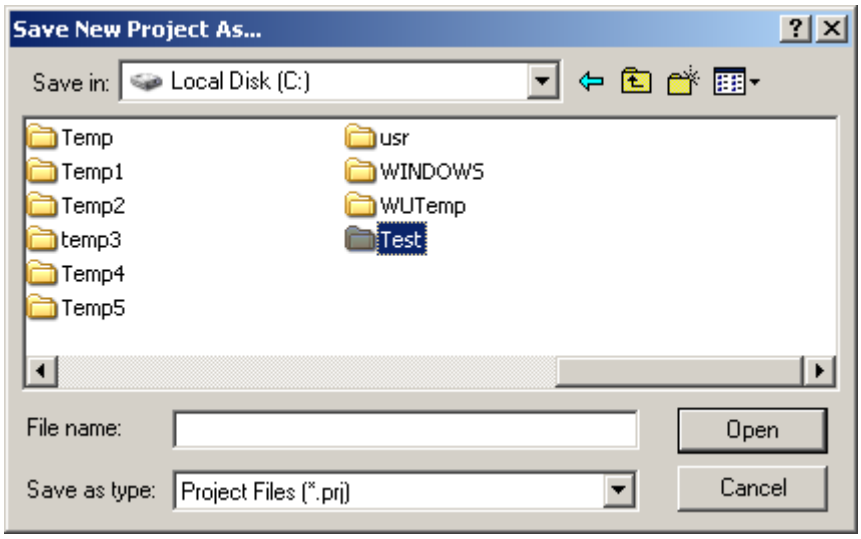


Starting a new Project:

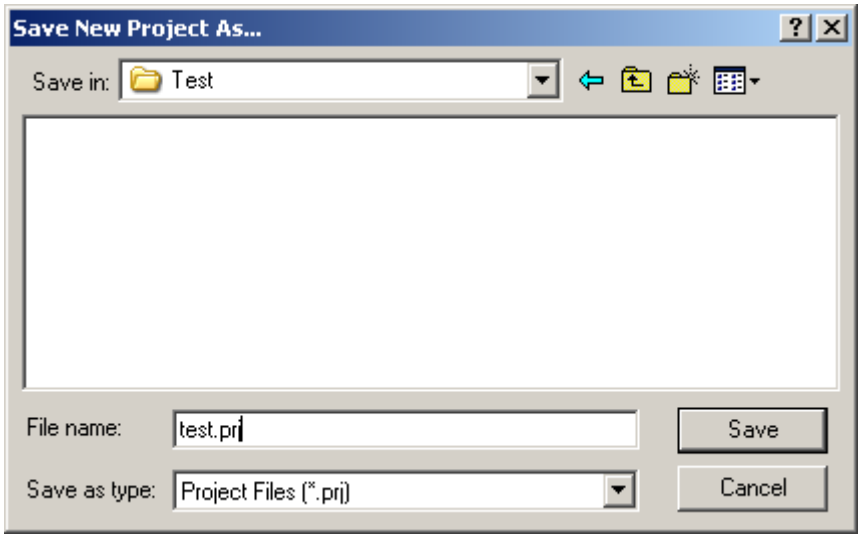
Once the compiler options are setup, a new project can be created. Click Project menu – New.



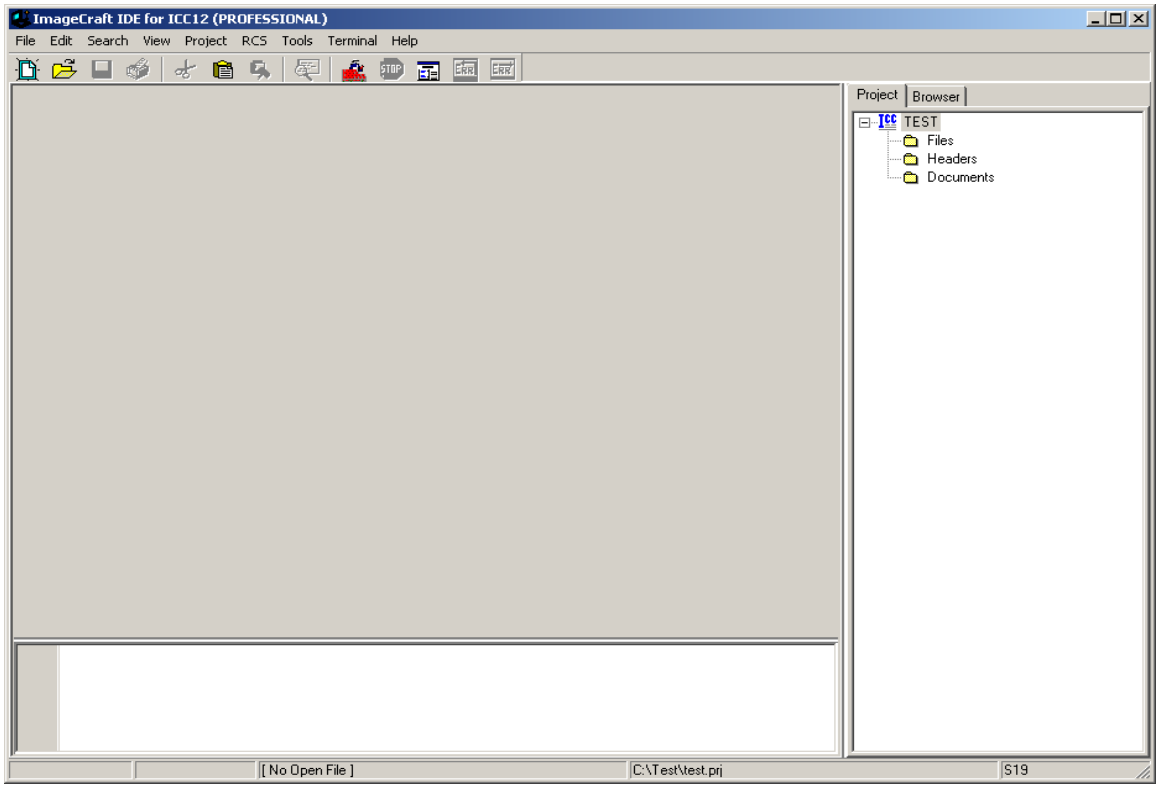
The ICC12 will prompt to save the new project. The user should decide whether to create a new directory to save the new project. In this example a new directory called ***Test*** is created and the file is saved as file ***test.prj***.



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

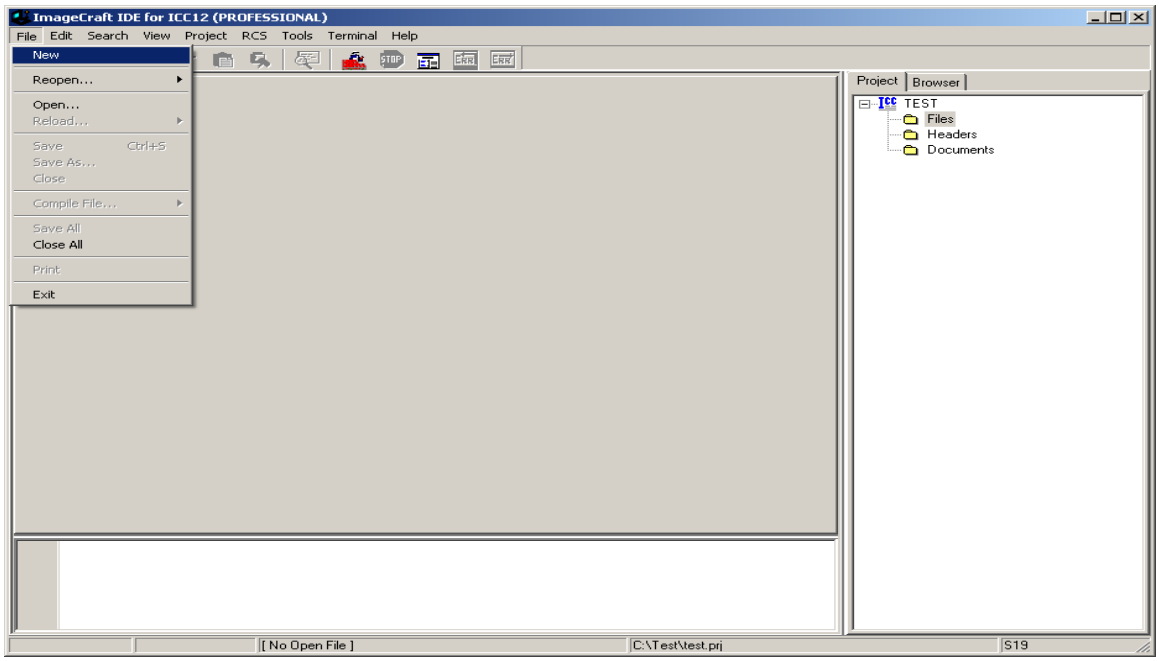


Note that the project window has changed to add Files, Headers and Documents.

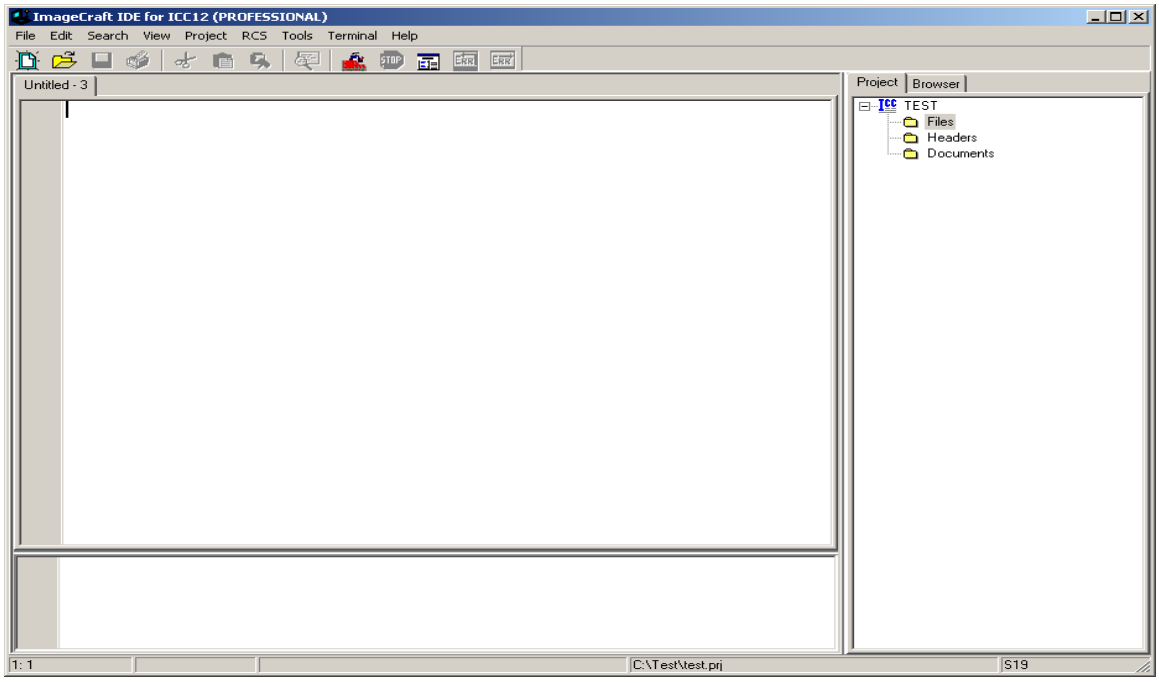


Creating a new file to the project:

