

LPC PRESSO Cetting started with NXP LPCXpresso Rev. 10 - 7 April 2011

User guide

Document information

Info	Content
Keywords	LPCXpresso, LPC1100, LPC1200, LPC1300, LPC1700, LPC2000, LPC3000
Abstract	LPCXpresso is a new, low-cost development platform available from NXP. This document is a brief overview on how to get started with LPCXpresso.



NXP Semiconductors

LPCXpresso

Getting started with NXP LPCXpresso

Revision	history
Rev	Date

Description

1101	Date	Description
10	20110407	Added Fig 43 and Fig 44.
9	20110301	Updated keywords and supported products throughout
		Updated Section 6.1
		Updated Section 6.2.6
		Updated Section 6.6.2
		Removed Fig 20
		Added Fig 40 and Fig 41
8	20110211	Updated Section 3.2
		• Added Fig 37, Fig 38, Fig 39, Fig 42
7	20100915	Updated Section 3.1
6	20100712	 Added 7.1 Schematics for LPCXpresso LPC1768 target side
		Added 6.1 Installing Eclipse plugins
5	20100604	 Added new products supported to Introduction section: LPC2929, LPC3250
		Removed 6.1.6 Download performance
		Updated 6.4.2 Optimization section
4	20100419	Updated Fig 45
3	20100315	Updated Fig 36
		Added Fig 37
2	20100311	Updated Section 3.1
		Added Section 6.2.6
1	20100111	Initial version

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

LPCXpresso

1. Introduction

LPCXpresso is a new, low-cost development platform available from NXP. The software consists of an enhanced, Eclipse-based IDE, a GNU C compiler, linker, libraries, and an enhanced GDB debugger. The hardware consists of the LPCXpresso development board which has an LPC-Link debug interface and an NXP LPC ARM-based microcontroller target. LPCXpresso is an end-to-end solution enabling embedded engineers to develop their applications from initial evaluation to final production.

The LPCXpresso IDE, powered by Code Red Technologies (<u>www.code-red-tech.com/lpcxpresso/</u>), is based on the popular Eclipse development platform and includes several LPC-specific enhancements. It is an industry-standard GNU toolchain with an optimized C library that gives engineers all the tools necessary to develop high-quality software solutions quickly and cost-effectively. The C programming environment includes professional-level features. There is syntax coloring, source formatting, function folding, on- and offline help, and extensive project management automation.

The LPCXpresso target board, jointly developed by NXP, Code Red Technologies, and Embedded Artists (<u>http://www.embeddedartists.com/products/lpcxpresso/</u>), includes an integrated JTAG debugger (LPC-Link), so there's no need for a separate JTAG debug probe. The target portion of the board can connect to expansion boards to provide a greater variety of interfaces, and I/O devices. The on-board LPC-Link debugger provides a high-speed USB to JTAG/SWD interface to the IDE and it can be connected to other debug targets such as a customer prototype. Users can also use the LPCXpresso IDE with the Red Probe JTAG adapter from Code Red Technologies.

Supported LPC products on the LPCXpresso platform:

- LPC1100: All products supported
- LPC1200: All products supported
- LPC1300: All products supported
- LPC1700: LPC1751 LPC1752 LPC1754 LPC1756¹ LPC1758¹ LPC1764 LPC1765¹ LPC1766¹ LPC1767¹ LPC1768¹ LPC1769¹
- LPC2000: LPC2109 LPC2134 LPC2142 LPC2362 LPC2929
- LPC3000: LPC3130 LPC3250



 The LPCXpresso platform will allow the user to build an executable of any size but it will restrict code download to 128 kB. These specific products contain more than 128 kB of flash memory. Easy upgrade options are provided for higher memory configurations. Please visit the LPCXpresso website for more details.

1.1 LPCXpresso IDE

LPCXpresso's IDE is a highly integrated software development environment for NXP's LPC Microcontrollers, which includes all the tools necessary to develop high quality software solutions in a timely and cost effective fashion. LPCXpresso is based on Eclipse with many LPC specific enhancements. It also features the latest version of the industry standard GNU tool chain with a proprietary optimized C library providing professional quality tools at low cost. The LPCXpresso IDE can build an executable of any size with full code optimization and it supports a download limit of 128 kB after registration. LPCXpresso supports the full embedded product design cycle by moving beyond chip evaluation boards and supporting development on external target boards.

	🐔 🖌 🏇 🖄 - İ 🏊 🖆 🛅 📮 🐌 - İ 🕜 🔉 İ 🇞 🗐 🛣 🗮 🔀 Develop
🏠 Project Expl 🙁 👬 Core Regist 🚼 Peripherals 🙄 🗖	C man_ledflash.c 🛛 C easyweb.c 🕒 Welcome
E 😫 🗸	43 int main (void) (
B 😂 RDB1768_adcpot	44 45 volatile static int i = 0 :
CB1768_AudioMeter	46 int loop;
RDB1768_EasyWeb	47
BUC RDB1768_EXTINC	48 // Set up board/processor
B BB1768 LCDb	49 TargetResetInit();
B BDB1768 LedFlash	50
🗄 🐇 Binaries	51 // Set P1_24, P1_25, P1_28, P1_29 to 00 - GPIO
🗷 🔊 Includes	52 PINSEL3 ε= (~OxOFOFO000);
🖨 😂 src	53 54 // Set GDTO - D1 24 D1 25 D1 29 D1 29 - to be outputz
🖶 🔟 cr_startup_lpc17.c	55 FIGIDIR = LED ALL:
H C leds.c	56
tal leds.h	57
B target c	58 // Enter an infinite loop, cycling through led flash sequence
	59 while(1) (
🙂 Quickstart Pan 🙁 🖉 Variables 💁 Breakpoints 🗖 🗖	60
	61 // Turn all leds on, then off, 5 times
🖉 Start here 🔹 👘	62 IOT (100p=1; 100p < 5; 100p++) (
Import Example project(s)	64 short delay(2000):
& Build 'BDB1768 LedElash' for Debug	65 leds all off();
Glaid Roos too_cod lash for beoldg	66 short delay(2000);
🧹 Clean 'RDB1768_LedFlash' for Debug	67)
🏂 Debug 'RDB1768_LedFlash'	68 short_delay(3000);
🖄 Quick Settings 🔹	K
	📮 Console 🕴 📲 Problems 🚦 Memory 📑 👘 🗝
📡 Project and File wizards 🛛 😵	No consoles to display at this time.
📡 Import and Export 🛛 😵	
📓 Build and Settings 🛛 😵	
E Debug and Run	
∏♦ Writable S	imart Insert 49 : 1 RDB1768 LedFlash NXP/LPC1768

Getting started with NXP LPCXpresso

1.2 LPCXpresso development board



1.3 LPC-LINK JTAG/SWD debugger

The LPCXpresso board contains a JTAG/SWD debugger called the "LPC-Link" and a target MCU. LPC-Link is equipped with a 10-pin JTAG header (highlighted on the above image) and it seamlessly connects to the target via USB (the USB interface and other debug features are provided by NXP's ARM9 based LPC3154 MCU). Cutting the tracks between the LPC-link and the target will make the LPC-Link a stand-alone JTAG debugger. This enables the LPCXpresso platform to be connected to an external target and used to develop for a wide variety of NXP's Cortex-M0, Cortex-M3, and ARM7/9 based applications. Currently supported microcontroller products include LPC1700, LPC1300, LPC1200, and LPC1100 series and specific members of the LPC2000 and LPC3000 families.

1.4 Integrated evaluation target

The target includes a small prototyping area and easily accessible connections for expansion. The LPCXpresso board with target can be used

- On its own for software development and benchmarking
- · Connected to an off-the-shelf baseboard for rapid proof-of-concepts
- Connected to customer-designed board for a full prototype

1.5 LPCXpresso partners

NXP has partnered with Code Red Technologies and Embedded Artists for the LPCXpresso platform. For added flexibility and higher memory configurations, the

LPCXpresso platform can easily be upgraded to include full-blown suites from Code Red Technologies and more advanced hardware kits from Embedded Artists. Please visit the LPCXpresso webpage for more information.





Getting started with NXP LPCXpresso

2. Evaluate, explore and develop



Users can envisage three stages from evaluation to product development. During evaluation, features and peripherals of the target MCU can be easily tested with the prototyping area and easily accessible connections on the target board. Complementing the target board are also easy-to-use example projects and a handy Getting Started guide. For rapid proof-of-concepts, users can get an off-the-shelf base board from Embedded Artists and quickly explore the next level of applications. And finally LPCXpresso users can seamlessly develop their final application by using the LPC-Link's 10-pin JTAG connector to attach any JTAG-capable custom board. This way, users can now enjoy the same user experience right from evaluation to product development.

