

# LPC4330-Xplorer





## **About NGX Technologies**

NGX Technologies is a premier supplier of development tools for the ARM7, ARM Cortex M0, M3 and M4 series of microcontrollers. NGX provides innovative and cost effective design solutions for embedded systems. We specialize in ARM MCU portfolio, which includes ARM7, Cortex-M0, M3 & M4 microcontrollers. Our experience with developing evaluation platforms for NXP controller enables us to provide solutions with shortened development time thereby ensuring reduced time to market and lower development costs for our customers. Our cost effective and feature rich development tool offering, serves as a testimony for our expertise, cost effectiveness and quality.

# **Contact Information:**

NGX Technologies Pvt. Ltd. No.216, 5th main Road, R.P.C. Layout, Vijayanagar 2nd Stage, Bangalore – 560 104 Phone : +91-80-40925507 email:sales@ngxtechnologies.com

# **CE certification:**

NGX Technologies LPC4330-Xplorer board has been tested for radiated emission as per EN55022 class a standard. The device is under the limits of the standard EN55022 class A and hence CE marked. No other test have been conducted other than the radiated emission (EN55022 class A standard). The device was tested with the ports like USB, Serial, and Power excluding the GPIO ports. Any external connection made to the GPIO ports may alter the EMC behavior. Usage of this device under domestic environment may cause unwanted interference with other electronic equipment's. User is expected to take adequate measures. The device is not intended to be used in and end product or any subsystem unless the user re-evaluates applicable directive/conformance.



# **Table of Contents**

1.0 INTRODUCTION	4
2.0 LPC4330-Xplorer Development Tool Setup	5
2.1 ULINK2/ME and KEIL	5
2.2 Installation & Configuration of KEIL software	5
2.3 Configuration of ULINK2/ME Debugger	9
2.4 Configuring External Quad Flash	10
2.5 Setup for ULINK2/ME and LPC4330 Xplorer Board	12
3.0 LPC4330 Xplorer firmware Development	17
3.1 Creating the sample (Blinky) project in KEIL	17
3.2 Executing the sample project in KEIL	33
4.0 Restoring Xplorer to Factory Defaults	35
4.1 ULINK2/ME and KEIL	35
5.0 Schematic & Board Layout	36
5.1 Schematic	36
5.2 Board layout	36
6.0 CHANGE HISTORY	37
6.1 Change History	37



# **1.0 INTRODUCTION**

This document is the 'User Manual' for LPC4330-Xplorer; a cost effective evaluation platform for NXP's LPC43xx (dual core Cortex M4 and M0) MCUs. This document reflects its contents which include system setup, debugging, and software components. This document provides detailed information on the overall design and usage of the board from a systems perspective.

Before proceeding further please refer the <u>Quick Start Guide</u> for Xplorer features, Xplorer Unboxing and Xplorer verification. Kindly refer to the <u>product page</u> for the latest information.

*Note: To restore the Factory Default for 'LPC4330 Xplorer Board' kindly refer to <u>section 4.0</u>* 



# 2.0 LPC4330-Xplorer Development Tool Setup

## 2.1 ULINK2/ME and KEIL

NGX's MCU evaluation platforms are not coupled tightly with any one particular combination of IDE and debugger. The following sections will explain the setup for KEIL and ULINK2/ME as the IDE and debugger respectively.

## 2.2 Installation & Configuration of KEIL software

The Installation of KEIL (MDK-ARM V4.23) software is explained below:

Step 1: Open the KEIL setup



Fig.1



Step 2: Click on Next.

Setup MDK-ARM V4.23	X
Welcome to Keil µVision Release 12/2011	
This SETUP program installs: MDK-ARM V4.23	
This SETUP program may be used to update a previous However, you should make a backup copy before proce	product installation. eding.
It is recommended that you exit all Windows programs be	fore continuing with SETUP.
Follow the instructions to complete the product installation	ı.
— Keil µVision4 Setup ————————————————————————————————————	<< Back Next >> Cancel

Fig.2

Step 3: To accept the agreement, click the check box and click Next.

