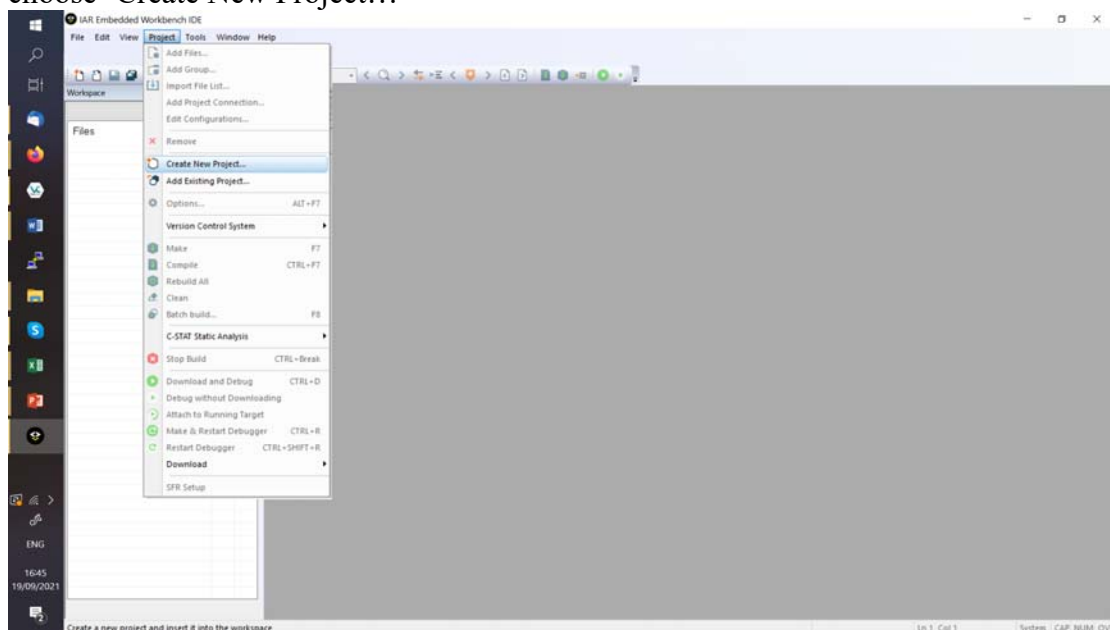


EE579 ADVANCED MICROCONTROLLER APPLICATIONS

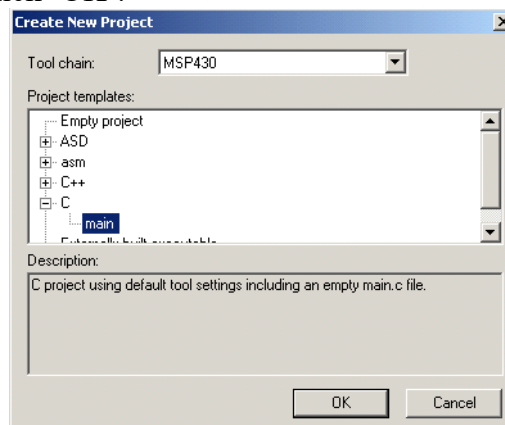
Introduction to the MSP430

In this introduction, we will use the IAR Embedded Workbench to develop a simple program using the IDE to flash an LED.

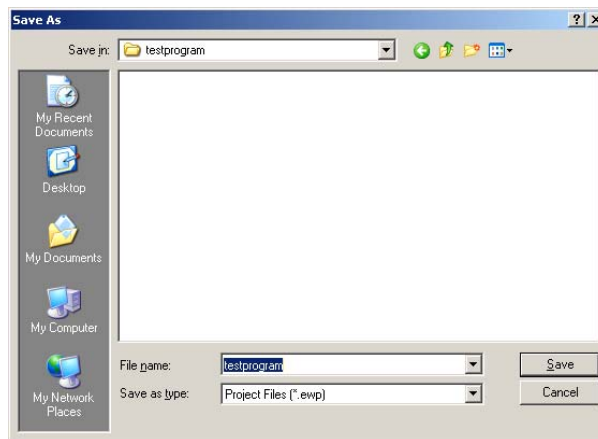
Open the IAR software. If you have never used the software you will see a welcome screen will come up with 'Information Center for MSP430'. Otherwise it will default to your last project. Close any open workspaces and then from the Project menu, choose 'Create New Project...'



When the 'Create New Project' dialog appears, click the little + next the 'C', and then select 'main'. Click 'OK'.

