Embedded ARM Development by IAR workbench (1)

Part 1 | Part 2 | Part 3

Step by step
Last Edited: APR-2010
Compiler for ARM, ARM Programming Tutorials, ARM Guide, ARM Projects, ARM Led Blinking Program, ARM Debugger, How to use Jlink for ARM, IAR Workbench Guide

IAR workbench is very nice development tools for ARM. You can download their kickstart version from their website. Kickstart version is free to download and it comes with restriction of 32KB code size limit. Once you download then you can easily install it. This article describe project creating and debugging.

Before going through in details of working of IAR workbench, let me explain basics of ARM application project.
ARM application can be of two types:

1.ROM Run Application
2.RAM Run Application

ROM Run Application

Rom run application will be downloaded to flash memory and it will start executing from there only. The executable file in this case will be pure binary file i.e. this will not contain debugging information. Extension of this file can be either bin or hex. (User can configure project options to generate any type of file). In IAR workbench there is predefined setting of project options for generating output file suitable for ROM run application, name of this configuration is Release configuration.
In Release configuration code is optimized in order to generate smaller and fast executable output file. And Rom version of linker script file will be used. In this file text section of code is directed to load into ROM memory.

RAM Run Application

Ram run application will be downloaded to ram memory and it will start executing from there only. The executable file in this case will contain debugging information. So that debugging will be easy. Purpose of type of application is only debugging and this will be not be finally released file. These files are of ELF format.(**Executable and Linkable Format**)

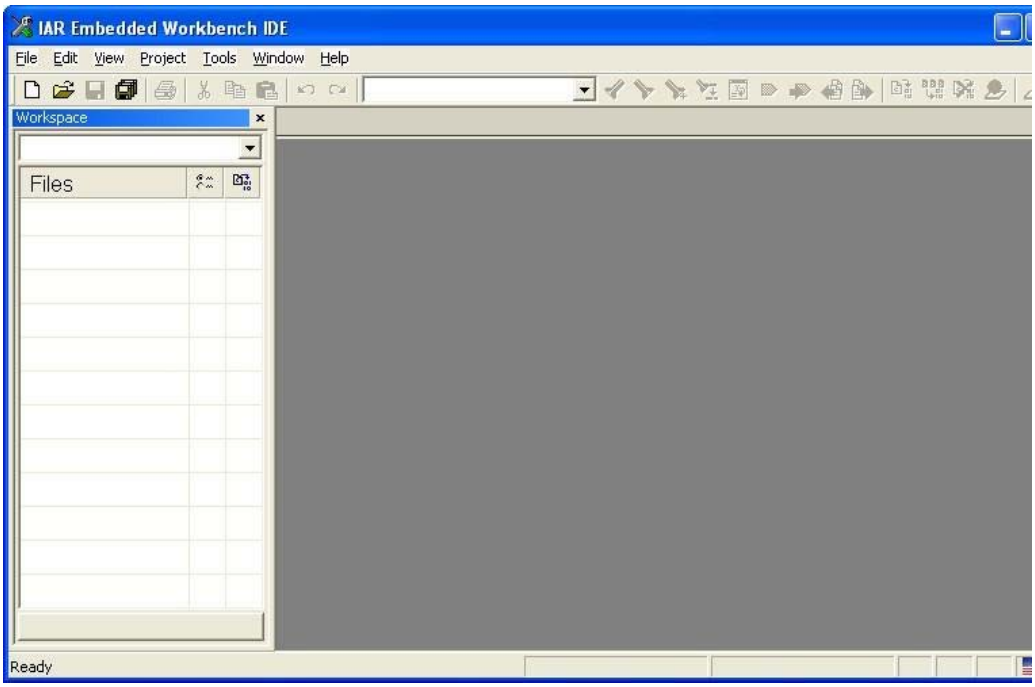
In general these files do not have any extension. In IAR workbench there is predefined setting of project options for generating elf output file le for RAM run application, name of this configuration is Debug configuration. And Ram version of linker script file will be used while building the project. In this file text section of code is directed to load into RAM memory.
Downloading of such a file need JTAG debugger or bootloader program.