Setup MDK-ARM V4.23
License Agreement Please read the following license agreement carefully.
To continue with SETUP, you must accept the terms of the License Agreement. To accept the agreement, click the check box below.
END USER LICENCE AGREEMENT FOR MDK-ARM THIS END USER LICENCE AGREEMENT ("LICENCE") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR SINGLE LEGAL ENTITY) AND ARM LIMITED ("ARM") FOR THE USE OF THE SOFTWARE ACCOMPANYING THIS LICENCE. ARM IS ONLY WILLING TO LICENSE THE SOFTWARE TO YOU ON CONDITION THAT YOU ACCEPT ALL OF THE TERMS IN THIS LICENCE. BY CLICKING "I AGREE" OR BY INSTALLING OR OTHERWISE USING OR COPYING T
Keil uVision4 Setup
<< Back Next >> Cancel

Fig.3



Step 4: Provide the destination path and click on Next

Setup MDK-ARM V4.23	×
Folder Selection Select the folder where SETUP will install files.	
SETUP will install µVision4 in the following folder. To install to this folder, press 'Next'. To install to a different folder. Destination Foldor C:\Keil	folder, press 'Browse' and select another B <u>r</u> owse
— Keil µVision4 Setup	<< Back Next >> Cancel

Fig.4

Step 5: Fill your personal information and click on Next

Setup MDK-ARM V4.2	3	
Customer Informal Please enter your	ion nformation.	
Please enter your n First Name:	ame, the name of the company for whom you work	and your E-mail address.
Last Name: Company Name:	xyz NGX Technologies Pvt Ltd	
<b>E-mail:</b> — Keil µVision4 Setup	abc@ngxtechnologies.com	
	Ka	ck Next>> Cancel

Fig.5



Step 6: Click on Next

Setup MDK-ARM V4.23	×
File installation completed	
μVision Setup has installed all files successfully.	
Retain current μVision configuration.	
Add example projects to the recently used project list.	
Preselect Example Projects for	
Simulated Hardware	<b>_</b>
— Keil μ∀ision4 Setup —	<< Back Next >> Cancel

Fig.6

Step 7: KEIL µVision4.23 setup is completed, click Finish.



Fig.7



## 2.3 Configuration of ULINK2/ME Debugger

The configuration flow of 'ULINK2/ME Debugger' is explained below:

Step 1: Open the KEIL blinky project downloaded from the website and click on the 'Target Options'.





Step 2: The window opens as shown in the following image, click on Debug and select the 'ULINK2/ME Cortex Debugger' as shown in the following image.

😨 Options for Target 'Internal SRAM'	
Device Target Output Listing User C/C++ Asm	Linkel Debug Utilities
C Use Simulator Settings	Use ULINK2/ME Cortex Debugger Settings
Load Application at Startup     Initialization File:      Edit	I✓         Load Application at Startup         I✓         Run to main()           Initialization File:          Edit
Restore Debug Session Settings         I reakpoints         I reakpoints	Restore Debug Session Settings         I reakpoints       I Toolbox         I Watch Windows         I Memory Display
CPU DLL: Parameter: SARMCM3.DLL -MPU	Driver DLL: Parameter: SARMCM3.DLL - MPU
Dialog DLL: Parameter: DCM.DLL pCM4	Dialog DLL: Parameter: TCM.DLL pCM4
ОК Са	ncel Defaults Help

Fig.9



Step 3: Next click on the 'Settings Option', the 'Cortex-M Target Driver Setup' window opens then select SW port. After selection of the SW port the ULINK2/ME detected is as shown in the following image.

Coptions for Target 'Internal SRAM' Device   Target   Output   Listing   User   C/C++   Asm   Linker Debug   Utilities	x i
C Use Simulator Settings © Use: ULINK2/ME Cortex Debugge	er 🗸 Settings
Cortex-M Target Driver Setup           Debug         Trace         Rash Download           ULINK USB - JTAG/SW Adapter         SW Device           Serial No:         V1530BINE           ULINK Version:         ULINK2           Device Family:         Cortex-M           Firmware Version:         V1.42           Image: SWJ         Cortex-M           Max Clock:         TML           Max Clock:         TML	Move Up Down
Debug       Connect & Reset Options       Do         Connect & Normal       ▼       Reset:       VECTRESET       ▼         I ✓       Reset after Connect       ✓       Cache Memory       □	wnload Options Verify Code Download Download to Flash
OK Cancel	Help

**Fig.10** 

(Note : The Cortex M4 can be programed using SW or JTAG, while the Cortex M0 is visible through JTAG port.)

# 2.4 Configuring External Quad Flash

Step 1: Open the KEIL blinky project, click on the 'Target Options'.



**Fig.11** 



Step 2: A window opens as shown in the following image, click on Utilities and select 'ULINK2/ME Cortex Debugger'.

Options for Target 'SPIFI 32MB Debug'	x
Device   Target   Output   Listing   User   C/C++   Arm   Linker   Debu Utilities	
Configure Rash Menu Command	III
Use Target Driver for Rash Programming	
ULINK2/ME Cortex Debugger Settings Update Target before Debugging	
Int Fle: Edt	
C Use External Tool for Rash Programming	
Command:	
Arguments:	
Bun Independent	
	- 11
OK Cancel Defaults He	
	<u> </u>

**Fig.12** 

Step 3: Select 'Update Target before Debugging' check box, click on 'Settings Option' and then in 'Cortex-M Target Driver Setup' click add and select 'LPC18xx/43xx S25FL032 SPIFI programing algorithm' and click add.

