1. **To interface LCD with ARM processor ARM7TDMI/LPC2148. Write and execute programs in C language for displaying text messages and numbers on LCD**

////// "LCD DISPLAY" To display the predefined data Date:21/01/2012 //////

#include<lpc214x.h>

#include<stdio.h>

//Function prototypes

void lcd\_init(void);

void wr\_cn(void);

void clr\_disp(void);

void delay(unsigned int);

void lcd\_com(void);

void wr\_dn(void);

void lcd\_data(void);

unsigned char temp1;

unsigned long int temp,r=0;

unsigned char \*ptr,disp[] = "pda,",disp1[] = "cse";

int main()

{

IO0DIR = 0x000000FC; //configure o/p lines for lcd

IO0PIN = 0X00000000;

delay(3200); //delay

lcd\_init(); //lcd intialisation

delay(3200); //delay

clr\_disp(); //clear display

delay(3200); //delay

//........LCD DISPLAY TEST.........//

temp1 = 0x80; //Display starting address of first line 1 th pos

lcd\_com();

ptr = disp;

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

temp1 = 0xC0; // Display starting address of second line 4 th pos

lcd\_com();

ptr = disp1;

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

while(1);

} //end of main()

// lcd initialisation routine.

void lcd\_init(void)

{

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x20; // change to 4 bit mode from default 8 bit mode

wr\_cn();

delay(3200);

// load command for lcd function setting with lcd in 4 bit mode,

// 2 line and 5x7 matrix display

temp = 0x28;

lcd\_com();

delay(3200);

// load a command for display on, cursor on and blinking off

temp1 = 0x0C;

lcd\_com();

delay(800);

// command for cursor increment after data dump

temp1 = 0x06;

lcd\_com();

delay(800);

temp1 = 0x80; // set the cursor to beginning of line 1

lcd\_com();

delay(800);

}

void lcd\_com(void)

{

temp = temp1 & 0xf0;

wr\_cn();

temp = temp1 & 0x0f;

temp = temp << 4;

wr\_cn();

delay(500);

}

// command nibble o/p routine

void wr\_cn(void) //write command reg

{

IO0CLR = 0x000000FC; // clear the port lines.

IO0SET = temp; // Assign the value to the PORT lines

IO0CLR = 0x00000004; // clear bit RS = 0

IO0SET = 0x00000008; // E=1

delay(10);

IO0CLR = 0x00000008;

}

// data nibble o/p routine

void wr\_dn(void) ////write data reg

{

IO0CLR = 0x000000FC; // clear the port lines.

IO0SET = temp; // Assign the value to the PORT lines

IO0SET = 0x00000004; // set bit RS = 1

IO0SET = 0x00000008; // E=1

delay(10);

IO0CLR = 0x00000008;

}

// data o/p routine which also outputs high nibble first

// and lower nibble next

void lcd\_data(void)

{

temp = temp1 & 0xf0;

temp = temp ;//<< 6;

wr\_dn();

temp= temp1 & 0x0f;

temp= temp << 4;

wr\_dn();

delay(100);

}

void clr\_disp(void)

{

// command to clear lcd display

temp1 = 0x01;

lcd\_com();

delay(500);

}

void delay(unsigned int r1)

{

for(r=0;r<r1;r++);

}

**2.Design and develop an assembly program to interface 4\*4 matrix keyboard. Using ARM TTDMI / LPC2148.**

/\*Program to demonstrate keyboard operation Date:11/11/2011

Takes a key from key board and displays it on LCD screen\*/

#include<lpc21xx.h>

#include<stdio.h>

/\*\*\*\*\*\*\* FUNCTION PROTOTYPE\*\*\*\*\*\*\*/

void lcd\_init(void);

void clr\_disp(void);

void lcd\_com(void);

void lcd\_data(void);

void wr\_cn(void);

void wr\_dn(void);

void scan(void);

void get\_key(void);

void display(void);

void delay(unsigned int);

void init\_port(void);

unsigned long int scan\_code[16]= {0x00EE0000,0x00ED0000,0x00EB0000,0x00E70000,

0x00DE0000,0x00DD0000,0x00DB0000,0x00D70000,

0x00BE0000,0x00BD0000,0x00BB0000,0x00B70000,

0x007E0000,0x007D0000,0x007B0000,0x00770000};

unsigned char ASCII\_CODE[16]= {'0','1','2','3',

'4','5','6','7',

'8','9','A','B',

'C','D','E','F'};

unsigned char row,col;

unsigned char temp,flag,i,result,temp1;

unsigned int r,r1;

unsigned long int var,var1,var2,res1,temp2,temp3,temp4;

unsigned char \*ptr,disp[] = "4X4 KEYPAD";

unsigned char disp0[] = "KEYPAD TESTING";

unsigned char disp1[] = "KEY = ";

int main()

{

// \_\_ARMLIB\_enableIRQ();

init\_port(); //port intialisation

delay(3200); //delay

lcd\_init(); //lcd intialisation

delay(3200); //delay

clr\_disp(); //clear display

delay(500); //delay

//........LCD DISPLAY TEST.........//

ptr = disp;

temp1 = 0x81; // Display starting address

lcd\_com();

delay(800);

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

//........KEYPAD Working.........//

while(1)

{

get\_key();

display();

}

} //end of main()

void get\_key(void) //get the key from the keyboard

{

unsigned int i;

flag = 0x00;

IO1PIN=0x000f0000;

while(1)

{

for(row=0X00;row<0X04;row++) //Writing one for col's

{

if( row == 0X00)

{

temp3=0x00700000;

}

else if(row == 0X01)

{

temp3=0x00B00000;

}

else if(row == 0X02)

{

temp3=0x00D00000;

}

else if(row == 0X03)

{

temp3=0x00E00000;

}

var1 = temp3;

IO1PIN = var1; // each time var1 value is put to port1

IO1CLR =~var1; // Once again Conforming (clearing all other bits)

scan();

delay(100); //delay

if(flag == 0xff)

break;

} // end of for

if(flag == 0xff)

break;

} // end of while

for(i=0;i<16;i++)

{

if(scan\_code[i] == res1) //equate the scan\_code with res1

{

result = ASCII\_CODE[i]; //same position value of ascii code

break; //is assigned to result

}

}

}// end of get\_key();

void scan(void)

{

unsigned long int t;

temp2 = IO1PIN; // status of port1

temp2 = temp2 & 0x000F0000; // Verifying column key

if(temp2 != 0x000F0000) // Check for Key Press or Not

{

delay(1000); //delay(100)//give debounce delay check again

temp2 = IO1PIN;

temp2 = temp2 & 0x000F0000; //changed condition is same

if(temp2 != 0x000F0000) // store the value in res1

{

flag = 0xff;

res1 = temp2;

t = (temp3 & 0x00F00000); //Verfying Row Write

res1 = res1 | t; //final scan value is stored in res1

}

else

{

flag = 0x00;

}

}

} // end of scan()

void display(void)

{

ptr = disp0;

temp1 = 0x80; // Display starting address of first line

lcd\_com();

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

ptr = disp1;

temp1 = 0xC0; // Display starting address of second line

lcd\_com();

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

temp1 = 0xC6; //display address for key value

lcd\_com();

temp1 = result;

lcd\_data();

}

void lcd\_init (void)

{

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x20;

wr\_cn();

delay(3200);

// load command for lcd function setting with lcd in 4 bit mode,

// 2 line and 5x7 matrix display

temp = 0x28;

lcd\_com();

delay(3200);

// load a command for display on, cursor on and blinking off

temp1 = 0x0C;

lcd\_com();

delay(800);

// command for cursor increment after data dump

temp1 = 0x06;

lcd\_com();

delay(800);

temp1 = 0x80;

lcd\_com();

delay(800);

}

void lcd\_data(void)

{

temp = temp1 & 0xf0;

wr\_dn();

temp= temp1 & 0x0f;

temp= temp << 4;

wr\_dn();

delay(100);

}

void wr\_dn(void) ////write data reg

{

IO0CLR = 0x000000FC; // clear the port lines.

IO0SET = temp; // Assign the value to the PORT lines

IO0SET = 0x00000004; // set bit RS = 1

IO0SET = 0x00000008; // E=1

delay(10);

IO0CLR = 0x00000008;

}

void lcd\_com(void)

{

temp = temp1 & 0xf0;

wr\_cn();

temp = temp1 & 0x0f;

temp = temp << 4;

wr\_cn();

delay(500);

}

void wr\_cn(void) //write command reg

{

IO0CLR = 0x000000FC; // clear the port lines.

IO0SET = temp; // Assign the value to the PORT lines

IO0CLR = 0x00000004; // clear bit RS = 0

IO0SET = 0x00000008; // E=1

delay(10);

IO0CLR = 0x00000008;

}

void clr\_disp(void)

{

// command to clear lcd display

temp1 = 0x01;

lcd\_com();

delay(500);

}

void delay(unsigned int r1)

{

for(r=0;r<r1;r++);

}

void init\_port()

{

IO0DIR = 0x000000FC; //configure o/p lines for lcd

IO1DIR = 0XFFF0FFFF;

}