We have created a sample project for blinking of LED. This application is written for Olimex LPC2148 Board. In this board two LEDs are connected at P0.10 and P0.11. You can easily modify code according to your LPC2148 board. This code should work for any LPC2xxx processor.

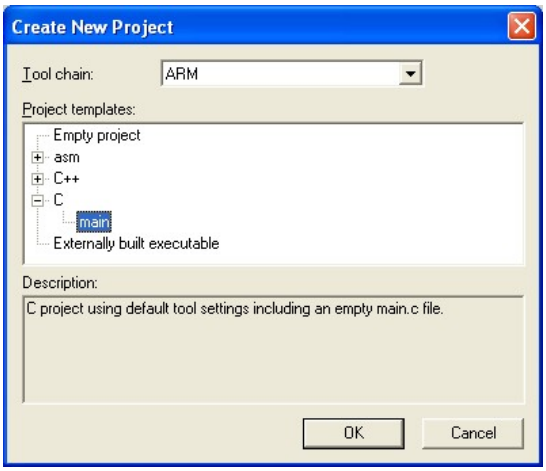
You can download from following location
Click Here
todo give the location of code.

Creating Project

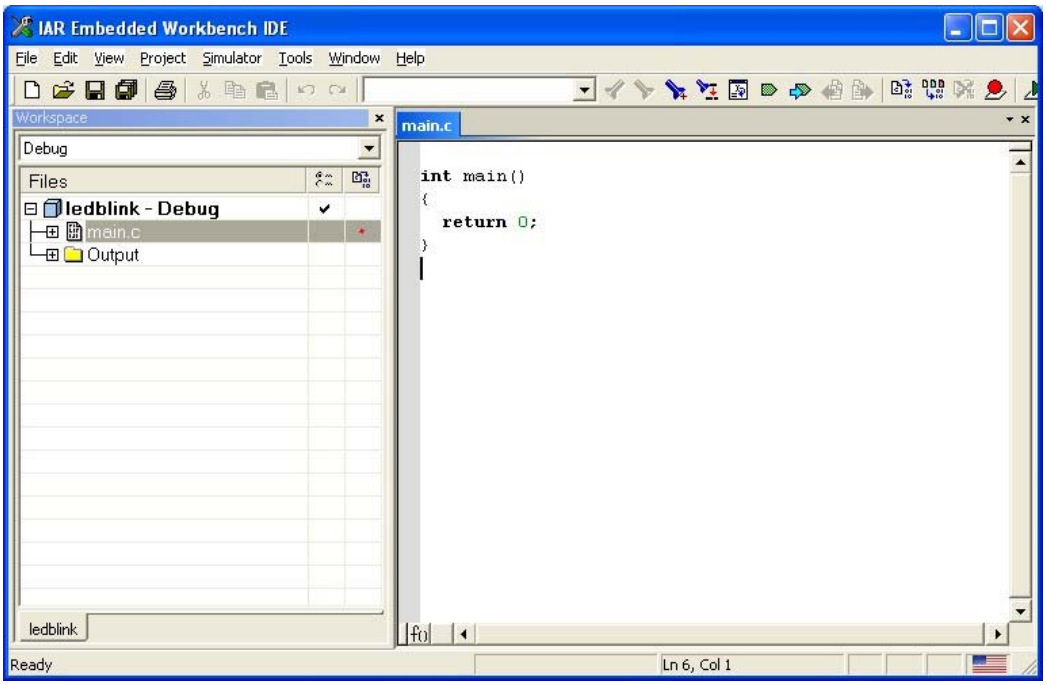
After installing IAR workbench open IAR workbench.
For that goto
Start > All Programs > IAR Systems > IAR Embedded Workbench for ARM 5.30 Kickstart > IAR Embedded Workbench