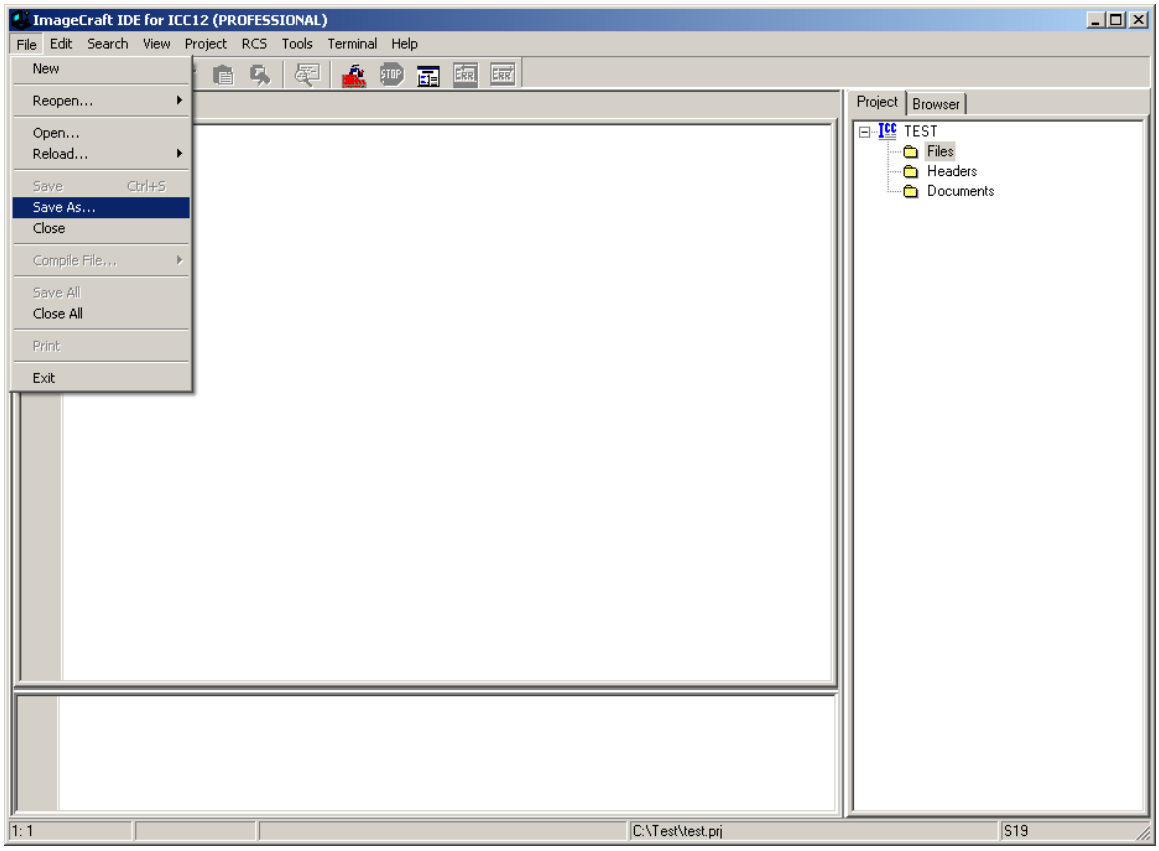
To add files to the project, click on the File menu – new as shown.



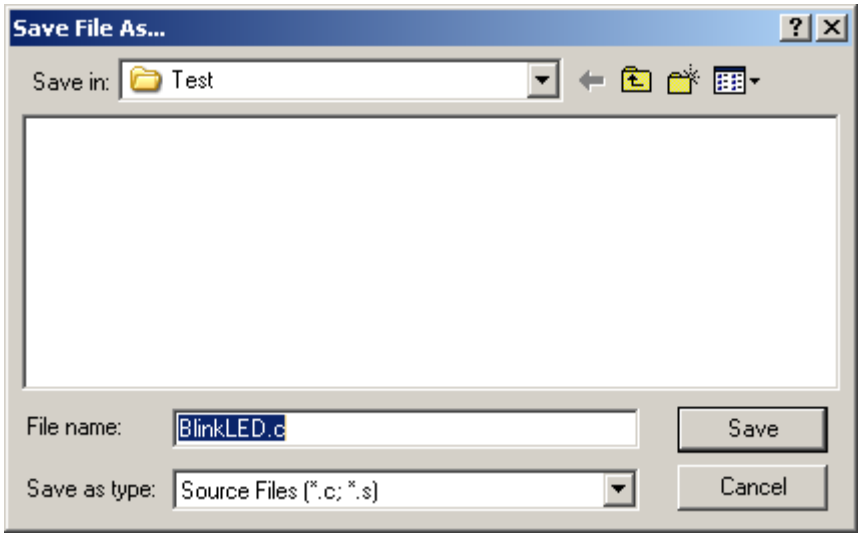
Note that ICC12 created an untitled file. Save the file as ***BlinkLED.C***.



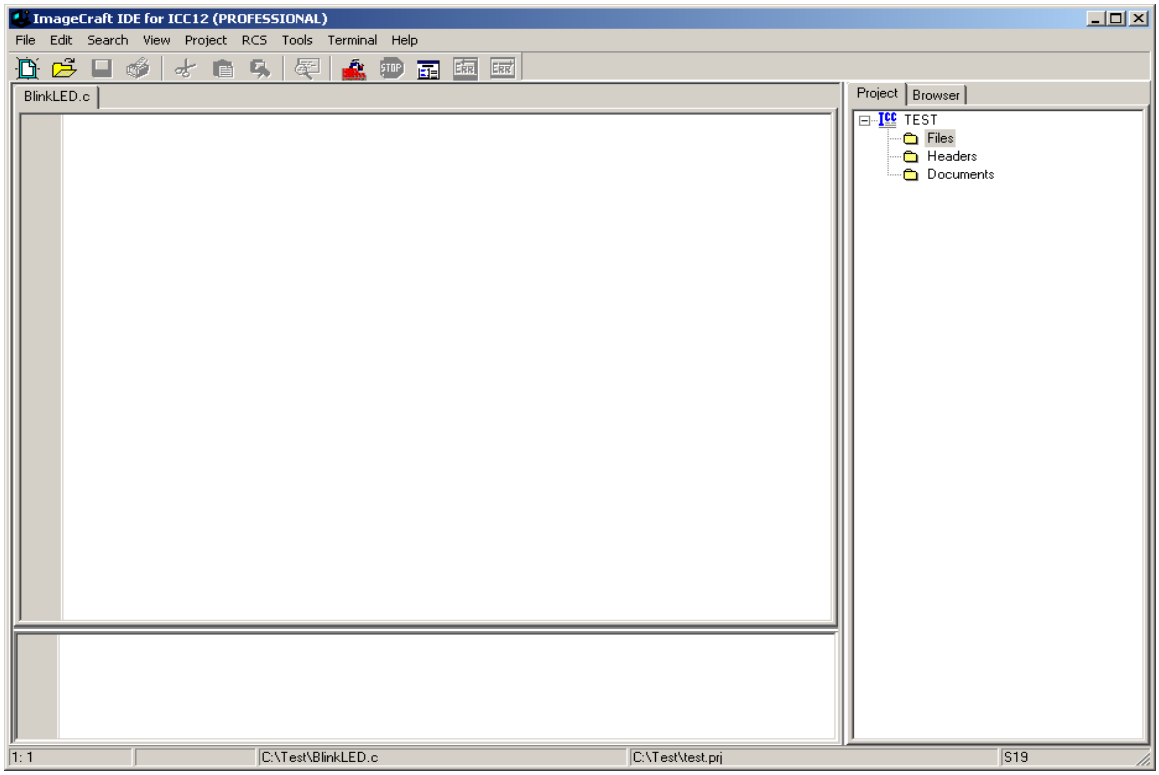
To save, click on File menu – Save As



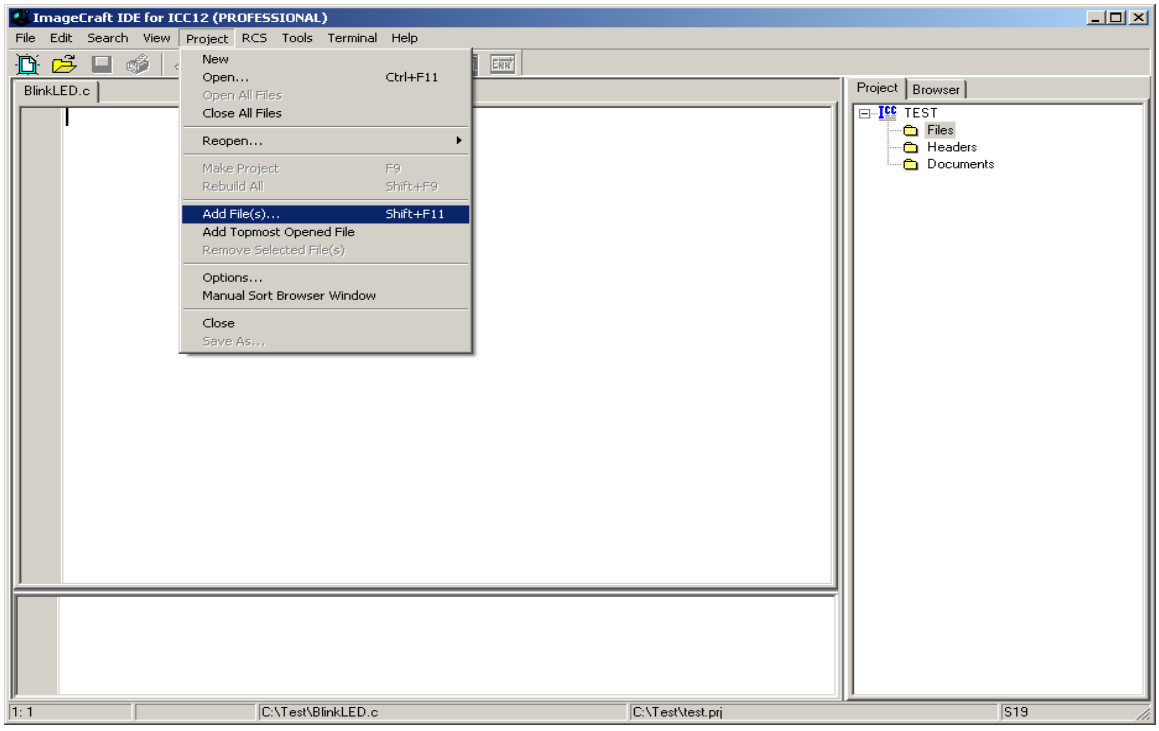
ICC12 will open an explorer window to help save the file. Type BlinkLED.c then press the save button.



