**NXP Semiconductor** 

# Getting started with LPCOpen: Running the demo applications

By: Technical Information Center



# About this document

This document will explain the steps to use LPCXpresso with the LPCOpen projects for your preferred device and platform.

The steps described in the document were done using the LPC54102 MCU like the one in the LPCXpresso Board for the LPC54100 family of MCUs, but the same principles are applicable to any LPC MCU.

# Software versions

The steps described in this document are valid for the following versions of the software tools:

- o LPCXpresso v8.1.4
- o LPCOpen v3.xx

# Contents

1. Overview and concepts	3
1.1 LPCOpen	3
1.1.1 Core driver library	3
1.1.2 Middleware	3
1.1.3 Examples	3
1.1.4 Using LPCOpen with an RTOS	3
2. Running the demo applications	4
2.1 Downloading a LPCOpen package	4
2.2 Importing the LPCOpen examples	6
2.3 Building and debugging blinky project1	1
Appendix A - References1	6

## 1. Overview and concepts

#### 1.1 LPCOpen

LPCOpen is an extensive collection of free software libraries (drivers and middleware) and example programs that enable developers to create multifunctional products based on LPC microcontrollers. Access to LPCOpen is free to all LPC developers.

#### **1.1.1 Core driver library**

The core driver library contains common chip-specific drivers. It is divided into two layers: a chip driver layer containing drivers optimized for a specific device or family, and a board layer containing board-specific functions and low-level setup code.

#### 1.1.2 Middleware

LPCOpen includes access to key middleware elements:

- SEGGER emWin graphics object library
- SWIM graphics library
- o LWIP open-source networking stack source code and examples
- USB libraries: USBD device library for all LPC devices and LPCUSBLib open-source USB host stack
   both use the USB ROM APIs or a Flash-based library

#### 1.1.3 Examples

LPCOpen includes an extensive set of examples designed to illustrate how to use core driver library functions and middleware. Examples demonstrate use of:

- Peripherals such as I2C, UART, SPI, and GPIO
- o USB host and device
- Ethernet use with an IP stack (LWIP)
- o emWIN and SWIM graphics libraries

#### **1.1.4 Using LPCOpen with an RTOS**

LPCOpen libraries are RTOS agnostic and can be used with a simple control loop. Examples are also included in each software download package for use with FreeRTOS.

## 2. Running the demo applications

### 2.1 Downloading a LPCOpen package

 Got to the link <u>http://www.nxp.com/lpcopen</u> and select the corresponding family of microcontrollers, for this example we chose the LPC54100 series:

LPCOpen ports for LPC Cortex-M series microcontrollers

- LPC800 Series
  LPC1100 Series
  LPC1300 Series
  LPC1500 Series
  LPC1700 Series
  LPC1800 Series
- LPC4000 Series
- LPC4300 SeriesLPC54100 Series
- LPC54100 Series
   LPC54110 Series
- On the next page you will find the latest available LPCOpen software package downloads along with older versions of the packages. Look for your board/device and click on the software download link, for this example we chose the package for the LPCXpresso IDE but there is also a package for the IAR and Keil IDEs:

Latest available LPCOpen v3.xx		Legacy LPCOpen 2.xx Older versions of LPCOpen 2.xx			2.xx	
atest available	e LPCOpen v3.x	x software pa	ckage downloads			
Supported Board(s) /Devices(s)	Software Download link	Toolchain <sup>1</sup>	Documentation download link <sup>2</sup>	Debugger(s) <sup>s</sup>	Related downloads	History
LPCXpresso LPC54102 board		LPCXpresso v7.9.0	Windows help file (chm)	Redlink / CMSIS-DAP	Link2 Windows driver	History
	★ v3.01.000 Release Date 08/04/2015	IAR EWARM 7.40.3	<ul> <li>PDF Document</li> <li>API migration (ROM to LPCOpen)</li> </ul>	CMSIS-DAP		
		Keil MDK- ARM v5.15				