Go to Project > Create New Project

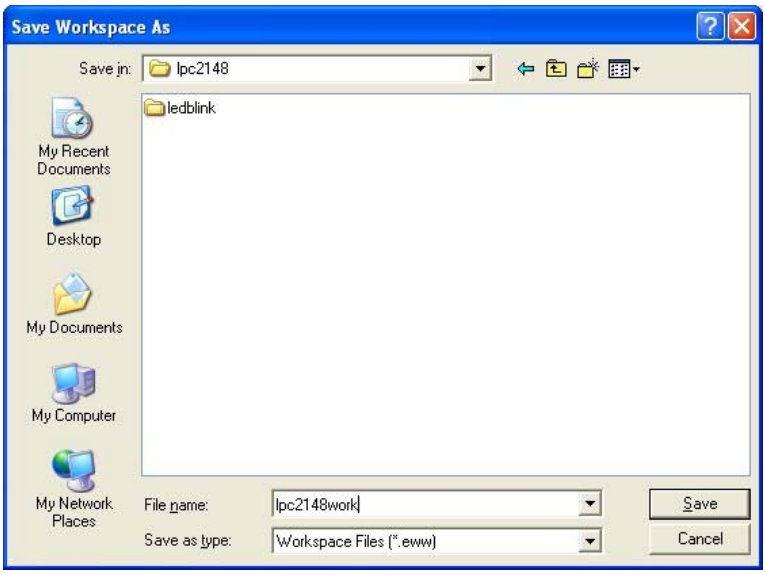


Here we will choose C main project and click Ok. After that you will see option to save project file. My preferred location is D:\lpc2148\ledblink and project name I have given ledblink. IAR project file extension name is ewp. so my project file is ledblink.ewp.



After saving project let us save workspace. workspace is a group of projects. For example, in case of lpc2148 board I can create more then one projects, so my all these project should goes under lpc2148 workspace. So its up to user how he manage project files.

I have selected workspace name as lpc2148work and its location will be in d:\lpc2148. For saving project workspace goto File > Save Workspace
and give lpc2148work name to workspace file.



Extension to workspace file is eww. now click on save button and save it.
Now let us add source code in main file. We are writing code for Led blinking.

```
#include "iolpc2148.h"
#define LED0 0x0400
#define LED1 0x0800
#include "main.h"

int main()
{
  /**** Initialization ****/
  PLLInit();
  SCS =0x3;
  /* make led line as output*/
  FIO0DIR |= LED0 | LED1;
  FIO0SET = LED0 | LED1;
  while(1)
    {
/* turn off LED*/
FIO0SET = LED0 | LED1  ;
delay();
/* turn on LED*/
```

```
        FIO0CLR = LED0 | LED1 ;
delay();
    }

}
```

keep main.h and iolpc2148.h files in your project directory.
There are two configuration for executable file, ram run or ram run as discussed in first paragraph of this article. Let us discuss building of RAM Run application first.
For RAM run application we have to add LPC2148_RAM.icf linker script file to the project.

Click here for Part2 of the article.

