

EM-LPC1300 Development Board User Guide

Rev. 1.0

Release: 2010-03-01



EMBEST CO., LIMITED

Address: Rm509, Luohu Science&Technology Building, 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	20100126	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.....	6
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.....	10
2.7.1 Compile the default factory routine.....	10
2.7.2 Download the default factory routine.....	10
CHAPTER 3 EM-LPC1300 DEVELOPMENT BOARD 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 SSP test.....	14
4.5 SysTick test.....	14
4.6 Timer32 test.....	14
4.7 UART test.....	15
4.8 USB CDC test.....	15
4.9 USB HID test.....	15
4.10 USB HID ROM test.....	16
4.11 USB Memory Storage test.....	16
4.12 USB Memory Storage Rom test.....	17
4.13 WDT test.....	17
4.14 I2C EEPROM test.....	18

4.15 I2C Temperature Sensor test 18

CHAPTER 5 SOFTWARE DEVELOPMENT AND EXAMPLES..... 19

5.1 MDK Introduction..... 19

5.2 Compile routine..... 19

 5.2.1 *Open a routine*..... 19

 5.2.2 *Compile the routine* 19

5.3 Debug and Download the routine using Emulator 21

 5.3.1 *Debug and Download the routine using ULINK2* 22

Chapter 1 Overview

1.1 The Microcontroller Introduction

The LPC13xx are ARM Cortex-M3 based microcontrollers for embedded applications featuring a high level of integration and low power consumption. The LPC13xx operate at CPU frequencies of up to 72 MHz. The peripheral complement of the LPC13xx series includes up to 32 kB of flash memory, up to 8 kB of data memory, USB Device, one Fast-mode Plus (FM+) I2C interface, one UART, four general purpose timers, and up to 42 general purpose I/O pins.

The LPC13xx operates in the -65 to $+150$ °C temperature range, from a 2.0 to 3.6 V power supply. A comprehensive set of power-saving mode allows the design of low-power applications.

Device Type	Flash (KB)	SRAM (KB)	USB 2.0	I ² C /FM+	A/D	Timer 32b/16bit	SSP	UART /485	Package
LPC1343FHN33	32	8	Device	1	8ch/10bit	2/2	1	1	HVQFN33
LPC1343FBD48	32	8	Device	1	8ch/10bit	2/2	1	1	LQFP48
LPC1342FHN33	16	4	Device	1	8ch/10bit	2/2	1	1	HVQFN33
LPC1313FBD48	32	8	—	1	8ch/10bit	2/2	1	1	LQFP48
LPC1313FHN33	32	8	—	1	8ch/10bit	2/2	1	1	HVQFN33
LPC1311FHN33	8	4	—	1	8ch/10bit	2/2	1	1	HVQFN33

1.2 Evaluation Boards Introduction

The EM-LPC1300 ARM-CM3 Board is a brand new, cost-effective but high-performance evaluation tool of the ARM Cortex-M3 based LPC13xx 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-M3 project.

You can use the Cortex-M3 based EM-LPC1300 Evaluation Board to generate and test application programs for the NXP LPC13xx 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

- LPC13xx ARM 32-bit Cortex™-M3 CPU, 72MHz maximum frequency
- 8 LED light tube, a power supply LED
- A mini-type USB socket, support for USB 2.0
- A BOOT button
- A RESET button
- 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

- SSP 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.
SSP	SSP LOOPBACK mode test.
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.
UART	Use UART to send and receive data through HyperTerminal.
USB_CDC	USB Virtual COM port example.
USB_HID	USB HID (Human Interface Device) demo.
USB_HID_rom	USB HID demo uses functions in rom.
USB_Mem	USB Memory Storage demo based on USB Mass Storage Class.
USB_Mem_rom	USB Memory Storage demo uses functions in rom.
WDT	WatchDog application example.
I2C_EEPROM	I2C EEPROM Read/write test
TemperatureSensor	I2C Temperature sensor test

Chapter 2 Getting Started

2.1 Product detailed list check

1. One EM-LPC1300 board
2. One crosslink serial cable
3. One USB cable
4. A product CD-ROM

2.2 Documents Description

File name / Item	Description	Attribute
EM-LPC1300 User Manual V1.0.pdf	User's manual	 671KB
EM-LPC1300 Board Schematic.pdf	Main board schematics.	 49KB
EM-LPC13xx Datasheet.pdf	Datasheet of LPC13xx	 981KB
EM-LPC13xx Reference Manual.pdf	Reference Manual of LPC13xx	 1.68MB
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: MDK4.01

2.4 Hardware resource requirements

When we test EM-LPC1300, 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 need 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 '5'. 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 ones 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 five.

```

*****
*                                     *
*   Welcome to Embest                 *
*                                     *
*   This is EM-LPC1300 test           *
*                                     *
*   arm.embedinfo.com                 *
*                                     *
*   0-LED test    1-BUTTON test    2-UART test    3-USB Memory test    *
*                                     *
*   4-I2C EEPROM test    5-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-LPC1300 --
-- LED1 blinky test --
    
```

● **1 option: test the BOOT button on the board.**

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

```

-- Basic GPIO Project V1.0 --
-- EM-LPC1300 --
-- 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-LPC1300 --

-- 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.

```
.df2eqf314jkj51455254/*-/4516445561...
```

- **3 option: USB Mass Storage test.**

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

```
-- Basic USB Mem Project V1.0 --

-- EM-LPC1300 --

-- USB Memory Storage test --
```

- 2) The USB Memory is automatically recognized by the host PC running Windows which will load a generic Mass Storage driver. Then you can find this device in PC hardware list. Open “my computer” icon on your desktop. You can find a new removable storage devices with name “LPC134x USB”. You can also open a readme.txt file from this device. The content of this file is the brief information about this demo. They are “This is a USB Memory Device demonstration for the NXP NXP13XX Board with Philips LPC1343.”

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

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

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

-- EM-LPC1300 --

-- 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.

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

- 1) After typing ‘5’, it displays the brief information about this test in the terminal.

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

-- EM-LPC1300 --

-- I2C Temperature Sensor test --

-I- Temperature Register value: 0x1100

-I- Configuration Register value: 0x0

-I- Tos Set Point Register value: 0x80
```

-I- Thyst Set Point Register value:0x1100

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.