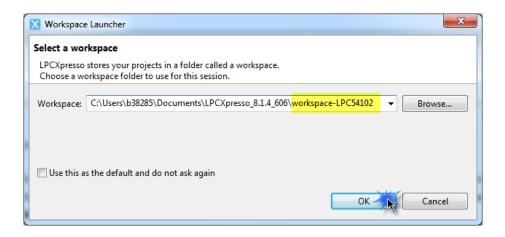
 The LPCOpen packages can also be found on the LPCXpresso installation folder, just make sure they are the latest package available:

C:\nxp\LPCXpresso\_8.1.4\_606\lpcxpresso\Examples\LPCOpen

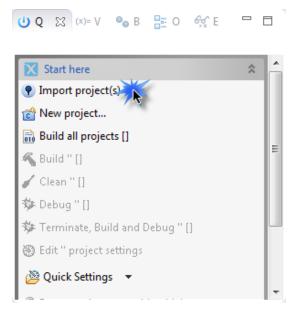
Organize 🔻 🛛 Include	in library ▼ Share with ▼ Burn New folder			i≡ <b>- □</b>
🛠 Favorites	Name	Date modified	Туре	Size
E Desktop	Ipc5410x_lpcxpresso_54102_lpcxpresso_3.01a.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,119 KB
L Downloads	Ipc5411x xpresso54114 Ipcxpresso v3.00b.001 41.zip	3/14/2016 7:01 AM	ALZip ZIP File	1.632 KB
😴 Dropbox	Ipcopen_2_05_lpcxpresso_nxp_lpcxpresso_1343.zip	3/14/2016 7:01 AM	ALZip ZIP File	480 KB
Recent Places	Ipcopen 2 05 Ipcxpresso nxp Ipcxpresso 1347.zip	3/14/2016 7:01 AM	, ALZip ZIP File	678 KB
	Ipcopen_2_06_lpcxpresso_manley_11u68.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,239 KB
📄 Libraries	Ipcopen_2_06_lpcxpresso_nxp_lpcxpresso_11u68.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,084 KB
Documents	Ipcopen_2_10_lpcxpresso_ea_devkit_1788.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,176 KB
🁌 Music	Ipcopen_2_10_lpcxpresso_ea_devkit_4088.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,221 KB
Pictures	Ipcopen_2_10_lpcxpresso_nxp_lpcxpresso_1769.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,021 KB
📄 Subversion	Ipcopen_2_12_lpcxpresso_hitex_eva_1850.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,003 KB
🛃 Videos	Ipcopen_2_12_lpcxpresso_hitex_eva_4350.zip	3/14/2016 7:01 AM	ALZip ZIP File	7,171 KB
	👜 lpcopen_2_12_lpcxpresso_keil_mcb_1857.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,146 KB
💺 Computer	👜 lpcopen_2_12_lpcxpresso_keil_mcb_4357.zip	3/14/2016 7:01 AM	ALZip ZIP File	7,318 KB
師 Primary (C:)	Ipcopen_2_12_lpcxpresso_ngx_xplorer_1830.zip	3/14/2016 7:01 AM	ALZip ZIP File	3,894 KB
	Ipcopen_2_12_lpcxpresso_ngx_xplorer_4330.zip	3/14/2016 7:01 AM	ALZip ZIP File	7,052 KB
📭 Network	Ipcopen_2_12_lpcxpresso_nxp_lpclink2_4370.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,669 KB
	👜 lpcopen_2_16_lpcxpresso_nxp_lpcxpresso_4337.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,899 KB
	👜 lpcopen_2_17_lpcxpresso_nxp_lpcxpresso_1125.zip	3/14/2016 7:01 AM	ALZip ZIP File	441 KB
	👜 lpcopen_2_18_lpcxpresso_nxp_lpcxpresso_1837.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,435 KB
	👜 lpcopen_2_19_lpcxpresso_lpc812max.zip	3/14/2016 7:01 AM	ALZip ZIP File	753 KB
	👜 lpcopen_2_19_lpcxpresso_nxp_lpcxpresso_812.zip	3/14/2016 7:01 AM	ALZip ZIP File	754 KB
	👜 lpcopen_2_19_lpcxpresso_nxp_lpcxpresso_824.zip	3/14/2016 7:01 AM	ALZip ZIP File	859 KB
	👜 lpcopen_2_20_lpcxpresso_nxp_lpcxpresso_1549.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,127 KB
	Ipcopen_v2_00a_lpcxpresso_nxp_lpcxpresso_11c24.zip	3/14/2016 7:01 AM	ALZip ZIP File	401 KB
	Ipcopen_v2_00a_lpcxpresso_nxp_lpcxpresso_11u14.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,086 KB
	Ipcopen_v2_03_lpcxpresso_nxp_lpcxpresso_11u37h.zip	3/14/2016 7:01 AM	ALZip ZIP File	678 KB
	📄 readme.txt	3/14/2016 7:01 AM	Text Document	1 KB

