

NXP LPC1114 Starter Kit User Guide

Rev. 1.0

Release: 2010-02-01



EMBEST CO., LIMITED

Address: Room 509, Luohu Sci&Tech Edifice, No.85, Taining Road,
Luohu District, Shenzhen, 518020, China

Telephone: +86-755-25621715

Fax: +86-755-25616057

E-mail: sales.en@embedinfo.com

Support: support.en@embedinfo.com

Website: <http://www.embedinfo.com/en>

Revision history

Rev	Date	Description
1.0	20100201	Initial version

Catalog

CHAPTER 1 OVERVIEW.....	5
1.1 The Microcontroller Introduction.....	5
1.2 Evaluation Boards Introduction.....	5
1.3 Hardware resources list.....	5
1.4 Software resources list.....	5
CHAPTER 2 GETTING STARTED.....	7
2.1 Product detailed list check.....	7
2.2 Documents Description.....	7
2.3 Version Information.....	7
2.4 Hardware resource requirements.....	7
2.5 Preparations.....	7
2.6 Use the default factory routine.....	8
2.7 Restore the default factory routine.....	9
2.7.1 Compile the default factory routine.....	9
2.7.2 Download the default factory routine.....	10
CHAPTER 3 NXP LPC1114 STARTER KIT INTRODUCTION.....	11
3.1 Board Interface Overview.....	11
3.2 Jumpers Settings.....	11
3.3 Hardware Interface Introduction.....	12
3.3.1 JTAG.....	12
3.3.2 UART.....	12
3.3.3 Mini USB Port.....	12
3.3.4 EEPROM.....	12
3.3.5 LM75 Temperature Sensor.....	12
3.3.6 LED.....	12
CHAPTER 4 SOFTWARE RESOURCES INTRODUCTION.....	13
4.1 LED Blinky test.....	13
4.2 GPIO test.....	13
4.3 PMU test.....	13
4.4 SysTick test.....	14
4.5 Timer32 test.....	14
4.6 WDT test.....	14
4.7 SSP test.....	15
4.8 UART test.....	15
4.9 I2C EEPROM test.....	15
4.10 I2C Temperature Sensor test.....	16
CHAPTER 5 SOFTWARE DEVELOPMENT AND EXAMPLES.....	17
5.1 MDK Introduction.....	17

5.2 Compile routine..... 17

 5.2.1 *Open a routine*..... 17

 5.2.2 *Compile the routine* 17

5.3 Debug and Download the routine using Emulator 19

 5.3.1 *Debug and Download the routine using ULINK2* 19

 5.3.2 *Download and Debug the routine using Colink2* 25

5.4 Download the routine using ISP..... 30

APPENDIX A: AFTER-SALE SERVICE..... 36

Chapter 1 Overview

1.1 The Microcontroller Introduction

The LPC11xx are ARM Cortex-M0 based microcontrollers (designed for 8/16-bit microcontroller applications) for embedded applications featuring a high level of integration and low power consumption. The LPC11xx operate at CPU frequencies of up to 50 MHz. The peripheral complement of the LPC11xx series includes up to 32 KB of flash memory, up to 8 KB of data memory, one Fast-mode Plus I2C-bus interface, one RS-485/EIA-485 UART, up to two SPI interfaces with SSP features, four general purpose timers, a 10-bit ADC, and up to 42 general purpose I/O pins.

The LPC11xx operates in the -65 to $+150$ °C temperature range, from a 1.8 to 3.6 V power supply. A comprehensive set of power-saving mode allows the design of low-power applications. It includes:

- eMetering
- Lighting
- Industrial networking
- Alarm systems
- White goods

1.2 Evaluation Boards Introduction

The LPC1114 Starter Kit is a brand new, cost-effective but high-performance evaluation tool of the ARM Cortex-M0 based LPC11xx controller family from NXP, allowing you to create and test working programs for this advanced architecture. The board has a wide range of interfaces making it a great starting point for your next Cortex-M0 project.

You can use the Cortex-M0 based LPC1114 KIT to generate and test application programs for the NXP LPC11xx microcontroller family. With this hands-on process, you can determine the hardware and software requirements for current and future product development.

1.3 Hardware resources list

- LPC1114x301 ARM 32-bit Cortex™-M0 CPU, 50 MHz maximum frequency
- 8 LED light tube, a power supply LED
- A mini-type USB socket, just power supply
- A BOOT button
- A RESET button
- SSP interface
- An RS232 connection socket (DB9) (support RS-485/EIA-485)
- I2C interface
- I2C interface connects to external EEPROM
- 10-bit ADC Module
- A JTAG/SWD debug interface
- Power Supply: USB powered

1.4 Software resources list

Project Name	Function Description
Blinky	Use 16-bits timer to achieve LED1 blinky.
GPIO	Use GPIO port to achieve input event interrupt. GPIO ports set as input






	event, single edge trigger, active high.
PMU	Configure PMU to put some of the peripheral in sleep mode.
SysTick	Configure the SysTick to generate a time base equal to 1 ms. According systick delay to achieve LED toggle.
Timer32	Use 32-bits timer to achieve LED1 blinky delay.
WDT	WatchDog application example.
SSP	SSP LOOPBACK mode test.
UART	Use UART to send and receive data through HyperTerminal.
I2C_EEPROM	I2C EEPROM Read/write test
TemperatureSensor	I2C Temperature sensor test

Chapter 2 Getting Started

2.1 Product detailed list check

1. One LPC1114 Starter Kit
2. One crosslink serial cable
3. One USB cable
4. A product CD-ROM

2.2 Documents Description

File name / Item	Description	Attribute
LPC1114 KIT User Manual V1.0.pdf	User's manual	 671KB
LPC1114 Starter Kit Schematic.pdf	Main board schematics.	 49KB
EM-LPC11xx Datasheet.pdf	Datasheet of LPC11xx	 314KB
EM-LPC11xx Reference Manual.pdf	Reference Manual of LPC11xx	 1.93MB
Other pdf format files of datasheet	Datasheet of other components of the board such as EEPROM ,RS232	 992KB

2.3 Version Information

- The version of the development tools: MDK4.10

2.4 Hardware resource requirements

When we test LPC1114 KIT, PC recommended the following configuration:

- 2.0GHz (or higher) of the CPU
- 512M RAM
- 2 USB interfaces
- A serial interface
- Windows XP operating system
- KEIL Integrated Development Environment installed

2.5 Preparations

- Jumper Settings: JP2 keep open.
- Serial Connection: Connect com of board and the COM of PC through serial port cable.
- USB Connection: Using USB cable, one end plugged into the Mini USB port on the board, the other end connected to PC.

- JTAG Debugger Connection: One end connected to JTAG interface on the board, the other end connected to PC.
- Serial Port Receive Settings: In the PC, run HyperTerminal serial communication program, select the serial port used and set the following parameters (to set status: Baud rate (115200), data bits (8 bits), stop bits (1 bit), parity bit (no), data flow control (no)).

2.6 Use the default factory routine

- Set environment according to section 2.5, power on the development board and press 'reset' button.
- At this time, the HyperTerminal may display the information about the function of this board, which needs to be tested. See it as below. Every item of board's functions has a number. They are sorted according the number from digital '0' to '4'. You need to type one number of them so as to test the corresponding function using PC keyboard after resetting the board.
- Note: after testing one function of the board, you need to reset the board by pressing the 'RESET' button of the board. It may redisplay the tip information, so you can choose the other one so as to test the other function.

Now we illustrate how to test every function of the board and the corresponding test phenomenon based on the number from zero to four.

```

*****
*
* Welcome to Embest
*
* This is EM-LPC1100 test
*
* arm.embedinfo.com
*
* 0-LED test      1-BUTTON test      2-UART test
*
* 3-I2C EEPROM test    4-I2C Temperature Sensor test
*
*****

Please input number from keyboard
    
```

- **0 option: test LED also test the function of Timer16 of the board.**

- 1) After typing '0' in the terminal using PC keyboard, the LED1(D2) starts to blinky. Time delay between two flashes is controlled by timer16.
- 2) Then, the brief information about this function test will be displayed in the terminal.

```

-- Basic Blinky Project V1.0 --

-- EM-LPC1100 --

-- LED1 blinky test --
    
```

- **1 option: test the BOOT button onthe board.**

- 1) After typing '1',it displays the brief information about this test in the terminal.

```

-- Basic GPIO Project V1.0 --

-- EM-LPC1100 --

-- GPIO port single edge trigger interrupt test --

-- Please press 'BOOT' button on the board! --
    
```

- 2) Pressing 'BOOT' button(K1) on the board can triggers one interrupt. Its interrupt service routine displays

one short message ‘PIOINT0_IRQHandler!’ in the terminal. See it as below.

```
PIOINT0_IRQHandler!

PIOINT0_IRQHandler!

PIOINT0_IRQHandler!
```

- **2 option: UART sends and receives data test.**

1) After typing ‘2’, it displays the brief information about this test in the terminal.

```
-- Basic UART Project V1.0 --

-- EM-LPC1100 --

-- Please input any key on the keyboard --
```

2) According tip message, you can type in any non-control character using PC keyboard. The corresponding letter can be displayed in the terminal. For example as below.

```
.fdf2eqf314jkj51455254/*-/4516445561...
```

- **3 option: I2C EEPROM test.**

1) After typing ‘3’, it displays the brief information about this test in the terminal.

```
-- Basic I2C EEPROM Project V1.0 --

-- EM-LPC1100 --

-- I2C Read/Write EEPROM test --

I2C EEPROM test success!
```

2) This testcase configures I2C working in master mode, and Read/Write EEPROM using I2C bus. Init buffer, send the buffer data to EEPROM, read them and compare. If the data read is the same as data written, then test success. It displays message “I2C EEPROM test success!” in the terminal.

- **4 option: I2C Temperature Sensor test.**

1) After typing ‘4’, it displays the brief information about this test in the terminal.

```
-- Basic I2C Temperature Sensor Project V1.0 --

-- EM-LPC1100 --

-- I2C Temperature Sensor test --

-I- Temperature Register value: 0x1380

-I- Configuration Register value: 0x0

-I- Tos Set Point Register value: 0x80

-I- Thyst Set Point Register value: 0x1380

I2C Temperature Sensor test success!
```

2) This project tests temperature sensor on the board using I2C bus, and read internal four registers. They are Temperature Register, Configuration Register, Tos Set Point Register and Thyst Set Point Register. If display message “I2C Temperature Sensor test success!”, then test success.

2.7 Restore the default factory routine

2.7.1 Compile the default factory routine

Compile the default factory routine according to the steps stated in the section 5.2.

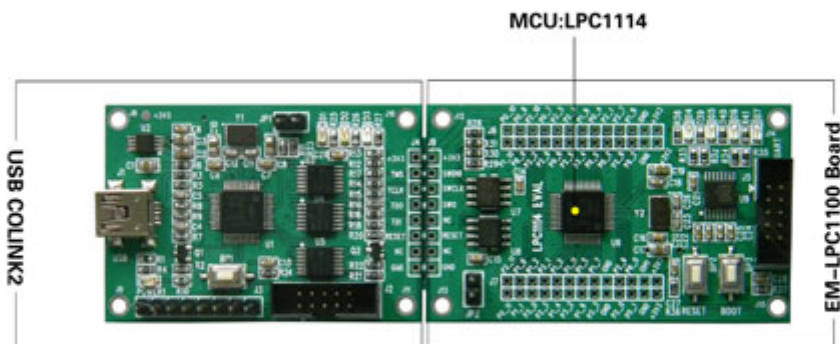
If you don't want to recompile the default factory routine, you can also use the image file All_LPC1100_test.hex or All_LPC1100_test.bin in the directory 02-Images of CD-ROM and download it to your board.

2.7.2 Download the default factory routine

- If you use ULINK2 emulator, refer to 5.3.1 section and download.
- If you use Colink2 integrated on the LPC1100 board, refer to 5.3.2 section and download.
- If you do not have any emulator and don't use Colink2, refer to 5.4 section and download.

Chapter 3 NXP LPC1114 Starter Kit Introduction

3.1 Board Interface Overview



J6	Processor External Round Cable	BP3	BOOT Button
J7	Processor External Round Cable	BP2	Reset Button
J8	JTAG	U6	LPC1114FBD48
D4	LED1	U7	Temperature Sensor
D5	LED2	U8	64KBit Serial I ² C Bus EEPROM
D6	LED3	U9	RS232 Transceiver
D7	LED4		

3.2 Jumpers Settings

ID	Name	Default Settings	Note
JP2	ISP	Open	Allow user to download the routine using ISP software when this jumper is closed.

3.3 Hardware Interface Introduction

3.3.1 JTAG

A standard 20-pin JTAG connector is implemented on the LPC1114 KIT for any ARM JTAG Emulator, such as ULINIK2, JLink, CoLink...

3.3.2 UART

The Universal Asynchronous Receiver Transmitter features a 9-pin UART that can be used for communication and trace purposes. It offers an ideal channel for ISP downloading.

3.3.3 Mini USB Port

A Mini USB AB interface, just supplies 5V voltage to the Board.

3.3.4 EEPROM

A 64KBit EEPROM is connected to the I2C bus in LPC1114 Starter Kit.

3.3.5 LM75 Temperature Sensor

A LM75 Temperature Sensor is connected to the I2C bus in LPC1114 Starter Kit.

3.3.6 LED

LPC1114 KIT provides 4 LEDs D4...7, they respectively connect with PIO2.0...3 IO pins, and they can be used for user output.

Chapter 4 Software Resources Introduction

There are 10 test cases in the CD-ROM provided. The directory is 03-software\examples. You can find 11 folders in this directory. The Common folder contains common source files and head files. The other 10 items are all test cases used to test the peripherals or functions of the LPC1114 Starter Kit.

Test cases' functions, most of those are easy to understand. You can get the details about testing introductions and phenomenon from the readme.txt files in the projects or get them by view the LPC1114 KIT User Manual V1.0.doc documents.