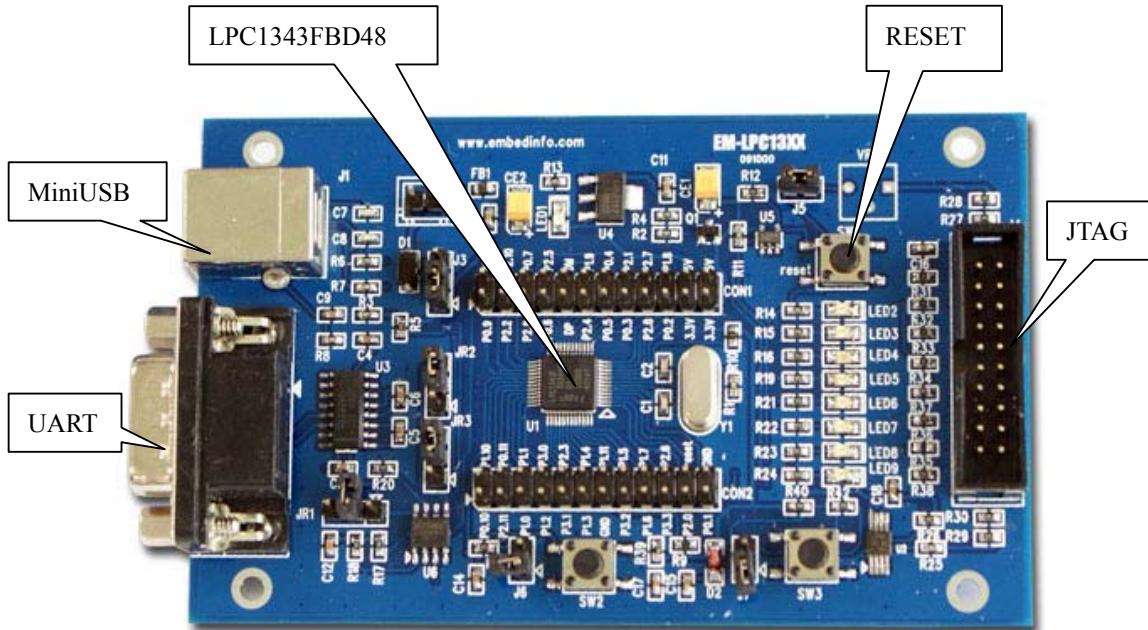
If you don't want to recompile the default factory routine, you can also use the image file All_LPC1300_test.hex or All_LPC1300_test.bin in the directory 02-Images of CD-ROM and download it to your board.

2.7.2 Download the default factory routine

- Use ULINK2 emulator, refer to 5.3 section and download.

Chapter 3 EM-LPC1300 Development Board Introduction

3.1 Board Interface Overview



J1	Mini USB B Connector	D4	LED3
J2	GND	D5	LED4
J3	UART	D6	LED5
J4	Processor External Round Cable	D7	LED6
J5	Processor External Round Cable	D8	LED7
J6	JTAG Interface	D9	LED8
U1	voltage regulator(1A output current)	K1	BOOT Button
U2	LPC1343FBD48 Chip	K2	Reset Button
D1	Power LED0	MN1	Temperature Sensor
D2	LED1	MN2	64Kbit Serial I ² C Bus EEPROM
D3	LED2	MN3	RS232 Transceiver

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 EM-LPC1300 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 is implemented to transport USB data, and it also supports USB full speed 2.0. In addition, supply 5V voltage to the Board through this Mini USB port.

3.3.4 EEPROM

A 64kbit EEPROM is connected to the I2C bus in EM-LPC1300 board.

3.3.5 LM75 Temperature Sensor

A LM75 Temperature Sensor is connected to the I2C bus in EM-LPC1300 board.

3.3.6 LED

EM-LPC1300 provides 8 LEDs D2...9, they respectively connect with PIO2.0...7 IO pins, and they can be used for user output. It also contains a power LED D0.

Chapter 4 Software Resources Introduction

There are 14 test cases in the CD-ROM provided. The directory is 03-software\examples. You can find 15 folders in this directory. The Common folder contains common source files and head files. The other 14 items are all test cases used to test the peripherals or functions of the EM-LPC1300 board.

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 EM-LPC1300 UserManual 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. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic Blinky Project V1.0 --
-- EM-LPC1300 --
-- LED1 blinky test --
```

Then, the LED1 of the board will blinky.

- Reference manual: EM-LPC13xx 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-LPC1300 --
-- 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-LPC13xx 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-LPC1300 --

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

After reset, system pLL, USB PLL, Watchdog oscillator, ADC and BOD will power-down. You can push "BOOT" button on the board to wakeup. After wakeup, LED1 will blinky.

- Reference manual: EM-LPC13xx Reference Manual.pdf

4.4 SSP test

- Source code location: 03-Software\Examples\04-SSP
- Test description: The SSP project is a simple program to test SSP LOOPBACK mode on the EM-LPC1300 board. 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-LPC1300 --

-- Synchronous Serial Communication test --

SSP Loopback Test success!
```

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

- Reference manual: EM-LPC13xx Reference Manual.pdf

4.5 SysTick test

- Source code location: 03-software\Examples\05-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. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic SysTick Project V1.0 --

-- EM-LPC1300 --

-- SysTick delay test --
```

LED1 on the board start to blinky immediately after reset.

- Reference manual: EM-LPC13xx Reference Manual.pdf

4.6 Timer32 test

- Source code location: 03-software\Examples\06-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. Open PC HyperTerminal and push RESET button, you will see the phenomenon below.

```
-- Basic Timer32 Project V1.0 --
```

```
-- EM-LPC1300 --
-- LED1 blinky delay controled by timer32 test --
```

LED1 on the board start to blinky immediately after reset.

- Reference manual: EM-LPC13xx Reference Manual.pdf

4.7 UART test

- Source code location: 03-software\Examples\07-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-LPC1300 --
-- Please input any key on the keyboard --
```