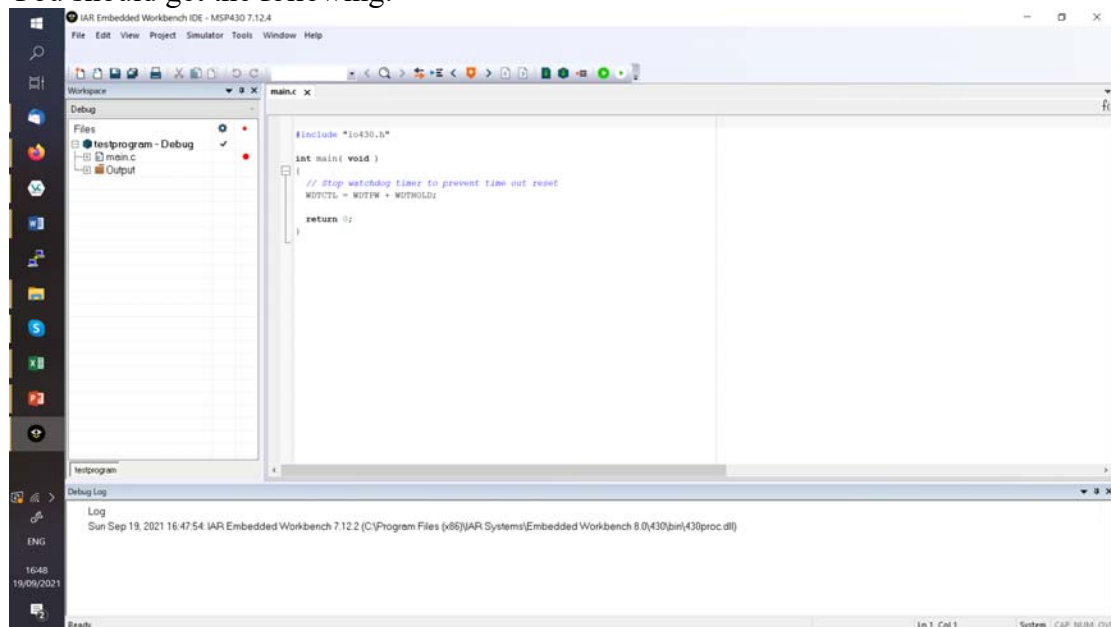


The 'Save as' dialog will appear. **Create a new directory** with the name 'testprogram', and then change to that. Then enter 'testprogram' as the name of the project.



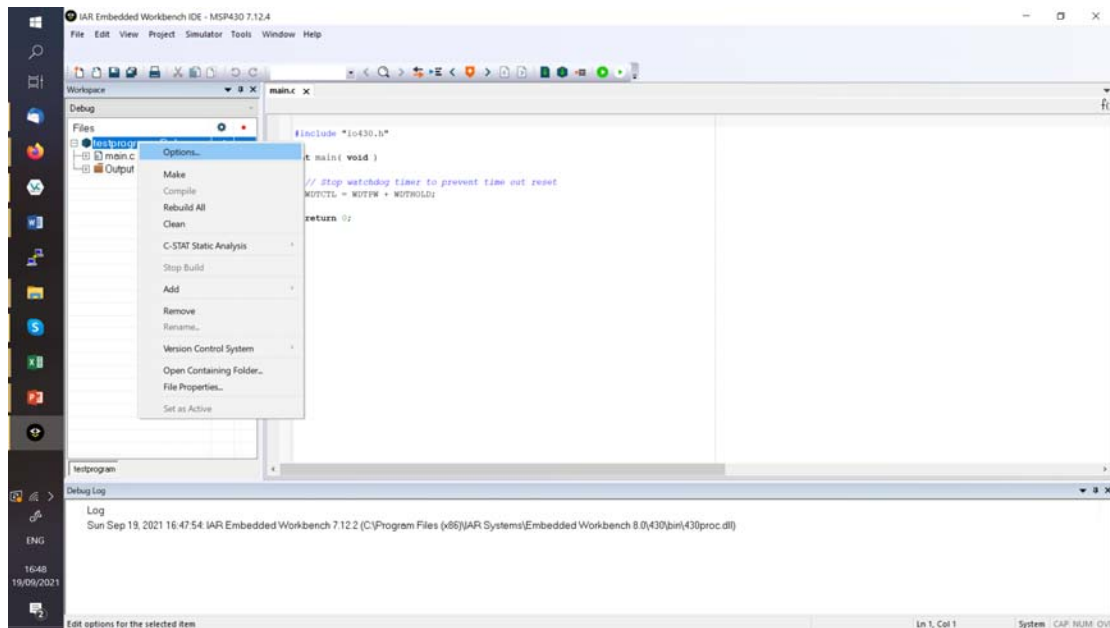
It is very important to create a new directory for each project, and store your programs in it. If you don't, you may overwrite the `main.c` file of your previous project because the IDE creates an empty C file for you each time you create a project.