4.1 LED Blinky test

- Source code location: 03-Software\Examples\01-Blinky
- Test description: This example describes how to use 16-bits timer to achieve LED1 blinky.
- Test phenomenon: Build the program and download it inside the evaluation board. Push RESET button, the LED1 of the board will blinky.
- Reference manual: EM-LPC11xx Reference Manual.pdf

4.2 GPIO test

- Source code location: 03-Software\Examples\02-GPIO
- Test description: This example describes how to use GPIO port to achieve input event interrupt. GPIO ports set as as input event, single edge trigger, active high.
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic GPIO Project V1.0 --
```

```
-- EM-LPC1100 --
```

```
-- GPIO port single edge trigger interrupt test --
```

Press 'BOOT' button on the board may trigger a interrupt. It displays message " PPOINT0_IRQHandler!" in the terminal.

- Reference manual: EM-LPC11xx Reference Manual.pdf

4.3 PMU test

- Source code location: 03-Software\Examples\03-PMU
- Test description: This example describes how to configure PMU to put some of the peripheral in sleep mode. It also demonstrates how to setup wakeup source before sleep or deep sleep.
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic PMU Project V1.0 --
```

```
-- EM-LPC1100 --
```

```
-- PMU sleep or deep sleep mode test --
```

Press 'BOOT' to wakeup system!

After reset, system pLL, Watchdog oscillator, ADC and BOD will power-down. You can push "BOOT" button on the board to wakeup. After wakeup, 8 LEDs will blinky.

- Reference manual: EM-LPC11xx Reference Manual.pdf

4.4 SysTick test

- Source code location: 03-software\Examples\04-SysTick
- Test description: This example shows how to configure the SysTick to generate a time base equal to 1 ms. According systick delay to achieve LED toggle.
- Test phenomenon: Build the program and download it inside the evaluation board. Push RESET button, LED1 on the board start to blinky immediately after reset.
- Reference manual: EM-LPC11xx Reference Manual.pdf

4.5 Timer32 test

- Source code location: 03-software\Examples\05-Timer32
- Test description: This example describes how to use 32-bits timer to achieve LED1 blinky delay.
- Test phenomenon: Build the program and download it inside the evaluation board. Push RESET button, LED1 on the board start to blinky immediately after reset.
- Reference manual: EM-LPC11xx Reference Manual.pdf

4.6 WDT test

- Source code location: 03-Software\Examples\06-WDT
- Test description: This example describes how to configure and use WDT. The Watchdog will generate a system reset if the user program fails to "feed" (or reload) the Watchdog within a predetermined amount of time. Writing 0xAA followed by 0x55 to WDFEED(Watchdog feed sequence register) reloads the Watchdog timer with the value contained in WDTC(Watchdog timer constant register).
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic WDT Project V1.0 --
-- EM-LPC1100 --
-- Feed watchdog timer to prevent it from timeout test --
```

LED1 starts blinky after reset. Press BOOT button, WatchDog can't be feeded, so it resets system. After system reset, LED1...4 are all turned on and LED1 resume blinky. Description Information is displayed in the terminal.

```
PIOINTO_IRQHandler!
-- Basic WDT Project V1.0 --
-- EM-LPC1100 --
-- Feed watchdog timer to prevent it from timeout test --
WatchDog Reset!
```

- Reference manual: EM-LPC11xx Reference Manual.pdf

4.7 SSP test

- Source code location: 03-Software\Examples\07-SSP
- Test description: The SSP project is a simple program to test SSP LOOPBACK mode on the LPC1114 Starter Kit. There are three modes in SSP: loopback, master or slave. Here are the combination of all the tests. You need to modify the setting in ssp.h first in order to configure the proper mode for testing. Please see readme file for more details.
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic SSP Project V1.0 --

-- EM-LPC1100 --

-- Synchronous Serial Communication test --

SSP Loopback Test success!
```

If display message “SSP Loopback Test success!”, then the test is successful.

- Reference manual: EM-LPC11xx Reference Manual.pdf

4.8 UART test

- Source code location: 03-software\Examples\08-UART
- Test description: This example describes how to use UART to send and receive data through HyperTerminal.
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic UART Project V1.0 --

-- EM-LPC1100 --

-- Please input any key on the keyboard --
```

Input any key using PC keyboard, the corresponding letter will display in the HyperTerminal.

- Reference manual: EM-LPC11xx Reference Manual.pdf, MAX3232CSE.pdf

4.9 I2C EEPROM test

- Source code location: 03-Software\Examples\09-I2C_EEPROM
- Test description: This testcase configures I2C working in master mode, and Read/Write EEPROM using I2C bus. Init buffer,send the buffer data to EEPROM,read them and compare. If the data read is the same as data written,then test success.
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button. If it displays message “I2C EEPROM test success!” in the terminal,then test success. Phenomenon is as follows.

```
-- Basic I2C EEPROM Project V1.0 --

-- EM-LPC1100 --

-- I2C Read/Write EEPROM test --

I2C EEPROM test success!
```

- Reference manual: EM-LPC11xx Reference Manual.pdf, M24C64.pdf

4.10 I2C Temperature Sensor test

- Source code location: 03-Software\Examples\10-TemperatureSensor
- Test description: This example describes how to configure I2C work in master mode. It test the temperature sensor LM75 and read the four registers' value. They are Temperature Register, Configuration Register, Tos Set Point Register and Thyst Set Point Register.
- Test phenomenon: Build the program and download it inside the evaluation board. Open PC HyperTerminal and push RESET button. If test success, it may display the four registers' value in the terminal. Phenomenon is as follows.