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

- Reference manual: EM-LPC13xx Reference Manual.pdf, MAX3232CSE.pdf

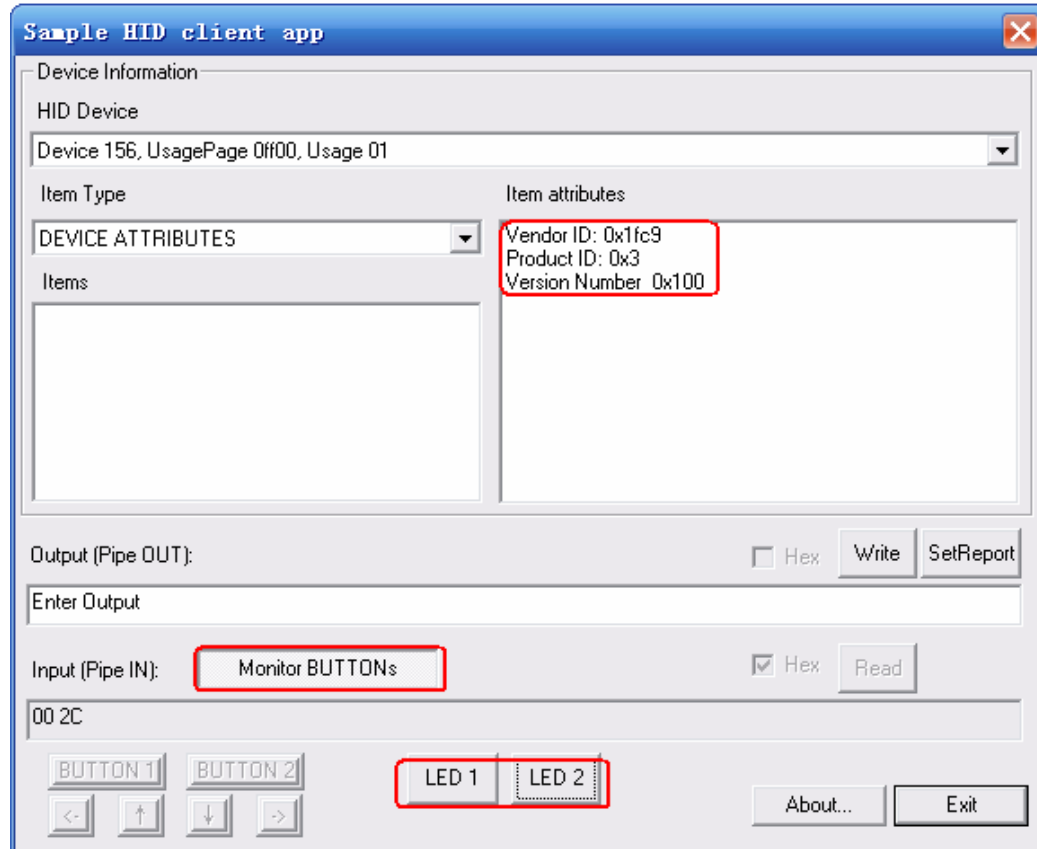
4.8 USB CDC test

- Source code location: 03-software\Examples\08-USB_CDC
- Test description: The Virtual COM port project is a demo program for using the EM-LPC1300 board. It demonstrates an USB Virtual COM port based on a Windows USB host driver (usbser.sys).
- Test phenomenon: Connect board with PC COM using USB cable. Build the program and download it inside the evaluation board. Push RESET button. The PC will install a virtual COM port on the PC (see Driver Installation). You can find drive file in directory drive/lpc134x-vcom.inf. After installation an additional port "LPC13xx USB VCom Port(COMx)" can be found under System/Hardware/Device Manager/Ports(COM&LPT). Number "x" is not fixed as different PC configuration may have different "x" displayed on the device manager. The USB host driver assigns "x" dynamically based on the existing COM port configuration of the system. Open two Hyperterminal windows. One with "LPC13xx USB VCom Port(COMx)". One with "Communications Port (COM1)". Connect PC port COM1 to the comport on the board and open "COM1" and "COMx". Data from COM1 will be echoed on "COMx" and visa versa. So, this is bi-directional communication between the physical COM port 1 on the board and the virtual COM port COMx on host PC.
- Reference manual: EM-LPC13xx Reference Manual.pdf

4.9 USB HID test

- Source code location: 03-software\Examples\09-USB_HID
- Test description: The HID(Human Interface Device) project is a demo program using the EM-LPC1300 board.
- Test phenomenon: Connect board with PC COM using USB cable. Build the program and download it inside the evaluation board. Push RESET button. The USB HID is recognized by the host PC running Windows which will load a generic HID driver. The board LEDs can then be accessed from the PC through a custom HID Client Program(hidTest.exe). You can check the Vendor ID, Product ID, Version

Number about this USB HID device. See it as figure below.



Click Monitor BUTTONs, then you can lighten LED1...4 on the board by clicking LED1 button and control LED2 on or off. by clicking LED2 button.

- Reference manual: EM-LPC13xx Reference Manual.pdf

4.10 USB HID ROM test

- Source code location: 03-software\Examples\10-USB_HID_rom
- Test description: This project is also a USB HID demo program using the EM-LPC1300 board. The only difference is that pll and pin init function program used are all placed in the BOOT ROM. They are download by ULINK2 or other debugging tools previously and their memory address are determined at the compile time.
- Test phenomenon: Connect board with PC COM using USB cable. Build the program and download it inside the evaluation board. Push RESET button. The USB HID is recognized by the host PC running Windows which will load a generic HID driver. The board LEDs can then be accessed from the PC through a custom HID Client Program(hidTest.exe). The more details are same to 4.9.
- Reference manual: EM-LPC13xx Reference Manual.pdf

4.11 USB Memory Storage test

- Source code location: 03-software\Examples\11-USB_Mem
- Test description: The Memory project is a demo program for the Embest EM-LPC1300 Board using the NXP LPC1343 Microcontroller. It demonstrates an USB Memory based on USB Mass Storage Class.

- Test phenomenon: Connect board with PC COM using USB cable. Build the program and download it inside the evaluation board. Push RESET button. The USB Memory is automatically recognized by the host PC running Windows which will load a generic Mass Storage driver. Then you can find this device in PC hardware list. Open “my computer” icon on your desktop. You can find a new removable storage devices with name “LPC134x USB”. You can also open a readme.txt file from this device. The content of this file is the brief information about this demo.
- Reference manual: EM-LPC13xx Reference Manual.pdf

4.12 USB Memory Storage Rom test

- Source code location: 03-software\Examples\12-USB_Mem_rom
- Test description: The Memory project is a demo program for the Embest EM-LPC1300 Board using the NXP LPC1343 Microcontroller. It demonstrates an USB Memory based on USB Mass Storage Class. The only difference is that pll and pin init function program used are all placed in the BOOT ROM. They are download by ULINK2 or other debugging tools previously and their memory address are determined at the compile time.
- Test phenomenon: Connect board with PC COM using USB cable. Build the program and download it inside the evaluation board. Push RESET button. The USB Memory is automatically recognized by the host PC running Windows which will load a generic Mass Storage driver. Then you can find this device in PC hardware list. Open “my computer” icon on your desktop. You can find a new removable storage devices with name “LPC134x USB”. You can also open a readme.txt file from this device. The content of this file is the brief information about this demo. The detail phenomenon are same to 4.11.
- Reference manual: EM-LPC13xx Reference Manual.pdf

4.13 WDT test

- Source code location: 03-Software\Examples\13-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-LPC1300 --
-- 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...8 are all turned on and LED1 resume blinky. Description Information is displayed in the terminal.

```
PIOINTO_IRQHandler!
-- Basic WDT Project V1.0 --
-- EM-LPC1300 --
-- Feed watchdog timer to prevent it from timeout test --
```

WatchDog Reset!