Configure @ Use	Rash Menu Command Target Driver for Rash Programming ULINK2/ME Cottex Debugger	date Target before Debug	ono	
C U	Debug   Trace Rish Download	Add Flash Programming Algor	ithm	
Ang	LOAD	Description HT32 Series Rish Options HT32 Series Rish Options HT32 Series Rish Options HT32 Series Rish Dual Rish LM3500 128:6 Rish LM3500 256:6 Rish LM3500 256:6 Rish LM3500 513:6 Rish LM3500 513:6 Rish LM3500 513:6 Rish LM3500 6:128:6 Rish LM3500 8:6 Rish LM3500 256:8 Rish LM4500 23:0 Rish LM4500 23:0 Rish LM4500 23:0 Rish	Device Type On-chip Rash On-chip Rash	Device Size 128k 4k 64M 128k 19k 256k 32k 384k 512k 64k 256k 32k 384k 512k 64k 256k 33k 64k

Fig.13

Click OK to complete the 'ULINK2/ME Debugger configuration'.



## 2.5 Setup for ULINK2/ME and LPC4330 Xplorer Board

#### **Option A: With ULINK-ME debugger**

To run the KEIL examples you will need the following and the image shows the each components:

- ULINK-ME
- 10-pin ribbon cable
- LPC4330 Xplorer Board
- 2 USB AM to Micro B cable



**Fig.14** 

Steps to setup the ULINK-ME and LPC4330 Xplorer Board: (*Note: Please refer <u>keil knowledgebase article</u> for Connecting ULINK2/ME 10-pin ribbon cable to NGX Xplorer)* 

Step 1: Connect one end of 10-pin ribbon cable to 'ULINK-ME 10-pin box header' as shown in the following image.







Step 2: Connect other end of 10-pin ribbon cable to '10-pin box header' of the LPC4330 Xplorer board as shown in the following image.



Fig.16

Step 4: Connect one end of 'USB AM to Micro B cable' to LPC4330 Xplorer board and other end to computer and connect one end of 'USB AM to Micro B' to ULINK-ME and other end to computer. As shown in the following image.



Fig.17

Step 5: The setup is now ready to be used for development with **KEIL IDE** and **ULINK-ME**.



#### **Option B: With ULINK2 debugger**

The Xplorer board has on board '10-pin SWD/JTAG box'; ensure that the ULINK2/ME must have '10-pin SWD/JTAG' support for development. The ULINK2 debugger, '20-pin to 10-pin adapter' and 10-pin ribbon cable are not a part of the LPC4330 Xplorer package, the user needs to buy separately.

To program/debug the KEIL examples you will need the following and the image shows the each components:

- ULINK2
- ARM JTAG to Cortex JTAG Adapter (20-pin to 10-pin Adaptor)
- 10-pin ribbon cable
- LPC4330 Xplorer Board
- One USB AM to Micro B cable



Fig.18

Steps to setup the ULINK2 and LPC4330 Xplorer Board: (*Note: Please refer <u>keil knowledgebase article</u> for Connecting ULINK2/ME 10-pin ribbon cable to NGX Xplorer)* 

Step 1: Connect one end of 10-pin ribbon cable to '20-pin to 10-pin adapter' as show in following image.



Fig.19



Step 2: Connect other end of 10-pin ribbon cable to '10-pin box header' of the LPC4330 Xplorer board as shown in the following image.





Step 3: Now connect the 'ULINK2 20-pin cable' to '20-pin to 10-pin adapter' as shown in the following image.



Fig.21



Step 4: Connect one end of 'USB AM to Micro B' cable to LPC4330 Xplorer board and other end to computer, connect one end of 'USB type B' to ULINK2 and other end to computer, as shown in the following image.



Fig.22

Step 5: The setup is now ready to be used for development with KEIL IDE and ULINK2.



# 3.0 LPC4330 Xplorer firmware Development

# 3.1 Creating the sample (Blinky) project in KEIL

Steps to create the sample (Blinky) project:

Step 1: Open a KEIL IDE.





Step 2: Click on Project->New uVision Project... as show below.



Fig.24



Step 3: Create a new folder in downloaded sample example folder and rename to Blinky and select Blinky folder click Open.

