

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)

An UGC Autonomous Institution, affiliated to JNTUH, Accredited by NAAC with 'A' Grade & NBA and Recipient of World Bank Assistance under TEQIP-II S.C.1.1



DEPARTMENT OF ELECTRONICS AND

COMMUNICATION ENGINEERING

LAB MANUAL

54310 – ADVANCED EMBEDDED SYSTEM LABORATORY I M.Tech II Semester (MR-15 Regulations)

Prepared By

Thota Ravi Theja, Assistant Professor, ECE Department, MREC (A). Verified By

Approved By

LIST OF EXPERIMENTS

Note: The following programs are to be implement on ARM based Processors using Keil IDE and Flash magic.

List of experiments:

- 1. Introduction to ARM Development Board & Software
- 2. Simple Assembly Program for
 - a. Addition | Subtraction | Multiplication | Division
- 3. Simple Assembly Program for
 - a. Operating Modes, System Calls and Interrupts
 - b. Loops, Branches
- 4. Write an Assembly programs to configure and control General Purpose Input/Output (GPIO) port pins.
- 5. Write an Assembly programs to read digital values from external peripherals and execute them with the Target board.
- 6. Program to demonstrate Time delay program using built in Timer / Counter feature on IDE environment
- 7. Program to demonstrates a simple interrupt handler and setting up a timer
- 8. Program demonstrates setting up interrupt handlers. Press button to generate an interrupt and trace the program flow with debug terminal.
- 9. Program to Interface 8 Bit LED and Switch Interface
- 10. Program to implement Buzzer Interface on IDE environment
- 11. Program to Displaying a message in a 2 line x 16 Characters LCD display and verify the result in debug terminal.
- 12. Demonstration of Serial communication. Transmission from Kit and reception from PC using Serial Port on IDE environment use debug terminal to trace the program.

INDEX

S.NO	Name Of The Experiments	Page No
1	Introduction to ARM Development Board & Software	1-20
2	Simple Assembly Program for Addition Subtraction Multiplication Division	21
3	Simple Assembly Program for a. Operating Modes, System Calls and Interrupts b. Loops, Branches	22
4	Write an Assembly programs to configure and control General Purpose Input/Output (GPIO) port pins.	23-26
5	Write an Assembly programs to read digital values from external peripherals and execute them with the Target board	27-28
6	Program to demonstrate Time delay program using built in Timer / Counter feature on IDE environment	28-29
7	Program to demonstrates a simple interrupt handler and setting up a timer	30-33
8	Program demonstrates setting up interrupt handlers. Press button to generate an ninterrupt and trace the program flow with debug terminal.	34-37
9	Program to Interface 8 Bit LED and Switch Interface	38-40
10	Program to implement Buzzer Interface on IDE environment	41-43
11	Program to Displaying a message in a 2 line x 16 Characters LCD display and verify the result in debug terminal.	44-47
12	Demonstration of Serial communication. Transmission from Kit and reception from PC using Serial Port on IDE environment use debug terminal to trace the program.	48-52

Experiment: 1

Introduction to ARM Board (LPC2148):

This section of the document introduces LPC2148 microcontroller board based on a 16bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine microcontrollers with embedded high-speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30% with minimal performance penalty. The meaning of LPC is Low Power Low Cost microcontroller. This is 32 bit microcontroller manufactured by Philips semiconductors (NXP).

Due to their tiny size and low power consumption, LPC2148 is ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale.

Features of ARM Microcontroller:

16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.8 kB to 40 kB of on-chip static RAM and 32 kB to 512 kB of on-chip flash memory; 128-bit wide interface/accelerator enables high-speed 60 MHz operation. In-System Programming/In-Application Programming (ISP/IAP) via on-chip boot loader software, single flash sector or full chip erase in 400 ms and programming of 256 Bytes in 1 ms 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 2kB of endpoint RAM. In addition, the LPC2148 provides 8 kB of on-chip RAM accessible to USB by DMA. One or two (LPC2141/42 vs, LPC2144/46/48) 10-bit ADCs provide a total of 6/14 analog inputs, with conversion times as low as 2.44 ms per channel.Single 10-bit DAC provides variable analog output (LPC2148 only) Two 32-bit timers/external event counters (with four captures 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.



Modules and Jumpers Relationship

Jumper	Related	Usage
J6	UART0 & UART1	Connecting all pins enables both UART0 and UART1 and pins 5 and 7 enable UART0.
J8	VREF voltage	Connecting this will set the VTEF voltage to .3V
J9	Test LED's	Connecting all pins enables test LED's Pins 3 to 9 are connected to SPI0 lines of LPC2148.
J10	ADC	This will enable the ADC interface
J11	JTAG	This will enable the debug mode on the microcontroller.
J12	Keyboard(PS/2)	This will enable the PS/2 connector.
J13	Keyboard(PS/2)	This will provide 5V supply to PS/2
J18	LCD	Connecting all pins enabled LCD. Pins 1 to 7 are data lines, 9 to 13 are control lines and pin 15 is 5V power pin.
J19	LCD Backlight	If pins 1 are 2 are connected the LCD back light will always stay PN and if pins 2 and 3 are connected the back can be controlled by firmware.
J22	Power supply to board	Connecting this will provide 3.3V supply to board.
J25	I2C	By connecting all pins it enables I2C interface and its status is displayed on LCD.
J26	Bootloader select	If pins 1 and 2 are connected, manual bootloader mode is selected and If pins 2 and 3 are connected auto bootloader mode is selected. UART0 to be used foe this purpose.
J27	RTC	Connect a battery to use RTC.

Introduction to KEIL software:

Introduction:

- 1. Keil was found in 1982 by Gunler and Reinhard keil.
- 2. Keil implemented the first c compiler design.
- Keil provides a board range of development tools, ANSK, macro assembler, debugger and linkers.
- 4. Keil micro vision debuggers accurately simulate on-chip peripherals (input/output, A/D converters, D/A converters).

Features:

- 1. Nine basic data types including 32-bit IEEE floating point.
- 2. Flexible, variable allocation with bit, data, bdata, idata, xdata and pdata of the memory types.
- 3. Interrupt functions may be written in C.
- 4. Fall use of the 8051 register banks.
- 5. Complete symbol and type information for source level debugger.
- 6. Use of AIMP and ACAII instructions.
- 7. Bit addressable data objects.
- 8. Built in interface for the RTX51 real time kernel.
- 9. Support for dual data pointers at the ATMEL, AED, EMPRESS, and DELL as semiconductor in line on Philips and transcend microcontroller.

4

KEIL µVision 4 Tool

How to work with keil?

How to create a new µProject?

Step 1: Give a double click on μ vision 4 icon on the desk top, it will generate a

window as shown below.



Step 2: To create new project go to project select new micro vision project.

🛛 🕅 🕅	sion4			
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	Pro	ject Fl <u>a</u> sh <u>D</u> ebug Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp	
	3 🗐 🗿 L		New µ⊻ision Project	
Project	5 M 🤣 F		New Multi-Project <u>W</u> orkspace Open Project <u>Close Project</u>	
			Export	
			Manage	•
			Select Device for Target	
			Remoye Item	
		1	Options Alt+F7	
			Clean <u>t</u> arget	
			Build target F7	
		1911	Rebuild all target files	
		9	Batch Build	
		۲	Translate Ctrl+F7	
			Stop build	
			1 C:\Documents and Settings\string\Desktop\Spectrum analyser(keil)\spectrum.uvproj 2 \\10.0.0.130\Documents\Real time data(Keil)\Source\Real_time.uvproj	

Step 3: select a drive where you would like to create your project.