```
-- Basic I2C Temperature Sensor Project V1.0 --  
  
-- EM-LPC1100 --  
  
-- I2C Temperature Sensor test --  
  
-I- Temperature Register value:0x1280  
  
-I- Configuration Register value:0x0  
  
-I- Tos Set Point Register value:0x80  
  
-I- Thyst Set Point Register value:0x1280  
  
Temperature Sensor test success!
```

- Reference manual: EM-LPC11xx Reference Manual.pdf, LM75.pdf

Chapter 5 Software development and examples

5.1 MDK Introduction

RealView MDK Development Suite is the latest software development tool of ARM Limited Corporation for Embedded Processors. MDK4.02 integrates the most advanced technology in this industry, including μ Vision4 IDE and RealView RVCT4.0 Compiler; RealView MDK supports ARM7, ARM9 and the latest Cortex-M3 Core Processor. It has a configuration wizard for startup code and integrates flash program module, powerful device simulation, performance analyzer and so on. You can obtain MDK software from the CD released with LPC1114 KIT in the directory of 04-tools\Realview MDK4.02, or you can download the latest version from Keil website www.keil.com. Double click the installation file setup.exe; finish Keil μ Vision4 installation under the guidance of the installation wizard. The installation interface as follows:



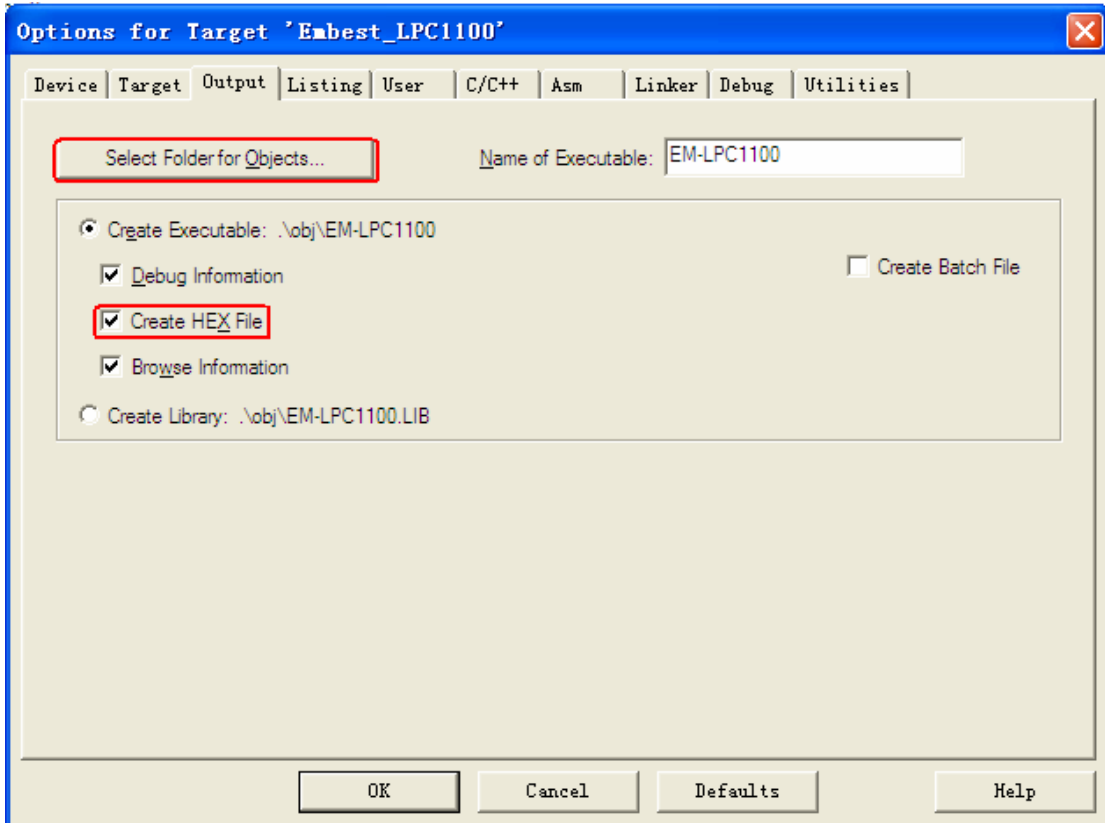
5.2 Compile routine

5.2.1 Open a routine

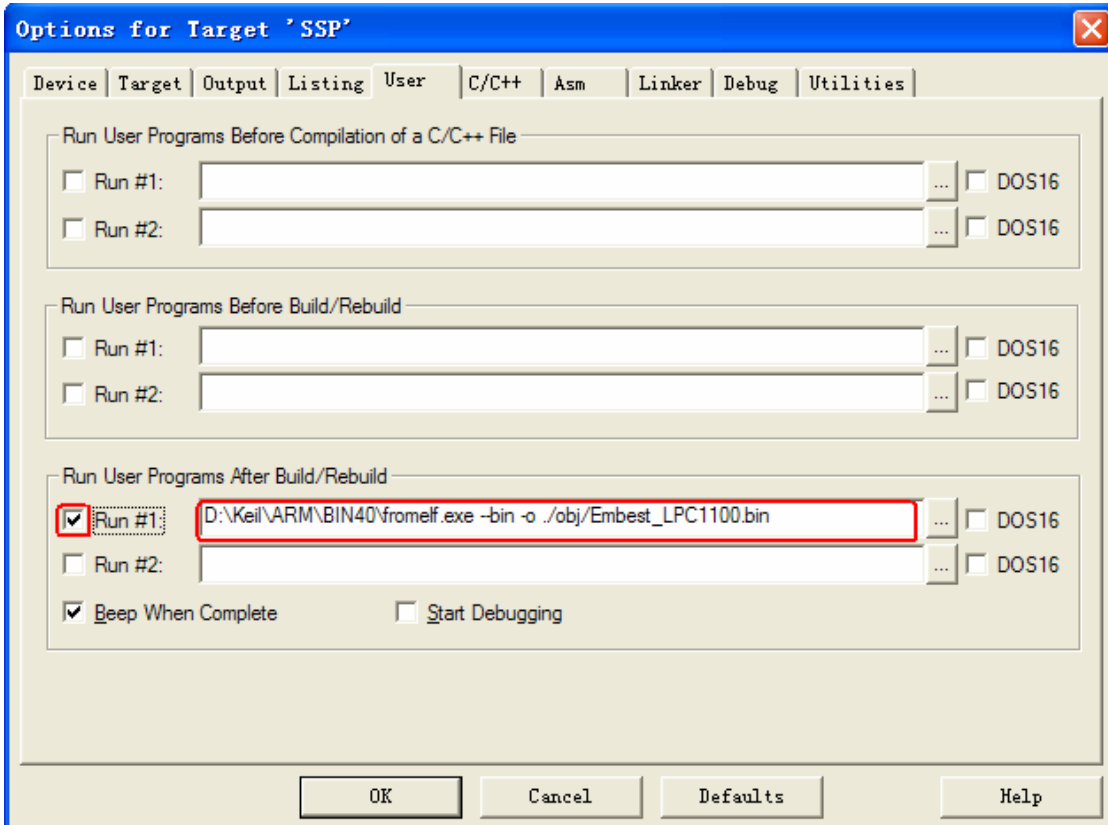
Enter in the folder of 03-software\Examples; Open the Embest_LPC1100.uvproj file in the project directory of one routine. For example: \03-Software\Examples\01-Blinky\project\Embest_LPC1100.uvproj.

5.2.2 Compile the routine