Organize 🔻 New folder				(
🔚 Libraries 🔦	Name	Date modified	Туре	
Documents	📙 Blinky	07-06-2012 PM 04:	File folder	
J Music	CMSISVZp10_LPC43xx_DriverLib	28-05-2012 AM 10	File folder	
Pictures	LPC4330_Xplorer_Audio	28-05-2012 PM 12:	File folder	
Videos	📙 LPC4330_Xplorer_Blinky	28-05-2012 AM 10	File folder	
🔏 Nagaraj	LPC4330_Xplorer_Default_Setting	28-05-2012 AM 10	File folder	
Computer	LPC4330_Xplorer_DualCore	28-05-2012 AM 10	File folder	
S (C:)	LPC4330_Xplorer_Ethernet	28-05-2012 AM 11	File folder	
New DVD RW Drive	LPC4330_Xplorer_ExtInt	28-05-2012 AM 11	File folder	
Local Disk (F:)				
Ele nome				
rile name:				
Save as type: Project F	iles (*.uvproj)			

Fig.25

Step 4: Give a project name, example: 'LPC4330\_Xplorer\_Blinky' and click Save.

🖉 🗢 📕 « Ipc4330_Xplorer_Keil 🕨 Blinky	✓ ♦ Search Blinky	Q
Organize 🔻 New folder	8==	• 0
Documents Music Music Pictures Videos Nagaraj Computer GOS (C:) Mex DVD RW Drive	Date modified Type No items match your search.	Siz
Local Disk (F:)     Local Disk (G:)     CD Drive (H:)     Filename: LPC4330_Xplorer_Blinky     Save as type: Project Files (*.uyproj)		•

Fig.26



Step 5: Select the NXP(found by Philips) and search for the controller.

Vondor: Actol		
Device:		
Toolset:		
Data base	Description:	
Infineon Lapis Semiconductor Luminary Micro Illington NxP (founded by Philips) LH75400 LH75401 LH75410 LH75410 LH7520 LH7524 Illington		*
	OK Cancel	Help

**Fig.27** 

Step 6: Select LPC4330 controller and click OK.



**Fig.28** 



Step 7: Click YES to copy startup file to project folder and add file to project.



**Fig.29** 

Step 8: Rename the 'Source Group1' to Startup.



Fig.30



Step 9: Double click on Startup.

Add File	es to Group 'Startup'		×
Name		Date modified	Туре

Fig.31

Step 10: Search 'system\_LPC43xx.c file' (in the sample examples downloaded folder from NGX website), the file will found at following path:

**'..\CMSISv2p10\_LPC43xx\_DriverLib\Core\Device\NXP\LPC43xx\Source\Templates'** select 'system\_LPC43xx.c file' and click Add as shown in the following image.

roject		<b>* 🔛</b>		
	arget 1 Startup 🟦 startup_LPC4	ł3xx.s		
ſ	Mdd Files to G	iroup 'Startup'	State of the second second	X
	Look in: 退 Ter	mplates	- ← 🗈 💣 🖬	
	Name		Date modified	Туре
	ARM GCC		28-05-2012 AM 10 28-05-2012 AM 10 28-05-2012 AM 10	File fo File fo File fo
	system_LPC	43xx.c	28-05-2012 AM 11	C File
	<ul> <li>✓</li> <li>File name: sy</li> <li>Files of type: C</li> </ul>	Type: C File Size: 2.10 KB Date modified: 28 stem_LPC+axc.c	-05-2012 AM 11:50	ld se

Fig.32



Step 10: Right click on Target to add a new group.



Fig.33

Step 11: Rename the 'New Group' to Drivers.



Fig.34



Step 12: Double click on Drivers and search the driver files (in the sample examples downloaded folder from NGX website) the driver files will found at following path:

**'..\CMSISv2p10\_LPC43xx\_DriverLib\src'** for blinky project we have to select lpc43xx\_cgu.c, lpc43xx\_gpio.c, lpc43xx\_scu.c, lpc43xx\_timer.c and lpc43xx\_utils.c driver files and click Add.

Look in the src	
Name	Date modified
pc43xx_cgu.c	18-01-2012 PM 06:
Ipc43xx_dac.c	18-01-2012 PM 06:
lpc43xx_emc.c	18-01-2012 PM 06:
lpc43xx_evrt.c	18-01-2012 PM 06:
lpc43xx_gpdma.c	18-01-2012 PM 06:
Ipc43xx_gpio.c	27-03-2012 PM 04:
File name: I"loc43ox utils c" "loc43o	c cau c." "loc43xx, apia c."

**Fig.35** 

Step 13: Create another new group and rename it as Main. Step 14: Click on New to create an empty document.



Fig.36



Step 15: Implement the C instructions need to blink a LED on Xplorer and save it to Main.c in 'Blinky folder' as shown in the following image.