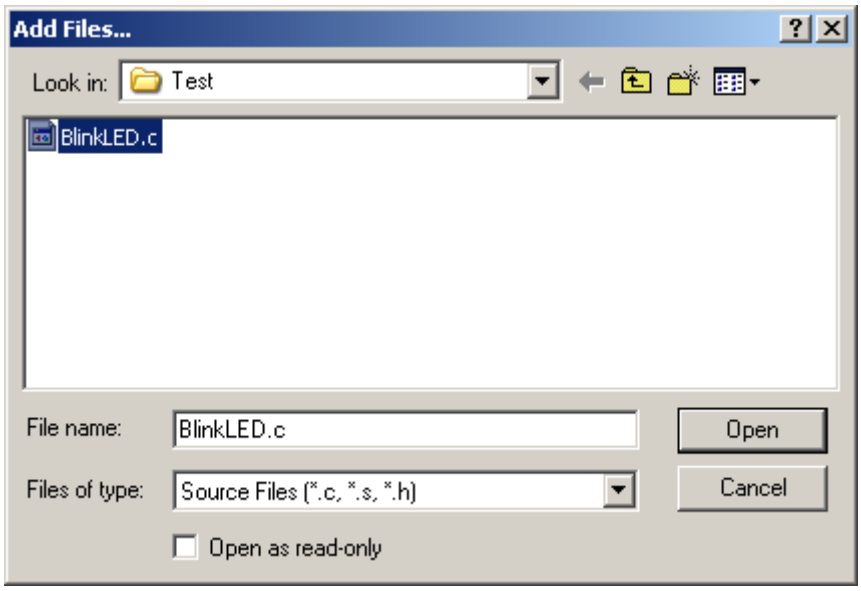
Note that ICC12 has renamed the file to BlinkLED.c.



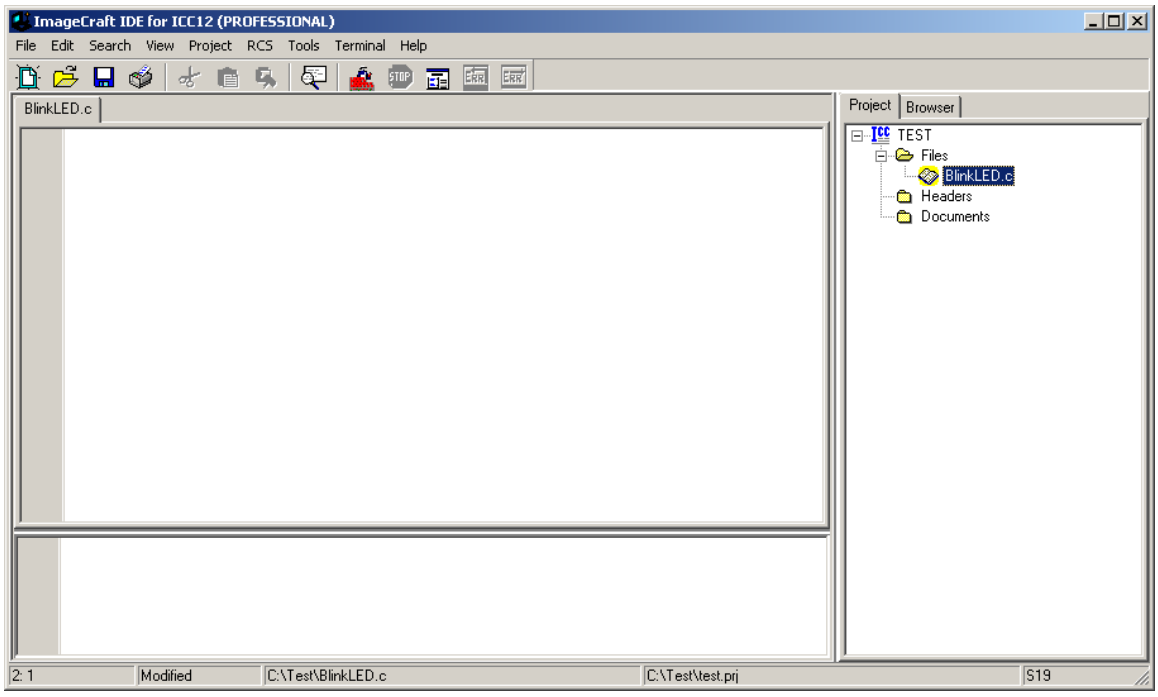
To add BlinkLED.c to the Project, click on the Project menu – Add File(s)



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

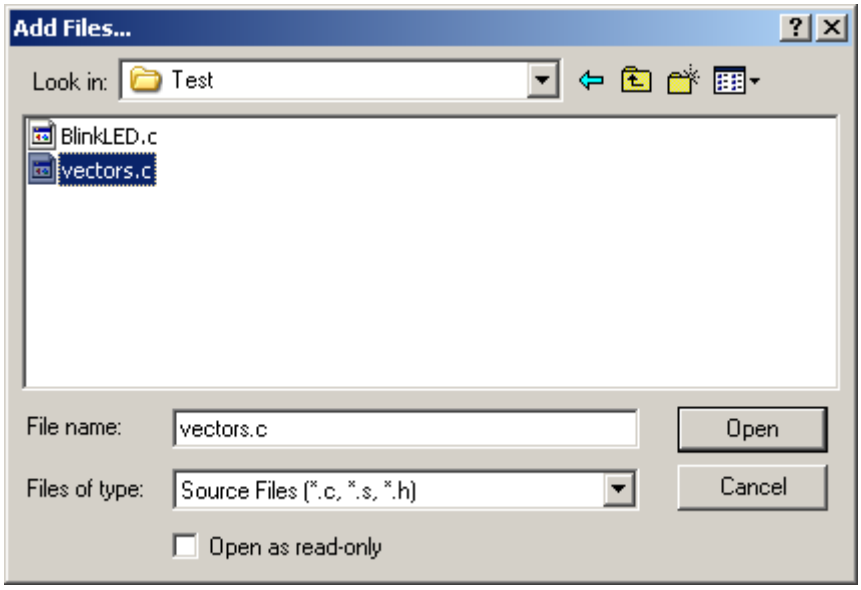


Note that the right window pane has changed to include BlinkLED.c under the Files Project.

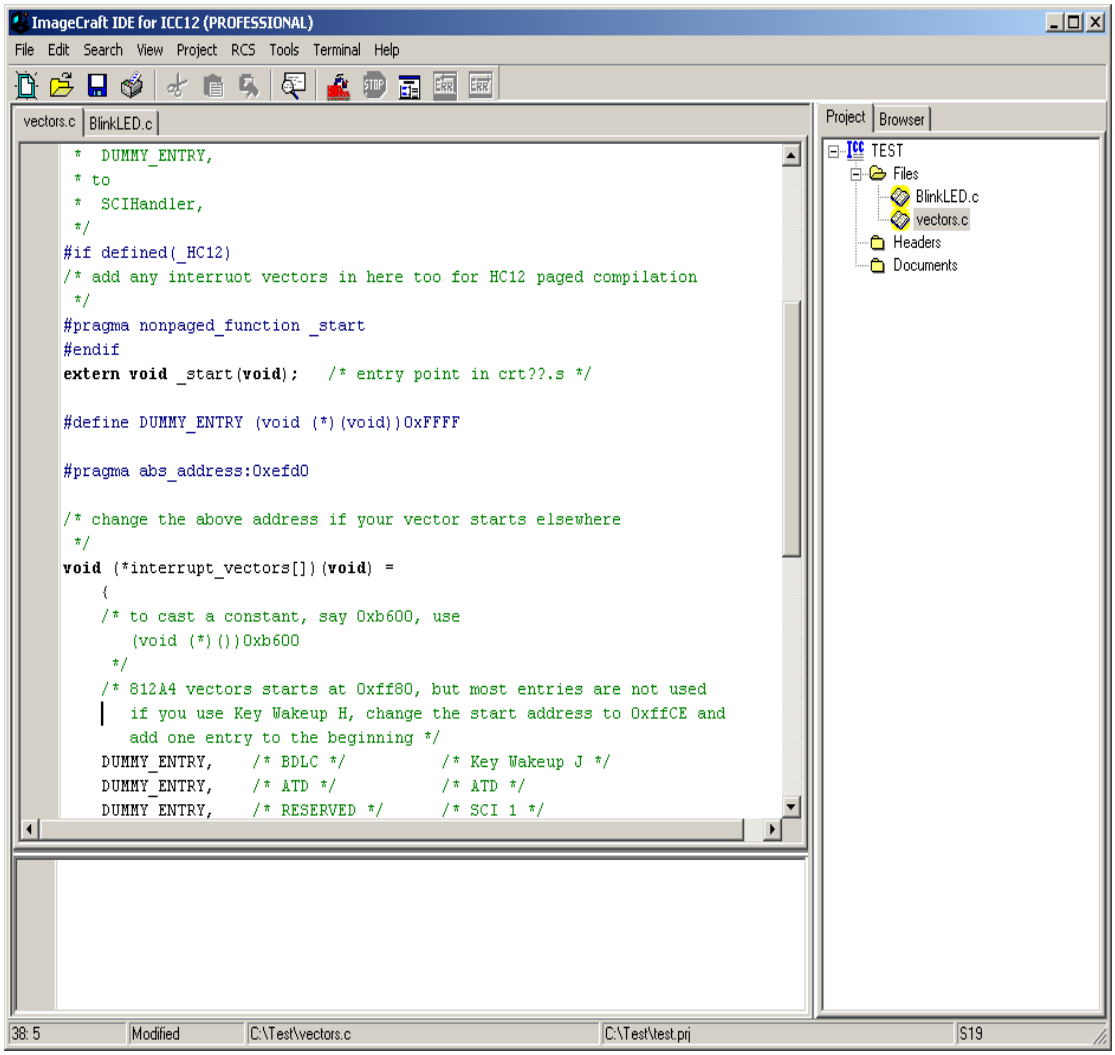


Locate ***vectors.c*** and copy file to Test directory. The major reason why this must be done is because of project to project dependency. It is not good to keep editing a single ***vectors.c*** if other projects are using this same file. It becomes a problem to keep track of the changes made to the different projects.

