

LPC1768-Xplorer

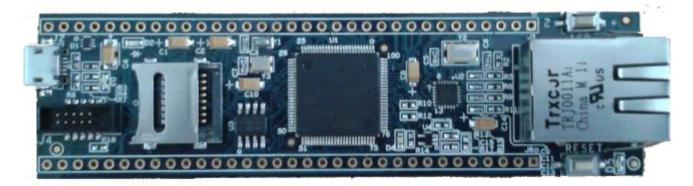


Fig.1



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 LPCXpresso software	5
2.3 Setup for NXP LPC-Link and LPC1768 Xplorer Board	
3.0 LPC1768 Xplorer firmware Development	
3.1 Executing the sample projects in LPCXpresso	
3.2 Creating the sample Blinky project in LPCXpresso	
4.0 LPC1768-Xplorer Programming	
4.1 Programming options	
4.1.1 Secondary bootloader (USB)	
4.1.2 Flashing the board using USB	
5.0 Restoring LPC1768-Xplorer to Factory Defaults	23
5.1 LPC-link and LPCXpresso	23
6.0 Schematic & Board Layout	26
6.1 Schematic	26
6.2 Board layout	26
7.0 CHANGE HISTORY	27
7.1 Change History	27



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 LPCXpresso and NXP LPC-Link as the IDE and debugger respectively.

2.2 Installation & Configuration of LPCXpresso software

For installation and configuration of LPCXpresso: Click here.

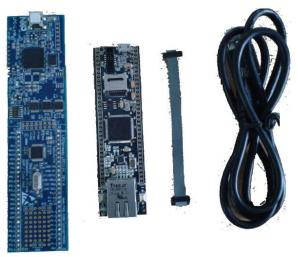
Note: We have used **LPCXpresso version 4.2.2_275** *while creating the User manual for this evaluation kit. Please ensure that you are using* **LPCXpresso version 4.2.2_275 or** *above.*

2.3 Setup for NXP LPC-Link and LPC1768 Xplorer Board

The Xplorer board has on board 10 pin SWD/JTAG box, the 10 pin cable is not a part of the LPC1768 Xplorer package, the user needs to buy 10 pin cable separately.

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

- NXP LPC-Link
- 10 pin cable
- LPC1768 Xplorer Board
- One USB AM to Micro B cable





Steps to setup the LPC-Link and LPC1768 Xplorer Board: (*Note: Please refer keil knowledgebase article for connecting 10-pin ribbon cable to NGX Xplorer*)

Step 1: Connect one end of 10-pin ribbon cable to 'LPCLink 10-pin connector'; the 10-pin ribbon cable

User Manual: LPC1768-Xplorer



header notch should facing towards the 'NXP LPCLink Mark' as shown in the following image.

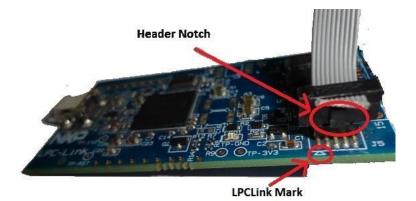
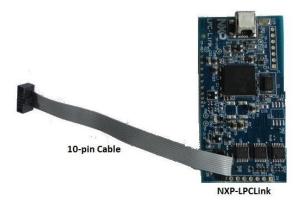


Fig.3





Step 2: Connect other end of 10-pin ribbon cable to '10-pin box header' of the LPC1768 Xplorer board and 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 mini B' to LPC-Link and other end to computer.

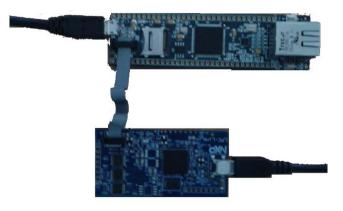


Fig.5 Step 4: The setup is now ready to be used for development with LPCXpresso and NXP LPCLink.