1) If you need the .hex format file, then you can configure MDK to build the hex file, Click Select Folder for Object... to specify the hex file's output directory. Or you can skip this step.

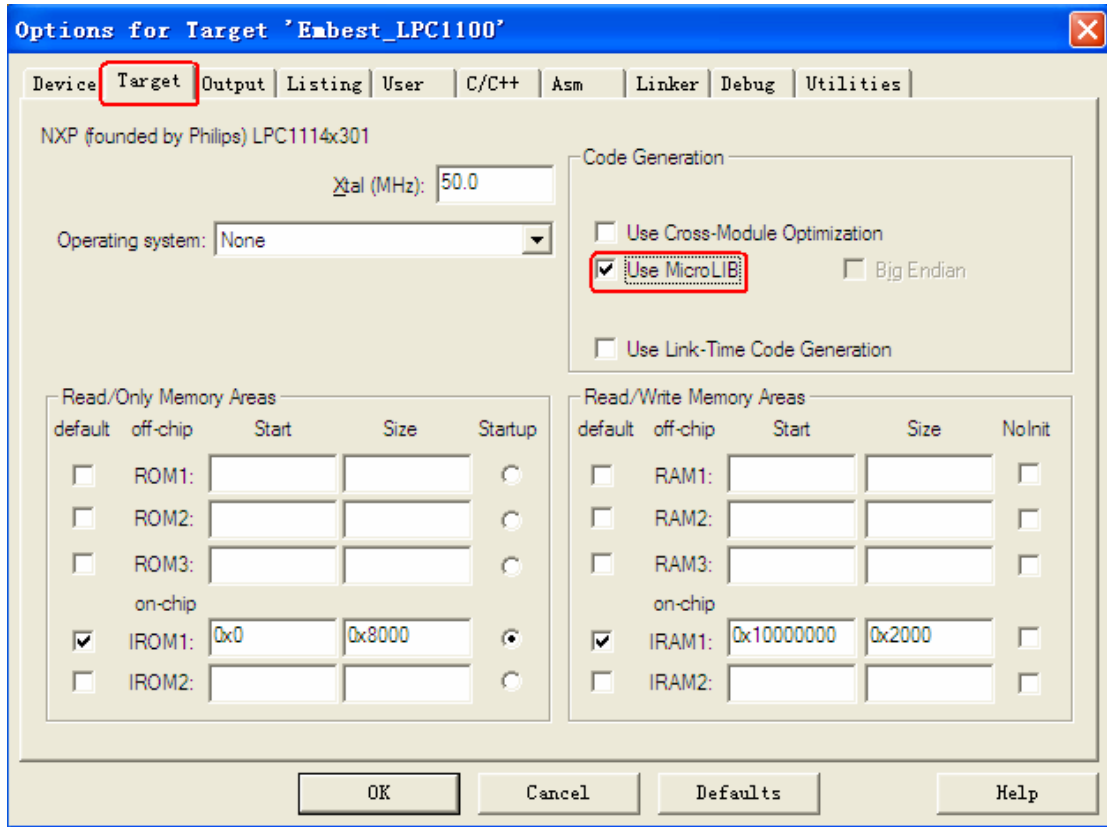


2) If you need the .bin format file, then you can configure MDK to build the bin file. Or you can skip this step.

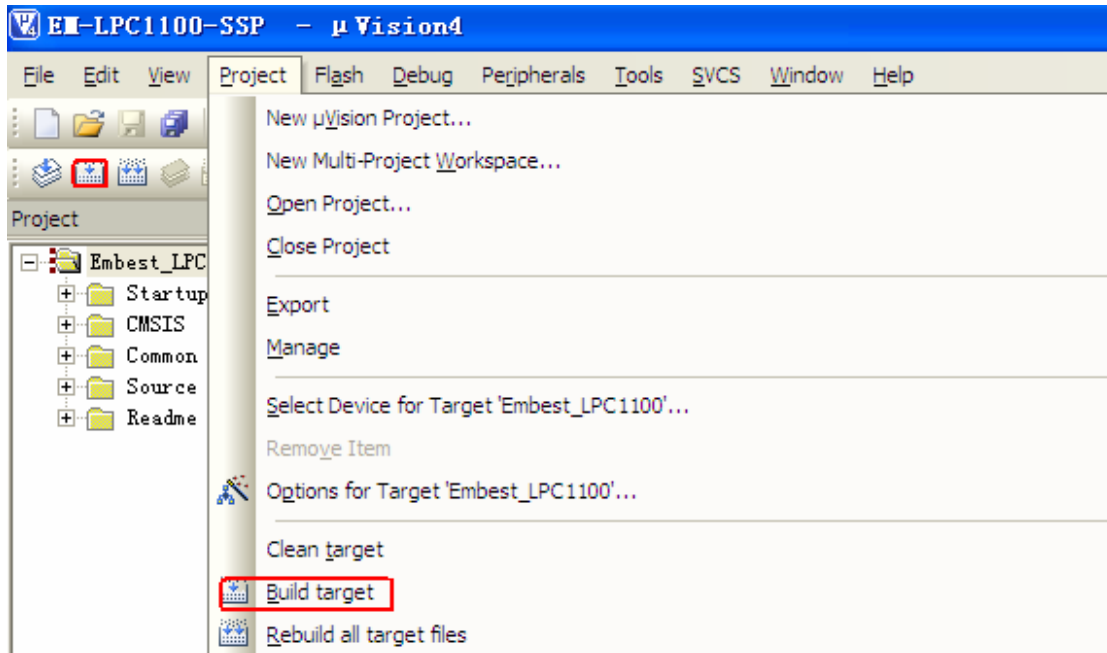


- D:\Keil\ARM\BIN40\fromelf.exe specify the path of fromelf.exe, it will convert axf file to bin file
- --bin -o export bin file
- ./obj/Embest_LPC1100.bin the directory and file name of the bin file you want to create
- ./obj/Embest_LPC1100.axf the directory and file name of the axf file you want to convert

3) Use MicroLIB. Through the serial port using the printf statement to print debugging information, the option "Use MicroLIB" need to be selected. Click Project->Options for Target and select this option in the "Target" tab.



4) Click project->build, or click the shortcut button to build the routine.



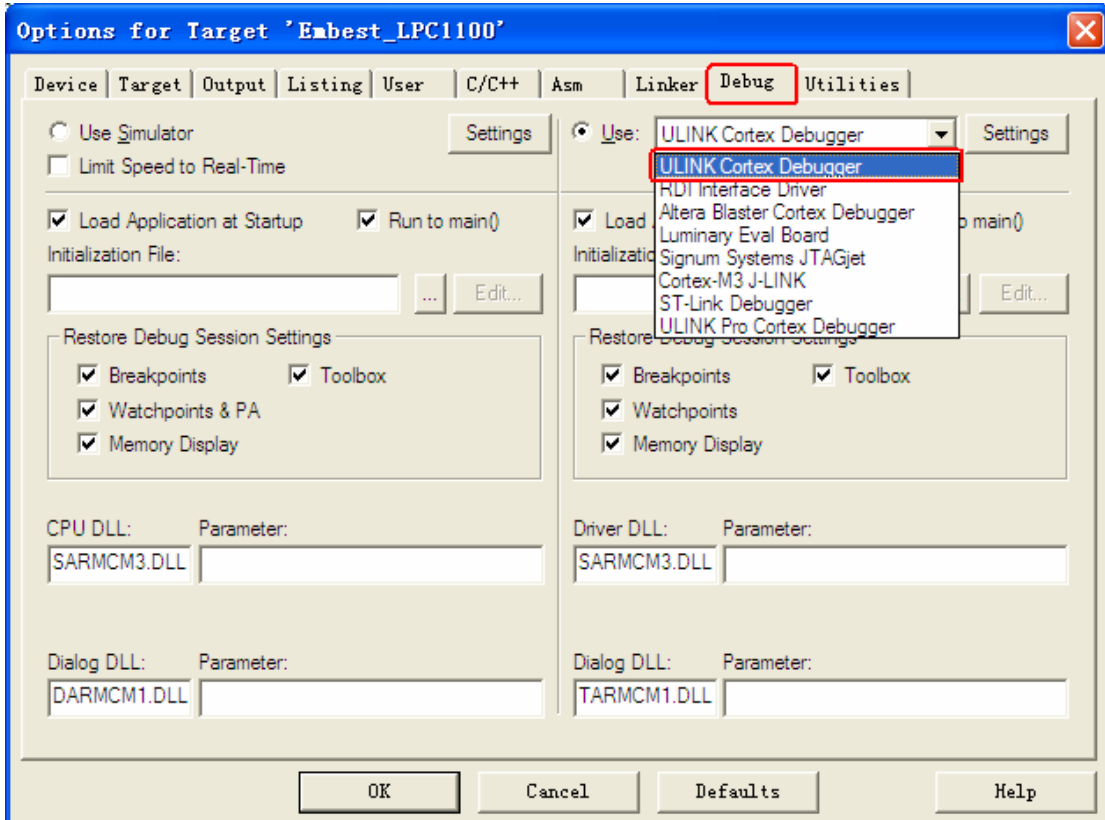
5.3 Debug and Download the routine using Emulator

The precondition for the next steps is that you have bought or owned a corresponding hardware Emulator.

5.3.1 Debug and Download the routine using ULINK2

1. Debug the routine using ULINK2

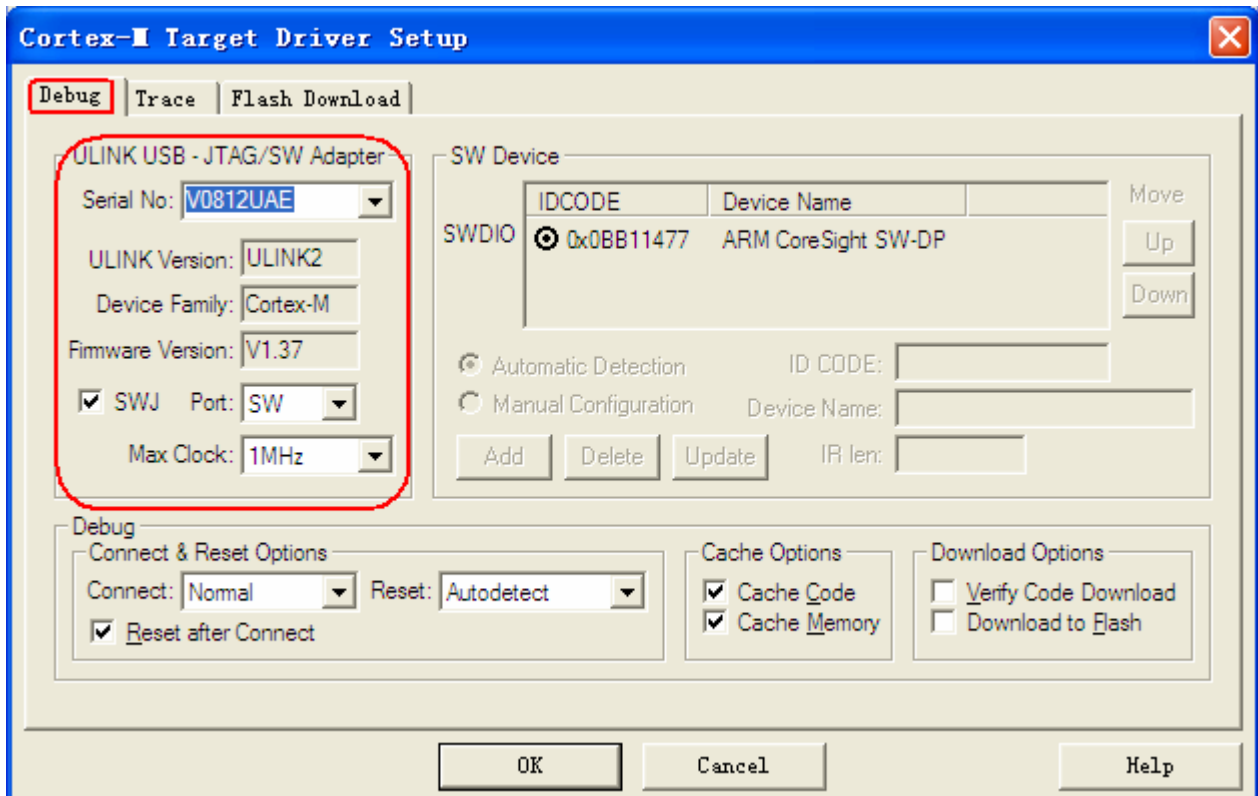
1) Choose Emulator



2) Check the ULINK2, optional.

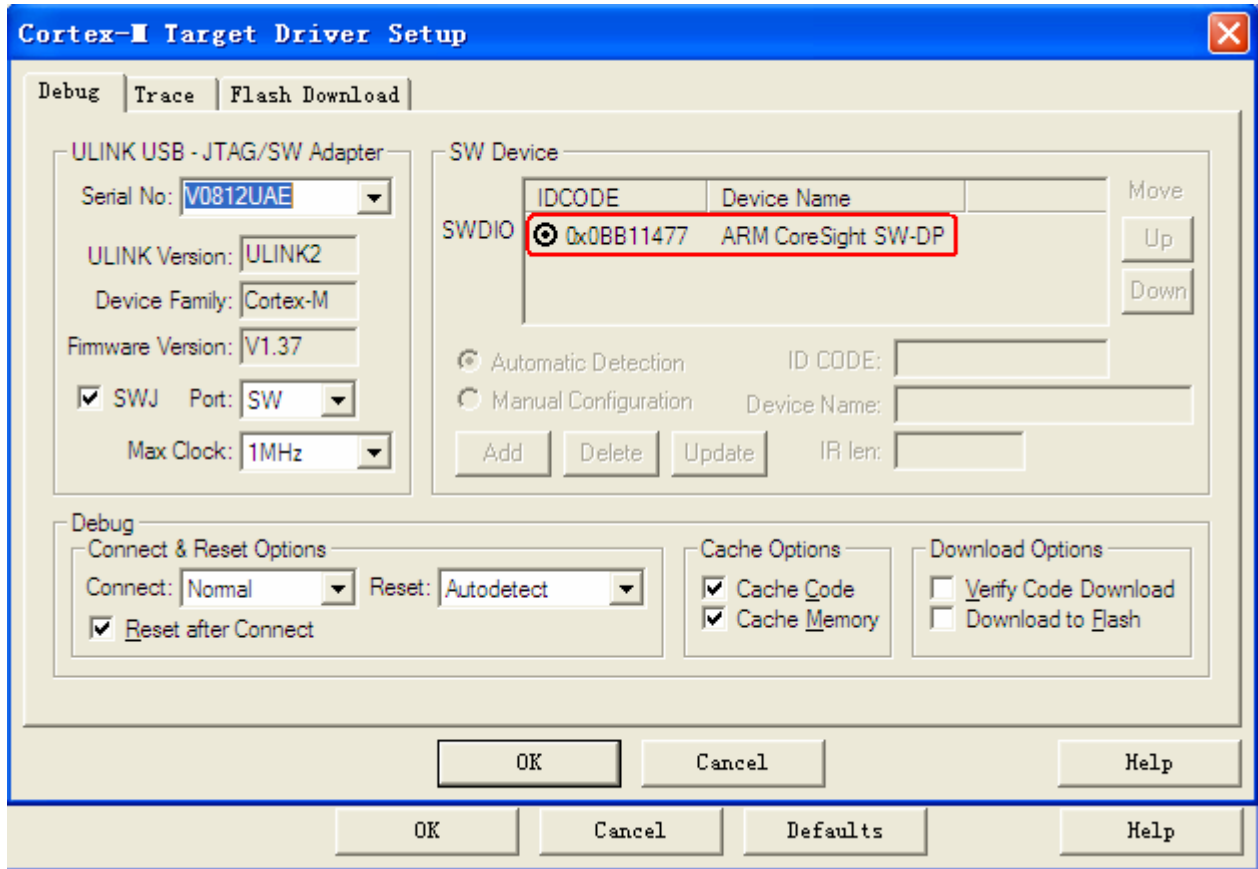
When ULINK2 connects to the Development Board, if the RUN and COM indicator lights first change to blue and then go out, and the USB indicator light has always been red, this proves that ULINK2 is no problem.

