

BlueBoard-LPC1343



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

CE certification

NGX Technologies BLUEBOARD-LPC1343 board have 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 behaviour. 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 BLUEBOARD-LPC1343 Development Tool Setup	5
2.1 IDE and debugger	5
2.2 Installation & Configuration of CoIDE software	5
2.3 Setup for NGX CoLink and BLUEBOARD-LPC1343 Board	
2.4 Configuration of NGX CoLink Debugger	9
3.0 BLUEBOARD-LPC1343 Software Development	9
3.1 Executing the sample projects	
3.2 Creating New project	
3.3 Bin File	
4.0 BLUEBOARD-LPC1343 Programming	
4.1 Programming options	19
4.1.1 On-Chip bootloader (USB or UART)	
4.1.2 Flashing the board using USB	
4.2 Flashing the Hex file through UART	
5.0 Schematic & Board Layout	
5.1 Schematic	
5.2 Board layout	
6.0 Trouble Shooting	
7.0 CHANGE HISTORY	
7.1 Change History	
8.0 REFERENCES	



1.0 INTRODUCTION

This document is the User Manual for the BLUEBOARD-LPC1343, 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 BLUEBOARD-LPC1343 features and BLUEBOARD-LPC1343 verification.

For BLUEBOARD-LPC1343 quick start guide Click here.

For the most updated information on the BLUEBOARD-LPC1343 board please refer to NGX' website.



2.0 BLUEBOARD-LPC1343 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 CoIDE and NGX CoLink as the IDE and debugger respectively.

2.2 Installation & Configuration of CoIDE software

The Installation of CoIDE software is explained below:

Note: We have used **CoIDE 1.4.2** while creating the User manual for this evaluation kit. Please ensure that you are using version 1.4.2 and above.

Step 1: Open CoIDE 1.4.2 setup







Step 2: CoIDE 1.4.2 information Click on Next





Step 3: Provide the destination path as shown in below image and Click on Next.



Fig. 4



Step 4: Click on Next.

🚏 Setup - CoIDE	<u>- 🗆 ×</u>
Select Start Menu Folder Where should Setup place the program's shortcuts?	
Setup will create the program's shortcuts in the following Start Menu folde	er.
To continue, click Next. If you would like to select a different folder, click Browse.	
CooCox Software Browse.	·-
< Back Next > C	Cancel

Fig. 5

Step 5:Click on Install.

🖶 Setup - CoIDE	
Ready to Install Setup is now ready to begin installing CoIDE on your computer.	
Click Install to continue with the installation, or click Back if you want to review or change any settings.	
Destination location: C:\CooCox\CoIDE	A
Start Menu folder: CooCox Software	
< Back Install	Cancel

Fig. 6



Step 6: Click on Finish.



Fig. 7

2.3 Setup for NGX CoLink and BLUEBOARD-LPC1343 Board

The BlueBoard-LPC1343 board has on board 10 pin SWD box, the NGX CoLink is not a part of the BlueBoard-LPC1343 package, the user needs to buy it separately.

To run the BlueBoard-LPC1343 examples you need have the following components and the image shows the each components:

- NGX CoLink.
- BlueBoard-LPC1343 Board.
- Mini USB cable.

NGX CoLink with cable as shown in below image.



Fig. 8



Connections of components are shown in the below image.





The above setup is ready to use for development in CoIDE .

2.4 Configuration of NGX CoLink Debugger

The configuration flow of NGX CoLinkEx Debugger is explained below:

Before configuring the NGX CoLink select the tool chain path as C:\CooCox\CoIDE\gcc\Sourcery G++ Lite\bin as shown in the below images.

User Manual: BlueBoard-LPC1343



🚺 CooCox CoIDE	
File Edit View Project Flash Debug Search Help	
Image: Second state sta	
Common Open Folder As Project Open Folder As Project M0 Creater Select Toolchain Path	
Boot Configuration Configuration	
Peripheral.ND Build Example Comm Build Example Comm Build Clean	
Project Signal F:\Technical\Dharmanna\BB_COIDE\BB_LPC11U14\BB_LPC11U14_Buzzer\BB_LPC11U14 F:\Technical\Dharmanna\BB_COIDE\BB_LPC11U14\BB_LPC11U14_Blinky\BB_LPC11U14	-
Error Comsis	
Ercipitation Erci	
Crefific_lib Syscalls Syscalls Duild.xml Ink.ld Main.c memory.ld	
Sign in to CooCox Writable Smart Insert 1:1	