3.0 LPC1768 Xplorer firmware Development

3.1 Executing the sample projects in LPCXpresso

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

Steps to execute the sample project:

1. Open LPCXpresso, Browse the folder which contains LPC1768_Xplorer_LPCXpresso.zip project and Click Ok.

🔛 Workspace	e Launcher
Select a w	orkspace
	stores your projects in a folder called a workspace. orkspace folder to use for this session.
Workspace:	F:\LPC1768-Xplorer_Sample_programs\LPC1768-Xplorer_LPCXpresso
Use this a	s the default and do not ask again
	Cancel

Fig.6

2. Click on Import and Export then Click on Import archived projects (zip).

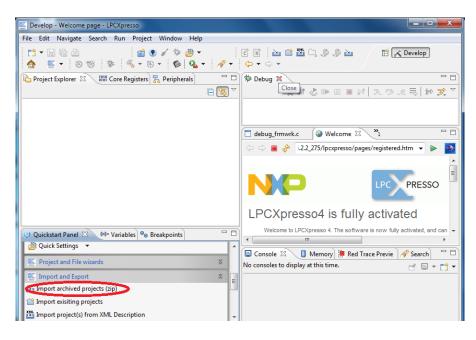


Fig.7



3. Click on Browse to select a archived project (zip) and Click on Open

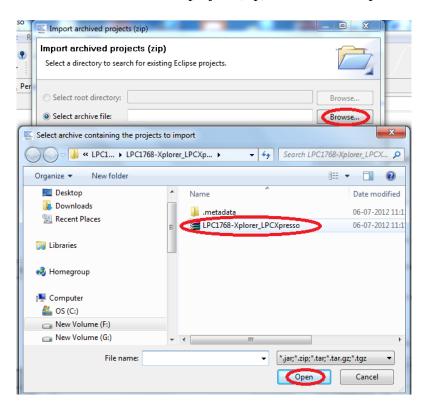


Fig.8

4. Click Finish.

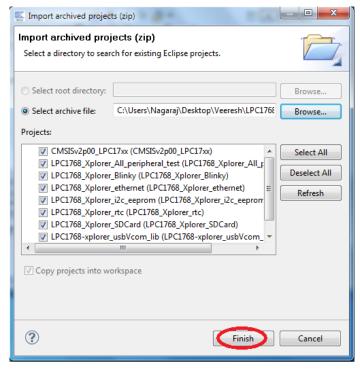


Fig.9



5. Right click on "LPC1768_Xplorer_Blinky" and left click on Build Project.

Develop - Welcome page		CV proces		
		New	+	
		Go Into		
C' N' L C A m C' N' ≧ 1		Open in New Window		🔀 Develop
📎 🐯 – 🕸	D	Сору	Ctrl+C	
(-) ▼ (-) ▼	Ê	Paste	Ctrl+V	
Project Explorer	×	Delete	Delete	Deb 🛛 🗖
		Move		
E CMSISv2p00_LPC17		Rename	F2	k 🛤 🕹 🕩 💷
b 😂 LPC1768_Xplorer_A	2	Import		
Description: De	4	Export		
D D LPC1768_Xplorer_et	_) 🛛 🚬 🖓 🖓
 b S LPC1768_Xplorer_i2 b S LPC1768_Xplorer_rt 		Build Project		> _⊗ •
EPC1768_Xplorer_SL		Clean Project		
N 1 PC1768-vnlorer us	8	Refresh	F5	≥ × × □
U Quickstart Panel		Close Project		consoles to display at 1
Project and File wizard		Close Unrelated Projects		₫ 🖳 ד 🔂 ד
Import and Export		Build Configurations	+	
🔤 Import archived projec		Make Targets	•	
😭 Import exisiting projec		Index	+	
□ [◆] 😂 LPC1768_>		Convert To		
		Run As	+	XP LPC1768
LE		Debug As	+	AF LFC1700

Fig.10

6. For selecting a debug option, right click on LPC1768_Xplorer_Blinky project, go to Lunch Configurations then select Open Current Launch Configurations.

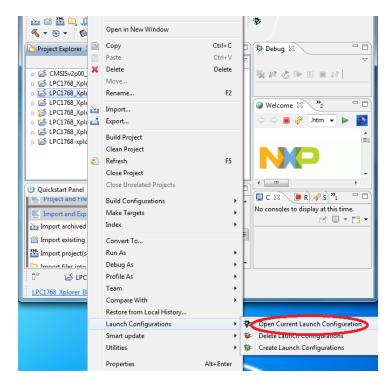


Fig.11



7. Select LPC1768_Xplorer_Blinky Debug then select Debugger and select NXP LPC17xx option and click on Apply.

E Debug Configurations		×
Create, manage, and run configurations		Ť.
Image: Second Secon	Name: LPC1768-Xplorer_led_blinky Debug Main Debugger E Source Common Stop on startup at: main Advance Debugger Options Target configuration NxP_LPC17x Script values Configuration Option	ed
C LPC1768-Xplorer_led_blinky Debug C LPC1768-Xplorer_red_blinky Releas C I PC1768-Xplorer_red_blinky Releas C LPC1768-Xplorer_red Release C LPC1768-Xplorer_sdcard Debug C LPC1768-Xplorer_sdcard Release Filter matched 21 of 21 items	Ally Crystal (XTAL) frequency and PLL value EVector catch Enable Red Trace Semihosting support III	false true Auto
?		Debug Close