In addition, you can use the next way to check ULINK2. Click the Settings button in the Debug TAB, if the red marked part appears, it proves that ULINK2 is no problem.

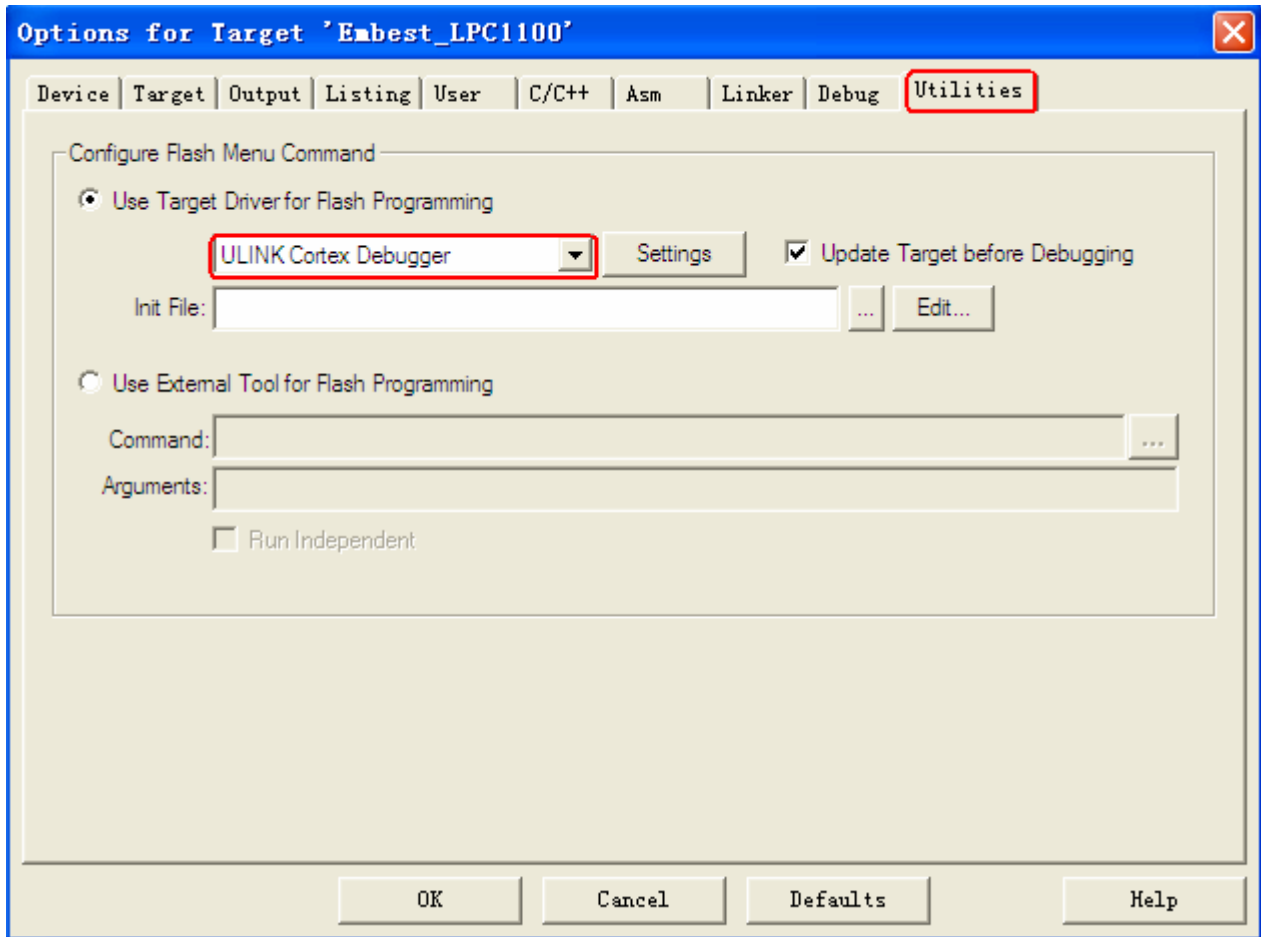


3) Check that whether ULINK2 can detect the development board or not, optional.

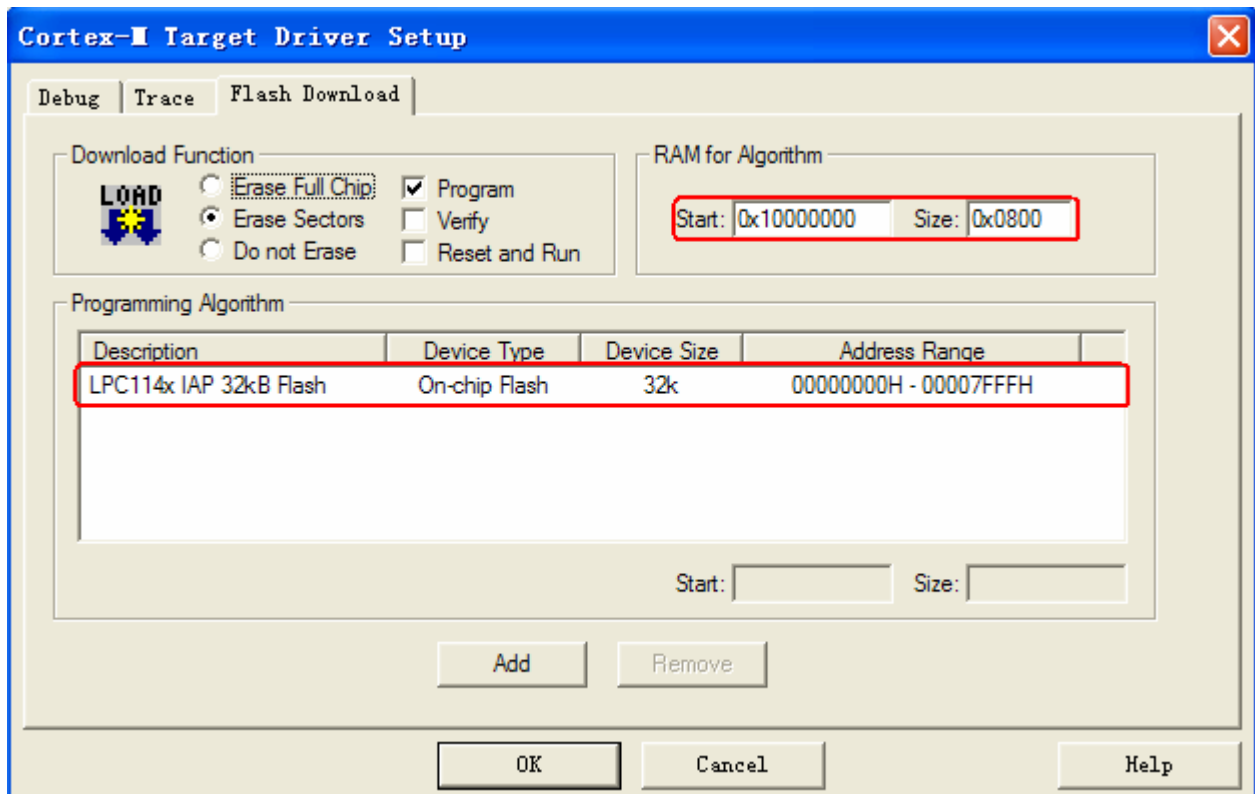
Click the Settings button in the Debug TAB, if the red marked part appears, it proves that ULINK2 has detected the development board.



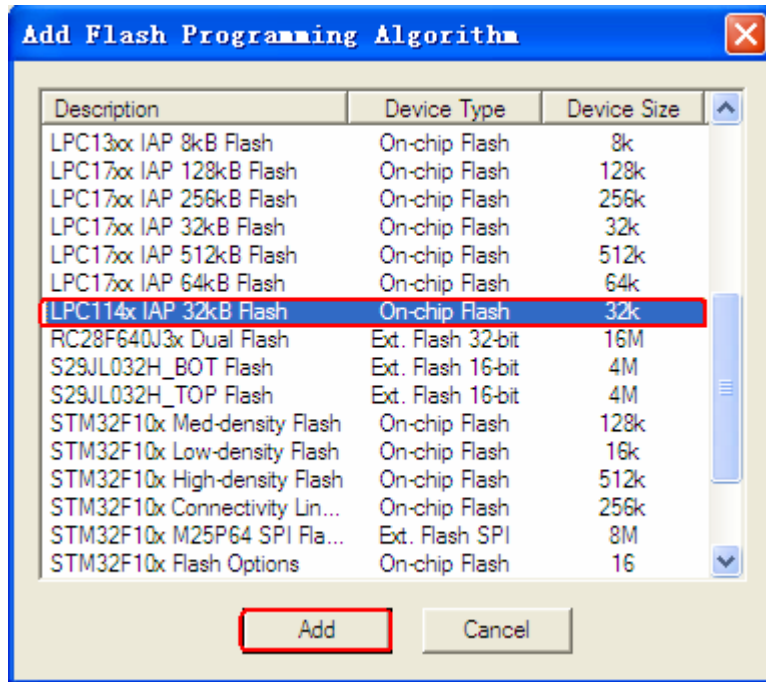
4) Set the Flash Programmer, configure the Utilities TAB.



Because MDK 4.02 version doesn't include LPC1114x301 download algorithm. You need some manually set. Copy LPC_IAP_CD0_32.FLM file from the directory 04-Tools in CD-ROM to KEIL install directory Keil\ARM\FIash. Then click Settings button, it appears:

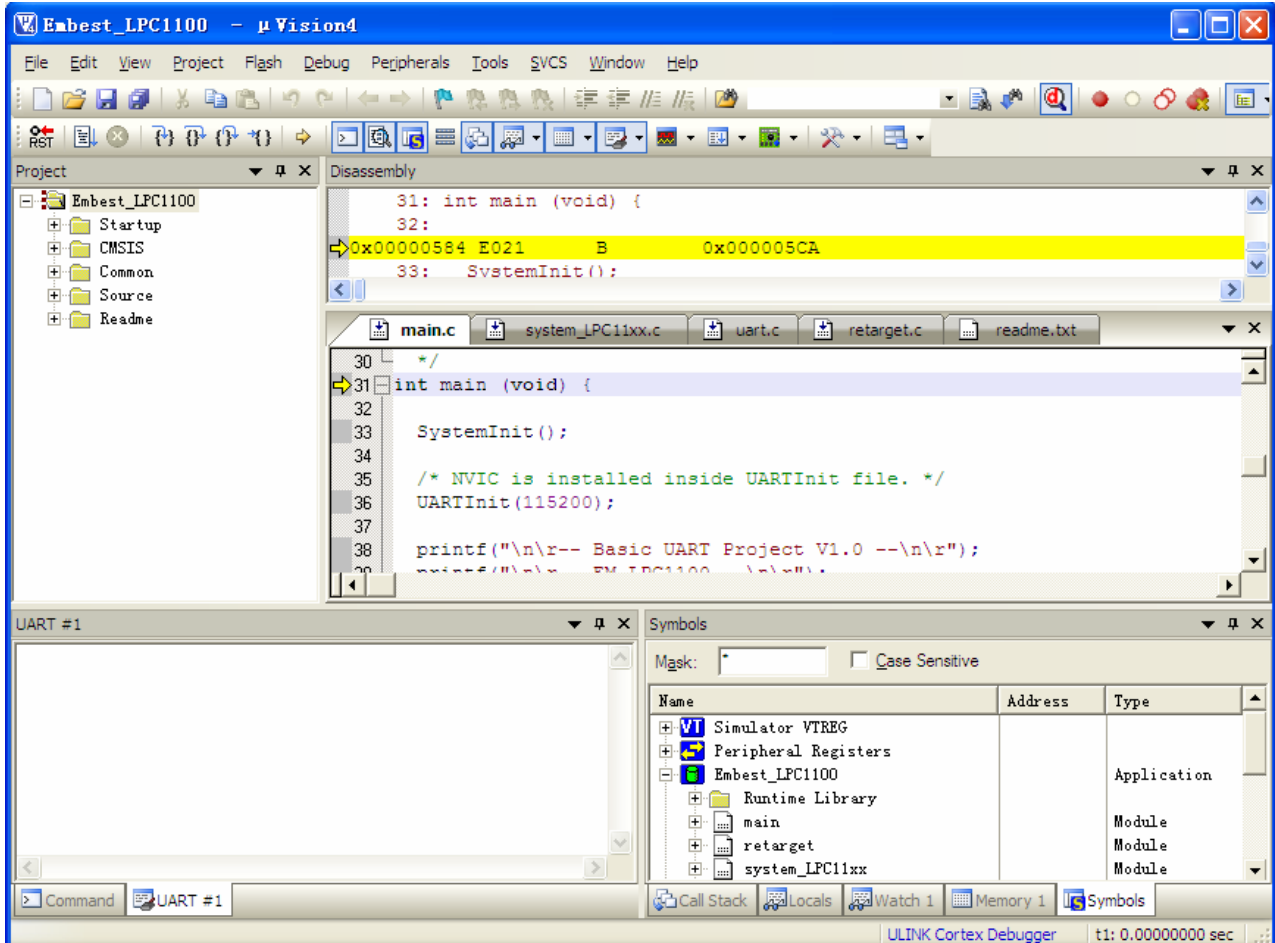


If nothing is in the Programming Algorithm, then you should add the corresponding Flash programming algorithm by clicking the Add button, as follows:

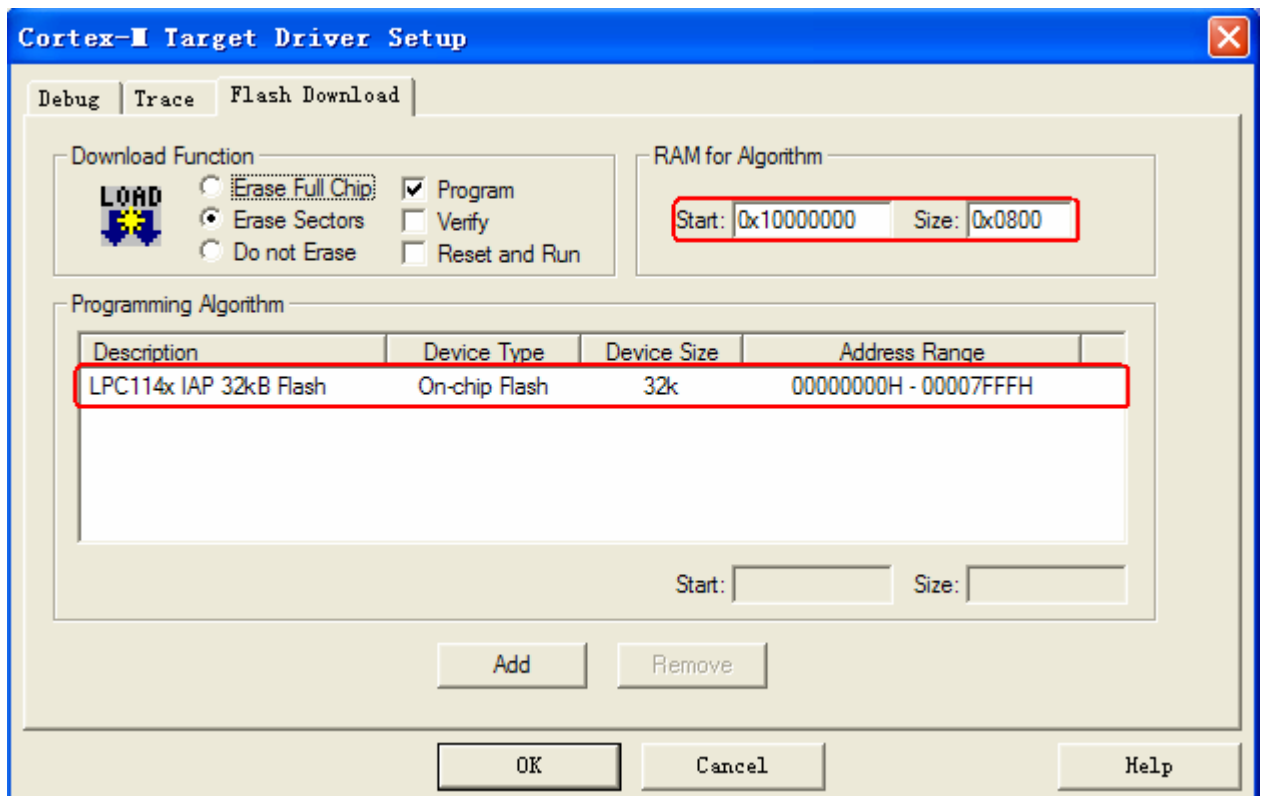
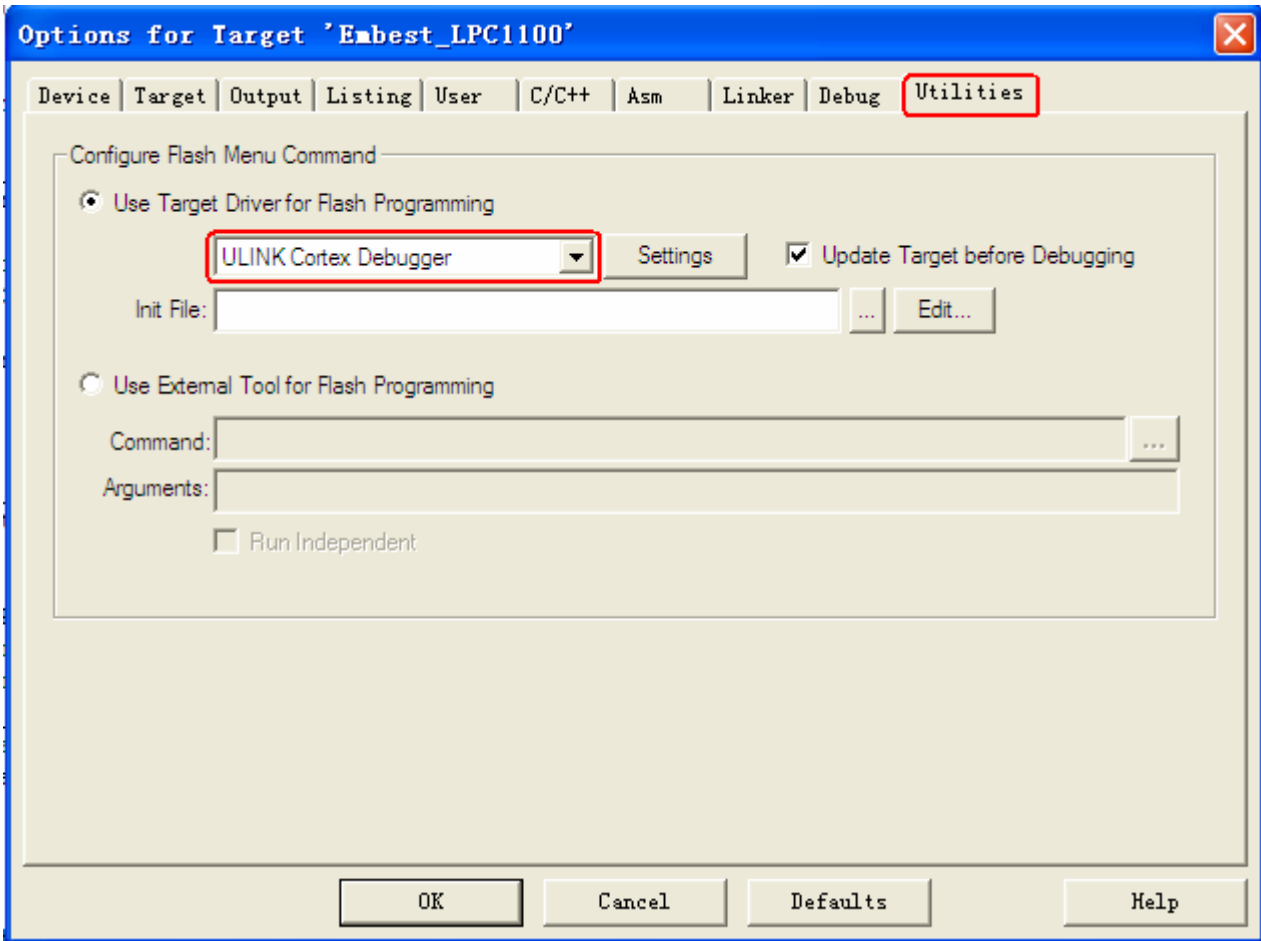


After you choose a Flash Programming Algorithm, then Click Add button.

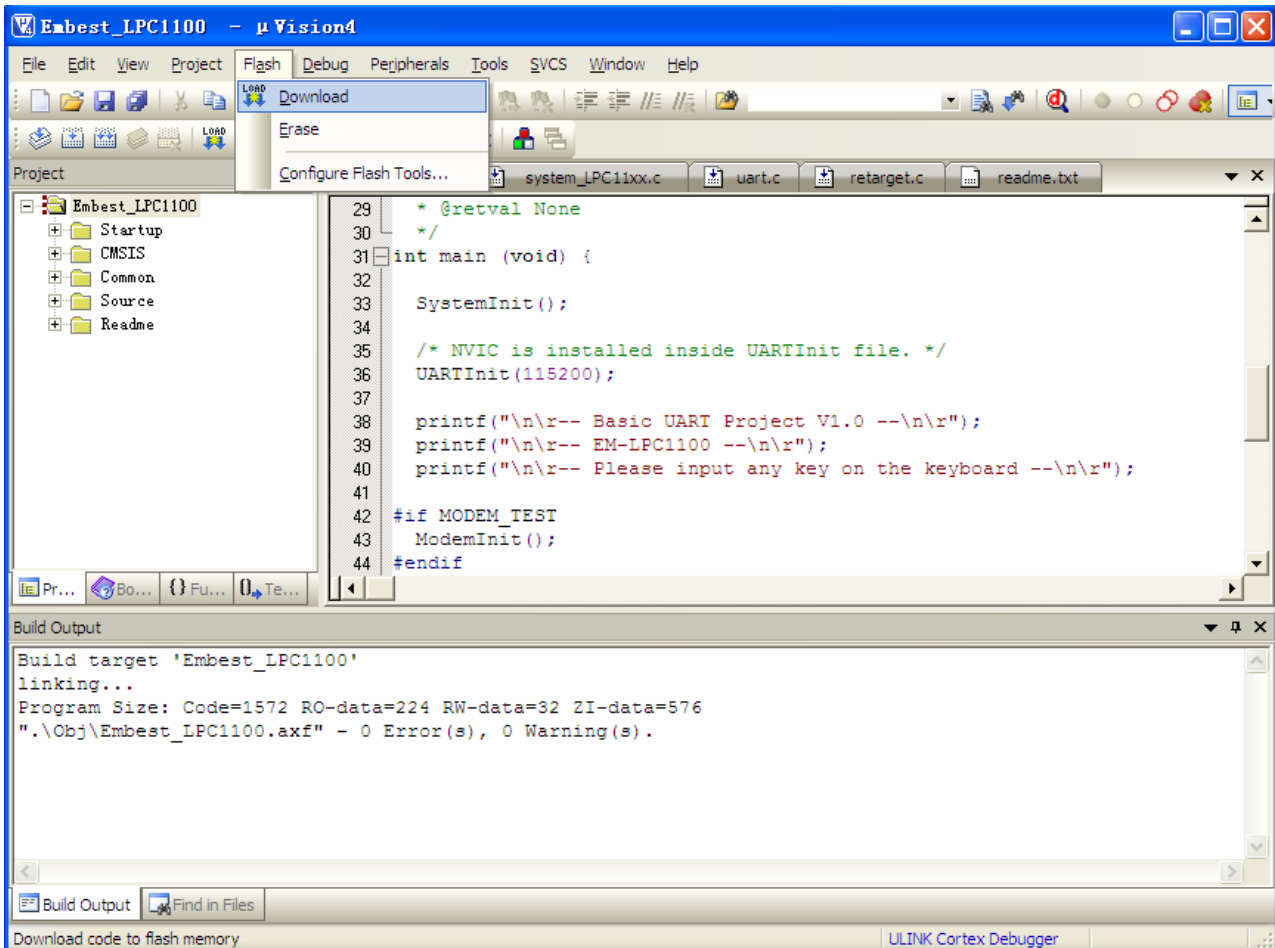
5) Start to Debug the routine by clicking shortcut button or clicking Debug->Start/Stop Debug Session, the status of the debug as follows:



2. Download the routine using ULINK2
 - 1) Check the Flash Programmer's setting



- 2) Start to Download by click Flash->Download or the shortcut button as follows:



5.3.2 Download and Debug the routine using Colink2

1. Application environment