- Reference manual: EM-LPC13xx Reference Manual.pdf

4.14 I2C EEPROM test

- Source code location: 03-Software\Examples\14-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-LPC1300 --
-- I2C Read/Write EEPROM test --

I2C EEPROM test success!
```

- Reference manual: EM-LPC13xx Reference Manual.pdf, M24C64.pdf

4.15 I2C Temperature Sensor test

- Source code location: 03-Software\Examples\15-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-LPC1300 --
-- 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

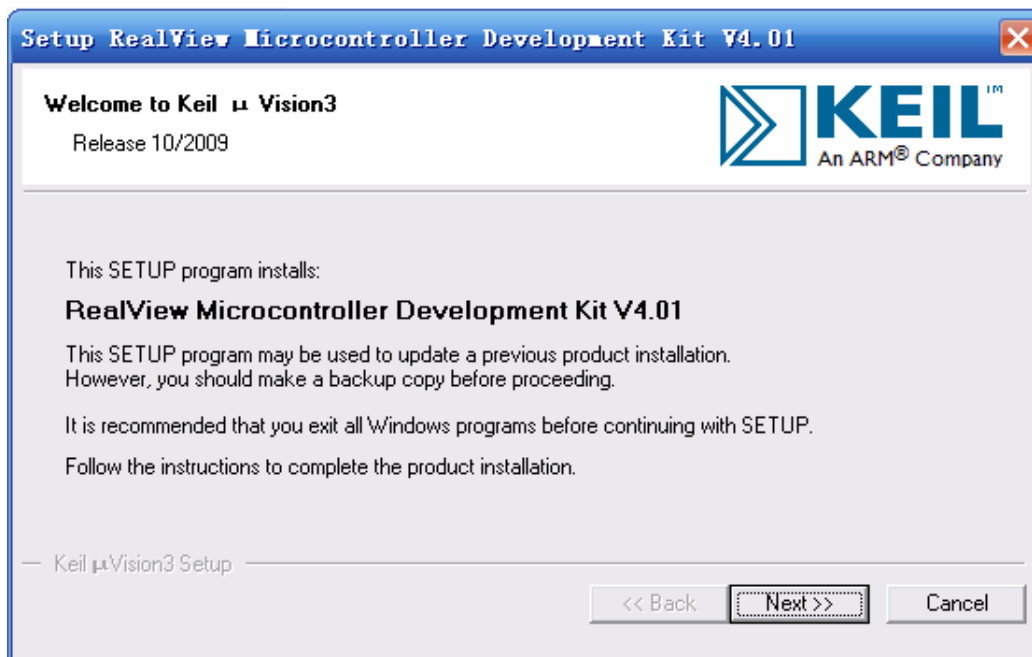
Temperature Sensor test success!
```

- Reference manual: EM-LPC13xx 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. MDK4.01 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 EM-LPC1300 Evaluation Board in the directory of 04-tools\Realview MDK4.01, 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:



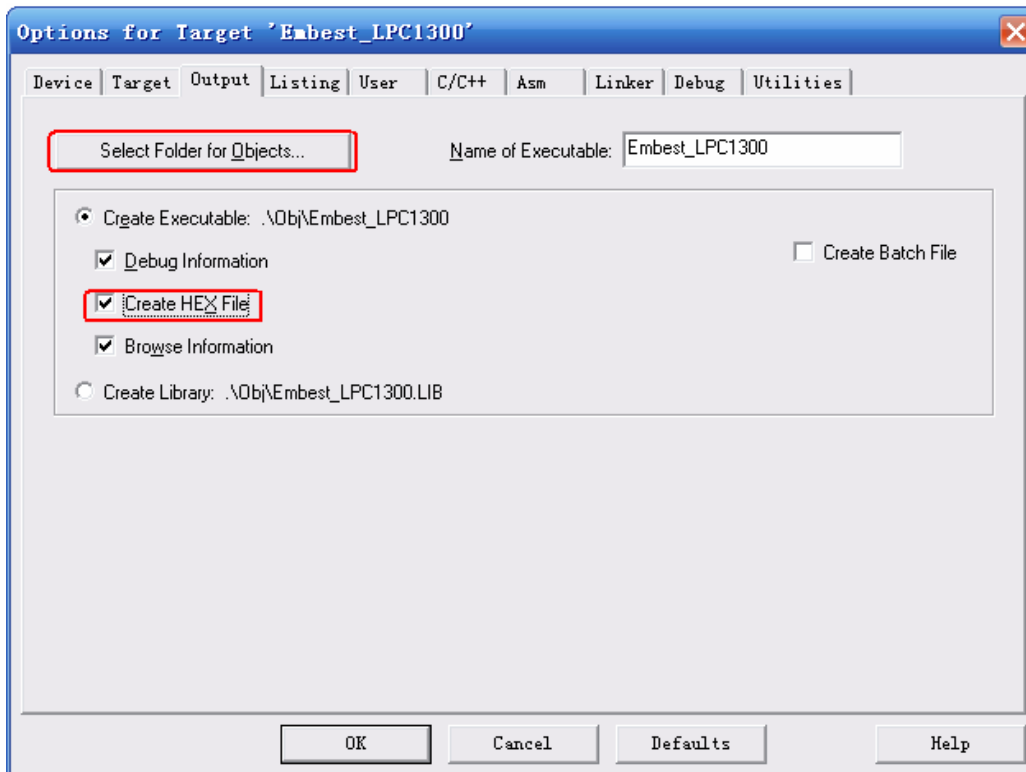
5.2 Compile routine

5.2.1 Open a routine

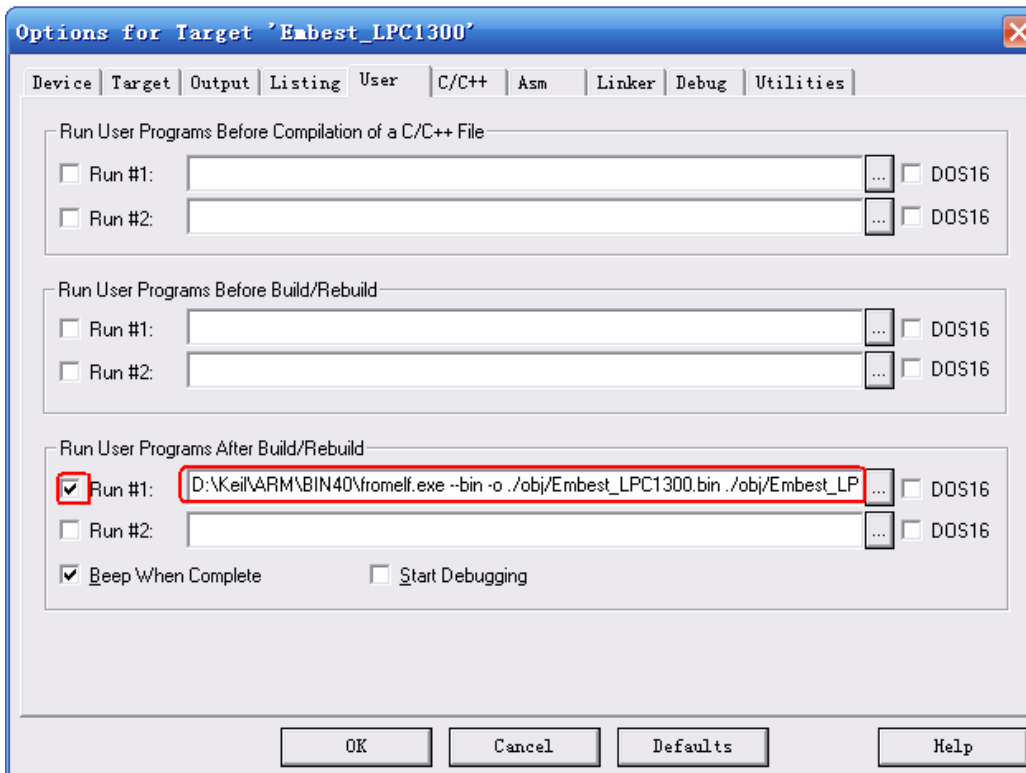
Enter in the folder of 03-software\Examples; Open the Embest_LPC1300.uvproj file in the project directory of one routine. For example: \03-Software\Examples\01-Blinky\project\Embest_LPC1300.uvproj.