Fig.12

8. After building project click on Debug.

Evelop - LPC1768_Xplorer_Blinky/src/blinky.c - LPCXpresso	
File Edit Source Refactor Navigate Search Run Project Window	Help
C1 - 2 R △ 0 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 -	C n Evelop
🔁 Project Explorer 🕱 📲 Core Registers 🛃 Peripherals 🦳 🗖	🏇 Debug 🛛 🗖 🗖
□ 🔄 🏹	~
CMSISv2p00_LPC17xx	🍇 🛤 🕹 🕪 💷 🔤 🙌
EPC1706_Aplorer_olinky	blinky.c 🖾 🔭 🗆
EPC1768_Xplorer_i2c_eeprom	
▷ EPC1768_Xplorer_rtc	45
▷ LPC1768_Xplorer_SDCard LPC1768-xplorer_usbVcom_lib	46 int main(void) {
	47
U Ouickstart Panel 🖄 😡= Variables) 💁 Breakpoints	🗐 C 🖾 🚺 M 🎦 R 🔗 S 📃 🗖
Project and File wizards	No consoles to display at this time.
Import and Export	
🚵 Import archived projects (zip)	
😭 Import exisiting projects	
Mt Import proiect(s) from XML Description	
□ D E LPC1768_Xplorer_Blinky	
LPC1768 Xplorer BlinkNXP LPC1768	

Fig.13



9. Click on Run and select Resume (F8) to start debugging the project. Here the two LED (D4) start blinking.

Develop - LPC1768_Xplorer_Blinky/src/blinky.c - I	LPCXpresso
File Edit Source Refactor Navigate Search	Run Project Window Help
	F8 Resume F8
2 🗠 📇 🕞 🖉 🦉 🗠 🛛 🧄 🕵	00 Suspend
🚳 💁 र 🥭 🖉 र 📝 🗇 र 🔿 र	📕 Terminate Ctrl+F2
	Step Into F5
Project Explorer 🛛 🛛 🚻 Core Registers 🔂 P	Step Over F6
	Step Return F7
EPC1768_Xplorer_All_peripheral_test	🖘 🛛 Run to Line 🛛 Ctrl+R
LPC1768_Xplorer_Blinky Binaries	😿 Use Step Filters
ncludes	Q External Tools + c 🛛 🔭 🗆 🗆
😕 src	46 int main(void) {
c blinky.c	47
cr_startup_lpc176x.c	248 // TODO in the cod
i Debug i Debug i Debug	♦ 49 LPC_GPI00->FIODIR
LPC1768 Xplorer Blinky Debug.launch	▼ ★ Ⅲ ►
🕘 Quickstart Panel 🕴 😡= Variables 💁 Break	repoints 🛛 🗖 📮 C 🖾 🚺 M 🎘 R 🛷 S 🖓 🗖
Project and File wizards	LPC1768_Xplorer_Blinky Debug [C/C++ MCU A
Import and Export	*
Import archived projects (zip)	📊 🛃 🖳 🕶 🔂 🕶
🕋 Import exisiting projects	A
Import project(s) from XML Description	▼
□ [◆] Writable	Smart Insert
LPC1768 Xplorer BlinkNXP LPC1768	

Fig.14



3.2 Creating the sample Blinky project in LPCXpresso

Step 1: Open an LPCXpresso v4.2.2_275 IDE.

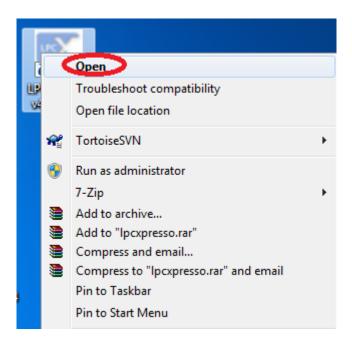


Fig.15

Step 2: Click Browse.. as show below.

Select a workspace			
LPCXpresso stores your projects in a folder ca Choose a workspace folder to use for this ses			
Workspace:		- (Browse
Use this as the default and do not ask agai	1		
		 ж П	Cancel

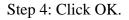
Fig.16



Step 3: Please select LPC1768-Xplorer_Examples_V1 folder which downloaded from website, as shown below image and click OK as shown in below image.