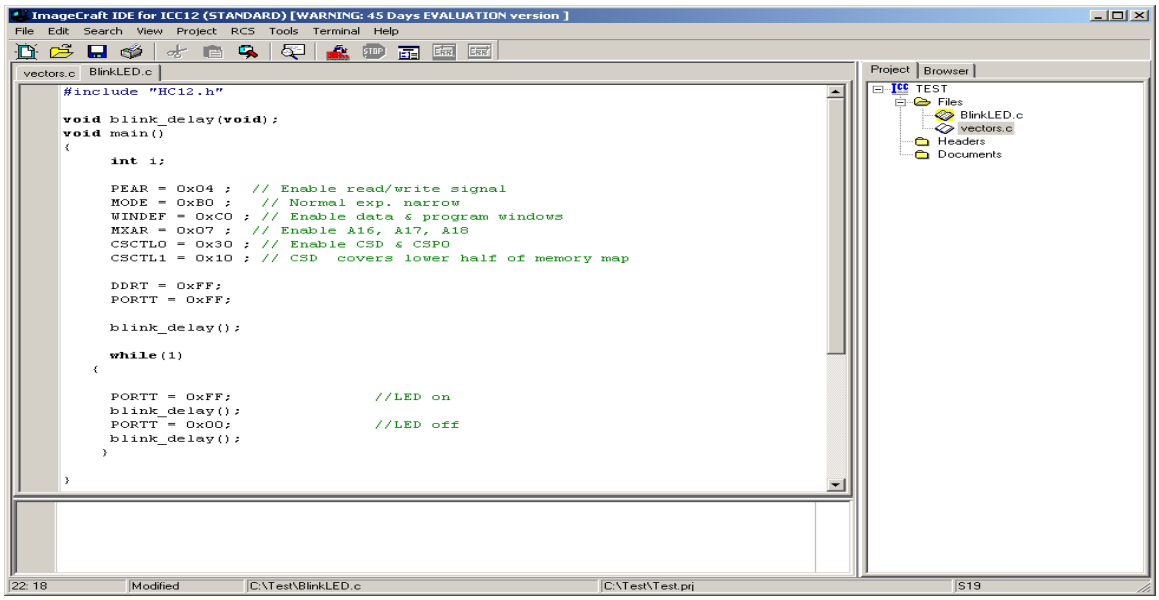
To add ***vectors.c*** to the Project, click on the Project menu – Add File(s)



Note that ICC12 has changed to include ***vectors.c.*** Note that the **vectors.c** was written for the 68HC912B32 and 812A4 MCUs.



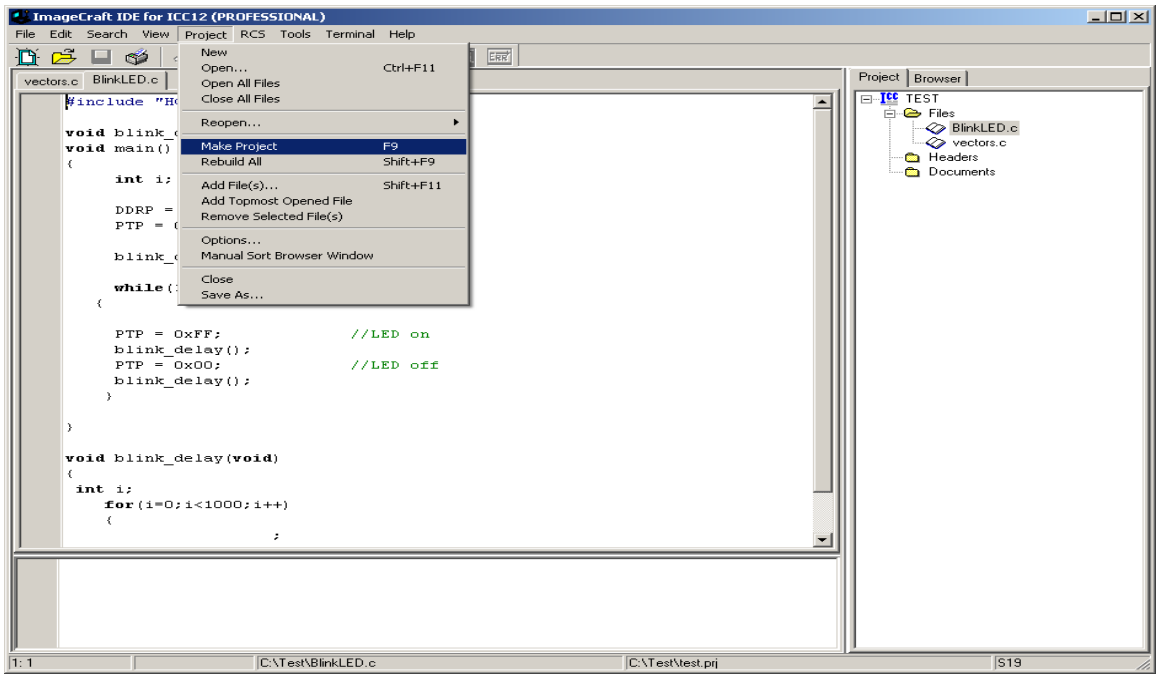
Write the codes below into BlinkLED.c file. Once it is written we can then compile/make/build the code.



