LPC2148 DEVELOPMENT BOARD





# **LPC2148 DEVELOPMENT BOARD**





# NOTICE

The contents of this manual are subject to change without notice. All efforts have been made to ensure the accuracy of contents in this manual. However, should any errors be detected, NEX Robotics welcomes your corrections. You can send us your queries / suggestions at

info@nex-robotics.com



Content of this manual is released under the Creative Commence cc by-nc-sa license. For legal information refer to: <u>http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode</u>



- Product's electronics is static sensitive. Use the product in static free Environment.
- Read the user manual completely before start using this product



#### **Recycling:**

Almost all the part of this product is recyclable. Please send this product to the recycling plant after its operational life. By recycling we can contribute to cleaner and healthier environment for the future generations.



# INDEX

1.0 INTRODUCTION	4
1.1 Board Features	4
1.2 LPC2148 Pro Development Board Overview	6
1.3 Power supply	7
1.4 In System Programming via UART0	8
2.0 KEIL µVISION4 IDE	9
2.1 Installing Keil uVision4 IDE	9
2.2 Using Flash Magic Tool	.12
2.2.1 Installing Flash Magic Tool	.12
2.3 Overview of Keil uVision4 IDE	.17
2.3.1 Create a Project in Keil for LPC2148 development board	.18
2.3.2 CODE WALKTHROUGH	.29
2.3.3 Configuring Startup.s File using Configuration wizard	.34
2.4 Using uVision Debugger in simulator mode	.37
3.1 Main Expansion Header	.44
3.2 LEDs	.45
3.3 USER SWITCHES	.45
3.4 LCD INTERFACE	.46
3.5 BUZZER	.46
3.6 IR RECEIVER	.46
3.7 TRIMPOTS	.47
3.8 UART1/XBEE/BLUETOOTH/WIFI	.47
3.9 SPI INTERFACE	.47
3.10 I2C INTERFACE	.48
3.11 ULN 2003	.48
3.12 L293D	.49
3.13 JTAG PORT	.50
3.14 RTC	50
4.0 SAMPLE PROGRAMS	.51
4.1 LED CHASER	.51
4.2 IO INTERFACING	.51
4.3 BUZZER	.52
4.4 LCD INTERFACING	.52
4.5 UART0 COMM 9600	.53
4.6 UART1 COMM 9600	.53
4.7 ADC	.54
4.8 I2C EEPROM	.54
4.9 SD MMC INTERFACE	.55
4.10 DC MOTOR CONTROL	.55
4.11 STEPPER MOTOR CONTROL	.56
4.12 RTC UART0	.56
4.13 XBEE/BLUETOOTH/WIFI WIRELESS COMMUNICATION	.57
4.14 BOARD DEMO	.58



LPC2148 DEVELOPMENT BOARD

# **1.0 INTRODUCTION**



### **1.1 Board Features**

LPC2148 Pro Development Board is a powerful development platform based on LPC2148 ARM7TDMI microcontroller with 512K on-chip memory. This board is powered by USB port and does not need external power supply. It is ideal for developing embedded applications involving high speed wireless communication (Zigbee / Bluetooth / WiFi), USB based data logging, real time data monitoring and control, interactive control panels etc. The on-chip USB controller provides direct high speed interface to a PC/laptop with speeds up to 12Mb/s. The UART boot loader eliminates need of an additional programmer and allows you to program using serial port. The on board peripherals include SD/MMC card interface, USB2.0 interface, 4Kbit I2C EEPROM, Xbee wireless module interface, ULN2003 500mA current sinking driver, L293D DC motor controller, 16X2 character LCD and many more. The on-chip peripherals and the external hardware on the development board are interconnected using pin headers and jumpers. The I/O pins on the microcontroller can be accessed from a 50 pin male header. The board is made from double sided PTH PCB board to provide extra strength to the connector joints for increased reliability.

#### Note:

1. Xbee / Bluetooth/ WiFi wireless module and SD/MMC card are not included with the board and it can be bought separately from Nex Robotics website.



#### **Specifications:**

- Microcontroller: LPC2148 with 512K on chip memory
- Crystal for LPC2148: 12Mhz
- Crystal for RTC: 32.768KHz
- 50 pin Berg header for external interfacing
- Wireless module adapter for 2.4GHz ZigBee (Xbee) / Bluetooth / WiFi connectivity
- 512 bytes of I2C external EEPROM
- USB Type B Connector
- SD / MMC card holder
- 10pin(2X5) FRC JTAG connector for flashing and debugging
- 50 Pin Expansion header
- 2x16 Character Alphanumeric LCD

- L293D 600mA Dual DC motor Driver
- ULN2003 500mA driver
- Dual RS232 UARTs for external communication
- Real-Time Clock with Battery Holder
- 2 Analog Potentiometers connected to ADC
- TSOP1738 IR receiver
- 4 USER Switches
- 4 USER LEDs
- Reset and Boot loader switches
- 3V button cell for on chip RTC
- ON/OFF switch
- Buzzer
- Schematics and Application examples in KEIL provided in the documentation CD

#### **Kit Contains:**

1-LPC2148 Pro Development Board 1-USB Cable 1-DB9 Serial Cable A set of 10 1-1 cables Documentation CD contains: I. Schematic II. Programming Software III. Example Codes

#### LPC2148 Features:

- 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
- 40 kB of on-chip static RAM and 512 kB of on-chip flash memory.
- In-System Programming/In-Application Programming (ISP/IAP) via on-chip boot loader software.
- Embedded ICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip Real Monitor software and high-speed tracing of instruction execution.
- USB 2.0 Full-speed compliant device controller with 2 kB of endpoint RAM.
- Two 10-bit ADCs provide a total of 14 analog inputs
- Single 10-bit DAC provides variable analog output
- Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
- Low power Real-Time Clock (RTC) with independent power and 32 kHz clock input.



- Multiple serial interfaces including two UARTs, two Fast I<sup>2</sup>C-bus (400 kbit/s), SPI and SSP with buffering and variable data length capabilities.
- Vectored Interrupt Controller (VIC) with configurable priorities and vector addresses.
- 60 MHz maximum CPU clock available from programmable on-chip PLL with settling time of 100 us.
- On-chip integrated oscillator operates with an external crystal from 1 MHz to 25 MHz
- Power saving modes include Idle and Power-down.
- Individual enable/disable of peripheral functions as well as peripheral clock scaling for additional power optimization.
- Processor wake-up from Power-down mode via external interrupt or BOD.
- Single power supply chip with POR and BOD circuit
- CPU operating voltage range of 3.0 V to 3.6 V (3.3 V +- 10 pct) with 5 V tolerant I/O pads.



# **1.2 LPC2148 Pro Development Board Overview**

- 1. LPC2148 Plug-in module
- 2. 3V cell holder for RTC
- 3. UART1 DB9 connector
- 4. UART0 DB9 connector
- 5. 50-pin expansion header
- 6. 2X5 JTAG header
- 7. ON/OFF slide switch
- 8. Boot loader switch
- 9. Microcontroller reset switch
- 10. USB connector B-type

- 11. Jumpers for LCD interface
- 12. Jumpers for Switches
- 13. Jumpers for Trim pots
- 14. Jumpers for Buzzer and IR Receiver
- 15. Jumpers for LEDs
- 16. Jumpers for selection between UART1 and Xbee
- 17. Jumpers for SPI SD/MMC interface

NEX Robotics Pvt. Ltd. www.nex-robotics.com



18. Jumpers for I2C EEPROM
 19. SD/MMC card socket

20. 16X2 character LCD

- 21. Four user switches
- 22. Two trim pots connected to ADC
- 23. Four user LEDs



- 24. Buzzer
- 25. TSOP1738 IR Receiver
- 26. Xbee module interface

- 27. ULN2003 driver side header
- 28. L293D o/p header
- 29. Jumpers for ULN2003



The USB powered version of LPC2148 development board is powered using USB port.

Slide the power switch to ON position to turn the board ON. The red LED will glow indicating that the board is powered.

**Warning:** Please make sure that the USB port on your PC or any other device is capable to source at least 400mA current.



# 1.4 In System Programming via UART0

LPC2148 microcontroller incorporates a capability to self program itself using the on-chip UART boot loader. This eliminates the need of an external programming hardware. The boot loader code is dedicated to use UART0 on the microcontroller. Once the code is loaded on the microcontroller UART0 is free to be used in the application. The host, in this case PC or Laptop, requires software to transmit the hex file to the microcontroller over the serial link. For demonstration purpose we will use Flash Magic programming utility. The installation procedure is explained in section 2.2.1.



To enter in to the boot load mode follow the steps mentioned below:

1. Connect a DB9 cable between the selected **COM** port on your PC and the **UART0** port on LPC2148 development board.

2. Plug in an USB cable to the development board and slide power switch to ON position.

3. The board enters bootloader mode when **BOOT** switch is held down at the time of **RESET**. To enter the bootloader mode, keep the **BOOT** switch pressed and then press and release **RESET** switch. You can release the **BOOT** switch after reset.



