

# LPC1768-Xplorer

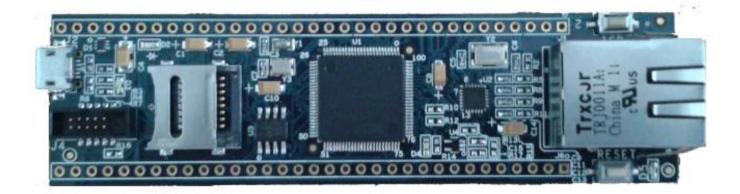


Fig.1



#### **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



# **Table of Contents**

1.0 INTRODUCTION	4
2.0 LPC1768-Xplorer Development Tool Setup	5
2.1 IDE and debugger	
2.2 Installation & Configuration of KEIL software	
2.3 Setup for ULINK2 and LPC1768-Xplorer Board	
2.4 Configuration of ULINK2 Debugger	
2.4 Configuring Internal Flash	
3.0 LPC1768-Xplorer Software Development	
3.1 Executing the sample projects in KEIL	
3.2 Creating New project	
4.0 LPC1768-Xplorer Programming	
4.1 Programming options	
4.1.1 Secondary bootloader (USB)	20
4.1.2 Flashing the board using USB	
4.1.3 Creating the Bin file compatible with Secondary USB bootloader	
4.0 Schematic & Board Layout	
4.1 Schematic	23
4.2 Board layout	
5.0 CHANGE HISTORY	
5.1 Change History	



# **1.0 INTRODUCTION**

This document is the User Manual for the LPC-1768-Xplorer, a low cost ARM Cortex-M3 based board by NGX Technologies. 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 quick start guide for LPC-1768-Xplorer features and hardware verification.

For LPC1768-Xplorer Quick Start Guide: Click here.

For the most updated information on the LPC-1768-Xplorer board please refer to NGX'website.



# 2.0 LPC1768-Xplorer Development Tool Setup

#### 2.1 IDE and debugger

As mentioned in the earlier section, 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 as the IDE and debugger respectively.

### 2.2 Installation & Configuration of KEIL software

The Installation of KEIL software is explained below:

Step 1: Open the KEIL setup

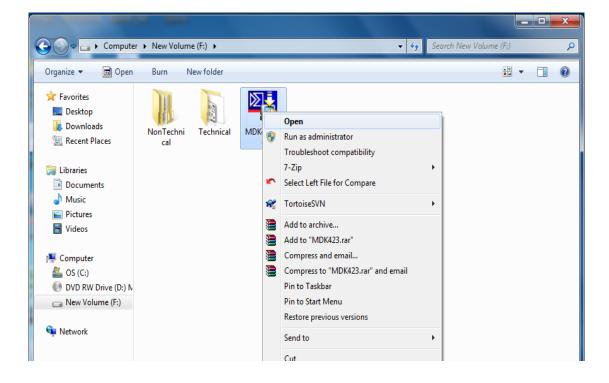


Fig.2



Step 2: Click on Next.

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

Fig.3

Step 3: Select Terms & conditions check box and click on 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 +
<ul> <li>✓ If agree to all the terms of the preceding License Agreement</li> <li>✓ Keil µVision4 Setup</li> </ul>
<< Back Next >> Cancel

Fig.4



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 differ folder. Destination 5.100. C:Weil	erent folder, press 'Browse' and select another B <u>r</u> owse
— Keil µVision4 Setup ————————————————————————————————————	<< Back (Next >> ) Cancel

Fig.5

Step 5: Fill your Personal information and Click on Next

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

Fig.6



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	
— Keil μVision4 Setup	<< Back Next>> Cancel

Fig.7

Step 7: Keil  $\mu$ Vision4.23 setup is completed. Click on Finish.