3. Installation

3.1 System requirements

Operating System	Microsoft® Windows - XP 32-bit or 64-bit (SP2 or greater) Microsoft® Windows - Vista 32-bit or 64-bit Microsoft® Windows - Windows 7 32-bit or 64-bit Linux - Ubuntu 9 and 10 Linux - Fedora Core 12 and 13
System RAM	512 MB minimum (1 GB recommended)
Hard Disk	225 MB of available space.
Screen/Display Adaptor	1024x768 minimum recommended
Internet Connection	High-speed internet is recommended to download and register the software

Note: Desktop virtualization tools supporting a linux or Windows guest with USB support can be used to run LPCXpresso on other computing platforms.

3.2 Installation process

LPCXpresso is installed into a single directory, of your choice. Multiple versions can be installed simultaneously without any issues. The installation process is to double-click the installer file after downloading. Then click "next" on the setup wizard. To install under linux, the downloaded file should be marked as executable first using chmod +r.

Setup - LPCXpresso	
NP	Welcome to the LPCXpresso Setup Wizard
	This will install LPCXpresso 3.0.3 [Build 59] on your computer.
LPC PRESSO	It is recommended that you close all other applications before continuing.
	Click Next to continue, or Cancel to exit Setup.
Powered by	
acode_red	
http://www.pxp.com/locxpresso	
	Next > Cancel
Fig 4. Setup wizard	

LPCXpresso



Read the license agreement then click next. There are a number of other screens on the setup wizard, but generally the default options can be accepted. After the install, an information file will be displayed. Click "Next." Congratulations! Your LPCXpresso installation is complete!

	Installing Please wait while Setup installs LPCXpresso on your computer.
	Extracting files C:\rwp\lpcxpresso\eclipse\plugins\org.eclipse.cdt.core_5.1.1.200909110608.jar
	LPCXpresso v3.0_beta3
Fig 6. Setup wiza	ırd

LPCXpresso

3.3 Activation

To activate your product from LPCXpresso, choose Help->Product activation->Create serial number and register. Once the wizard is open, click "Copy to clipboard" to copy the LPCXpresso serial number into the clipboard. This serial number is based on your machine's hardware and operating system configuration, but contains no personally identifiable information. Now click the button to open the registration activation page. This should display a web form. After completing the form, you will receive an activation code via email within a few minutes. Highlight the activation code in your email program, and select Copy to place it into the Windows clipboard. Now, choose select Help->Product activation->Enter activation code from within LPCXpresso. Paste the product activation code into the Product activation dialog by right clicking in the Activation code field and choosing "Paste." Then click the "OK" button. You should receive a dialog confirming acceptance of the activation code. It is also possible to complete LPCXpresso activation on a PC that is offline as long as another PC has access to the Internet. Refer to Fig 7 for the process.



4. Getting familiar with the LPCXpresso IDE

LPCXpresso IDE is based on the Eclipse IDE framework and many of the core features are described well in generic Eclipse documentation and in the help files found in the help menu of the product. Further documentation and pointers to useful documents are also available on the Code Red Technologies Wiki at <u>http://lpcxpresso.code-red-tech.com/LPCXpresso/softwareknowledgebase</u>.

4.1 Layout of the LPCXpresso desktop

LPCXpresso's Desktop contains many windows. Each window is called a View, because it displays a particular view of data in the LPCXpresso environment. This data could be source code, hex dumps, disassembly, memory contents, or more. Views can be opened, moved, docked, and closed, and the layout of the currently displayed Views can be saved and restored. A specific configuration of Views is called a 'Perspective.' Typically, LPCXpresso operates in a single perspective under which both the code development & debug sessions operate as shown on the next page. The single perspective greatly simplifies the Eclipse environment and enhances the entire LPCXpresso experience.

All Views in the Perspective can be moved around by dragging and dropping. If a View is accidentally closed, it can be restored by selecting it from the Show View dialog. The Show View dialog can be opened from the Show View Other... option in the Window menu.



Getting started with NXP LPCXpresso



E Develop - NXP_LPC13xx_Blinky/src/blinky.c - LPCXpresso	
File Edit Source Refactor Navigate Search Project Run Window Help	
i ➡ • 🖩 🖮 i @ • 않 않 ♂ 👔 i 🔮 ≪ 🧳 🎄 💩 • i ù i 🍫 i ≪ • ⊛ • i ⊗ 🔳 i ❷ 🖋 • i 🅖 i ⇔ • ↔ •	u 🖆 📇 📮 🥼 🧯 🕢 💭 🔅 🎯 😾 🖽 🔀 🗶 Develop
Proje 🕱 📲 Core 🔒 Perip 🖓 🗖 🕥 Welcome 🚺 blinky.c 🗙	- B
<pre> I/*********************************</pre>	x C file for NXP LPC13xx Family Microprocessors XF Semiconductor Prelimnary version, first Release LPC13xx definitions */
15 (Daut 92 Maying P. R. B. 16#define TEST TIMER NIM	0 /* 0 or 1 for 16-bit timers only */
O Quick 22 W- Vana G Brea 11 10 #define 1251_111Ek_Non	5 , 5 St 1 IST 10 Sic Simers Shiry ,
Starthere	timer16_0_counter;
Import Example project(s) 19 extern volatile uint32_t	timer16_1_counter;
K Build 'NXP_LPC13xx_Blinky' for Debui	
Clean 'NXP_LPC13xx_Blinky' for Debu	
No Debug NYP LPC 13vy Blinky	
) 🔒 📑 🖬 🖬 - 😁 - 🗖
C-Build [NXP_LPC13xx_Blinky]	
Project and Project and Second Structure NXF_LFC13xx_Blinky.axf >NXF_L text data bss of 3048 13 120 33	PC13xx_Blinky. ec hex fi 81 c6d NX
Writable Smart Insert 1:1	NXP_LPC13xx_BlinkyNXP/LPC1343

Fig 9. Single perspective (develop)

- 1. Project Explorer View: The 'Project Explorer' gives you a view of all the projects in your current 'Workspace'. A 'Workspace' is a collection of projects that are stored in a single Workspace Directory on your computer.
- 2. Editor: On the upper right is the editor which allows modification and saving of source code as well as setting breakpoints in debug mode.
- 3. Console and Problems Views: On the lower right are the Console and Problems Views. The Console View displays status information on compiling and debugging, as well as program output. The Problem View (available by changing tabs) shows all compiler errors and will navigate the Editor View to the error location.
- 4. Quick Start View: Below, the 'Quick Start' view has fast links to commonly used features. This is the best place to go to find options such as Build, Debug, and Import.

LPCXpresso Getting started with NXP LPCXpresso

4.1.2 Single perspective (debug)

Develop - N	XP_LPC13xx_Blinky/sro	/blinky.c - LPCXpresso	
e Edit Source	e Refactor Navigate Sea	ch Project Run Window Help	
∎ • 🖫 🖻 🕸 🗄 🔨 •	⊡ ∂ • 2 29 ⊗ • ⊘ 1 2 ,	C C C C Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q	
Proje	re 🛛 🛃 Perip 🗖 🗖	🗱 Debug 🛿 🦉 🦓 👘 👛 🖉 🕼 🗟 👘 🕱 🏟 🗸	
Register	Core Registers 🛛 Iı 🛧	E C NXP_LPC 13xx_Blinky (Debug) [C/C++ MCU Application]	^
1919 r0	0x10000088	ACU GDB Debugger (11/6/09 1:55 PM) (Suspended)	
0101 r 1	0x10000088	□ □ m [®] Thread [0] (Suspended)	
0101 r2	0x0		2
888 r3	0x0	Welcome R blinky.c 🕅	
0101 r4	0x40048000	20	
1111 r5	0xefffd432	21/* Main Program */	
1919 r6	0x'	22	
1010 r7	0	23int main (void) {	
1111 r8	0.		
8181 r9	Oxt	25 SystemInit():	
0101 r 10	0x0	26	
0101 r11	0x0 🗸	27 /* Config CLKOUT, mostly used for debugging. */	
(>	28 CLKOUT Setup (CLKOUTCLK SRC MAIN CLK);	-
U Quick 🕅	(X)= Varia 🔍 🗣 Brea 🗋 🗖 🗖	29 LPC_ICCON->PI00_1 &= ~0x07; 30 LPC_ICCON->PI00_1 = 0x01; /* CLK_OUT */	
	^	31	
👫 Start hei	re 🏾 🌋	32 /* Enable AHB clock to the GPIO domain. */	
📍 Import Exa	mple project(s)	33 LPC SYSCON->SYSAHBCLKCTRL = (1<<6);	
Ruid NVP I	PC13vv, Blinky' for Debu	34	
Gound HAP_L	- CIDXX_DIIIKY IOI DEDU	35 /* TEST_TIMER_NUM is either 0 or 1 for 16-bit timer 0 or 1. */	
🧹 Clean 'NXP_	LPC13xx_Blinky' for Debu	36 init timer16(TEST TIMER NUM. TIME INTERVAL);	~
🕉 Debua 'NXP	LPC13xx Blinky'	2	J
		📮 Console 🕴 🚼 Problems) 🚺 Memory 📃 📮 🖉 📑 🖝 🖻 🔻 📑 🕈	
🖄 Quick Setti	ngs 🔻	NVP IPC13vx Block (Datus) [C/(++ MCILAnglication] C/Dog ments and Sattings)136(My Dog ments larger bats 3/unressed	
		The is a serve_server, the construction of the potential and acting.	
🔛 Project a	ind Fi 📃 🔇		
1177		4)	
Import a	nd Ex, 📎		~
107 IN 191			>
=0		: : : NXP LPC13vx Blinkv NXP/LPC134	3