# 2.0 KEIL μVISION4 IDE

The LPC2148 microcontroller is supported by various commercially available IDEs for compiling and debugging of the code. Keil being one of them is the widely used IDE for LPC family of microcontrollers. The  $\mu$ Vision4 IDE is Windows-based software development platforms that combines a robust editor, project manager, and make facility.  $\mu$ Vision4 integrates all tools including the C compiler, macro assembler, linker/locator, and HEX file generator. The evaluation version of Keil  $\mu$ Vision4 IDE is used for demonstrating the sample codes that are included in the CD.

The open source community has been doing a lot in the development of open source tools for ARM architecture based microcontrollers. The open source tools are available at zero cost and are being improved with time. Eclipse being one of them and is most commonly used IDE due to its unique features like auto complete, project tree, etc. It requires GCC tool chain for code compilation.

# 2.1 Installing Keil µVision4 IDE

To install Keil  $\mu$ Vision4 IDE, Go to "Software" folder in the documentation CD and locate "mdk412.exe" file. Click on "mdk412.exe" to start the installation process. Once the installation process is started Keil  $\mu$ Vision welcome screen will appear.

Please read the instructions on the welcome window and click Next>> to start the installation.

Welcome to Keil μVision Release 6/2010	
This SETUP program installs:	
MDK-ARM V4.12	
This SETUP program may be used to update a pr However, you should make a backup copy befor	revious product installation. e proceeding.
It is recommended that you exit all Windows prog	rams before continuing with SETUP.
Follow the instructions to complete the product in	stallation.
Keil µVision4 Setup	
	KK Back Next >> Cancel

Please read the license agreement carefully. If it is acceptable click the check box and click **Next>>** to continue.



To continue with SETUP, you must accent the terms of the Licens	
agreement, click the check box below.	e Agreement. To accept the
End-User License Agreement for AF Development Tools	RM Keil Software
THIS END USER LICENCE AGREEMENT ("LICENCE") BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR S ARM LIMITED ("ARM") FOR THE USE OF THE SOFTW, LICENCE. ARM IS ONLY WILLING TO LICENSE TH CONDITION THAT YOU ACCEPT ALL OF THE TERM CLICKING "I AGREE" OF BY INSTALLING OF OTHE	IS A LEGAL AGREEMENT SINGLE LEGAL ENTITY) AND ARE ACCOMPANYING THIS E SOFTWARE TO YOU ON MS IN THIS LICENCE. BY
I agree to all the terms of the preceding License Agreement	

Select the destination folder where setup will install files. It is always recommended to select the default location. To create backup of old installation select the backup option and click **Next>>** to continue.

Select the folder where SETLIP will install files	KEIL
SETUP will install µVision4 in the following folder.	
To install to this folder, press 'Next'. To install to a different folder, pre folder. Destination Folder	ss 'Browse' and select another
C:\Keil	Browse
Update Installation: Create backup tool folder	



In the next window enter your information and click Next>> to continue.

Customer Informal	tion	KEI	Th
Please enter your	information.	Tools by AF	RM
Please enter your r	name, the name of the company for w	hom you work and your E-mail address.	
First Name:	Embedded Systems Group		
Last Name:	Nex		
Last Name: Company Name:	Nex Nex Robotics		
Last Name: Company Name: E-mail:	Nex Nex Robotics info@nex-robotics.com		

On clicking next the file copying process will begin. Wait till setup is complete.

Setup MDK-ARM V4.12	
Setup Status	
$\mu Vision$ Setup is performing the requested operations.	
Install Files	
Installing Clock.ini.	
•	
— Keil μVision4 Setup	Cancel



Click Finish to complete installation process.

ully.
<u>×</u>
1

# 2.2 Using Flash Magic Tool

The LPC series of microcontrollers are preloaded with the boot loader firmware which allows self programming of microcontrollers using serial port. Flash magic is a utility which provides an interface for reading, writing and verifying the flash memory of the microcontroller.

# 2.2.1 Installing Flash Magic Tool

To install **Flash Magic**, Go to "Software" folder in the documentation CD and locate "**FlashMagic.exe**" file. Click on "**FlashMagic.exe**" to start the installation process. Once the installation process is started Flash Magic welcome screen will appear.





Please read the license agreement carefully. If it is acceptable click the radio button and click **Next>>** to continue.

🕼 Setup - Flash Magic	
License Agreement Please read the following important information before continuing.	
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.	
FLASH MAGIC LICENSE EMBEDDED SYSTEMS ACADEMY, INC.	
You should carefully read the following terms and conditions before using this software. Unless you have a different license agreement signed by Embedded Systems Academy, Inc. ("ESA") your use, distribution, or installation of this copy of Flash Magic indicates your acceptance of this license.	
If you do not agree to any of the terms of this License, then do not install, distribute or use this copy of Flash Magic.	~
<ul> <li>I accept the agreement</li> <li>I do not accept the agreement</li> </ul>	
Embedded Systems Academy, Inc.	Cancel

Select the destination folder and click **Next>** to continue.

🔞 Setup - Flash Magic	
Select Destination Location Where should Flash Magic be installed?	
Setup will install Flash Magic into the following folder.	older olick Browse
C:\Program Files\Flash Magic	Browse
At least 11.6 MB of free disk space is required.	
Embedded Systems Academy, Inc. — — — — — — — — — — — — — — — — — — —	Next > Cancel

In the next window choose the appropriate folder and click **Next>** to continue.



🕼 Setup - Flash Magic	
Select Start Menu Folder Where should Setup place the program's shortcuts?	
Setup will create the program's shortcuts in the follo To continue, click Next. If you would like to select a different	wing Start Menu folder. folder, click Browse.
(Flash Magic	Browse
Embedded Systems Academy, Inc. —	
Kack	Next > Cancel

Select the desired option and click **Next>** to continue.

🔞 Setup - Flash Magic	
Select Additional Tasks Which additional tasks should be performed?	
Select the additional tasks you would like Setup to perform while install then click Next. Additional icons: Create a <u>d</u> esktop icon Create a <u>Quick Launch icon</u>	ing Flash Magic,
Embeddad Systems Academy, Inc	> Cancel

Setup is now ready to begin installing flash magic. Click Install to continue.



1층 Setup - Flash Magic	
<b>Ready to Install</b> Setup is now ready to begin installing Flash Magic 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:\Program Files\Flash Magic Start Menu folder: Flash Magic Additional tasks: Additional icons: Create a desktop icon Create a Quick Launch icon	2
	<u>s</u>
	Cancel

Wait till setup installs Flash Magic on your computer.

Setup - Flash Magic	
Installing	
Please wait while Setup installs Flash Magic on you	ir computer.
Extracting files	
C:\Program Files\Flash Magic\Drivers\Pcandrv.exe	•
shaddad Sustams Acadamu Inc.	
ibedded Systems Academy, Inc. ———————	



Click Finish to complete the installation process





# 2.3 Overview of Keil µVision4 IDE

To start Keil IDE click Start>Programs>Keil µVision4. The initial screen will appear followed by the main window.

🕎 µVision4		E B 🛛
File Edit View Project I	Flash Debug Peripherals Tools SVCS Window Help	
🗋 💕 🖬 🖉   X 🖦 I	8. ウ ♡ / - →   巻 象 教   準 準 准 服   ❷	
0 I I 0 0 I   X	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	
Project 🧭 🗴	Teeti 🛛	
PROJECT	EDITOR	2
Build Output		
	OUTPUT	2 2
For Help, press F1		L1 C1 CAP NUM SCRU OVR R/W

The Keil IDE main window in basic configuration is mainly divided into three areas.

Editor - It is the area where .c and .h files of the project are edited.Project Explorer- It shows the project tree.Output Window- This window shows the messages related to compiling, project building and debugging.