Setup MDK-ARM V4.23	<b>— X</b> —
Keil μVision4 Setup completed MDK-ARM V4.23	
μVision Setup has performed all requested operations successfu Launch Driver Installation: "ULINK Pro Driver V1.0"	lly.
✓ Show Release Notes.	
— Keil μVision4 Setup	<< Back Finish Cancel

Fig.8



## 2.3 Setup for ULINK2 and LPC1768-Xplorer Board

The Xplorer board has on board '10-pin SWD/JTAG box'; ensure that the ULINK2/ME must havpin SWD/JTAG' support for development. The ULINK2 debugger, '20-pin to 10-pin adapter' and pin ribbon cable are not a part of the LPC1768 Xplorer package, the user needs to buy separately. To program/debug the KEIL examples you will need the following and the image shows the components:

- ULINK2
- ARM JTAG to Cortex JTAG Adapter (20-pin to 10-pin Adaptor)
- 10-pin ribbon cable
- LPC1768 Xplorer Board
- One USB AM to Micro B cable



Fig.9

Steps to setup the ULINK2 and LPC1768 Xplorer Board: (*Note: Please refer keil knowledgebase article 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.10



Step 2: Connect other end of 10-pin ribbon cable to '10-pin box header' of the LPC1768 Xplorer board as shown in the following image.

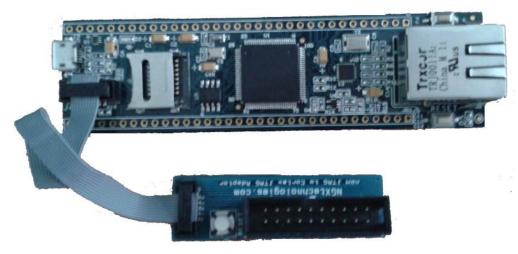


Fig.11

Step 3: Now connect the 'ULINK2 20-pin cable' to '20-pin to 10-pin adapter', connect one end of 'USB AM to Micro B' cable to LPC1768 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.12** 

Step 4: The setup is now ready to be used for development with KEIL IDE and ULINK2.



### 2.4 Configuration of ULINK2 Debugger

The configuration flow of ULINK2 Debugger is explained below:

Step 1: Open the Keil Workspace, then by clicking on the **target** option, the window opens as shown below. Next click on Debug and then select the ULINK2/ME Cortex Debugger as shown in the image.

Voptions for Target 'FLASH'	<b>X</b>	
Device   Target   Output   Listing   User   C/C++   Asm	Linker Debug Utilities	
C Use Simulator Settings Limit Speed to Real-Time	Use: ULINK2/ME Cortex Debugger	
Load Application at Startup Run to main() Initialization File: Edit	Load Application at Startup     Run to main() Initialization File:      Edit	
Restore Debug Session Settings         Image: Contract of the section		
I Memory Display	Memory Display	
CPU DLL: Parameter: SARMCM3.DLL MPU	Driver DLL: Parameter: SARMCM3.DLL -MPU	
Dialog DLL: Parameter: DARMP1.DLL pLPC1768	Dialog DLL: Parameter: TARMP1.DLL pLPC1768	
OK Cancel Defaults Help		

Fig.13

Step 2: Next Click on the settings option, the Cortex-M Target Driver Setup window opens then select JTAG port. After selection of the JTAG port the ULINK2 detected is as shown in the image below