Se	elect Workspace Directory	x
	Select the workspace directory to use.	
	LPC1768-Xplorer_LPCXpresso	*
	🛛 📙 .metadata	
	D CMSISv1p30_LPC17xx	
	D LPC1768-Xplorer_all_peripheral_test	
	D LPC1768-Xplorer_ethernet	
	LPC1768-Xplorer_extint	
	LPC1768-Xplorer_i2c_eeprom	-
	LPC1768-Xplorer_led_blinky	
	LPC1768-Xplorer_rtc	
	LPC1768-Xplorer_sdcard	
	LPC1768-xplorer_usbVcom_lib	-
	• III	Þ.
	Folder: LPC1768-Xplorer_LPCXpresso	
	Make New Folder Canc	el

Fig.17



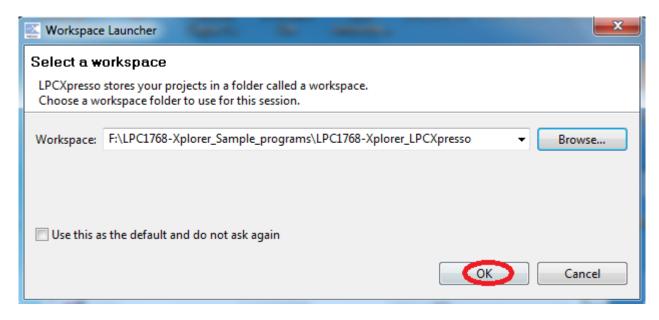


Fig.18



Step 5: Click on File -> New -> Project... as shown below image.

File	Edit	Navigate	Search	Run	Project	Window	Help					
	New					Alt	+Shift+N		Project),© 🔤 🗍	
	Open I	File						EŶ.	Other	Ctrl+N		
	Close						Ctrl+W					
	Close	All				Ctrl-	Shift+W					
	Save						Ctrl+S					
<u>.</u>	Save A	ls										
6	Save A	All.				Ctr	+Shift+S					
	Revert											
	Move.											
	Renam	ne					F2					
8	Refres	h					F5					
5	Print						Ctrl+P					
	Switch	Workspace	e					·				
	Restart	t										

Fig.19

Step 6: Select LPCXpresso C Project and Click Next as shown below image.

Kew Project	
Select a wizard	
Wizards:	
type filter text	
 ▷ General ▷ C/C++ ▷ C++ Project ▷ C++ Project ▷ CPCXpresso C++ Project ⓒ LPCXpresso C++ Project ⓒ Makefile Project with Existing Code ▷ ▷ CVS 	
Rext > Finish	Cancel

Fig.20



Step 7: Select NXP LPC1700 projects -> C Project and click Next as shown below image.

Wizard		
⊿ NX	P	
	NXP LPC1100 projects	
	NXP LPC1102 projects	
	NXP LPC11E00 projects	
	NXP LPC11U00 projects	
	NXP LPC1200 projects	
	NXP LPC1300 (12bit ADC) projects	
	NXP PC1300 projects	
-	NXP LPC1700 projects	 _
	C Static Emple C project	_
	FreeRTOS Project	
	Semihosting C Project	
N	NXP LPC177x/8x projects	
	NXP LPC1800 projects	
V		_

Fig.21

Step 8: Give Project name is Blinky and Click Next.

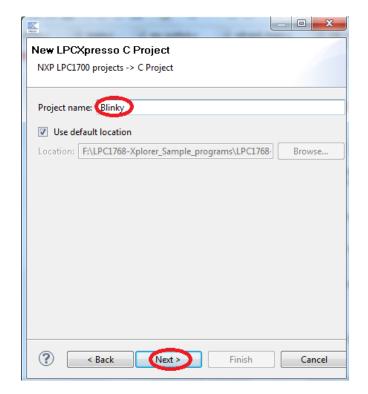


Fig.22



Step 9: Select the target MCU is LPC1768 and click Next as shown in below image.

New LPCXpresso C Project	
Select the target MCU	
Target selection	
NXP LPC1768	^
LPC1758	
LPC1759	
LPC1763	
LPC1764	
LPC1765	
LPC1766	E
LPC1767	
1PC1768	
LPC1769	
LPC1772	
LPC1774	
LPC1776	T
Constant Constan	Cancel

Fig.23

Step 10: Click Finish.