## 2.2 Importing the LPCOpen examples

- Open LPCXpresso, create a new workspace and click on OK:



- On the **Quick Start panel** click on **Import project(s)**:



The Import project(s) wizard will open. There are two ways to import projects, import projects contained in archives or unpacked projects, for this example chose the Project archive (zip) option and click on Browse...:

Import project(s)	_ <b>0</b> X
Import project(s) Select the examples archive file to import.	
Projects are contained within archives (.zip) or are unpacked within a directory. Select yo project archive or root directory and press <next>. On the next page, select those project wish to import, and press <finish>.</finish></next>	
Project archives for LPCOpen and 'legacy' examples are provided.	
Project archive (zip)	
Archive	Brows
Project directory (unpacked)	
Root directory	Browse
LPCOpen	
LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrolle	rs.
LPCXpresso includes the LPCOpen packages which can be imported directly by pressir button in the Project archive (zip) section, above, and navigating to the Examples/LPC	
Alternatively, press the button below to Browse the LPCWare.com website for latest res	ources.
Browse LPCOpen resources on LPCWare.com	
(?)     < <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

- Search for the downloaded **\*.zip** file or locate the package included on the LPCXpresso installation, select it and click on **Open**:

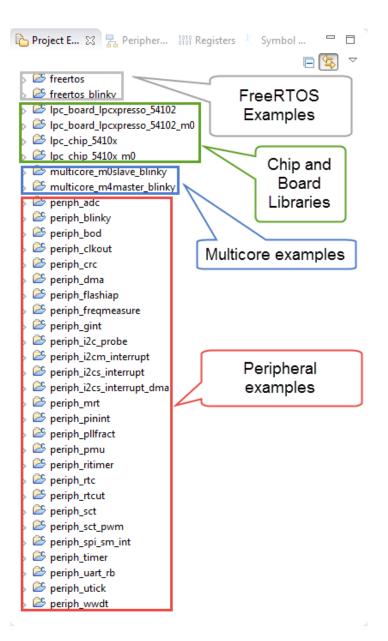
Select the archive containing the projects to import						
G → Kamples → LPCXpresso_8.1.4_606 → Ipcxpresso → Examples → LPCOpen → → ↔ Search LPCOpen						
Organize 👻 New f	older				· 🔳 🔞	
🔆 Favorites	<b>^</b>	Name	Date modified	Туре	Size	
🧮 Desktop		💷 lpc5410x_lpcxpresso_54102_lpcxpresso_3.01a.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,119	
鷆 Downloads		Ipc5411x_xpresso54114_lpcxpresso_v3.00b.001_41.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,632	
💱 Dropbox	=	👜 lpcopen_2_05_lpcxpresso_nxp_lpcxpresso_1343.zip	3/14/2016 7:01 AM	ALZip ZIP File	480	
🖳 Recent Places		👜 lpcopen_2_05_lpcxpresso_nxp_lpcxpresso_1347.zip	3/14/2016 7:01 AM	ALZip ZIP File	678	
		👜 lpcopen_2_06_lpcxpresso_manley_11u68.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,239	
🥽 Libraries		👜 lpcopen_2_06_lpcxpresso_nxp_lpcxpresso_11u68.zip	3/14/2016 7:01 AM	ALZip ZIP File	1,084	
Documents		👜 lpcopen_2_10_lpcxpresso_ea_devkit_1788.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,176	
🚽 Music		👜 lpcopen_2_10_lpcxpresso_ea_devkit_4088.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,221	
Pictures		👜 lpcopen_2_10_lpcxpresso_nxp_lpcxpresso_1769.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,021	
🗟 Subversion		👜 lpcopen_2_12_lpcxpresso_hitex_eva_1850.zip	3/14/2016 7:01 AM	ALZip ZIP File	4,003	
📕 Videos	Ψ.	( III			•	
Fi	le nar	me: lpc5410x_lpcxpresso_54102_lpcxpresso_3.01a.zip	-	*.zip	-	
			ĺ	Open	Cancel	