(Note: Please refer Downloaded 'LPC4330\_Xplorer\_Blinky example')



**Fig.37** 

Step 16: Double click on 'Main Group' and select 'Main.c file' and click on Add

Look III.	Blinky	<u>_</u> ← 🗈 📸 .
Name	*	Date modified Ty
Ma	in.c	07-06-2012 PM 04: C
∢ [ File name	e: Main.c	Add

**Fig.38** 



Step 17: Click 'Target Option', make following changes for 'Internal SRAM' as shown in the following image.

Device	Target 0	utput Listing	User C/C	:++ Asm	Linker	Debug	Utilities		
NXP <mark>(</mark> fou	nded by Pl	hilips) LPC <mark>4</mark> 330	Xtal (MHz): 1	12.0	Code C	Generatior	1		
Operatin	g system:	None		•	E.	lse Cross-l	Module Op <mark>t</mark> imiza	ation	
System-	/iewer File	(.Sfr):		_		lse MicroL	IB ľ	🗌 Big Endian	
SFD\N	KP\LPC43	xx\LPC43xx.SF	R		Floati	ing Point H	Hardware:	Use FPU	
- Pood	Ophy Mom	ani Amaa			- Pood	Write Mor		Use PPU	
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	Nol
	ROM1:			- C		RAM1:			
	ROM2:			- c		RAM2:			- r
	ROM3:	Í	<u> </u>	- c		RAM3:	Ì.		
1	on-chin					on chin	-		
	IROM1:	0x10000000	0x18000	•		IRAM1:	0x10080000	0x20000	
	A DE DE AND	All and a second se	Sector Sector	0		IDAM2	0~20000000	IOx10000	

**Fig.39** 

For 'External SPIFI Flash', make following changes as shown in the following image.

XP (tound	led by Pl	hilips) LPC4330	MILLION LINE	20		eneration			
)perating :	system:	None			Γυ	se Cross-M	odule Optimiza	ition	
vstem-Vie	wer File	(.Sfr):				se MicroLII	в Г	🗌 Big Endian	
SFD\NXP	LPC43	x\LPC43xx.SFI	R		Floati	ng Point Hi	ardware:	Not Used	-
							<	Not Used	_
Read/Or	nly Memo	ory Areas			Read/	Write Mem	ory Areas	5 	
default (	off-chip	Start	Size	Startup	default	off-chip	Start	Size	Nolnit
Γ	ROM1:			C		RAM1:			
	ROM2:			- c		RAM2:			
Г	ROM3:			- c	Г	RAM3:			
_	on unip	-	-			on ohin	-		
<b>V</b> I	ROM1:	0x14000000	0x18000			IRAM1:	0x10080000	0x20000	П
	ROMZ			C		IRAM2:	UX20000000	0x10000	

Fig.40



Step 18: In 'Output Option' select 'Create HEX File' check box.

Select Folder for Objects	Name of Executable: LPC4330_Xplorer_Blinky
Create Executable: \LPC4330_Xplo     ✓ Debug Information     ✓ Create HEX File     ✓ Browse Information     ✓ Create Library: \LPC4330 Xplorer E	orer_Blinky
C Create Library: \LPC4330 Xplorer 8	Blinky I IB

Fig.41

Step 19: Click 'C/C++ Option', in Define type 'CORE\_M4', click on 'Include paths' to include drivers header file path as shown in the following image.

Prep	processor Symbols		
[	Defile: CORE_M4		
Und	define:		
- Lan	guage / Code Generation	x ) ?	ngs:
idei	Browse for Polder		pecified>
ietu	Select Folder:		numb Mode
	F:\pc4330_Xplorer_Keil_v4_53\CMSISv2p10_LPC43xx_D	river	
	CMSISv2p10_LPC43xx_DriverLib settings	<b>^</b>	
	▷ 🕌 Core		
	docs_nxp_driverlib		nc -I C:\Keil\ARM
	jinc jinc		rowse "*.crf"
	Description of the second s	+	-
		Land I and a second	Help



Fig.42

Step 20: In 'Asm Option', in Define type 'NO\_CRP' for assembly control symbols.

wice   ruig	et   Output   Listing   User   C/C+t Asm Linker   Debug   Utilities
Conditiona	Assembly Control Symbols
Define	NO_CRP
Undefine:	
Language	/ Code Generation
	Split Load and Store Multiple
Read-C	Only Position Independent
☐ Read-V	Nrite Position Independent
and the second second	
I Thumb	Mode
□ Thumb	Mode mings
□ Thumb	Mode mings
I Thumb	Mode mings
Include Paths Misc	Mode mings
Include Paths Controls	Mode mings
Include Paths Misc Controls Assembler control	Mode mings
Include Paths Misc Controls Assembler control	Mode mings

#### Fig.43

Step 21: In 'Debug Option', select 'ULINK2/ME Cortex Debugger' radio button and select 'Load Application at Startup' and 'Run to main()' check boxes, click on 'Initialization File': to select 'Internal SRAM.ini file', select 'Internal SRAM.ini file' and click Open as shown in the following image.