Step 4: Create a new folder and name it with your project name.

👿 µVision4				
<u>File Edit View Project Flash D</u> eb	ug Pe <u>r</u> ipherals <u>T</u> ool	ls <u>S</u> VCS <u>W</u> indow <u>H</u> elp		
- C 🖉 🖉 🖇 🖓 🖓 🔄 🖓	4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	🛝 🖅 🚝 //E //E 🏄 📃	* 🔍 (*	
: 🕸 🏝 🖨 🥔 🚉 🔐	- 8 4	5 2		
Project 👻 A 🗙	Create New Project	t » Local Disk (D:)	 + E * II 	
	My Recent Documents Desktop My Documents My Computer	adc_code GLCD gnud RFID t_sc tiny encryption in keil_after modifi touch dwnid tsx tsy <u>New Folder</u>		
Build Output	My Network Places Sav	name: ve as type: Project Files (*.uvp	roj)	Dpen Cancel

Step 5: Open that project folder and give a name of your project executable file and save it.

Advanced Embedded Syste	m Lab]					
🕱 µVision4						
<u>File Edit View Project Flash Deb</u>	oug Pe <u>r</u> ipherals	<u>T</u> ools <u>S</u> VCS	<u>W</u> indow <u>H</u> elp			
i 🗅 💕 🗐 🏈 l 🗴 🔁 🔂 i 🤊 🖓 i	and the W	四限目標目	= //# //# 🗠 🧖	- IA	* Q 0 0 6	
	- 1	168				
Project 👻 🕈 🗙						
						2
	Create New Pr	oject			? 🛛	
	Save in:	C demo				
]	
	Mu Recent					
	Documents					
	Desktop					
	My Documents					
	My Computer					
	My Network	File name:	demoproj	×	Save	
Build Output	Places	Save as type:	Project Files (*.uvproj)	•	Cancel	
		1999 - C. S. S.				

Step 6: After saving it will show some window there you select your microcontroller company i.e NXP from Phillips.



Step 7: Select your chip as LPC2148.



Step 8: After selecting chip click on OK then it will display some window asking to add STARTUP file. Select YES.



Step 9: A target is created and startup file is added to your project target and is shown below.

[Advanced Embedded System Lab]				
🔣 demoproj - µVision4				
Eile Edit View Project Flash Debug Peripher Image: Image to the state of the state o	als Iools SVCS Window Help	- 🔍 \land 🔍 🗠 🔿 🍕 🔚 - 1 👟		
Project ■ I arget I Source Group 1 Startup.s				

Step 10: To write your project code select a new file from FILE menu bar or

click on icon.

W d	emoproj - µVision4	
Eile	Edit View Project Flast	Debug Peripherals Tools SVCS Window Help
	<u>N</u> ew Ct	1+N 🗛 🗠 🛝 🛝 🕸 🕸 ///////////////////////////
1	Open Ct	1+0
	Close	
a	Save	<mark>ri+S</mark>
	Save <u>A</u> s	
ø	Save All	
	<u>D</u> evice Database	
	License <u>M</u> anagement	
	P <u>r</u> int Setup	
\mathbf{a}	Print C	d+P
	Print Preview	
	<u>1</u> main	
	2 UART0_115200	
	<u>3</u> Startup	
	<u>4</u> target	
	5 Os_cpu_c	
	6 OS_CORE	
	<u>7</u> main	
	<u>8 \\10.0.130\Documents\\LC</u>	D
	9 UARTO	
	10 interrupt	
	Exit	

Step 11: It will display some text editor, to save that file select SAVE option from FILE menu bar.

[Advanced Embedded Sy	stem Lab]	
🕱 demoproj - µVision4		
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject Fl <u>a</u> sh	<u>Debug</u> Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp	
🗋 🔂 🛃 🗶 🕺 🖄 🛍 👘		🔹 🔜 🥐 🔍 🗠 🔿 🔗 🛃
😸 🔛 🥔 🔜 🔤 Target 1	🖻 🔊 📥 🗟	
Project • 4 ×		

Step 12: By giving a file name with extension .C for c files and save it.

🔣 demoproj 🛛 🛛 🖳 🛛 🖳		
<u>File Edit View Project Flash Deb</u>	bug Peripherals Tools SVCS Window He	
🐼 🏔 🧼 🗛 🙀 Target 1	🛋 🔊 📥 📇	
Project 🔻 🕈 🗙	Text1	
Target 1 Source Group 1 Startup.s	Save As	
	Save in: 🗀 demo	
	My Recent Documents Desktop My Documents My Computer	
Build Output	My Network File name: abc.c Places	Save
build output	Save as type: All Files (*.*.	") Cancel

Step 13: Write the code of your project and save it.

[Advanced Embedd	ded System	Lab]		
🔣 demoproj - µVision4 🛛				
File Edit View Project	Flash Debug	Peripherals Tools SVCS Window	w Help	
			11= 1000	
	Target 1	🖾 🔊 💼 🔁		
Project	• • × 🗋	abc.c*		
Source Group 1	01 02 04 05 06 07 08 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 3 4 ↓	<pre>#include <lpc214x.h> void delay(int); int main() { unsigned int a,1; PINSEL0 = 0x00000000; PINSEL1 = 0x00000000; PINSEL2 = 0x00000000; a=0x00010000; while(1) { //TOSET1 = 0x000F0000; delay(10); loCLR1 = a; delay(10); loCLR1 = a; delay(10); a = a<<1; } for(i=0;i<=3;i++) {</lpc214x.h></pre>	//enable a // Set Pl. //turn on //turn of	ill the pins as GPIO 16, P1.17, P1.18, P1.19 as Output : LED f LED
Build Output				
P				

Step 14: To add our c file to target give a right click on Source Group, choose "ADD files to Group "option.

🕊 demoproj - µVision4	
<u>File Edit View Project Flash Debug Peripherals Tools SVCS</u>	Window Help
	i li li li 🖄 🔽 🔹 🕄 🖓 🖓 🔍 🔍 🖉 🖉 🖓
🐼 🖾 🧼 🔛 🙀 Target 1 💽 🔊 📥 🔁	
Project 🔍 🔍 🗋 abc.c	
Target 1 O1 #include <lpc214x.1< th=""><th>∃></th></lpc214x.1<>	∃>
Sta 🔊 Options for Group 'Source Group 1' Alt+F7	
Open File	
Open List File	DO; //enable all the pins as GPIO
Open <u>M</u> ap File	bo; po:
Rebuild all target files	D: // Set P1.16, P1.17, P1.18, P1.19 as Output
Build target F7	and and a substantian constraining constraining constraining constraining a substantiant tark substantiants
Tr <u>a</u> nslate File	
Et Stop build	0000;
A <u>d</u> d Group	
Add Files to Group 'Source Group 1'	//turn on LED
Remove Group 'Source Group 1' and its Files	//turn off LED
Anage <u>c</u> omponents	
Show Include File Dependencies	
23 { 24 a = a>>1;	
Build Output	

Step 15: It displays some window there select the file you have to add and click on ADD option.

[Advanced Embedded System Lab]						
🕱 demoproj - µVision4						
<u>File E</u> dit <u>V</u> iew <u>P</u> roject Fl <u>a</u> sh <u>D</u> eb	<u>Elle Edit View Project Fla</u> sh <u>D</u> ebug Peripherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp					
: 🗋 🚰 🖟 🥥 🕺 🕹 🗠 🖓 🖓	· (4) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2					
Project - 4 X						
Target 1 Source Group 1	01 #include <lpc214x.h></lpc214x.h>					
Startup.s	Add Files to Group 'Source Group 1'					
	Look in: 🔁 demo					
	in abc	Pat.				
Build Output	File name: abc Files of type: C Source file (*.c)	Add Close				

Step 16: The file will be added to our target and it shows in the project window.