# 2.3.1 Create a Project in Keil for LPC2148 development board

1. To create a new project, Select **Project>New uVision Project** from the main menu.



2. Create a new directory and name it as **First\_Project**. Click open to enter in to this directory.

Create New Pro	oject				? 🛛
Save in:	🗇 Local Disk (C	2)	•	- •	•
My Recent Documents	Cocuments an Keil Program Files WINDOWS First_Project	d Settings			
My Documents					
My Computer					
My Network Places	File name: Save as type:	Project Files (*.uvp	roj)	• •	Open Cancel



3. Inside this directory create a new project and name it as **First\_Project** and click **Save** to continue.

Create New Pro	oject				? 🔀
Save in:	🔁 First_Projec	t		🕈 🗈 💣 🗉	•
My Recent Documents					
My Documents					
My Computer					
My Network	File <u>n</u> ame:			•	( <u>S</u> ave
Flaces	Save as type:	Project Files (*.)	uvproj)	•	Cancel

4. In the next window locate NXP (founded by Philips) tree and expand it.

CPU	- 1 A
Vendor: Actel Device: Toolset:	
Data base Description: Energy Micro Freescale Semiconductor Fujitsu Microelectronics Control Navoton Navoton NAXP (founded by Philips) EM773FHN33 Control LH75400 Control LH75410 Control LH75411 Control LH7541 Control LH75411 Control LH7541 Co	
□ H79520 □ H79524 □ H79524	<u>×</u>



5. Now select target device as LPC2148 and click OK to continue.

Select Device for Target 'Target 1'.		×
CPU Vendor: NXP (founded by Philips) Device: LPC2148 Toolset: ARM	Description	
LPC2132/01     LPC2134     LPC2134     LPC2134     LPC2134     LPC2136     LPC2136     LPC2138     LPC2138     LPC2138     LPC2141     LPC2142     LPC2144     LPC2144     LPC2148	ARM7TDMI-S based high-performance 32-bit RISC Microcontroller with TF 512KB on-chip Flash ROM with In-System Programming (ISP) and In-Applix Two 10bit ADCs with 14 channels, USB 2.0 Full Speed Device Controller Two UARTs, one with full modem interface. Two 32-bit timers, Watchdog Timer, PWM unit, Real Time Clock with optional battery backup, Brown out detect circuit General purpose I/O pins. CPU clock up to 60 MHz, On-chip crystal oscillator and On-chip PLL	The second secon
LPC2194	OK Cancel Help	

6. Click **Yes** to copy **Startup.s** file to project folder. This file configures stack, PLL and maps memory as per the configurations in the wizard. It is discussed in the later sections.

µVision			×
?	Copy Philips LPC2100 Startup Co	de to Project Folder and Add F	ile to Project ?
	Yes	No	



7. Observe the project explorer area in the main window.



8. Now click **Project>Manage>Components**, Environments, Books from the main menu to ensure compiler settings.

🕲 First_Project - µ\	/ision4
File Edit View Pro	oject Flash Debug Peripherals Tools SVCS Window Help
	New µVision Project k in the second secon
Project	Close Project
🖃 🚵 Target 1 -	Export
1 Start	Manage Components, Environment, Books
	Select Device for Target 'Target 1'
	Remove Item
Å	Options for Target 'Target 1' Alt+F7
	Clean target
12	Build target F7
(##	Rebuild all target files
e 1997	Batch Build
	Translate, Ctrl+F7
	Stop build



9. In the Folders/Extensions tab ensure the compiler settings are as shown in the fig. below. If you have installed keil software at a different location then change Tool Base Folder location. Click OK to continue.

roject Components	Folders/Extensions Books			
Development Tool I	Folders		Default File Ext	ensions:
🗖 Use Settings fro	m TOOLS INI:		C Source:	*.c
Tool Base Fo	Ider C:\Keil\ARM\		C++ Source:	*.cpp
BIN: CONKe	IVARM\BIN\		Asm Source:	*.s*; *.src; *.a*
INC:			Object:	*.obj
LIB:			Library:	*.lib
Regfile:			Document:	*.txt; *.h; *.inc
Use RealView Compiler Use GNU Compiler	RealView Folder: BIN40 GNU-Tool-Prefix: arm-hone-eabi- GNU-Tool Folder: C.\Program Files	NCodeSourcery\So	urcety G++ Lite\	
		Contral		Liale

10. Now click **File>New** to create a new file.



 $\setminus$ 



11. Save the new file in to the same folder that was created earlier and name it as **main.c** and click **Save** to continue.

Save As		? 🛛
Save in: My Recent Documents Desktop My Documents My Computer	First_Project First_Project First_Project First_Project_Target 1.dep Startup.s	
My Network Places	File name: main.cl  Save as type: All Files (*.*)	Save Cancel

12. Now add "**main.c**" to the source group by right clicking on the **Source Group 1** from the project explorer and select the highlighted option as shown in the fig. below.





13. Select "main.c" file to be added and click ADD to continue.

Add Files to	Group 'Source Group 1'			? 🛛
Look in: 🙆	First_Project	•	+ E d	• 📰 <del>-</del>
main				
			-	
File name:	main		— (C	Add
Files of type:	C Source file (*.c)		- T	Close

14. Observe that "main.c" file is added to the source group in the project explorer window.

	B ゥ ┍ ← → 作 除 除 限 律 律 ///// /// ///////////////////	
🖉 🖾 🕮 🥔 拱 🙀	Target 1 🛛 🗹 🎊 📥 🖷	
Project 😗 🖸	a main.c 🛛	
Startup.s	01 #10-1002 (190214X.10) 02 03 #define LED1 ON() IO1SET=(1<<16) 04 #define LED2_ON() IO1SET=(1<<17) 05 #define LED3 ON() IO1SET=(1<<18) Group 'Source Group 1'	
Look jn: 🔯 F	irst_Project 🔽 🕂 🖽 -	
🗊 main		



15. Right click **Target1** in the project explorer window and select the highlighted option as shown in the fig. below.

🖫 First_Project - µVision4	
File Edit View Project Flash Debug Peripherals T	Tools SVCS Window Help
9 (+ + ) ° (2 (2 (2 (2 (2 (2 (2 (2 (2 (2 (2 (2 (2	🎗 🖪 🐧 🛊 連 🎼 🎼 🙆
🗇 🕮 🥔 🔜 🙀 Target 1 🛛 🛛 🖌	8 🛔 👫
Project 🥑 🗵 🚺 main.c 🛛	
Options for Target 'Target 1' Alt+F7 Open File Open List File Open .\First_Project.Map Rebuild all target files Build target F7 Translate File Stop build Add Group Add Files to Group Remove Item	<pre>&gt;() IOISEI=(1&lt;&lt;16) '() IOICLR=(1&lt;&lt;17) '() IOICLR=(1&lt;&lt;18) '() IOICLR=(1&lt;&lt;19) .gned char j) .; )00; i++);</pre>

16. In the appearing window select Target tab and set Xtal. frequency as 12MHz.

XP Itor.	nded by P	hilins)   PC214	3						
(			( IT	-	Code C	ieneration	(		
			Xtal (MHz):	20	ARM	l-Mode	•		
Operat	ing system	None		•	ΓU	se Cross-N	Aodule Optimiza	tion	
					ΓU	se MicroL	IB L	T Big Endian	
					ΓU	se Link-Ti	me Code Gener	ation	
Read/	Only Mem	ory Areas			-Read/	Write Men	nory Areas		
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	Nolnit
Г	ROM1:			C	Г	BAM1:			Г
Г	ROM2:			C	Г	RAM2:		<b></b>	
П	ROM3:		1	- C	Г	RAM3:		-	Г
	on-chip	,				on-chip		,	
	IROM1:	0x0	0x80000	•		IRAM1:	0x40000000	0x8000	F
-	IBOM2			- c	Г	IRAM2:			- Γ



17. In the **Output** tab ensure that **Create HEX File** option is selected.

Options for Target 'Target 1'	
Device   Targe Output Listing   User   C/C++   Asm   Linker   De	ebug   Utilities
Select Folder for Objects Name of Executable:	First_Project
Create Executable: .\First_Project     Debug Information     Create HEX File     F Browse Information     Create Library: .\First_Project.LIB	T Create Batch File
OK Cancel	Defaults Help

18. In the **Linker** tab ensure that the highlighted option is selected and click **OK** to continue.

Options for T	arget 'Target 1'	-		
Device Targ	et   Output   Listing   User   C/C++   Asm	Linker Debug	Utilities	
🔽 Use Men	nory Layout from Target Dialog			
☐ Make ☐ Make ☐ Don'i	e RW Sections Position Independent e RO Sections Position Independent - Search Standard Libraries	R/O Base: R/W Base	0x00000000 0x40000000	
Repo	rt 'might fail' Conditions as Errors	disable Warnings:		
Scatter File Misc				Edit
controls				
Linker control string	cpu ARM7TDMI *.ostrictscatter "First_F autoatsummary_stderrinfo summarysizes	<sup>p</sup> roject.sct'' smapxrefcallgrap	hsymbols	<u>^</u>



19. Now since the project is almost setup we can start writing code in the "**main.c**" file that was created earlier. For demonstration purpose you can copy the following code and paste it in the "**main.c**" file. This code is written to blink LEDs on LPC2148 development board in the sequential order starting from LED1. Refer jumper settings for LEDs from the hardware description section 3.2.

```
#include <lpc214x.h>
#define LED1 ON() IO1SET=(1<<16)</pre>
#define LED2 ON() IO1SET=(1<<17)</pre>
#define LED3 ON() IO1SET=(1<<18)</pre>
#define LED4 ON() IO1SET=(1<<19)</pre>
#define LED1_OFF() IO1CLR=(1<<16)</pre>
#define LED2_OFF() IO1CLR=(1<<17)
#define LED3_OFF() IO1CLR=(1<<18)
#define LED4_OFF() IO1CLR=(1<<19)</pre>
void Delay(unsigned char j)
{
 unsigned int i;
 for(;j>0;j--)
 {
  for(i=0; i<60000; i++);</pre>
 }
}
int main (void)
{
 PINSELO = 0 \times 00000000;
                                  // Enable GPIO on all pins
 PINSEL1 = 0 \times 00000000;
 IO1DIR = (1<<19) | (1<<18) | (1<<17) | (1<<16); // Set P1.16, P1.17,
P1.18, P1.19 as Output
 while(1)
 {
  LED1 ON();
  Delay(25);
  LED1_OFF();
LED2_ON();
  Delay(25);
  LED2 OFF();
  LED3 ON();
  Delay(25);
  LED3_OFF();
  LED4 ON();
  Delay(25);
  LED4 OFF();
 }
 return(0);
}
```