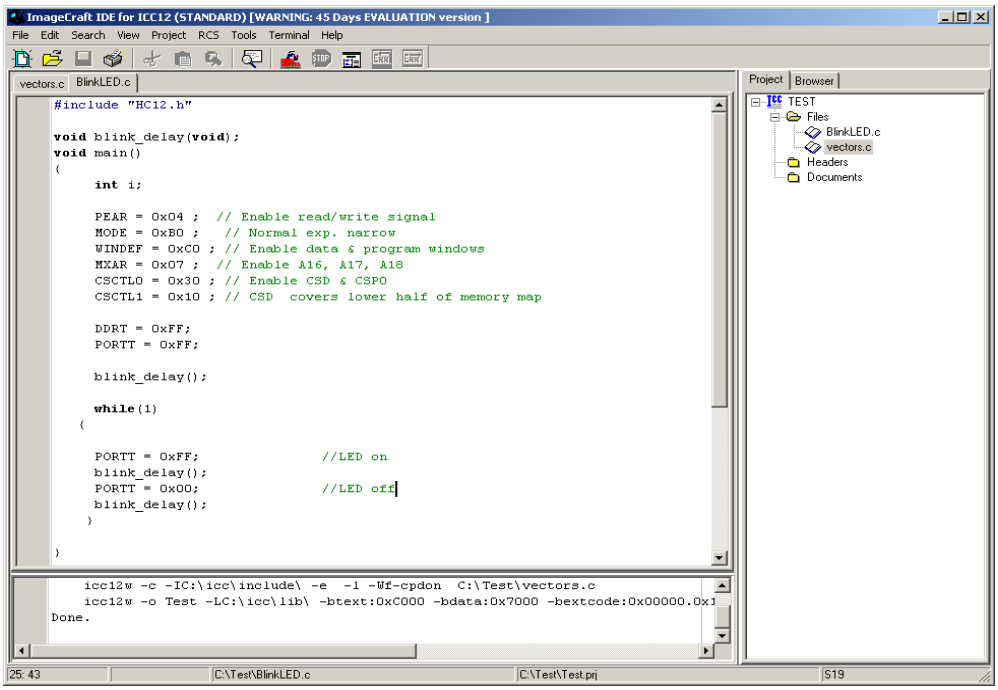

```
#include "HC12.h"
void blink_delay(void);
void main()
{
        int i;

        COPCTL = 0x00 ;       // Disable COP
        PEAR = 0x04 ; // Enable read/write signal
        MODE = 0xB0 ;   // Normal exp. narrow
        WINDEF = 0xC0 ; // Enable data & program windows
        MXAR = 0x07 ;// Enable A16, A17, A18
        CSCTL0 = 0x30 ; // Enable CSD & CSP0
        CSCTL1 = 0x10 ; // CSD  covers lower half of memory map

        DDRT = 0xFF;
        PORTT = 0xFF;

        blink_delay();

        while(1)
  {

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

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

Compiling/Build/Make the file:

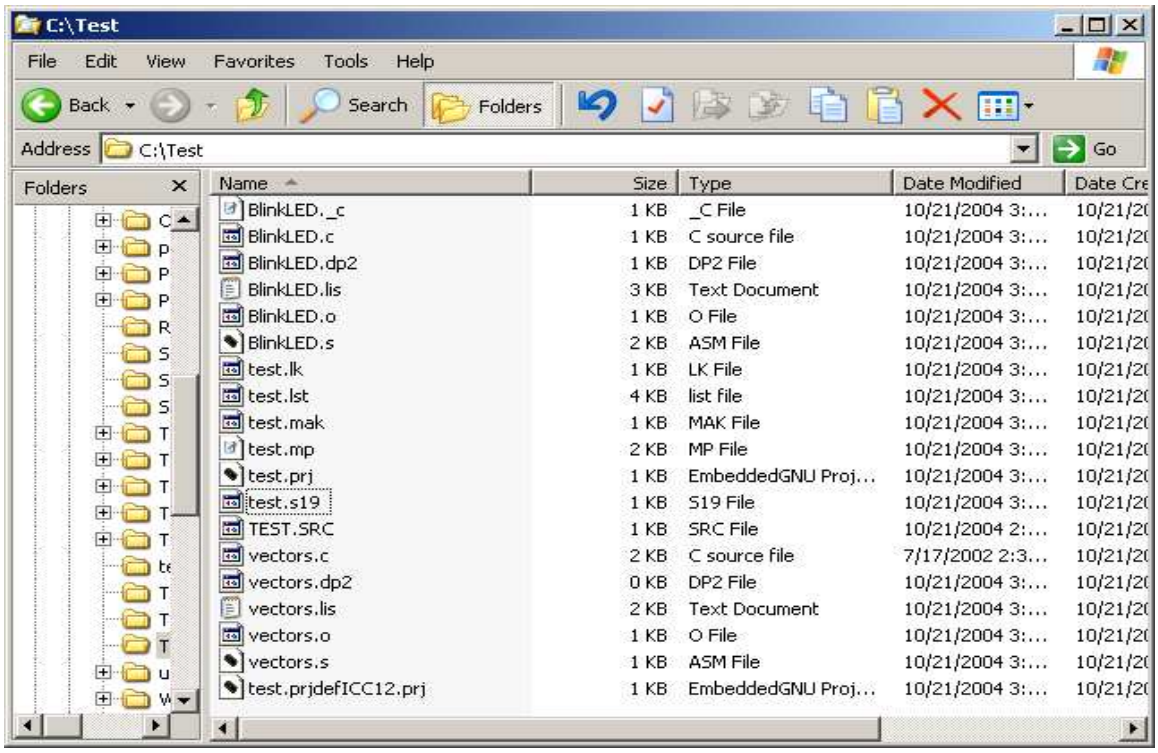
To make the file click Project menu – make project as shown.



Note the bottom window pane will show messages to display how the build progressed. Any errors, if any, are shown in this window. The build was without error so we can progress to erasing and programming the Adapt812DX.



Note the other extraneous files are created after a make.



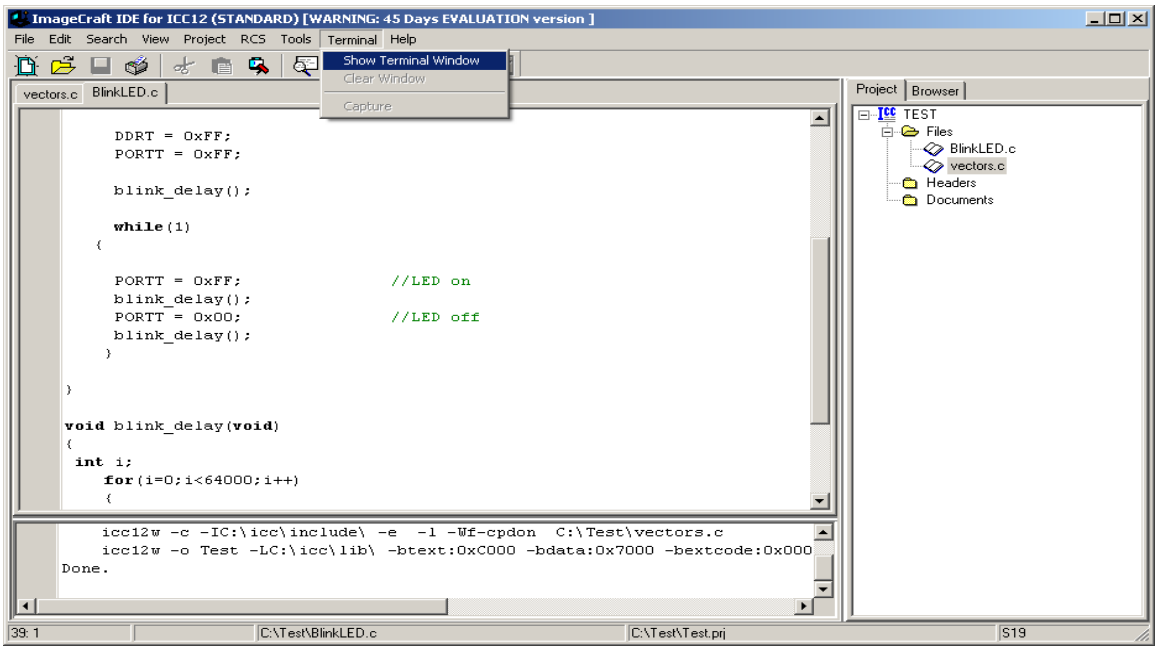
Using WordPad to check the content of ***test.s19*** file. Note that the S-records are of different lengths. In reality the test.s19 record is an S2 type.

S20F01C000CF0C0016C07387CE70008EB8
S21101C00B700027056A000820F6CEC078CD2B
S21201C01870008EC0782706180A307020F516C4
S20801C026C02A20FE08
S21101C02A34B7751B9EC6047B000AC6B07BAA
S21201C037000BC6C07B0037C6077B0038C6303C
S21201C0457B003CC6107B003DC6FF7B00AFC6ED
S21101C053FF7B00AE4A8000042010C6FF7B74
S21001C06000AE4A8000047900AE4A800061
S20B01C06C0420EEB757303D3A
S21201000034B7751B9ECC00006C1E2007EC1E4C
S21201000EC300016C1EEC1E8CFA0025F2B757DB
S20601001C300AA2
S21201FFD0FFFFFFFFFFFFFFFFFFFFFFFFFF2B
S21201FFDEFFFFFFFFFFFFFFFFFFFFFFFFFF1D
S21201FFECFFFFFFFFFFFFFFFFFFFFFFFFFF0F
S20A01FFFAFFFFFFFC0003F
S20901C0731D0016073D4B
S903C0003C

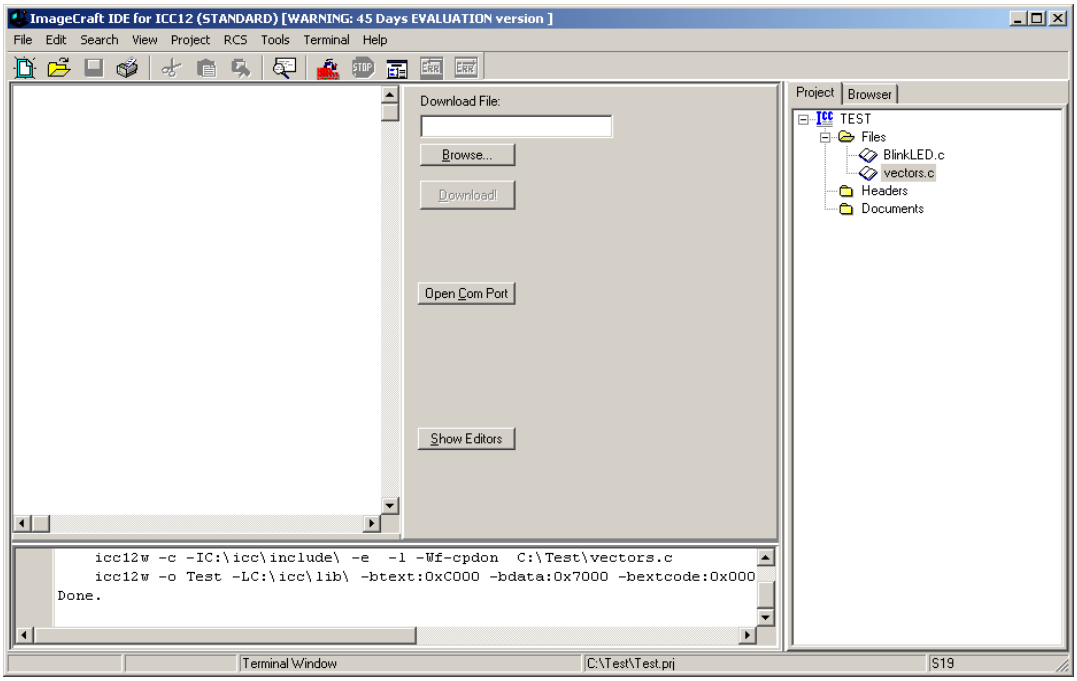
Programming the Adapt812DX:

Any terminal program can be use to communicate to the FLASH Loader. In this example ICC12 is used. Connect Serial cable to any available PC COM port and the other end to Adapt812DX.

Click Terminal menu – Show Terminal window as shown

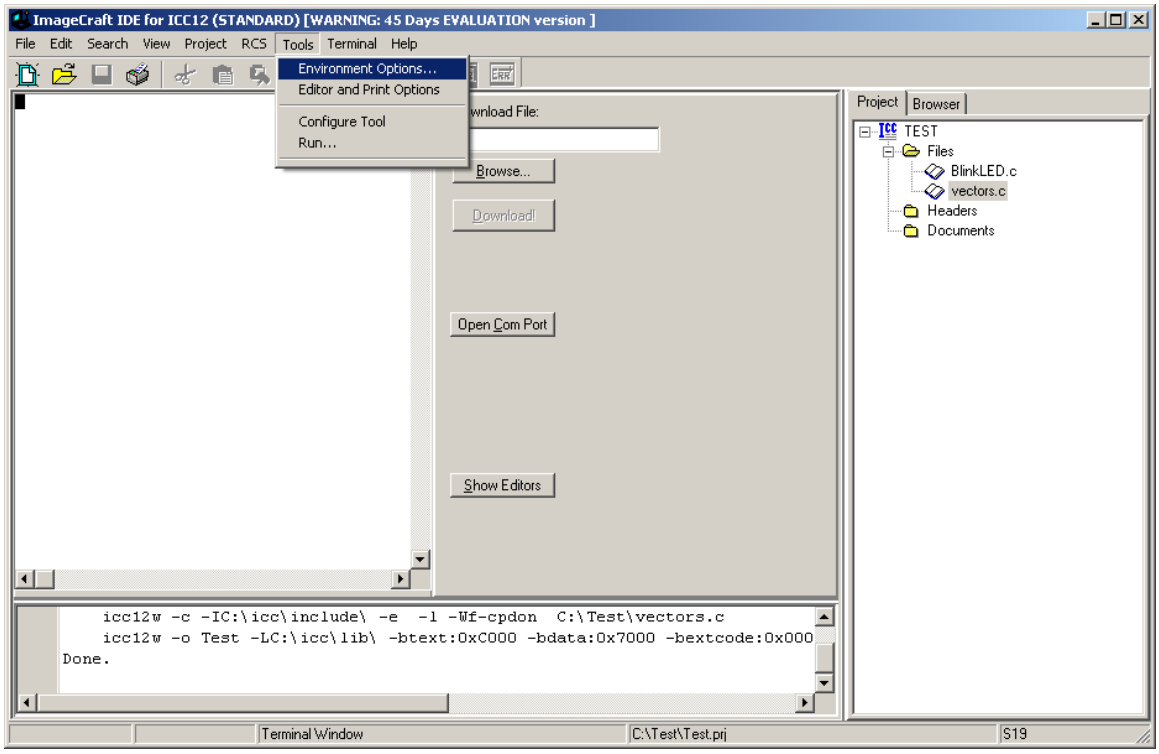