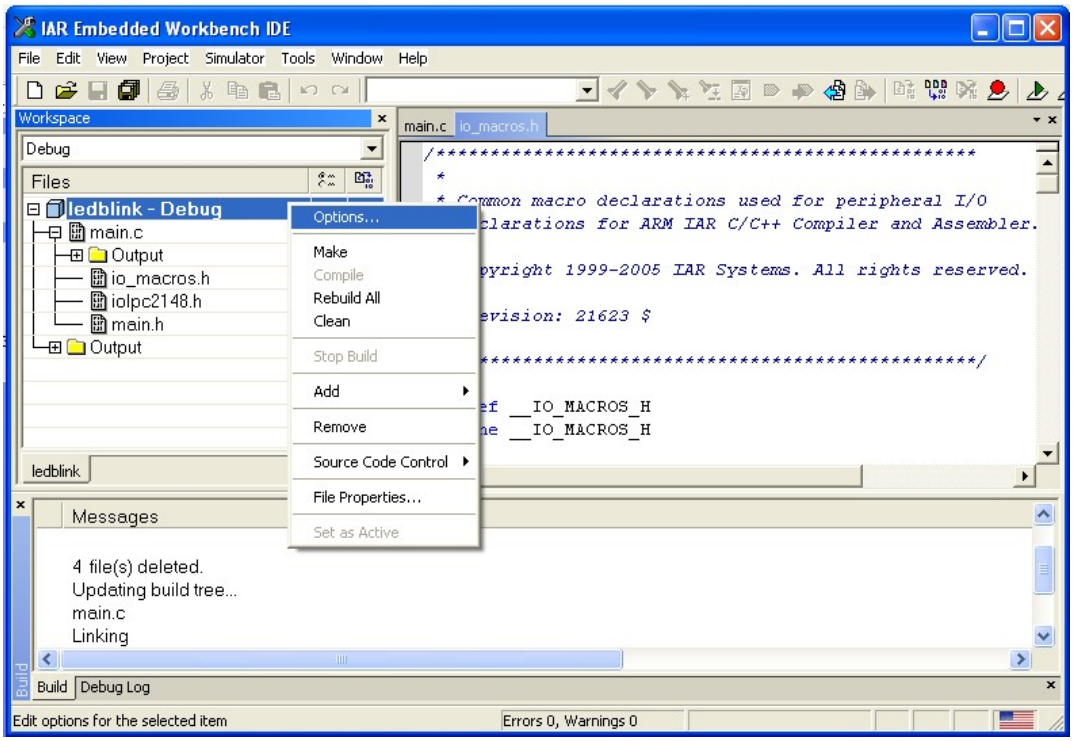
Part 1 | **Part 2** | **Part 3**

Embedded ARM Development by IAR workbench (2)

Part 1 | Part 2 | Part 3

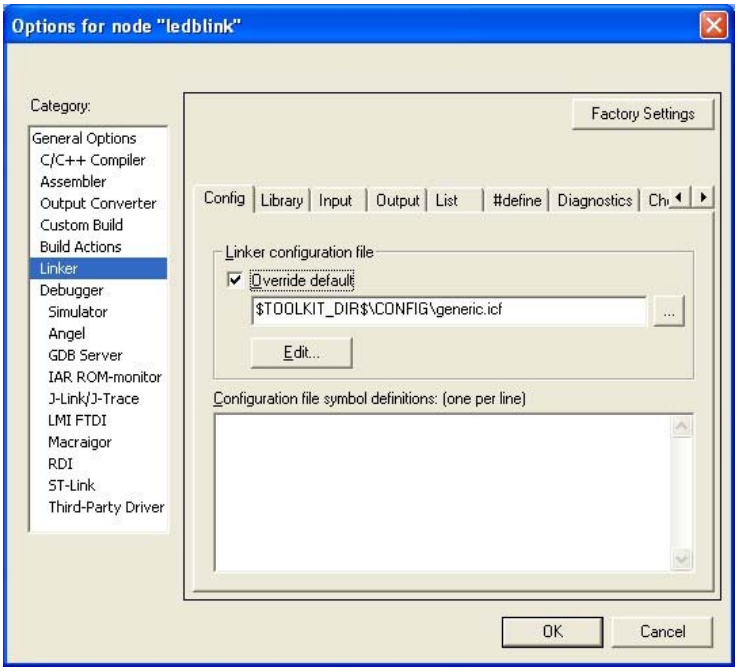
Step by step
Last Edited: APR-2010
Compiler for ARM, ARM Programming Tutorials, ARM Guide, ARM Projects, ARM Led Blinking Program, ARM Debugger, How to use Jlink for ARM, IAR Workbench Guide

For adding linker script file right click on project options and choose options as shown in above picture. or choose Project > options or use (ALT + F7).