Fig. 10



Fig. 11



File Edit View Project Flash Debug - Search Help
📫 🗎 🏙 🏭 🏇 🤐 🦉 🏑 毘 🛷 - 🐓 - 🖓 - 🦙 🔶 - 🔿 -
🗖 Components 🕴 👘 🖬 🔝 👘 🖬 Repository 👘 🖬 🖇 👘 🗖
Device [LPC1114v201]
Browse For Folder
PC
Erowse Default oid Browse Default oid
Sourcery G++ Lite
Pr OK
🕂 🗁 c 🗈 🕒 libexec
Ender: bin
Make New Folder OK Cancel
Sign in to CooCox Writable Smart Insert 1:1
Fig. 12
CooCox CoIDE
File Edit View Project Flash Debug · Search Help
📑 📄 🖫 🎬 🏭 🏇 🤐 🎬 🛷 閉 🛷 • 🧏 - 🎘 - 🤚 - 😓 -
🗖 Componen 🕱 🗖 🗒 Repository 🗙 👘 🗖 🔯 Help 🕸 👘 🗖
Device [LPC1343] Step 3 Select Components [NXP / LPC1343]
Common Select Toolchain Path
Taalahain Datha

CooCox CoIDE		
File Edit View Project Flash	Debug - Search Help	
] 📑 🗎 🔚 🏙 🊟 🔅 🏇	🮯 🛎 🖉 閉 🔗 • ½ • ⅔ • ↔ ↔ ↔	
Componen 🛛 🗖	🗮 Repository 🗙 📃 🔯 Help 😫	
Device [LPC1343]	Step 3 Select Components [NXP / LPC1343]	⇒ &
🛛 🕕 Select Toolchain	Path	×
Toolchain Path		
C:\CooCox\CoIDE\g	cc\Sourcery G++Lite\bin Defau Defau	lit
Project		
E 🚝 BE	ОК	
Ē		
E Cmsis_boot		
E···		
	Manufacturers Chips Peripherals Drivers Others	
Sign in to CooCox		

Fig. 13



Step 1: Open CoIDE project for example: BB_LPC1343_Blinky then click on Debug Configuration as shown in the below image.

CooCox CoIDE			
File Edit View Project	Flash Debug Search He	φ	
] 📑 📄 🔛] 🎬 🏙	1 * 🛞 1 🖉 🗄	팀] 🔗 •] 🖢 • 원 • 🥲	$\Leftrightarrow \bullet \Rightarrow \bullet$
	Repository X Debug Configuration		Hel 🛛 🗖 🗖
Device [LPC1343]	Step 3 Select Compor	_	← ⇒ &
Common			<u>^</u>
C Library	COMMON		
Grand CMSIS con	C Library	Implement the minimal funct	
	Retarget printf	Implementation of printf(), s	
	Semihosting	Implementation of Semihost	
	CMSIS core	CMSIS core for Cortex M3 V	
BB_LPC1343_Blinky	🗆 ВООТ		
Do_crecis to_billiky	Cmsis Boot	CMSIS Boot for LPC13xx	
🗄 🗁 cmsis_boot	•		
E Debug	Manufacturers Chips Perip	erals Drivers Others	-1
Sign in to CooCox	😂 BB_LPC1343_Blinky		

Fig.14

Step 2: Click on Example: BB_LPC1343_Blinky.configuration then select ColinkEx adapter, select SWD port and 1M Max Clock(Hz) as shown in the image below.

🚺 Debug Configurations	×
Create, manage, and run configurations	Ť.
type filter text Debugger Down Cortex-M Application Hardware Cortex-M Application Hardware Loc BB_LPC1343_Blinky.confid Adapter ColinkEx Startup Startup Unit to main ColinkEx	Load
Advance Reset M de SYSRESETREC TargetInfo Host IP Address: 127 .	
Filter matched 3 of 3 items	Apply Revert



Step 3: Select Download option then click on default and apply then click on close as shown in image.

Debug Configurations	×
Create, manage, and run con	figurations
type filter text	Image: Constraint of the second se
Filter matched 3 of 3 items	Apply Revert
	Close

Fig.16

3.0 BLUEBOARD-LPC1343 Software Development

3.1 Executing the sample projects

The sample projects are provided with the available kit. Steps to execute the sample projects:

- 1. Open the project folder.
- 2. Then open the file project_name.cob eg BB_LPC1343_Blinky.cob.