20. Now build the project by clicking on **Rebuild** button on the main toolbar.

🛚 First_Project - µVision	4
File Edit View Project	Flash Debug Peripherals Tools SVCS Windo
	非計 図 おお ひ ( ~ ~ ) ( ~ ) ( 2
🕸 🕮 🖉 🖳 🙀	Target 1 🛛 😒 🌋 🗂 📇
Project Rebuild	main.c 🛛
🖃 🛅 Target 🛛 Rebuild all t	arget files #include <1pc214x.h>
📄 🔄 Source Group 1	02
Startup.s	03 #define LED1_ON() IO1SET=(1<-
🔚 main.c	04 #define LED2_ON() IO1SET=(1<-
	05 #define LED3_ON() IO1SET=(1<-
	06 #define LED4 ON() IO1SET=(1<-
	07
	08 #define LED1 OFF() IO1CLR=(1-
	09 #define LED2_OFF() IO1CLR=(1-

21. You can alternatively build project by clicking on **Project>Build Target** from the main menu.

🕱 First_Project	- µVision4				
File Edit View	Project Flas	n Debug Per	ipherals Tools	SVC	S Window Help
□ 🚰 🖬 🗭 ⊗ 🖾 🖽 🧼 Project	New µVis New Mul Open Pro	ion Project ti-Project Worksp ject iect	bace		谭 谭 //ミ //泉 🙆
Target 1	Export Manage	Sectorie			T= (1<<16) T= (1<<17)
	Select De Remove I	vice for Target 'Ta tem	arget 1'		:T= (1<<18) :T= (1<<19)
	Clean tar	or Target 'Target	1' /	Alt+F7	LR=(1<<16) LR=(1<<17)
	Build tar	jet		F7	LR= (1<<19)
	<ul> <li>Rebuild a</li> <li>Batch Bu</li> </ul>	II target files			ar j)
	Stop buil	C:\First_Project\r d	main.c C	trl+F7	
	1 C\First	Project\First Pro	iect.uvoroi		-);



22. You can observe the build process in the output window. If any errors, rectify it by double clicking on it and you will be pointed to the erroneous line.



# 2.3.2 CODE WALKTHROUGH

#### #include <lpc214x.h>

The first line of code includes the header file for LPC2148 microcontroller. These file contains the definitions of special function registers related to this device. This file can be found in Keil uVision 4 install directory.

```
#define LED1_ON() IO1SET=(1<<16)
#define LED2_ON() IO1SET=(1<<17)
#define LED3_ON() IO1SET=(1<<18)
#define LED4_ON() IO1SET=(1<<19)</pre>
```

The above 4 lines define the macro functions for turning ON LEDs.

For e.g. when LED1\_ON(); is called, it will execute IO1SET=(1<<16). This will set 16 th bit on PORT1

```
#define LED1_OFF() IO1CLR=(1<<16)
#define LED2_OFF() IO1CLR=(1<<17)
#define LED3_OFF() IO1CLR=(1<<18)
#define LED4_OFF() IO1CLR=(1<<19)</pre>
```

The above 4 macro functions are used to turn OFF respective LED. Please refer LPC2148 user manual for more information on GPIO.

```
void Delay(unsigned char j)
{
    unsigned int i;
    for(;j>0;j--)
    {
        for(i=0; i<60000; i++);
    }
}</pre>
```

The **Delay(unsigned char j)** function is used to provide some finite delay between Led ON and OFF events.

```
NE®ROBOTICS
```

```
int main (void)
PINSEL0 = 0 \times 00000000;
                           // Enable GPIO on all pins
PINSEL1 = 0x0000000;
 IO1DIR = (1<<19) | (1<<18) | (1<<17) | (1<<16); // Set P1.16, P1.17,
P1.18, P1.19 as Output
 while(1)
 {
 LED1 ON();
 Delay(25);
 LED1 OFF();
 LED2 ON();
 Delay(25);
 LED2 OFF();
 LED3 ON();
 Delay(25);
 LED3 OFF();
 LED4 ON();
 Delay(25);
 LED4 OFF();
 }
return(0);
}
```

It is the main function of the code. The first and second line will initialize GPIO function on all PORT0 pins.

```
IO1DIR = (1<<19) | (1<<18) | (1<<17) | (1<<16);
The above statement will make port1 pins P1.16, P1.17, P1.18, P1.19 as output.
```

while (1) statement will continuously execute the code inside its loop.

The first line LED1\_ON(); will turn on LED1 and it will remain ON till delay function is executed. The third line will turn OFF LED1 and turn ON LED2 and it will remain ON till delay function is executed. Similarly as the code progresses the LEDs will turn ON & OFF and it will appear as if LEDs are chasing each other.

The next step is to load the hex file in to the microcontroller using Flash magic utility.

1. Click Start>Programs>Flash Magic>Flash Magic to start Flash Magic utility.

2. Click Select Device to select LPC2148 microcontroller.



🌸 Flash Magic	- NON PRODUCTION US	E ONLY
File ISP Option:	s Tools Help	
۵ 🗖 🍳	🎸 🗸 📕 💊 🛛	
Step 1 - Communi	cations	Step 2 - Erase
Select Device	89C51RA2xx	Erase block 0 (0x0000-0x0FFF)
COM Port:	~	
Baud Rate:	9600 💌	
Interface:	None (ISP)	Caracity Caracity Clic
Oscillator (MHz):		Erase all Flash+Security+Ciks
Step 3 - Hex File		
Hex File:		Browse
Modified	t Unknown	more info
Step 4 - Options		Step 5 - Start!
Verify after prog	ramming Set Security Bit	Start
Gen block chec	h 🔄 Set Security Bit 7 ksums 🔲 Set Security Bit 7	
Execute	Prog Clocks Bit	
Technical on-line	articles about 8051 and XA pro	gramming
www.esacademy.e	com/fag/docs	
		0

3. Expand the ARM7 tree and select LPC2148 from the displayed list of microcontrollers.





4. After selecting the microcontroller, select the COM port that you are going to use and ensure that other settings are as shown in the figure below.

🏶 Flash Magic	- NON PRODUCTION US	SE ONLY	
File ISP Option	s Tools Help		
۵ 🗔 🔍 🕲	i 🍏 🗸 🎩 💊 🛛 🕺	國 🔞 😂	
Step 1 - Communi	cations	Step 2 - Erase	
Select Device	LPC2148	Erase block 0 (0x000000-0	x000FFF)
COM Port:	СОМ 1 🛛 🖌	Erase block 2 (0x002000-0	xoolfff) == )xoo2FFF)
Baud Rate:	9600 💌	Erase block 3 (0x003000-0 Erase block 4 (0x004000-0	0x003FFF) 0x004FFF1
Interface:	None (ISP)	Erase block 5 (0x005000-0	0x005FFF) 🕑
Oscillator (MHz):	12	Erase all Elash+Code Br	1 Prot
Hex File: Modified	t Unknown	more info	Browse
Step 4 - Options		Step 5 - Startl	
Verify after prog Fill unused Flas Gen block chec Execute	ramming) h ksums	St	art
Visit the "Flash Ma	agic" home page for info on th com/software/flashmagic	e latest revision	6
		0	

5. To select the hex file click **Browse** and point to the folder that was created earlier and select the hex file.

🌸 Flash Magic	- NON PRODUCTION US	E ONLY	
<u>File ISP Option</u>	s <u>T</u> ools <u>H</u> elp		
🖻 🗖 🔍 🗿	I 🍪 🗸 🎩 🔈 😽 I	A 🕄 🔁	
Step 1 - Communi	cations	Step 2 - Erase	
Select Device	LPC2148	Erase block 0 (0x000000-1	0x000FFF)
COM Port:	СОМ 1 🛛 🖌	Erase block 2 (0x002000-)	Dx001FFF) 🔤
Baud Rate:	9600 💌	Erase block 3 (0x003000-0 Erase block 4 (0x004000-0	0x003FFF) 0x004FFF)
Interface:	None (ISP)	Erase block 5 (0x005000-0	0x005FFF)
Oscillator (MHz):	12	Erase all Flash+Code River Erase blocks used by H	d Prot lex File
Hex File: C:\First Modified	_Project\First_Project.hex I: Saturday, August 14, 2010, 7	2:14:13 PM more info	Browse
Step 4 · Uptions		Step 5 - Start	
Verify after prog Fill unused Flast Gen block cheo Execute	ramming h :ksums	St	art
Visit the "Flash Ma	agic" home page for info on the	a latest revision	
www.esacademu	com/software/flashmanic		1
	sent centralen ndon magio	10 million (1997)	



6. Before loading the hex file prepare LPC2148 development board by doing the following hardware setup.

i. Connect a DB9 cable between the selected **COMx** port on your PC and the **UART0** port on LPC2148 development board.

ii. Plug in USB cable to the development board and slide ON/OFF switch to ON position.iii. Enter in to boot load mode by keeping the BOOT switch pressed and then press RESET switch, release RESET switch and then release BOOT switch.