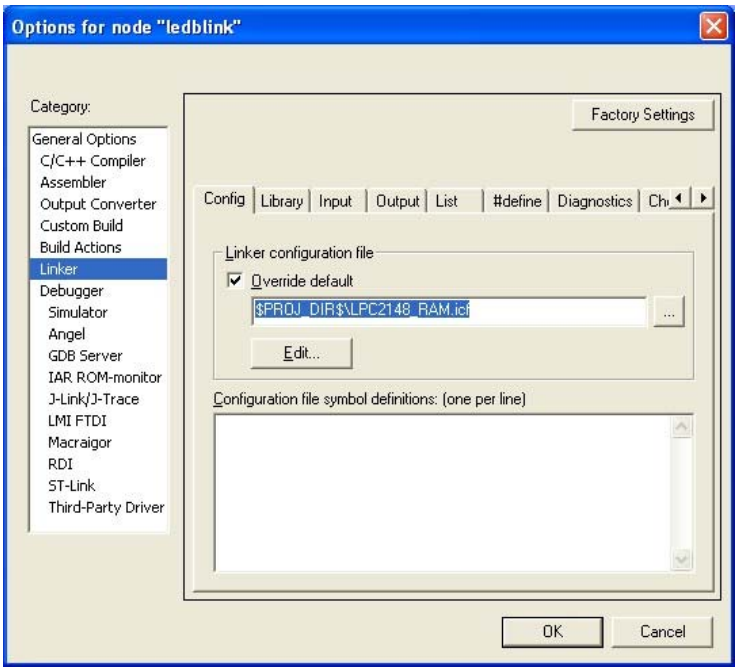


After that options for node “ledblink” window will open. Here goto Linker > Config > Linker configuration file override default and give path of LPC2148_RAM.icf file. For specifying path you can use IAR global variable $PROJ_DIR$ to specify path of project directory. So according to that complete path will be following

$PROJ_DIR$\LPC2148_RAM.icf

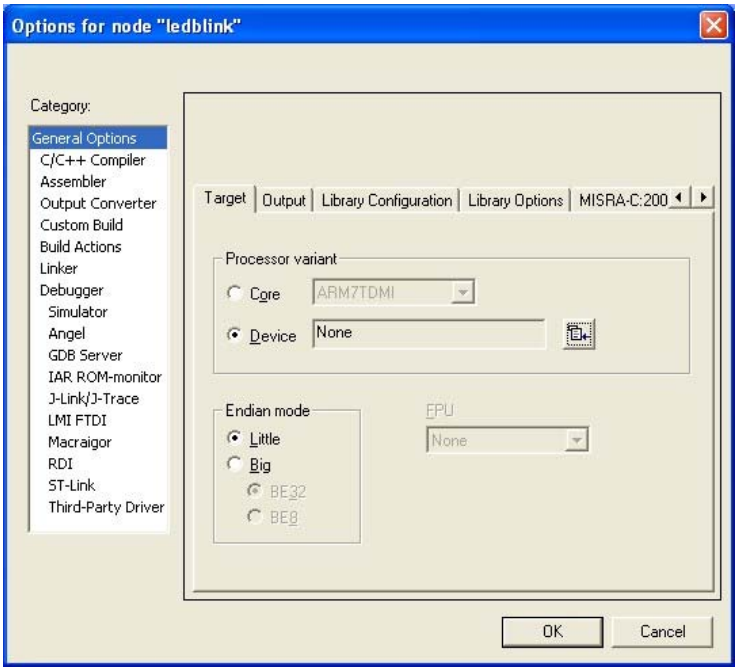


after setting of linker script file this window should be like following:

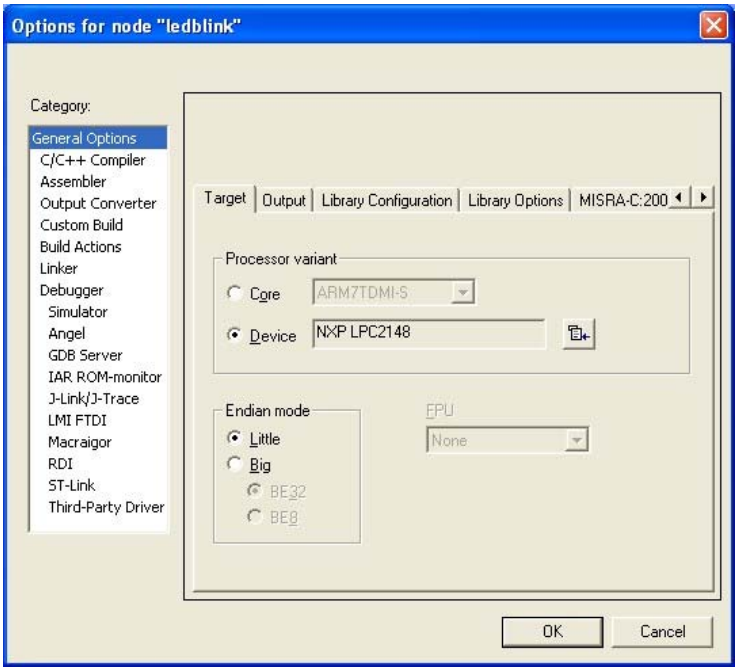


After that we have to choose project options which are following:

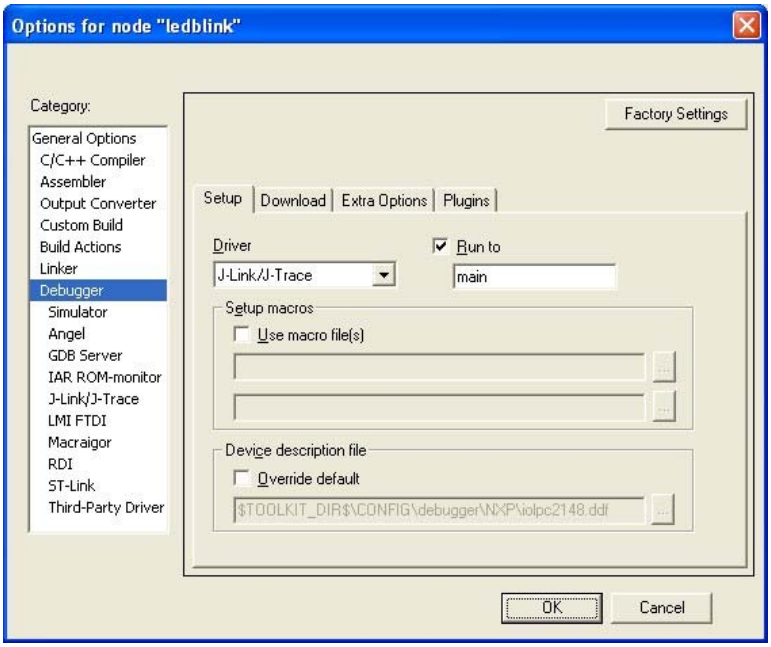
Processor options
Project > options > General options > Target > Processor variant > Device
click on button next to Device text box



and choose NXP > NXP LPC2148.