Fig 10. Single perspective (debug)

- 1. Core Register View: This shows all of the registers in the processor core. Registers that have changed from step to step are highlighted in yellow.
- 2. Debug View: This shows you the stack trace and the debug toolbar. Using the icons at the top of the view, you can step through code or execute at full speed. In the 'stopped' state, you can click on any particular function and inspect its local variables in the right hand panel on the Variables tab.
- 3. Editor: In here you will see the code you are executing and can step from line to line. By pressing the 'i' icon at the top of the Debug view, you can switch to stepping by assembly instruction. Clicking in the left margin will set and delete breakpoints.
- 4. Console View: On the lower right is the Console View. The Console View displays status information on compiling and debugging, as well as program output.
- 5. Quick Start View: Below, the 'Quick Start' view has fast links to commonly used features. This is the best place to go to find options such as Build, Debug, and Import.

4.1.2.1 Peripheral views

LPCXpresso includes full, annotated and interactive debug views of all the peripherals. Access to the views is found on the Peripherals View (click the Peripherals tab) behind the Core Registers view. Each peripheral can be selected, and it is displayed in the 'Memory' view which is located behind the 'Console' view at the bottom of the LPCXpresso desktop.

LP





4.2 Connecting the target

To begin development, the LPCXpresso can be connected to a PC using a USB 2.0 A/Mini-B cable.



If you are debugging a prototype board or a target containing a different MCU, see the Appendix for a pinout to connect the debugger section of the LPCXpresso to an external target.

LPCXpresso

5. Blinky: Build, download and debug

5.1 Importing the blinky project using the Quickstart panel

Example projects make great starting points for your own embedded projects.

In LPCXpresso, Quickstart Panel brings together all of the most frequently used operations for Embedded Microcontroller development. The Quickstart Panel is by default located in the lower left of LPCXpresso.

• On the Quickstart Panel, click on the 'Start Here' sub-panel, and click on 'Import Example project(s)'.

	U Quickstart Pan 😵 🕺 We Variables) 💁 Breakpo		
	Start here Import Example project(s) Fuild " () Clean " () Suild all projects () Clean " () Clean		
	Project and File wizards		
	Import and Export Build and Settings	*	
Fig 14 Quickstart page	□ [◆] Done		

• Click on 'Browse' and find the LPCXpresso examples directory. If this is the first time you have used the Importer, then it should automatically be at the right directory; otherwise, the default install directory is 'C:\nxp\lpcxpresso\Examples'. Once in this directory, pick an archive for your particular evaluation board and click 'Open' and then 'Next'.

Getting started with NXP LPCXpresso

📫 - 🔛 🖄 🕘 🥒 🖓 🚵	₫• 월 8 8 8 ¹ 8- 9 9 - 9 -	2 🖆 📇 🔾 🔎	0	1 🕸 🔛 🔀 De	velop
Project Expl 🕺 🚻 Core Regist 🛃 Periphe	🖾 Import Example proje	ct(s)			
	Import Example project Select the examples archive fi	(s) le to import.		1	► ●
	Projects are supplied in project each archive. After selecting y exist in the workspace) will be	archive format (.zip). Multiple proj our archive, press Next. A list of p presented. Select those you wish t	ects are contained within ojects (that don't already o import, and press Finish.	RESSO powered b	/ Ecod e
	Project archive		Browse		
	Select the ar	chive containing the projec	ts to import	? 🛛	
	Look i	n: 🔁 Examples	O Ø	F 📂 🛄 •	
🕑 Quickstart Pan 🛛 🖓 🖓 Variables 🔍 Breakpo	My Recent Documents	LPC11xx_CMSIS_1.30.zip LPC13xx_CMSIS_1.30.zip LPC114LPCXpresso_Exan	nples 1.0. zip		
Start here	R		pestorap		
Import Example project(s) Suild "()	Desktop				
🖌 Clean " ()					
Build all projects ()	My Documents				~
Quick Settings •	2				- 📬 • 🗖 🗆
Project and File wizards	My Computer				
Import and Export	*	Flanama: IPC13/3.1	PCYnraeen Evamplae1.0 tin		
E Build and Settings	Ny Network	Files of type: *.zip	a expressed and the transp	Cancel	

Fig 15. Examples directory

- You will then be presented with a list of projects within the archive as shown. Make sure that the 'Blinky' Project and the CMSIS Project are checked.
- Click 'Finish' and the two Projects will be imported into the current workspace by LPCXpresso.
- Now click 'Build all projects (Debug)' on the Quickstart Panel to build the Blinky example and CMSIS library.

5.2 Debugging/running 'blinky' on your LPCXpresso board

In LPCXpresso, when you start to debug, your program will automatically download to the target and be programmed into flash memory.

To start debugging on your target, simply highlight the project in the Project Explorer and then in the Quickstart Panel select 'Debug project 'Blinky'. LPCXpresso may then show a dialog asking you to select which executable (Release or Debug) to use. Select 'Debug,' Then click on 'OK' to continue with the download and debug of the 'Debug' build of your project.

LPCXpresso

17 of 51

LPCXpresso Getting started with NXP LPCXpresso



You may also enter debug mode by clicking the bug icon on the top LPCXpresso toolbar.



You are then presented with the debug view and toolbar and have run control over the code running on your target. The toolbar will pop up above the code window.



You can now do the following with the buttons towards the top of the 'Debug' view:



LPCXpresso

6. LPCXpresso IDE tips and tricks

6.1 Installing Eclipse plugins

The LPCXpresso IDE contains many of the features of the Eclipse open-source IDE from http://www.eclipse.org. The browse and install plugin function is present in the help menu. To access it, choose Help -> Install New Software. This will display the Eclipse Install Software dialog which will allow browsing and installing of Eclipse plugins.

6.2 Debugging tips

6.2.1 Debug features not enabled

All of the LPCXpresso features are context-sensitive. If features are disabled, doublecheck that you are navigated into a .c file in an open project on the Project Explorer View, or some menu items and toolbar buttons may be disabled. If your workspace contains projects that create libraries such as CMSIS, please note that debug features will be disabled if you are currently editing a .c file that is part of a library project.

6.2.2 Incorrect registers displayed or errors starting debug

Make sure that the correct NXP LPC microcontroller part is selected in LPCXpresso. The current part number is displayed in the status bar at the bottom of the LPCXpresso window. It can be changed by holding down the Ctrl key and clicking. A dialog will appear allowing selection of the correct part number.