7. After ensuring the hardware setup in step 6 click start to begin programming the hex file.

🌸 Flash Magic	- NON PRODUCTION US	E ONLY	
File ISP Option	s <u>T</u> ools <u>H</u> elp		
۵ 🗔 🔍	i 🍪 🗸 🎩 🔈 😽 🛽	a 😮 😂 👘	
Step 1 - Communi	cations	Step 2 - Erase	
Select Device	LPC2148	Erase block 0 (0x000000	1-0x000FFF)
COM Port:	СОМ 1 🛛 🔽	Erase block 2 (0x002000	-0x001FFF)
Baud Rate:	9600	Erase block 3 (0x003000 Erase block 4 (0x004000	)-0x003FFF) )-0x004FFF)
Interface:	None (ISP)	Erase block 5 (0x005000	)-0x005FFF)
Oscillator (MHz):	12	Frase all Flash+Lode I	Hd Prot Hex File
Hex File: C:\First	_Project\First_Project.hex		Browse
Modified	t: Saturday, August 14, 2010, 1	2:14:13 PM <u>more ir</u>	10
Step 4 - Options		Step 5 - Start!	
Verify after prog Fill unused Flast Gen block chec Execute	ramming h sksums		Start
Visit the "Flash Ma www.esacademy.	agic" home page for info on the com/software/flashmagic	latest revision	•
1		U	

8. Observe the status bar at the bottom of the Flash Magic tool. Wait for the "**Finished**" status and press "**Reset**" on the LPC2148 development board and observe the LEDs.

orep 4 - options	arep a sarare
Verify after programming     Fill unused Flash     Gen block checksums     Execute	Start
On-Line training classes for microcontro Internetworking www.esacademy.com/fag/classes	llers and embedded networking and
Finished	2

If not successful then repeat again from step 6.



# 2.3.3 Configuring Startup.s File using Configuration wizard

The "**startup.s**" is the collection of assembly instructions that is supposed to be executed at the start-up of the application code. The start up file contains information about the PLL settings, peripheral clock divider, stack configuration, etc. This file can be found in the Keil installation directory. On creating a new project keil allows you to copy this file to the project source folder so that you can modify the settings as per your application requirements without modifying the original file. To execute the start up code on reset please ensure that following setting is done. To do this setting refer step 18 in section 2.3.1.

Lise Mer	nory Layout from Target Dialog			
I Mak I Mak I Don' I Rep	RW Sections Position Independent RO Sections Position Independent Search Standard Libraries nt 'might fail' Conditions as Errors	R/O Base: R/W Base disable Warnings:	0x00000000	
				1
Scatter File				Edit
Scatter File Misc controls				Edit

1. In the Project explorer double click on "startup.s" file.



2. The startup.s file will open up as assembly file. To open the wizard click **Configuration Wizard** as shown in figure below.



🔣 First_Project - µVision	on4	🗖 🖬 🗖
File Edit View Project	: Flash Debug Peripherals Tools SVCS Window Help	
🗋 💕 🖬 🖉 🕺 🐿	3 巻 ク で ( ← ⇒) 作 存 為 我 律 存 川 版 29 3 3 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3	
* * * * * * * *	Target 1 × K A 5	
Project (9 ×	X A mains A Startups	
🖃 🎦 Target 1		
E 🗃 Source Group 1	002 :/* STARTUP.S: Startup file for Philips LPC2000 */	-
The startup.s	003 :// construction Witard in Context Manu 200 s/	
	005 ;//***********************************	
	006 //* This file is part of the uVision/ARM development tools. */	
	000 :/* This software may only be used under the terms of a valid, current, */	
	009 ://* end user licence from KEIL for a compatible version of KEIL software */	
	010 // development cools. Nothing else gives you the light to use this software. */	
	012	
	013	
	015 ; * The STARTUP.S code is executed after CPU Reset. This file may be	
	UT6 /* translated with the following SET symbols. In uVision these SET 117 /* symbols are entered under Unitions - BSM - Define.	
	018 / *	
	019 ; * REMAP: when set the startup code initializes the register MEMMAP 020 ; f which organize the settings of the CBU configuration nine The	
	021 ; * startup and interrupt vectors are remaped from:	
	022 /* 0x0000000 default setting (not remapped)	
	023 /* 0/240000000 when EXING NOE 15 as at	
	025 / *	
	020 ) * Links moust when set the device is configured for code execution 027 ;* from external memory starting at address 0x80000000.	
	028 / *	
	US /* FRAM_MODE: when set the device is configured for code execution 130 ** from op-chip RAM starting at address 0x40000000.	
	031 2 *	
	032 /* EXTERNAL MODE: when set the FIN2SEL values are written that enable	
	034 / */	
	0.05	
	Use 037 ; Standard definitions of Mode bits and Interrupt (I & F) flags in PSRs	
	038	
	043 Hode Juk Rod Vil	
	041 Mode_IRQ EQU 0x12	-
■ P ● B   O F U. T		
Build Output		
		<u></u>
		~
S		2
For Help, press F1		imulation L1 C1 CAP NUM SCRU OVR R/W

3. Expand the PLL Setup tree to observe the default the PLL settings.

ption	Value	
Stack Configuration (Stack Sizes in Bytes)		
- Heap Configuration		
g. VPBDIV Setup		
PLL Setup		
MSEL: PLL Multiplier Selection	5	
- PSEL: PLL Divider Selection	2	
MAM Setup	<u> </u>	
3- External Memory Controller (EMC)		

To include the PLL code in the startup file select the checkbox next to PLL setup heading. Here MSEL setting indicates that the onboard crystal frequency (12MHz) will be multiplied by 5. This indicates that the CPU core frequency will be 60MHz and this is the maximum operating frequency for LPC2148 microcontroller. Care should be taken while selecting the multiplier value to ensure that the CPU core frequency is not beyond the maximum value. When using the PLL setup the minimum on board crystal frequency should be at least 10 MHz.

The PLL divider selection maintains the frequency of internal current controlled oscillator in the range of 150-320MHz. For more information please refer LPC2148 user manual. The default setting for **MSEL** is 5 and for **PSEL** it is 2. So the core frequency (FCLK) is 60MHz.

To change any setting just click on it and make use of the up/down arrows.



ption	Value	
Stack Configuration (Stack Sizes in Bytes)		
- Heap Configuration		
- VPBDIV Setup		
PLL Setup	-F	
- MSEL: PLL Multiplier Selection		
	2	
MAM Setup	M Value	
- External Memory Controller (EMC)		

4. Now expand VPBDIV Setup tree and observe its settings. It is used to prescale the peripheral clock (PCLK).

ion	Value
Stack Configuration (Stack Sizes in Bytes) Heap Configuration	
VPBDIV Setup	Г
	VPB Clock = CPU Clock / 4
XCLKDIV: XCLK Pin	XCLK Pin = CPU Clock / 4
PLL Setup	
MAM Setup	
- External Memory Controller (EMC)	F

Observe that in the default mode it is not included but at reset the VPBDIV loads the default value that is  $\frac{1}{4}$  of the CPU clock. Using the configuration wizard you can set the peripheral clock as:  $\frac{1}{4}$  of CPU clock or

<sup>1</sup>/<sub>2</sub> of CPU clock or equal to CPU clock.

For other settings please refer LPC2148 microcontroller user manual. After you have done using configuration wizard rebuild the project.



# 2.4 Using uVision Debugger in simulator mode

To use Debugger in simulator mode first we need to setup the debugger so as to use simulator.

1. Right click **Target 1** and select the highlighted option as shown in the figure below.



2. In the debug tab make sure that the highlighted settings are done. Click **OK** to continue.

ptions for Tar	rget 'Target 1'			
Device   Target	Output Listing User C/C++ Asm	Linker Debug	Utilities	
<ul> <li>Use Simulat</li> <li>Limit Speed</li> </ul>	or Settings to Real-Time	C Use: ULIN	IK ARM Debugger 💽 Settings	
✓ Load Applic Initialization File	ation at Startup 🛛 🔽 Run to main()	✓ Load Applic Initialization File:	ation at Startup 🦵 Run to main()	
	Edit		Edit	
Frestore Debu ↓ Breakpo ↓ Watch \ ↓ Memory	g Session Seturgs ints IV Toolbox Vindows & Performance Analyzer Display	Hestore Debug Session Settings     F Breakpoints     F Toolbox     Watch Windows     Memory Display		
CPU DLL:	Parameter:	Driver DLL:	Parameter:	
SARM.DLL	-cLPC2100	SARM.DLL		
Dialog DLL:	Parameter:	Dialog DLL:	Parameter:	
DARMP.DLL	-pLPC2148	TARMP.DLL	-pLPC2148	
	OKCa	ancel De	efaults Help	



3. Start Debug session by clicking on **Debug>Start/Stop Debug Session** from the main menu. Alternatively you can press Ctrl+F5 to toggle between starting and stopping of debug session.