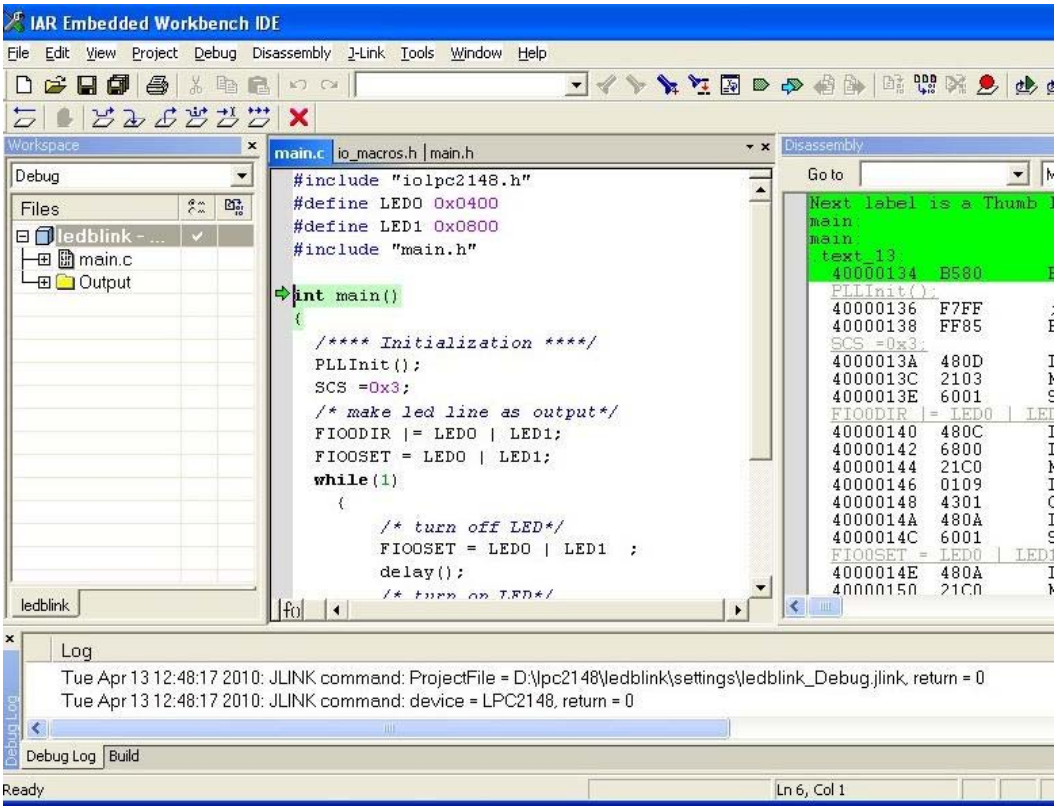


Now choose j-Link as a debugger. For that go to
Project > options > Debugger > Setup > Driver
select J-Link/J-Trace from drop down menu. Left all options as it is. and click on OK.



now project is ready to build.

To build project pres F7.
and Start debugger. For that go to Project > Download and Debug or press (Ctrl + D).
Code will start downloading on target. Once downloading is completed then you can see debugger view of IAR workbench. Now you can see that code is stopped at main.



Click here for Part3 of the article.

Part 1 | **Part 2** | **Part 3**

Receive Regular Updates
Want to receive regular updated of Tutorials published on this site, Register for free subscription of cost Infoletter
Click Here for registration.

TOP

Embedded ARM Development by IAR workbench (3)

Part 1 | Part 2 | Part 3

Step by step
Last Edited: APR-2010
Compiler for ARM, ARM Programming Tutorials, ARM Guide, ARM Projects, ARM Led Blinking Program, ARM Debugger, How to use Jlink for ARM, IAR Workbench Guide