🕱 demoproj - µVision4		
<u>File Edit View Project Flash D</u>	ebug Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u>	elp
i 🗋 😂 🗔 🥥 l X 🗈 🛍 l 🤊 r	(今今 魯魯魯魯 律律 [🛎 💽 کې د 🍳 🔍 🖉 🐛 🖉
] 🧐 🔛 🛗 🥔 🔣 🙀 Target 1	🖻 🛣 🔚 🔁	
Project 👻 🕈 🗙	abc.c	
🖃 🔚 Target 1	01 #include <lpc214x.h></lpc214x.h>	
E 🔄 Source Group 1	02 void delay(int);	
Startup.s	03 int main()	
abc.c	04 🗐 (
4.5	05 unsigned int a, i;	
	06 PINSELO = 0x00000000; //	enable all the pins as GPIO
	07 PINSEL1 = 0x00000000;	
	08 PINSEL2 = 0x00000000;	
	09 IODIR1 = 0x000F0000; //	' Set P1.16, P1.17, P1.18, P1.19 as Output
	10 a=0x00010000;	
	11 while (1)	
	12 (
	13 //IOSET1 = 0x000F0000;	
	14 for (i=0;i<=3;i++)	
	15 (
	16 IOSET1 = a; /	//turn on LED
	17 delay(10);	
	18 IOCLR1 = a; /	//turn off LED
	19 delay(10);	
	20 a = a<<1;	
	21)	
	<pre>22 for (i=0; i<=3; i++)</pre>	
	23 (
	24 a = a>>1;	
Ruild Output		
Build Output		

Step 17: Now give a right click on target in the project window and select "Options for Target".



Step 18: It will show some window, in that go to output option and choose Create Hex file option by selecting that box.



Step 19: In the same window go to Linker option and choose Use Memory Layout from Target Dialog by selecting the box, and click OK.

💟 demoproj - µVision4		
<u>File Edit View Project Flash Debug</u>	Peripherals Tools SVCS Window Help	
i 🗋 📂 🖼 🍠 l X. 🗈 🙉 l 🖉 🖉 i		- 🔍 🖉 🧶 💿 🔿 🔗 💽 - 🔍
: 😂 🏦 🎮 🧼 📇 📴 Target 1	• 🔊 🛔 🗟	
Project 👻 🕂 🗙	+] abcc	
E Target 1		
🖻 🔄 Source Group 1 Option	s for Target 'Target 1'	
Startup.s		
tevice termination in the termination of termination	Target Output Listing User C/C++ Asm Linker Debug	Utilities
	Ise Memory Layout from Target Dialog	
r	Make BW Sections Position Independent	0.00000000
i i i i i i i i i i i i i i i i i i i	Make BD Sections Position Independent	
	R/W Base R/W Base	0x40000000
1	Z Devent les tandard Elorates disable Warnings:	
	Heport might rail Conditions as Errors	
	catter	
	File	E OIt
	Misc	
č		
	Linker	rw-base 0x40000000strictautoat - 📉
	controlinfo sizesinfo totalsinfo unusedinfo veneers	
Build Output	sound 1	
Build target 'Target 1'		
assembling Startup.s	OK Cancel Def	faults Help
compiling abc.c		
linking		
Program Size: Code=660 RO-data	=16 RW-data=U ZI-data=1256	

"demoproj.axf" - O Error(s), O Warning(s).

Step 20: Now to Compile your project go to Project select Build Target option or press F7 or click on (Build) icon or (Build All) icon.

Edit View	Pro	ject Flas	<u>D</u> ebug	Peripherals	Tools	<u>s</u> vcs	Window	Help			
🗃 🗐 👩 I		New µ <u>V</u> is	ion Project.								
		New Mult	i-Project <u>W</u>	orkspace							
t		Open Pro	ject								
Target 1		<u>Close Pro</u>	ect								
Source G	re	Export									•
abc.c		<u>M</u> anage								•	
		Select De	vice for Tar	get 'Target 1'							hing as CRTO
		Remo <u>v</u> e F	ile 'abc.c'								pins as GPIU
	1	O <u>p</u> tions f	or File 'abo						Alt	t+F7	
		Clean tar	get								17, P1.18, P1.19 as Output
		<u>B</u> uild targ	let							F7	
		<u>R</u> ebuild a	ll target file	es							
	-	Batch Bui	ld								
	۲	Tr <u>a</u> nslate	D:\demo\al	bc.c					Ctrl	I+F7	
		Stop b <u>u</u> il	đ								
	-	1 D:\dem	o\demopro	j.uvproj							
		2 C:\Docu	ments and	Settings\All U	Jsers\Do	cuments	led\ledb.uv	/proj			
		3 C:\Docu	ments and	Settings\strin	g\Deskt	op\Spect	rum analys	er(keil)\sp	ectrum.uvpro	oj	
		4 \\10.0.0	130\Docun	nents\Real tim	e data(K	eil)\Sour	ce\Real_tim	e.uvproj			
		5 C:\Docu	ments and	Settings\strin	g\Deskt	op\RFID\	ARMrfid.uv	proj			
Jutput		6 C:\Docu	ments and	Settings\strin	g\Deskt	op/glcd i	new workin	ig final∖gl	cdarm.uvpro	Ú	
Jucput		7 C:\Docu	ments and	Settings\strin	ig\Deskt	op\touch	n dwnld\dn	ldglcd.uv	proj		
		8 C:\Docu	ments and	Settings\strin	ig\Deskt	op\add\a	ad.uvproj				
		<u>9</u> E:\anil\a	irm_main_p	roj_data\glcd	new\glo	darm.uv	proj				
		10 C:\Doc	uments an	d Settings\All	Users\D	ocument	s\adc_code	ADC.uvp	roj		

Step 21: In the build OUT PUT window you can see the errors and warnings if there exits. And here Your project Hex file will be created.

🕱 demoproj - µVision4		
<u>File Edit View Project Flash D</u>	ebug Peripherals Tools SVCS Window	Help
0 0 0 0 X 1 0 0 1 X 1 0 0 1 0 0	~~ 四四四四二十二	🖄 🚽 🔍 🖉 🖉 🖉 🖉
i 🥸 🖄 🕮 🧼 洪 i 🙀 i Target 1	• 🔊 🛔 🗟	
Project 👻 🕈 🗙	abc.c	
Image: 1 Image: Source Group 1 Image: Startup.s Image:	<pre>01 #include <lpc214x.h> 02 void delay(int); 03 int main() 04 { 05 unsigned int a,i; 06 PINSEL0 = 0x00000000; 07 PINSEL1 = 0x00000000; 08 PINSEL2 = 0x00000000; 10 a=0x00010000; 11 while(1) 12 { 13 //IOSET1 = 0x000F0000; 14 for(i=0;i<=3;i++) 15 { 16 IOSET1 = a; 17 delay(10); 18 IOCLR1 = a; 19 delay(10); 20 a = a<<1; 21 } 21 for(i=0;i<=3;i++) 23 { 24 a = a>>1; 4</lpc214x.h></pre>	//enable all the pins as GPIO // Set P1.16, P1.17, P1.18, P1.19 as Output //turn on LED //turn off LED
Build Output		
Build target 'Target 1' linking Program Size: Code=932 RO-c FromELF: creating hex file. "demoproj.axf" - 0 Error(s)	data=16 RW-data=0 ZI-data=1256 , 0 Warning(s).	