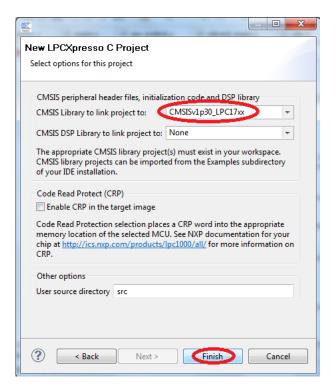


Fig.24



Step 11: The New Blinky Project is created, double click on main.c file as shown below image.

Develop - Welcome page - LPCXpresso	
File Edit Navigate Search Run Project Window Help	
) 🖌 🅸 ▼ 🔹 🖆 👘 🔛 🛣 Develop 🕸 🔦 ▼ 🛞 ▼ 🕼 🖗 💁 ▼
🎦 Project Exp 🖾 👭 Core Regist 🛃 Peripherals 🦳 🗖	🗈 timer_utilitis.c 🛛 🕲 Welcome 🛛 🔭 👘 🗖
□ 🔄 🏹	
➢ Blinky ▷ D Includes ☑ cr startup_lpc176x.c ☑ main.c ☑ cr startup_lpc176x.c	
🙂 Quickstart Pa 🛛 🕪= Variables 💁 Breakpoints 🖓 🗖	LPCXpresso4 is fully activated
Start here	Welcome to LPCXpresso 4. The software is now fully activated, and c generate and download applications containing up to 128KB of code ir
Import project(s)	
Build all projects [Debug]	📮 Console 🛛 🧻 Memory 📳 Red Trace Preview 📃 🗆
🐔 Build 'Blinky' [Debug] 👻	No consoles to display at this time. 📑 📃 👻 📑
□ [◆] 😂 Blinky	
Blinky NXP LPC1768	

Fig.25

Step 12: After double click the main.c file will open in LPCXpresso as shown in below image

E Develop - Blinky/src/main.c - LPCXpresso	1	
File Edit Source Refactor Navigate Search Run	Projec	ct Window Help
	e 🕐) 🧹 🏇 👺 🔻 🔂 😭 🔛 🔛 🔀 Develop
🕹 🖆 💯 💭 💭 🕍 🛛 🧴 🐳 📎		
	_	
陷 Project Exp 🕺 🔐 Core Regist 🛃 Peripherals 🗧	- 0	Welcome 🚺 main.c 🛛 🎇 6 🗖 🗖
E 🖉	∇	
😂 Blinky 🔊 Includes	-	28 29 // TODO: insert code here 30
😂 src		31 // Enter an infinite loop, just increm
cr_startup_lpc176x.c		<pre>32 volatile static int i = 0 ;</pre>
C main.c	-	33 while(1) {
		34 i++;
		35 } 36 return 0 ;
🕵 Start here 🖈		
📸 New project		38 👻
Import project(s)		< III +
Build all projects [Debug]		📮 Console 🛛 🧯 Memory 🗿 Red Trace Preview 📃 🗖
K Build 'Blinky' [Debug]	-	No consoles to display at this time. 📑 📃 👻 📩
□ [◆] Writable S	mart l	: Insert 8 : 23
Blinky NXP LPC1768		

Fig.26



Step 13: Write a C program for LED blinky, after the program Click on Save as shown in below image. (*Note: Writing a C code for Blinky project Please refer Downloaded LPC1768-Xplorer_led_blinky example*)

Develop - Blinky/src/main.c - LPCXpresso	
File Edit Source Refactor Navigate Sear	rch Run Project Window Help
Image: Save All (Ctrl+Shift+S) Image: Save All (Ctrl+Shift+S)	 2 ✓ 3 3 3 4 5
🎦 Proje 🖾 📲 Core 🔀 Perip 📃 🗆	🗈 timer_utilitis.c 🛛 Welcome 🔂 *main.c 🖾 🔭 🗖 🗖
 ▶ ⓐ Includes ▲ ☞ src ▶ ⓓ cr_startup_lpc176x.c ▶ ⓓ cr_startup_lpc176x.c ▶ ⓓ cr_startup_lpc176x.c ▶ ⓓ CMSISv1p30_LPC17xx ♥ ☑ CMSISv1p30_LPC17xx ♥ ☑ Quick ※ (M= Varia ● Break ♥ ☑ Quick ※ (M= Varia ● Break ♥ Import project ♥ Import project(s) ➡ Build all projects [] ♥ Build all projects [] ♥ Clean "[] 	<pre>58 } 59 } 60 61 int main(void) { 62 63 // TODO: insert code here 64 LPC_GPIOO->FIODIR =TEST_LED; 65 // Enter an infinite loop, just incrementing a 66 67 while(1) 68 { 69 test_led(); 70 } 71 72 } 73</pre>
□ □ ↓ 0 items selected	
LPC1768-	:

Fig.27

Step 14: To include LPC17xx.h file in your project right click on your Project and click on Properties.