Fig.44



For 'External SPIFI flash', make following changes as shown in the following image.

1.000			
C++ Asm	Linker Debug Utilities		
Settings	JOJse: ULINK2/ME Cortex Debugger 💌 _	Settings	
) main()	oad Application at Startup	🔛 Select Target Debugger Initialization File	×
1	Initialization File:	🛛 💭 🗢 🖟 🖌 🖓 🕹 🕹 🕹 🕹 🕹 🕹 🕹 🕹 🕹 🕹 🕹 🕹 🕹	Q
Edit	Restore Debug Session Settings	Organize ▼ New folder 🗮 ▼ [	1 0
	Preakpoints Toolbox	Favorites Ame Date modified	Туре
r.	Watch Windows	5	INI File
	Memory Display	Desktop SPIFI 32MB Debug.ini 08-06-2012 PM 01:	INI File
	Driver DLL: Parameter: SARMCM3.DLL -MPU	☐ Libraries ☐ Documents ↓ Music	
	Dialog DLL: Parameter:	i pictures Videos Nagaraj	
	I from the second secon	👰 Computer	
Ca	ancel Defaults	🗣 Network 🔤 Control Panel 👻 K 🔤	•
	ш	File name: SPIFI 32MB Debug.ini - Ini Files (*.ini)	• ncel

Fig.45

(*Note: The 'ini file' will found at following path: ..\LPC4330\_Xplorer\_Blinky\Keil*)

Step 22: In 'Utilities Option', select 'Use External Tool for Flash Programing'.

	tput   Listing   User   C/C++   Asm   Linker   Debug Utilities
-Configure Flash M	enu Command
C Use Target D	river for Flash Programming
U	LINK2/ME Cortex Debugger 🔄 Settings 🔽 Update Target before Debugging
Init File:	Edit
<ul> <li>Use External</li> </ul>	Tool for Flash Programming
Command:	<u></u>
Arguments:	
	Run Independent

Fig.46



For 'External SPIFI Flash', select 'Use Target Driver for Flash Programing' as 'ULINK2/ME Cortex Debugger' and select 'Update Target before Debugging' check box then click 'Setting Option', remove the existing programing algorithm and click Add select 'LPC18xx/43xx S25FL032 SPIFI Flash algorithm' as shown in the following image.

<ul> <li>Se Target Driver for I</li> </ul>	Flash Programming		
ULINK2/M	IE Cortex Debugger 💽 Settings	date Target before Debugging	
Init File:	Cortex-M Target Driver Setup		
C Use External Tool for	Debug Trace Flash Download		
Command:	□ Download Function	RAM for Algorithm	
Arguments:	LOAD C Erase Full Chip 🔽 Progra	im l	
E Duele	Erase Sectors 🔽 Verify	Start: 0x10000000 Size	0x8000
I HUITH	Uo not Erase   Reset	and Run	
	Programming Algorithm	Add Hash Programming Algori	thm
	Description Device	Type	Device Type Device Size
		HT32 Series Flash	On-chip Flash 128k
		HT32 Series Flash Options	On-chip Flash 4k
		K8P5615UQA Dual Flash	Ext. Flash 32-bit 64M
		LM3Sxxx 128kB Flash	On-chip Flash 128k
		LM3Sxxx 16kB Flash	On-chip Flash 16k
		LM3Sxxx 256kB Flash	On-chip Flash 256k
		LM3Sxxx 32kB Flash	On-chip Flash 32k
		LM3Sxxx 384kB Flash	On-chip Flash 384k
		LM3Sxxx 512kB Flash	On-chip Flash 512k
11	( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )	Add LM3Sxxx 64kB Flash	On-chip Flash 64k
		LM3Sxxx 8kB Flash	On-chip Flash 8k
0. Te		LM4Fxxx 128kB Flash	On-chip Flash 128k
to a survey of the survey of the	×	LM4Fxxx 256kB Flash	On-chip Flash 256k
		OK LM4Fxxx 32kB Flash	On-chip Flash 32k
		LIMAGener CALL D. F.	C/1.

