NXP LPC4330 Xplorer: Blinky Lab

MDK Version 5 Tutorial



Spring 2015, V 1.0

Abstract

5.

This tutorial shows how to create the Blinky project for the Cortex-M4 in the LPC4330 on the NXP LPC4330 Xplorer development board.

Create a New Project for the Xplorer Board

- I. In the main μ Vision menu, select **Project** \rightarrow **New** μ **Vision Project...** The 'Create New Project' window opens up.
- 2. Create a new directory called **Blinky** and enter **Blinky** for the File name. Press **Save**.
- 3. In the 'Select Device for Target' window select LPC4330:Cortex-M4 and press OK.
- 4. In the 'Manage Run-Time Environment' window select the following Software Components:
 - a. CMSIS:RTOS (API):Keil RTX
 - b. Board Support (LPC4330-Xplorer):LED (API):LED
 - c. When done, press **Resolve** and afterwards **OK**.
 - The Project window should look like this —

Add user code templates main.c and Thread.c

- 6. Right-Click Source Group I and select Add New Item to Group 'Source Group I'...
- In the upcoming window, select User Code Template and then expand CMSIS. Select CMSIS-RTOS 'main' function and click Add:

Add New Item to Group 'Source Gro	up 1'	alast	×
C File (.c)	Add template nie(s) to the pr	oject.	
**	Component	Name	
C++ File (.cpp)	🖃 🚸 CMSIS		<u> </u>
	RTOS:Keil RTX	CMSIS-RTOS 'main' function	
	RTOS:Keil RTX	CMSIS-RTOS Mail Queue	
h Header File (.h)	RTOS:Keil RTX	CMSIS-RTOS Memory Pool	
	RTOS:Keil RTX	CMSIS-RTOS Message Queue	
Text File (.txt)	RTOS:Keil RTX	CMSIS-RTOS Mutex	
Image File (*)	-RTOS:Keil RTX	CMSIS-RTOS Semaphore	
anage rie (.)	-RTOS:Keil RTX	CMSIS-RTOS Thread	
User Code Template	RTOS:Keil RTX	CMSIS-RTOS Timer	
	RTOS:Keil RTX	CMSIS-RTOS User SVC	-
ype: User Code Templa lame: osObjects.h main .ocation: C:\01_workspace	.c //DKv5\Atmel\Binky	Close	



8. Repeat the process and choose **CMSIS-RTOS Thread**. You now should see a *main.c* and a *Thread.c* file below the Source Group 1.

Configure CMSIS-RTOS RTX

- 9. Open RTX_Conf_CM.c, select the Configuration Wizard tab and press Expand All.
- 10. Change the **RTOS Kernel Timer input clock frequency [Hz]** to **180000000** as the LPC4330 runs on 180 MHz.

Add the Blinky code

II. Change *main.c* as follows:

```
#define osObjectsPublic
                                        // define objects in main module
#include "osObjects.h"
                                        // RTOS object definitions
#include "LPC43xx.h"
                                        // Device header
#include "Board LED.h"
                                        // ::Board Support:LED
extern int Init blink LED (void);
int main (void) {
 osKernelInitialize ();
                                       // initialize CMSIS-RTOS
 LED Initialize();
 Init blink LED();
                                        // start thread execution
 osKernelStart ();
 while(1);
}
```

12. Change *Thread.c* as follows:

```
#include <cmsis os.h>
                                          // CMSIS RTOS header file
#include "Board LED.h"
                                          // ::Board Support:LED
void blink LED (void const *argument);
                                                        // thread function
osThreadId tid blink LED; // thread id
osThreadDef (blink LED, osPriorityNormal, 1, 0); // thread object
int Init blink LED (void) {
  tid blink LED = osThreadCreate (osThread(blink LED), NULL);
  if(!tid blink LED) return(-1);
  return(0);
}
void blink_LED (void const *argument) {
  while (1) {
   LED On (0);
   osDelay(500);
   LED Off (0);
   osDelay(500);
   osThreadYield();
  }
ł
```

Debug Adapter

In this workshop we are using the **LPC-Link2 debug adapter** with J-Link firmware. You will need a Mini-USB cable to connect the LPC-Link2 with the PC that runs the development tools. The LPC-Link2 Debug Adapter should be configured as described below.

Download and Install J-Link Software & Documentation Pack for Windows

Visit <u>www.segger.com/jlink-software.html</u> and download the latest version of the J-Link software and documentation pack for Windows. The ZIP file contains an EXE file that needs to be installed on your computer before the configuration of the LPC-Link2 that is described in the next step.