- 1) Colink2 can only be used in Windows XP operation system, supports SP3 and higher version.
- 2) Debugger and the target board should using the common GND.
- 3) When debugging using Colink2 integrated on the LPC1100 development board, it is recommended to use MDK4.03 version, because MDK4.03 just began full support the LPC1100.

2. Colink2 for LPC1100 JTAG Debugger Features

- Only support LPC1100 Series currently(Can support other Cortex-M0/M3 MCU later)
- Support SW/JTAG debug and Download
- Download speed up to 110KB/s
- Flexible debugging socket, support 20/10/8 pins connector
- Support Keil MDK and CooCox full series software
- Debugging conditions: use SW debug, the LPC1114 need to run at 50MHz frequency.

3. Install the Colink2 for MDK driver.

1) Install USB driver

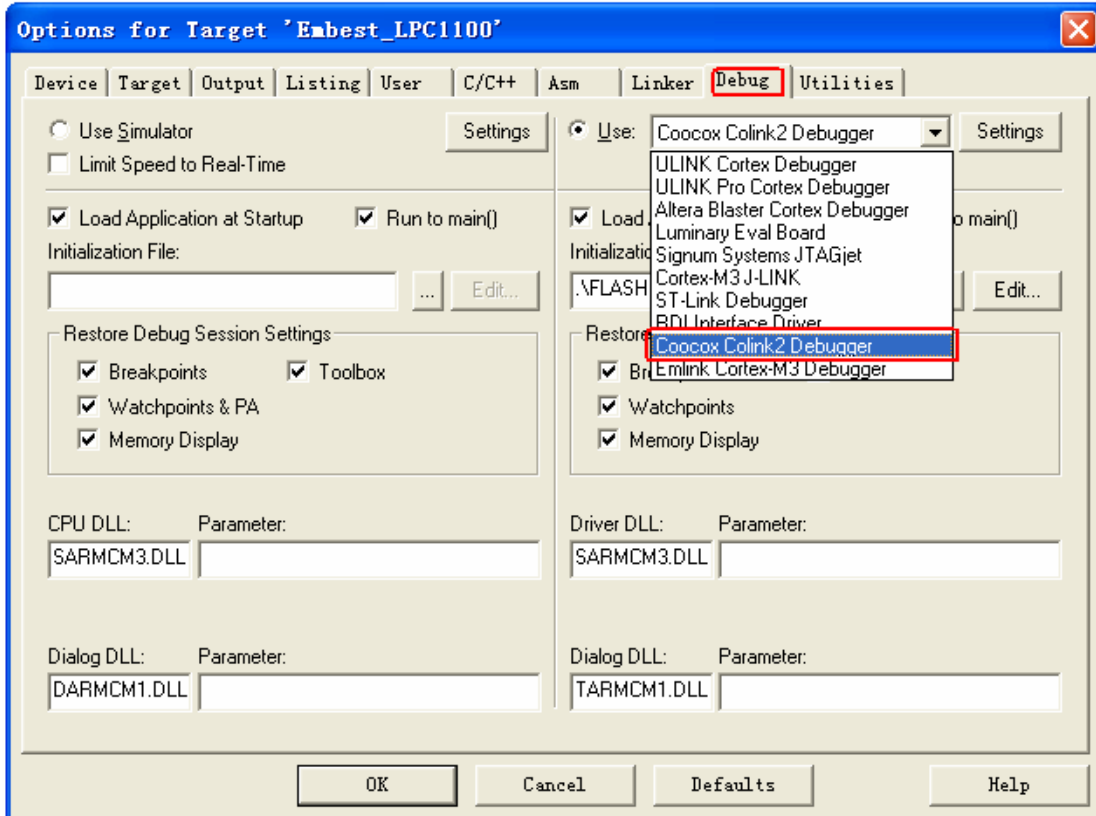
- Power on Colink2 for LPC1100 board, pop-up found hardware dialog box, wait for the pop-up hardware installation wizard;
- Choose "From a list or specific location (Advanced)" → "Search for the best driver in these locations(S)" → Check the "include this location in the search (O)" → "Browse"(point to the driver directory of the CD-ROM, 04-Tools\colink_for_lpc1100_driver), then the next step to complete the installation.
- After a successful installation, indicating a successful hardware installation, you can use.

2) Configure MDK

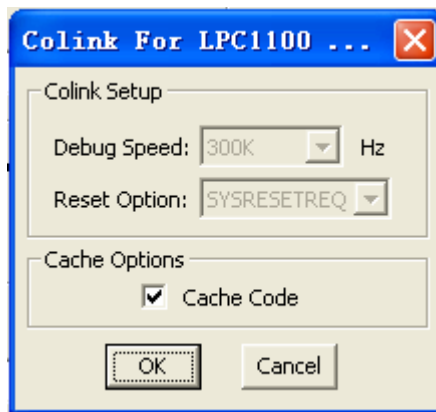
- Copy colink2.dll and colink-agdi-lpc1100.dll from directory 04-Tools\colink_agdi_for_lpc1100 of CD-ROM to BIN directory of the Keil Installation path. For instance: D:\Keil\ARM\BIN\. You can download the latest plugin from CooCox formal website www.coocox.org.