Device       Target       Output       Listing       User       C/C++       Asm       Linker       Debug       Utilities         C       Use Simulator       Settings       Image: ULINK2/ME Cortex Debugger       Settings       Settings         Cortex-M Target Driver Setup       Image: ULINK2/ME Cortex Debugger       Settings       Image: ULINK2/ME Cortex Debugger       Settings         Debug       Trace       Rash Download       Image: ULINK USB - JTAG/SW Adapter       JTAG Device Chain       Image: ULINK Version: ULINK2         Serial No:       V0168AVR       Image: UDCODE       Device Name: Image: Image: UPCODE       Image: Image: Image: UPCODE       Image: Imag	Coptions for Target 'FLASH'
Cortex-M Target Driver Setup       X         Debug       Trace       Rash Download         ULINK USB - JTAG/SW Adapter       JTAG Device Chain         Serial No:       V0168AVR       Image: Cortex-Marget Driver Setup         ULINK Version:       ULINK2         Device Family:       Cortex-M         Firmware Version:       V1.42         If SWJ Port       JTAG         Max Clock:       1MHz         Add       Delete         Update       IR len:         Debug       Connect & Reset Options         Connect:       Normal         Reset:       Autodetect	Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Debug       Trace       Rash Download         ULINK USB - JTAG/SW Adapter       JTAG Device Chain         Serial No:       [V0168AVR]         ULINK Version:       [ULINK2]         Device Family:       Contex-M         Firmware Version:       [V1.42]         If SWJ Por       JTAG         Max Clock:       1MHz         Add       Delete         Update       IR len:         Debug       Connect & Reset Options         Connect:       Normal         Reset:       Autodetect	C Use Simulator Settings © Use: ULINK2/ME Cortex Debugger 🔍 Settings
ULINK USB - JTAG/SW Adapter       JTAG Device Chain         Serial No: V0168AVR       IDCODE         ULINK Version: ULINK2       IDCODE         Device Family: Cortex-M       IDCODE         Firmware Version: V1.42       IV         IV       SWJ         Max Clock: IMHz       Add         Debug       Connect & Reset Options         Connect & Reset Options       Cache Options         Download Options       Verify Code Download	Cortex-M Target Driver Setup
Connect & Reset Options Connect: Normal Reset: Autodetect Connect: Normal Verify Code Download	ULINK USB - JTAG/SW Adapter Serial No: V0168AVR  ULINK Version: ULINK2 Device Family: Cortex-M Firmware Version: V1.42 SWJ Port JTAG
OK Cancel Help	Connect & Reset Options Connect: Normal  Reset: Autodetect Reset after Connect Stop after Bootloader Cache Options Download Options Verify Code Download Download to Rash

Fig.14

Note: The Cortex M3 can be programed using SW or JTAG.



## 2.4 Configuring Internal Flash

Step 1: Open the Keil Workspace, then by clicking on the **target** option the widow opens as shown below, click on Utilities and select ULINK2/ME Cortex Debugger

V Options for Target 'FLASH'	x
Device   Target   Output   Listing   User   C/C++   Asm   Linker   Debug Utilities	
Configure Flash Menu Command	
Use Target Driver for Flash Programming	
ULINK2/ME Cortex Debugger Settings ✓ Update Target before Debugging	
Init File: Edit	
C Use External Tool for Rash Programming	
Command:	
Arguments:	
Run Independent	
OK Cancel Defaults Help	

**Fig.15** 

Step 2: Now Click on Settings, then in Cortex-M Target Driver setup click add and select FLASH on LPC17xx Flash programing algorithm and click add.

Options for Target 'FLASH'			×	
Device Target Output Listing User C/C++ Asm Configure Flash Menu Command © Use Target Driver for Flash Programming ULINK2/ME Cortex Debugger	Settings	before Debugging		
Cortex-M Target Driver Setup	Add Flash Programming Algorit	ih ma	23	x
Debug Trace Flash Download	Add Flash Programming Algori	nm		
Download Function         Image: Constraint of the sectors         Image: Constraint o	Description           LPC122xIAP 64kB Flash           LPC11xx/13xx IAP 8kB Flash           LPC12xIAP 80kB Flash           LPC12x IAP 96kB Flash           LPC17xx IAP 128kB Flash           LPC17xx IAP 256kB Flash           LPC17xx IAP 256kB Flash           LPC17xx IAP 512kB Flash           LPC17xx IAP 512kB Flash           LPC17xx IAP 512kB Flash           LPC17xx IAP 512kB Flash           MB9kF500 256kB Flash           MB9kF502 128kB Flash           MB9kFxx1 248kB Flash           MB9kFxx2 128kB Flash           MB9kFxx5 384kB Flash           MB9kFxx5 384kB Flash           MB9kFxx5 384kB Flash           MB9kFxx5 384kB Flash	Device Type On-chip Flash On-chip Flash	Device Size 64k 80k 96k 256k 32k 512k 64k 256k 64k 256k 64k 128k 256k 384k 512k	
	Cancel	Cancel	Help	

Fig.16

Click OK to complete the ULINK2 Debugger configuration.