Note that ICC12 will immediately change to show terminal window.

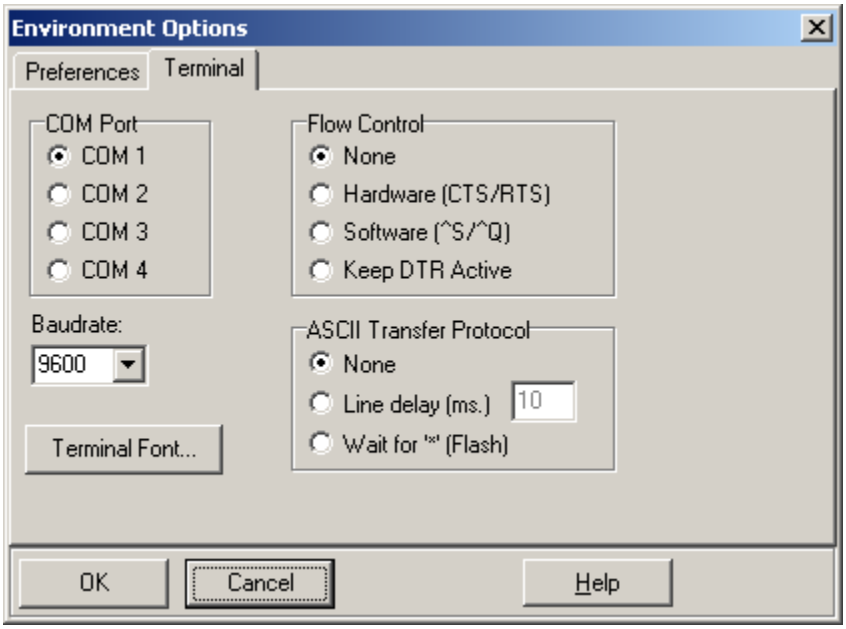


Setup the terminal BAUD by clicking on Tools menu – Environment options as

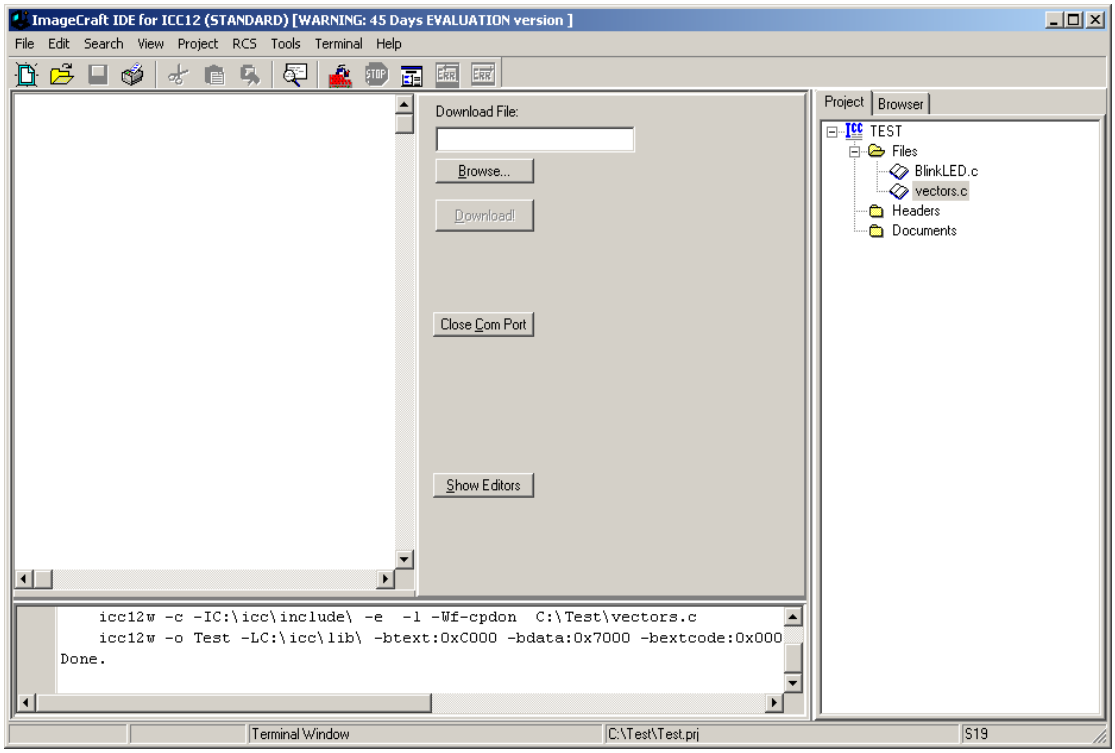
shown.



Select the Terminal Tab and the set the BAUD = 9600, Flow control = None, ASCII Transfer Protocol = None. Select the COM that the serial cable is connected to at the back of the PC.



In the middle of ICC12 is the Open COM Port button. Push the button to open COM port as shown. Note that it will change to Close Com Port.



The MCU must boot up in single chip mode in order for the Loader to work. Set the various jumpers as indicated below.

Jumper settings for MODA and MODB for single chip mode.

JB1 – pin 2 and 3 shunted – MODB
JB5 – shunted – MODA
SW3 to SGL (single chip) position.

Jumper settings for FLASH and RAM with REV0 to REV2A

JB3 – pin 1 and 2 shunted
JB4 – CSP0* and F shunted
JB6 – CSD* and R shunted

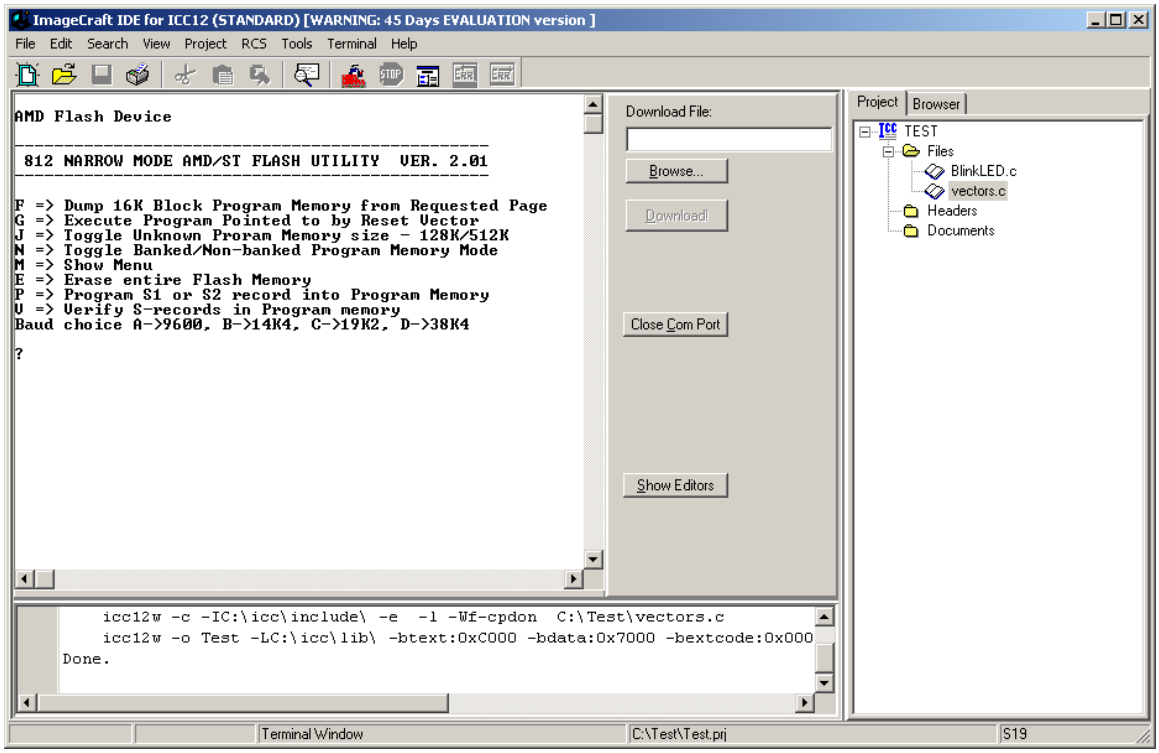
Jumper settings for FLASH and RAM with REV3

JB3 – pin 1 and 2 shunted (CSD* RAM enable)
JB6 – pin 1 and 2 shunted (CSP0* FLASH select)
JB7 – pin 1 and 2 shunted (RAM write enable)

Connect Serial Cable to COM 1 at the back of PC and to the Adapt812DX. Slide SW2 Run/Boot switch to Run position. Make sure SW3 is in SGL position.

Power up the Adapt812DX board with a known good power supply, making sure the PWR LED is on. Once the board is powered, ICC12 Terminal should display the MXFlash Loader menu as shown.

Notice that it detected the FLASH to be ***AMD Flash Device***. If the Loader cannot detect a FLASH memory it will display unknown Memory and will assume that the memory is RAM.

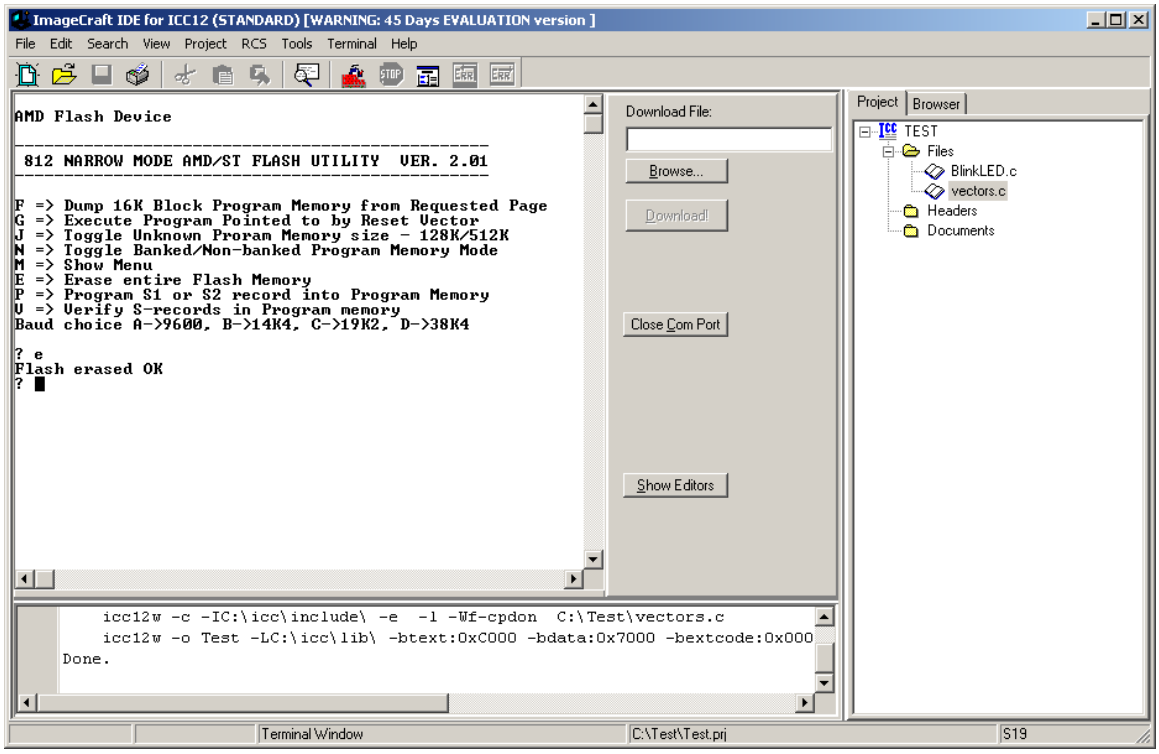


There are several things to check if the Menu will not show up as shown above.

- Do not use NULL cable. Use a straight thru cable.
- Check and verify the cable is connected to the correct COM port.
- Re-boot the PC if necessary as other application may have corrupted the OS.
- Use known good serial cable.
- If there are other problems, test the COM port with another device that is known to work.
- If the FLASH type is not recognized re-check jumper settings. If still unrecognized then the board has a problem.