🔛 Develo	Move Rename	F2	
File Edit	Nename	12	ect Window Help
i 🚽 🚽 🔛	Import		🗸 🏇 🗢 📑 🖍 📑 🔀 Develop
	Export		7 6 100 100 100 100
S Q	Build Project		
Proje	Clean Project		tis.c 🚱 Welcome 🕼 main.c 🕱 🔭 🗖 🗖
2	Refresh	F5	}
6 🔁 BD	Close Project		
	Close Unrelated Projects		
Þ 😂 LI	Build Configurations	•	main(void) {
> 😂 LF	Make Targets		TODO: insert code here
> 📂 Li	Index	, r	LPC GPIOO->FIODIR =TEST LED;
⊳ <u>6</u> 5 Ц	Index	•	/ Enter an infinite loop, just incrementing a
	Convert To		
U Quick	Run As	+	hile(1) E
💽 Star	Debug As	+	test_led();
😭 New	Profile As	+	
🔹 💽 Impc	Team	+	🖇 🚺 Memory 📒 Red Trace Preview 📃 🗖 🗖
🗟 🗟 Build	Compare With	•	/] ↓ ☆ 🗐 🖬 🖬 🖬 = ▼ 🗊 ▼
K Build	Restore from Local History		data bss dec hex filename 🔺
Clear	Launch Configurations	+	0 0 1124 464 Blinky.axf
	Smart update	•	
	Utilities	+	
Blinky	Properties	Alt+Enter	

Fig.28



Step 15: Select C/C++ Build ->Settings then Includes, click on Add then click on workspace to select the .h files path as shown below image.

Properties for Blinky		
Resource Builders 7/C++ Build Build Variables Discovery Options Environment Logging MCU settings Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings	Settings Settings Include paths (-1) Symbols Symbols Includes Optimization Debugging Warnings Warnings Miscellaneous Target MCU Assembler General Target MCU Inker General Libraries Miscellaneous Shared Library Settings Target	
?	ОК	Cancel

Fig.29

Step 16: Select LPC17xx.h file path then click OK as shown below image.

		x
Select one or more Workspace Folders		
-		
⊳ 🚝 Blinky		
CMSISv1p30_LPC17xx		
🔺 🦾 Core		
CMSIS		
a 🗁 Device		
a 🗁 NXP		=
LPC17xx		
Include I		
Source		
Debug		
> 🗁 docs		
b 🗁 inc		
▷ b c c c c c c c c c		
EPC1768-Xplorer_all_peripheral_test		
Description: De		
DESCRIPTION DESCRIPTON DESCRIPTION DESCRIPTON DESCRIPANTON DESCRIPTON DESCRIPANT		Ψ.
ОК	Canc	el

Fig.30



Step 17: Click OK and OK as shown below image.

Properties for Blinky type filter text	Settings
Resource Builders C/C++ Build Build Variables Discovery Options Environment Logging MCU settings Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings	MCU C Compiler Preprocessor Symbols Includes Optimization Debugging Warnings Miscellaneous Target MCU Linker General Libraries Miscellaneous Shared Library Settings Include files (-include)
	€ III →
?	Cancel

Fig.31

Step 18: Right Click on Blinky Project and click on Build Project, build must be error free.

Develop	New	
File Edit	Go Into	ct Window Help
	Open in New Window	✓ ॐ ※ ▼ C h C h C b <p< th=""></p<>
	Copy Ctrl-	+C 🕸 🗞 🛪 🛞 🛪 🥬 🖓 🖬
	Paste Ctrl-	
🕒 Proje 🗙	Delete Dele	ete s.c 🥥 Welcome 📝 main.c 🛛 🔭 🗖 🗖
	Move	}
	Rename	F2
⊳ 🚰 CN ⊳ 🚰 LPi 🚵	Import	main(void) {
> 😂 LP 🛃	Export	TODO: insert code here
▷ 25 LP ▷ 25 LP	Build Project	LPC_GPIO0->FIODIR =TEST_LED;
	Clean Project	Enter an infinite loop, just incrementing a
		F5 ile(1)
U Quick	Close Project	
💽 Start	Close Unrelated Projects	<pre>test_led();</pre>
📸 New p	-	
🕐 Impor	Build Configurations	,
🗟 Build	Make Targets	·
	Index	