DEBUGGER:

We can debug and also simulate the program using debugger option, just click on

icon or can also select fron menu bar go to Debug option and in that the first one "Start/Stop debug".

When you click on that it will show you one window as shown below, click ok.



After clicking OK it will enter simulation screen as show below,



The above figure consists of 5 windows as shown in figure

Window 1: This window is called as REGISTER window.

Window 2: This is ASSEMBLY window, where we can see assembly code for

our corresponding C code.

Window 3: This is Editor Window where we can write our C code or Edit it.

Window 4: This is serial communication output window where we can see any output

from ARM on serial terminal.

Window 5: This window is Call Stack where we can see the local variables updating which is store in stack memory.

These are the default windows we can see other windows which is present in debugger are shown in figure below

Real_time - pV	lision4		. 🛛 🛛
Ele Edit View	Project Flash	h Debug Perjaherals Tools SVCS Window Help	
i 🗋 🗃 🗃 🌒 👘	102 -	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	
2 0 B	0.0.0		
Registers	¥ 0	X Disassembly	▼ 3 ×
Register	Value	41: (~
Current		= \$20x0000050 E32D4010 STMDB	
RD P1	0x40001500	43:	
- R2	0x80000000	0x000005E4 E59F0074 LDR RD,[PC,#0x0074]	
R3	0x00005550	0x00005E8 E59F1066 LDR R1,[PC,#0x0068]	
R4	0x00000000	44: Initolio:	
R6	0x40001900	0x000005F0 EBFFFFE4 BL Initpl1(0x00000568)	
87	0x00000000	45: InitUart0():	
RB	0x00000000	CV000005F4 FB0000C5 B! TwitHert0(0v00000614)	>
R9	0x00000000	Processor Province Processor Draw as Draw day Draw as Draw and Processor Processor Processor Processor	
- R11	0x00000000	1 20 A ALEM L MANY C 400 M L 40 ALEM	
R12	0x000005e0	U33 US_VUENT SEED;	-
R13(SP)	0x40002000	035 OS EVENT *sem3;	
H14(LH)	0x000001c0	036 INTBU status_sem1;	
E CPSR	0x600000df	U3/ INTRO status semi:	80
E SPSR	0x000000#		
* User/System		(40) int main (void) Resident Come Size Limit Six	
E Interupt			
E Supervisor		042 Plasteb = UX000300051 //using ducl.1 ### Restricted Version with 32768 Byte Code	
Abort Abort		044 Initpl1(): *** Currently used: 21956 Bytes (674)	
Lindefined		045 InitUart0();	
PC \$	0x000005E0	046 initid();	_
Mode	System	044 Internet(), The set time data acquisition n	
States	6937 0.00015015	049 CGInit();	
i sec	CHOCHOOD D	050 OSTaskCreate(TestTask1, (void *)0, sTestTask1, ASSIGN BreakDisable BreakEnable BreakKill	
		(b) OSIASKLTEATE(lestlask2, (VOLG ')), Alestlask2stack[lask5tklengn-1],]); (57) OSTASKLTEATE (TextTask3, (VOLG ')), AlestTask3Stack[Task5tklengn-1],]);	
		053 OSStart();	
		054 return 0:	
Symbols		x Mesory 1 x Watch 1 x Coak	×
Mask: *	□ □ Case S	Serolive D. Address (Dx400000000 Value Value Value Value	
Name	Address	Ture	
* VI Sinulator V		. 0x00000008: 18 F0 9F E5 18 F0 9F E5	13
🕂 🛃 Peipheral		L 0x00000010: 18 F0 9F E5 80 5F 20 B9	
E E Loc214K_u	2	Applesion 0.000000018: F0 F7 1F E5 18 F0 FF E5 4	13
÷ □ LCD	2	Mode L. 0000000025: 04 53 00 04 45 00 00 04	
🕀 🛄 nan		Module 1,0x00000030: 48 00 00 00 00 00 00 00	
⊕ [] 05_C		Modde0x00000038: 00 00 00 4C 00 00 00	
⊯⊟ 0€"cb∵	1		
1		000 3	
		070 Tx string("Semaphorel not created");	-1
	sters		· ·
		Real-Time Agent: Target Stopped Simulation t1: 0.00015015 sec CAP NUM SC	R. DVR. R.W.
H start		🖄 bel der - Menselt 💦 Shared Fanneerts 👘 Staal here - storest 👘 Staal - here 14 - a. 😪 ers here - Store - Stores de la Staal - Store - Stores de la Staal - Store - Stores de la Staal - Stores d	12:17 PM
- and -	and the second		

Window 1 is Command window which will display any errors or warning when we build the file.

Window 2 is Symbol window where we can see each and every variables of .c files like lcd.c, main.c etc

Window 3 is a memory window where we can see values stored in a

particular memory.

Window 4 is Watch window where we can see run time changes in the variables similar to symbol window but in symbol window we cannot see Global variables, in this window we can all variables.

Window 5 is a local window in this we can see function's local variables which are having local scope.

Execution of program:

For executing the program we need to press this Icon or we can directly press F5 function key from keyboard. This execution will be done all at a time we can also execute the program step wise by pressing F11 or clicking on this icon is for "step one line", for step over the current line click on **O** icon or press F10 function key, to step out of the current function press ctrl+F11 or click on icon.

Downloading Hex file onto ARM microcontroller:

Flash Magic Tool

To program the Microcontroller, Flash Magic tool is used. Generally, the microcontroller is in one of the two modes. One is RUN mode and the other is PROGRAMMING mode. In RUN mode microcontroller executes the application present in the microcontroller flash memory. In PROGRAMMING mode, microcontroller programs its flash memory in synchronization with Flash Magic.

To enter in to the programming mode, Hold down SW2(isp) and SW3(reset), then release SW3 first and finally SW2. To enter in to Run Mode, press the SW3(reset) after programming is over.

籋 Flash Magic	- NON PRODUCTION US	E ONLY	
File ISP Option	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 (0x000000-0	x000FFF)
COM Port:	COM 1	Erase block 1 (0x001000-0 Erase block 2 (0x002000-0	x002FFF)
Baud Rate:	9600	Erase block 3 (0x003000-0 Erase block 4 (0x004000-0	x003FFF)
Interface:	None (ISP)	Erase block 5 (0x005000-0	<u>x005FFF1 </u>
Oscillator (MHz):	12	Erase all Flash+Code Rd Erase blocks used by He	l Prot ex File
Step 3 - Hex File Hex File: C:\Doc Modified	uments and Settings\All Users' d: Saturday, September 3, 2011	\Documents\ARM.cd1&2\a\ , 6:47:43 PM <u>more info</u>	Browse
Step 4 - Options		Step 5 - Start!	
Verify after prog Fill unused Flasl Gen block chec Execute	ramming h sksums	Sta	art
Microcontrollers fro	om NXP Semiconductors Main	web page at:	
www.nxp.com/pro	ducts/microcontrollers		
		0	

Downloading Hex file onto microcontroller:

To program the flash memory, first keep the microcontroller in PROGRAMMING mode. Launch the Flash Magic Tool. Select the COM1, Baud rate as 19200, device as LPC2148; Oscillator Freq (MHz) as 12, in Communication block. Select the box erase all Flash + Code Rd Prot in Erase block. Select the box Verify after programming in Options Block. Select the hex file in Hex File block. Hold down SW2 (isp) and SW3 (reset), then release SW3 first and finally SW2 .Then click Start Button in Start Block.

Experiments: 2 & 3

Arithmetic and Branching Operations

Aim:

To verify arithmetic and branching operations by writing an assembly language program.