**Fig.47** 

Step 23: In 'Debug Option', click on Edit... the 'Internal SRAM.ini file' will open in editor and OK.

levice   Target	Output   Listing   User   C/C++   Asm	Linker Debug	Utilities	
C Use Simulator Settings ☐ Limit Speed to Real-Time		Use: ULINK	2/ME Cortex Debugger 💌 Settings	
Load Applicat	ion at Startup 🔲 Run to main()	✓ Load Applicat Initialization File:	tion at Startup 🔽 Run to main()	
	Edit		Ninternal SRAM.ini	
- Restore Debug Breakpoin Watch W Memory D	Session Settings ts 🔽 Toolbox indows & Performance Analyzer lisplay	Restore Debug Breakpoin Watch Wi Memory D	Session Settings ts 🔽 Toolbox indows isplay	
CPU DLL:	Parameter:	Driver DLL:	Parameter:	
SARMCM3.DLL	-MPU	SARMCM3.DLL	-MPU	
Dialog DLL:	Parameter:	Dialog DLL:	Parameter:	
DCM.DLL	pCM4	TCM.DLL	pCM4	

**Fig.48** 



For 'External SPIFI Flash', click on Edit... the 'SPIFI 32MB Debug.ini file' will open in editor and OK.

C Use Simulator Settings				
Limit Speed to	o Real-Time	1		
Load Application at Startup Initialization File:		I Load Application at Startup I Run to main() Initialization File:		
	Edit	SPIFI 32MB D	ebug.ini Edit)	
Restore Debug Session Settings		Restore Debug	Restore Debug Session Settings	
🔽 Breakpoints 🔽 Toolbox		I Breakpoin	🔽 Breakpoints 🔽 Toolbox	
🔽 Watch W	indows & Performance Analyzer	🔽 Watch W	🔽 Watch Windows	
✓ Memory Display		Memory Display		
CPU DLL:	Parameter:	Driver DLL:	Parameter:	
SARMCM3.DLL	-MPU	SARMCM3.DLL	-MPU	
Dialog DLL:	Parameter:	Dialog DLL:	Parameter:	
DCM.DLL	-pCM4	TCM.DLL	-pCM4	

Fig.49

Step 24: Give the correct '.axf file' path in 'Internal SRAM.ini file' and file name should be same as shown in the following image.

1		
2 FUNC void Setup (unsigned	int region) {	
3 region &= 0xFFFF0000;		// grant Starly Determined
4 SP = RDWORD (region);	125	// Setup Stack Pointer
5 PC RDWORD (region + 4		// Setup Program Counter
6WDWORD(OXEODOED08, reg	lon);	// Secup vector lable offset Registe
/ _ /		
9 LOAD T.PC4330 Xplorer Bli	NO ANT INCREMENTAL	
10	in the second second	
11	Options for Target 'Target 1'	
12 Setup( scatterload);	0	
13	Device Target Output histing	User C/C++ Asm Linker Debug Utilities
	Ó Ó	
	Select Folder for Objects	Name of Executable (LPC4330 Xplorer Blinky
	Create Executable: .\LPC	4330_Xplorer_Blinky
	Debug Information	
	Create HEX File	

Fig.50



For 'External SPIFI Flash', Give the correct '.axf file' path in 'SPIFI 32MB Debug.ini file' and file name should be same as shown in the following image.



**Fig.51** 

Step 25: Click on 'Build (F7)' to build a blinky project, the build should be error free.



**Fig.52** 



Step 26: Click on Debug -> 'Start/Stop Debug Session'.

File Edit View Project Flash	ebug Peripherals Tools SV(	CS Window	Help
🖹 🖸 🍘 🗶 🖉 🖉 🖉 🖉	Start/Stop Debug Session	Ctrl+P	🗐 //👷 🖄
😵 🖭 🖽 🥔 畏 🙀 🛛 Targe 😫	T Reset CPU	The description of the second	
Project 🛛 🛱 📓	Run	F5	
🖃 🔚 Target 1	) Stop		
🔁 🦰 Startup	} Step	F11	*** PRIVAT
a stattup_LPC43xx.c 0	Step Over	F10	//LEDUSB
Drivers	Step Out	Ctrl+F11	//TEDUCE
ian ∰ 🚰 🛗 👘	} Run to Cursor Line	Ctrl+F10	// LEDOSE
\$	Show Next Statement		
	Breakpoints	Ctrl+B	*** PRIVAT
	Insert/Remove Breakpoint	F9	



Step 27: Click on 'Run (F5)', on Xplorer board the LED starts blinking.



Fig.54



#### 3.2 Executing the sample project in KEIL

Please note that the sample programs are available once the product is registered.

Steps to execute the sample project in 'Internal SRAM':

Step 1: Open project folder.

Step 2: Open project\_name.uvproj file (Example: LPC4330\_Xplorer\_Blinky.uvproj.)



Fig.55

Step 3: This launches the IDE and double click on 'Main.c file', click on build, build must error free. Click on Debug -> 'Start/Stop Debug Session'.



**Fig.56** 



Step 4: Click Run (F5) to execute from the 'Internal SRAM', two LED's (D2 and D3) on Xplorer should blink.