Fig.32



Step 19: After successful build, click on Debug as shown in below image

Develop - Blinky/src/main.c - LPCXpresso					
File Edit Source Refactor Navigate Sea	arch Run Project Window Help				
Image: Second secon					
Proje 🛛 🚻 Core 🔀 Perip 🗖 🗖	🛛 🖻 timer_utilitis.c 🛛 🥹 Welcome 🔂 main.c 🖾 🍡 S 👘 🗖				
Image: Start here	<pre>50</pre>				
② New project ■					
Import project(s)	🗏 🖾 Console 🛛 🚺 Memory 📳 Red Trace Preview 🛛 🖓 🗖				
🗟 Build all projects [Debug]	C-Build [Blinky] 🕹 🗘 🕼 📓 🖓 🖃 🛪 📬 🗸				
🔏 Build 'Blinky' [Debug]	text data bss dec hex filename 🔺				
🧹 Clean 'Blinky' [Debug] 📼	1124 0 0 1124 464 Blinky.axf				
□◆					
Blinky NXP LPC1768					

Fig.33

Step 20: Click Resume for free running, the LED D4 is starts blinking on LPC1768-Xplorer.

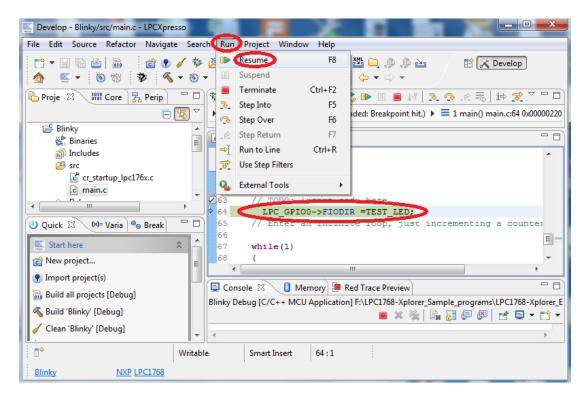


Fig.34



4.0 LPC1768-Xplorer Programming

4.1 Programming options

LPC1768-Xplorer can be programmed using the

- Secondary USB bootloader (USB)
- Debugger (NXP LPC-LINK)

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:

- ▲ **Press and hold** the ISP button
- ▲ Press the RESET button and release it
- \checkmark Now release the ISP button
- \checkmark 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. 35



Flashing the bin file(Drag & drop).

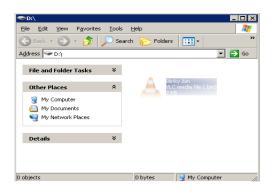


Fig. 36

5.0 Restoring LPC1768-Xplorer to Factory Defaults

5.1 LPC-link and LPCXpresso

To restore the factory defaults for the Xplorer, the user needs the LPC-link to flash **LPC1768-Xplorer_all_peripheral_test.bin** file in LPCXpresso.

Steps to restore the factory defaults for Xplorer:

Step 1: Open LPCXpresso, Import the LPC1768-Xplorer_LPCXpresso.zip, as explained above in the *section 3.1*. Click on the **program flash.**

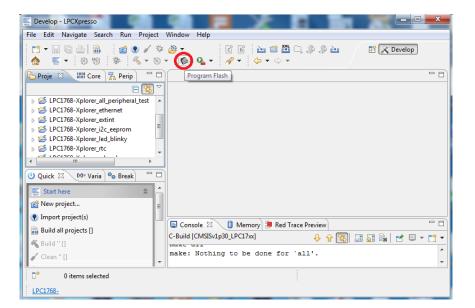


Fig.37

Step 2: Click on Browse.



Program Flash	
Program target flash: LPC17xx	Samaan
Options	
Display progress log	Reopen on completion
Reset target on completion	Repeat on completion
Connection Options	
Speed 250	
Use JTAG interface	
Code Read protect	
CRP selection None	•
Program flash memory Erase flash memory	
Select file	- Browse
Base address	
Erase Options	
Mass erase	Erase only required pages
	OK Cancel

Fig.38

Step 3: Select *.bin and then select the following path..**Binaries****LPCXpresso****Bin****LPC1768-Xplorer_all_peripheral_test.bin** and click open.