🖫 First_Project - µVision4						
File Edit View Project Flash	Debug Peripherals Tools SVCS Wind	ow_Help				
	Start/Stop Debug Session Ctrl+	F5   // <sub>R</sub>   22				
🔮 🔛 🕮 🥔 🔜 🕌 Targe	Reset CPU					
roject 🥑 🗙 🥖	国, Run	F5				
🖃 🛅 Target 1 🛛 👘 🛛 🖓	Stop					
E Source Group 1 0;	{}} Step F	11 .				
± the main.c	⊕ Step Over F	10				
0!	{}+ Step Out Ctrl+F	11				
	⁺{} Run to Cursor Line Ctrl+F	10				
0	⇒ Show Next Statement	6)				
0:	Breakpoints Ctri	- 7) +B 8)				
1	Insert/Remove Breakpoint	F9 9)				

4. Click **OK** to continue.



5. Go to **Peripherals>System Control Block>Phase Locked Loop 0** to observe the PLL settings on the microcontroller.

🗓 First_Projec	t - µVision	4			
File Edit Viev	w Project	Flash Debug	Peripherals Tools SVCS W	indow	Help
	1 X Ca	19 C (	System Control Block	er	Memory Accelerator Module
RST   💷 🥹	(+) (+) (+)	*{} 🌩 💽	Pin Connect Block	6	Phase Locked Loop 0
Register	Value	25:	GPIO Slow Interface	1	Phase Locked Loop 1
E Current R0	0x00	Cx000002 0x000002 0x000002	UART		Power Control
	0x00 0x00 0x00	26: 0x000002	I2C Interface SPI Interface	3	External Interrupts Reset
R4 R5	0x00 0x00	Mair	SSP Interface Timer	•	System Control & Status
R6 R7 R8	0x00 0x00 0x00	07 08 #def	Pulse Width Modulator A/D Converter	, 16	5)
1 1000000		U9 #def		17	1



6. In the **Crystal Oscillator & Processor Clock** section observe that XTAL is 12MHz and CCLK is 60MHz. This is because the debugger has already executed the start up code when we started the debug session. It also ensures that PLL settings that we had done in the configuration wizard of the startup code are properly working.

Phase Locked Loop 0 (PLL0)	X
Control Register PLLOCON: CONS PLLE PLLC	
Configuration Register PLL0CFG: 0x24 MSEL: 5  PSEL: 2	1
Status Register	
PLLOSTAT: 0x0724 MSEL: 5 - PSEL: 2 -	I
PLLE PLLC PLOCK	
Feed Register PLL0FEED: 0x55	
Crystal Oscillator & Processor Clock	1
XTAL: 12.000000 MHz Crystal Oscillator (Fos	c)
CLOCK: 60.000000 MHz Processor Clock (CCL	.К)

7. Now click Peripherals>Pin Connect Block to include pin connect block,

🕱 First_Project - µVision4							
File Edit Viev	v Project Fl	ash Debug	Peripherals Tools SVCS Wir	ndow Help			
🗋 💕 🖬 🖪	XGR	2 5 6	System Control Block	• //= //= 🖄			
😫 🖾	P 0 ()	*()   🗢   🖂	Vectored Interrupt Controlle	- 🔜 + 💷 +			
Registers	<i>9</i> ×	Disassembly	CPIO Slow Interface	_			
Register	Value	25:	GPIO Fast Interface	// Enable C			
E Current B0	0x00.	0x000002	UART	<pre>,#0x0000000 ,[PC,#0x009</pre>			
B1	0x00	0x000002	I2C Interface	• / [R1]			
B2	0x00	0x000002	SPI Interface	▶,[R1,#0x000			
B4	0x00	<	SSP Interface				
R5	0x00	📩 mair	Timer	•			
R6	0x00		· · · · · · · · · · · · · · · · · · ·	-			



8. Similarly select Port 1 window.

First_Proje	ct -µVision4 w Project I	l Flash Debug Pe	ripherals Tools SVCS Wind	ow Help
🗋 🚅 🖬 🕼	ነ∉ %   የነዋ የ	8 9 0   1) 0   E V	System Control Block Vectored Interrupt Controller Pin Connect Block	▶  /≞ //╦   200 
Registers	, G ×	Disassembly	GPIO Slow Interface	Port 0
Register	Value	25: ¢ <mark>0x000002</mark>	GPIO Fast Interface	Port 1
RO	0x00	0x000002	UART	▶,[PC,#0x0090]
	0x00	0x000002	I2C Interface	• / [R1]
- R2	0x00	26: 0x000002	SPI Interface	,[R1,#0x0004]

9. Arrange the windows as shown in figure below.



10. Now use **Step** function to step through the code. Alternatively you can press Ctrl+F11. Observe the changes in the Pin connect block and Port1 block as you step through the code.

🖫 First_Projec	:t - µVision4	1					
File Edit View	w Project	Flash Debug	Peripherals	Tools	svcs v	Vindow	Help
0 🖌 🖬 🕯	1 X G1	8 9 9	(a a) [Pa	12 13	18	尾律 /	l≣   j
👬 🗄 🚳	የ የ	*{} 🗢 🔽		6	-	• 🛃 •	
Registers	{+} Step	(F11) embly					
Register	Step	one line 25:	PINSELO	= 0x00	000000	); /	/ E1
<b>Current</b> R0 R1 R2 R3 R4	0x00 0x00 0x00 0x00 0x00	Cx000002 0x000002 0x000002 26: 0x000002	58 E3A00 50 E59F1 60 E5810 PINSEL1 64 E5810	0000 M 090 L 0000 S = 0x00	IOV DR TR 000000 TR	RO, R1, RO, ); RO,	#OxC [PC, [R1] [R1,
	0x00 0x00 0x00	<b>main</b> .	c 🗵 🟦 :	Startup.s			



11. When you encounter the delay function you can simply use Step Out function. It will execute the delay function and take execution to the next line immediately after the delay function.

🛚 First_Proj	ject - µVisior	14			
File Edit V	/iew Project	Flash Debug	Peripherals	Tools SVCS	Window Help
🗋 🚰 🛃	Ø X G	100	(a. a) [P	12 12 12	
RST 🗐 🚳	0 6 6	₹}   ⇒   Σ		- <b>B</b>	💽 😼 • 🔳 • 🛛
Registers	9	Disassembly	d+ E11)	_	
Register	Value	Step out of t	the current fur	tion 60000;	i++);
Current R0 R1 R2 R3 B4	0x00 0xE0 0x40 0x40 0x00	0x000002 0x000002 0x000002	) 22C E3A0: 230 EA000 234 E281: 238 E59F2	1000 MOV 0000 B 1001 ADD 2080 LDR	R1,#0x000 0x000002: R1,R1,#0: R2,[PC,#0
R5	0x40 0x00	🛃 mair	n.c 🔀 🔝	Startup.s	

12. Now press the Reset button to Reset the CPU and observe the PLL window.

👿 First_Project - µVision	4	
File Edit View Project	Flash Debug Peripherals Tools	SVCS Win
	日とうの	内内律
👫 🗉 🔕   Pr OF O		<b>-</b>
Registers 🕜 🗙	Disassembly	
Ben Rest Reset	239: Vectors	LDR
Reset the CPU dido	➡Ox00000000 E59FF018	LDR
	240:	LDR
P1 0.00	0x00000004 E59FF018	LDR
B2 0.00	241:	LDR
B2 0v00	0x0000008 E59FF018	LDR
B4 0v00	<	
R5 0x00	👔 main.c 🔀 🛃 Startup	i.s



13. Observe that CCLK is also reset and now it is equal to XTAL frequency. Similarly the GPIO and pin connect block vales are also reset.

	Pin Conne	ct Block	
hase Locked Loop 0 (PLL0)	Pin Conne           Pin Name           P0.0           P0.1           P0.2           P0.3           P0.4           P0.5           P0.6           P0.7           P0.8           P0.9           P0.11           P0.12           P0.13           P0.14           P0.15           Selected P           P0.0:	Ct Block           Function           GPI0 Port 0.0           GPI0 Port 0.1           GPI0 Port 0.2           GPI0 Port 0.3           GPI0 Port 0.4           GPI0 Port 0.5           GPI0 Port 0.6           GPI0 Port 0.7           GPI0 Port 0.9           GPI0 Port 0.10           GPI0 Port 0.11           GPI0 Port 0.12           GPI0 Port 0.13           GPI0 Port 0.14           GPI0 Port 0.15           Yin           GPI0 Port 0.0	
Crystal Oscillator & Processor Clock XTAL: 12.000000 MHz Crystal Oscillator (Fosc) CLOCK: 12.000000 MHz Processor Clock (CCLK)	Pin Select PINSELD: PINSEL1:	0x00000000 PINSEL2: 0x00000030 0x15400000	
General Purpose Input/Output 1 (GPIO 1) - Slow In           GPI01         31         Bits         24         23         Bits           I01DIR:         0x000000000         1	16 15	Bits 8 7 Bits 0	
Pins: 0xFFFF0000 [			

14. Now let us setup a break point at line no. 25 i.e. at the beginning of the main function. It will halt the execution when the debugger encounters this line.