Erasing:

To erase the FLASH the command is ***E***.



Programming:

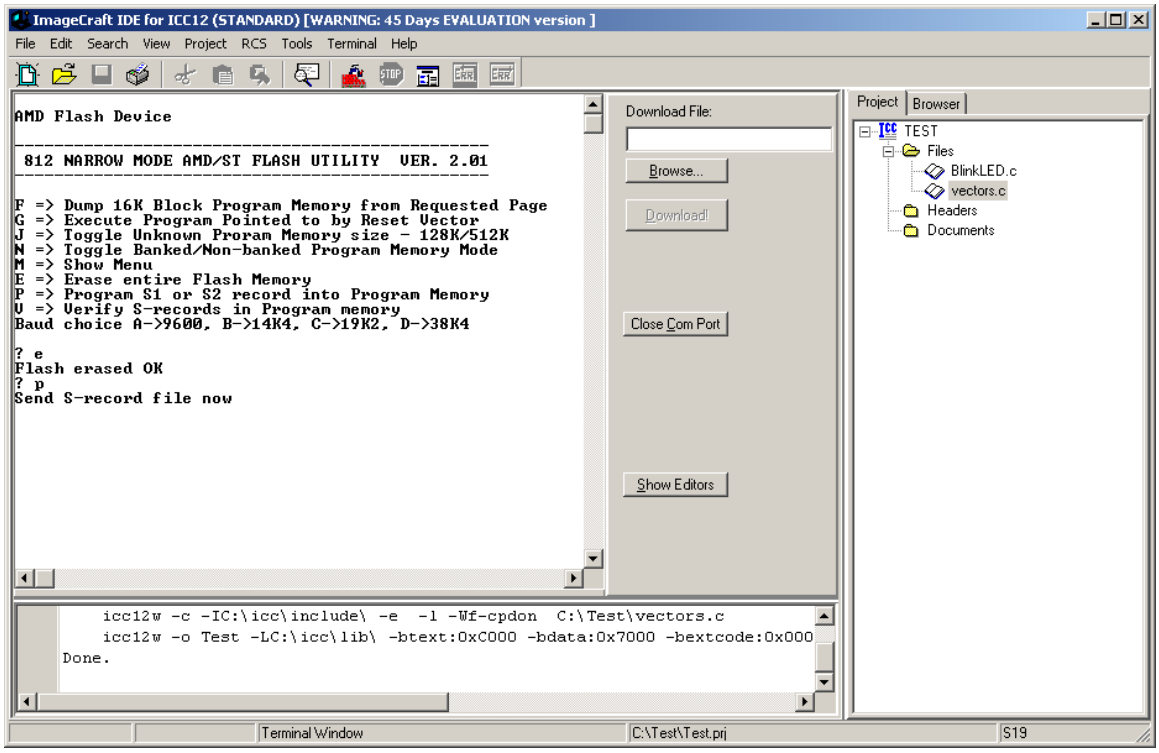
There are 2 options for programming the FLASH. The 1st is whether the S-record is BANKED. The command is ***N*** to toggle from BANKED to NON-BANKED. ICC12 creates mixed S1 and S2 record. This is generally called BANKED mode. One should always check the S19 record to see if there are S1 and S2 mixture. Other compiler creates BANKED S2 too. One should experiment whether to program the S-record as BANKED or NON-BANKED.

By choosing the ***N*** command the loader will always toggle from previous state to the next. By default it is set for BANKED mode.

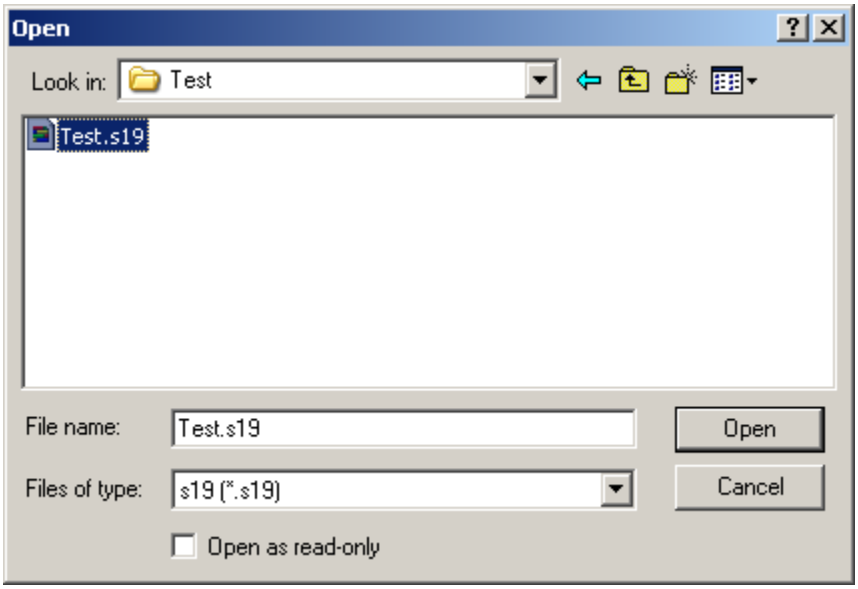
If the S-record contain only addresses from $C000 to $FFFF then it should always be programmed as NON-BANKED. Linear S2 record must always be programmed as NON-BANKED.

Note: For ICC12 the S2 record is programmed as BANKED.

Type P to initiate program.

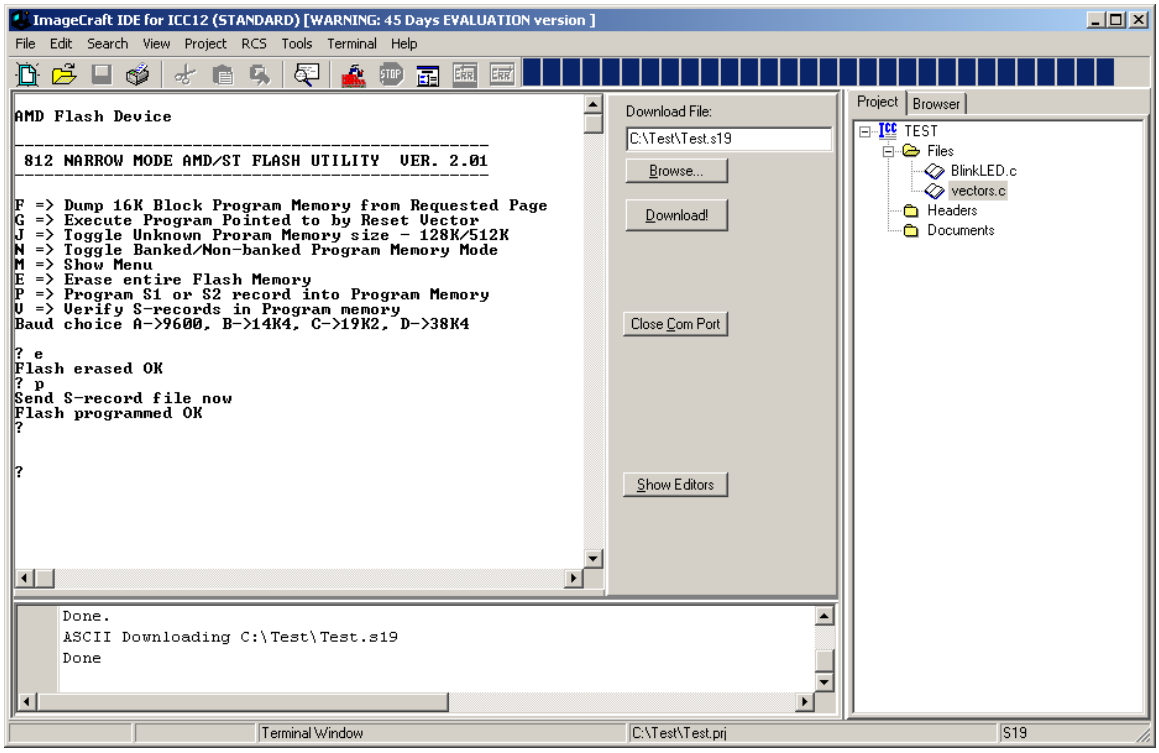


Use the ***Browse*** button to help locate the file to be uploaded.



Select the Test.s19 file then press the Open button.

Once selected the ***Download*** button will now become active.



After programming, slide SW3 to EXP (MODA = 1) then press the RESET button. The programmed application in now running in expanded NARROW mode.

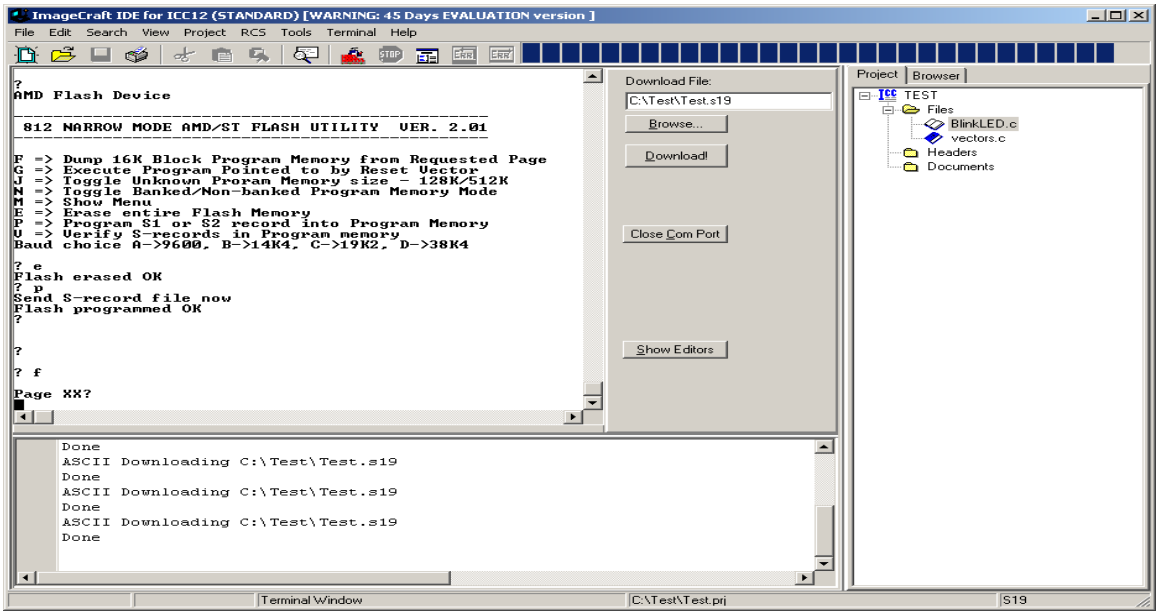
Checking DATA in Load mode:

The command is ***F*** for memory dumping DATA from FLASH. This is used to check certain parts of the memory.

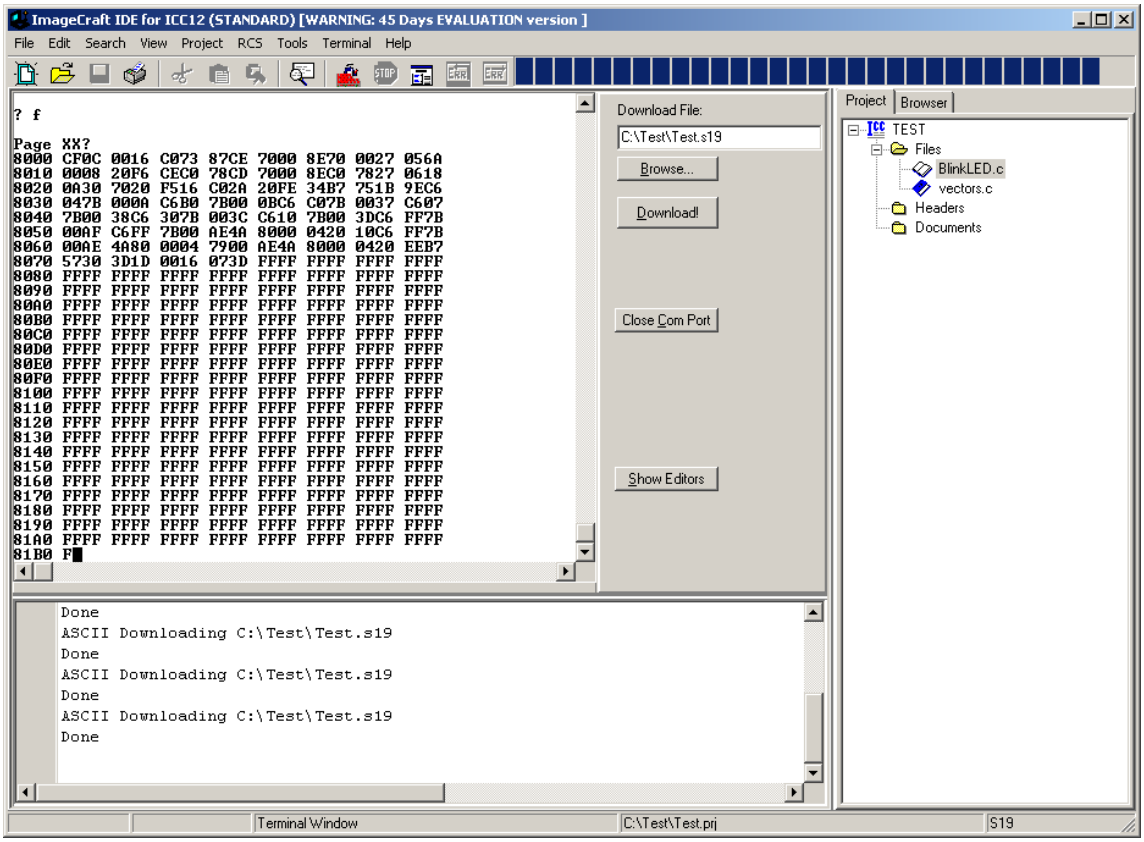
For example, one may want to check the address from $C000 to $FFFF. How is this possible? There are few things that needs to be understood. The memory block $C000 to $FFFF is always present in the memory map. This is where the ISR and start of code resides. When servicing an ISR it is possible to access a different PPAGE. With 128Kbyte Flash, the $C000 to $FFFF can be access at PPAGE = $07 in the page window $8000 to $BFFF.

For 512Kbyte Flash, the $C000 to $FFFF can be access at PPAGE = $1F in the page window $8000 to $BFFF.

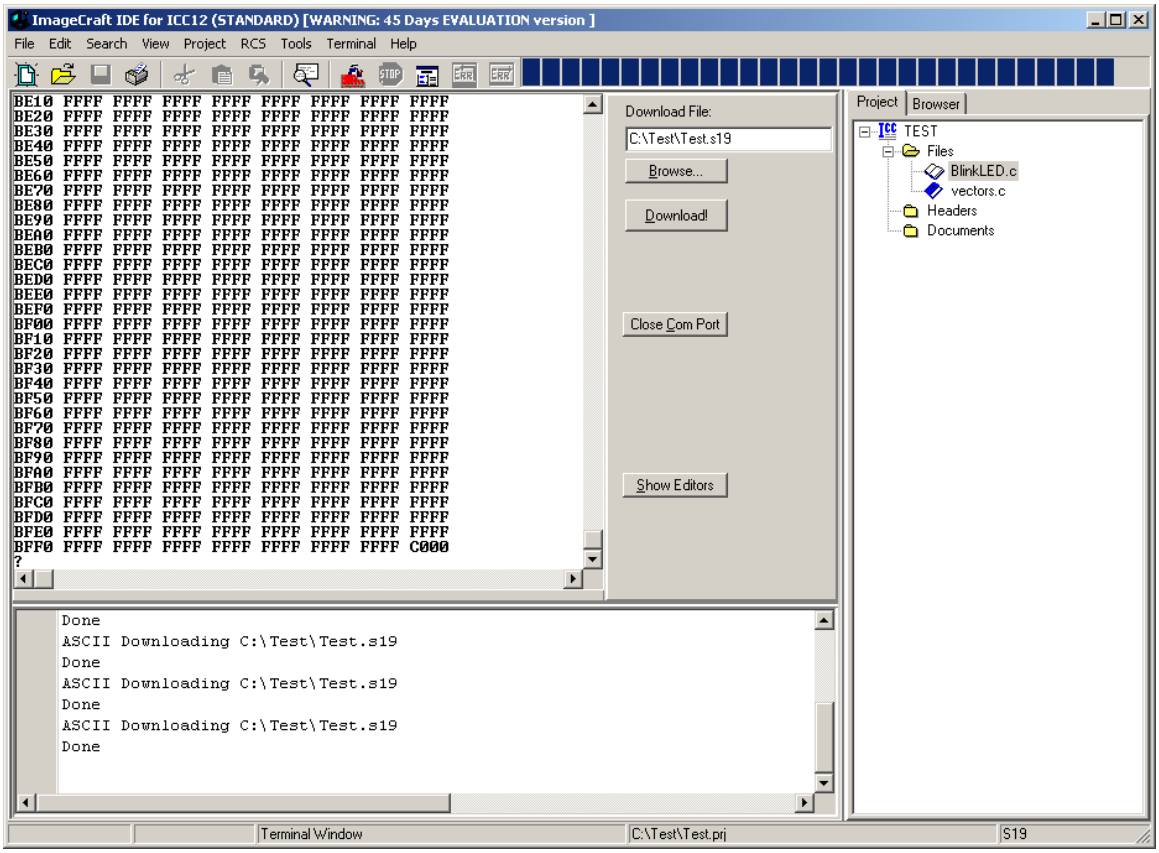
Here is an example to dump $C000 to $FFFF with a 128Kbyte Flash. Select ***F*** command then type the value ***07***