You should get the following:

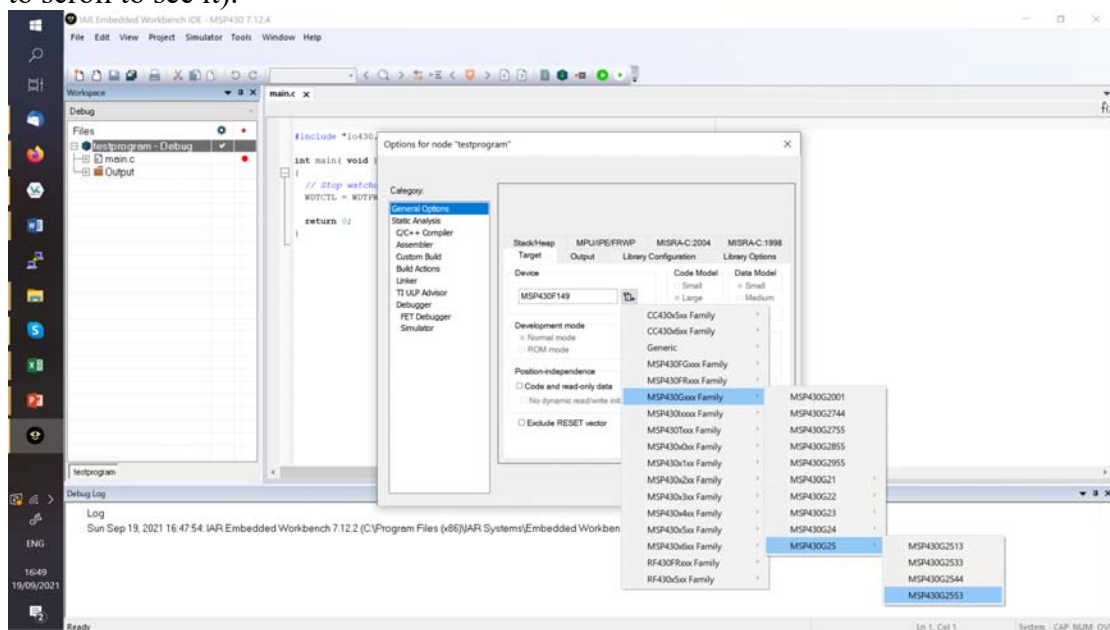


The IDE has created a skeleton C program. The consists only of a main function. The IDE has assumed no parameters will be supplied – hence the ‘(void)’ – but an integer value will be returned. The value is a fixed one, 0, returned by the line ‘return 0;’ which is the convention for programs which execute correctly. The line ‘#include “io430.h”’ tells the compiler the names of all the devices on the MSP430 chip.

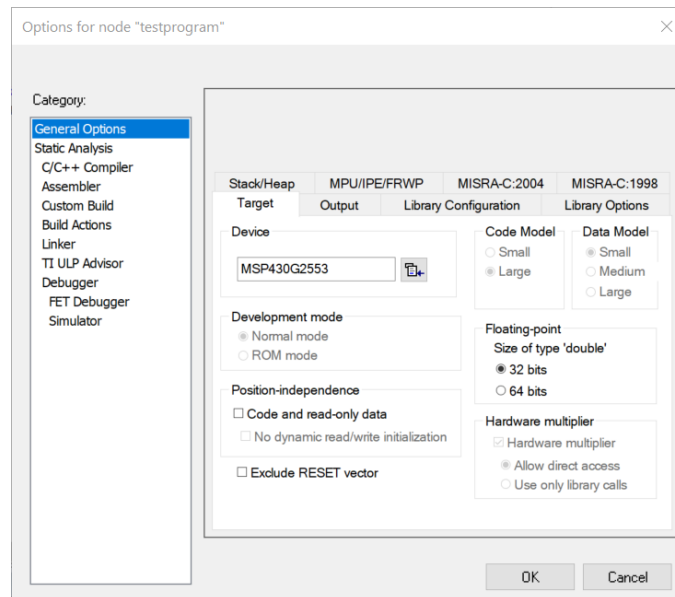
Right click on ‘testprogram’ in the Workspace window on the left hand side, and select ‘Options’.