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.

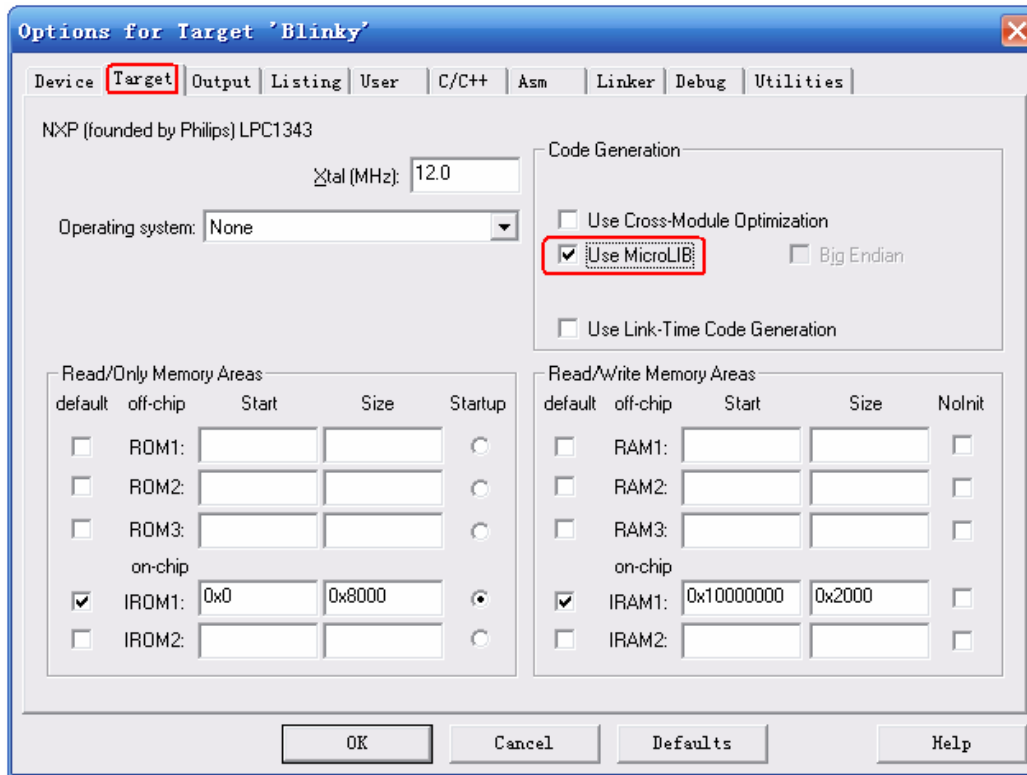


2) If you need the .bin format file, then you can configure MDK to build the bin file. Or you can skip this step.

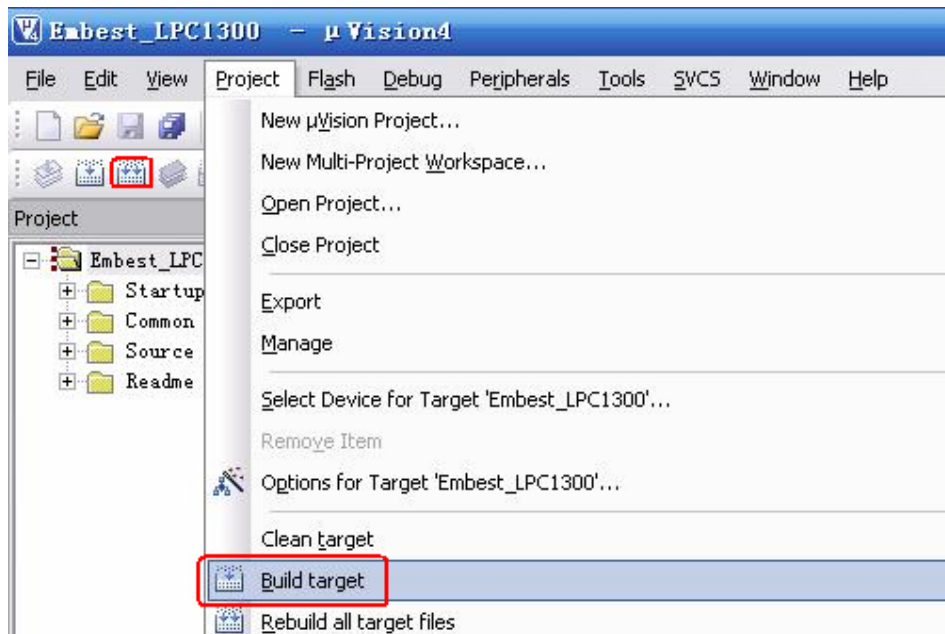


- D:\Keil\ARM\BIN40\fromelf.exe specify the path of fromelf.exe, it will convert axf file to bin file
- --bin -o export bin file
- ./obj/Embest_LPC1300.bin the directory and file name of the bin file you want to create
- ./obj/Embest_LPC1300.axf the directory and file name of the axf file you want to 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.