# 3.0 LPC1768-Xplorer Software Development

## 3.1 Executing the sample projects in KEIL

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

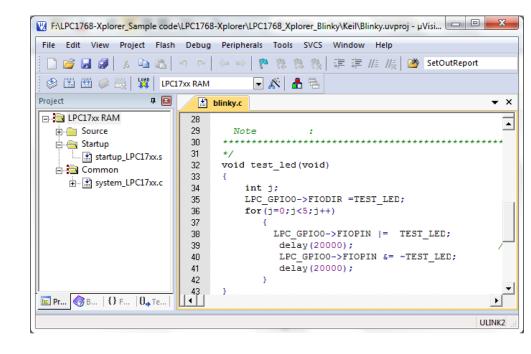
Steps to execute the sample project:

Step 1: Open the project folder.

Step 2: Then open the file project\_name.uvproj (Example: Blinky.uvproj.)

Sector 2 → 2 → 2 → 2 → 2 → 2 → 2 → 2 → 2 → 2	Xplorer_Blinky 🕨 Keil 🕨 🔻 😽 Sea	rch Keil 🔎
Organize 👻 🔣 Open	▼ Burn New folder	i 🕶 🖬 🔞
☆ Favorites	Name	Date modified Type
📃 Desktop	🐌 Lst	19-04-2012 18:34 File folder
📕 Downloads 🗮	퉬 Оbj	20-04-2012 12:24 File folder
🔚 Recent Places	Blinky.uvgui.NGX12	20-04-2012 12:44 NGX12 File
	Blinky.uvgui_NGX12.bak	20-04-2012 12:24 BAK File
🔚 Libraries	Blinky.uvopt	20-04-2012 12:44 UVOPT File
Documents	Blinky	19-04-2012 19:13 µVision4 Project
J Music	Blinky_LPC17xx Flash.dep	20-04-2012 12:44 DEP File
Pictures	Blinky_uvopt.bak	20-04-2012 12:24 BAK File
Subversion	Blinky_uvproj.bak	30-12-2011 10:50 BAK File
Videos		
		4
Blinky µVision4 Project		reated: 19-04-2012 18:30

**Fig.17** 



Step 3: This launches the IDE

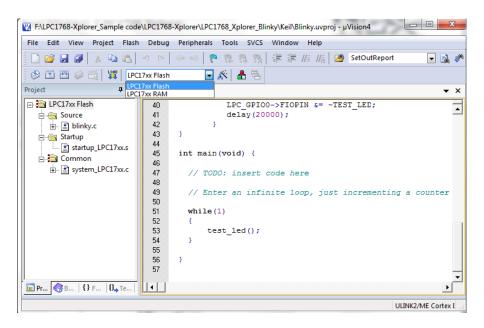
Fig.18



If you are using ulink2 debugger, click Debug on menu bar then click on start/stop debug session; click Run (F5) to execute from the Internal RAM, LED's (D4) on Xplorer should blink.

Steps to execute the sample project in Internal Flash:

Step 4: Select LPC17xx Flash option as shown in the below image.



**Fig.19** 

Step 5: Click on Build to build the project as shown in the below image

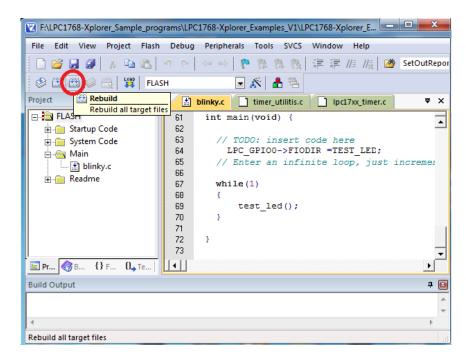


Fig. 20



Step 6: Click on LOAD, the executable is loaded into internal flash. The program can be run by pressing RESET switch or the program can be debugged from the flash by clicking start/stop debug session. Click Run (F5) to execute from the Flash, LED (D4) on Xplorer should blink.