BB_LPC1343	_Blinky	<u>- </u>
GO ~)	▼BB_LP ▼ BB_LP ▼ 🛛 ▼ 🚱	Search BB_LPC1343 😥
Organize 🔻 🌔	🚺 Open 🔻 Burn New folder	
☆ Favorites	Name ^	Date modified
	BB_LPC1343_Blinky.elf.xcodeproj	04-06-2012 15:5
📃 Desktop	鷆 cmsis	04-06-2012 15:5
	퉬 cmsis_boot	04-06-2012 15:5
	i Debug	04-06-2012 15:5
	🌗 lpc13xx_lib	04-06-2012 15:5
	鷆 syscalls	04-06-2012 15:5
	cproject	20-06-2012 12:
	project	07-06-2012 15:2
	BB_LPC1343_Blinky.cob	20-06-2012 12:3
	Dinneyro	01-06-2012 15:5
	•	Þ
BB	3_LPC1343_Blinky.cob Date modified: 20-0	6-2012 12:35
Co	IDE Project Size: 719	bytes

Fig. 17



3. This launches the IDE

🚺 CooCox CoIDE			
File Edit View Project Flash	Debug - Search Help		
📫 🗎 🔚 🛗 🎬 🖄	• 🎯 🛎 🛷] 🎟] 🔗	•] ½ • № • ← + + +	
Compone 🛛 🗖 🗖	🗒 Repository 🗙	□ □	
Device [LPC1343]	Step 3 Select Componer	nts [NXP / LPC 1343]	$\leftarrow \Rightarrow \mathfrak{S}$
Common	1		
- Boot	C Library	Implement the minimal functionality required to allow	
Cmsis Boot	Retarget printf	Implementation of printf(), sprintf() to reduce memor	
⊡ · Peripheral.NXP	Semihosting	Implementation of Semihosting GetChar/SendChar	
Ormmon Header File SYSCON	CMSIS core	CMSIS core for Cortex M3 V1.30	
GPIO (with 2 examp	🗆 ВООТ		
	Cmsis Boot	CMSIS Boot for LPC13xx	
	PERIPHERALNXP		
Project 🛛 📃 🗖	Common Header Files	Common Usedas Files fra LDC12 multiment	
- 45 ▽		,	
BB LPC1343 Blinky	SYSCON	LPC13xx System Configuration Driver	
	PMU	LPC13xx Power Management Unit Driver	
	GPIO	LPC13xx General Purpose I\O Driver	
🗄 🗁 Debug	IOCON	LPC13xx I\O Configuration Driver	
⊡ ⊡ lpc13xx_lib	UART	LPC13xx Universal Asynchronous Receiver Transmitt	
syscalls blinky.c	I2C	LPC13xx I2C-bus Interface Driver	
.c blinky.h	WDT	LPC13xx Watch Dog Driver	
			의
link.ld	Manufacturers Chips Peripher	als Drivers Others	-
Sign in to CooCox			

Fig. 18

4. Build the project as shown below image.



Fig. 19



5. Click the Debug button on the IDE and the project is loaded onto the controller and ready for debugging as shown in below image.

CooCox CoIDE		
File Edit View Project	Flash Debug Search Help	
	💌 - <u>- </u>	
🗖 😋 🕱 🗸 🦳 🗖	Repository X 👘 🗖 🕼	Hel 🛛 🗖 🗖
Device [LPC1343]	Start Debug (Ctrl+F5) Step 3 Select Components [NXP / LPC1343] 1	& b ← →
C Library		
CMSIS core		
⊡ · Boot _ [C Library Implement the minimal funct	
	Retarget printf Implementation of printf(), £	
Pr X - D	Semihosting Implementation of Semihost	
	CMSIS core CMSIS core for Cortex M3 V	
E 🔄 ▽ BB_LPC1343_Blinky ▲	🗆 воот	
E cmsis	Cmsis Boot CMSIS Boot for LPC13xx	
🗄 🗁 cmsis_boot	•	
🗄 🗁 Debug 📃 💌		
	Manufacturers Chips Peripherals Drivers Others	_
Sign in to CooCox	BB_LPC1343_Blinky	

Fig. 20

3.2 Creating New project

Follow the below steps, for creating new project:

Step 1: After launching CooCox CoIDE, select manufacturer, for example: NXP as shown in image,



Fig. 21



Step 2: Select chip, for example:LPC1343x301, the right side of CoIDE will display the corresponding information of the chip as shown in below image.



Fig. 22

Step 3: Check the components which you want to use, for example, check GPIO component. CoIDE will promote you to create a new project as shown in below image.



Fig. 23



Step 4: Input the project's name, CoIDE will create a project contained startup code and main.c file for you. The components you selected in third step will be added to your project, too as shown in image.

CooCox CoIDE File Edit View Project	🚺 New Project	
] 📑 🗈 🔛] 🕮 🛎	Denie et	-0
Device []	S Project name: Blinkyl Image: Use default location Location: C:\CooCox\CoIDE\workspace\Blinky	IO: 11xx eral ose
Pr X		river ew are 4 borts port 1 ind up
Sign in to CooCox	Finish Cancel	al se I/O ach

Fig. 24

Step 5: Start writing your code in main as shown in below image.



Fig. 25



Step 6: Build the project as shown below image.

🚺 CooCox CoIDE		
File Edit View Project	Flash Debug Sea	arch Help
] 📑 🗎 🔛 🎬 🎬	🏇 🔐 😐 ,	Ø ▦ ┩ • カ₂ • ヤ + ♥ + ♥ • ♥ •
	Repository 🕅	
Device [LPC1343]	Step 3 selec	t Components [NXP / LPC1343] 1
- • C Library	E COMMON	
GMSIS core Goot	C Library	Implement the minimal funct
	Retarget pr	rintf Implementation of printf(), s
Pr X D	Semihosting	g Implementation of Semihost
	CMSIS core	e CMSIS core for Cortex M3 V
BB_LPC1343_Blinky	🗆 ВООТ	
Er CIS IS Dilliky	Cmsis Boot	CMSIS Boot for LPC13xx
🗄 🗁 cmsis_boot	•	
Ebug	Manufacturers Chi	ps Peripherals Drivers Others
Sign in to CooCox	😕 BB_LPC 1343_Bli	inky

Fig. 26

Step 7: Click the Debug button on the IDE and the project is loaded onto the controller and ready for debugging as shown in below image.



Fig. 27



3.3 Bin File

For bin file follow the below steps:

Step 1: Go to CoIDE workspace (project saved location) Open project folder.

Step 2: Open Debug folder in that two folders are there bin and obj folders.

Step 3:Open bin folder in that bin and hex files are there as shown in below image:



Fig. 28

4.0 BLUEBOARD-LPC1343 Programming

4.1 Programming options

BlueBoard-LPC1343 can be programmed using the

- On-chip bootloader (USB or UART)
- Debugger (NGX CoLink)



4.1.1 On-Chip bootloader (USB or UART)

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

Getting the board in programming mode:

Theory: The On-chip 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 BlueBoard-LPC1343 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
- A Press the RESET button and release it
- ▲ Now release the ISP button
- \checkmark The board is in the programming mode

We know that the on-chip bootloader can be used with USB or UART. Please note that if you have connected a USB cable to the board the USB bootloader is activated else the UART bootloader is activated. Meaning, if you have connected the USB cable as your power source then you cannot use UART bootloader, you need to use an alternate external power source (DC jack) to enable UART bootloader. If a particular MCU supports USB bootloading it is highly recommended to use the same for programming.

Programming through USB is the most convenient way to program the BlueBoard-LPC1343. The LPC1343 has an on-chip USB bootloader support which 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 dragn-drop your binary to this drive.

Note: Not all NXP USB MCUs support USB boot loading. For example the LPC11U14 does not support USB bootloader although it has support for USB on the chip.

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. Press SW4, then SW5; release SW5, then SW4, the mass storage device will appear on your screen. On the board LED D1 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. 29



Flashing the bin file(Drag & drop).



4.2 Flashing the Hex file through UART

Step 1: Connect the serial cable to the PC as well as to the board UART and open the flash magic tool.

Step 2: Input all the parameters as shown in below Fig.

🚓 Flash Magic - NON PRODUCTION USE ONLY				
File ISP Options Tools Help				
🖻 🖬 🔍 🗿 🐗 🖌 📕 🔈 😻 🖳 🚱 😂				
Step 1 - Communi	cations	Step 2 - Erase		
Select Device	LPC1343	Erase block 0 (0x000000-0x000FFF)	*	
COM Port:	COM 13 🔹	Erase block 1 (0x001000-0x001FFF) Erase block 2 (0x002000-0x002FFF)	=	
Baud Rate:	9600 🔹	Erase block 3 (0x003000-0x003FFF) Erase block 4 (0x004000-0x004FFF)		
Interface:	None (ISP) 🔹	Erase block 5 (0x005000-0x005FFF)	-	
Oscillator (MHz):	12	Erase all Flash+Code Rd Prot Image: State of the		
Step 3 - Hex File Hex File: G:\BLUE BOARD_LPC1000_Series\BLUE BOARD_LPC1343\BLL Browse Modified: Thursday, December 8, 2011, 17:37:24				
Step 4 - Options	Step 4 - Options Step 5 - Start			
Verify after programming Start Fill unused Flash Gen block checksums Execute Execute				
Rotating, fully customizable, remotely updated Internet links. Embed them in your application!				
		0		

Fig. 31

Step 3: Click **Start** to flash the hex file. Press Reset switch to run. *NOTE: Make sure that the Board is not powered through USB.*



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



Fig. 32



6.0 Trouble Shooting

For newly created bin file you should update crc.

Unfortunately, the checksum generated is not correct and unless the checksum of the .bin file is modified, the firmware will be rejected by the USB bootloader. Thankfully, this is relatively easy to fix. There is free utility to fix the checksum. A pre-compiled version for Windows (named lpcrc.exe) is also located in the root folder. To fix the checksum, simply go into the command-line and go to the root folder where both the lpcrc.exe tool and your .bin file are located (the file will be named 'firmware.bin' unless you have modified the Makefile), and enter the following command:

Fixing the firmware.bin Checksum

lpcrc firmware.bin

🔤 C:\WINDOWS\system32\cmd.exe			
Directory of C:\Work\ngx\Blueboard_LPC11U24\BLUEBOARD_LPC11U24_Documents\BLUEBO			
ARD_LPC11U2	4_Documents	\Blue Board_LP	C11U24\Source_programs\BLUEBOARD_LPC11U24_KE
IL_with_GLC	D\Blinky\Ke	ilworkspace\Ob,	j
10,000,00014	OF-E4 DM	(510)	
12/09/2011 12/09/2011		<dir> <dir></dir></dir>	•
12/09/2011			12 Blinky.axf
12/09/2011			00 Blinky.bin
12/08/2011			38 blinky.crf
12/08/2011			66 blinky.d
12/09/2011			17 Blinky.hex
12/09/2011			75 Blinky.htm
12/09/2011		3	39 Blinky.lnp
12/08/2011		33,6	24 blinky.o
12/09/2011			72 Blinky.plg
10/10/2011			79 Blinky.sct
12/09/2011			89 Blinky.tra
11/09/2011			19 ExtDll.iex
03/22/2011 12/08/2011			42 lpcrc.exe
12/08/2011			44 startup_LPC11Uxx.o
12/08/2011	12.47 FN 19.40 DM	34,4	58 system_lpc11uxx.crf
12/08/2011	12-49 PM	27 2	56 system_lpc11uxx.d 40 system_lpc11uxx.o 270 butcm_lpc11uxx.o
12/00/2011	17 File(s) 228	070 bytes
	2 Dir(s		904 bytes free
			61 <i>x</i> 3000 1100
C:\Work\ngx\Blueboard_LPC11U24\BLUEBOARD_LPC11U24_Documents\BLUEBOARD_LPC11U24_D			
ocuments\Blue_Board_LPC11U24\Source_programs\BLUEBOARD_LPC11U24_KEIL_with_GLCD\B			
linky\Keilworkspace\Obj>lpcrc Blinky.bin			
succesfully updated crc to: effff9b3			

Fig. 33



7.0 CHANGE HISTORY

7.1 Change History

Rev	Changes	Date (dd/mm/yy)	By
1.0	Initial release of the manual	11/07/2012	Dharmanna Gaded

8.0 REFERENCES

In addition to this document, the following references are included on the NGX BLUEBOARD-LPC1343 product and can also be downloaded from <u>www.ngxtechnologies.com</u>:

• NGX BLUEBOARD-LPC1343 schematic for the Development board.

Additional references include:

- Information on development tool being used:
 - CoIDE 1.4.2, <u>http://www.coocox.org/CooCox_CoIDE.htm</u>
 - Flash magic, <u>http://www.flashmagictool.com/</u>

About this document:

Revision History

Version: V1.0 author: Dharmanna Gaded

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 times and without notice in abding without limitation enables and market descriptions.

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.