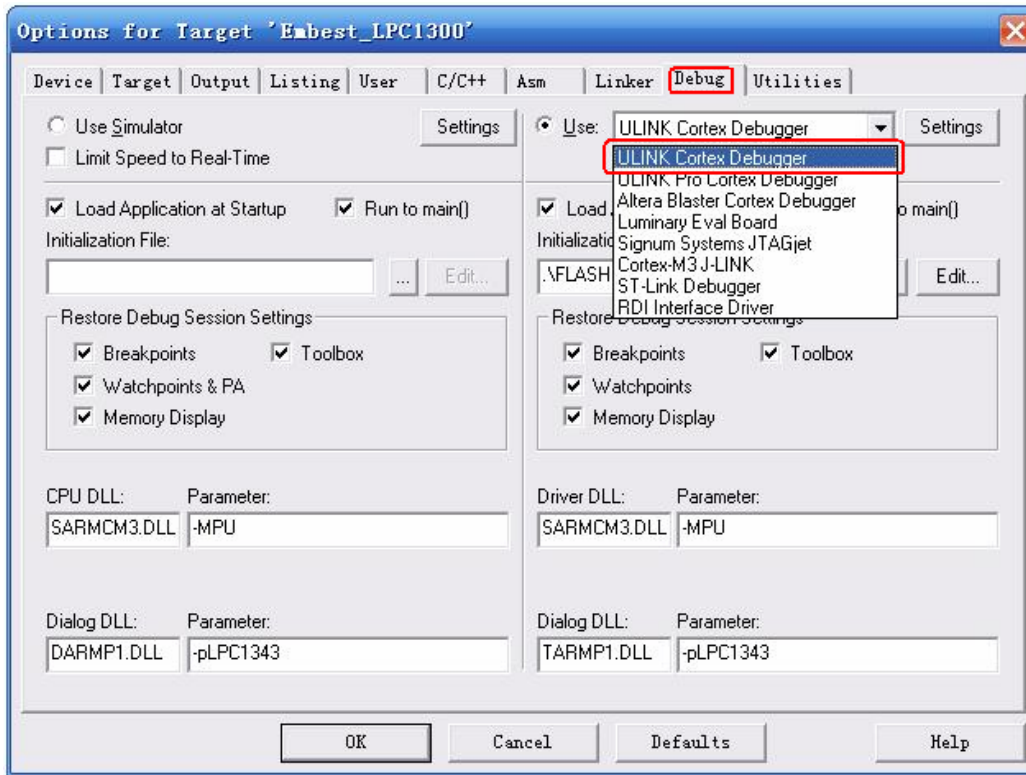
5.3 Debug and Download the routine using Emulator

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

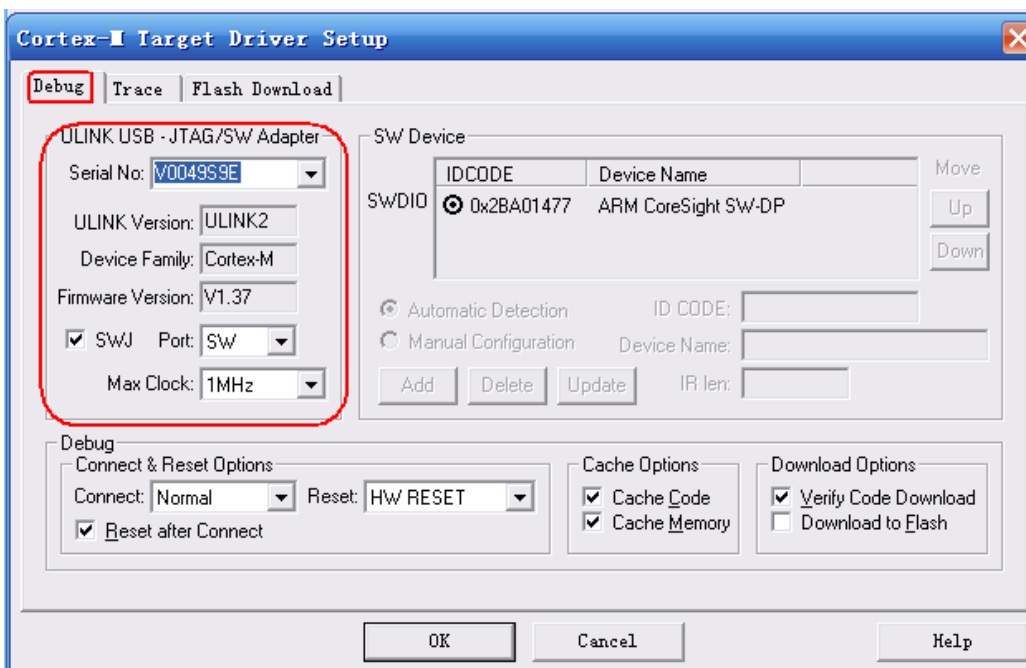
1) Choose Emulator



2) Check the ULINK2, optional.

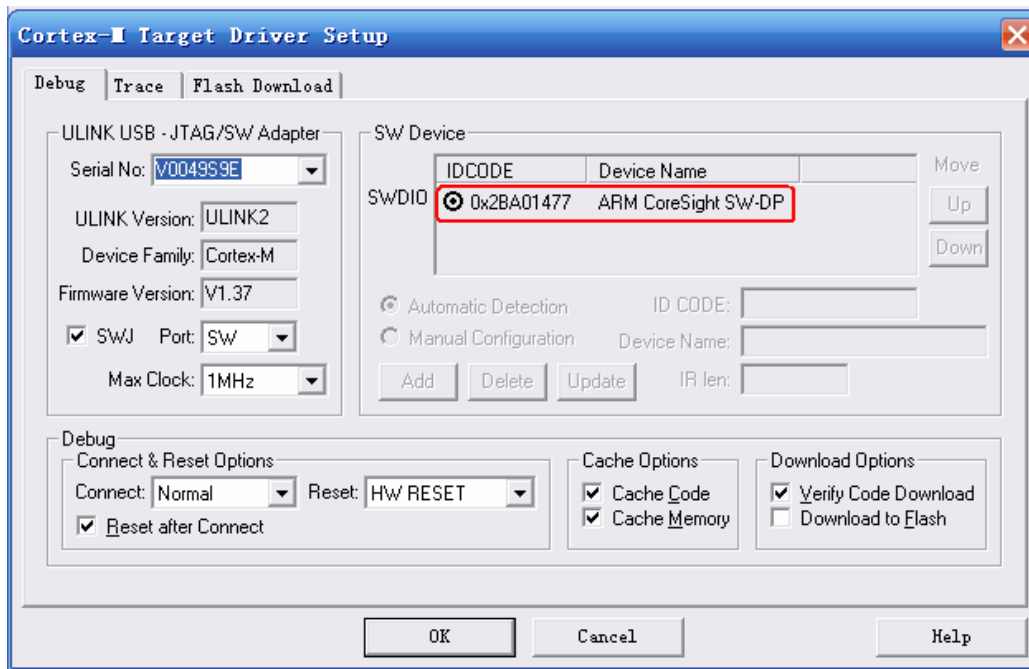
When ULINK2 connects to the Development Board, if the RUN and COM indicator lights first change to blue and then go out, and the USB indicator light has always been red, this proves that ULINK2 is no problem.

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.