F:\LPC1768-Xplorer_Sample code	e\LPC1768-Xplorer\LPC1768_Xplorer_Blinky\Keil\Blinky.uvproj - μVisi	x
File Edit View Project Flash	n Debug Peripherals Tools SVCS Window Help	
🗋 🗋 🛃 🕼 👗 🛅 🛍	- 9 (2) (4>) 🥐 🎘 🤼 🎘 🐺 🏣 🎼 //絵   20 SetOutReport	
	117xx Flash 💽 🔊 📥 🗟	
Project 🛛 🗜 📧	blinky.c	▼ ×
Pr Pr	<pre>22 23 Description : 24 25 Input : 26 27 Output : None 28 30 ************************************</pre>	•
	ULI	NK2 🔡

**Fig.21** 

Step 7: To debug the code click on Debug option then click on Start/Stop Debug session as shown in the below image. Press F5 to free run or press F10 to line by line debug.

F:\LPC1768-Xplorer_Sample_prog	rams\	LPC1768-Xplorer_Examples_\	/1\LPC1768-Xpl	orer_E 😐 😐 🗙
File Edit View Project Flash	Deb	ug Peripherals Tools S	/CS Window	Help
🗋 🖸 🖬 🕼 🕷 🕹 🛍	Q	Start/Stop Debug Session	Ctrl+F5	🗉 //👷 1 🏙 SetOutRepo
🛛 🕸 🕮 🧼 🔜 🙀 🛛 FLAS	RST	Reset CPU		
Project 🛛 🗜 🗵	H.	Run	F5	'xx_timer.c ₹ ×
E FLASH	0	Stop		******
i fire fire fire fire fire fire fire fir	{*}	Step	F11	
i System Code	9	Step Over	F10	
blinky.c	{}-	Step Out	Ctrl+F11	
	7{}	Run to Cursor Line	Ctrl+F10	
	\$	Show Next Statement		L - TROT LED.
		Breakpoints	CtrI+B	N  = TEST_LED;
	٠	Insert/Remove Breakpoint	F9	N &= ~TEST_LED;
	0	Enable/Disable Breakpoint	Ctrl+F9	
E Pr 🕜 B {} F 🛛 Te	8	Disable All Breakpoints		
Build Output	<b>.</b>	Kill All Breakpoints	Ctrl+Shift+F9	
		OS Support	Þ	÷ •
		Execution Profiling	Þ	-
<		MamaniMan		- F
Enter or leave a debug session		Memory Map		the second se

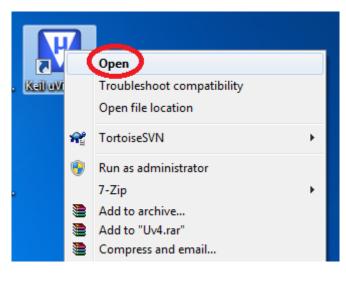
Fig. 22



## **3.2 Creating New project**

Follow the below steps, for creating new project:

Step 1: Open the keil IDE.





Step 2: Click on to the Project tab - new uvision project.

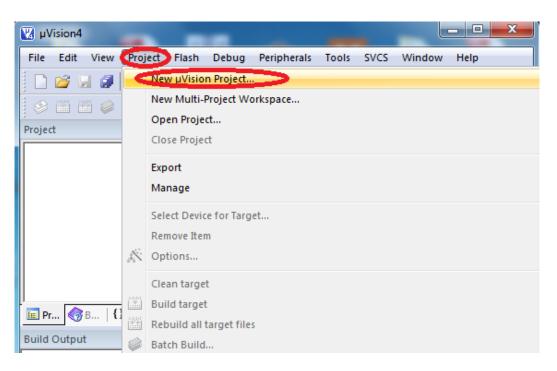


Fig. 24



Step 3: Give project name then click Save.

Create New Project		-				×
🚱 🔾 🗢 📔 🕨 New folder		•	<b>4</b> 9	Search New fol	der	٩
Organize 🔻 New folder						0
☆ Favorites	<b>^</b>	Name		^		Date
📃 Desktop	=		No i	r search.		
Downloads	-		,,			
🕮 Recent Places						
🧊 Libraries						
🤣 Homegroup						
🖳 Computer						
🚈 OS (C:)	-	•				۰.
File name: blinky						•
Save as type: Project Files (*.uvproj)						•
) Hide Folders				Save	Canc	el

Fig. 25

Step 4: Select the controller.