LPCXpresso

Resource Image: Target: Build Variables Image: Target: Build Variables Image: Target: Descrivery Options Image: Target: MCL settings Image: Target: Tool Chain Editor Image: Target: Project References Image: Target: References Image: Target: Run/Debug Settings Image: Target: Targe	type filter text	MCU settings	(\$ * \$ * *	-
C(-+ € fuld Poid/Svripes Descrivery Obtons Poid/Svripes Descrivery Obtons Poid/Svripes Settings Tool Chain Editor Tool Chain Editor Project References Run/Debug Settings Terget architecture Cortex-m3 Terget architecture Cortex-m3 Terget architecture Cortex-m3 Terget architecture Size Size Terget architecture Size Terget architecture Size Terget architecture Size Terget Size Size Size Terget Size - Resource Builders	Target		-	
Build Variables Build Variables Discovery Qotons Environment MCU settingsi Settingsi Project References LPC1313 LPC13142 LPC1343 References LPC1343 BLC2000 BLC2000 BLC1300 LPC1313 LPC1343 References LPC1343 LPC3000 BLC2000 BLC2000 BLC2000 BLC2000 BLC333 LPC1342 LPC300 BLC2000 BLC20000 BLC20000 BLC20000 BLC200000 BLC200000	C/C++ Build	NYP LPC 1343		1
□ Discovery Options If UPC2000 ■ UPC1000 If UPC1000 MCU settings If UPC1300 □ Settings IPC1313 □ Cold Anis Editor IPC1313 IPC(-4 General IPC1313 IPC1942 IPC1343 IPC1942 IPC1343 IPC1300 IS IPC1342 IPC1343 IPC1343 IPC1300 IS IPC1300 IS IPC1342 IPC1342 IPC1300 IPC1342 IPC1300 IS IPC1300 IS IPC1342 IPC3000 IPC3000 IS IPC3000 IS IPC3000 IS IPC3000 IS IPC3000 IS IPC3000 IPC3000 IPC3000 IS IPC3000 IPC3000 IPC3000 IPC3000 IPC3000 IPC3000 IPC3000 IPC3000 IPC3000 IPC1000 IPC3000 IPC10000000 IPC3000 I	-Build Variables	E NXP		
Environment ⊞ LPC1700 Bridge LPC131 Col Chain Editor LPC1313 Bridge LPC1313 LPC190 LPC1313 Project References LPC1342 LPC190 LPC1343 Refactoring History LPC1343 Run/Debug Settings LPC1300 LPC1900 LPC1342 UPC1342 LPC1343 LPC1900 LPC1900 LPC1900 LPC19000 LPC1900 LPC19000 LPC19000 LPC19000 LPC19000 LPC19000 LPC1900000 LPC190000 LPC1900000000 LPC1900000 LPC19000000000000000000000000000000000000	- Discovery Options	■ LPC2000		
MCU settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical content of the settings Image: Classical cont	Environment	⊥PC1700		
Bit of College LCC1311 LP void Chain Editor LCC1313 LP void Chain Editor LCC1313 LP void Chain Editor LCC1323 LP void Chain Editor LCC13242 LP void Chain Story LCC1343 RunyDebug Settings III LPC 1000 III LPC 2000 IIII LPC 2000 IIII LPC 2000 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	MCU settings	⊟ LPC 1300		
B: C/C++ General L-C1313 -Project References L-C1343 -References L-C1343 B: LPC100 B: LPC100 B: LPC3000 B: LPC3000 B: LPC3000 B: LPC3000 Ither script Create linker script Vices cript Memory T Name Fiesh MFlesh22 0x0 RAM RanLoc8 0x1000000 0x2000 0x2000	- Tool Chain Editor	LPC1311		
Project Rofferences LPC1343 LPC1343 LPC1343 LPC1343 LPC1343 LPC1343 LPC1343 LPC1300 LPC100 LPC100	€ C/C++ General	LPC1313		
Refactoring History Bi LPC100 Run/Debug Settings Bi LPC3000 Target architecture cortex-m3 Linker script Create linker script Memory T Name Location Flaich MFlash32 0x0 RAM RemuceS 0x1000000	Project References	LPC1342		
Run/Debug Settings Image: Context-m3 Target architecture Context-m3 Univer script Create linker script Memory T Name Location Size RAM RamLoc8 0x30000 RAM RamLoc8 0x20000	- Refactoring History	EPC1345		
Target architecture contex-m3 Unker script Create linker script Of Create linker script Memory T Memory T Name Location Size Flash 0x8000 RAM Remice3 0x10000000 0x2000	Run/Debug Settings	E LPC3000		
Memory T Name Location Size Flash MFlash32 0x0 0x8000 RAM RamLoc8 0x10000000 0x2000		Target architecture cortex-r	n3	
Flash MFlash32 0x0 0x8000 RAM RamLoc8 0x10000000 0x2000		Memory T Name	Location Size	
		Flash MFlash32 RAM RamLoc8	0x0 0x8000 0x10000000 0x2000	
	(?)		OK Cancel	

6.2.3 Optimization issues

When optimization is enabled, it will reorder code. What this means is that the code from multiple C lines will be intermingled. In addition, assignments and initializations might be pulled out of loops so they are only executed once. Changes like these will make the code confusing to debug. Some symptoms you might see are breakpoints that only work the first time through, or seeing the debugger's current line indicator fail to advance or even move backwards when you click step. It is best to always use –O0 for debugging. Since optimization can make such a big difference in code size and performance, it is a good idea to test your project with optimization and plan for a final build that is optimized.

6.2.4 Displaying assembly instructions

Click the i-> icon. This changes the Instruction Stepping Mode to step by processor instructions, rather than source lines. This also shows the disassembly view around the current instruction.

6.2.5 Exiting debug mode and stopping debugging

To stop debug press the 'Stop' button (red square) shown in the toolbar at the top of the debug view.

6.2.6 Recovery of board

After playing around with the LPCXpresso board, especially when trying out new PLL settings, reconfiguring the SWDIO/SWDCLK pin functions, disabling AHBCLKCTRL bits, or trying power down modes, the board may be disabled and no longer enter debug mode. This is caused by code on the on-board flash that incorrectly disables the system clocks or the debug port soon after reset before the debugger can connect to the core. The easiest solution to this is to load a working project into LPCXpresso, ground the ISP pin (see the chip User's Manual for details) and then try to enter debug mode.

Grounding the ISP pin during reset will put the target MCU into In-System Programming (ISP) mode. It will wait for a command through the serial port or the USB port. This temporarily prevents the troublesome code in flash from starting. Although ISP is designed to enable serial and USB updates, while ISP is running, the LPCXpresso toolchain is able to connect to the Cortex core and reprogram the flash. After the flash is

reprogrammed, disconnect the ISP pin (pull it high or allow it to float) and stop debugging. Now you should be able to debug code again.

6.3 Datasheet browser

The LPCXpresso IDE comes with an integrated web browser that will direct viewers to the datasheet of the device. Just click on the button below to see the browser in action.

minieror	controllore (- I DCY proces	
v Help	Juntouersy - Le CApresso	
• 🤅 📸	• 😂 😂 🖍 ½ 🛌 📬 🗮 🛄 💭 ½ ??] ⓒ timer16.c (ⓒ blinky.c (ⓒ gpio.c (ⓒ clkcor	🔎 İ 🛞 🐮 İ 🤹 İ 🐔 - 🕲 - İ 🌮 İ 🌮 - İ 🌮 - İ 🍄 - ⇒ 🛛 🖹 🔀 Develop ıfig.c. 🔞 Welcome 👔 http://www.standardics.nxp.com/products/lpc1000/datasheet/lpc1311.lpc131 🖄 🦵
⊑ 🙀 ▽	🗢 🔿 🔳 🔗 http://www.standardics.nxp.com/product	s/lpc1000/datasheet/lpc1311.lpc1313.lpc1342.lpc1343.pdf 🔹 🗸 💽
	🖶 🛅 🖏 🛧 🐥 1 53 🖲 🤅	66.8% • 🛃 🛃 Find •
ints 🗖	Bookmarks Image: Second state in the second state in the second state in the second state in the second state in the second state in the second state in the second state in the second state in the second state is the second state in the second state is the second state in the second state is the seco	Image: Display interface with the second
≡	C-Build [LPCXpresso1343_blinky]	
*		
8		
*		
	J.	LPCXpresso1343 blinky NXP/LPC1343

Fig 22. Integrated web browser

6.4 Code size

6.4.1 printf

When optimizing a project for size, if you are using printf, make sure that Redlib is selected as the standard library in the Projects Properties dialog. This option must be set in two different places, to configure both the header files, and the libraries. It should also be set both for the Debug target and the Release target. To get to the header file option, select settings in the C/C++ Build folder in the tree on the left. Make sure that the Tool Settings tab is active, and then select Target under MCU C Compiler in the tree on the

LPCXpresso Getting started with NXP LPCXpresso

right. Under the Configuration drop-down, select the Debug or Release target. Under Use headers for C library, choose Redlib. This is currently the default setting.

type filter text	13xx_Blinky Settings		
Resource - Builders - C/C++Build - Build Variables - Discovery Options - Environment which Setup	Configuration Debug [Active]	Build Artifact 🔛 Binary Parsers <table-cell> 📀</table-cell>	anage Configurations
-Settings Normal Editor - C/C ++ General - Project References - Refactoring History - Run/Debug Settings	MCU C Compiler Preprocessor Symbols Directories Optimization Wernings Warnings Wiscellaneous Target MCU Assembler @ General	Architecture Cortex-M3 Thumb mode Enable Thumb intervenders Use headers for Orbrary Redlib Redlib Newlib	
n 23 Header file opti	ion	and the second second second second second second second second second second second second second second secon	a fallen an eg talle samen eg an efter

To reconfigure the standard library setting for the linker, in the same dialog, select Target under MCU Linker. In this case, there are several options for Redlib. The default is Redlib (semihost) which allows full I/O to the PC through the LPCXpresso console.