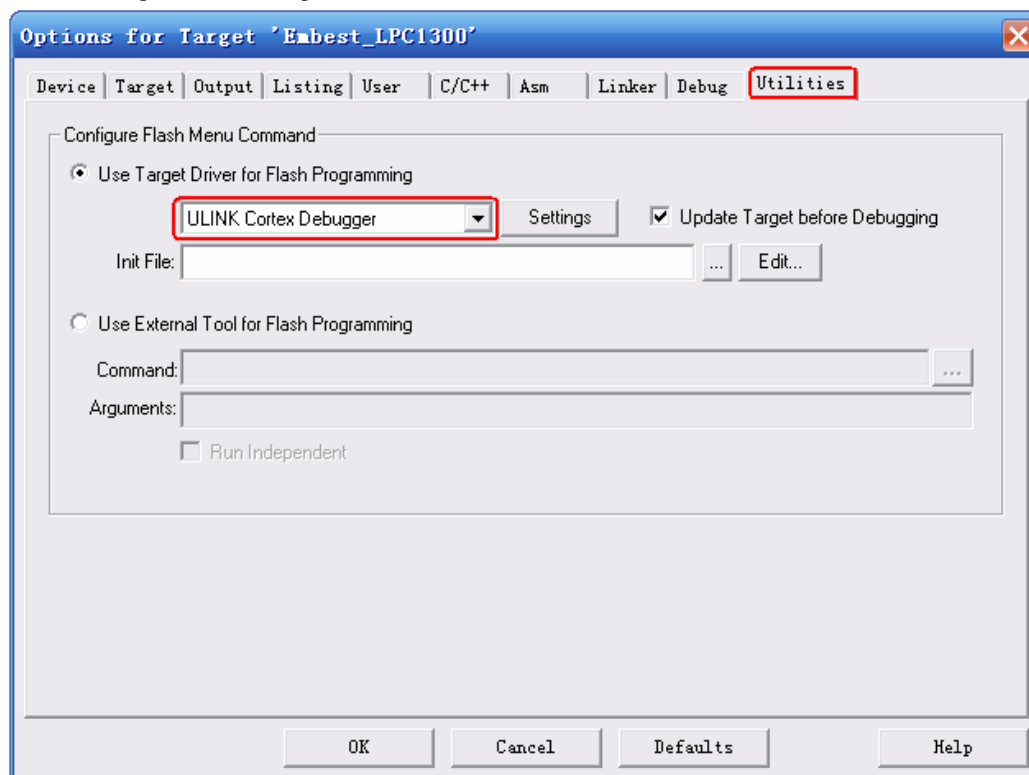


3) Check that whether ULINK2 can detect the development board or not, optional.

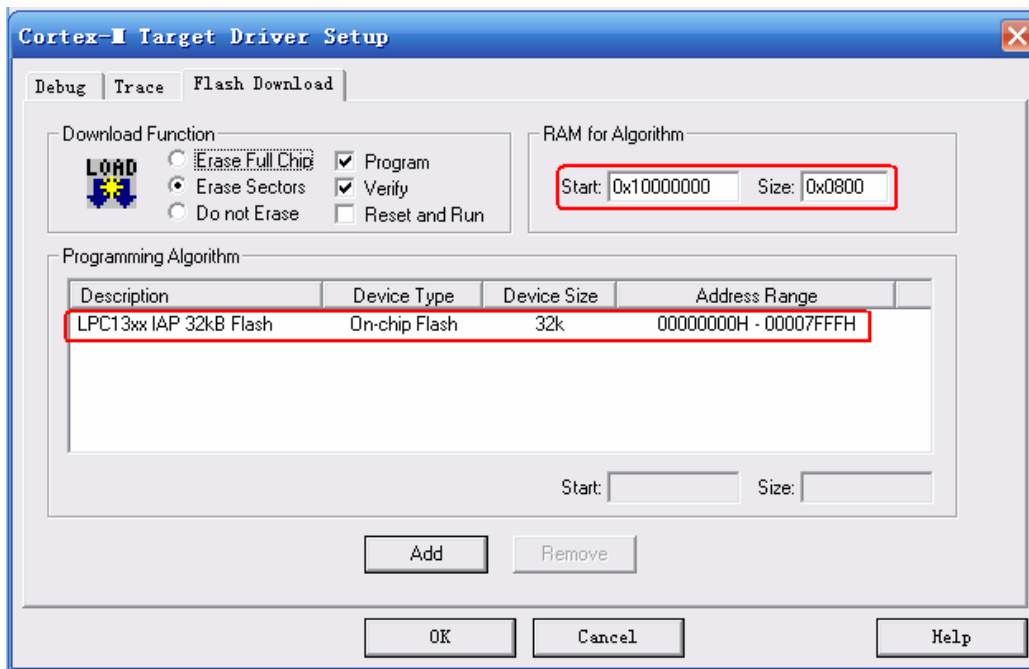
Click the Settings button in the Debug TAB, if the red marked part appears, it proves that ULINK2 has detected the development board.



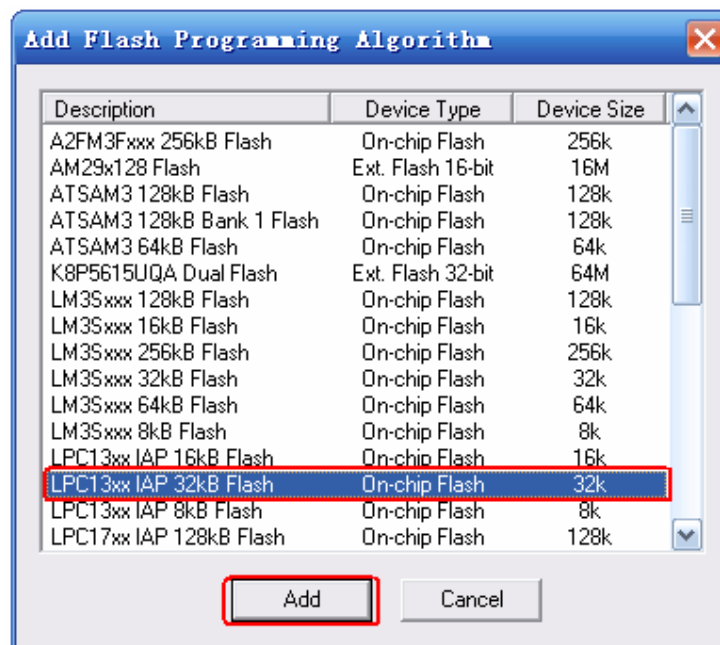
4) Set the Flash Programmer, configure the Utilities TAB.