Select Device for Target 'Target 1'	and the same planet	x
CPU Vendor: NXP (founded by Philips) Device: LPC1768 Toolset: ARM		
Data base	Description:	
	ARM 32-bit Cortex-M3 Microcontroller with MPU, CPU clock up to 100MH 512kB on-chip Flash ROM with enhanced Flash Memory Accelerator, In-System Programming (ISP) and In-Application Programming (IAP), 64kB RAM, Nested Vectored Interrupt Controller, Eight channel General purpose DMA controller, AHB Matrix, APB, Ethemet 10/100 MAC with RMII interface and dedicated DMA, USB 2.0 full-speed Device controller and Host/OTG controller with DMA, CAN 2.0B with two channels, Four UARTs, one with full Modem interface Three I2C serial interfaces, Three SPI/SSP serial interfaces, I2S interface General purpose I/O pins, 12-bit ADC with 8 channels, 10-bit DAC, Four 32-bit Timers with capture/compare, Standard PWM Timer block, Motor control PWM for three-phase Motor control, Quadrature Encoder, Watchdog Timer, Repetitive Interrupt Timer, Brown-out detect circuit, Power-On Reset, Power Management Unit, Wakeup Interrupt Controller, Crystal oscillator, 4MHz internal RC oscillator, PLL,	. =
	Cancel H	elp

Fig. 26



Step 5: Go to file – new, & start writing the code.

V C	:\Users\N0	GX12\	Desktop	New fol	der\blink	y.uvproj - µVis	ion4			- 0	X	<u> </u>
File	Edit V	'iew	Project	Flash	Debug	Peripherals	Tools	SVCS	Window	Help		
	New			Ctrl+I	N (2)	$\leftarrow \Rightarrow   p$	14 1	4 12		//ミ //袁	2	Se
6	Open			Ctrl+0			<u> </u>					
	Close											
	Save			Ctrl+	S							
	Save As											
Ø	Save All											
	Device Da	tabas	e									
	License M	lanag	ement									
	Print Setu	ID										
a	Print			Ctrl+	Р							
	Print Prev	iew										
	d b Balas				- 10							
	1 blinky										д	
	2 blinky											-
	3 memory											
	4 F:\Blue			4\\sd								
	5 LPC11U										Þ	-
2	6 BB-LPC1		-									
L	7 glcd_pla	attorn	n_ipc1102	4								

Fig. 27

Step 6: Save the file with some name.

🕅 Save As	COMPANY AND	-	×
New folder		✓ 4 Search New for	lder 🔎
Organize 🔻 New folder			= • 🔞
🙀 Favorites	A Nam	ne	Date
🧮 Desktop		blinky.plg	25-0
\rm Downloads		blinky.uvgui.NGX12	21-0
💯 Recent Places		blinky.uvopt	21-0
	<b>W</b>	blinky	25-0
词 Libraries		blinky_Target 1.dep	25-0
🤣 Homegroup	asm	startup_LPC17xx	17-1
🖳 Computer			
🚨 OS (C:)	<b>▼</b>	III	F
File names, blinky.c			•
Save as type: All Files (*.*)			•
🔿 Hide Folders		Save	Cancel

Fig. 28



Step 7: Add the file to the source group as shown in the below image.

C:\Users\NGX12\De	esktop∖New folder∖blinky.uvproj - µVision4	
File Edit View P	roject Flash Debug Peripherals Tools SVCS V	Vindow Help
🕴 🗋 😂 🖬 🖉 🛛	2 山間 ク () ← →   10 円 間 () 2	🗏 🎼 //👷 🛛 🖄 SetOutReport
🔅 🕮 🖉 븛	🛛 🙀 Target 1 🕞 🔊 📥 🖷	
Project	7 🖸 🗋 blinky.c	<b>▼</b> ×
⊡ - 🔁 Target 1	1	
E Sour	Options for Group 'Source Group 1' Alt+F7	
	Open File	
	Open List File	
	Open Map File	
***	Rebuild all target files	
· · · · · · · · · · · · · · · · · · ·	Build target F7	-
<mark>шр </mark>	Translate File	<u> </u>
Build Output	Stop build	P 🔝
	Add Group	*
	Add Files to Group 'Source Group 1' Remove Group 'Source Group 1' and its Files	
<u> </u>	Manage Components	Ψ
Add Files to current	Show Include File Dependencies	4
Add thes to current as	Jee oroup	ht.