Tools:

(1)PC

(2)Keil Microvision4

Program:

AREA Arithmetic, CODE, READONLY

ENTRY

start

MOV	r0, #2
MOV	r1, #5
ADD	r2, r0, r1
MUL	r3, r0, r1
SUBGT	r4, r0, r1
SUBLT	r5, r1, r0
CMP	r1, #0
MOV	r8, r0
BNE	divd

here MOV r8, r7

stop B stop

divd

SUB r8, r8, r1

ADD r7, #1 CM r8, #0 P BEQ here BNE divd

END

Output:

<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project Flash	<u>D</u> ebug Pe <u>r</u> ipherals	<u>I</u> ools <u>S</u> VC	S <u>W</u> indow	<u>H</u> elp		
i 🗋 💕 🔛 🥔 🗌	X 🗈 🔁 🔊 🖞	PH 🗢 🚽 🔽	anana 10	E ∰E //≞ //∰	2	- 🗟 🦸	n 🔍 🕯
i 🎼 🗐 🖓	0⁺ () * ⁴ () ♦		3 👼 • 🔳 •	- 🔜 -	💷 • 🛒 • 🎌 • 🔜 •		
Registers	▲ ġ ×	Disassembly					
Register	Value 🔺	19:	SUB	r8, r8,	, r1		
Current		0x0000002C	E0488001	SUB	R8,R8,R1		
R0	0x00000002	20:	ADD	r7, #1	Contractor Description		
B1	0x00000005	0x00000030	E2877001	ADD	R7,R7,#0x00000001		
B2	0x00000007	21:	CMP	r8, #0			
R3	0x0000000a	0x00000034	E3580000	CMP	R8,#0x0000000		
B4	Oxffffffd	22:	BEQ	here			
R5	0x00000000	0x00000038	OAFFFFF9	BEQ	0x0000024		
R6	0x00000000	23:	BNE	divd			
B7	0x00000002	C x 0000003C	1AFFFFFA	BNE	0x0000002C		
R8	0xfffffff8	000000040	0000000	ANDRO	חם חם חם		
R9	0x00000000	1200					
R10	0x00000000	arithmatic	.s				
B11	0x00000000	17					
B12	0x00000000 —	19 dive					
	0x00000000	10 01100	SIIB *	8 18 11			
B14 (LB)	0x00000000	20	30D T	7 #1			
R15 (PC)	0x0000003c	20	(1)(D) 1	9 #O			
. E CPSR	0xa00000d3	21	BEO h	0 , 0			
E SPSR	0x00000000	22	BNF 4	lize			
	200 - 201 -	24 FND	DHE U	1140			
E Fast Interrupt	-	24 END					
📴 Project 📰 Regi	sters						
Command		R				▼ ₽ ×	Call Stack
Running with (Code Size Lim	nit: 32K				~	Stook Free
Load "G:\\ECA	D-VLST Lab Ma	nual\\cvcle1	nrograms o	utnuts\\ar	ithmatic.AXF"		Stack Fra
acas or those	· · · · · · · · · · · · · · · · · · ·	manual logorer	brodromo o	ashaon () at	· · · · · · · · · · · · · · · · · · ·		

Result:

Experiment: 4

Configuration and Controlling Of GPIO Ports

Aim:

To write an assembly level program for Configuring and Controlling of GPIO ports

Tools Used:

(1) PC

(2)Keil microvision4

Program:

AREA myprogram, CODE, READONLY

ENTRY

PINSEL0 EQU 0xE002C000

PINSEL1 EQU 0xE002C004

PINSEL2 EQU 0xE002C014

IODIR0 EQU 0xE0028008

IOSET0 EQU 0xE0028004

IOCLR0 EQU 0xE002800C

IODIR1 EQU 0xE0028018

IOSET1 EQU 0xE0028014

IOCLR1 EQU 0xE002801c

MOV r0, #0 LDR r1, =PINSEL0

STR r0, [r1]

MOV r0, #0 LDR r1, =PINSEL1 STR r0, [r1]

MOV r0, #0 LDR r1, =PINSEL2 STR r0, [r1]

MOV r0, #0xFFFFFFFF LDR r1, =IODIR0 STR r0, [r1]

MOV r0, #0xFFFFFFFF LDR r1, =IODIR1 STR r0, [r1] stop

MOV r0, #0xFFFFFFFF LDR r1, =IOSET0 STR r0, [r1]

MOV r0, #0XFFFFFFFF LDR r1, =IOSET1 STR r0, [r1]

MOV r3, #0 BL DELAY MOV r3, #0 MOV r0, #0xFFFFFFF LDR r1, =IOCLR0 STR r0, [r1]

MOV r0, #0XFFFFFFFF LDR r1, =IOCLR1 STR r0, [r1]

MOV r3, #0

BL DELAY

MOV r3, #0

B stop

DELAY

LDR r5, =0xFFFFFF

;MOV r4, #255

HERE

CMP R3, R5

;HERE_1

;CMP r4, r6

;ADD r6, #1

;BNE HERE_1

ADD R3, #1

BNE HERE

BX LR

END

Output:

Set:

- GPI00	- 31 Bits	24	1 23	Bits	16	15	Bits	8 7	Bits	0
100DIR: 0x0000000F					Ē	ΓΓΓ		ר רח		~~
100SET: 0x00000000				ТГГГ						
100CLR: 0x00000000	ГГГГГ					ГГГ				
IOOPIN: 0xF2FFFFF0			যথ্য	নিম্ব	মান	বিত্র	ঘৰ্ষ	াম্য য		
Pins: 0xF2FFFFF0	-	V	~~~		~~	~~~	~~~~	~ ~ ~	~~	

Clear:

GPIO0	- <u>31 B</u>	lits	24 23	Bits	16	15	Bits	8	7	Bits	0
005ET: 0x0000000F										মনান	-
00CLR: 0x00000000	ГГГГ				ГГ	ГГГ			ГГГ	ГГГ	ГГ
IOOPIN: 0xF2FFFFFF	ঘ্যম্য		াবন ন	বিত্রবিত্	ঘত	নবা	ালবাল	য	ঘ্যম	বিব	নন
Pins: 0xF2FFFFFF	~~~~	~	~~	~~~~	~~	~~~		~~	~~~	~~~	~

Results:

Experiment: 5 & 6

Timer Subroutine and External Value input in ARM7

Aim:

To write program to demonstrate time delay using built in timer/counter features on IDE environment.