Configure the LPC-Link2 as J-LINK debugger

Visit <u>www.lpcware.com/lpclink2</u> to obtain the latest LPC-Link Configuration Tool. After installation, run the tool and follow the on-screen instructions to program your LPC-Link2 with the "LPC-Link2 J-Link debugger" firmware.

E DFU programmer/LPC-Link 2 Configuration Tool		
File Help		
Select an image Clark2 24thk debugger Clark2 24thk debugger image Urc Link 2 Segger J-Link debugger support to LPC-Link 2 acrists Instructions: Remove Jumper JP1 per the image. Connect the board is 0458 to the system running this program. Pers the Program Without to program this image Into the bord (in 0458 to the system running this image Into the bord (in 0458 to the system r	Staf Program Detension 1: Press the program button to program the board	Verly Program Operation
Ado program currently selected Link2 image on board detection Enable run fine debug messages	Program the Link2 with the currently selected image	
PROG mode v NO CONN		

Note: This software requires the .NET framework to be present on your PC.

Configure the Target Options

- 13. Click on 🔊 or press ALT+F7
- 14. Enter the following in the **Read/Only Memory Areas:**
- 15. Select the **Debug** tab and choose **J-LINK / J-TRACE Cortex**. Press **Settings**.
- 16. Configure **Debug** and **Flash Download** as follows:

Cortex JLink/JTrace Target Driver Setup	Cortex JLink/JTrace Target Driver Setup
Debug Trace Rash Download J-Link / J-Trace Adapter JTAG Device Chain SN: provide: JLink / J-Trace Adapter Device: J-Link / J-Trace Adapter IDCODE Device: J-Link / J-Trace Adapter IDCODE Device: J-Link / J-Trace Adapter IDCODE Device: J-Link / J-Trace Adapter Up Device: J-Link LPCXpresso Up HW: V1.00 dt: V4.96h	Debug Trace Rash Download Download Function Logad C Erase Full Chip V Program C Erase Sectors V Verify C Do not Erase Reset and Run Descence Machine
FW: J-Link LPCXpresso V2 complet Pott: Max Clock: Vidio G JTAG 10MHz Auto Chk Manual Configuration Delete Update IR IR Connect: Normal Reset Reset Process Cache Options Connect: Normal Reset Reset Process Conhect Code Process Condot Code Process Connect: Normal Reset Reset Reset Process Connect: Verify Code Download Download Options Process Connect: Verify Code Download Download to Rash Interface TCP/IP Network Network Settings	Description Device Size Device Type Address Range LPC18xv43xx S25FL032 SPIFI 4M Ext. Rash SPI 1400000H - 143FFFFH Use the Add button to add the LPC18xx/43xx S25FL032 SPIFI Size: algorithm Add Remove
Scan IP-Address Pot (Auto: 0) State: ready 127 . 0 . 0 . 1 0 OK Cancel Apply	OK Cancel Apply

Read/Only Memory Areas

ROM2

ROM3:

Start

ROM1: 0x14000000

Size

0x10000

Startup

œ

default off-chip

 $\overline{\mathbf{v}}$

П

Build the Project and run it on the Target

- 17. Right-click on *startup_LPC43xx.s* (*Startup*) and choose **Options for Component Class 'Device'**
- 18. Select the C/C++ tab and enter USE_SPIFI=1 in the Define box and click OK:

🔀 Options for Component Class 'Device'		×
Software Component Software Component GPIO SCU Startup	System Startup for NXP LPC4300 Series Bundle: Vendor: Keil Properties Memory C/C++ Asm Preprocessor Symbols Define: USE_SPIFI=1 Undefine:	

- 19. Go to File \rightarrow Save All
- 20. Go to **Project** → **Build Target** (or press **F7**)
- 21. Connect the Mini-USB cable to the LPC-Link 2 and the Micro-USB cable to a USB connector on the LPC4330-Xplorer board
- 22. Connect the two boards with the flat cable (make sure the red mark on the cable is on the right side of the connectors of each board)
- 23. Go to **Flash** \rightarrow **Download** to flash the project to the target
- 24. Go to **Debug** → **Start/Stop Debug Session** (or press **CTRL+F5**)
- 25. Go to **Debug** \rightarrow **Run** (or press **F5**) to run the project on the target. LED **D3** will start flashing.

More information

For a detailed description on how to setup multi-core projects on the LPC4430, please refer to AN 272: <u>http://www.keil.com/appnotes/docs/apnt_272.asp</u>