

实验二 段式 LCD 模块原理以及接口技术应用报告

1. 实验要求

- 1) 先读懂实验例程的功能；
- 2) 结合硬件接口，理解显示函数底层显示功能；
- 3) 在段式 LCD 上显示小组成员的年月日 24.0305 和 SJTU 等信息。

2. 设计思路

当进行 MSP430 微控制器的 LCD 显示实验时，首先需要准备好系统时钟，并选择合适的晶振作为系统时钟源。然后，需要配置与段式液晶相连的 IO 口，确定它们的工作模式。此外，还需要清空 LCD 寄存器，并启动 LCD 模块。

完成这些配置可通过 MSP430 内部 LCD 驱动器相关的寄存器。其中，P5SEL 寄存器用于设置端口功能选择，LCDBPCTL0 寄存器用于配置 COM 口的驱动方式，LCDBCTL0 寄存器用于控制 LCD 的基本设置，LCDBMEMCTL 寄存器用于控制 LCD 内存的保护方式，LCDBCTL0 寄存器用于启动 LCD 控制器。

在 dr_lcdseg.c 文件中，通过增加 SEG_CTRL_BIN 数组进行"S""J""T""U"的显示，而日期和小数点的显示可通过 LCDSEG_DisplayNumber 进行。

在主程序中，完成逻辑：

1. 初始化常量、LCD 寄存器和时钟，关闭看门狗时钟。
2. 显示 24.0305，延时后擦除，显示 SJTU，延时后擦除并循环此步骤。

3. 实验代码

```
3 /*
4 #include <msp430f6638.h>
5 #include <stdint.h>
6 #include <stdio.h>
7 #include <string.h>
8 #include "dr_lcdseg.h" //调用段式液晶驱动头文件
9 #define XT2_FREQ 4000000
10 #define MCLK_FREQ 16000000
11 #define SMCLK_FREQ 4000000
12 void initClock()
13 {
14     while(BAKCTL & LOCKIO) //解锁XT1引脚操作
15     BAKCTL &= ~(LOCKIO);
16     UCSCTL6 &= ~XT1OFF; //启动XT1, 选择内部时钟源
17     P7SEL |= BIT2 + BIT3; //XT2引脚功能选择
18     UCSCTL6 &= ~XT2OFF; //启动XT2
19     while (SFRIFG1 & OFIFG) //等待XT1、XT2与DCO稳定
20     {
21         UCSCTL7 &= ~(DCOFFG+XT1LFOFFG+XT2OFFG);
22         SFRIFG1 &= ~OFIFG;
23     }
24     UCSCTL4 = SELA__XT1CLK + SELS__XT2CLK + SELM__XT2CLK; //避免DCO调整中跑飞
25     UCSCTL1 = DCORSEL_5; //6000kHz~23.7MHz
26     UCSCTL2 = MCLK_FREQ / (XT2_FREQ / 16); //XT2频率较高, 分频后作为基准可获得更高的精度
27     UCSCTL3 = SELREF__XT2CLK + FLLREFDIV_16; //XT2进行16分频后作为基准
28     while (SFRIFG1 & OFIFG) //等待XT1、XT2与DCO稳定
29     {
30         UCSCTL7 &= ~(DCOFFG+XT1LFOFFG+XT2OFFG);
31         SFRIFG1 &= ~OFIFG;
32     }
33     UCSCTL5 = DIVA__1 + DIVS__1 + DIVM__1; //设定几个CLK的分频
34     UCSCTL4 = SELA__XT1CLK + SELS__XT2CLK + SELM__DCOCLK; //设定几个CLK的时钟源
35 }
36
37 void main(void)
38 {
39     unsigned char i,num1;
```

```
37 void main(void)
38 {
39     unsigned char i,num1;
40     int32_t num2;
41     WDTCTL = WDTPW | WDTHOLD; // 停止看门狗
42     initClock(); // 配置系统时钟
43     initLcdSeg(); // 初始化段式液晶
44     while(1) // 进入程序主循环
45     {
46         LCDSEG_DisplayNumber(240305,3); //显示日期, 这里小数点选择点在第3位
47         __delay_cycles(MCLK_FREQ*5); //延时
48         LCDSEG_DisplayNumber(000000,0);
49         LCDSEG_SetDigit(0,-1);
50         LCDSEG_SetDigit(5,-1);
51         LCDSEG_SetDigit(4,-1);
52         LCDSEG_SetDigit(3,-1);
53         LCDSEG_SetDigit(2,-1);
54         LCDSEG_SetDigit(1,-1);
55         //显示SJTU
56         LCDSEG_SetDigit(4,13);
57         LCDSEG_SetDigit(3,14);
58         LCDSEG_SetDigit(2,15);
59         LCDSEG_SetDigit(1,16);
60         __delay_cycles(MCLK_FREQ*5); //延时
61         LCDSEG_SetDigit(4,-1);
62         LCDSEG_SetDigit(3,-1);
63         LCDSEG_SetDigit(2,-1);
64         LCDSEG_SetDigit(1,-1);
65     }
66 }
67 }
68
69
```