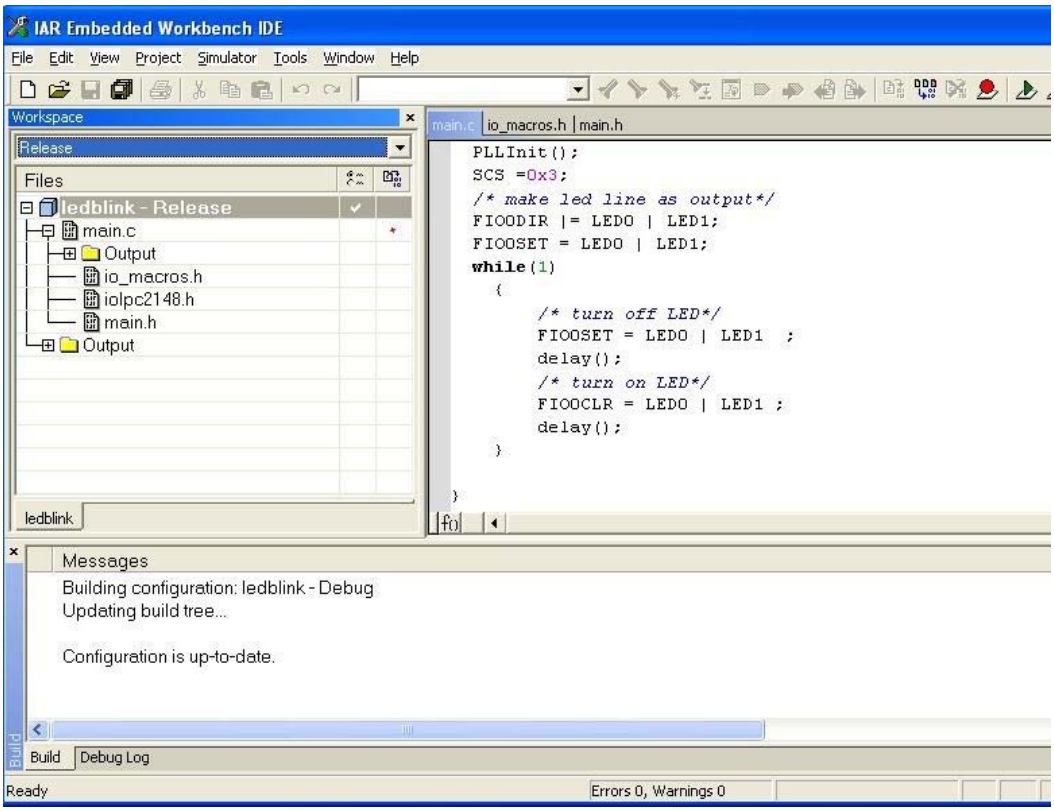
We can do single stepping . Some of the useful commands are following:

Step over F10
Step into F11
Step out Shift + F11
Go F5

Press F5 and you can see LED must be blinking on Board.

ROM run program or Flash run program

For making program to execute from run IAR has Release configuration. For that choose Release from drop down button below the Workspace window.



For release configuration we have to make following changes

1.Optimization

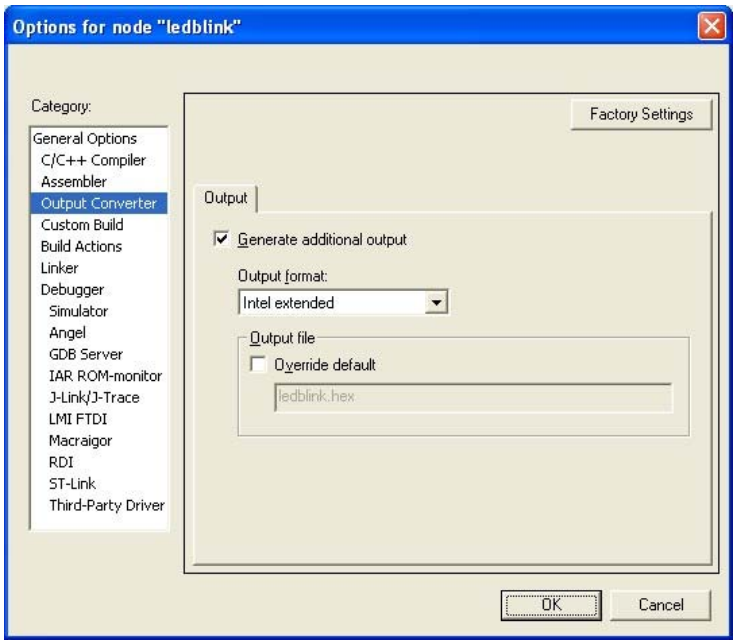
choose none optimization.
Project > options > C/C++ Compiler > Optimization > Level

choose none

2.Linker script file for flash
Project > options > Linker > Config > Linker configuration file
choose LPC2148_flash.icf file
$PROJ_DIR$\LPC2148_flash.icf

3.Generate HEX file

Project > options > Output Converter > output
check on Generate additional output
choose intel extended



now build the project and you can see that hex file is generated in Release folder of project directory.

This file can be downloaded into flash memory by flash magic software.

Part 1 | **Part 2** | **Part 3**