- Click on **Next** and then **Finish**:

Import project(s)	
Import project(s) Select the examples archive file to import.	
Projects are contained within archives (.zip) or are unpacked within a directory. Select you project archive or root directory and press <next>. On the next page, select those projects wish to import, and press <finish>.</finish></next>	
Project archives for LPCOpen and 'legacy' examples are provided.	
Project archive (zip)	
Archive C:\nxp\LPCXpresso_8.1.4_606\lpcxpresso\Examples\LPCOpen\lpc5410x_lpcxpr	ess Browse
Project directory (unpacked)	
Root directory	Browse
LPCOpen LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers	i.
LPCXpresso includes the LPCOpen packages which can be imported directly by pressing button in the Project archive (zip) section, above, and navigating to the Examples/LPCO	pen directory.
Alternatively, press the button below to Browse the LPCWare.com website for latest reso	ources.
Browse LPCOpen resources on LPCWare.com	
? < Back Next > Einish	Cancel

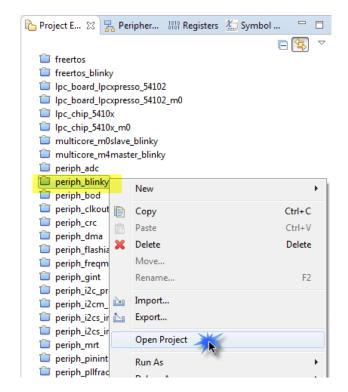
Import project(s)	- • ×
Import project(s) Select a directory to search for existing Eclipse projects.	
Projects:	
Image: Construct of the sector of the sec	Select All
<u> </u>	
Working sets          Image: Working sets         Working sets:	S <u>e</u> lect
? < <u>Back</u> Next > Einish	Cancel

The example projects along with the chip and board libraries are now shown on the **Project Explorer** window:

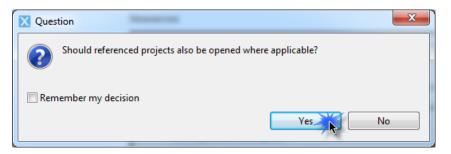


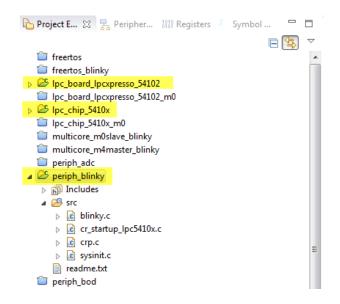
## 2.3 Building and debugging blinky project

After importing the projects to the workspace these can be closed to avoid having all of them opened in the workspace. Then we can open only one of them, for this example we will use the periph\_blinky project. Right click on the project and click on Open:



A pop-up window will appear asking if the referenced projects should also be opened, click on
 Yes, you should see the periph\_blinky project opened along with the M4 chip and board libraries:

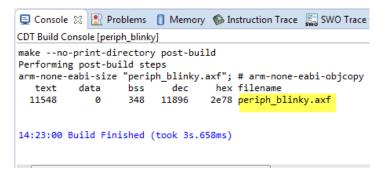




 The next step is to build the project, select the project, go to the Quick start panel and click on Build 'periph\_blinky' [Debug]:

<ul> <li>multicore_m4master_blinky</li> <li>periph_adc</li> <li>periph_blinky</li> <li>pin Includes</li> <li>pin Includes</li> <li>pin Includes</li> <li>pin Includes</li> <li>pin cr_startup_lpc5410x.c</li> <li>pin crp.c</li> <li>pin sysinit.c</li> <li>readme.txt</li> <li>periph_bod</li> <li>periph_clkout</li> <li>periph_crc</li> <li>periph_dma</li> <li>periph_flashiap</li> <li>periph_freqmeasure</li> </ul>	
Deperiph_gint	
U Quic 🔀 🗱 Varia 💁 Brea 📴 Outli 💇 Expr	
X Start here	*
• Import project(s)	
📸 New project	
🗟 Build all projects [Debug]	
Suild 'periph_blinky' [Debug]	
🧹 Clean 'periph_blinky' [Debug]	
🕸 Debug 'periph_blinky' [Debug]	
🏇 Terminate, Build and Debug 'periph_blinky' [Debug]	