- Add this sentence ' TDRV50=BIN\colink-agdi-lpc1100.dll("CooCox Colink For LPC1100 Debugger") ' in the [ARMADS] section of Tools.ini file in the path of Keil's installation.
- Choose "CooCox Colink2 Debugger" when degugging.

4. Debug the routine using Colink2

1) Choose the Emulator.

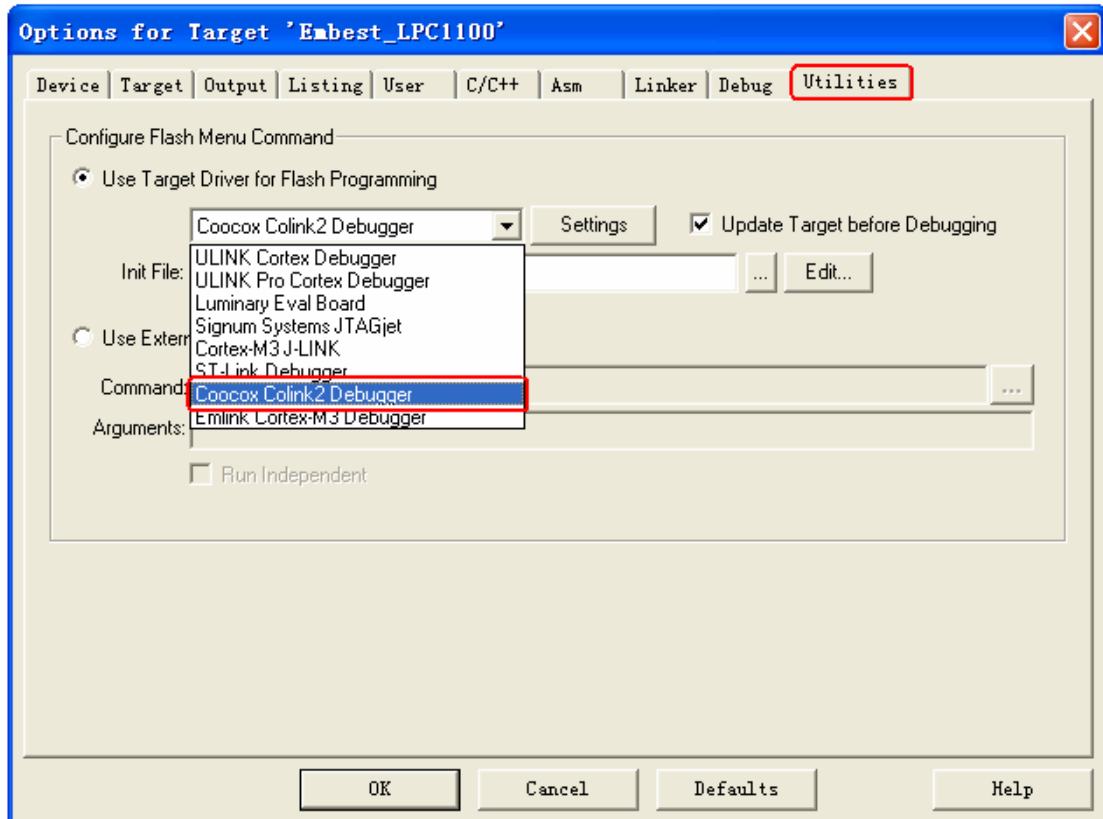


2) Click the Settings button to configure Colink2, as follows:

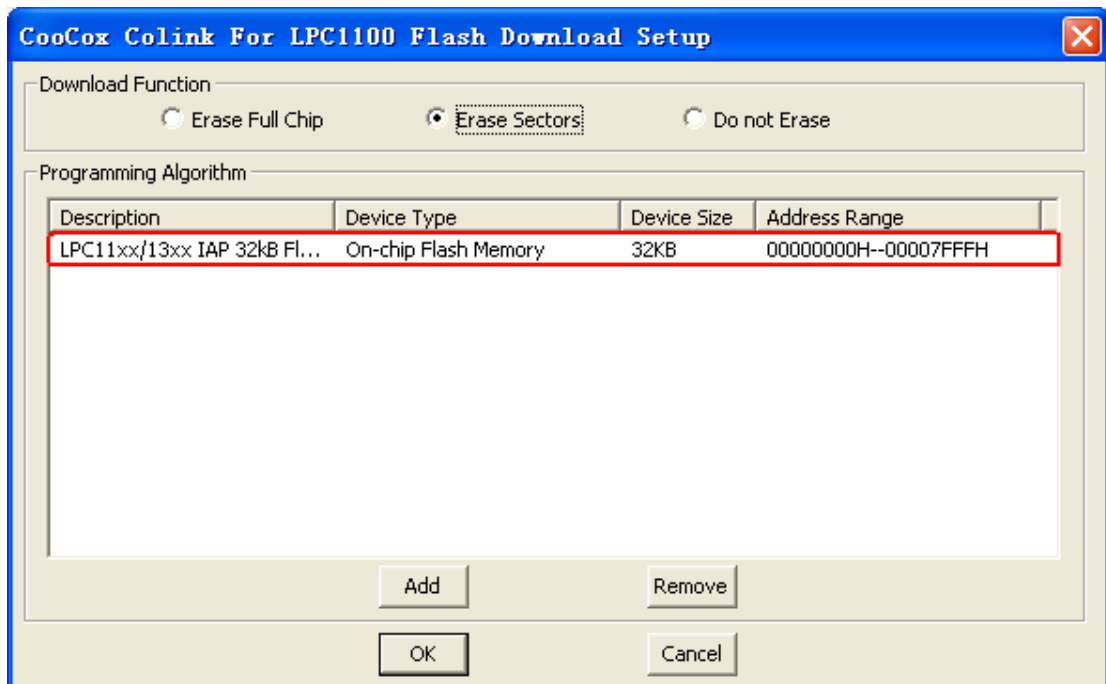


3) Set the Flash Programmer.

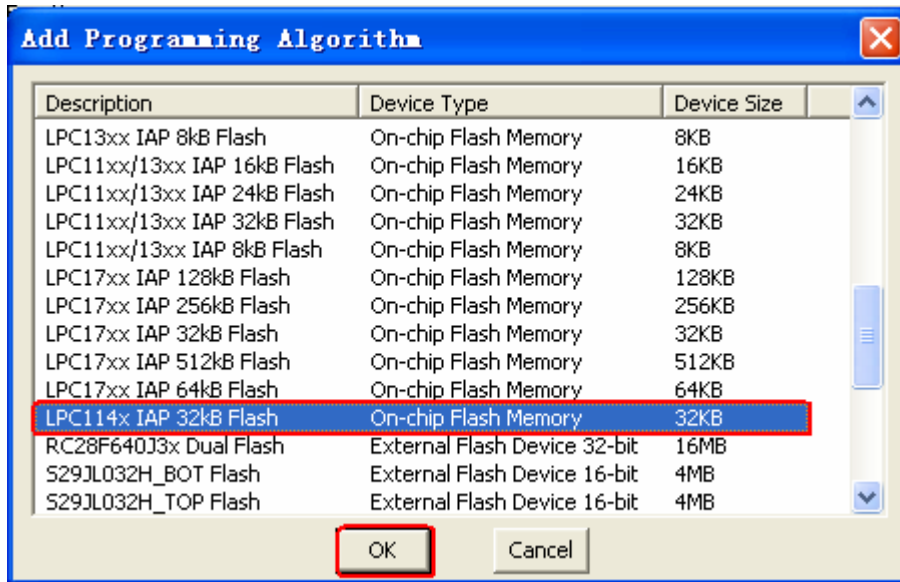
Configure the Utilities TAB, as follows:



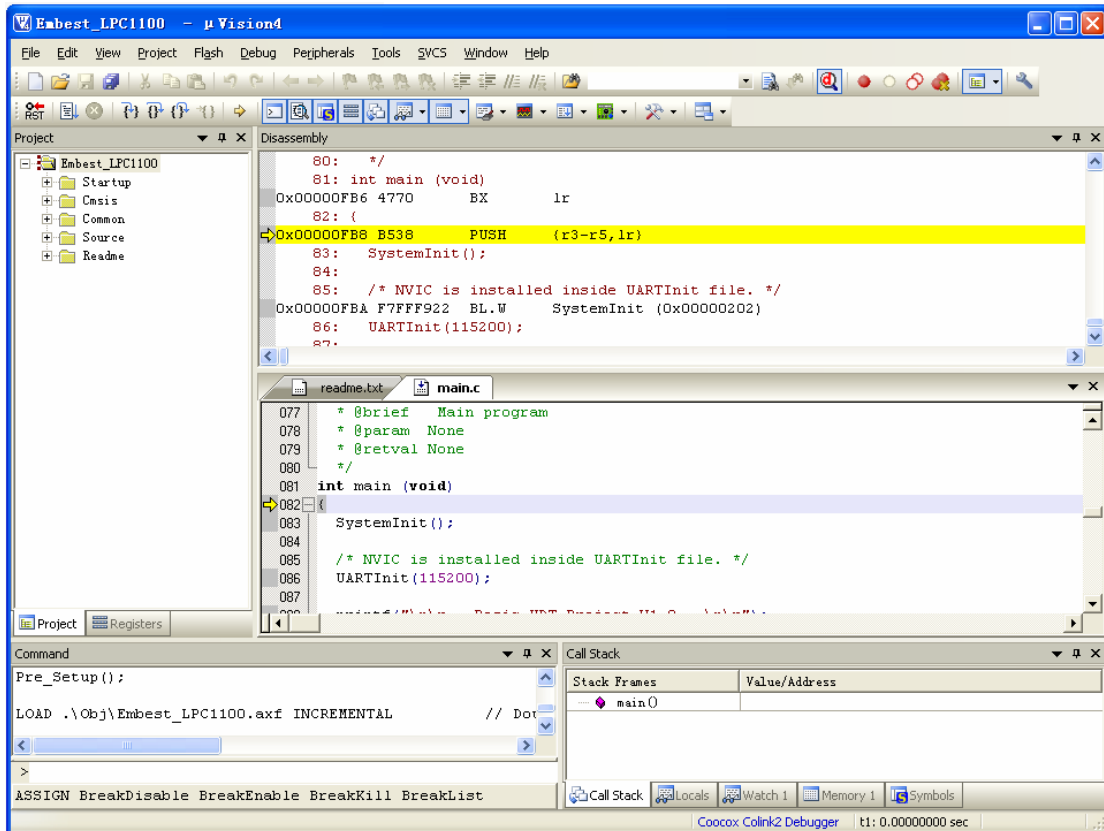
Then click the Settings buttons, it will show the next Dialog.



If nothing is in the Programming Algorithm, then you should add the corresponding Flash programming algorithm by clicking the Add button, as follows:

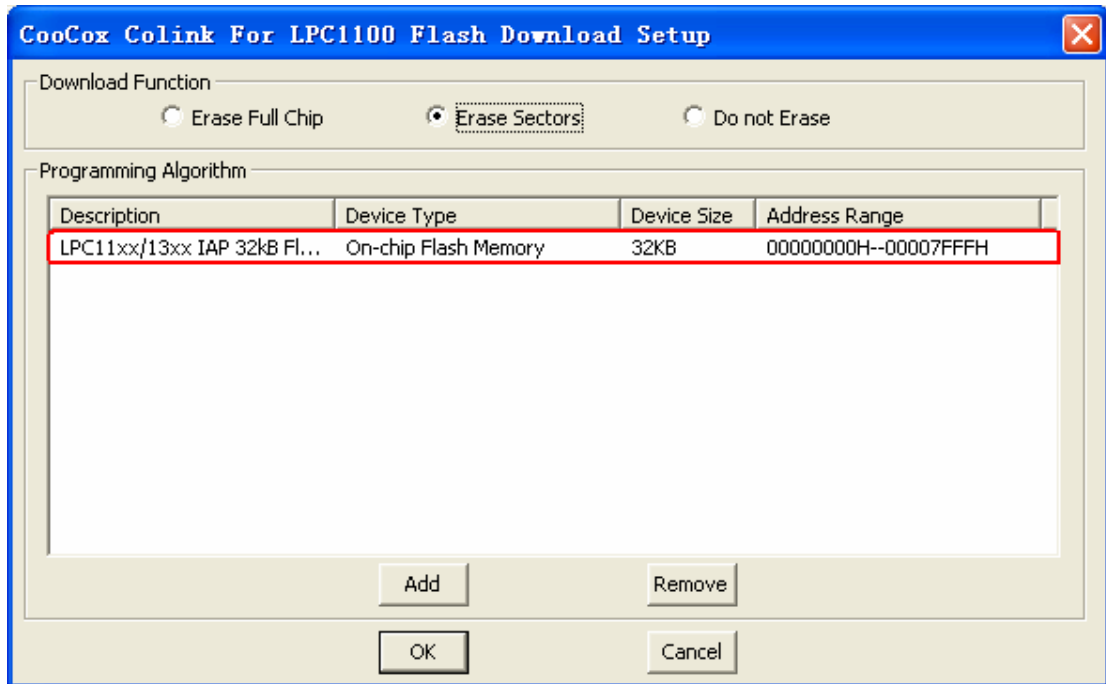
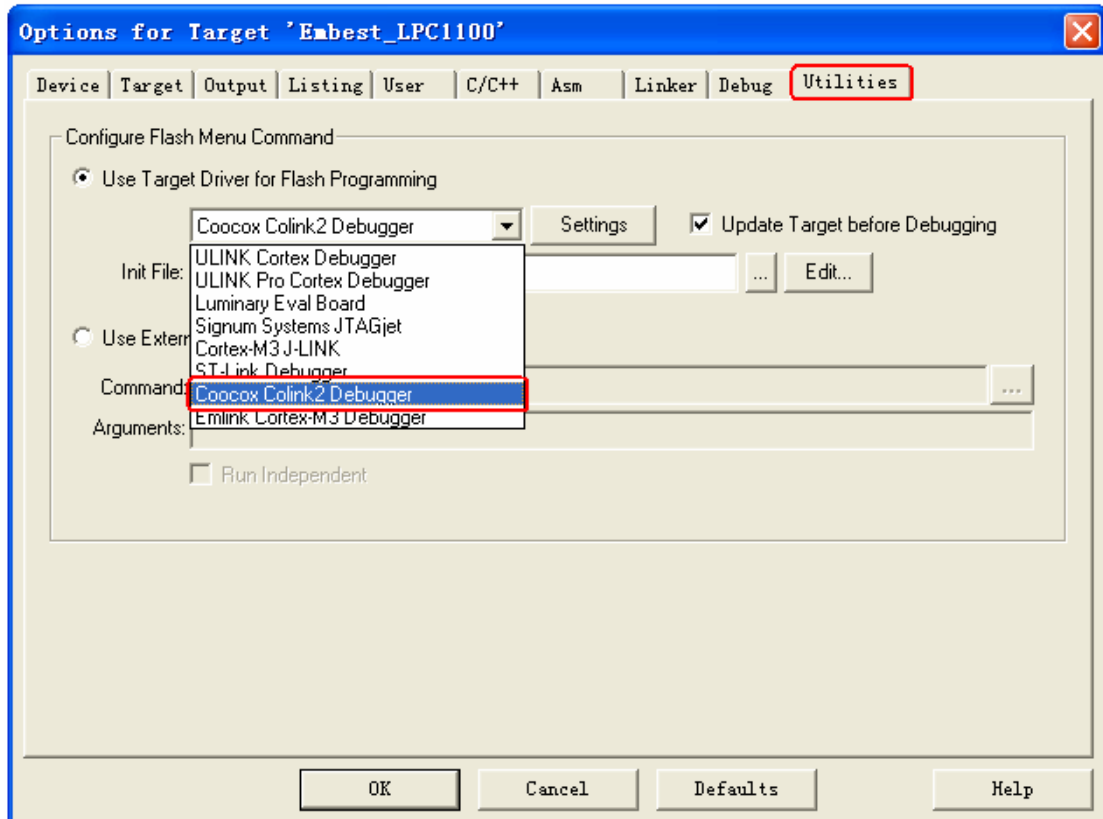


4) Start to debug the routine by clicking the shortcut button  or clicking Debug->Start/Stop Debug Session, the status of the debug is as follows:

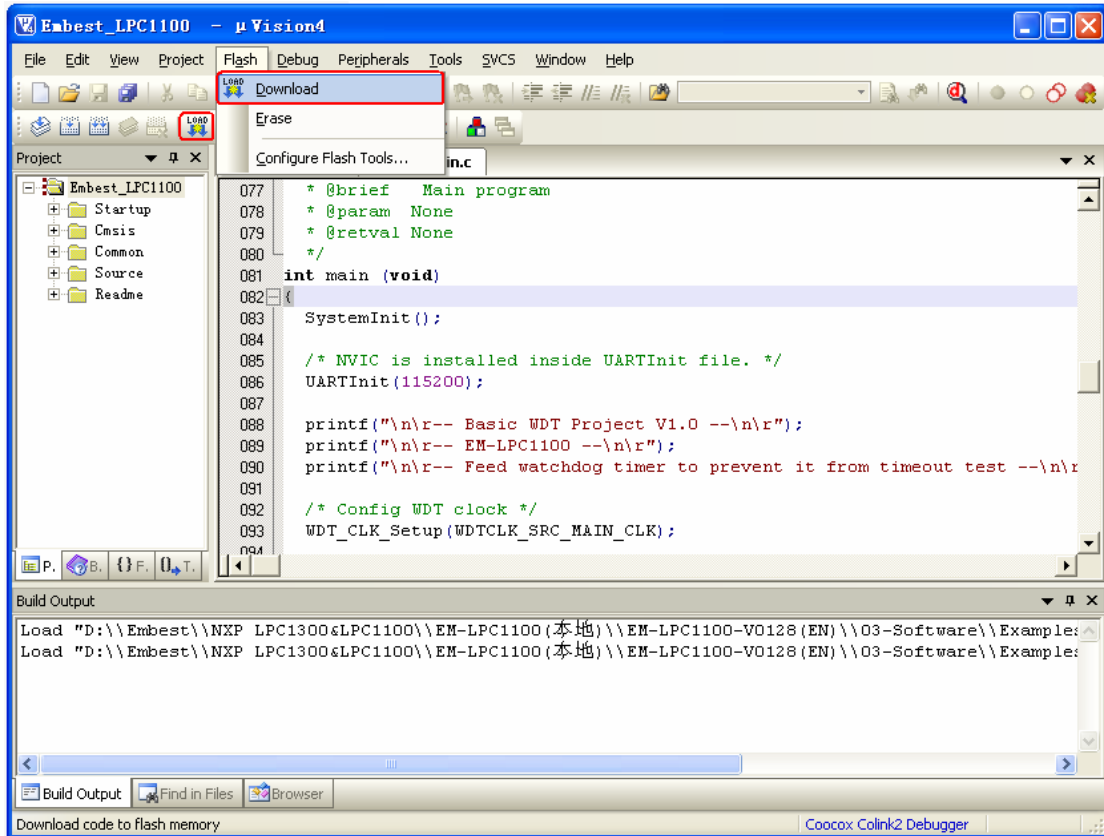


5. Download the routine using Colink2

1) Check the Flash Programmer's setting.

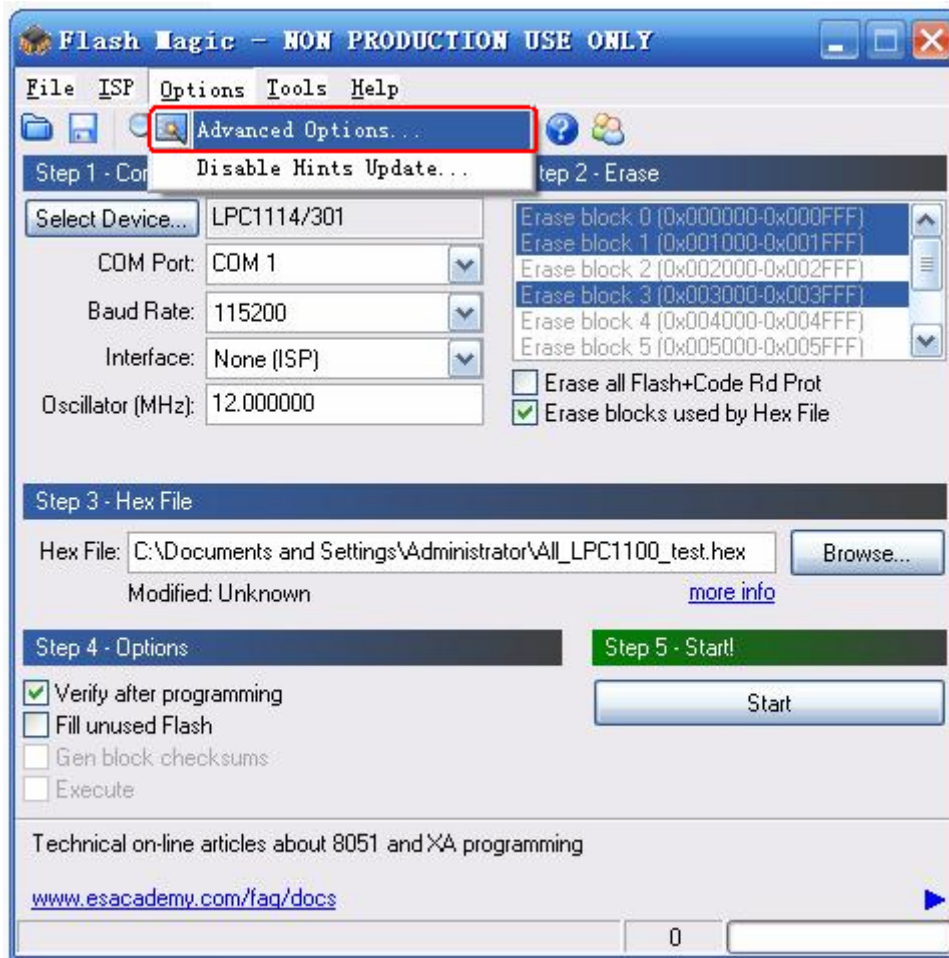


2) Start to download by clicking Flash->Download or the shortcut button, as follows:

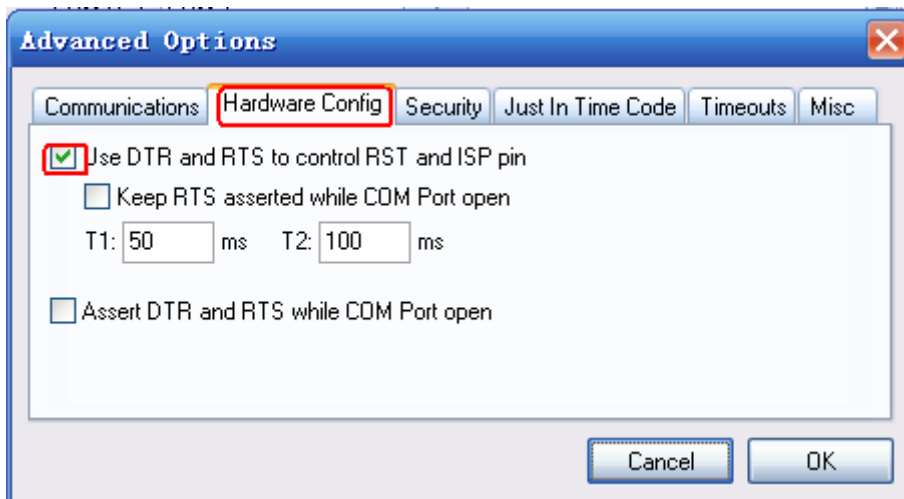


5.4 Download the routine using ISP

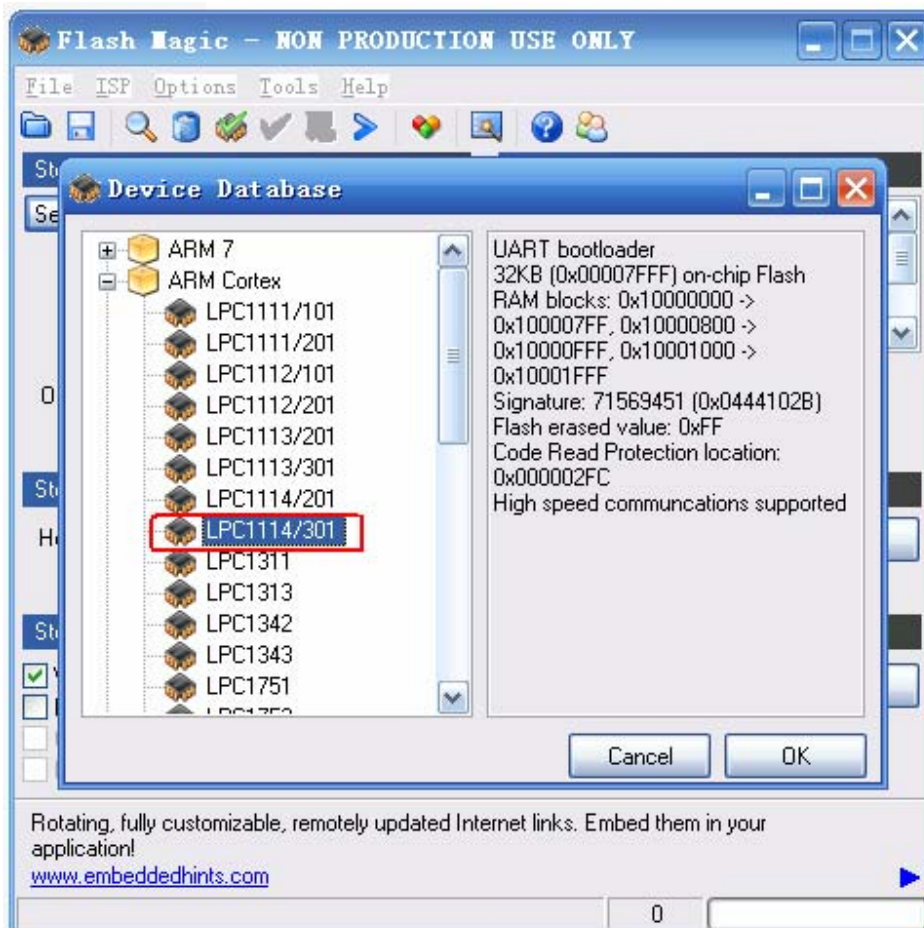
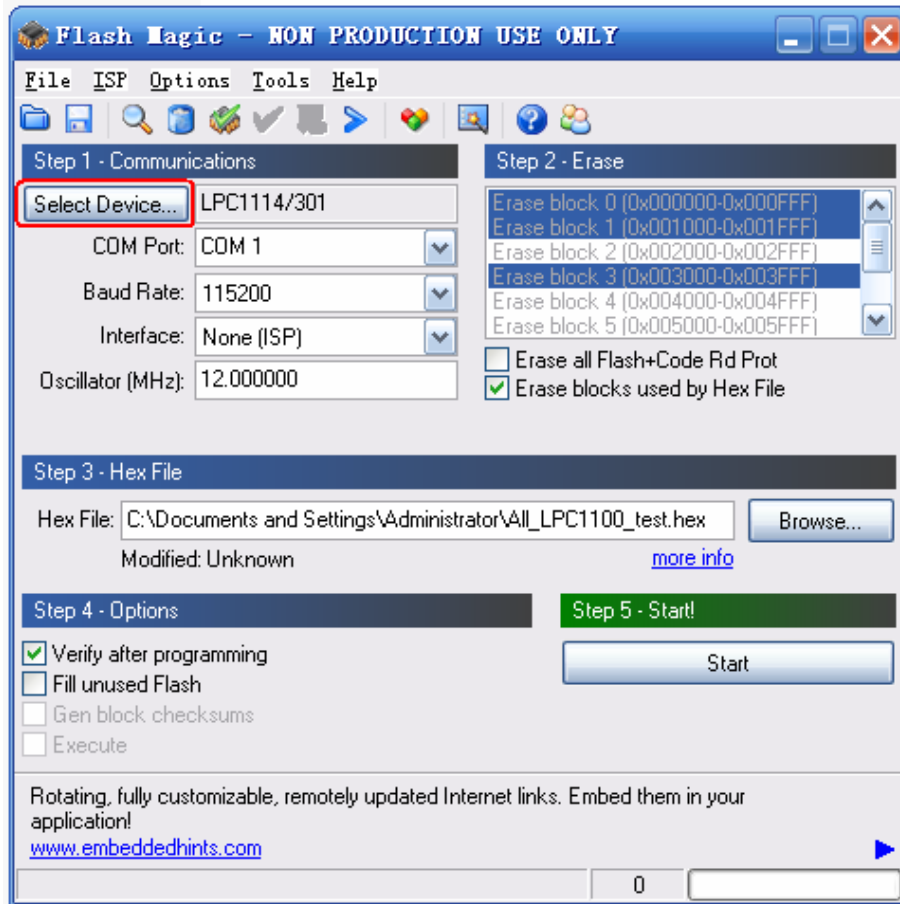
1. Install FlashMagic. Double click FlashMagic V5.53.exe in the directory 04-Tools of CD-ROM and install it. Or you can download the latest version in the internet and install it.