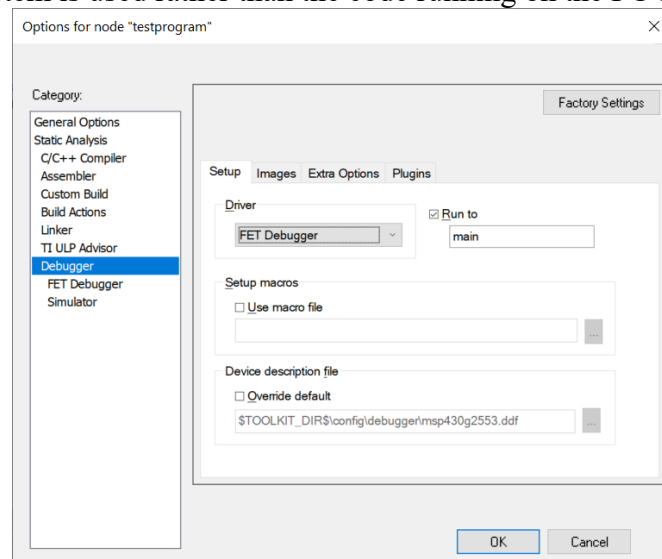
Under 'General options', click to the right of the 'Device' box and select 'MSP430Gxxx family' and then 'MSP430G2553' to select the correct chip for the target board (it is right at the bottom of the MSP430Gxxx family list so you may need to scroll to see it).



You should get the following:



Then select 'Debugger' and change 'Simulator' to 'FET Debugger', so that the development system is used rather than the code running on the PC alone.



Click 'OK'

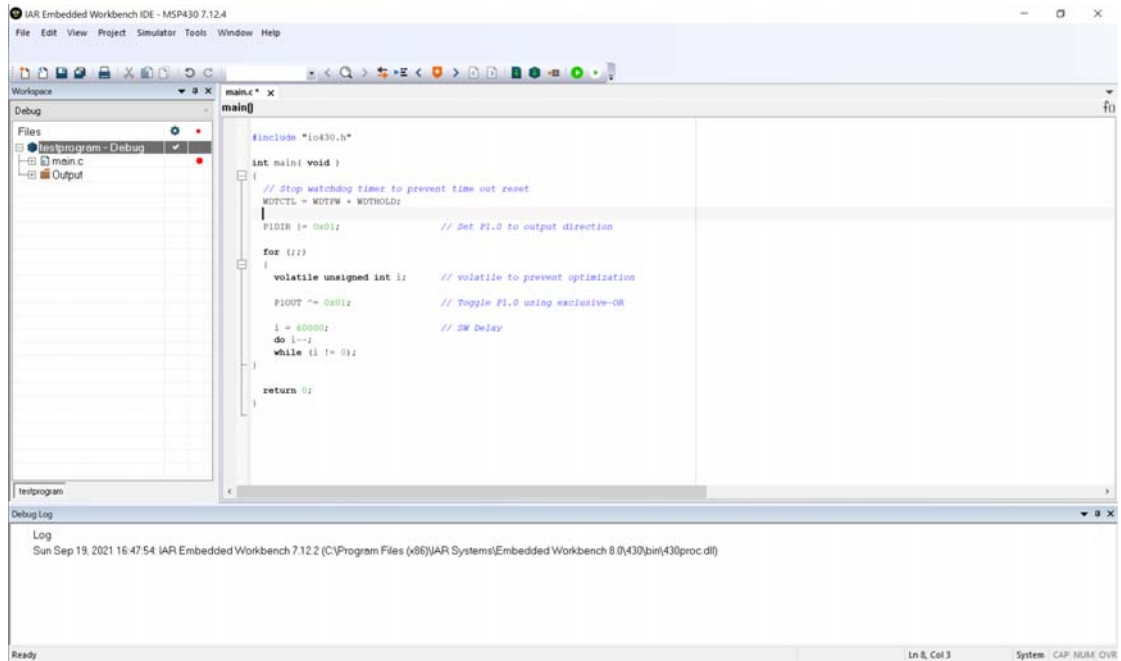
Add the following code to the program:


```
P1DIR |= 0x01; // Set P1.0 to output direction


for (;;)
{
    volatile unsigned int i; // volatile to prevent optimization


    P1OUT ^= 0x01; // Toggle P1.0 using exclusive-OR


    i = 60000; // SW Delay
    do i--;
    while (i != 0);
}
```




Save everything ( top left).

Hit build (). The system will complain about an unreachable statement, but ignore that. It has noticed that the loop will never end and so the code will never reach the 'return 0;' statement. This isn't a problem, but the compiler warns you to make sure that that is what you really wanted to do.

Hit make and start debugger ()

Run the code (). The red light (only) should flash.

Hit () to stop the code.