Tools:

```
(1)PC(2)keil microvision4
```

Program:

```
#include <LPC214X.H>
void timer_delay(void);
```

int main()

```
{
```

```
PINSEL0 = 0x00000000;
IODIR0 = 0xFFFFFFFF;
while(1)
```

{

```
IOOSET = 0x55;
timer_delay();
IOOCLR = 0x55;
timer_delay();
IOOSET = 0xAA;
timer_delay();
IOOCLR = 0xAA;
timer_delay();
```

}

}

```
void timer_delay()
```

{

T0TCR = 0x02;

T0TC = 0;

TOPC = 0;TOPR = 15000;

```
T0MR0 = 1000;
```

```
T0MCR = 0x02;
```

T0TCR = 1;

while(T0TC != T0MR0);

}

Output:

Set:

General Purpose Inpu	t/Output 0 (GPIO 0) - Slow Interface 🛛 🛛 🔯
GPIO0 IOODIR: 0xFFFFFFFF	31 Bits 24 23 Bits 16 15 Bits 8 7 Bits 0 VVVVVVVV VVVVVVV VVVVVVVV VVVVVVVV VVVVVVVV VVVVVVVVV VVVVVVVVV VVVVVVVVV VVVVVVVVVV VVVVVVVVVVV VVVVVVVVVVV VVVVVVVVVVVVV VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
100SET: 0x00000000	
100CLR: 0x00000000	
100PIN: 0x00000000	
Pins: 0x70000000	

Clear:

IOODIR: 0xFFFFFFFF	- 31	Bits	24	23	Bits	16 VV	15	Bits	8	7	Bits	
00SET: 0x00000055	ГГГГ			ГГГ	ГГГ	ГГ	ГГГ	ГГГ	ГГ		-	
00CLR: 0x00000000				ГГГ	тгг	ГГ	ГГГ	ГГГ	ГГ	ГГГ	ггг	ГГ
100PIN: 0x00000055				ГГГ	ГГГ	ГГ			ГГ		বাৰ	
Pins: 0x70000055		v 1		ГГГ	ГГГ		ГГГ		ГГ		~~~	

Timer0:

Timer 0			X
Prescaler PR: 0x00003A98 PC: 0x00000000	Timer TCR: 0x00000001 TC: 0x00000000	Enable	Interrupt Register
Match Channels MCR: 0x00000002 MR0: 0x000003E8	EMR: 0x00000000	MR2: 0x00000000	MB3: 0x0000000
Interrupt on MR0 ✓ Reset on MR0 ✓ Stop on MR0 EMC0: Nothing ▼	Interrupt on MR1 Reset on MR1 Stop on MR1 EMC1: Nothing - E	Interrupt on MR2 Reset on MR2 Stop on MR2 EMC2: Nothing •	Interrupt on MR3 Reset on MR3 Stop on MR3 EMC3: Nothing
External Match 0	External Match 1 MR1 Interrupt	External Match 2 MR2 Interrupt	External Match 3 MR3 Interrupt
Capture Channels CCR: 0x00000000 CR0: 0x000000000	CR1: 0x00000000	CR2: 0x00000000	CR3: 0x00000000
Fising Edge 0 Falling Edge 0 Interrupt on Event 0 CAP0.0 CR0 Interrupt	Rising Edge 1 Falling Edge 1 Interrupt on Event 1 CAP0.1 CR1 Interrupt	Rising Edge 2 Falling Edge 2 Interrupt on Event 2 CAP0.2 CR2 Interrupt	Rising Edge 3 Falling Edge 3 Falling Edge 3 Interrupt on Event 3 CAP0.3 CR3 Interrupt
Count Control CTCR: 0x00000000	Mode: Timer	Counter	Input: CAP0.0 💌

Result:

Experiment: 7

Simple Interrupt Handler with Time Delay

Aim:

To write a program to demonstrate a simple interrupt handler and setting up timer features on IDE Environment.

Tools:

```
(1)PC
(2)Keil microvision 4
```

Program:

#include <LPC214X.H>

```
void Timer_0_Init(void);
void Timer_0_ISR(void) __irq;
unsigned char g_flag = 0;
int main()
```

```
{
```

}

{

```
PINSEL0 = 0x0000000;
      PINSEL1 = 0x0000000;
      PINSEL2 = 0x0000000;
      IO0DIR = 0x000F0000;
      Timer_0_Init();
      T0TCR = 0x01;
  while(1);
void Timer_0_Init(void)
      T0TCR = 0x02;
      T0TC = 0x00;
      TOPR = 15000;
      TOPC = 0;
      TOMR0 = 1000;
```

```
TOMCR = 0x03;
```

```
[ Advanced Embedded System Lab]
       VICIntSelect |= 0x0000000;
       VICVectCntl0 \mid = 0x20|4;
VICVectAddr0 = (unsigned)Timer_0_ISR;
VICIntEnable |= (1<<4);
}
void Timer_0_ISR(void) __irq
{
      if(VICIRQStatus & 0x00000010)
       {
if(T0IR == 0x01)
              {
                    T0IR = 0x01;
      if(g_flag == 0)
       {
         IO0SET = 0x00003F00;
         g_flag = 1;
       }
      else
       {
         IOOCLR = 0x000003F00;
         g_flag = 0;
       }
     }
VICVectAddr = 0x00000000;
}
```

Output:

Reset:

General Purpose Inpu	it/Output 0 (GPIO	0) - Sla	w Inte	rfa	ce					D
GPI00 100DIR: 0x000F0000	31 Bits	24	23	Bits	16	15	Bits	8	7	Bits	_0
100SET: 0x00000000			ГГГ		Т	ГГГ		Т	ГГГ	ГГГ	
100CLR: 0x00000000			ГГГ		Т	ГГГ		Т	ГГГ	ГГГ	
IOOPIN: 0x82F0FFFF	וחחחק	77	ঘ্যম	VLL	-	যথ	নবব	যাহ	অঅঅ	অঅঅ	ঘ্য
Pins: 0xF2F0FFFF	-	•	~~~	V	Г	~~~		~~		-	~~

Enable:

General Purpose Inpu	t/Output 0 (GPIO 0) - Slow Interface	<
GPI00 IO0DIR: 0x000F0000	31 Bits 24 23 Bits 16 15 Bits 8 7 Bits 0	1
100SET: 0x00000000		
100CLR: 0x00000000		
IOOPIN: 0xF2F0FFFF	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
Pins: 0xF2F0FFFF		

VIC reset

Channel	Sour	ce	Name	Туре	Vector	IntEnable	RawInt	1
0	Wate	:hdog	WDINT	IRQ	00000000	H O	0	
1	SWI	nterrupt		IRQ	0000000	IH O	0	
2	DbgC	CommRx		IRQ	00000000	H O	0	
3	Dbg0	CommTx		IRQ	00000000	H O	1	
4	Time	r O		IRQ	0000000	H O	0	
5	Time	r1.		IRQ	0000000	H O	0	-
6	UAR	ТО		IRQ	00000000	H O	0	
2	UAR	11		IHU	00000000	H U	U	
8	PWM	1		IRQ	00000000		U	
3	CDIO				00000000	н U U 0	0	
11	SSP			IRQ	00000000	H 0	0	
☐ IntE	inable tInt	IntSele	^{ct} IR	Q Slot:	v	CDefVectAddr:	0×00000000)
VICVec	tAddr:	0x00000000	VICInt	Select:	0x00000000	VICRawIntr:	0x00000008	3
VICS	oftInt:	0x00000000	VICIntE	nable:	0x00000000	VICIRQStatus:	0x00000000)
VICSoftInt	Clear:	0x00000000	VICIn	tEnClr:	0x00000000	VICFIQStatus:	0x00000000)
								-

VIC enable

Vectored	Interrupt Contr	oller (Vi	c)				×
Channel	Source	Name	Туре	Vector	IntEnable	RawInt	~
0	Watchdog	WDINT	IRQ	00000000	DH 1	0	
1	SW Interrupt		IRQ	0000000	OH O	0	
2	DbgCommRx		IRQ	00000000	OH O	0	
3	DbgCommTx		IRQ	00000000	OH O	1	
4	Timer 0		IRQ 0	00000128	3H 1	1	
5	Timer 1		IRQ	00000000	OH O	0	
6	UARTU		IRU	00000000	JH U	U	
6	DARTI		IRQ	00000000	JH U	U	
0				00000000	ישר או חים ה	0	
10	SPIN		IRO	00000000	יה ט וע ח	0	
11	SSP		IRQ	00000000	он о	0	~
Selected I	Interrupt: Watchdog nable 🔽 IntSele Int 🔽 RawIn	t ^{ict} IR	Q Slot:	• V	ICDefVectAddr:	0×00000000	0
VICVect	Addr: 0x00000128	VICInt	Select: Oxi	0000000	VICRawIntr:	0x00000018	3
VICS	oftInt: 0x00000000	VICIntE	nable: 0xl	00000011	VICIRQStatus:	0x00000010)
VICSoftInt	Clear: 0x00000000	VICIn	tEnClr: Ox	00000000	VICFIQStatus:	0x00000000)
					VICProtection:	0x00000000)

Result:

Experiment: 8

Interrupt Handler Aim:

To Write a Program to demonstrate Setting up interrupt handlers and generate an interrupt when button is pressed.