Select &file			X
Goo ♥ ↓ ≪ Binaries ▶ L	PCXpresso 🕨	Bin 👻 🍫 Search Bin	م
Organize 🔻 New folder		:== :==	• 🔟 🔞
🧮 Desktop	*	Name	Date modified
🐌 Downloads		LPC1768-Xplorer_all_peripheral_test.bin	11-05-2012 17:5
📃 Recent Places	=	LPC1768-Xplorer_ethernet.bin	11-05-2012 17:5
		LPC1768-Xplorer_extint.bin	11-05-2012 15:1:
🛜 Libraries		LPC1768-Xplorer_i2c_eeprom.bin	11-05-2012 15:0
🤞 Homegroup		LPC1768-Xplorer_led_blinky.bin	10-05-2012 19:0
		LPC1768-Xplorer_rtc.bin	11-05-2012 15:0
J. Computer		LPC1768-Xplorer_sdcard.bin	11-05-2012 15:10
🏭 OS (C:)			
👝 New Volume (F:)			
👝 New Volume (G:)	-	•	÷.
File name	: LPC1768-X	plorer_all_peripheral_te 👻 🔭	-
		Open	Cancel

Fig.39

Step 4: click OK.



E.	×
Program Flash	
Program target flash: LPC17xx	1 automation
Options	
Display progress log	Reopen on completion
Reset target on completion	Repeat on completion
Connection Options	
Speed 250	
Use JTAG interface	
Code Read protect	
CRP selection None	•
Program flash memory Erase flash memory	у
Select file Binaries\LPCXpresso\Bin\LPC1	768-Xplorer_all_peripheral_test.bin 👻 Browse
Base address	
Erase Options	
O Mass erase	Erase only required pages
	Cancel

Fig.40

Step 5: Wait till the download process is completed then RESET the Xplorer, to restore the Xplorer to factory defaults.

E Develop - LPCXpresso						
File Edit Navigate Search Run Project Window Help						
C1 ▼ □ C2 C1 C2 C2						
Proje 🛛 👫 Core 🚦	🛃 Perip 📃 🗖					
Progress Information						
➢ Blinky ➢ CMSISv1p30_LP ➢ LPC1768-Xplore	S CMSISv1p30_LP LPC1768-Xplore 0 Writing 43940 bytes to 0000 in Flash (assumed clock: 4.0MHz)					
EPC1768-Xplore						
EPC1768-Xplore						
-C 1001700 V 1						
	Cancel Details >>					
藍 Start here						
📸 New project	Ξ					
Import project(s)						
🗟 Build all projects []		📮 Console 🛿 🥼 Memory 📳 Red Trace Preview				
🔏 Build '' []		C-Build [CMSISv1p30_LPC17∞] ↓ ☆ 🔄 🔜 🖓 💌 = -	<u> 1</u>			
Clean " []	-	make: Nothing to be done for `all'.	*			
□ [◆] 55%: Downloaded 24576 bytes						
LPC1768-						

Fig.41

The same way we can flash the any bin file to the target board.



6.0 Schematic & Board Layout

6.1 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.

6.2 Board layout

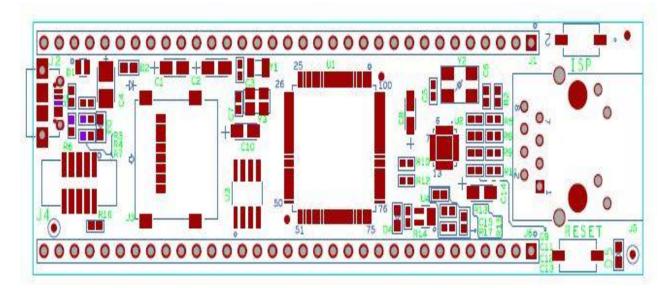


Fig.42

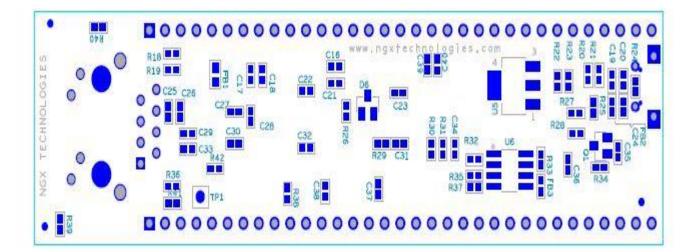


Fig. 43



7.0 CHANGE HISTORY

7.1 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.