Fig. 29

Step 8: Select the file and click on Add as shown in the below image.

K Add Files to Group 'Source Group 1'	×
Look in: 🚺 New folder 💌	← 🗈 📸 📰 ▾
Name	Date modified Ty
blinky	20-06-2012 11:18 C
•	4
File name: blinky	Add
Files of type: C Source file (*.c)	▼ Close

Fig. 30

Hence building the code will create the Hex file.

Step 9: To build, download and debug follow the steps 5, 6 and 7 in section 3.1.



# 4.0 LPC1768-Xplorer Programming

#### 4.1 Programming options

LPC1768-Xplorer can be programmed using the

- Secondary USB bootloader (USB)
- Debugger (ULINK2)

#### 4.1.1 Secondary bootloader (USB)

In order to program the board through USB we need to get the board under programming mode.

Getting the board in programming mode:

Theory: The secondary bootloader looks for a logic LOW to be present on a pre-defined PIN (ISP pin) during reset. If the ISP pin is held LOW and reset signal is provided to the MCU, the MCU enters into programming mode.

Practical:

On the LPC1768-Xplorer the RESET and ISP signals are connected to buttons provided on the board. Look for the RESET and ISP marking on the board. Therefore to enter into programming mode:

- A **Press and hold** the ISP button
- ▲ Press the RESET button and release it
- ▲ Now release the ISP button
- ▲ The board is in the programming mode

Programming through USB is the most convenient way to program the LPC1768-Xplorer. The USB bootloader makes programming the board very simple. You don't require any PC application to program using USB bootloader. Once the board enters the programming mode it appears as a drive on your Windows machine and all you need to do is just drag-n-drop your binary to this drive.

#### 4.1.2 Flashing the board using USB

The pre-build binaries can be used to flashing onto to the board for each peripheral by using the USB bootloader as a Mass storage device. After loading the USB bootloader firmware to the LPC1768-Xplorer board, Press ISP, then RESET; release RESET, then ISP, the mass storage device will appear on your screen. On the board LED D3 glows. Remove firmware.bin file and then place your bin file and then press reset switch to execute the specific code written on to the flash.



Fig. 31



Flashing the bin file(Drag & drop).

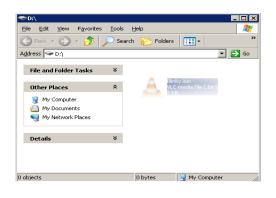


Fig. 32

#### 4.1.3 Creating the Bin file compatible with Secondary USB bootloader

Steps to create bin file that is compatible with the free loader secondary USB bootloader are as follows

Step 1: Open the project, open the Target options..

	\LPC1768-Xplorer_Keil_Sample programs\LPC1768-Xpl
🗋 😂 🛃 🍠 👗 🛍 🛍 🔊	○   ← →   作 換 換 換 律 導 //: //( 20 SetOutReport
Project 4 State	blinky.c
Image: Plast pla	<pre>2 2 3 // TODO; insert code here 4 LPC_GPIO0-&gt;FIODIR =TEST_LED; 5 // Enter an infinite loop, just incrementi 6 7 while(1) 8 { 9 test_led(); 0 } 1 2 }</pre>
<b>E Pr ③</b> B   <b>3</b> F   <b>0</b> →Te   <b>1</b>	
Build Output  Configure target options	

Fig. 33

Step 2: Change the start address of on-chip IROM1 memory from 0 to 0x2000 as shown in fig. below. This is the area where user program is placed and has to run when the controller is reset.



