

```

#include<REG51.h>
//#define Enable 0x08
//#define RS 0x04

sbit en = P3^3;
sbit rs = P3^2;

void DelayMs(delay)
{
    int i,j;
    for(i=0;i<delay;i++)
    {
        for(j=0;j<100;j++);
    }
}

void write_lcd_data(value)
{
    /* data for lcd */
    P0 = value;
    rs = 1;
    en = 1;
    DelayMs(10);
    en = 0;
}

void write_lcd_command(value)
{
    /* command for lcd */
    P0 = value;
    rs = 0;
    en = 1;
    DelayMs(10);
    en = 0;
}

void main(void)
{
    P0 = 0x00;
    P3 = 0x00;
    while(1)
    {

```

```

        write_lcd_command(0x38);
//function set
        DelayMs(100);
write_lcd_command(0x0E);           //display off
DelayMs(100);
write_lcd_command(0x01);           //display clear
DelayMs(100);
write_lcd_command(0x06);           //entry mode set
DelayMs(100);
write_lcd_command(0x0F);           //display on
DelayMs(100);
write_lcd_command(0x80);           //set address
counter value
DelayMs(100);

write_lcd_data('M');
DelayMs(100);
write_lcd_data('I');
DelayMs(100);
write_lcd_data('C');
DelayMs(100);
write_lcd_data('R');
DelayMs(100);
write_lcd_data('O');
DelayMs(100);
write_lcd_data('C');
DelayMs(100);
write_lcd_data('O');
DelayMs(100);
write_lcd_data('N');
DelayMs(100);
write_lcd_data('T');
DelayMs(100);
write_lcd_data('R');
DelayMs(100);
write_lcd_data('O');
DelayMs(100);
write_lcd_data('L');
DelayMs(100);

```

```
write_lcd_data('L');  
DelayMs(100);  
write_lcd_data('E');  
DelayMs(100);  
write_lcd_data('R');  
DelayMs(100);  
    }  
}
```