2. Keep JP2 jumper closed and press RESET button on the board.
3. Close all the HyperTerminal application and ensure the connection between UART interface of the board and the COM of PC using serial cable.
4. Open FlashMagic and set as follows:
 - 1) Click menu "Options"->"Advanced Options..."



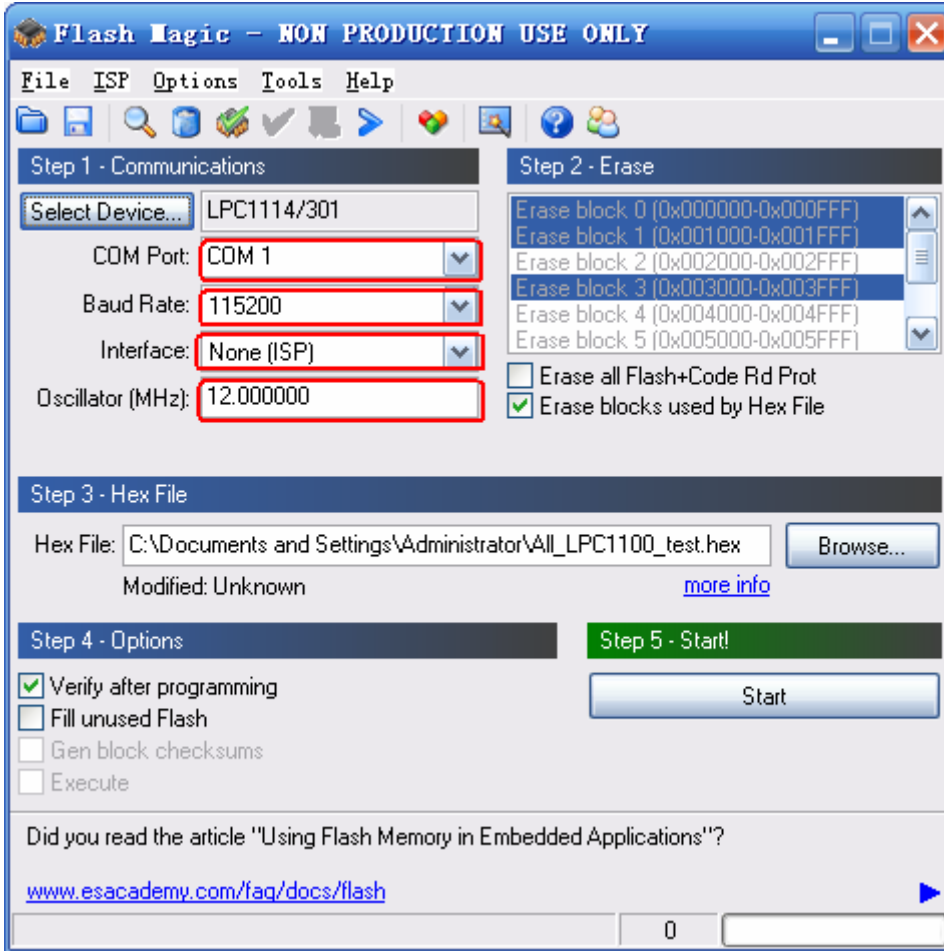
2) Open the interface as shown below. Switch to the Hardware Config tab. Ensure the next figure tick option selected, then click OK.



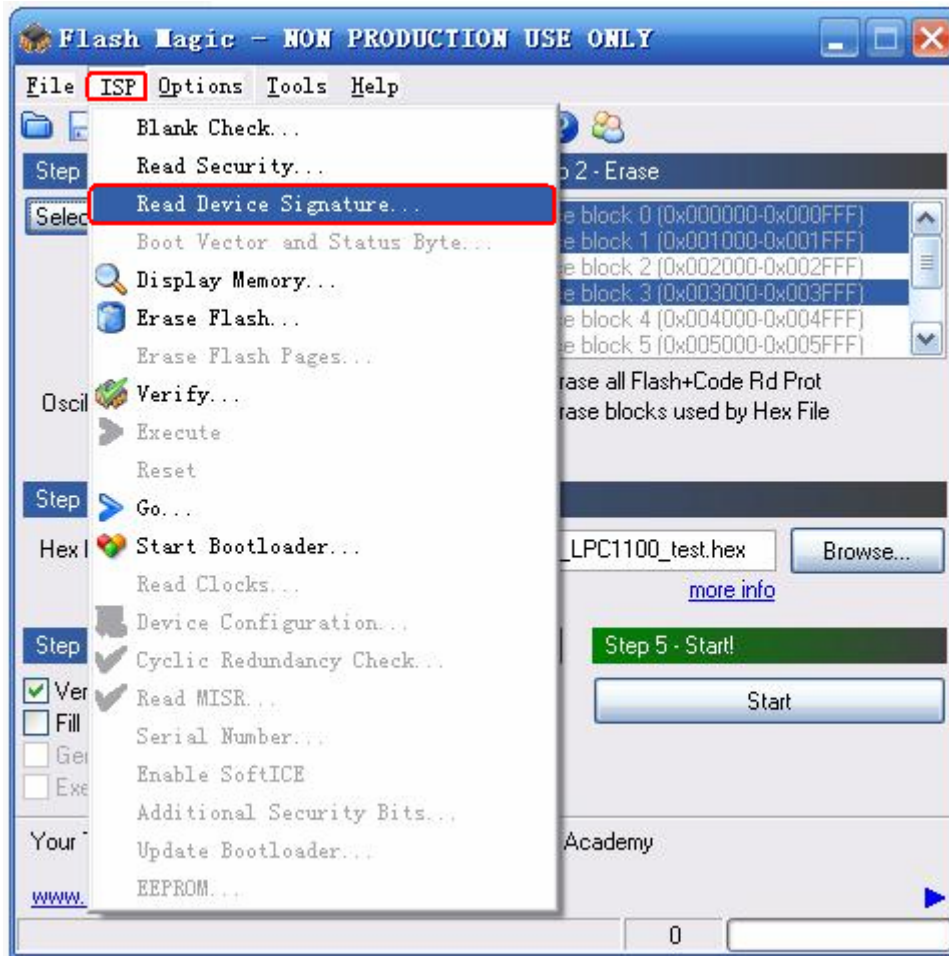
3) Click “Select Device” button and choose the device LPC1114/301. Click OK.



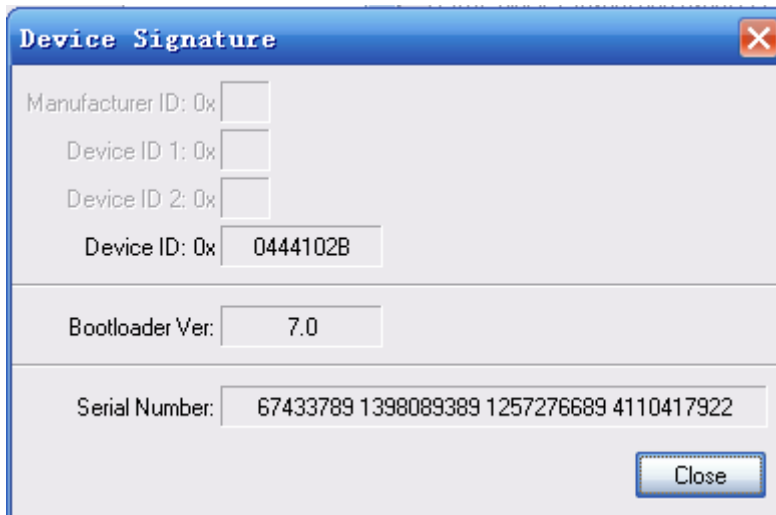
4) Then select serial port, set the baud rate, choose the interface type and fill in the oscillator frequencies.



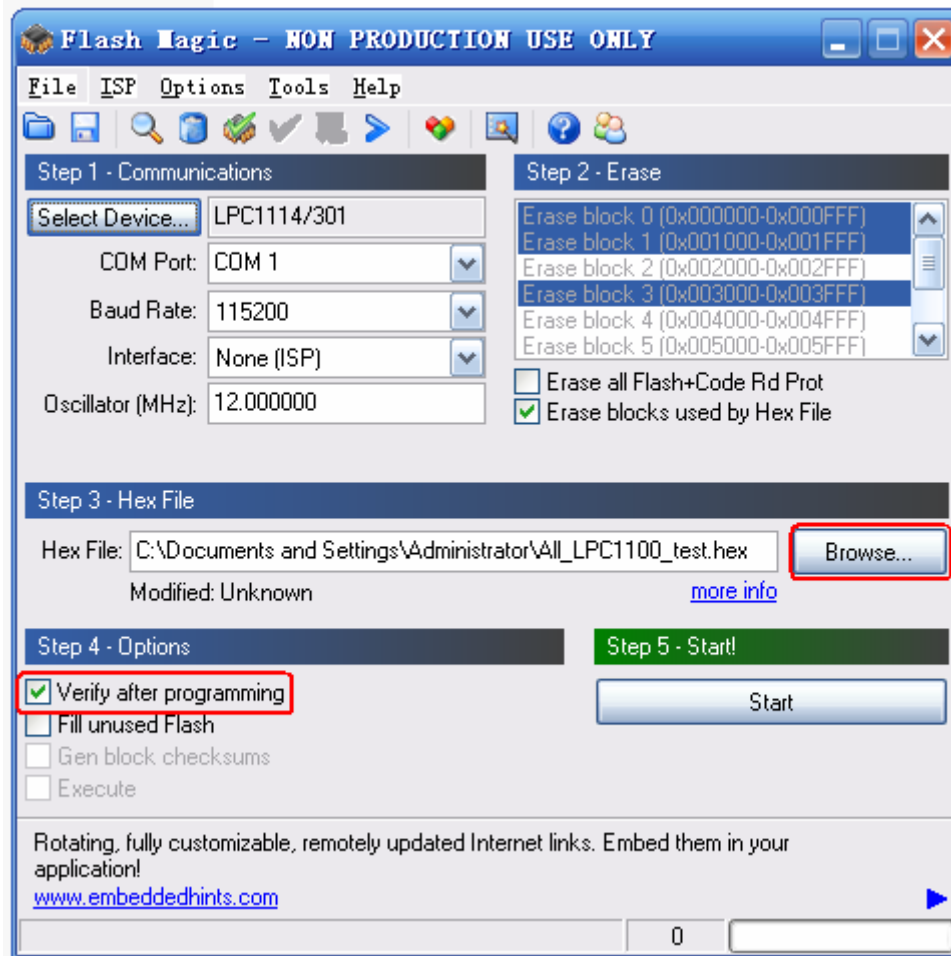
5. After configuration, click menu “ISP”->” Read Device Signature...” and read the device.



If read success, it displays the Device ID, Bootloader Version and Serial Number. See it as below.



6. After completing step 5, click “Browse...” button and choose the target HEX file. Ensure the option “Verify after programming” selected.



7. Click “start”button and start to download. After download complete, be sure the jumper JP2 disconnected. After reset, the development board start to run the application just downloaded.

Appendix A: After-sale Service

Embest is at your service, and we have special Technical Support Engineers to provide support and consultation in forms of telephone, E-mail, Fax and so on.

- TEL: +86-755-25621715
- FAX: +86-755-25616057
- Technical Support E-mail: support.en@embedinfo.com