	Settings $\bigcirc \bullet \bullet \bigcirc \bullet$
Resource Builders ⊂ C/C++ Build −Build Variables −Discovery Options −Environment ↑CO Sectors −Settings Tool Sectors Project References Refactoring History Run/Debug Settings	Configuration: Debug [Active] Manage Configurations Tool Settings Build Steps Build Artifact Binary Parsers Error Parsers MCU C Compiler Architecture Cortex-M3 Image Configurations Preprocessor Symbols Image Configuration Image Configuration Debugging Optimization Image Configuration Image Configuration Debugging Miscellaneous Redib (none) Image Configuration Miscellaneous Target Redib (nonost) Image Configuration Miscellaneous Target Miscellaneous Image Configuration Miscellaneous Saraget Miscellaneous Image Configuration Miscellaneous Shared Library Settings Shared Library Settings Image Configuration

The printf implementation in Redlib is about half the size of the implementation in Newlib. A smaller printf library can be used in Redlib if floating point formatting strings are not used. To select this smaller library, define the symbol CR_INTEGER_PRINTF to the compiler (i.e. -DCR_INTEGER_PRINTF). To save even more space, avoid using printf or

any C standard library functions and select Redlib (none). Depending on your printf settings and code, this could free up 10K to 20K of flash memory.

6.4.2 Optimization

Optimization can do a lot to save flash memory. It can be configured in the same dialog as the C standard library. Choose "Optimization" under "MCU C Compiler" in the "Tool Settings" tab. Higher levels of optimization will typically result in higher performance, but may result in larger code size. It is best to use –O0 for debugging and higher levels for Release. For best code size try –Os –mword-relocations. To further reduce code, add --gc-sections to the project linker flags. This causes the linker to remove unused functions from the compiled code. --gc-sections is enabled by default in new projects created by the project wizard. If you are working with an existing project, you may need to manually add this option to your project. --gc-sections is safe to use in both Release and Debug builds. There are many optimization options available for GCC. Visit http://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html to see all of them.

6.5 Showing hidden views

A view is an on-screen representation of something in the IDE. A view can be source code, the project tree, or a debug window. If you accidentally close a view, you can open it again by going to the Window menu and choosing Show View and Other. It is a good idea to browse through the Show View window to see what is available.



This will present a dialog allowing you to pick a view and display it.

Getting started with NXP LPCXpresso



6.6 Creating a 'skeleton' project in a new Workspace

LPCXpresso includes several project Templates to help get started quickly.

6.6.1 Create a new Workspace

From the 'File' menu hover over 'Switch Workspace' and then select 'Other...' from the bottom of the list. You will then see the 'Workspace Launcher' dialog shown below.

Enter or browse to the new path for your workspace. We have called our new workspace 'NewWorkspace'.

Select a w	orkspace
LPCXpresso s Choose a wo	tores your projects in a folder called a workspace. /kspace folder to use for this session.
Workspace:	C:\Documents and Settings\nxp21346\My Documents\pcxpresso\NewWorkspace V Browse
Copy Sett	ings
	OK Cancel

Then click on OK to re-open LPCXpresso with this new workspace selected.

6.6.2 Create the 'Skeleton' project

- If you are using a Cortex-based part, first, import the CMSIS header files for the chip family you are planning to work with. To do this, click "Import Example Project" again and navigate to the CMSIS<version/part>.zip. Once this project is added to your workspace, click "Build all projects (Debug)" in the Quickstart Panel.
- Click on the 'Projects and File Wizards' tab of the 'Quickstart Panel'.
- Click on 'MCU project wizards' and select "Create NXP Project' for your architecture.
- Enter a project name when the dialog appears. In this case we will use 'MyProject' then click 'Next.'

LPCXpresso Getting started with NXP LPCXpresso

U Quickstart Pan 🛛 🔲 Variables	● _● Breakpoints □□
Start here	× ^
Project and File wizards	*
e MCU project wizards 🔻	
₩₽ NXP LPC1100 project templates	
NXP LPC1300 project templates	🔸 🚝 Create NXP LPC 1300 Project
NVP NXP LPC1700 project templates	🕨 🚰 Create NXP LPC 1300 C 😽 Project (not sup
NRP NXP LPC2000 project templates	Create NXP LPC 1300 Hello World Project
h Create New C'Header He	No consoles to display a
Import and Export	
A REAL PROPERTY AND A REAL	and the second second second second second second second second second second second second second second second

 If you are using a Cortex-based part, the wizard will ask if you would like to use CMSIS. CMSIS stands for Cortex Microcontroller Software Interface Standard. CMSIS defines a common way to access peripheral registers and to define interrupts. Please check the "Use CMSIS" box and click "Next."

CMSIS Selec	tion	
Select whether appropriate CN	to use CMSIS and which version to use. Note: The ISIS base project must exist in the workspace.	
Use CMSIS		
CMSIS version	CMSISv1.30 LPC13xx	~

• The next section will ask you for "Source file information." You may fill out your Author and Copyright text.

• Then click next and this dialog appears so that you can select what build configuration you want to create. By default both 'Debug' and 'Release' are to be created which is the most commonly used setting.

	C Project
	Select platforms and configurations you wish to deploy on
	Project type: Executable Toolchains: Code Red MCU Tools Configurations:
	♥ ØØDebug Select all ♥ ØØRelease Deselect all
	Advanced settings
	Use "Advanced settings" button to edit project's properties. Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on property pages.
	Cancel
g 30. Select build	configuration

• Finally, the 'SelectProcessorType' dialog appears. In here you can select your target microcontroller. Then click on finish and you have a project for your selected microcontroller. In this example, we are selecting an LPC1343. You should make sure to select the microcontroller that is on your target board. After clicking "Finish," your sample project will be created.

Getting started with NXP LPCXpresso

et target processor type for project	<pre></pre>
	1
Farget	
NXP LPC 1343	
NXP	
LPC1300	
LPC1311	
LPC1313	
LPC1342	
LPC1343	
Target architecture cortex-m3	
inker script	
✓ Create linker script	
Memory Type Name Location S	ze
Flack MElack22 0.0	:8000
Flash MiFlash32 0x0 05	0000

Congratulations! You have created your first project!

LPCXpresso User guide

LPCXpresso

le Edit Source Refactor Navigate Search Pr	oject	Run Window Help
📬 • 🔚 🗁 📾 🧯 😗 🧹 🏇 💩 • 🐲 🕴 🔨 • 🗞 • 🗄 🗛 • 🍲 🖉 🖋	- CÎ	• 😂 C3 C 🔐 i 🏊 🖀 🎦 📮 i ⑦ 🔉 i ⊗ 🐯 🔛 K Develop ∥ i ↔ • ↔ -
🗅 Project E 🛛 🔐 Core Regi 🛃 Periphera 🖞	- 8	Welcome 🚺 main.c 🛛 🖓
	$\overline{\mathbf{N}}$	1/*
CMSISv 1p30_LPC13xx MyProject W Minores Includes P P r C r c r c r c min.c		<pre>2 3 Name : main.c 4 Author : 5 Version : 6 Copyright : (C) Copyright 7 Description : main definition 8</pre>
9 Quickstart 🖄 (M= Variables) 💁 Breakpoin		<pre>10 11#ifdefUSE_CMSIS 212#include "LPC13xx.h" 13#endif 14 215// TODO: insert other include files here 16</pre>
Start here		217// TODO: insert other definitions and declarations here 18
Import Example project(s) Build all projects (Debug) Suild 'MyProject' (Debug)	=	19 int main(void) { 20 21 // TODO: insert code here 22
🧹 Clean 'MyProject' (Debug)		
∜≽ Debug 'MyProject' (Debug)		C-Build [MyProject]
🦉 Quick Settings 🔹		MyProject.axf MyProject.hex ; text data bss dec hex filename
Project and File wizards		916 0 4 920 398 MyProject.axf
📸 MCU project wizards 🔹	~	
□◆		MyProject NXP_LPC1343

Fig 32. Project creation complete

Getting started with NXP LPCXpresso

7. Appendix

7.1 LPCXpresso target side schematics

LPCXpresso User guide

NXP Semiconductors

LPCXpresso Getting started with NXP LPCXpresso



User guide

Rev. 10 — 7 April 2011

User guide

32 of 51



LPCXpresso Getting started with NXP LPCXpresso

NXP

Semiconductors

User guide

LPCXpresso



NXP

LPCXpresso Getting started with NXP LPCXpresso

Fig 35. Schematic for the LPCXpresso LPC1769 target side (3 of 3)

FIG 35.

mbed

IF+

VOUT (3.3V out)