Immediately after it would begin to dump the DATA beginning at $8000 to $BFFF. Please note that PPAGE = $07 at $8000 to $BFFF is the same as $C000 to $FFFF



Below is at the end of DATA dump.



Note that at address $BFFE:BFFF is equal to $FFFE:$FFFF. The value is seen to be $C000 the start of code.

This tool is implemented to help diagnose where are the DATA being programmed too from an S-record file.

Executing Program in Load Mode:

The command is ***G*** to execute the already program code. The Loader will fetch the address at $BFFE:$BFFF (PPAGE set at the Last page) then start executing the code at that address. For example, if the value was $C000, the loader removes the EEPROM out of the memory map, read the vector address and make Register X equal $C000 and jump to $C000 with Register X.

Note:

In load mode certain registers are initialized. The stack is set to $0C00, MODA:MODB in expanded narrow mode including the various expanded register controls. Lastly the COP control is disabled.

Problems:

Q. Programmed okay but will not appear to run.

A. Toggle from BANKED to NON-BANKED or vice-versa then erase and re-program again.

Q. Tried to program in BANKED or NON-BANKED, Flash programmed OK but will not work.
A. There are 3 things to check.
1. COP control needs to be disable if not being serviced.
2. Check if the Stack is initialized
3. Check if the Power ON reset vector is correctly pointing to start of code.

Q. Programmed OK and runs with G command but not in Expanded mode.
A. Check the stack is initialized and COP control disabled if not being serviced. In load mode the stack is initialized and COP disabled. Make sure to slide SW3 to Expanded Narrow mode.

Q. During programming the red LED comes on
A. The Loader is unable to program into a memory location. The reasons are many. The S-record may contain code that are manipulating the registers. The S-record may contain addresses that are outside the bound of the FLASH chip. Lastly, the FLASH memory maybe faulty.

Q. Programming Rev3 is different from Rev2A.
A. Verify the jumper settings FLASH and RAM

JB3 – pin 1 and 2 shunted (CSD* RAM enable)
JB6 – pin 1 and 2 shunted (CSP0* FLASH select)
JB7 – pin 1 and 2 shunted (RAM write enable)