 The corresponding chip and board libraries will be built and the file periph\_blinky.afx generated:



- Now select the project and go to menu **Run > Debug Configurations**:

Se <u>a</u> rch <u>P</u> roject	Run	<u>W</u> indow <u>H</u> elp		
N - 🗟 🎋		Resume		4
		Suspend		ł
Peripher 1919 Ro		Terminate		L
	19	Disconnect		ŀ
	Ъ.	Step Into		I
p_lpc5410x.c	P	Step Over		I
p_ipes izexie	.Ē	Step Return		I
	=>[	Run to Line		I
	ক	Use Step Filters	Shift+F5	I
	Q	Run Last Launched	Ctrl+F11	l
	첺	Debug Last Launched	F11	I
		Run History	•	l
		Run As	+	I
sure		Run Configurations		
e		Debug History	•	
errupt		Debug As	+	I
rrupt		Debug Configurations		
rrupt_dma			715	l

 The Debug Configurations window will open, double click on the "C/C++ (NXP Semiconductors) MCU Applications" option to create a new connection and program the MCU trough the Onboard Link2 debug probe and click on Debug:

X Debug Configurations		X
Create, manage, and run configurations Create, manage, and run configurations  Vill be created  type filter text	Name: periph_blinky Debug Main Common Source C/C++ Application: Debug\periph_blinky.axf Project: periph_blinky Build (if required) before launching Build configuration: Select Autom Enable auto build	e Debugger Variables Search Project Browse Browse
Filter matched 8 of 8 items	● Use workspace settings	Configure Workspace Settings o a terminal. Revert Apply Debug Close

A new window will show up and it will display the connected emulators, select the LPC-LINK2
 CMSIS-DAP emulator and click on OK:

Connect to emulator: LinkSer 1 emulator found. Select the emulator to use	ver		
Name	Serial number/	Туре	Manufacturer
LPC-LINK2 CMSIS-DAP V5.134	A00000002	LinkServer	NXP Semiconductors
Emulator search options Search for LinkServer again	Search for any er	nabled emulat	tor Cancel

 After this a new message will show up, indicating that 2 SWD devices were found, select the Cortex-M4 device which is the one used by the example:

X	]		A Real Property lines	The local sectors	×		
s	SWD Configuration						
		ole SWD Devices de Cortex-M4' has bee					
	Device	Name	TAP Id	Details			
	<b>V</b> 0	Cortex-M4	0x2ba01477	APID:24770011			
	1	Cortex-M0	0x2ba01477	APID:24770011			
	OK Cancel						

- You should now be able to **debug** the program and **step** through the code:

X Develop - Ipc_chip_5410x/src/sysinit_5	Dx.c - LPCXpresso	
File       Edit       Source       Refactor       Navigate       Search       Project       Run       Window       Help         Image: I		
	· U · • M · • M · • · · · · · · · · · · · ·	
🔁 P 🔀 🛃 P 🕮 R 🕹 S 💻 🗖	to Debug 🛛	🍇 🕅 🖬 🔻 🗖 🗖
🗏 🔄 🗸	Fer periph_blinky Debug [C/C++ (NXP Semiconductors) MCU Application]     Are periph_blinky.axf [LPC54102J512 (cortex-m4)]	Â
▷ iap.c ▲	Thread #1 <main> (Suspended : Signal : SIGSTOP:Stopped (signal))</main>	=
pinint_5410x.c	0x25febffe Chip Clock SetMainClockSource() at clock 5410x.c:291 0xbd2	
⊳ 💼 pll_5410x.c ⊳ 🗊 ring_buffer.c	Chip_Clock_seturainClockSource() at Clock_3410x.c:2910x862 Chip_SetupIrcClocking() at sysinit_5410x.c:106 0x1c50	
▷ ining_burier.c ▷ ic ritimer_5410x.c ≡	Board_SetupClocking() at board_sysinit.c:199 0x844	+
⊳ 💼 rtc_ut.c ⊳ 🗊 sct_5410x.c	🏽 Welcome 💽 0x25febffe 💽 clock_5410x.c 🔀 sysinit_5410x.c 🛛 🔭 🗖 🗖	🔤 Disassembly 🔀 📃 🗖
sct_pwm_5410x.c	<pre>95 pllError = Chip_Clock_SetupPLLData(&amp;pllConfig, &amp;pllSetup); 96 if (pllError == PLL ERROR SUCCESS) {</pre>	Enter location here 👻 🔹 🟠
b spi_common_5410x.c	97 pllSetup.flags = PLL_SETUPFLAG_WAITLOCK   PLL_SETUPFLAG_ADGV(	$\bigtriangledown$
▷ c spim_5410x.c ▷ c spis_5410x.c	<pre>98 pllError = Chip_Clock_SetupSystemPLLPrec(&amp;pllSetup); 99 }</pre>	00001c22: ; <undefin 00001c26: : <undefin< td=""></undefin<></undefin 
▷ stopwatch_5410x.c	100 101 /* Set system clock divider to 1 */	96 if (pllError ==
syscon_5410x.c	<pre>102 Chip_Clock_SetSysClockDiv(1);</pre>	00001c28: ; <undefin 00001c2c: ; <undefin< td=""></undefin<></undefin 
4 III >>	103 104⊖ /* Set main clock source to the system PLL. This will drive 24MH;	97 pllSetup.fla 00001c30: ; <undefin< td=""></undefin<>
(U) Q 🖾 (x)= V 💁 B 🎇 🗖 🗖	105 for the main clock and 24MHz for the system clock */	98 pllError = C
	107 108⊖ /* ASYSNC SYSCON needs to be on or all serial peripheral won't w	00001c38: ; <undefin< th=""></undefin<>
🔀 Start here 🕆 📤	109 Be careful if PLL is used or not, ASYNC_SYSCON source needs to	00001c3c: ; <undefin 00001c40: ; <undefin< td=""></undefin<></undefin 
Import project(s)	110 selected carefully. */ 111 Chip SYSCON Enable ASYNC Syscon(true);	102 Chip_Clock_SetSy 00001c44: ; <undefin< td=""></undefin<>
📸 New project	112 Chip_Clock_SetAsyncSysconClockDiv(1); 113 Chip_Clock_SetAsyncSysconClockSource( <i>SYSCON_ASYNC_IRC</i> );	00001c48: ; <undefin< td=""></undefin<>
🗟 Build all projects [Debug]	113 Chip_clock_setAsyncsysconclocksource(srscow_Async_inc); 114 }	106 Chip_Clock_SetMa
K Build 'lpc_chip_5410x' [Debug]	< Þ	<
🧹 Clean 'lpc_chip_5410x' [Debug]	🚍 Console 🕱 🖹 Problems 🔋 Memory 🏇 Instruction Trace 🎆 SWO Trace Config 📼 Power M	easurement Tool 🛷 Search 🛛 🗖 🗖
🕸 Debug 'Ipc_chip_5410x' [Debug]		💥 🖹 🔛 🖓 🦃 😾 🖬 🕶 🗂 🕶
🅸 Terminate, Build and Debug 'Ipc	eriph_blinky Debug [C/C++ (NXP Semiconductors) MCU Application] periph_blinky.axf	*
Edit 'lpc_chip_5410x' project setti	4	7 ( 4
/lpc_chip_5410x/src/sysinit_5410x.c Wri	ole Smart Insert 106 : 1	UNXP LPC54102J512 (lpc_chip_5410x)
,		· INT CONTRACTOR

# **Appendix A - References**

- LPCOpen webpage: http://www.nxp.com/lpcopen
- LPCXpresso webpage: <u>http://www.nxp.com/lpcxpresso</u>
- LPCXpresso Board for the LPC54100 family of MCUs: <u>http://www.nxp.com/products/software-and-tools/hardware-development-tools/lpcxpressoboards/lpcxpresso-board-for-the-lpc54100-family-of-mcus:OM13077</u>