Tools:

(1) PC

(2) Keil microvison 4

Program:

```
(a)extinterrupthandler.c
```

```
#include <LPC214X.H>
```

```
#include "Exp 7.h"
```

```
void ExtInt_Init(void);
```

```
void ExtInt_Service(void)__irq
```

{

```
UART_0_Send_String("External Interrupt has arrived \r\n");
```

EXTINT $\mid= 4;$

VICVectAddr = 0;

}

```
void ExtInt_Init(void)
```

{

EXTMODE $\mid=4;$

EXTPOLAR = 4;

VICVectCntl0 = 0x20 | 16;

VICVectAddr0 = (unsigned long) ExtInt_Service; VICIntEnable |= 1<<16;

```
[ Advanced Embedded System Lab]
```

```
}
```

{

```
int main()
```

PINSEL0 |= 0x80000005;

```
PINSEL1 = 0x00000000;
```

ExtInt_Init();

```
UART_0_Init();
```

UART_0_Send_String("* External Interrupt Demo

```
(n\langle r''); DelayMs(100);
```

```
DelayMs(100);
```

```
while(1);
```

}

(b)Exp7.h

#define UART_0_BPS 115200

#define Fpclk 15000000

void UART_0_Init(void)

```
{
```

unsigned int Baud16;

U0LCR = 0x83;

Baud16 = ((Fpclk / 16)/ UART_0_BPS);

U0DLL = Baud16 % 256;

U0DLM = Baud16 / 256;

U0LCR = 0x03;

```
[ Advanced Embedded System Lab]
```

```
}
void UART_0_Send_Char(unsigned char value)
{
    U0THR = value;
while((U0LSR & 0x40 ) == 0);
}
void UART_0_Send_String(unsigned char *data)
{
    while(*data != '\0')
        UART_0_Send_Char(*data++);
}
void DelayMs(unsigned int count)
{
    unsigned int i,j;
    for(i=0;i<count;i++)
    {
        for(j=0;j<1000;j++);
    }
}</pre>
```

Output:



[Advanced Embedded System Lab]	
🛃 Terminal v1.6 by Bray++	
Connect COM Port Baud rate Data bits Parity Stop Bits Handshaking Disconnect © COM1 © 300 © 4800 © 38400 © 5 © none © 1 © none Disconnect © COM2 © 600 © 9600 © 56000 © 6 © odd © RTS/CTS About © COM4 © 2400 © 19200 © 115200 © 7 © mark © 2 © RTS/CTS + X0N/X0FF	
Settings	
Receive CLEAR Reset Counter 13 \$ Counter = 4	
⊔şRRRRRRRRRRRRR™áŇ+Éą±0%ąŇ+ÉÉŐÁŇ⊡0+µ″⊡©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©	
Transmit CLEAR SET DTR CLR DTR CLR RTS Send File	
MRITS embedded systems laboratory	

Result:

Experiment: 9

Interface 8-Bit LED and Switch

Aim:

To Write a program to Interface 8-bit LED and Switch Interface

Tools:

(1)PC

(2)Keil Microvision 4

Program:

(a)LED.c

#include <LPC214X.H>

#include "Exp8.h"

unsigned int count =0;

int main()

{

PINSEL0 = 0x00000000;

PINSEL1 = 0x00000000;

IODIR0 = 0x00003F00;

```
IODIR1 = 0x00C00000;
while(1)
{
    if(SW1)
    {
        count = count+1;
        while(SW1);
//pattern_1_on;
        pattern_off;
    }
    if(count == 1)
    {
        IO0SET=0x00003F00;
        IO1SET=0x00C00000;
    }
}
```

```
}
else if(count == 2)
{
        IO0SET=0x00001200;
        IO1SET=0x00C00000;
}
else if(count == 3)
{
        IO0SET=0x00002D00;
        count=0;
}
}
```

(b)Exp8.h

#define pattern_1_on IO0SET=0x00003F00;IO1SET=0x00C00000;

#define pattern_1_off

IO0CLR=0x00003F00;IO1CLR=0x00C00000; #define pattern_2_on

IO0SET=0x00001200;IO1SET=0x00C00000; #define pattern_2_off

IO0CLR=0x00001200;IO1CLR=0x00C00000; #define pattern_3_on

IO0SET=0x00002D00; #define pattern_3_off

IO0CLR=0x00002D00;

#define pattern_off IO1CLR = 0x00C00000;IO0CLR=0x00003F00;

#define SW1 (IOPIN0&(1<<15))

Output:

Pattern: 1

GPID0 IOODIR: 0x00008F00	- 31 Bits	24	23	Bits	16 15	Bits	8 VIVIV	7	Bits
00SET: 0x00003F00			ГГГ		ГГ	<u> </u>	<u> অত্য</u>	ГГГ	
00CLR: 0x00000000		ГГГ	ГГГ	ГГГГ	ГГ			ГГГ	ТГГГ
IOOPIN: 0xF2FF7FFF	াস্বস্থ		ম্বর	নব্য	া য	াবব্য	যথ	অঅঅ	অঅঅঅ
Pins: 0xF2FF7FFF	-	1	~~~	~~~~		~~~~	~~~	~~~	

Pattern: 2

GPIOU IOODIR: 0x0000BF00	- 31 Bits	24	23	Bits	16	15 🔽	Bits	8 VVV	7	Bits	0
00SET: 0x00001200			ГГГ					<u></u>	ГГГ		
00CLR: 0x00000000											
IOOPIN: 0xF2FF52FF	ান্যব্যর্থ	- 기 기	বিব	াববাবা	যাত			<u> </u>	নবাব	বিবিধ	ঘত
Pins: 0xF2FF52FF	-	•	~~~		-				~~~		~~

Pattern: 3

- GPI00	31 Bits 24.23 Bits 16.15	Bite 8 7 Bite 0
100DIR: 0x00003F00	קיזן זידיזידייין דידיזידייי	าาาาาาาาา จัจจจัง
100SET: 0x00002D00		
IOOCLR: 0x00000000		
IOOPIN: 0x82FFEDFF		תקתקקקק קוקקי
Pins: 0xF2FFEDFF		

Result:

Experiment: 10

Buzzer Interfacing

Aim: To write a program to implement Buzzer Interface on IDE environment

Tools: (1) PC

(2) Keil Microvision 4

Program:

(a)buzzer.c

#include <LPC214X.H>

#include "Exp7.h"

int main()

```
{
```

PINSEL0 = 0x0000000;

PINSEL2 = 0x00000000;

IODIR0 = 0x00000800;

```
IODIR1 = 0x000F0000;
```

while(1)

```
{
```

if(SW1==0)

while (SW1 == 0)

BuZZer_ON;

else

BuZZer_OFF;

}

}

(b)Exp7.h

#define BuZZer_OFF IO0SET=(1<<11)</pre>

#define BuZZer_ON IO0CLR=(1<<11)

#define SW1 (IO0PIN&(1<<15))

Output:

Buzzer-off:

GPIO0

General Purpose Inpu	ut/Output 0 (GPIO 0) - Slow Interface	2
GPI00	- 31 Bits 24 23 Bits 16 15 Bits 8 7 Bits	0
100SET: 0x00000800		
100CLR: 0x00000000		
IOOPIN: 0x82FFFFFF	וסטסטטט מטטטטטט מטטטטט וטררררט	7
Pins: 0xF2FFFFFF		~

GPIO1

ID1DIR: 0x000F0000	- 31 Bits	24 23	Bits 1	6 15	Bits	8 7	Bits	0
01SET: 0x00000000					тгг			T
01CLR: 0x00000000						Г		T
IO1PIN: 0xFFF00000	অব্যব্য	াম্ম মহা				-г г		T

Buzzer-on

GPIO0

GPI00 IOODIR: 0x00000800	- 31 Bits	24	23 ГГГГ	Bits 1	6 15	Bits	8	7	Bits	0
00SET: 0x00000000			ГГГГ					ГГГ		
DOCLR: 0x00000000			ГГГГ					ГГГ	ГГГГ	Т
OOPIN: 0x82FFF7FF			নব্ব	াববাব	াবন হ	- যথ	ঘৰ্ষ	ব্যব	নবব	ঘচ
Pins: 0xF2FFF7FF	-	V	~~~		-	~~	~~~	~~~	~~~	~~

GPIO1

General Purpose Input/Output 1 (GPIO 1) - Slow Interface 🛛 🛛 🗙									
GPI01 I01DIR: 0x000F0000	31 Bits 24 23 Bits 16 15 Bits 8 7 Bits 0								
I01SET: 0x00000000									
I01CLR: 0x00000000									
IO1PIN: 0xFFF00000									
Pins: 0xFFF00000									

Result:

Experiment: 11

LCD Display Interfacing Aim:

To write a program for Displaying message in a 2 line * 16 characters LCD display and

verify the result in debug terminal.

Tools: (1)PC (2)Keil Microvision 4 Program:

(a)LCD.c:

#include <LPC214X.H>

#include "Exp6.h"

int main(void)

{

PINSEL0 = 0x0000000;

PINSEL1 = 0x00000000;

PINSEL2 = 0x00000000;

IO1DIR = 0x01C00000;

IO0DIR = 0x00003C00;

LCD_init();

LCD_Command(0x80);

LCD_Send_String(" UNISTRING TECH ");

LCD_Delay(10);

LCD_Command(0xC0);

LCD_Send_String("SOLUTION PVT LTD");

while(1);

}

(b)Exp6.h

#define LCD_RS_DATA IO1SET=(1<<24)
#define LCD_RS_CMD IO1CLR=(1<<24)
#defineLCD_WR IO1CLR=(1<<23)
#defineLCD_RD IO1SET=(1<<23)
#define LCD_EN_HI IO1SET=(1<<22)
#defineLCD_EN_LOW IO1CLR=(1<<22)</pre>

void LCD_Delay(unsigned int s)

{

```
unsigned int i = 0;
```

while(s-->0)

i = 0;

{

}

while(i++<60000);

}

void LCD_Data(unsigned int data)

{

unsigned char temp;

LCD_RS_DATA;

LCD_WR;

temp = data;

```
IO0SET = (temp & 0xF0)<<6;
```

LCD_EN_HI;

LCD_EN_LOW;

IO0CLR = (temp & 0xF0) <<6;

IO0SET = (data & 0x0F)<<10;

LCD_EN_HI;

LCD_EN_LOW;

IO0CLR = (data & 0x0F)<<10;

LCD_Delay(10);

}

```
void LCD_Send_String(unsigned char *data)
```

{

while(*data)

LCD_Data(*data++);

}

void LCD_Command(unsigned int data)

```
{
```

unsigned char temp;

LCD_RS_CMD;

LCD_WR;

temp = data;

IO0SET = (temp & 0xF0)<<6; LCD_EN_HI; LCD_EN_LOW;

```
}
```

Output:



Result:

Experiment: 12

Serial Communication Interfacing Aim:

To write a program to demonstrate serial driver's Transmission from kit and Reception From

PC using Serial Port on IDE environment.

Tools: (1)PC (2) Keil Microvision4

Program:

(a)serialdriver.c

#include <LPC214X.H>

```
void Tx_string(unsigned char *);
```

void InitUart0(void);

void main()

{

PINSEL0=0x05;

InitUart0();

Tx_string("Hello world\n");

while(1);

}

(b)serialdriver.c

//#include "LPC2148.h"

#include <LPC214X.H>

#define DESIRED_BAUDRATE 19200

```
[ Advanced Embedded System Lab]
#define CRYSTAL_FREQUENCY_IN_HZ 12000000
#define MAX_PCLK (CRYSTAL_FREQUENCY_IN_HZ*5)
#define PCLK (MAX_PCLK/4)
#define DIVISOR (PCLK/(16*DESIRED_BAUDRATE))
void InitUart0(void)
```

```
{
```

U0LCR=0x83;

VPBDIV=0x00;

U0DLL=DIVISOR&0xFF;

U0DLM=DIVISOR>>8;

U0LCR=0x03;

U0FCR=0x05;

}

```
void Tx_char(char ch)
```

```
{
    if (ch=='\n')
        {
            while (!(U0LSR&0x20)) { }
            U0THR='\r';
        }
        while (!(U0LSR&0x20)) { }
        U0THR=ch;
    }
```

unsigned char Rx_char(void)

```
char ch;
while (!(U0LSR&0x01)) { }
ch=U0RBR;
```

```
[ Advanced Embedded System Lab]
       return ch;
}
int Tx_string(char *s)
{
       int i=0;
       while(s[i]!='0')
       {
               Tx_char(s[i]);
               i++;
       }
       return(i);
}
void Tx_num(unsigned int num)
{
       char str[9];
       int i=0,temp;
 for(i=0;i<8;i++)
       temp=num%16;
   if(temp>9)
          str[7-i]=(0x37)+temp;
   else
           str[7-i]=(0x30)+temp;
   num=num/16;
  }
 str[8]='\0';
 Tx_string(str);
}
void Tx_num_dec(int num)
{
       char str[12];
       int i=0,temp,length;
 if(num<0)
  {
       num=-num;
   str[0]='-';
   i=1;
   length=11;
  }
 else
  {
       i=0;
   length=10;
 for(;i<length;i++)</pre>
```

```
[ Advanced Embedded System Lab]
{
    temp=num%10;
    if(temp>9)
        str[length-1-i]=(0x37)+temp;
    else
        str[length-1-i]=(0x30)+temp;
    num=num/10;
    }
    str[length]='\0';
    Tx_string(str);
}
```

Output:

Flash magic tool



Terminal

🛃 Terminal	v1.6 by Bra	1 y ++						
<u>Connect</u> Disconnect <u>A</u> bout Quit	COM Port COM1 COM2 COM3 COM4	Baud rate 300 600 1200 2400	 4800 9600 14400 19200 	 38400 56000 57600 115200 	Data bits C 5 C 6 C 7 G 8	Parity rone odd reven rmark rmark rmark	Stop Bits • 1 • 1.5 • 2	Handshaking none RTS/CTS XON/XOFF RTS/CTS + XON/XOFF
Settings Auto Dis/I	Connect 8	▼ Font S	ize 🥅 Tir	ne				
Receive CLEAR	Reset Coun	ter 13 🚖	Counter =	7				
Helio world Helio world Ş≪ë⊡é″ɱ'5)`He Helio world Helio world Helio world Helio world	ello world							

Result: