**EXPERMIMENT NO: 2**

**Subject: Embedded System Design**

**Subject code: ET 3271**

**Name:**

**Gr.No.:**

**Roll No.:**

**Class: TY Div : Batch:**

**Title :** 2 X 16 character LCD Interfacing

**Aim :**To interface 16X2 alphanumeric LCD to LPC 2148

**Objective** :To Interface 16 x 2 LCD character display to LPC2148 microcontroller in 8-bit mode and write an embedded C program to display a message on LCD

**Hardware**: LPC2148 Development Kit, PC

**Software** : Keil IDE microvision , Philips LPC2000 flash utility.

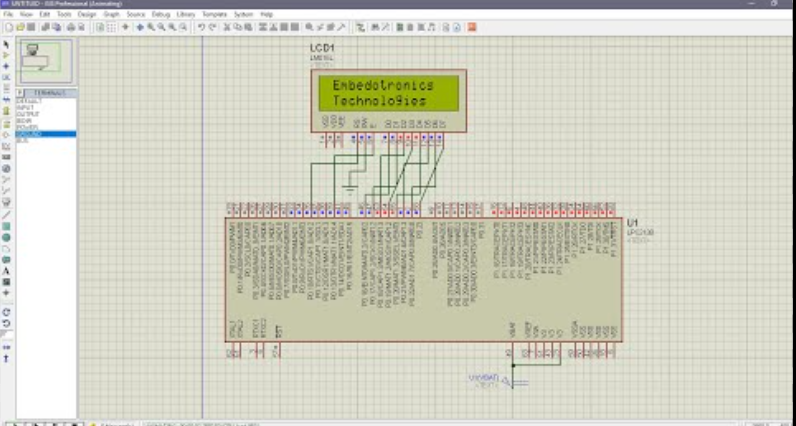
**Theory:**

2 X 16 character LCD is connected to the LPC 2148through the port1 . LCD is connected in the 8-bit mode. (data lines D0 to D7 ).

**Features of LCD:**

1. Easy interface.
2. Built-in Dot Matrix LCD controller with font 5X7 dots.
3. Display data RAM for 80 characters.
4. Character generator ROM, which provides 160, characters with font 5X7 dots and 32 characters with font 5X10 dots.
5. Both display data and character generator RAM can be read from the MPU.
6. Internal automatic reset circuit at power ON.
7. Built in oscillator circuit.

**LCD Interfacing Diagram: ( proteus simulation diagram)**

****

**LCD Pin assignment with LPC 2148:**

| | **16x2 Character LCD** | |  | | | --- | --- | --- | --- | |  |  | |  | | **LPC2148( Pins used)** | | **Description** | | --- | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D0 |  | LCD Data Bit 0 |
| D1 |  | LCD Data Bit 1 |
| D2 |  | LCD Data Bit 2 |
| D3 |  | LCD Data Bit 3 |
| D4 |  | LCD Data Bit 4 |
| D5 |  | LCD Data Bit 5 |
| D6 |  | LCD Data Bit 6 |
| D7 |  | LCD Data Bit 7 |
| RS |  | Register select |
| R/W |  | Read/Write |
| EN |  | enable |

**LCD commands used:**

| SR no | Command | Hex value |
| --- | --- | --- |
| 1 | Function set: 8-bit, 2line, 5 x 7 dots | 38h |
| 2 | Clear display | 01h |
| 3 | Entry mode | 06h |
| 4 | Display on cursor on | 0eh |
| 5 | Line address | 80h |

**Algorithm:**

1. Declare a function to send commands to initialize LCD in 8-bit mode.
2. Declare a function to send data to display on LCD.
3. Declare a function to send either data or commands in 8-bit format on data line D0-D7.
4. Declare a delay function.
5. Call all these function from main function in order to display desired message on LCD

**Procedure:**

* 1. Write C program using any available editor
  2. Compile and Link the program
  3. Make .hex file
  4. Download .hex file into the LPC2148 using LPC2000 flash utility.
  5. Execute the program on the target board and observe the displayed message on LCD

**Program**:

(Development board)

#include<lpc214x.h>

#include <stdint.h>

#include <stdlib.h>

#include <stdio.h>

#define LCD\_PORT 0x00FF0000

#define EN 1<<10

#define RS 1<<11

#define RW 1<<20

#define LCD\_SHIFT 16

void delay(unsigned int time)

{

int i,j;

for(i=0;i<100;i++)

for(j=0;j<100;j++);

}

void LCD\_strobe()

{

delay(100);

IOSET0 = EN;

delay(100);

IOCLR0 = EN;

delay(100);

}

void LCD\_data(unsigned char ch)

{

IOCLR1 = LCD\_PORT;

IOSET1 = ch<<LCD\_SHIFT;

IOSET0 = RS;

IOCLR0 = RW;

LCD\_strobe();

}

void LCD\_data\_down(unsigned char ch)

{

IOCLR1 = LCD\_PORT;

IOSET1 = ch<<LCD\_SHIFT;

IOSET0 = RS;

IOCLR0 = RW;

LCD\_strobe();

}

void LCD\_cmd(unsigned char ch)

{

IOCLR1 =LCD\_PORT;

IOSET1 = ch<<LCD\_SHIFT;

IOCLR0 = RS;

IOCLR0 = RW;

LCD\_strobe();

}

void LCD\_init(void)

{

PINSEL0 &= 0xFF0FFFFF;

PINSEL1 &= 0xFFFFFCFF;

PINSEL2 &= 0xFFFFFFF3;

IODIR0 |= RS | EN | RW;

IODIR1 |= LCD\_PORT;

LCD\_cmd(0x38);

LCD\_cmd(0x06);

LCD\_cmd(0x0C);

LCD\_cmd(0x01);

LCD\_cmd(0xC0);

//LCD\_data('Hello');

LCD\_data('S');

LCD\_data('o');

LCD\_data('u');

LCD\_data('r');

LCD\_data('j');

LCD\_data('a');

LCD\_data('d');

LCD\_data('i');

LCD\_data('p');

LCD\_data(' ');

LCD\_data('!');

}

int main()

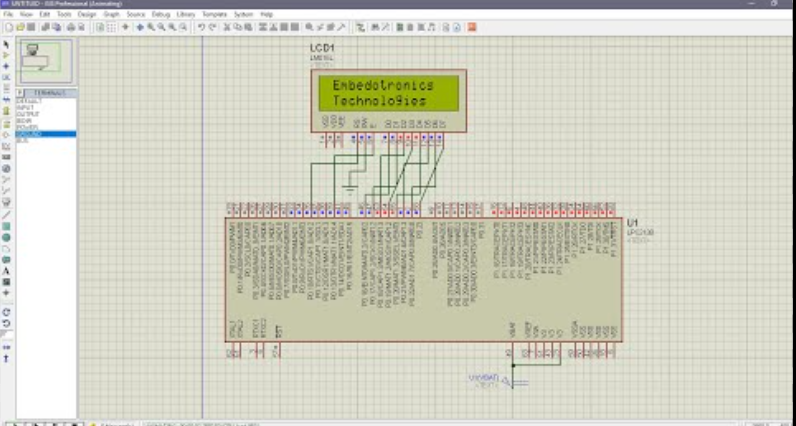
{

LCD\_init();

while(1);

}

**Output:** proteus simulation output

****

Development board Output



**Conclusion:**

The experiment which aimed to test the LPC2148 microcontroller's ability to interface with an LCD and display data, was a resounding success. The connection between the microcontroller and the LCD was established through a suitable interface and the appropriate control signals were sent to display the data. During the experiment, various steps such as initialization, writing data, and adjusting the LCD's display properties were executed with great results. This experiment highlights the importance of understanding LCD interfacing and the versatility of the LPC2148 microcontroller in working with different peripheral devices.