15. Now click Run to start the execution.

🖫 First_Projec	:t - µVisio	n4					
File Edit Viev	w Project	Flash	Debug	Perip	oherals	Tools	SVC
i 🗋 💕 🖬 🕯	1 X D	8	5 (2)	-(m. 1	+ P	17	A 8
Ret 💷 🛛	P ()	}- <b>-</b> {}	4	D		181.	- 🔍
Registers 🗐 Run	(F5)	Disa	issembly				
Register	code execu	tion	25:	PII	VSELO	= 0x1	00000
Current R0 R1 R2 R3	0x00 0x00 0x00 0x00 0x00		x000002 x000002 x000002 26: x000002	258 25C 260 PII 264	E3A00 E59F1 E5810 NSEL1 E5810	0000 1090 0000 = 0x1	MOV LDR STR DOODO STR
	0x00 0x00		📩 mair	n.c 🗴		Startup	.s

16. After clicking on RUN observe the change in the CCLK field. The execution has also halted at the breakpoint that was setup earlier. Now again if you click Run the code will continue to execute.







#### 3.1 Main Expansion Header

0.3 0.5 00.7 00.1 00.11 00.15 00.15 P0.31 P1.17 P1.19 19 21 23 25 25 333 P0. P0. P0. P0. P0. Ы. R Ц ā H ര P0.0 P0.0 P0.2 P0.2 P0.10 P0.12 P0.15 P0.20 P0.20 P0.20 P1.16 P1.20 P1.2 2 2

All port pins on LPC2148 microcontroller are accessible on the 50 pin main expansion header. 50 pin FRC wire can also be connected to this connector. The expansion header allows user to interface external peripherals to LPC2148 microcontroller. Apart from port pins, it also includes 3.3V supply which can be utilised to power external devices. When using the expansion header, it is necessary to disconnect the on-board peripherals connected to these pins by removing the jumpers which links these pins to the on-board peripherals. Refer following sections for jumper settings.



Note: 1. Total current drawn by all external peripherals must not be more than 150mA.

2. Do not connect any external supply to the 3V3 pin on the expansion header

# **3.2** LEDs



LPC2148 development board has 4 user programmable LEDs. The port pins for LEDs are as shown in the above figure. Jumper D1 is associated with Led D1 and so on. Jumper positions are as shown in the above figure. Alternatively you can make connection between any port pin and LED directly using 1to1 cables. All LEDs are common ground with series resistor of 1K.

Note: 1. Port pins P1.16, P1.17, P1.18 and P1.19 are also used for L293D direction inputs.

### **3.3 USER SWITCHES**



LPC2148 development board has 4 user programmable switches. The ports pins and jumper settings are as shown in the above figure. When a switch is pressed it will pull the associated pin to logic '0'. Port pins P0.15 and P0.30 features alternate function of external interrupts. Jumper SW1 is associated with switch SW1 and so on.



# **3.4 LCD INTERFACE**



The LPC2148 development board is included with 4-bit HD44780 based LCD interface. It can be used to interface 16x1, 16x2, 20x4 characters LCD display. The board is shipped with 16x2 character LCD display. The jumper positions and port pins are as shown in the above figure. The **B**/L jumper setting is used to control LCD backlight. On removing this jumper LCD backlight will turn OFF.

## 3.5 BUZZER



Buzzer is connected to P0.11 of LPC2148 microcontroller. Logic '0' on P0.11 will turn ON buzzer. P0.11 is open drain type and it requires external pull-up for proper operation. The LPC2148 development board have a pull-up on buzzer side. If using this pin for some other purpose then it is recommended to use a pull-up of 10Kohms on the external board.

# 3.6 IR RECEIVER

	-
~	0
	Θ
	$\odot$
	n

There is a TSOP1738 IR receiver on LPC2148 development board. The jumper position is as shown in the above figure. Alternatively IR receiver can be connected to any port pin using 1-1 cable.



# **3.7 TRIMPOTS**



LPC2148 development board has two trimpots connected to AD0.1 and AD0.2 on the LPC2148 microcontroller. The jumper positions are as shown in the above figure. Jumper AN1 represents trimpot AN1 and AN2 represents trimpot AN2. The trimpot voltage range lies between 0-3.3V.

#### **3.8 UART1/XBEE/BLUETOOTH/WIFI**



The LPC2148 development board has a provision for interfacing XBee / Bluetooth / WiFi wireless modules. The wireless modules use UART for the communication with microcontroller. The UART1 on LPC2148 microcontroller can be used for this purpose. The jumper positions for the desired operation are shown in above figures. If jumper positions are as per fig.3.1 then UART1 is selected for Xbee wireless communication. If jumper positions are as per fig.3.2 then UART1 is selected for RS-232 serial communication.

### **3.9 SPI INTERFACE**



The LPC2148 microcontroller features 2 SPI interfaces i.e. SPI0 and SPI1. On LPC2148 development board SPI0 is used for interfacing SD/MMC card. The SPI expansion NEX Robotics Pvt. Ltd. 47
www.nex-robotics.com



header contains 4 signals related to SPI0. The jumper positions and associated pins for SPI-SD/MMC interface are shown in the above figure. To use SPI0 for the external interfacing, remove these jumpers. The SPI0 pin conventions on the expansion header are with respect to master i.e. LPC2148 microcontroller.

# 3.10 I2C INTERFACE



The I2C0 peripheral on LPC2148 development board is used for interfacing 24LCxxx EEPROM memories from microchip. The above figure describes the jumper positions and pins used for I2C interface.

## 3.11 ULN2003



The LPC2148 development board features a high current ULN2003 driver. It has 7 darlington transistor array. Each transistor can sink current up to 500mA. This device is useful for driving a wide range of loads including solenoids, relays, DC motors, LED displays filament lamps, thermal print heads and high power buffers etc. *For more information on ULN2003 refer to data sheet included in the CD*. Fig.3.3 describes the jumper positions for logic inputs to the ULN2003 driver. To use any channel insert a jumper in its respective position. Fig.3.4 describes the expansion header for connecting relays, stepper motors, LEDs or any inductive load. Logic '1' on input channel will drive the load connected to the corresponding channel. The application circuit used in the experiment is shown in the figure below.



The cathode of LED is connected to the pin named as '1' on Driver header and I/O pin connected is connected to pin named as '1' on Logic header by placing a jumper in its position.



# 3.12 L293D 600mA Dual DC MOTOR DRIVER



Fig.3.5 Supply and motor connections

The LPC2148 development board has one L293D motor driver IC. L293D is designed to provide bidirectional drive currents of up to 600mA at voltages from 5V to 36V. *For more information on L293D motor driver IC refer device data sheet included in the CD.* 

Each motor driver channel requires two I/O pins for direction control and 1 I/O pin for velocity control of a DC motor. The motor power supply is separated from the logic supply. The logic supply is provided from the on board power supply. The motor supply should be provided externally as shown in fig.3.5.



The L293D direction control port and LED port uses same pins on LPC2148 microcontroller. So in order to use L293D driver all the jumpers on the LED header should be inserted as shown in the fig.3.6.

The velocity of the DC motor can be varied using the PWM port on the LPC2148 microcontroller or it can be fixed to maximum velocity by connecting velocity pins to 5V. For this purpose, LPC2148 development board has a header for selecting the source of velocity signal.



If jumper positions are as shown in fig. 3.7a, PWM channels 4 & 6 are selected as source of velocity signal. If jumper position are as shown in fig. 3.7b, +5V is selected.



Note: PWM channels 4 & 6 are multiplexed with UART1 RX and TX signals on LPC2148 microcontroller. When PWM function is enabled on these pins do not insert any jumper in the XBEE/UART header.

# 3.13 JTAG PORT



The LPC2148 development board has a 10pin (2X5) JTAG box header for on chip debugging. JTAG debugging requires pulling P1.26 low on reset in order to enter into debug mode. To achieve this there is a 2 pin jumper on the development board as shown in the figure above.

# 3.14 RTC



The LPC2148 microcontroller features an in-built RTC and it provides Seconds, Minutes, Hours, Day of Month, Month, Year, Day of Week, and Day of Year. It has a dedicated 32.768 kHz oscillator or programmable prescaler from VPB clock. It also has a dedicated power supply pin that can be connected to a battery or to the board supply of 3.3V. LPC2148 development board has battery holder for providing battery back up to the internal RTC.



# 4.0 SAMPLE PROGRAMS

The following sample programs are included in the CD. The sample programs are written in keil uVision IDE version 4 by using the Real View complier.