Options		et 'FLASH'	ograms(Er CI		r_rten_ou	mpic pre	ograms\LPC17	00 Xpi	
Device Target) Output   Listing   User   C/C++   Asm   Linker   Debug   Utilities									
NXP (founded by Philips) LPC1768 Xtal (MHz): 12.0									
Operating	system:	None		-	🗆 U	se Cross-I	Module Optimiza	tion	
	îewer File			_	<b>▼</b> U	se MicroL	IВ Г	Big Endian	
1									
Read/	Only Memo	ny Areas			Read/	Write Men	nory Areas		
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	NoInit
				0		RAM1:			
	ROM1:								
	ROM1: ROM2:		Í	0		RAM2:			
						RAM2: RAM3:			
	ROM2:			- ~ I	· ·				_
	ROM2: ROM3:	0x2000	Qx80000	- ~ I	· ·	RAM3:	Qx10000000	0x8000	_
	ROM2: ROM3: on-chip	0x2000	0x80000	c		RAM3: on-chip	0x10000000 0x2007C000	0x8000 0x8000	

Fig. 34

Step 3: Insert the user command in Run #1 ticking the check box. After compile and build the entire project, from elf it creates hex file along with the hex file there will be another file with the extension .axf. This user command is used to convert .axf file to .bin. Here in fig. Above project\_name.axf is converted to project\_name.bin which is used for drag drop programming with the USB boot-loader that allows you to enumerate the board as a Mass Storage Device.

👿 F:\LPC1768-Xplorer_Sample_programs\LPC1768-Xplorer_Keil_Sample programs\LPC1768-Xpl.	. 🗖		х
Options for Target 'FLASH'			x
Device   Target   Output   Listing User   C/C++   Asm   Linker   Debug   Utilities			
- Run User Programs Before Compilation of a C/C++ File			
□ Run #1:	🗖	DOS16	3
□ Run #2:	🗆	DOS16	5
Run User Programs Before Build/Rebuild			
□ Run #1:	🗖	DOS16	3
E Run #2:		DOS16	5
Run User Programs After Build/Rebuild			
I I Run #1: fromelf -bin -o "\$L@L.bin" "\$L@L.axf"	🗖	DOS16	3
□ Run #2:		DOS16	5
✓ Beep When Complete ☐ Start Debugging			
OK Cancel Defaults		Help	

Fig. 35



# 4.0 Schematic & Board Layout

#### 4.1 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.

## 4.2 Board layout

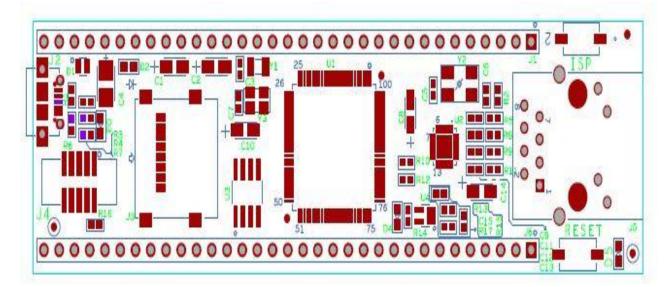


Fig.36

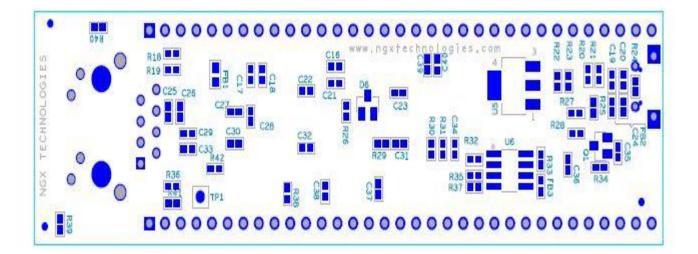


Fig. 37



# **5.0 CHANGE HISTORY**

# **5.1 Change History**

Rev	Changes	Date (dd/mm/yy)	Ву
1.0	Initial release of the manual	25/06/2012	Veeresh Tumbaragi



#### About this document:

#### **Revision History**

Version: V1.0 author: Veeresh Tumbaragi

#### **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.