VU (5.0V USB out)

RD- (Ethernet)

RD+ (Ethernet)

TD- (Ethernet)

TD+ (Ethernet)

D- (USB)

D+ (USB)

CAN-RD

CAN-TD

PWMOUT0

PWMOUT1

PWMOUT2

PWMOUT3

PWMOUT4

PWMOUT5

UART3-TX / I2C2-SDA

UART3-RX / I2C2-SCL

≰¥闘 SPI2-MC GNDX AINO AIN1 AIN2 AIN4 AIN5

LPC1114 Target Side

LED

훭닞兴

			(superset o	of mbed pinning)		
mbed	LPCX	presso	Dual row ho	LPCXpress		
GND	GND		C J6-1		VOUT (+3	.3V out) if self else +3.3V input
VIN (4.5-14V)	VIN (4.5-5.5)	/)	EXT_POWX_C J6-2	(J6-29	not used	olo volo v nipu
VB (battery supply)	not used			(J6-30	not used	
nR (reset)	PIO0_0	RESET	C J6-4	(J6-31	not used	
SPI1-MOSI	PIO0_9	MOSI/SWO	PIO0_9C J6-5	(J6-32	not used	
SPI1-MISO	PIO0_8	MISO	PIO0_8C J6-6	(J6-33	not used	
SPI1-SCK	PI02_11	SCK	C J6-7	(J6-34	not used	
GPIO	PIO0_2	SSEL	PIO0_2C J6-8	(J6-35	not used	
UART1-TX / I2C1-SDA	PIO1_7	TXD	PIQ1_7C J6-9	(J6-36	USB_DM	Note wrong tex in silkscreen
UART1-RX / I2C1-SCL	PIO1_6	RXD	C J6-10	PIQ2_5(J6-37	USB_DP	Note wrong tex in silkscreen
SPI2-MOSI	PIO0_7		C J6-11	(J6-38	PIO0_1	in an autour out
SPI2-MISO	PIO2_0		C J6-12	PIQ0_3(J6-39	PIO0_3	
SPI2-SCL / UART2-TX	PIO2_1		C J6-13	FIQ0_5(J6-40	PIO0_5	I2C-SD/
UART2-RX	PIO2_2		C J6-14	PIQ0_4 (J6-41	PIO0_4	I2C-SCI
AINO	PIO0_11	AD0	C J6-15	PIQ1_9(J6-42	PIO1_9	
AIN1	PIO1_0	AD1	PIO1_0C J6-16	PIO1_10(J6-43	PIO1_10	
AIN2	PIO1_1	AD2	PIO1_1C J6-17	PIO1_11(J6-44	PI01_11	
AINS / AOUT	PIO1_2	AD3	C J6-18		PIO2_3	
AIN4	PIO1_3	AD4/SWDIO	PIO1_3C J6-19	PIO3_4(J6-46	PIO2_4	Note wrong tex in silkscreen
AIN5	PIO1_4	AD5	PIO1_4C J6-20	PIO3_5(J6-47	PIO2_5	Note wrong tex in silkscreen
	PIO1_5		PIO1_5C J6-21	PIO2_6(J6-48	PIO2_6	
	PIO1_8		PIO1_8(J6-22	PIO2_7(J6-49	PIO2_7	
	PIO0_6		PIO0_6C J6-23	PIO2_8(J6-50	PIO2_8	
	PIO0_10	SWCLK	PIO0_10C J6-24	PIO2_9(J6-51	PIO2_9	
	PIO3_0		C J6-25		PIO2_10	
	PIO3_1		PIO3_1C J6-26	PIO3_3(J6-53	PIO3_3	
	PIO3_2		PIO3_2(J6-27	GNDX(J6-54	GND	
	•				· · · · · · · · · · · · · · · · · · ·	

Expansion Connector

Fig 36. Schematic for the LPCXpresso LPC1114 target side²

R31 12K R36 0R

R35 0R

R33 0R

R32 0R

Note: SWO does not exist on LPC1114

BESET/PIO0_0

VCT16B0_CAP0 PIO0_3 PIO0_4/SCL PIO0_5/SDA PIO0_6/SCK0 PIO0_7/CTS

16 22

23

34 35

30

PIO0_1/CLKOUT/CT3280_MAT2 PIO0_2/SSEL0/CT16B0_CAP0

PIO0_8/MISO0/CT16B0_MAT0 PIO0_9/MOSI0/CT16B0_MAT1

SWCLK/PIO0_10/SCK0/CT16B0_MAT2 TDI/PIO0_11/AD0/CT32B0_MAT3

TMS/PIO1_01AD1/CT3281_CAP0 TDO/PIO1_11/AD2/CT3281_MAT0 TRST/PIO1_21/AD2/CT3281_MAT0 SWD0/PIO1_31/AD4/CT3281_MAT0 PIO1_41/AD5/CT3281_MAT3WA/KEUP PIO1_6/TRXD0/CT3280_CAP0 PIO1_6/TRXD0/CT3280_MAT0 BIO1_6/TRXD0/CT3280_MAT0

PIO1_6HXD/CT3259_MAT0 PIO1_7/TXD/CT3250_MAT1 PIO1_8/CT1661_CAP0 PIO1_9/CT1661_MAT0 PIO1_10/AD6/CT1661_MAT1 PIO1_11/AD7

PIO2_0/DTR/SSEL1 PIO2_1/DSR/SCK1

PIO2_2/DCD/MISO1 PIO2_3/RI/MOSI1

PIO2_4 PIO2_5 PIO2_6 PIO2_7

PI02_8 PI02_9 PI02_10

PIO2 11/SCK0

PIO3_0/DTR PIO3_1/DSR PIO3_2/DCD PIO3_3/RI PIO3_3/RI

PI03_4 PI03_5

37 43 PIO3_1 PIO3_2 PIO3_3 PIO3_4 PIO3_5

2100_1 PIO0_2

PIO0_3 PIO0_4 PIO0_5 PIO0_6

PIC0 5 PIC0 7 PIC0 8 PIC0 9 PIC0 10 PIC0_11

PIO1_0 PIO1_1

PIO1_2 PIO1_3 PIO1_4 PIO1_5 PIO1_6 PIO1_6

PIO1_8 PIO1_9

PI01 11

PIO2 0

PIO2_1

PIO2 8

PIO3 0

PI02 9

2. Design and layout compatible with LPC1343 version. Therefore, PIO2_4/5 and PIO3_4/5 swap. LPC1114 does not have USB, but LPC1343 does. Therefore R37/38. LPC1114 does not have SWO, but PIO0_9 is connected (since LPC1343 has SWO there).

Getting started with NXP LPCXpresso PCXpress

Ö

From LPC-LINK Side

VIO 3V3X JTAG TMS SWDIOX

JTAG TOLK SWOLKX

U7 LPC1114

XTALIN XTALOUT

VDDIO

VDDCORE

VSSIO VSS

JTAG TDO SWOX

JTAG RESETX

EXT POWX OND

Y2 12MHz

C38 180

C40

T10n

GNDX

C42

10n

41

GNDX

GNDX

GND

C39 100n

GNDX

C41

GNDX

100n

-In I

C37

GND

VIO 3V3X

34 o,



PCXpress

Ö



Fig 37. Schematic for the LPCXpresso LPC1343 target side

© NXP B.V. 2011. All rights reserved 35 of 51

i discia

₽

INIO

User guide



Fig 38. Schematic for the LPCXpresso LPC11C24 target side, part 1

₽

NXP Semiconductors

LPCXpresso Getting started with NXP LPCXpresso

						ے LPC-LII	NK side							
						Expansio (superse	on Conne t of mbe	ector d pinning)					
mbed	LPCXpresso		Dual row h	oles (2x27), 100	mil spacing							LPCX	presso	mbed
GND	GND	GNDX	J6-1							J6-28)	VIO_3V3X	VOUT (+3.3)	/ out) if self	VOUT (3.3V out)
VIN (4.5-14V)	VIN (4.5-5.5V)	EXT_POWX								J6-29)-		not used	e +3.3 v input	VU (5.0V USB out)
VB (battery supply)	not used		-C J6-3							J6-30 >		not used		IF+
nR (reset)	PIO0_0 RESET	PIO0_0								J6-31)-		not used		IF-
SPI1-MOSI	PIO0_9 MOSI0/SWO	9	J6-5							J6-32)-		not used		RD- (Ethernet)
SPI1-MISO	PIO0_8 MISO0		C J6-6							J6-33)-		not used		RD+ (Ethernet)
SPI1-SCK	PIO2_11 SCK0	_PIO2_11	J6-7							J6-34)-		not used		TD- (Ethernet)
GPIO	PIO0_2 SSEL0	PIO0_2	J6-8							J6-35 >		not used		TD+ (Ethernet)
UART1-TX / I2C1-SDA	PIO1_7 TXD	PIO1_7	(J6-9							J6-36)-		not used		D- (USB)
UART1-RX / I2C1-SCL	PIO1_6 RXD		(J6-10							J6-37) —		not used		D+ (USB)
SPI2-MOSI	PIO0_7	PIO0_7	(J6-11							J6-38) ——	PIO0_1	PIO0_1		CAN-RD
SPI2-MISO	PIO2_0	PIO2_0	C J6-12							J6-39)	PIO0_3	PIO0_3		CAN-TD
SPI2-SCL / UART2-TX	PIO2_1	PIO2_1	 J6-13							J6-40) ——	PIO0_5	PIO0_5	I2C-SDA	UART3-TX / I2C2-SDA
UART2-RX	PIO2_2	PIO2_2	(J6-14							J6-41)	PIO0_4	PIO0_4	2C-SCL	UART3-RX / I2C2-SCL
AINO	PIO0_11 AD0	PIO011	(J6-15							J6-42)	PIO3_3	PIO3_3		PWMOUT0
AIN1	PIO1_0 AD1	PIQ1_0	C J6-16							J6-43)	PIO1_10	PIO1_10		PWMOUT1
AIN2	PIO1_1 AD2	1	C J6-17	GNDX				VIO 3V3	,	J6-44) ——	PIO1_11	PIO1_11		PWMOUT2
AIN3 / AOUT	PIO1_2 AD3	_PIO1_2	C J6-18						`	J6-45)	PIO2_3	PIO2_3		PWMOUT3
AIN4	PIO1_3 AD4/SWDIO	_PIO1_3	C J6-19	-OPAD1	PAD5	-OPAD9	-OPAD13	PAD17	OPAD21	J6-46)	<u>PIO2_6</u>	PIO2_6		PWMOUT4
AIN5	PIO1_4 AD5	PIO1_4	(J6-20	-OPAD2	-OPAD6	-OPAD10	-OPAD14	-OPAD18	-OPAD22	J6-47) —		not used		PWMOUT5
	PIO1_5	PIO1_5	C J6-21	-OPAD3	-OPAD7	-OPAD11	-OPAD15	-OPAD19	-ØPAD23	J6-48) —		not used		
	PIO1_8	8	(J6-22	-OPAD4	-OPAD8	GNDX	OPAD16	-OPAD20	-OPAD24	J6-49) ——	PIO2_7	PIO2_7		1
	PIO0_6	PIO0_6	C J6-23			GINDA				J6-50)	PIO2_8	PIO2_8		1
	PIO0_10 SWCLK		(J6-24							J6-51) —		not used		1
	PIO3_0	PIO3_0	(J6-25							J6-52) ——	PIO2_10	PIO2_10		1
	PIO3_1	PIO3_1	(J6-26							J6-53) —		not used		1
	PIO3_2	PIO3_2	(J6-27							J6-54)	GNDX	GND		1

All information provided in this document is subject to legal disclaimers Rev. 10 — 7 April 2011

Fig 39. Schematic for the LPCXpresso LPC11C24 target side, part 2



Fig 40. Schematic for the LPCXpresso LPC1200 target side, part 1

LPCXpresso Getting started with NXP LPCXpresso

NXP

Semiconductors

User guide

All information provided in this document is subject to legal disclaimer Rev. 10 — 7 April 2011

39 of 51



Getting started with NXP LPCXpresso

Fig 41. Schematic for the LPCXpresso LPC1200 target side, part 2

0 Z

© NXP B.V. 2011. All rights





NXP Semiconductors

LPCXpresso Getting started with NXP LPCXpresso



NXP

Semiconductors

Getting started with NXP LPCXpresso



© NXP B.V. 2011. All rights

Fig 44. Schematic for the LPCXpresso LPC11U14 target side, part 2

LPCXpresso

7.2 LPCXpresso PCB pinout and dimensions

The schematics of the LPCXpresso Target and the LPC-LINK debug connector appear in Fig 36 and Fig 37. The LPCXpresso board was designed to be pin compatible with NXP mbed. LPCXpresso can be powered either through the debug mini-USB port, by 3.3V applied to the board, or by 5V applied to the USB connector. A cable for the 10-pin mini JTAG connector on the LPC-LINK debugger portion of LPCXpresso can be purchased from Digi-Key, part number FFSD-05-D-06.00-01-N.

Dimensions: LPCXpresso LPC1343's outer dimensions are 1.35x5.45 inches. It contains two rows of holes 900 mil apart. Each row has 27 connections and holes are drilled at a 100 mil pitch.



7.3 Enabling USB connectivity "to LPC1343 target"

The LPCXpresso board is simple yet flexible. Here is a way to configure it to support the development of USB devices using the LPC1343 or other USB-capable NXP microcontroller. The LPC1343 has a USB phy on-chip. To connect the microcontroller to a USB port, it is only necessary to add a USB connector and a pullup resistor.

Note: This simple connection does not implement NXP Soft-Connect to allow soft disconnection and connection to the USB bus. Because of this, the USB connection must be plugged into the PC near the time the USB peripheral is initialized, or after. If the USB port is connected when the LPC USB peripheral is not initialized, the pullup resistor will notify the PC that a USB device is present, yet the microcontroller will not respond because it has not been initialized. This will trigger windows to generate an error regarding a malfunctioning USB device. Unplug and re-plug the device to dismiss the error.

Note 2: Rather than building a cable or wiring a USB Type-A connector, you could take an existing A-B USB cable and cut off the B connector. Then the A side of the cable could be stripped and soldered onto the LPCXpresso board.

LPCXpresso Getting started with NXP LPCXpresso





© NXP B.V. 2011. All rights reserved.

7.4 Terminology

LPCXpresso

The Code Red Technologies IDE (Integrated Development Environment) based on Eclipse with our own extensions for embedded development.

SWD

Serial Wire Debugging (Single Wire Debugging). This is a debug connection technology available on the Cortex-M3 that allows debug through just 2-wires unlike 5 for JTAG.

ELF (Executable and Linking Format)

This is the object code file format used by our development tool chain and most microprocessor tool chains.

Workspace

LPCXpresso organizes groups of projects into a 'Workspace'. A workspace is stored as a directory on your host PC and has subdirectories containing individual projects.

Project

An LPCXpresso project. A project contains all of the .c and .h files to build a single microcontroller flash image.

Perspective

In LPCXpresso, a perspective is a particular collection of 'Views' that are grouped together to be suitable for a particular use. For example the 'C/C++ programming' perspective and the 'Debug' perspective.

View

A 'View' is a window in LPCXpresso that shows a particular file or activity. A view could be of a C source code file or something live such as a disassembly window or register dump. A 'Perspective' is the layout of many 'Views'.

Semi-hosting

The ability to use IO on your debugger host system for your target embedded system. For example a 'printf' will appear in the console window of the debugger.

Debug Target

The system being debugged. LPCXpresso includes a target microcontroller on-board, but can also be connected to external targets.

Redlib[™]

The optimized Code Red Technologies C runtime library (non-GNU). LPCXpresso includes both Redlib and Newlib libraries.

7.5 File Icons and their meaning

Table 1. File icons and their mear	ning
Icon	Meaning
.0	C language source file.
.h	C language header file.
· 👌	Project Makefile. Note that LPCXpresso may automatically generate these.
٦	C language source file that has been excluded from the project build. – Hollowed out character in icon.
	General text file. Often used for files such as linker script files which end with the extension '.ld'.
i 🥻 (The blue double arrows at the top of the icon denote that this particular source file has different project build properties than the rest of the project.
<u>.</u>	The orange/gold 'storage drive' denotes that this file is connected to a CVS repository.
Main.c 1.1 (ASCII -kkv)	The right arrow denotes that the file or directory has been modified locally and may need committing to the CVS repository, i.e., the local copy is more up to date than the repository.
榜	This is an executable file.
	Directory containing executables.
1 Contraction of the second se	C Project Directory (Project Explorer View)
E.	C Project Directory linked to a CVS repository. (Project Explorer View)

Getting started with NXP LPCXpresso

7.6 References

- [1] NXP LPCXpresso http://www.nxp.com/lpcxpresso
- [2] NXP LPCZone http://www.nxp.com/lpczone
- [3] NXP Microcontrollers http://www.nxp.com/microcontrollers
- [4] Code Red Technologies Wiki <u>http://lpcxpresso.code-red-tech.com/LPCXpresso/softwareknowledgebase</u>
- [5] Code Red Technologies LPCXpresso page <u>http://www.code-red-tech.com/lpcxpresso</u>
- [6] Embedded Artists AB http://www.embeddedartists.com
- [7] C, A Reference Manual, Fifth Edition. Samuel P. Harbison III, Guy L. Steele Jr. Prentice Hall.
- [8] The Definitive Guide to the ARM Cortex-M3, Joseph Yiu. Newnes
- [9] ARM Cortex-M3 Technical Reference Manual. Revision: r2p0.(ARM DDI 0337G). http://infocenter.arm.com/help/index.jsp
- [10] ARM Cortex-M0 Technical Reference Manual. Revision: r1p0.(ARM DDI 0413D). http://infocenter.arm.com/help/index.jsp
- [11] ARMv7-M Architecture Reference Manual (ARM DDI 0403) http://infocenter.arm.com/help/index.jsp
- [12] ARMv6-M Architecture Reference Manual (Cortex-M0/LPC11) http://infocenter.arm.com/help/index.jsp

LPCXpresso

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

8.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

Getting started with NXP LPCXpresso

9. List of figures

Fig 1.	LPCXpresso IDE4
Fig 2.	LPCXpresso development board5
Fig 3.	Product development stages7
Fig 4.	Setup wizard8
Fig 5.	Setup wizard9
Fig 6.	Setup wizard9
Fig 7.	LPCXpresso activation with offline PC10
Fig 8.	Show view/other menu11
Fig 9.	Single perspective (develop)12
Fig 10.	Single perspective (debug)13
Fig 11.	Display of peripheral view showing selection of ADC peripheral14
Fig 12.	Display of memory view showing detail of ADC peripheral registers14
Fig 13.	USB 2.0 A/Mini-B cable15
Fig 14.	Quickstart panel16
Fig 15.	Examples directory17
Fig 16.	Debug
Fig 17.	Bug icon
Fig 18.	Debug toolbar
Fig 19.	Debug buttons19
Fig 20.	Current part number20
Fig 21.	Selecting correct part number21
Fig 22.	Integrated web browser22
Fig 23.	Header file option23
Fig 24.	Reconfigure library setting23
Fig 25.	Show view window24
Fig 26.	Pick a view to display25
Fig 27.	Workspace launcher25
Fig 28.	Enter project name26
Fig 29.	CMSIS selection26
Fig 30.	Select build configuration27
Fig 31.	Select processor type28
Fig 32.	Project creation complete29
Fig 33.	Schematic for the LPCXpresso LPC1769 target side (1 of 3)31
Fig 34.	Schematic for the LPCXpresso LPC1769 target side (2 of 3)32
Fig 35.	Schematic for the LPCXpresso LPC1769 target side (3 of 3)33
Fig 36.	Schematic for the LPCXpresso LPC1114 target side
Fig 37.	Schematic for the LPCXpresso LPC1343 target side
Fig 38.	Schematic for the LPCXpresso LPC11C24
	target side, part 1

 Schematic for the LPCXpresso LPC11C24 target side, part 2	Fig 39.
 Schematic for the LPCXpresso LPC1200 target side, part 1	Fig 40.
1. Schematic for the LPCXpresso LPC1200 target side, part 2	Fig 41.
2. Dimensioned drawing of LPCXpresso LPC1227, LPC1343, LPC1114/301, LPC1114/302, LPC11C24, LPC1769, LPC176840	Fig 42.
 Schematic for the LPCXpresso LPC11U14 target side, part 141 	Fig 43.
 Schematic for the LPCXpresso LPC11U14 target side, part 242 	Fig 44.
5. LPCXpresso LPC-LINK JTAG/SWO pinout43	Fig 45.
6. USB retrofit schematic44	Fig 46.
7. LPC1343 target connected to Type-A USB cable44	Fig 47.

10. List of tables

LPCXpresso User guide

Getting started with NXP LPCXpresso

11. Contents

1.	Introduction3
1.1	LPCXpresso IDE4
1.2	LPCXpresso development board5
1.3	LPC-LINK JTAG/SWD debugger5
1.4	Integrated evaluation target5
1.5	LPCXpresso partners5
2.	Evaluate, explore and develop7
3.	Installation8
3.1	System requirements8
3.2	Installation process8
3.3	Activation10
4.	Getting familiar with the LPCXpresso IDE11
4.1	Layout of the LPCXpresso desktop11
4.1.1	Single perspective (code development)12
4.1.2	Single perspective (debug)13
4.1.2.1	Peripheral views14
4.2	Connecting the target15
5.	Blinky: Build, download and debug16
5.1	Importing the blinky project using the Quickstart
	panel16
5.2	Debugging/running 'blinky' on your LPCXpresso
	board17
-	
6.	LPCXpresso IDE tips and tricks20
6. 6.1	LPCXpresso IDE tips and tricks
6. 6.1 6.2	LPCXpresso IDE tips and tricks
6. 6.1 6.2 6.2.1	LPCXpresso IDE tips and tricks
6. 6.1 6.2 6.2.1 6.2.2	LPCXpresso IDE tips and tricks
6. 6.1 6.2 6.2.1 6.2.2	LPCXpresso IDE tips and tricks
6. 6.1 6.2 6.2.1 6.2.2 6.2.3	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying concerning the instantiana 21
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.3 6.2.4 6.2.5	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug 21
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.3 6.2.4 6.2.5 6.2.5	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting debug 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Recovery of board 21
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.3 6.2.4 6.2.5 6.2.6 6.3	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting debug 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Recovery of board 21 Datasheet browser 22
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4 1	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.1 6.4.2	LPCXpresso IDE tips and tricks20Installing Eclipse plugins20Debugging tips20Debug features not enabled20Incorrect registers displayed or errors starting20Optimization issues21Displaying assembly instructions21Exiting debug mode and stopping debugging21Datasheet browser22Code size22printf22Optimization24
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5	LPCXpresso IDE tips and tricks20Installing Eclipse plugins20Debugging tips20Debug features not enabled20Incorrect registers displayed or errors starting20Optimization issues21Displaying assembly instructions21Exiting debug mode and stopping debugging21Datasheet browser22Code size22printf22Optimization24
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5 6.6	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5 6.6	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24 Creating a 'skeleton' project in a new Workspace 25
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5 6.6	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24 Creating a 'skeleton' project in a new Workspace 25 Create a new Workspace 25
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5 6.6 6.6.1 6.6.2	LPCXpresso IDE tips and tricks20Installing Eclipse plugins20Debugging tips20Debug features not enabled20Incorrect registers displayed or errors starting20Optimization issues21Displaying assembly instructions21Exiting debug mode and stopping debugging21Datasheet browser22Code size22printf22Optimization24Showing hidden views24Creating a 'skeleton' project in a new Workspace25Create the 'Skeleton' project25
 6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5 6.6 6.6.1 6.6.2 7. 	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24 Creating a 'skeleton' project in a new Workspace 25 Create a new Workspace 25 Create the 'Skeleton' project 25 Appendix 30
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.5 6.6 6.6.1 6.6.2 7. 7.1	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24 Creating a 'skeleton' project in a new Workspace 25 Create a new Workspace 25 Create the 'Skeleton' project 25 Appendix 30
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.4.2 6.5 6.6 6.6.1 6.6.2 7. 7.1 7.2	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24 Creating a 'skeleton' project in a new Workspace 25 Create a new Workspace 25 Create the 'Skeleton' project 30 LPCXpresso target side schematics 30 LPCXpresso PCB pinout and dimensions 43
6. 6.1 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.3 6.4 6.4.1 6.5 6.6 6.6.1 6.6.2 7. 7.1 7.2 7.3	LPCXpresso IDE tips and tricks 20 Installing Eclipse plugins 20 Debugging tips 20 Debug features not enabled 20 Incorrect registers displayed or errors starting 20 Optimization issues 21 Displaying assembly instructions 21 Exiting debug mode and stopping debugging 21 Datasheet browser 22 Code size 22 printf 22 Optimization 24 Showing hidden views 24 Creating a 'skeleton' project in a new Workspace 25 Create a new Workspace 25 Create the 'Skeleton' project 30 LPCXpresso target side schematics 30 LPCXpresso PCB pinout and dimensions 43 Enabling USB connectivity "to LPC1343 target"

		43
7.4	Terminology	45
7.5	File Icons and their meaning	46
7.6	References	47
8.	Legal information	48
8.1	Definitions	48
8.2	Disclaimers	48
8.3	Trademarks	48
9.	List of figures	49
10.	List of tables	50
11.	Contents	51

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

© NXP B.V. 2011.

All rights reserved.

For more information, please visit: http://www.nxp.com For sales office addresses, please send an please send an email to: salesaddresses@nxp.com

Date of release: 7 April 2011 Document identifier: LPCXpresso