Then click Settings button, it appears:

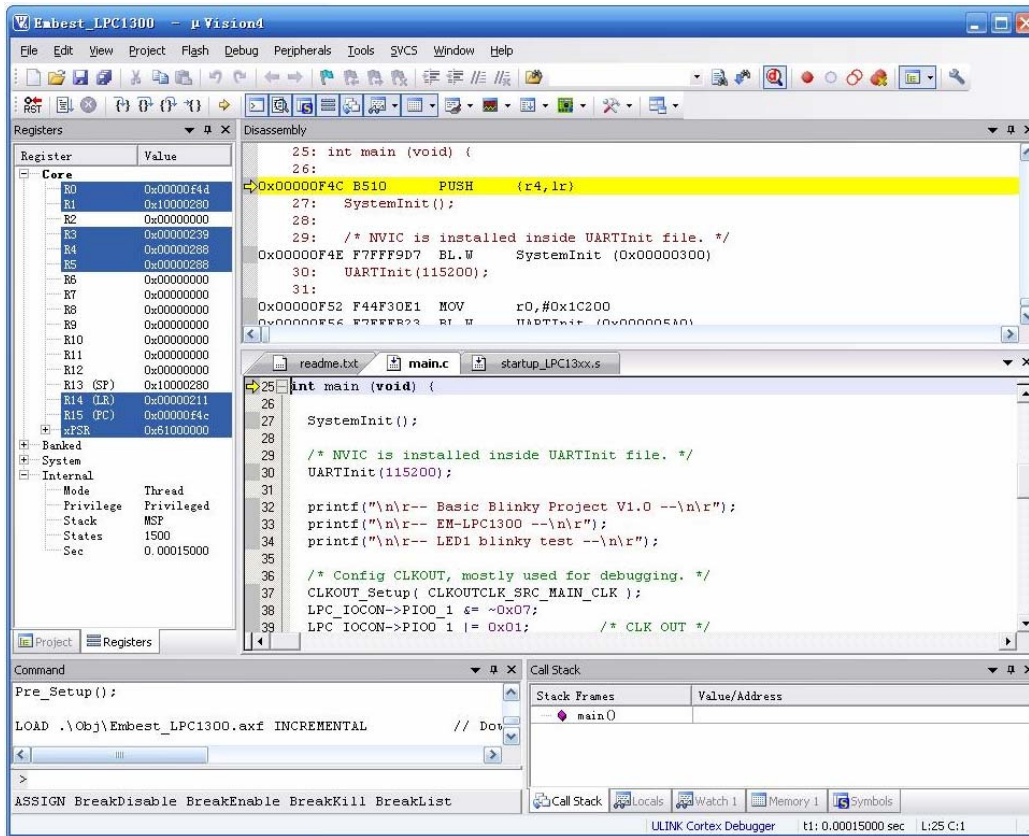


If nothing is in the Programming Algorithm, then you should add the corresponding Flash programming algorithm by clicking the Add button, as follows:

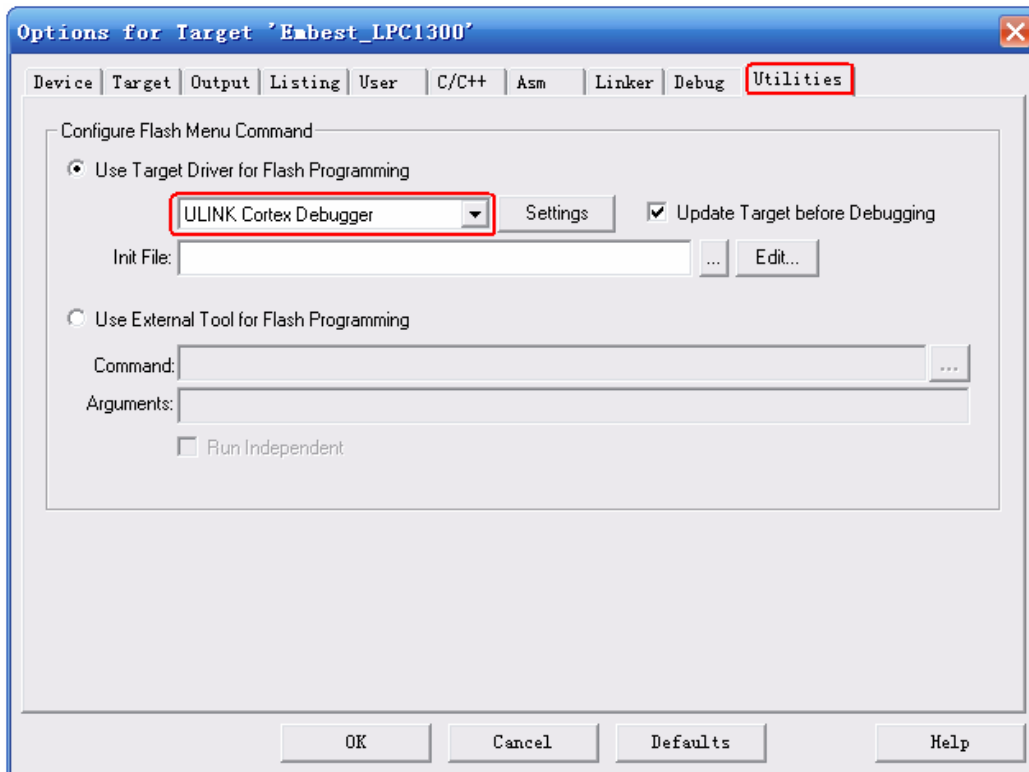


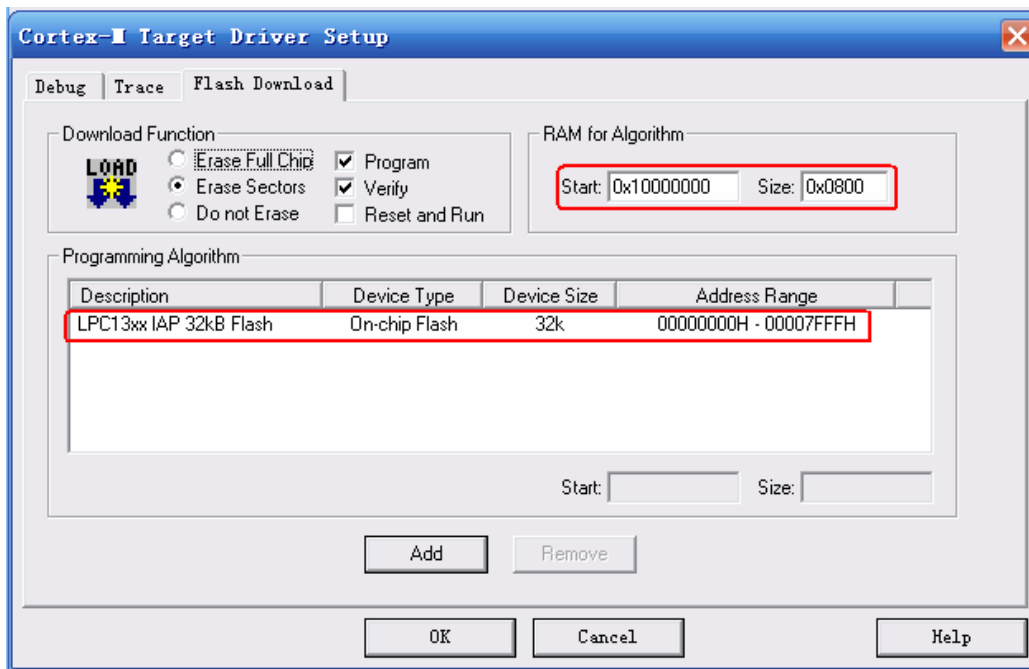
After you choose a Flash Programming Algorithm, then Click Add button.

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:

