



嵌入式系统设计实验报告

实验:	实验 4 触摸屏和显示字实验
专业:	计算机科学与技术
班级:	1 班
姓名:	姚怀聿
学号:	22920202204632

2022 年 12 月 19 日

目 录

一、	实验目的	3
二、	实验方案设计	3
1.	基础实验	3
2.	进阶实验	3
三、	实验过程	4
1.	基础实验	4
2.	进阶实验	6
四、	实验结果	8
五、	实验分析	10
六、	心得体会	10

一、实验目的

- 1、 掌握串口通讯的工作原理与配置方法，掌握定时器的原理。
- 2、 学会串口调试工具的使用方法，学会触摸屏的使用与软件配置方法。

二、实验方案设计

1. 基础实验

实验要求:

在触摸屏上定义好 5 个 LED 开关，使用串口功能发送指令，利用显示屏的按键开关控制电路板上的亮灭。

2. 进阶实验

实验要求:

我选择在实验箱的矩阵 LED 上写一个“田”字。将该字按照笔画顺序，按照逐个灯亮起的模式写该字，同笔画的 LED 灯亮起间隔为 3s，上一笔画的最后 LED 灯和下一笔画的 LED 灯亮起的间隔为 6s。

三、实验过程

首先，对于所有的实验，我们都需要进行一些初始化的设置，这个函数在 main.c 中调用：

```
20 int main(void)
21 {
22     system_init();
23
24     while(1){
25         test1();
26         //test2();
27     };
28 }
```

```
36
37 //系统初始化 (systick、定时器、GD_XLL、HC595、DIWIN )
38 void system_init(void) {
39     systick_config();
40     TIM3_Init(29999, 9999);
41     gd_XII_systeminit();
42     HC595_GPIO_Config();
43     diwen_init();
44     usart0Init(EVAL_COM0, 115200U);
45 }
46
```

TIM3_Init(29999, 9999)实现 3 秒定时

```
9 void TIM3_Init(uint16_t arr, uint16_t psc) {
10     timer_parameter_struct timer_initpara;
11
12     rcu_periph_clock_enable(RCU_TIMER3);
13     timer_deinit(TIM3); //复位定时器
14
15     nvic_irq_enable(TIM3_IRQn, 3, 3); //配置NVIC分组
16
17     //定时器TIM3初始化
18     timer_initpara.prescaler = psc; //设置用来作为TIMx时钟频率除数的预分频值
19     timer_initpara.alignedmode = TIMER_COUNTER_EDGE; //对齐
20     timer_initpara.counterdirection = TIMER_COUNTER_UP; //TIM向上计数模式
21     timer_initpara.period = arr; //设置在下一个更新事件装入活动的自动重装载寄存器周期的值
22     timer_initpara.clockdivision = TIMER_CKDIV_DIV1; //设置时钟分割:TDTS = Tck_tim
23     timer_init(TIM3, &timer_initpara);
24
25     timer_interrupt_enable(TIM3, TIMER_INT_UP); //使能指定的TIM3中断,允许更新中断
26
27     timer_enable(TIM3); //使能TIM3
28 }
```

1. 基础实验

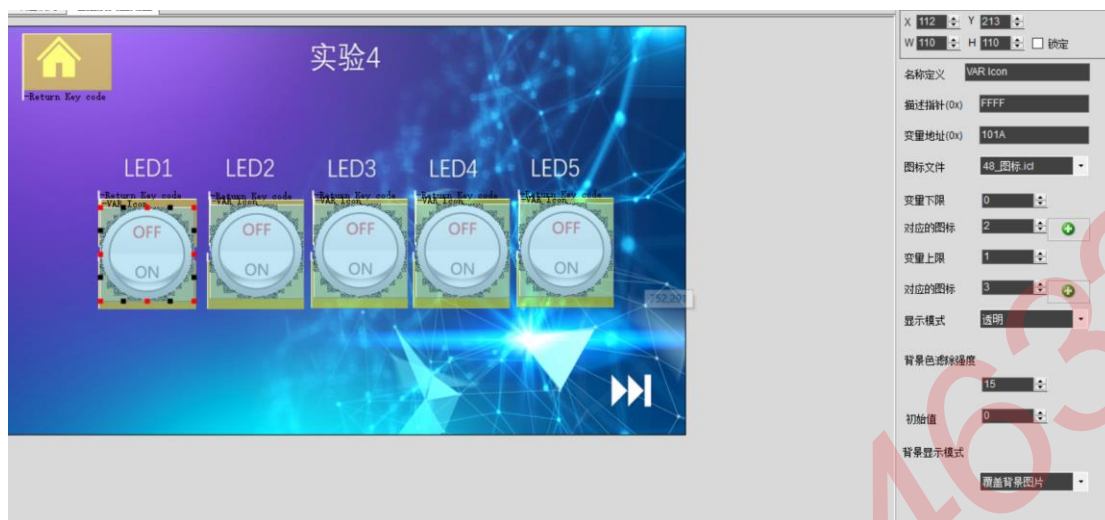
在 test.c 文件中编写 test2()函数:

只是简单的判断语句:

```
94  
95 //实验二 串口触摸屏控制LED  
96  
97 void test2(void) {  
98     while(uart5_rx_buffer[5] == 0x1A) {  
99         delay_lms(100);  
100         while(uart5_rx_buffer[8] == 0x01) {  
101             gd_eval_led_on(LED1);  
102         }  
103         while(uart5_rx_buffer[8] == 0x00) {  
104             gd_eval_led_off(LED1);  
105         }  
106     }  
107     while(uart5_rx_buffer[5] == 0x11) {  
108         delay_lms(100);  
109         while(uart5_rx_buffer[8] == 0x01) {  
110             gd_eval_led_on(LED2);  
111         }  
112         while(uart5_rx_buffer[8] == 0x00) {  
113             gd_eval_led_off(LED2);  
114         }  
115     }  
116 }
```



将每一个按键的变量地址和变量值都设置好（需要和程序中的对应）。这里还需要注意一点，需要将数据自动上传选项勾上，否则无法获得触摸得到的数据:



然后将这个文件夹放到实验箱配套的 sd 卡中，然后插入 sd 卡，就可以将触摸屏页面更换。编译、运行程序即可实现触摸屏上按键控制 LED 灯的功能：

42_图标库	2022/11/28 9:13	文件夹	
48_图标	2022/11/28 9:13	文件夹	
DWIN_SET	2022/11/28 9:13	文件夹	
ICON	2021/11/4 8:34	文件夹	
image	2021/11/4 8:34	文件夹	
TFT	2022/11/28 9:13	文件夹	
DisplayConfig.xls	2022/12/12 19:45	Microsoft Excel ...	5 KB
DWprj.hmi	2022/12/19 17:17	HMI 文件	1 KB
DWprj.tft	2022/12/19 17:17	TFT 文件	3 KB
TouchConfig.xls	2022/12/12 19:45	Microsoft Excel ...	6 KB

2. 进阶实验

先将“田”字的每一笔画定义好：

```

12  /*
13  00000000
14  11111110
15  10010010
16  11111110
17  10010010
18  11111110
19  00000000
20  00000000 // 田
21  */
22  uint8_t row_clear[] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}; //8*8LED清屏
23  uint8_t col_clear[] = {0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff};
24  uint8_t row1[] = {0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
25  uint8_t core1[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f}; //田字第一笔画*/
26  uint8_t row2[] = {0x3e, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x3e, 0x00};
27  uint8_t core2[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f}; //田字第二笔画*/
28  uint8_t row3[] = {0x3e, 0x0a, 0x0a, 0x0a, 0x0a, 0x0a, 0x0a, 0x3e, 0x00};
29  uint8_t core3[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f}; //田字第三笔画*/
30  uint8_t row4[] = {0x3e, 0x0a, 0x0a, 0x3e, 0x0a, 0x0a, 0x3e, 0x00};
31  uint8_t core4[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f}; //田字第四笔画*/
32  uint8_t row5[] = {0x3e, 0x2a, 0x2a, 0x3e, 0x2a, 0x2a, 0x3e, 0x00};
33  uint8_t core5[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f}; //田字第五笔画*/
34

```

timer.c 文件中定义一个变量 `tim3_count`，用于记录当前是第几个 3 秒，每有一个 3 秒到达，该变量的值就加 1:

```

6  volatile uint16_t tim3_count;
7  u8 led_check=0;

54 void TIMER3_IRQHandler(void)
55 {
56     if (SET==timer_interrupt_flag_get(TIMER3,TIMER_INT_UP)) //检查指定的TIM中断发生与否:TIM 中断源
57     {
58         tim3_count++;
59         timer_interrupt_flag_clear(TIMER3, TIMER_INT_UP); 注意这里还需要将标志清空
60     }
61 }

```

然后再 test.c 文件中继续完成 `test1()` 函数:

因为田字共有 5 笔，且实验要求当前字最后一笔和下一个字的第一笔中间要间隔 6 秒，也就是两个三秒，所以当来到第五个 3 秒时，将 LED 屏清空，然后再等待 3 秒写下一个字的第一笔，这样就实现了中间间隔 6 秒的效果:

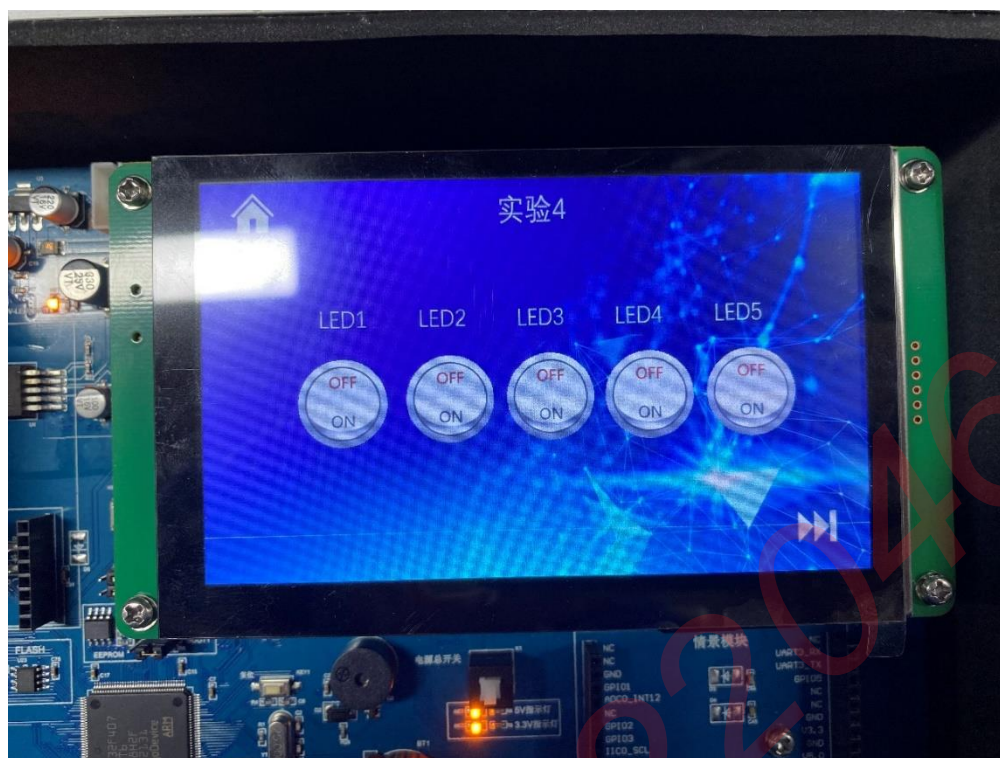

```

47 //实验一 LED写汉字
48 void test1() {
49     uint8_t i;
50     if(tim3_count == 0) {
51         for(i = 0; i < 8; i++) {
52             HC595_Send_Byte(row1[i]);
53             HC595_Send_Byte(core1[i]);
54             HC595_CS();
55         }
56     }
57     if(tim3_count == 1) {
58         for(i = 0; i < 8; i++) {
59             HC595_Send_Byte(row2[i]);
60             HC595_Send_Byte(core2[i]);
61             HC595_CS();
62         }
63     }
64     if(tim3_count == 2) {
65         for(i = 0; i < 8; i++) {
66             HC595_Send_Byte(row3[i]);
67             HC595_Send_Byte(core3[i]);
68             HC595_CS();
69         }
70     }
71     if(tim3_count == 3) {
72         for(i = 0; i < 8; i++) {
73             HC595_Send_Byte(row4[i]);
74             HC595_Send_Byte(core4[i]);
75             HC595_CS();
76         }
77     }
78     if(tim3_count == 4) {
79         for(i = 0; i < 8; i++) {
80             HC595_Send_Byte(row5[i]);
81             HC595_Send_Byte(core5[i]);
82             HC595_CS();
83         }
84     }
85     if(tim3_count == 5) {
86         for(i = 0; i < 8; i++) {
87             HC595_Send_Byte(row_clear[i]);
88             HC595_Send_Byte(cal_clear[i]);
89             HC595_CS();
90         }
91     }
92     if(tim3_count == 6) tim3_count = 0;
93 }
94

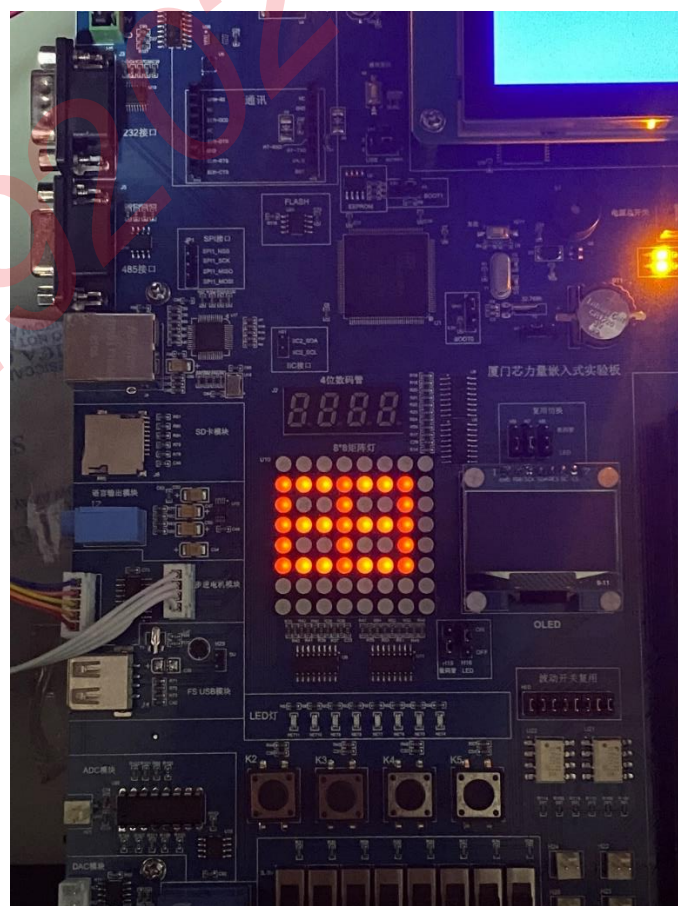
```

四、实验结果

1. 基础实验



2. 进阶实验



五、实验分析

本次实验都是与屏幕有关的实验，该实验中我接触到了定时器这个东西，学会了使用定时器引发中断从而实现一些功能。当一个定时到来时，还要记得将标志位清空，否则会看不到效果。整个实验完成较为曲折，第一周因为老师的读卡器没带，将触摸屏的实验推迟到了下一周，然后刚好赶上疫情，线上上课，又冲掉了一次实验课。上周连做两个实验，还好都做完了。实验整体不难，过程太过于曲折了-_-||

六、心得体会

本次实验主要是使用 LED 以及触摸屏。LED 的实验比较简单，助教老师在课上都已经讲的很清楚了；但是一开始我还是显示不出来，后来才发现是硬件上没有将跳线帽正确接好。将硬件安好之后，还是不行，检查代码发现是在写中断函数的时候，忘记将标志清空了少了一个函数，就无法达到实验要求，可见，每一步都要头脑清晰。在实现触摸屏实验的时候，一直无法实现，后来在助教老师的提醒下，将数据自动上传勾选，顺利实现了实验。