Steps to execute the sample project in 'External Quad Flash' (SPIFI 32MB Debug):

Step 1: Select 'SPIFI 32MB Debug Option' and click on build as shown in the following image.



**Fig.57** 

Step 2: The program can be debugged from the flash by clicking Debug -> 'Start/Stop Debug Session', click Run (F5) to execute from the 'External Quad Flash' OR click on LOAD, the executable is loaded into 'SPIFI 32MB flash' then press RESET switch twice to run program from 'External Quad Flash', the two LED's (D2 and D3) should start blinking on Xplorer.



Fig.58



# 4.0 Restoring Xplorer to Factory Defaults

## 4.1 ULINK2/ME and KEIL

To restore the factory defaults for the Xplorer, user needs the ULINK2/ME debugger to program the Xplorer to default firmware (i.e. **LPC4330\_Xplorer\_PeripheralTest.uvproj**).

Steps to restore the factory defaults for Xplorer:

Step 1: Open LPC4330\_Xplorer\_Default\_Setting folder and double click on LPC4330\_Xplorer\_PeripheralTest.uvproj project.

Coord Web Contract Co	e 🕨 LPC4330_Xplorer_Default_Setting 👻 🍫	Search LPC4330 🔎
Organize 🔻 🔣 Open 👻 Burn	New folder	:= • 🚺 🔞
🔆 Favorites	Name LPC4330_Xplorer_PeripheralTest.axf	1
🧮 Desktop	E LPC4220_Xpt.rc_PeripheralTecteropt	1
🥃 Libraries	LPC4330_Xplorer_PeripheralTest.uvproj	
Documents		
👌 Music		
Pictures		
🛃 Videos		
🎉 Nagaraj		
💻 Computer		
👊 Network		
🥶 Control Panel		
🗑 Recycle Bin		
🐌 Desktop		•
LPC4330_Xplorer_Peripheral	Test.uvproj Date modified: 11-04-2012 PM 07:07 Size: 27.9 KB	

**Fig.59** 

Step 2: Click on LOAD, the **LPC4330\_Xplorer\_PeripheralTest.axf** will flash on to 'External Quad Flash'. Now, RESET twice to restore the Xplorer to factory default.

🔽 Fr.Xplorer_release/LPC4330,Xplorer_Examples_VTLPC4330,Xplorer_Default_Setting/LPC4330,Xplorer_PeripheralTest.uvproj – µVision4	
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help	
□ 20 より入って(4-5)を充充方(定定)//// 20 GRO_Cleadnt - しょうの() - ○ クターー・ヘ	
S C T T C C C C C C C C C C C C C C C C	
Project 9 III	
🔁 SPIFI 32MB Debug	
Buscher	2 🖬
Load "F:\\Xplorer_release\\LPC4330_Xplorer_Examples_V1\\LPC4330_Xplorer_Default_Setting\\LPC4330_Xplorer_FeripheralTest.x47"	*
Erase Done.	
Programming Bourd	
	Ψ.
4	b

**Fig.60** 



# 5.0 Schematic & Board Layout

## **5.1 Schematic**

This manual will be periodically updated, but for the latest documentations please check our <u>website</u> for the latest documents. The Board schematic and sample code are available after the product has been registered on our website.

## 5.2 Board layout



SILKSCREEN TOP SIDE

Fig.61



Fig. 62



# 6.0 CHANGE HISTORY

# 6.1 Change History

Rev	Changes	Date (dd/mm/yy)	Ву
1.0	Initial release of the manual	29/03/2012	Ashwin Athani
1.1	• Added section for Restoring Xplorer to factory defaults in KEIL	10/04/2012	Nagaraj Baddi
1.2	<ul> <li>The Keil User Manual separated.</li> <li>Creating the sample Blinky project in KEIL added</li> <li>Setup for ULINK2/ME and LPC4330 Xplorer added.</li> </ul>	08/06/2012	Nagaraj Baddi



#### About this document:

#### **Revision History**

Version: V1.2 author: Nagaraj Baddi

#### **Company Terms & Conditions**

#### Legal

NGX Technologies Pvt. Ltd. provides the enclosed product(s) under the following conditions:

This evaluation board/kit is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION, and EDUCATION OR EVALUATION PURPOSES ONLY and is not considered by NGX Technologies Pvt. Ltd to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL and therefore may not meet the technical requirements of these directives or other related directives.

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies NGX Technologies from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

NGX Technologies currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. NGX Technologies assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages.

No license is granted under any patent right or other intellectual property right of NGX Technologies covering or relating to any machine, process, or combination in which such NGX Technologies products or services might be or are used.



#### Disclaimers

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

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

#### Trademarks

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