```
/*--------------------------------------------------------------------------------+
|                                                                                 |
|                          LCD 16x2 Driver (HD44780)                              |
|                                                                                 |
| Name: Driver_Lcd16x2.c                                                          |
| Purpose:                                                                        |
|   This is the function code used to modify the standard c function "printf" by  |
|   replacing the function "putchar" to suit the LCD driver HD44780. By adding this |
|   file, LCD panel with a chip HD44780 can print out a string using the standard c |
|   function "printf". \n, \r, and roll down will be included.                    |
|                                                                                 |
| By Dillian Wong, last modify data 19/9/03                                       |
+--------------------------------------------------------------------------------*/

/*_____________ I N C L U D E S _____________________________________________*/

#include <reg51.h>
#include <stdio.h>

/*_____________ M A C R O S _____________________________________________*/

#define DISP_BUSY   0x80
#define DISP_FUNC   0x38
#define DISP_ENTRY  0x06
#define DISP_CNTL   0x08
#define DISP_ON     0x04
#define DISP_CURSOR 0x02
#define DISP_CLEAR  0x01
#define DISP_HOME   0x02
#define DISP_POS    0x80
#define DISP_LINE2  0x40

/*_____________ H A R D W A R E _____________________________________________*/

sbit ENABLE   =  P3^2;                    // display enable output (EN)
sbit RDWR     =  P3^1;                    // display access mode output (RW)
sbit REGSEL   =  P3^0;                    // display register select output (DI)
sfr  DISPDATA =  0xA0;                    // data bus to the display panel,P2

/*_____________ P R O T O C O L _____________________________________________*/

void disp_cmd(unsigned char);
void disp_init(void);
unsigned char disp_read(void);
void disp_write(unsigned char);
```

```
unsigned char disp_read_char(void);
void disp_copyLine2toLine1(void);
void disp_clrLine2(void);


/*_____________ F U N C T I O N _____________________________________________*/

/*
    writes a given command to the LCD panel and waits to assure that the command
    was completed by the panel.
*/
void disp_cmd(unsigned char cmd) {
    DISPDATA=cmd;
    REGSEL=0;
    RDWR=0;
    ENABLE=1;
    ENABLE=0;
    while (disp_read() & DISP_BUSY);                // wait for the Lcd
}

/*
    sends the correct data sequence to the display to initialize it for use.
*/
void disp_init(void) {

    while (disp_read() & DISP_BUSY);
    disp_cmd(DISP_FUNC);                            // set the display for an 8
                                                    // bit bus, 2 display lines,
                                                    // and a 5x7 dot font
    disp_cmd(DISP_ENTRY);                           // set the character entry
                                                    // mode to increment display
                                                    // address for each
                                                    // character, but not to scroll
    disp_cmd(DISP_CNTL | DISP_ON);                  // turn the display on, cursor of
f
    disp_cmd(DISP_CLEAR);                           // clear the display
}

/*
    reads from the LCD panel status register.
*/
unsigned char disp_read(void) {
    unsigned char value;
    DISPDATA=0xFF;
    REGSEL=0;
    RDWR=1;
    ENABLE=1;
    value=DISPDATA;
    ENABLE=0;
    return(value);
}

/*
    writes a data byte to the LCD panel.
*/
void disp_write(unsigned char value) {
    DISPDATA=value;
    REGSEL=1;
    RDWR=0;
```

```
    ENABLE=1;
    ENABLE=0;
}

/*
    read data from CF or DD Ram
*/
unsigned char disp_read_char(void) {
    unsigned char value;
    DISPDATA=0xFF;
    REGSEL=1;
    RDWR=1;
    ENABLE=1;
    value=DISPDATA;
    ENABLE=0;
    return(value);
}

/*
    reads data from DD Ram in line2, and write char of these datas in line1
*/
void disp_copyLine2toLine1(void){
    unsigned int i;
    unsigned char k[16];
    disp_cmd(DISP_POS | DISP_LINE2);
    for(i=0;i<16;i++){
        k[i]=disp_read_char();
        while (disp_read() & DISP_BUSY);
    }
    disp_cmd(DISP_HOME);
    for(i=0;i<16;i++){
        disp_write(k[i]);
        while (disp_read() & DISP_BUSY);
    }
}

/*
    clear the display of Line2
*/
void disp_clrLine2(void){
    unsigned int i;
    disp_cmd(DISP_POS | DISP_LINE2);
    for(i=0;i<16;i++){
        disp_write(' ');
        while (disp_read() & DISP_BUSY);
    }
}

/*************************************************************
Function:    putchar
Description: This routine replaces the standard putchar
             function. Its job is to redirect output to the LCD panel.
Parameters:  c - char. This is h\next character to write to the
             display.
Returns:     The character just written.
*************************************************************/
char putchar(char c) {
    /*
    HD44760 16x2 character LCD display
```

```
    the following .no is variable flag refer to
    +-----------------------------------------------+
    |  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 |
    | 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 |
    +-----------------------------------------------+
    */
    static unsigned char flag=0;

    if (flag <=15){
        // standard format for the nextline
        if(c=='\n'){
                disp_clrLine2();
                disp_cmd(DISP_POS | DISP_LINE2);
                flag=16;
                return(c);
        }
        // standard format for return the line
        if(c=='\r'){
                disp_cmd(DISP_HOME);
                flag=1;
                return(c);
        }
    }

    if(flag==16){
            disp_cmd(DISP_POS | DISP_LINE2);
    }

    if(flag>=16){
        // standard format for the nextline
        if(c=='\n'){
                disp_copyLine2toLine1();
                disp_clrLine2();
                disp_cmd(DISP_POS | DISP_LINE2);
                flag=16;
                return(c);
        }
        // standard format for return the line
        if(c=='\r'){
                disp_cmd(DISP_POS | DISP_LINE2);
                flag=16;
                return(c);
        }
    }

    if (flag>31){
        disp_copyLine2toLine1();
        disp_clrLine2();
        disp_cmd(DISP_POS | DISP_LINE2);
        flag=16;
    }


    disp_write(c);                          // write the character to
                                            // the display
    while (disp_read() & DISP_BUSY);        // wait for the display
    flag++;
    return(c);
```

```
}
```

```
/*-------------------------------------------------------------------------------------+
| Name: DemoMain.c                                                                     |
| Purpose:                                                                             |
| Demonstrate the usage of 16x2 character LCD that drived by a chip HD44780. The C     |
| files code will modify the standard C function "printf" by replacing function       |
| "putchar". Hence, originally function "printf" will print a character out in RS232   |
| port, in this program, however, printf will modified to print a character out in     |
| standard 16x2 character LCD that drived by a chip HD44780.                           |
+-------------------------------------------------------------------------------------*/
#include <reg51.h>
#include <stdio.h>

void disp_init(void);

/*
    Demonstrate the usage of 16x2 character LCD that drived by a chip HD44780.
*/
void main(void) {


    disp_init();
    printf("Hello\n");
    printf("It is Good");

    while(1);

 }
```

UMPS

After KEIL C complier the file, we use the execute file Hin2bin to convert the hex file to binary file
Here is the command code >> Hin2bin Driver.hex