以下为 dr_lcdseg.c 文件的主要更改部分。

```

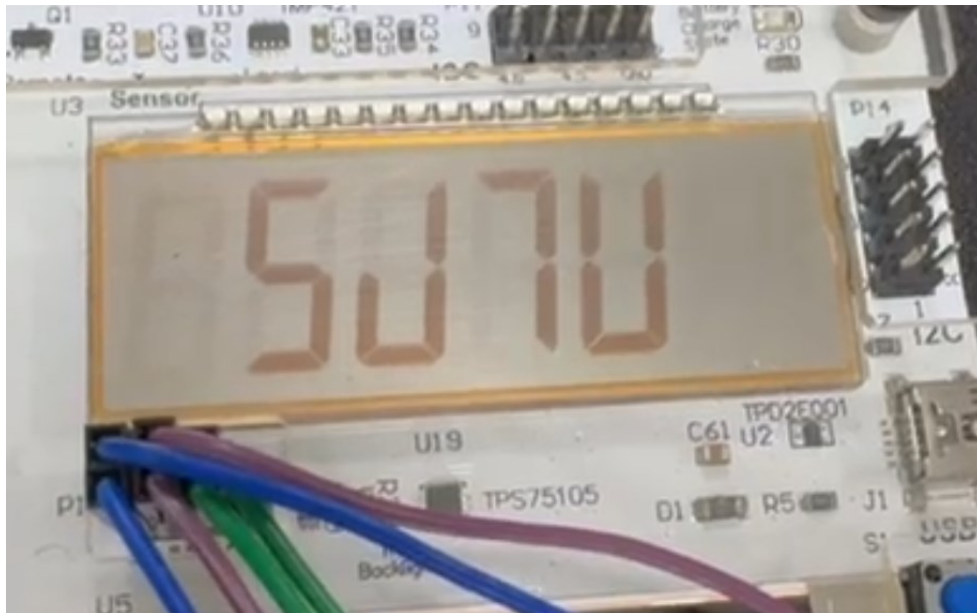
25 0x7C, //display b
26 0x40, //display -
27 0x6D, //display S
28 0x1E, //display J
29 0x07, //display T
30 0x3E, //display U
31};
32
33 void initLcdSeg()
34{
35 //端口设定
36 P5SEL |= BIT3 + BIT4 + BIT5; //P5.3 .4 .5作为LCD的COM
37 LCDBPCTL0 = 0x0FFF; //S0~S11所在端口作为LCD的段选
38 //控制器设定
39 LCDBCTL0 = LCDDIV_21 + LCDPRE_4 + LCD4MUX; //ACLK, 21*4分频, 合48.76Hz
40 LCDBMEMCTL |= LCDCLRM; //清空LCD存储器
41 LCDBCTL0 |= LCDSON + LCDON; //启动LCD模块
42}
43
44 void LCDSEG_SetDigit(int pos, int value) //value不在0~16时熄灭,16显示-
45{
46 if(pos < 0 || pos > 6)
47 return;
48 pos = 5 - pos;
49
50 uint8_t temp, mem;
51 if(value < 0 || value > 20)
52 temp = 0x00;
53 else
54 temp = SEG_CTRL_BIN[value];
55
56 const static uint8_t map[7] ={ BIT7, BIT6, BIT5, BIT0, BIT1, BIT3, BIT2 };
57
58 mem = LCDMEM[pos];
59 mem &= 0x10; //清空控制数字段的位
60 int i;

```

4. 实验结果

以下为 lcd 显示实验结果的照片记录，具体结果可见附件视频。





5. 分析与思考

通过对比 msp430 与 c51 中 lcd 屏显示的控制方式，可发现通过函数封装常用操作和以数组存储类似逻辑的数值映射，能够极大简化程序并提升可读性，降低许多设备的使用门槛。同时，由于不同型号的硬件常存在差异，统一驱动函数可简化不同型号硬件的控制和使用。