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

```
// LCD INTERFACING

#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[] = "BMSIT&M",disp1[] = "LCD INTERFACING";

int main()
{
      PINSEL0 = 0X00000000;         // configure P0.0 TO P0.15 as GPIO
      IO0DIR = 0x000000FC;        //configure o/p lines for lcd [P0.2-P0.7]

      lcd_init();                 //lcd initialisation
      delay(3200);                      // delay  1.06ms

    clr_disp();                         //clear display
      delay(3200);                // delay  1.06ms


      temp1 = 0x81;               //Display starting address of first line 2nd pos
      lcd_com();                        //function to send command to lcd

      ptr = disp;                             // pointing data
    while(*ptr!='\0')
    {
            temp1 = *ptr;
      lcd_data();                   //function to send data to lcd
            ptr ++;
    }

      temp1 = 0xC0;          // Display starting address of second line 1st pos
      lcd_com();                        //function to send command to lcd

      ptr = disp1;                  // pointing second data
      while(*ptr!='\0')
    {
      temp1 = *ptr;
      lcd_data();                   //send data to lcd
            ptr ++;
    }
      while(1);

}  //end of main()

// lcd initialisation routine.
```

```
void lcd_init(void)
{
      temp = 0x30;                  //command to test LCD voltage level
      wr_cn();
      delay(3200);

      temp = 0x30;                   //command to test LCD voltage level
      wr_cn();
      delay(3200);

      temp = 0x30;                  //command to test LCD voltage level
      wr_cn();
      delay(3200);

      temp = 0x20;    // change to 4 bit mode from default 8 bit mode
      wr_cn();
      delay(3200);

      temp1 = 0x28; // load command for lcd function setting with lcd in 4 bit mode,
      lcd_com();    // 2 line and 5x7 matrix display
      delay(3200);

      temp1 = 0x0C; // load a command for display on, cursor on and blinking off

      lcd_com();
      delay(800);

      temp1 = 0x06;  // command for cursor increment after data dump
      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;            //masking higher nibble first
    wr_cn();
    temp = temp1 & 0x0f;              //masking lower nibble
    temp = temp << 4;
    wr_cn();
    delay(500);                             // some delay
}

// 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;      //E=0
}

// 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;  //E=0
}

// data o/p routine which also outputs high nibble first
// and lower nibble next

void lcd_data(void)
{
    temp = temp1 & 0xf0;            //masking higher nibble first
    temp = temp ;
    wr_dn();
    temp= temp1 & 0x0f;             //masking lower nibble
    temp= temp << 4;                //shift 4bit to left
    wr_dn();
    delay(100);
}

void clr_disp(void)                // function to clear the LCD screen
{
    temp1 = 0x01;
    lcd_com();
    delay(500);
}

void delay(unsigned int r1)       // delay function using for loop
{
     for(r=0;r<r1;r++);
}
```