The hex file can be loaded on the microcontroller flash memory using Flash Magic utility. Detailed instructions for loading a hex file using Flash magic utility are provided in section 2.3.2.

The default clock settings for all sample programs:

Clock Settings		
FOSC	12MHz	
PLL	M=5, P=2	
CCLK	60MHz	
PCLK	15MHz	

## 4.1 LED CHASER

Hardware Setup: Insert LED jumpers D1-D4.

Load the LED\_CHASER.hex file from LPC2148>codes>LED CHASER folder using flash magic on the microcontroller flash memory.

**Description:** After loading the hex file, press reset switch. This application code will blink D1-D4 LEDs sequentially. The code explains how to use GPIO registers of LPC2148 microcontroller.

# 4.2 IO INTERFACING

Hardware Setup: Insert LED jumpers D1-D4. Insert Switch jumpers SW1-SW4

Load the IO\_INTERFACING.hex file from LPC2148>codes> IO INTERFACING folder using flash magic on the microcontroller flash memory.

**Description:** After loading the hex file, press reset switch. This application demonstrates simple IO interfacing on LPC2148 development board. The program basically explains how to read a pin and control the output. To see the result, press any switch and observe the corresponding LED glow.

# 4.3 BUZZER

Hardware Setup: Insert SW4 Jumper. Insert Buzzer Jumper

Load the BUZZER.hex file from LPC2148>codes>BUZZER folder using flash magic on the microcontroller flash memory.



**Description:** After loading the hex file, press reset switch. This application code explains how to use external hardware interrupts. Switch SW4 is used to trigger Ext interrupt EINT3. When switch SW4 is pressed buzzer beeps at different speeds.

# 4.4 LCD INTERFACING

Hardware Setup: Insert all LCD jumpers.

Load the LCD\_INTERFACING.hex file from LPC2148>codes>LCD INTERFACING folder using flash magic on the microcontroller flash memory.

**Description:** After loading the hex file, press reset switch. This application code demonstrates LCD interface on LPC2148 development board. LCD will display "Nex Robotics ARM LPC2148".

# 4.5 UART0 COMM (9600 bps)

**Hardware Setup:** Connect a DB9 cable between PC and UART0. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Enable terminal to receive data in string format.

Load the UART0\_COMM\_9600.hex file from LPC2148>codes>UART0 COMM 9600 folder using flash magic on the microcontroller flash memory.

**Description:** This application code demonstrates UART0 peripheral on LPC2148. On reset, it sends a string of characters to the terminal software running on the PC. It then asks for a character to be sent from the PC to complete the communication test.

# 4.6 UART1 COMM (9600 bps)

Hardware Setup: Refer jumper settings for using UART1 from section 3.8.

Connect a DB9 cable between PC and UART1. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Enable terminal to receive data in string format.

Load the UART1\_COMM\_9600.hex file from LPC2148>codes>UART1 COMM 9600 folder using flash magic on the microcontroller flash memory.



**Description:** This application code demonstrates UART1 peripheral on LPC2148. On reset, it sends a string of characters to the terminal software running on the PC. It then asks for a character to be sent from the PC to complete the communication test.

# 4.7 ADC

Hardware Setup: Insert trimpot jumpers AN1 & AN2.

Connect a DB9 cable between PC and UART1. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Enable terminal to receive data in string format.

Load the ADC.hex file from LPC2148>codes>ADC folder using flash magic on the microcontroller flash memory.

**Description:** After loading the hex file, press reset switch. This application code demonstrates ADC interfacing on LPC2148. The two trimpots are connected to AD0.1 and AD0.2 analog input channels of the microcontroller. The digital data is used to represent the measured voltage and it is sent to the terminal software running on the PC. AN1 is connected to AD0.1 and AN2 is connected to AD0.2. By varying the potentiometer you can observe the change in values on serial terminal software running on the PC.

# 4.8 I2C EEPROM

Hardware Setup: Insert LED jumpers D1-D4. Insert I2C jumpers SDA & SCL

Connect a DB9 cable between PC and UART0. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Load the I2C\_EEPROM.hex file from LPC2148>codes> I2C EEPROM folder using flash magic on the microcontroller flash memory.

**Description:** This application code explains reading and writing on I2C EEPROM 24LC04. On reset, microcontroller will write 16 bytes of DATA on the EEPROM 24LC04 and read back the same data. The read data is then sent to UART0, which can be observed on the receive window of terminal software.



# 4.9 SD MMC INTERFACE

Hardware Setup: Insert SPI jumpers SCK, SDO, SDI and /CS.

Insert SD/MMC card into the socket. Connect a DB9 cable between PC and UART1. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Enable terminal to receive data in string format.

Load the SD\_MMC\_INTERFACE.hex file from LPC2148>codes> SD MMC INTERFACE folder using flash magic on the microcontroller flash memory.

**Description:** This application code demonstrates SD/MMC interface on SPI bus. On reset microcontroller writes a string of data as "NEX ROBOTICS" on the memory card and reads back the same data. This data is sent to UARTO, which can be observed on receive window of terminal software.

# 4.10 DC MOTOR CONTROL

Hardware Setup: Insert LED jumpers D1-D4.

Insert Switch jumpers SW1-SW4 Insert PWM jumpers for +5V Connect Motor i/p supply (12V) to the V pin on L293D header

Load the DC\_MOTOR\_CONTROL.hex file from , LPC2148 > codes >DC MOTOR CONTROL folder using flash magic on the microcontroller flash memory.

**Description:** This application demonstrates, how to use L293D for controlling DC motor. SW1 is used to Start/Stop Motor1 connected across terminals A & B on the L293D o/p header. SW2 is used to control direction of Motor1. SW3 is used to Start/Stop Motor2 connected across terminals C & D on the L293D o/p header. SW4 is used to control direction of Motor2.

# 4.11 STEPPER MOTOR CONTROL

**Hardware Setup:** Insert ULN2003 jumpers 1-4. Connect common wire on the stepper motor to external supply as per the motor's rating. Connect the other 4 wires i.e. Coil 1 to coil 4 to the inputs 1 to 4 on the ULN header. If stepping sequence is inappropriate then alter the connection sequence.

Load the STEPPER\_MOTOR\_CONTROL.hex file from LPC2148>codes>STEPPER MOTOR CONTROL folder using flash magic on the microcontroller flash memory.



**Description:** This application code drives unipolar stepper motor using ULN2003 driver on the LPC2148 development board

### **4.12 RTC UART0**

**Hardware Setup:** Connect a DB9 cable between PC and UART0. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Load the RTC\_UART0.hex file from LPC2148>codes>RTC UART0 folder using flash magic on the microcontroller flash memory.

**Description:** This application code demonstrates RTC peripheral on LPC2148. It transmits current time in 24h format on UART0. Enable terminal to receive data in string format. By pressing reset switch, the current time of RTC can be displayed on receive window of terminal software.

#### 4.13 XBEE/BLUETOOTH/WIFI WIRELESS COMMUNICATION

**Hardware Setup:** Refer jumper settings for using Xbee from section 3.8. Connect Xbee USB adapter module to PC and install drivers. To install drivers refer Xbee USB adapter module documentation available in the CD. Setup the terminal software with following settings:

Baudrate	9600
Databits	8
Parity	None
Stopbits	1

Enable terminal to receive data in string format.

Load the XBEE\_WIRELESS\_COMM.hex file from LPC2148 > codes > XBEE WIRELESS COMM folder using flash magic on the microcontroller flash memory.

**Description:** This application code uses Xbee wireless module for communication over UART1 peripheral on LPC2148. The jumpers settings are required to be changed in order to select Xbee for UART1 on LPC2148 development board(see section 3.8). We will also require a Xbee USB adapter module for receiving and transmitting data on the PC side. Enable terminal to receive data in string format. After loading the hex file, press reset switch. on microcontroller board, The microcontroller board will send the below message, which can be observed on the receive window of terminal software on PC, where the Xbee USB module is connected,



"Nex Robotics PVT LTD ARM7LPC214x Development Board Communication Test Send any character to continue"

As instructed in the last line of the above message, please type any character in transmit window of terminal software. This character is received by the development board via Xbee module present on board. The development board will reply by sending the message as "Test Passed" to the terminal software running on PC. This message can be observed in receive window of terminal software.

## 4.14 BOARD DEMO

#### Hardware Setup:



1. Insert jumpers as highlighted in fig. 4.1

2. Insert any SD/MMC card of size up to 2GB in the SD/MMC slot.

3. Load hex file using flash magic and follow the instructions on LCD display.

Load the BOARD\_DEMO.hex file from LPC2148>codes>BOARD DEMO folder using flash magic on the microcontroller flash memory.

**Description:** The Demo program is used to test all the peripherals on the development board. After loading the hex file refer instructions on the LCD display.

LPC2148 DEVELOPMENT BOARD



# References

Volume 1: LPC214x User manual AN10406-Accessing SD/MMC card using SPI on LPC2000 AN10404-Initialization code/hints for the LPC2000 family AN10331-Philips LPC2xxx family phase lock loop SanDisk SD Card Product Manual