

工程实践与科技创新 II-A

(信工电科教学班)

实验报告

实验 4 点阵字符显示和用户操作界面设计

实验 5 发送-接收联合系统实验

组号： 25B

成员姓名： 戴天杰(组长)、张烨忱

撰写统稿： 戴天杰

完成时间： 2021 年 06 月 22 日

目 录

1. 实验 4 点阵字符显示和用户操作界面设计	2
1.1 实验目的	2
1.2 实验主要器材和设备	2
1.3 实验任务技术解决方案的简要说明	2
1.3.1 实验任务 4_1	2
1.3.2 实验任务 4_2	3
1.3.3 实验任务 4_3	4
1.4 实验核心代码清单	6
1.4.1 实验任务 4_1	6
1.4.2 实验任务 4_2	19
1.4.3 实验任务 4_3	26
2. 实验 5 发送-接收联合系统实验	36
2.1 实验目的	36
2.2 实验主要器材和设备	36
2.2.1 嵌入式系统基本开发环境	36
2.2.2 仪器工具	36
2.2.3 主要元件类器材	36
2.3 实验过程	37
2.3.1 实验任务 5_1——编程控制 FM 模块	37
2.3.2 实验任务 5_2——电压信号发生与测量显示	37
2.3.3 实验任务 5_5——信号调理和频率测量	37
2.3.4 实验任务 5_7——接收端系统整合	37
2.3.5 实验任务 5_8——联合实验	38
2.4 实验信号波形记录	38
2.5 实验现场考核记录	39
2.6 实验核心代码清单	39
2.6.1 实验任务 5_1	39
2.6.2 实验任务 5_7 & 5_8	44
3. 学习心得和意见建议	58
3.1 实验四学习心得	58
3.2 实验五学习心得	58
3.2.1 电路板的焊接	58
3.2.2 组员间的协作	59
3.3 《工程实践与科技创新 II-A》拓展实验部分的意见建议	60
4. 课程代码开源链接	61
5. 致谢	62

1. 实验 4 点阵字符显示和用户操作界面设计

1.1 实验目的

- 学习如何查阅厂商技术资料，掌握点阵液晶显示屏电路模块的应用开发方法；
- 学会基于有限状态机逻辑设计用户操作界面功能的一般方法；
- 学习借助专业测量工具仪器评估实验作品技术性能的实验技巧；
- 培养在技术团队中以专业态度扮演个人角色和履行分工职责的能力。

1.2 实验主要器材和设备

- 电脑；TM4C1294NCPDT 实验板卡；A2000TM4 扩展板；
- 调频发射实验单元板或调频接收实验单元板(两种单元板均带有 JLX12864G-086-PC 型液晶显示屏电路模块)；
- 台式稳压电源或 USB 接口供电设备(输出电压 DC5V，输出电流 $>0.5A$)；数字示波器；多用电表。

1.3 实验任务技术解决方案的简要说明

1.3.1 实验任务 4_1

实验任务 4_1 是最基础的部分。我们首先根据 canvas 上李安琪老师的操作，把液晶屏模块厂商资料《JLX12864G-086-PC 带字库 IC 的编程说明书》中的例程按照 TM4C1294NCPDT 编程特性与代码识别要求进行针对性改写。

实验任务 4_1 整体来说较为简单，在改写完编程说明书例程后，将代码整段移植到基础实验 2 的代码中，再利用 switch...case...语句，便可实现液晶屏轮换显示，每页保持 2-3 秒后自动切换，并利用 display_GB2312_string 函数增加了反白功能，具体而言，将其中某页的 1、3 行改为反白显示，2、4 行维持原状。最后，在展示自行设计的图片环节，我们利用相关网站，将上海交通大学校徽图片转为 16 进制的多维数组，将数组拷贝到 JLX12864.c 文件中，即可实现相关视觉效果。

实验 4_1 的流程图如图 1 所示。

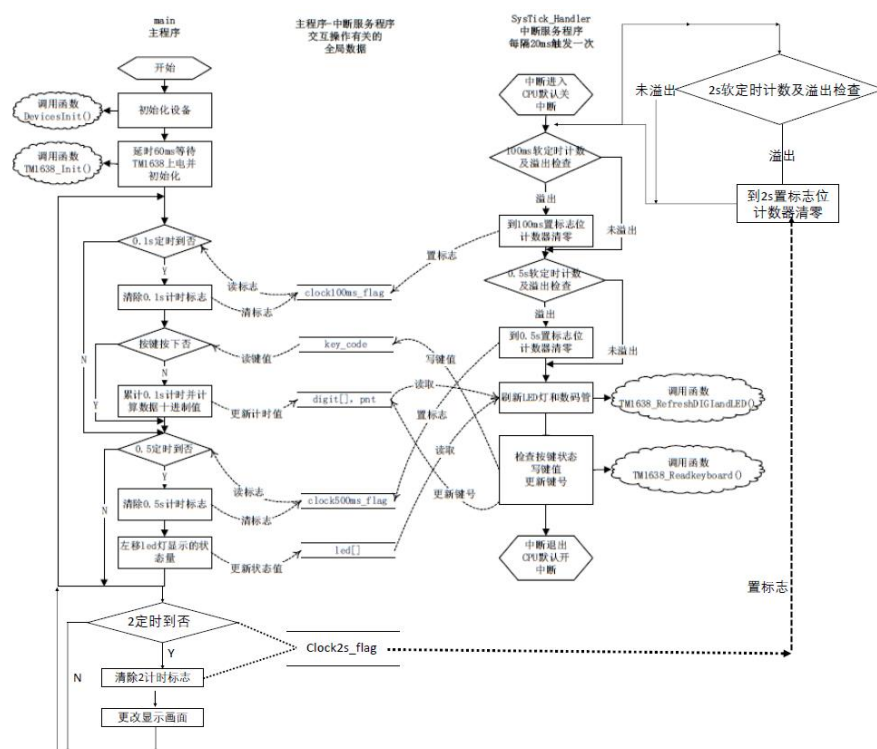


图 1 实验任务 4_1 FSM 的流程化展示

1.3.2 实验任务 4_2

显然，在刚才的任务 4_1 中是不带按键功能的，也就是没有所谓的人机交互操作。任务 4_2 则要求我们设计一个带交互的实例，一方面让我们初探 UI 界面的程序实现，另一方面则增进后续 UI 界面代码实现的熟练度。

图 2 给出了 FSM 状态转移切换示意图。显然，一共有 4 个主状态：ACT0、ACT001、ACT003、ACT005，在编程实战中，我们将该四个主状态单独写在了 mode_2.c 文件里，利用 display_GB2312_string 函数实现显示屏上不同主状态、不同区域块的内容、反白等视觉效果。

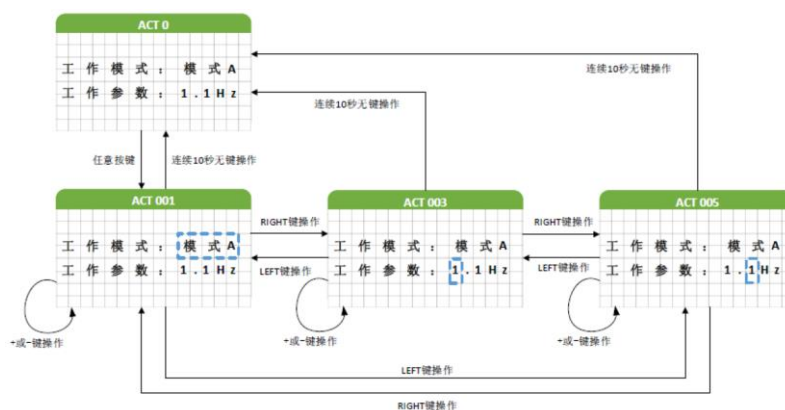


图 2 实验任务 4_2 FSM 的形象化表达

至于同一主模式下内部利用按键控制相关参数增减变换,我们将实现代码写在了主程序 exp4_2.c 里,利用基础实验部分利用中断读取按键相关代码实现按键读取功能,利用 switch...case, if...else... 语句实现不同子状态的选择。根据报告指示要求,代码段 1 为 ACT005 的相关代码。其余部分具体代码详见 1.4.2 节。

```
if (lcd_act == 0 && key_code != 0) {
    ++lcd_act;
    if_refresh = true;
    return;
}
if (key_code_pre != 4 && key_code == 4) {
    lcd_act = (--lcd_act + 2) % 3 + 1;
    if_refresh = true;
    return;
}
if (key_code_pre != 6 && key_code == 6) {
    lcd_act = (--lcd_act + 1) % 3 + 1;
    if_refresh = true;
    return;
}

if (key_code_pre != 8 && key_code == 8){
    switch (lcd_act) {
        case 1: let = (let + 2) % 3; break;
        case 2: integer = (integer + 9) % 10; break;
        case 3: decimal = (decimal + 9) % 10; break;
    }
    if_refresh = true;
}
if (key_code_pre != 2 && key_code == 2){
    switch (lcd_act) {
        case 1: let = ++let % 3; break;
        case 2: integer = ++integer % 10; break;
        case 3: decimal = ++ decimal % 10; break;
    }
    if_refresh = true;
}
}
```

代码段 1 实验任务 4_2 ACT005 程序化实现

1.3.3 实验任务 4_3

实验任务 4_3 是实验四最后一个部分,也是实验五发射机、接收机屏幕显示效果的任务来源。在该部分,任务要求我们在实验任务 4_2 的基础是编程实现一个多画面、多层次菜单的用户操作界面实例,实验任务 4_3 的状态机分析模型文字列表描述如表 1 所示。

表 1 实验任务 4_3 FSM 的文字化描述

NO.	状态代号	画面现象	触发条件	转移前操作	下一状态
1	ACT0	开机初始画面,不显示光标	任意按键	“设置”做反白效果(光标)	ACT005
2	ACT005	光标在“设置”的位置	“确定”键操作	移到 ACT1 的“工作参数”	ACT100
			5 秒无键操作	“设置”反白解除	ACT0
3	ACT100	光标在“工作模式”的位置	“←”键操作	移到“返回”	ACT102
			“→”键操作	移到“工作参数”	ACT101

NO.	状态代号	画面现象	触发条件	转移前操作	下一状态
4	ACT101	光标在“工作参数”的位置	“确定”键操作	移到 ACT2 的“模式#”	ACT201
			“←”键操作	移到“工作模式”	ACT100
			“→”键操作	光标移到“返回”	ACT102
			“确定”键操作	移到 ACT3 的工作参数的个位	ACT301
5	ACT102	光标在“返回”的位置	“←”键操作	移到“工作参数”	ACT101
			“→”键操作	移到“工作模式”	ACT100
			“确定”键操作	显示初始画面	ACT0
6	ACT201	光标在“模式#”的位置	“←”键操作	移到“取消”	ACT203
			“→”键操作	移到“确定”	ACT202
			“+”键操作	“模式#”中字母正序循环切换	ACT201
			“-”键操作	“模式#”中字母逆序循环切换	ACT201
7	ACT202	光标在“确定”的位置	“←”键操作	移到“模式#”	ACT201
			“→”键操作	移到“取消”	ACT203
			“确定”键操作	保存更改；移到 ACT1“工作模式”	ACT100
8	ACT203	光标在“取消”的位置	“←”键操作	移到“确定”	ACT202
			“→”键操作	移到“模式#”	ACT201
			“确定”键操作	撤销更改；移到 ACT1“工作模式”	ACT100
9	ACT301	光标在工作参数个位处	“←”键操作	移到“取消”	ACT306
			“→”键操作	移到工作参数的十分位	ACT303
			“+”键操作	个位正序切换	ACT301
			“-”键操作	个位逆序切换	ACT301
10	ACT303	光标在工作参数十分位处	“←”键操作	移到工作参数的个位	ACT301
			“→”键操作	移到“确定”	ACT305
			“+”键操作	十分位正序切换	ACT303
			“-”键操作	十分位逆序切换	ACT303
11	ACT305	光标在“确定”的位置	“←”键操作	移到工作参数的十分位	ACT303
			“→”键操作	移到“取消”	ACT306
			“确定”键操作，且参数值合法	保存更改；移到 ACT1“工作参数”	ACT101
			“确定”键操作，且参数值非法	显示警示画面 ACT4	ACT4
12	ACT306	光标在“取消”的位置	“←”键操作	移到“确定”	ACT305
			“→”键操作	移到工作参数的个位	ACT301
			“确定”键操作	移到 ACT1 的“工作参数”	ACT101

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```
//*****  
//  
// Copyright: 2020-2021, M****A****B****E****L****  
// File name: exp2_0.c  
// Description:  
  
//      1. ****λϣ****γ4λ****Q****'****^****λÖ****λθ0.1****  
//      2. ****λϣ8****ηLED****'****☐****任L0.5****任1Σ**  
//      3. ****üa****^****Jλ****'****P****  
//          ****'****ij****'****ü1t**  
//          ****λ****^****沙****^****r****Q****^****  
  
// Author:   M****A****B****E****L****  
// Version:  1.0.0.20201228  
// Date:    ****2020-12-28  
  
// History**  
  
//  
//*****  
  
//*****  
//  
// u1**  
//  
//*****  
  
#include <stdint.h>  
#include <stdbool.h>  
  
#include "inc/hw_memmap.h"           // ****랜**  
  
#include "inc/hw_types.h"            // ****.랜王j****  
  
#include "driverlib/debug.h"         // ****  
  
#include "driverlib/gpio.h"         // ****IO랜**
```



```

◆◆◆◆ 8x16 ◆◆ ASCII ◆◆
display_GB2312_string(3,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0); //◆◆'h'◆◆ 16x16 ◆◆◆◆◆◆ 8x16 ◆◆ ASCII
◆◆.◆◆◆◆◆◆"

display_GB2312_string(5,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0);
display_GB2312_string(7,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0);
break;
case 6:
    lcd_act=0;
    clear_screen(); //clear all dots
    display_GB2312_string(1,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0);
    display_GB2312_string(3,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0);
    display_GB2312_string(5,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0);
    display_GB2312_string(7,1,"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆",0);
    break;
default: break;
}
TEST_L;
}

}

}

//*****
//
// ◆◆◆◆◆◆void GPIOInit(void)
// ◆◆◆◆◆◆GPIO'◆◆◆◆◆◆PortK◆◆◆◆◆◆PK4,PK5'◆◆◆◆◆◆PortM◆◆◆◆◆◆PM0'◆◆◆◆◆◆
// ◆◆PK4◆◆◆◆◆◆TM1638◆◆◆◆◆◆STB◆◆◆◆◆◆PK5◆◆◆◆◆◆TM1638◆◆◆◆◆◆DIO◆◆◆◆◆◆PM0◆◆◆◆◆◆TM1638◆◆◆◆◆◆CLK◆◆
// ◆◆◆◆◆◆◆◆◆◆◆◆
// ◆◆◆◆◆◆◆◆◆◆◆◆
//
//*****
void GPIOInit(void)
{
    //◆◆◆◆◆◆TM1638oT:◆◆
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK); // '◆◆.◆◆ K
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOK)); // ◆◆◆◆◆◆ K◆◆◆◆◆◆

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOM); // '◆◆.◆◆ M
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOM)); // ◆◆◆◆◆◆ M◆◆◆◆◆◆

    // ◆◆◆◆◆◆ K◆◆j◆◆4,5◆◆◆◆◆◆PK4,PK5◆◆◆◆◆◆I◆◆◆◆◆◆◆◆ PK4-STB PK5-DIO
    GPIOPinTypeGPIOOutput(GPIO_PORTK_BASE, GPIO_PIN_4|GPIO_PIN_5);
    // ◆◆◆◆◆◆ M◆◆j◆◆0◆◆◆◆◆◆PM0◆◆◆◆◆◆I◆◆◆◆◆◆◆◆ PM0-CLK
    GPIOPinTypeGPIOOutput(GPIO_PORTM_BASE, GPIO_PIN_0);
}

//*****
//
// ◆◆◆◆◆◆SysTickInit(void)
// ◆◆◆◆◆◆◆◆◆◆◆◆SysTick◆◆◆◆◆◆
// ◆◆◆◆◆◆◆◆◆◆◆◆
// ◆◆◆◆◆◆◆◆◆◆◆◆
//
//*****
void SysTickInit(void)
{
    SysTickPeriodSet(ui32SysClock/SYSTICK_FREQUENCY); // ◆◆◆◆◆◆◆◆◆◆◆◆,◆◆h◆◆◆◆◆◆20ms
    SysTickEnable(); // SysTick'◆◆◆◆◆◆

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```
SysTickIntEnable();          // SysTick
}

//*****
//
// void DevicesInit(void)
// 系统初始化函数，包括GPIO、SysTick、中断控制器等
//
//
//
// *****
void DevicesInit(void)
{
    // 25MHz 晶振，PLL 倍频 2 倍，得到 50MHz
    ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ | SYSCTL_OSC_MAIN |
                                        SYSCTL_USE_PLL | SYSCTL_CFG_VCO_480),
                                        20000000);

    GPIOInit();              // GPIO
    SysTickInit();           // SysTick
    IntMasterEnable();       // 开启中断

}

//*****
//
// void SysTick_Handler(void)
// SysTick 中断处理函数
//
//
//
// *****
void SysTick_Handler(void)    // 10ms 中断
{
    // 0.1s 中断标志
    if (++clock100ms >= V_T100ms)
    {
        clock100ms_flag = 1; // 0.1s 中断标志
        clock100ms = 0;
    }

    // 0.5s 中断标志
    if (++clock500ms >= V_T500ms)
    {
        clock500ms_flag = 1; // 0.5s 中断标志
        clock500ms = 0;
    }

    // 2s 中断标志
    if (++clock2s >= V_T2s)
    {
        clock2s_flag = 1; // 2s 中断标志
        clock2s = 0;
    }

    // 数码管 LED 刷新
    TM1638_RefreshDIGandLED(digit, pnt, led);

    // 扫描数码管 0-9 的段码
    // 数码管段码
    key_code = TM1638_Readkeyboard();
}
```

```
    digit[5] = key_code;
}
```

JLX12864.c

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/debug.h"
#include "driverlib/gpio.h"
#include "driverlib/pin_map.h"
#include "driverlib/sysctl.h"

#define LCD_CS_H GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4,GPIO_PIN_4)
#define LCD_CS_L GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_4,0)
#define LCD_SCK_H GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_5,GPIO_PIN_5)
#define LCD_SCK_L GPIOPinWrite(GPIO_PORTB_BASE,GPIO_PIN_5,0)
#define LCD_SDA_H GPIOPinWrite(GPIO_PORTC_BASE,GPIO_PIN_4,GPIO_PIN_4)
#define LCD_SDA_L GPIOPinWrite(GPIO_PORTC_BASE,GPIO_PIN_4,0)
#define LCD_RST_H GPIOPinWrite(GPIO_PORTK_BASE,GPIO_PIN_0,GPIO_PIN_0)
#define LCD_RST_L GPIOPinWrite(GPIO_PORTK_BASE,GPIO_PIN_0,0)
#define LCD_RS_H GPIOPinWrite(GPIO_PORTK_BASE,GPIO_PIN_1,GPIO_PIN_1)
#define LCD_RS_L GPIOPinWrite(GPIO_PORTK_BASE,GPIO_PIN_1,0)

#define LCD_ROM_IN_H GPIOPinWrite(GPIO_PORTK_BASE,GPIO_PIN_2,GPIO_PIN_2)
#define LCD_ROM_IN_L GPIOPinWrite(GPIO_PORTK_BASE,GPIO_PIN_2,0)
#define LCD_ROM_SCK_H GPIOPinWrite(GPIO_PORTC_BASE,GPIO_PIN_5,GPIO_PIN_5)
#define LCD_ROM_SCK_L GPIOPinWrite(GPIO_PORTC_BASE,GPIO_PIN_5,0)
#define LCD_ROM_CS_H GPIOPinWrite(GPIO_PORTC_BASE,GPIO_PIN_4,GPIO_PIN_4)
#define LCD_ROM_CS_L GPIOPinWrite(GPIO_PORTC_BASE,GPIO_PIN_4,0)

#define LCD_ROM_OUT GPIOPinRead(GPIO_PORTC_BASE,GPIO_PIN_6)

#define TEST_H GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1,GPIO_PIN_1)
#define TEST_L GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1,0)

uint8_t bmp1[];

uint8_t jiong1[] = { //-- 0000: 00 --
//-- 0000 12; 00000000j000j000i0000 x 00=16x16 --
0x00,0xFE,0x82,0x42,0xA2,0x9E,0xA8,0x82,0x86,0xA8,0xB2,0x62,0x02,0xFE,0x00,0x00,
0x00,0x7F,0x40,0x40,0x7F,0x40,0x40,0x40,0x40,0x40,0x7F,0x40,0x40,0x7F,0x00,0x00
};

uint8_t lei1[] = { //-- 0000: 00 --
//-- 0000 12; 00000000j000j000i0000 x 00=16x16 --
0x00,0x80,0x80,0xBF,0xA5,0xA5,0xA5,0x3F,0xA5,0xA5,0xBF,0x80,0x80,0x80,0x00,
0x7F,0x24,0x24,0x3F,0x24,0x24,0x7F,0x00,0x7F,0x24,0x24,0x3F,0x24,0x7F,0x00
};

////'00GPIO
void LCD_PORT_init(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOC));
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK);
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOK));
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOB));
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOE));

    GPIOPinTypeGPIOOutput(GPIO_PORTK_BASE,GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2);
    GPIOPinTypeGPIOOutput(GPIO_PORTC_BASE,GPIO_PIN_5 | GPIO_PIN_4);
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE,GPIO_PIN_4 | GPIO_PIN_5);
    GPIOPinTypeGPIOOutput(GPIO_PORTC_BASE,GPIO_PIN_4);
    GPIOPinTypeGPIOInput(GPIO_PORTK_BASE,GPIO_PIN_6);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOF));
}
```

```
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE,GPIO_PIN_1);
}

//*****
void LCD_delay(int n_ms)
{
    int j,k;
    TEST_L;
    for(j=0; j<n_ms; j++)
        for(k=0; k<110; k++);
    TEST_H;
}

//*****
void LCD_delay_us(int n_us)
{
    int j,k;
    for(j=0; j<n_us; j++)
        for(k=0; k<1; k++);
}

//LCD 命令
void transfer_command_lcd(int data1)
{
    char i;
    LCD_CS_L;
    LCD_RS_L;
    for(i=0; i<8; i++)
    {
        LCD_SCK_L;
        //delay_us(10); //*****
        if(data1&0x80) LCD_SDA_H;
        else LCD_SDA_L;
        LCD_SCK_H;
        //delay_us(10); //*****
        data1=data1<<=1;
    }
    LCD_CS_H;
}

//LCD 数据
void transfer_data_lcd(int data1)
{
    char i;
    LCD_CS_L;
    LCD_RS_H;
    for(i=0; i<8; i++)
    {
        LCD_SCK_L;
        if(data1&0x80) LCD_SDA_H;
        else LCD_SDA_L;
        LCD_SCK_H;
        data1=data1<<=1;
    }
    LCD_CS_H;
}

//LCD 初始化
void initial_lcd()
{
    LCD_PORT_init();
    LCD_RST_L;
    LCD_delay(100);
    LCD_RST_H;
    LCD_delay(100);

    transfer_command_lcd(0xe2); //*****
    LCD_delay(5);
    transfer_command_lcd(0x2c); //***** 1
    LCD_delay(50);
    transfer_command_lcd(0x2e); //***** 2
    LCD_delay(50);
    transfer_command_lcd(0x2f); //***** 3
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```

LCD_delay(5);
transfer_command_lcd(0x23); //0x23
transfer_command_lcd(0x81); //0x81
transfer_command_lcd(0x28); //0x28
transfer_command_lcd(0xa2); //1/9 bias
transfer_command_lcd(0xc8); //i.e.
transfer_command_lcd(0xa0); //i.e.
transfer_command_lcd(0x40); //'Yh'n'
transfer_command_lcd(0xaf); //''

}
void lcd_address(uint32_t page,uint32_t column)
{
    column=column-0x01;
    transfer_command_lcd(0xb0+page-1); //0xb0+page-1 8 64 8 x
    transfer_command_lcd(0x10+(column>>4&0xf)); //e.k 4
    transfer_command_lcd(column&0xf); //e.j 4
}
//0
void clear_screen()
{
    unsigned char i,j;
    for(i=0; i<9; i++)
    {
        transfer_command_lcd(0xb0+i);
        transfer_command_lcd(0x10);
        transfer_command_lcd(0x00);
        for(j=0; j<132; j++)
        {
            transfer_data_lcd(0x00);
        }
    }
}
//' 128x64 e
void display_128x64(uint8_t *dp)
{
    uint32_t i,j;
    for(j=0; j<8; j++)
    {
        lcd_address(j+1,1);
        for (i=0; i<128; i++)
        {
            transfer_data_lcd(*dp); //д LCD,д 8 л 2 1
            dp++;
        }
    }
}
//' 16x16 e 2 16x16 e
void display_graphic_16x16(uint8_t page,uint8_t column,uint8_t *dp)
{
    uint32_t i,j;
    for(j=0; j<2; j++)
    {
        lcd_address(page+j,column);
        for (i=0; i<16; i++)
        {
            transfer_data_lcd(*dp); //д LCD,д 8 л 2 1
            dp++;
        }
    }
}
//' 8x16 e ASCII, 8x16 e
void display_graphic_8x16(uint8_t page,uint8_t column,uint8_t *dp)
{
    uint32_t i,j;

```

```

        for(j=0; j<2; j++)
        {
            lcd_address(page+j,column);
            for (i=0; i<8; i++)
            {
                transfer_data_lcd(*dp); //液晶 LCD, 8 位数据, 2 字节
                dp++;
            }
        }
    }

// 5x8 ASCII, 5x8
void display_graphic_5x8(uint8_t page,uint8_t column,uint8_t *dp)
{
    uint32_t i;
    lcd_address(page,column);
    for (i=0; i<6; i++)
    {
        transfer_data_lcd(*dp);
        dp++;
    }
}

// 液晶 IC
void send_command_to_ROM( uint8_t datu )
{
    uint8_t i;
    for(i=0; i<8; i++ )
    {
        LCD_ROM_SCK_L;
        LCD_delay_us(10);
        if(datu&0x80)LCD_ROM_IN_H;
        else LCD_ROM_IN_L;
        datu = datu<<1;
        LCD_ROM_SCK_H;
        LCD_delay_us(10);
    }
}

// 液晶 IC
static uint8_t get_data_from_ROM( )
{
    uint8_t i;
    uint8_t ret_data=0;
    for(i=0; i<8; i++)
    {
        LCD_ROM_SCK_L;
        //delay_us(1);
        ret_data=ret_data<<1;
        if( LCD_ROM_OUT )
            ret_data=ret_data+1;
        else
            ret_data=ret_data+0;
        LCD_ROM_SCK_H;
        //delay_us(1);
    }
    return(ret_data);
}

// 液晶 LCD, 8 位数据, 2 字节
void get_and_write_16x16(unsigned long fontaddr,uint8_t page,uint8_t column,bool inverse)
{
    uint8_t i,j,disp_data;
    LCD_ROM_CS_L;
    send_command_to_ROM(0x03);

    send_command_to_ROM((fontaddr&0xff0000)>>16); // 8 位, 24 位
    send_command_to_ROM((fontaddr&0xff00)>>8); // 8 位, 24 位
    send_command_to_ROM(fontaddr&0xff); // 8 位, 24 位
    for(j=0; j<2; j++)
    {
        lcd_address(page+j,column);
        for(i=0; i<16; i++ )
        {

```



```

        if( inverse)
            disp_data=~get_data_from_ROM();
        else
            disp_data=get_data_from_ROM();
        transfer_data_lcd(disp_data); //LCD, ýd 1 2e2 1
    }
}
LCD_ROM_CS_H;
}

//page, column)
void get_and_write_8x16(unsigned long fontaddr, uint8_t page, uint8_t column, bool inverse)
{
    uint8_t i, j, disp_data;
    LCD_ROM_CS_L;
    send_command_to_ROM(0x03);
    send_command_to_ROM((fontaddr&0xff000)>>16); //k 8 λ, 24 λ
    send_command_to_ROM((fontaddr&0xff00)>>8); // 8 λ, 24 λ
    send_command_to_ROM(fontaddr&0xff); //j 8 λ, 24 λ
    for(j=0; j<2; j++)
    {
        lcd_address(page+j, column);
        for(i=0; i<8; i++)
        {
            if( inverse)
                disp_data=~get_data_from_ROM();
            else
                disp_data=get_data_from_ROM();
            transfer_data_lcd(disp_data); //LCD, ýd 1 2e2 1
        }
    }
    LCD_ROM_CS_H;
}

//page, column)
void get_and_write_5x8(unsigned long fontaddr, uint8_t page, uint8_t column, bool inverse)
{
    uint8_t i, disp_data;
    LCD_ROM_CS_L;
    send_command_to_ROM(0x03);
    send_command_to_ROM((fontaddr&0xff000)>>16); //k 8 λ, 24 λ
    send_command_to_ROM((fontaddr&0xff00)>>8); // 8 λ, 24 λ
    send_command_to_ROM(fontaddr&0xff); //j 8 λ, 24 λ
    lcd_address(page, column);
    for(i=0; i<5; i++)
    {
        if( inverse)
            disp_data=~get_data_from_ROM();
        else
            disp_data=get_data_from_ROM();
        transfer_data_lcd(disp_data); //LCD, ýd 1 2e2 1
    }
    LCD_ROM_CS_H;
}

//*****
unsigned long fontaddr=0;
void display_GB2312_string(uint8_t page, uint8_t column, uint8_t *text, bool inverse)
{
    uint8_t i= 0;
    while((text[i]>0x00))
    {
        if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
        {
            // GB2312 V IC ej '
            //Address = ((MSB - 0xb0) * 94 + (LSB - 0xa1)+ 846)*32+ BaseAdd; BaseAdd=0
            //8  λ T r _ :: t d .
            fontaddr = (text[i]- 0xb0)*94;
            fontaddr += (text[i+1]-0xa1)+846;

```

[illegible]


```
// History
//
//*****

//*****
//
// u1
//
//*****
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h" // 寄存器
#include "inc/hw_types.h" // 寄存器, 寄存器
#include "driverlib/debug.h" // 寄存器
#include "driverlib/gpio.h" // IO
#include "driverlib/pin_map.h" // TM4C MCU 引脚
#include "driverlib/sysctl.h" // 系统控制
#include "driverlib/systick.h" // SysTick Driver
#include "driverlib/interrupt.h" // NVIC Interrupt Controller Driver

#include "tm1638.h" // TM1638 7-segment display
#include "JLX12864_2.c"
#include "modes_2.c"

//*****
//
// 寄存器
//
//*****
#define SYSTICK_FREQUENCY 50 // SysTick 50Hz 20ms

#define V_T10ms 5 // 0.1s 20ms
#define V_T50ms 25 // 0.5s 20ms
#define V_T2s 100 // 2s 20ms
#define V_T10s 500 // 10s 20ms
//*****
//
// 寄存器
//
//*****
void GPIOInit(void); // GPIO
void SysTickInit(void); // SysTick
void DevicesInit(void); // MCU
//*****
//
// 寄存器
//
//*****
uint8_t s_jtu[128*8];
uint8_t bit[8];
// 寄存器
uint8_t clock10ms = 0;
uint8_t clock50ms = 0;
uint8_t clock2s = 0;
uint16_t clock10s = 0;

// 寄存器
uint8_t clock10ms_flag = 0;
uint8_t clock50ms_flag = 0;
uint8_t clock2s_flag = 1;
uint8_t clock10s_flag = 0;
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```
//LCD
uint8_t lcd_act = 0;
bool if_refresh = true;    // 刷新标志

// 测试计数器
uint32_t test_counter = 0;

// 8个LED
uint8_t digit[8]={ ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' };

// 8个LED 1 0
uint8_t pnt = 0x04;

// 8个LED 1 0 0 0 0 0 0 0
uint8_t led[] = {1, 1, 1, 1, 1, 1, 1, 0};

// 键值
uint8_t key_code = 0;

// 系统时钟
uint32_t ui32SysClock;

// 字母数字
uint8_t* letter[3]={"A", "B", "C"};
uint8_t* number[10] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};
uint8_t let = 0;
uint8_t integer = 0;
uint8_t decimal = 0;

//*****
//
// *****
int main(void)
{
    uint8_t temp,i;
    uint8_t row;
    uint8_t line;

    DevicesInit();    // MCU初始化

    while (clock100ms < 3);    // 延时60ms, TM1638初始化
    TM1638_Init();    // TM1638
    initial_lcd();

    for (row=0;row<128;++row)
    {
        for(line=0;line<8;++line)
        {
            sjtu[line*128+127-row]=~bmp1[row*8+line];
            bit[0]=sjtu[line*128+127-row]%2;
            bit[1]=sjtu[line*128+127-row]%4/2;
            bit[2]=sjtu[line*128+127-row]%8/4;
            bit[3]=sjtu[line*128+127-row]%16/8;
            bit[4]=sjtu[line*128+127-row]%32/16;
            bit[5]=sjtu[line*128+127-row]%64/32;
            bit[6]=sjtu[line*128+127-row]%128/64;
            bit[7]=sjtu[line*128+127-row]/128;
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240


```
// *****
// *****
//
//*****
void GPIOInit(void)
{
    //TM1638T
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK);          // ' K
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOK)){};    // L. K

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOM);          // ' M
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOM)){};    // L. M

    // K j 4,5 PK4,PK5 I PK4-STB PK5-DIO
    GPIOPinTypeGPIOOutput(GPIO_PORTK_BASE, GPIO_PIN_4|GPIO_PIN_5);
    // M j 0 PM0 I PM0-CLK
    GPIOPinTypeGPIOOutput(GPIO_PORTM_BASE, GPIO_PIN_0);
}

//*****
//
// *****
// SysTickInit(void)
// *****
// SysTick
// *****
// *****
// *****
// *****
// *****
void SysTickInit(void)
{
    SysTickPeriodSet(ui32SysClock/SYSTICK_FREQUENCY); // , h 20ms
    SysTickEnable();          // SysTick'
    SysTickIntEnable();       // SysTick*
}

//*****
//
// *****
// void DevicesInit(void)
// *****
// CU' , T h ä GPIO' , SysTick*
// *****
// *****
// *****
// *****
// *****
void DevicesInit(void)
{
    // ' #25MHz h 7 PLL 20MHz
    ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |SYSCTL_OSC_MAIN |
        SYSCTL_USE_PLL |SYSCTL_CFG_VCO_480),
        20000000);

    GPIOInit();          // GPIO'
    SysTickInit();       // SysTick*
    IntMasterEnable();    // *
}

//*****
//
// *****
// void SysTick_Handler(void)
// *****
// SysTick*
// *****
```



```
// *****
//
//*****
void SysTick_Handler(void)      // 120ms
{
    uint8_t key_code_pre = key_code;    // "j"

    // 0.1s
    if (++clock100ms >= V_T100ms)
    {
        clock100ms_flag = 1; // 0.1s
        clock100ms = 0;
    }

    // 0.5s
    if (++clock500ms >= V_T500ms)
    {
        clock500ms_flag = 1; // 0.5s
        clock500ms = 0;
    }

    // 2s
    if (++clock2s >= V_T2s)
    {
        clock2s_flag = 1; // 2s
        clock2s = 0;
    }

    // 10s
    if (++clock10s >= V_T10s){
        clock10s_flag = 1;
        clock10s = 0;
    }

    // *LED:
    TM1638_RefreshDIGIandLED(digit, pnt, led);

    // 0-9-10
    // 1638_Readkeyboard()
    key_code = TM1638_Readkeyboard();

    digit[5] = key_code;

    // *n:
    // 0:w
    // 1:w'
    // 2:wλ
    // 3:wCλ

    // 0 1
    if (lcd_act == 0 && key_code != 0) {
        ++lcd_act;
        if_refresh = true;
        return;
    }

    // ', 1-3
    if (key_code_pre != 4 && key_code == 4) {
        lcd_act = (--lcd_act + 2) % 3 + 1;
        if_refresh = true;
        return;
    }

    // ', 1-3
```

```
if (key_code_pre != 6 && key_code == 6) {
    lcd_act = (--lcd_act + 1) % 3 + 1;
    if_refresh = true;
    return;
}

// 00000000n0

if (key_code_pre != 8 && key_code == 8){
    switch (lcd_act) {
        case 1: let = (let + 2) % 3; break;
        case 2: integer = (integer + 9) % 10; break;
        case 3: decimal = (decimal + 9) % 10; break;
    }
    if_refresh = true;
}
if (key_code_pre != 2 && key_code == 2){
    switch (lcd_act) {
        case 1: let = ++let % 3; break;
        case 2: integer = ++integer % 10; break;
        case 3: decimal = ++ decimal % 10; break;
    }
    if_refresh = true;
}
}
```

modes_3.c

```
#include "JLX12864_2.c"
#include "tm1638.h"

// g'0
void mode0(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    display_GB2312_string(3, 1, "0000g'00", 0);
    display_GB2312_string(3, 81, "g'", 0);
    display_GB2312_string(3, 113, letter, 0);
    display_GB2312_string(5, 1, "0000000000", 0);
    display_GB2312_string(5, 81, integer, 0);
    display_GB2312_string(5, 89, ".", 0);
    display_GB2312_string(5, 97, decimal, 0);
    display_GB2312_string(5, 105, "Hz", 0);
}

// g'1
void mode1(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    display_GB2312_string(3, 1, "0000g'00", 0);
    display_GB2312_string(3, 81, "g'", 1);
    display_GB2312_string(3, 113, letter, 1);
    display_GB2312_string(5, 1, "0000000000", 0);
    display_GB2312_string(5, 81, integer, 0);
    display_GB2312_string(5, 89, ".", 0);
    display_GB2312_string(5, 97, decimal, 0);
    display_GB2312_string(5, 105, "Hz", 0);
}

// g'2
void mode2(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    display_GB2312_string(3, 1, "0000g'00", 0);
    display_GB2312_string(3, 81, "g'", 0);
    display_GB2312_string(3, 113, letter, 0);
    display_GB2312_string(5, 1, "0000000000", 0);
    display_GB2312_string(5, 81, integer, 1);
    display_GB2312_string(5, 89, ".", 0);
    display_GB2312_string(5, 97, decimal, 0);
    display_GB2312_string(5, 105, "Hz", 0);
}
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```

}

// g'3
void mode3(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    display_GB2312_string(3, 1, "g'", 0);
    display_GB2312_string(3, 81, "g'", 0);
    display_GB2312_string(3, 113, letter, 0);
    display_GB2312_string(5, 1, "g'", 0);
    display_GB2312_string(5, 81, integer, 0);
    display_GB2312_string(5, 89, ".", 0);
    display_GB2312_string(5, 97, decimal, 1);
    display_GB2312_string(5, 105, "Hz", 0);
}

```

1.4.3 实验任务 4_3

exp4_3.c

```

//*****
//
// Copyright: 2020-2021, M...
// File name: exp2_0.c
// Description:
// 1. ...
// 2. ...
// 3. ...
// ...
// Author: M...
// Version: 1.0.0.20201228
// Date: 2020-12-28
// History:
//
//*****

//*****
//
// u1
//
//*****
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h" // ...
#include "inc/hw_types.h" // ...
#include "driverlib/debug.h" // ...
#include "driverlib/gpio.h" // ...
#include "driverlib/pin_map.h" // TM4C...
#include "driverlib/sysctl.h" // eT...
#include "driverlib/systick.h" // SysTick Driver
#include "driverlib/interrupt.h" // NVIC Interrupt Controller Driver

#include "tm1638.h" // ...
#include "JLX12864_2.c"
#include "modes_3.c"

//*****
//
// ...
//

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240


```
// 定义键码
uint8_t key_code = 0;

// 定义系统时钟
uint32_t ui32SysClock;

// 定义字母和数字数组
uint8_t* letter[3]={"A", "B", "C"};
uint8_t* number[10] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};
uint8_t let = 0;
uint8_t integer = 1;
uint8_t decimal = 0;

uint8_t let_temp = 0; // 定义临时变量
uint8_t int_temp = 1;
uint8_t dec_temp = 0;

// 定义函数指针
void (*func[5][4])(uint8_t*, uint8_t*, uint8_t*) = {0};

// *****
// *****
// *****
// *****
int main(void)
{
    uint8_t temp,i;
    uint8_t row;
    uint8_t line;

    DevicesInit(); // MCU 初始化

    while (clock100ms < 3); // 等待 60ms, TM1638 初始化
    TM1638_Init(); // 初始化 TM1638
    initial_lcd();

    for (row=0;row<128;++row)
    {
        for(line=0;line<8;++line)
        {
            sjtu[line*128+127-row]=~bmp1[row*8+line];
            bit[0]=sjtu[line*128+127-row]%2;
            bit[1]=sjtu[line*128+127-row]%4/2;
            bit[2]=sjtu[line*128+127-row]%8/4;
            bit[3]=sjtu[line*128+127-row]%16/8;
            bit[4]=sjtu[line*128+127-row]%32/16;
            bit[5]=sjtu[line*128+127-row]%64/32;
            bit[6]=sjtu[line*128+127-row]%128/64;
            bit[7]=sjtu[line*128+127-row]/128;
            sjtu[line*128+127-row]=bit[0]*128+bit[1]*64+bit[2]*32+bit[3]*16+bit[4]*8+bit[5]*4+bit[6]*2+bit[7];
        }
    }

    // *****
    func[0][0] = mode0_0;
    func[0][1] = mode0_1;
    func[1][0] = mode1_0;
    func[1][1] = mode1_1;
    func[1][2] = mode1_2;
    func[2][0] = mode2_0;
    func[2][1] = mode2_1;
    func[2][2] = mode2_2;
    func[3][0] = mode3_0;
    func[3][1] = mode3_1;
    func[3][2] = mode3_2;
    func[3][3] = mode3_3;
    func[4][0] = mode4;
```

```

while (1)
{
    if (clock100ms_flag == 1)    // 0.1ms延时
    {
        clock100ms_flag = 0;
        // 0.1ms延时
        if (key_code == 0)
        {
            if (++test_counter >= 10000) test_counter = 0;
            digit[0] = test_counter / 1000;    // 千位
            digit[1] = test_counter / 100 % 10;    // 百位
            digit[2] = test_counter / 10 % 10;    // 十位
            digit[3] = test_counter % 10;    // 个位
        }
    }

    if (clock500ms_flag == 1)    // 0.5ms延时
    {
        clock500ms_flag = 0;
        // 0.5ms延时
        temp = led[0];
        for (i = 0; i < 7; i++) led[i] = led[i + 1];
        led[7] = temp;
    }

    if (if_refresh || (if_count_5s && clock5s_flag) || (if_count_10s && clock10s_flag))
    {
        if (if_count_5s && clock5s_flag){
            lcd_act = 3;
            lcd_subact = 0;
            if_count_5s = false;
        }
        if (if_count_10s && clock10s_flag){
            lcd_act = 0;
            lcd_subact = 0;
            if_count_10s = false;
        }
        if_refresh = false;
        TEST_H;

        func[lcd_act][lcd_subact](letter[let], number[integer], number[decimal]);

        if (lcd_act == 0 && lcd_subact == 1) {if_count_10s = true; if_count_5s = false; clock10s = 0; clock10s_flag = false;}
        else if (lcd_act == 4) {if_count_5s = true; if_count_10s = false; clock5s = 0; clock5s_flag = false;}
        else if_count_5s = if_count_10s = false;
        TEST_L;
    }
}

//*****
//
// 初始化GPIO
// 配置GPIO引脚为推挽输出模式
// 配置GPIO引脚为推挽输出模式
// 配置GPIO引脚为推挽输出模式
// 配置GPIO引脚为推挽输出模式
// 配置GPIO引脚为推挽输出模式
// 配置GPIO引脚为推挽输出模式
// 配置GPIO引脚为推挽输出模式
//*****
void GPIOInit(void)
{
    //配置GPIO引脚为推挽输出模式

```

```

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK);                // 'K
while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOK)){};        // K

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOM);                // 'M
while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOM)){};        // M

// K4,5,PK4,PK5,I PK4-STB PK5-DIO
GPIOPinTypeGPIOOutput(GPIO_PORTK_BASE, GPIO_PIN_4|GPIO_PIN_5);
// M0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103,1104,1105,1106,1107,1108,1109,1110,1111,1112,1113,1114,1115,1116,1117,1118,1119,1120,1121,1122,1123,1124,1125,1126,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,1139,1140,1141,1142,1143,1144,1145,1146,1147,1148,1149,1150,1151,1152,1153,1154,1155,1156,1157,1158,1159,1160,1161,1162,1163,1164,1165,1166,1167,1168,1169,1170,1171,1172,1173,1174,1175,1176,1177,1178,1179,1180,1181,1182,1183,1184,1185,1186,1187,1188,1189,1190,1191,1192,1193,1194,1195,1196,1197,1198,1199,1200,1201,1202,1203,1204,1205,1206,1207,1208,1209,1210,1211,1212,1213,1214,1215,1216,1217,1218,1219,1220,1221,1222,1223,1224,1225,1226,1227,1228,1229,1230,1231,1232,1233,1234,1235,1236,1237,1238,1239,1240,1241,1242,1243,1244,1245,1246,1247,1248,1249,1250,1251,1252,1253,1254,1255,1256,1257,1258,1259,1260,1261,1262,1263,1264,1265,1266,1267,1268,1269,1270,1271,1272,1273,1274,1275,1276,1277,1278,1279,1280,1281,1282,1283,1284,1285,1286,1287,1288,1289,1290,1291,1292,1293,1294,1295,1296,1297,1298,1299,1300,1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321,1322,1323,1324,1325,1326,1327,1328,1329,1330,1331,1332,1333,1334,1335,1336,1337,1338,1339,1340,1341,1342,1343,1344,1345,1346,1347,1348,1349,1350,1351,1352,1353,1354,1355,1356,1357,1358,1359,1360,1361,1362,1363,1364,1365,1366,1367,1368,1369,1370,1371,1372,1373,1374,1375,1376,1377,1378,1379,1380,1381,1382,1383,1384,1385,1386,1387,1388,1389,1390,1391,1392,1393,1394,1395,1396,1397,1398,1399,1400,1401,1402,1403,1404,1405,1406,1407,1408,1409,1410,1411,1412,1413,1414,1415,1416,1417,1418,1419,1420,1421,1422,1423,1424,1425,1426,1427,1428,1429,1430,1431,1432,1433,1434,1435,1436,1437,1438,1439,1440,1441,1442,1443,1444,1445,1446,1447,1448,1449,1450,1451,1452,1453,1454,1455,1456,1457,1458,1459,1460,1461,1462,1463,1464,1465,1466,1467,1468,1469,1470,1471,1472,1473,1474,1475,1476,1477,1478,1479,1480,1481,1482,1483,1484,1485,1486,1487,1488,1489,1490,1491,1492,1493,1494,1495,1496,1497,1498,1499,1500,1501,1502,1503,1504,1505,1506,1507,1508,1509,1510,1511,1512,1513,1514,1515,1516,1517,1518,1519,1520,1521,1522,1523,1524,1525,1526,1527,1528,1529,1530,1531,1532,1533,1534,1535,1536,1537,1538,1539,1540,1541,1542,1543,1544,1545,1546,1547,1548,1549,1550,1551,1552,1553,1554,1555,1556,1557,1558,1559,1560,1561,1562,1563,1564,1565,1566,1567,1568,1569,1570,1571,1572,1573,1574,1575,1576,1577,1578,1579,1580,1581,1582,1583,1584,1585,1586,1587,1588,1589,1590,1591,1592,1593,1594,1595,1596,1597,1598,1599,1600,1601,1602,1603,1604,1605,1606,1607,1608,1609,1610,1611,1612,1613,1614,1615,1616,1617,1618,1619,1620,1621,1622,1623,1624,1625,1626,1627,1628,1629,1630,1631,1632,1633,1634,1635,1636,1637,1638,1639,1640,1641,1642,1643,1644,1645,1646,1647,1648,1649,1650,1651,1652,1653,1654,1655,1656,1657,1658,1659,1660,1661,1662,1663,1664,1665,1666,1667,1668,1669,1670,1671,1672,1673,1674,1675,1676,1677,1678,1679,1680,1681,1682,1683,1684,1685,1686,1687,1688,1689,1690,1691,1692,1693,1694,1695,1696,1697,1698,1699,1700,1701,1702,1703,1704,1705,1706,1707,1708,1709,1710,1711,1712,1713,1714,1715,1716,1717,1718,1719,1720,1721,1722,1723,1724,1725,1726,1727,1728,1729,1730,1731,1732,1733,1734,1735,1736,1737,1738,1739,1740,1741,1742,1743,1744,1745,1746,1747,1748,1749,1750,1751,1752,1753,1754,1755,1756,1757,1758,1759,1760,1761,1762,1763,1764,1765,1766,1767,1768,1769,1770,1771,1772,1773,1774,1775,1776,1777,1778,1779,1780,1781,1782,1783,1784,1785,1786,1787,1788,1789,1790,1791,1792,1793,1794,1795,1796,1797,1798,1799,1800,1801,1802,1803,1804,1805,1806,1807,1808,1809,1810,1811,1812,1813,1814,1815,1816,1817,1818,1819,1820,1821,1822,1823,1824,1825,1826,1827,1828,1829,1830,1831,1832,1833,1834,1835,1836,1837,1838,1839,1840,1841,1842,1843,1844,1845,1846,1847,1848,1849,1850,1851,1852,1853,1854,1855,1856,1857,1858,1859,1860,1861,1862,1863,1864,1865,1866,1867,1868,1869,1870,1871,1872,1873,1874,1875,1876,1877,1878,1879,1880,1881,1882,1883,1884,1885,1886,1887,1888,1889,1890,1891,1892,1893,1894,1895,1896,1897,1898,1899,1900,1901,1902,1903,1904,1905,1906,1907,1908,1909,1910,1911,1912,1913,1914,1915,1916,1917,1918,1919,1920,1921,1922,1923,1924,1925,1926,1927,1928,1929,1930,1931,1932,1933,1934,1935,1936,1937,1938,1939,1940,1941,1942,1943,1944,1945,1946,1947,1948,1949,1950,1951,1952,1953,1954,1955,1956,1957,1958,1959,1960,1961,1962,1963,1964,1965,1966,1967,1968,1969,1970,1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025,2026,2027,2028,2029,2030,2031,2032,2033,2034,2035,2036,2037,2038,2039,2040,2041,2042,2043,2044,2045,2046,2047,2048,2049,2050,2051,2052,2053,2054,2055,2056,2057,2058,2059,2060,2061,2062,2063,2064,2065,2066,2067,2068,2069,2070,2071,2072,2073,2074,2075,2076,2077,2078,2079,2080,2081,2082,2083,2084,2085,2086,2087,2088,2089,2090,2091,2092,2093,2094,2095,2096,2097,2098,2099,2100,2101,2102,2103,2104,2105,2106,2107,2108,2109,2110,2111,2112,2113,2114,2115,2116,2117,2118,2119,2120,2121,2122,2123,2124,2125,2126,2127,2128,2129,2130,2131,2132,2133,2134,2135,2136,2137,2138,2139,2140,2141,2142,2143,2144,2145,2146,2147,2148,2149,2150,2151,2152,2153,2154,2155,2156,2157,2158,2159,2160,2161,2162,2163,2164,2165,2166,2167,2168,2169,2170,2171,2172,2173,2174,2175,2176,2177,2178,2179,2180,2181,2182,2183,2184,2185,2186,2187,2188,2189,2190,2191,2192,2193,2194,2195,2196,2197,2198,2199,2200,2201,2202,2203,2204,2205,2206,2207,2208,2209,2210,2211,2212,2213,2214,2215,2216,2217,2218,2219,2220,2221,2222,2223,2224,2225,2226,2227,2228,2229,2230,2231,2232,2233,2234,2235,2236,2237,2238,2239,2240,2241,2242,2243,2244,2245,2246,2247,2248,2249,2250,2251,2252,2253,2254,2255,2256,2257,2258,2259,2260,2261,2262,2263,2264,2265,2266,2267,2268,2269,2270,2271,2272,2273,2274,2275,2276,2277,2278,2279,2280,2281,2282,2283,2284,2285,2286,2287,2288,2289,2290,2291,2292,2293,2294,2295,2296,2297,2298,2299,2300,2301,2302,2303,2304,2305,2306,2307,2308,2309,2310,2311,2312,2313,2314,2315,2316,2317,2318,2319,2320,2321,2322,2323,2324,2325,2326,2327,2328,2329,2330,2331,2332,2333,2334,2335,2336,2337,2338,2339,2340,2341,2342,2343,2344,2345,2346,2347,2348,2349,2350,2351,2352,2353,2354,2355,2356,2357,2358,2359,2360,2361,2362,2363,2364,2365,2366,2367,2368,2369,2370,2371,2372,2373,2374,2375,2376,2377,2378,2379,2380,2381,2382,2383,2384,2385,2386,2387,2388,2389,2390,2391,2392,2393,2394,2395,2396,2397,2398,2399,2400,2401,2402,2403,2404,2405,2406,2407,2408,2409,2410,2411,2412,2413,2414,2415,2416,2417,2418,2419,2420,2421,2422,2423,2424,2425,2426,2427,2428,2429,2430,2431,2432,2433,2434,2435,2436,2437,2438,2439,2440,2441,2442,2443,2444,2445,2446,2447,2448,2449,2450,2451,2452,2453,2454,2455,2456,2457,2458,2459,2460,2461,2462,2463,2464,2465,2466,2467,2468,2469,2470,2471,2472,2473,2474,2475,2476,2477,2478,2479,2480,2481,2482,2483,2484,2485,2486,2487,2488,2489,2490,2491,2492,2493,2494,2495,2496,2497,2498,2499,2500,2501,2502,2503,2504,2505,2506,2507,2508,2509,2510,2511,2512,2513,2514,2515,2516,2517,2518,2519,2520,2521,2522,2523,2524,2525,2526,2527,2528,2529,2530,2531,2532,2533,2534,2535,2536,2537,2538,2539,2540,2541,2542,2543,2544,2545,2546,2547,2548,2549,2550,2551,2552,2553,2554,2555,2556,2557,2558,2559,2560,2561,2562,2563,2564,2565,2566,2567,2568,2569,2570,2571,2572,2573,2574,2575,2576,2577,2578,2579,2580,2581,2582,2583,2584,2585,2586,2587,2588,2589,2590,2591,2592,2593,2594,2595,2596,2597,2598,2599,2600,2601,2602,2603,2604,2605,2606,2607,2608,2609,2610,2611,2612,2613,2614,2615,2616,2617,2618,2619,2620,2621,2622,2623,2624,2625,2626,2627,2628,2629,2630,2631,2632,2633,2634,2635,2636,2637,2638,2639,2640,2641,2642,2643,2644,2645,2646,2647,2648,2649,2650,2651,2652,2653,2654,2655,2656,2657,2658,2659,2660
```

邮 编：200240


```
if (key_code_pre != 2 && key_code == 2){
    if (lcd_act == 2 && lcd_subact == 0){
        let = ++let % 3;
        if_refresh = true;
    } else if (lcd_act == 3 && lcd_subact == 0){
        integer = integer % 5 + 1;
        if_refresh = true;
    } else if (lcd_act == 3 && lcd_subact == 1){
        decimal = ++decimal % 10;
        if_refresh = true;
    }
}
}
if (key_code_pre != 4 && key_code == 4){
    switch (lcd_act) {
        case 1:
            lcd_subact = (lcd_subact + 2) % 3;
            if_refresh = true;
            break;
        case 2:
            lcd_subact = (lcd_subact + 2) % 3;
            if_refresh = true;
            break;
        case 3:
            lcd_subact = (lcd_subact + 3) % 4;
            if_refresh = true;
            break;
    }
}
if (key_code_pre != 6 && key_code == 6){
    switch (lcd_act) {
        case 1:
            lcd_subact = (lcd_subact + 1) % 3;
            if_refresh = true;
            break;
        case 2:
            lcd_subact = (lcd_subact + 1) % 3;
            if_refresh = true;
            break;
        case 3:
            lcd_subact = (lcd_subact + 1) % 4;
            if_refresh = true;
            break;
    }
}
if (key_code_pre != 5 && key_code == 5) key5();
}

void key5(){
    switch (lcd_act) {
        case 0:
            switch (lcd_subact) {
                case 1:
                    lcd_act = 1;
                    lcd_subact = 0;
                    if_refresh = true;
                    break;
            }
            break;
        case 1:
            switch (lcd_subact) {
                case 0:
                    let_temp = let;
                    lcd_act = 2;
                    lcd_subact = 0;
                    if_refresh = true;
                    break;
                case 1:
                    int_temp = integer;
                    dec_temp = decimal;
                    lcd_act = 3;
                    lcd_subact = 0;
            }
        }
    }
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```

        if_refresh = true;
        break;
    case 2:
        lcd_act = 0;
        lcd_subact = 0;
        if_refresh = true;
        break;
    }
    break;
case 2:
    switch (lcd_subact) {
        case 1: // j
            lcd_act = 1;
            lcd_subact = 2;
            if_refresh = true;
            break;

        case 2: // d
            lcd_act = 1;
            lcd_subact = 2;
            let = let_temp; //
            if_refresh = true;
            break;
    }
    break;
case 3:
    switch (lcd_subact) {
        case 2: // j
            if (10 * integer + decimal >= 10 && 10 * integer + decimal <= 90) { //
                lcd_act = 1;
                lcd_subact = 2;
            }
            else {
                lcd_act = 4;
                lcd_subact = 0;
            }
            if_refresh = true;
            break;

        case 3: // d
            lcd_act = 1;
            lcd_subact = 2;
            integer = int_temp;
            decimal = dec_temp;
            if_refresh = true;
            break;
    }
    break;
default:
    break;
}
}
}

```

modes_3.c

```

#include "JLX12864_2.c"

// g'0_0
void mode0_0(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 17, "g'", 0);
    display_GB2312_string(3, 49, letter, 0);
    display_GB2312_string(3, 81, integer, 0);
    display_GB2312_string(3, 89, ".", 0);
    display_GB2312_string(3, 97, decimal, 0);
    display_GB2312_string(3, 105, "Hz", 0);
    display_GB2312_string(7, 1, "0.00", 0);
}

// g'0_1

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```
void mode0_1(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 17, "g'", 0);
    display_GB2312_string(3, 49, letter, 0);
    display_GB2312_string(3, 81, integer, 0);
    display_GB2312_string(3, 89, ".", 0);
    display_GB2312_string(3, 97, decimal, 0);
    display_GB2312_string(3, 105, "Hz", 0);
    display_GB2312_string(7, 1, "◆◆◆◆", 1);
}

// g'1_0
void mode1_0(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 17, "◆◆◆◆g'", 1);
    display_GB2312_string(5, 17, "◆◆◆◆◆◆◆◆", 0);
    display_GB2312_string(7, 97, "◆◆◆◆", 0);
}

// g'1_1
void mode1_1(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 17, "◆◆◆◆◆g'", 0);
    display_GB2312_string(5, 17, "◆◆◆◆◆◆◆◆", 1);
    display_GB2312_string(7, 97, "◆◆◆◆", 0);
}

// g'1_2
void mode1_2(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 17, "◆◆◆◆◆g'", 0);
    display_GB2312_string(5, 17, "◆◆◆◆◆◆◆◆", 0);
    display_GB2312_string(7, 97, "◆◆◆◆", 1);
}

// g'2_0
void mode2_0(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆g'◆◆", 0);
    display_GB2312_string(3, 81, "g'", 1);
    display_GB2312_string(3, 113, letter, 1);
    display_GB2312_string(7, 1, "j◆◆", 0);
    display_GB2312_string(7, 97, "d◆◆", 0);
}

// g'2_1
void mode2_1(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆g'◆◆", 0);
    display_GB2312_string(3, 81, "g'", 0);
    display_GB2312_string(3, 113, letter, 0);
    display_GB2312_string(7, 1, "j◆◆", 1);
    display_GB2312_string(7, 97, "d◆◆", 0);
}

// g'2_2
void mode2_2(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆g'◆◆", 0);
    display_GB2312_string(3, 81, "g'", 0);
    display_GB2312_string(3, 113, letter, 0);
    display_GB2312_string(7, 1, "j◆◆", 0);
    display_GB2312_string(7, 97, "d◆◆", 1);
}
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```
}

// g'3_0
void mode3_0(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆◆◆◆◆◆", 0);
    display_GB2312_string(3, 81, integer, 1);
    display_GB2312_string(3, 89, ".", 0);
    display_GB2312_string(3, 97, decimal, 0);
    display_GB2312_string(3, 105, "Hz", 0);
    display_GB2312_string(7, 1, "j◆◆", 0);
    display_GB2312_string(7, 97, "d◆◆", 0);
}

// g'3_1
void mode3_1(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆◆◆◆◆◆", 0);
    display_GB2312_string(3, 81, integer, 0);
    display_GB2312_string(3, 89, ".", 0);
    display_GB2312_string(3, 97, decimal, 1);
    display_GB2312_string(3, 105, "Hz", 0);
    display_GB2312_string(7, 1, "j◆◆", 0);
    display_GB2312_string(7, 97, "d◆◆", 0);
}

// g'3_2
void mode3_2(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆◆◆◆◆◆", 0);
    display_GB2312_string(3, 81, integer, 0);
    display_GB2312_string(3, 89, ".", 0);
    display_GB2312_string(3, 97, decimal, 0);
    display_GB2312_string(3, 105, "Hz", 0);
    display_GB2312_string(7, 1, "j◆◆", 1);
    display_GB2312_string(7, 97, "d◆◆", 0);
}

// g'3_3
void mode3_3(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆◆◆◆◆◆", 0);
    display_GB2312_string(3, 81, integer, 0);
    display_GB2312_string(3, 89, ".", 0);
    display_GB2312_string(3, 97, decimal, 0);
    display_GB2312_string(3, 105, "Hz", 0);
    display_GB2312_string(7, 1, "j◆◆", 0);
    display_GB2312_string(7, 97, "d◆◆", 1);
}

void mode4(uint8_t* letter, uint8_t* integer, uint8_t* decimal){
    clear_screen();
    display_GB2312_string(3, 1, "◆◆◆◆◆◆◆◆◆◆b◆", 0);
}
```

2. 实验 5 发送-接收联合系统实验

2.1 实验目的

- 练习掌握工程原型试制电路的焊装和调测的基本操作技巧；
- 学习掌握嵌入式系统多种接口技术（模数转换 ADC 接口、PWM 信号发生接口、I2C 总线接口等）的工程应用方法；
- 初步掌握电子测量设计中调试校准的基本技巧；
- 通过设计实验获得设计开发较复杂工程电子系统的初步经验。

2.2 实验主要器材和设备

2.2.1 嵌入式系统基本开发环境

电脑；TM4C1294NCPDT 实验板卡；A2000TM4 扩展板；调频发射实验单元板或调频接收实验单元板。

2.2.2 仪器工具

台式稳压电源；数字示波器；多用电表；函数信号发生器。

2.2.3 主要元件类器材

具体元件类器材清单详见表 2。

表 2 实验 5 所涉及的基本元器件

型号或名称	规格与封装	数量	备注
50KΩ	多圈精密可调电阻	2	可调分压电阻
Z3.3V/0.5W	插脚	1	稳压二极管
LM75B 模块	专用模块	2	IIC 接口温度传感器
TLV2372	DIP	2	运放芯片
IC 插座 8 脚	DIP	2	与 TLV2372 配用
TL431	插脚 TO-92	2	基准电压源
一单元洞洞板	（定制）一单元	2	焊装自制电路
香蕉头、螺母、垫圈	三件套装	8 套	与洞洞板配用
φ0.8 焊锡	φ0.8	1 米 2 根	焊接材料
φ0.5 导线	单芯硬线 φ0.5	1 米 2 根	焊连电路用
单排针	20 芯 2.54mm	2 条	做接线点或检测点，可掰开用
杜邦头跳接线	成品	5	线路连接用

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

色环电阻	0.25W 系列, 插脚	若干	根据设计选用
陶瓷电容	系列, 插脚	若干	根据设计选用

2.3 实验过程

2.3.1 实验任务 5_1——编程控制 FM 模块

在实验任务 5_1 环节, 我们需要利用 UART 串口向接收端发送工作参数设置指示。我们采用的是 UART 6 串口, 利用 AT 指令集中的“AT+FREQ=890”以及“AT+VOL=15”两条指令可以将接收端的工作频率设为 89.0MHz, 音量设为最大值 15。具体代码详见 2.6.1 节。

2.3.2 实验任务 5_2——电压信号发生与测量显示

在实验任务 5_2 环节, 最主要的任务是焊接硬件电路。这部分电路是发射端、接收端共同需要焊接的电路。在基准电压发生电路部分, 我们选用阻值为 $2.4\text{k}\Omega$ 的电阻 R2 和阻值为 $10\text{k}\Omega$ 的 R5。测评时, 测量 P1 点电压, 显示结果为 3.10000V, 精度极高, 满足 $\pm 0.01\text{V}$ 的误差要求。

在可调分压和电压跟随电路部分, 当我们连续调节变阻器 R4 时, 测量 P2、P3 的电压, 观察可得电压值连续可变, 且两点电压基本一致, 即实现了电压跟随功能。

最后, 利用课程提供的例程, 借助开发底板上的数码管显示测试位点电压值。观测可得, 数码管显示的 P3 电压与利用多用电表测出的结果非常接近, 电压示数误差小于 0.01V ; 再用 P2 接相关 ADC 端口, 重复上述步骤, 我们会观察发现当电压值增大时, 误差也随之增大。这体现出了电压跟随电路输入阻抗高、输出阻抗低的特点, 即输出电压不受后级电路阻抗的影响。

2.3.3 实验任务 5_5——信号调理和频率测量

在实验任务 5_5 环节, 我们按照课程指示焊装信号调理和频率测量电路。焊接完成后, 我们使用函数信号发生器, 设“负载阻抗”为高阻状态 (HIZ), 依次输入不同的波形, 使用双通道或多通道示波器, 通道 1 固定观测 P4 点波形, 并设为“触发源”, 同时用其他通道分别观察 TP1、TP2、TP3、P5 点波形。具体波形详见 2.4 节。

2.3.4 实验任务 5_7——接收端系统整合

在实验任务 5_7 环节, 最核心的任务是相关代码的书写。根据课程指示要求, 我们需要以实验四用过的液晶屏和按键作为正式的 UI 按键, 将接收端的各项功能整合成一体化产品。

根据要求, P3 点的电压读数显示于显示屏上, 精度为 0.001V , 能够自动动态刷新; 从 FM 模块

声道接收来自发射端的频率编码信号，按编码规则，解码还原温度数值信息并显示于界面首页，显示精度为 0.1°C ，并能自动动态刷新；通过按键进入第二页画面，设置工作频率，在用户按下“确定”键后自动判断参数合法性，数值合法的区间规定为 $[88.0\text{MHz}, 108.0\text{MHz}]$ ，若数值非法则能够自动弹出警告界面。

借鉴拓展实验四的编程思路，屏幕的显示状态由 `lcd_act` 和 `lcd_subact` 共同确定，`lcd_act` 的值为分别代表电压温度显示、载频设置、超范围提醒；`lcd_subact` 则是根据不同的 `lcd_act` 决定光标的位置。我们对加减、左右和确定键分别编程，在修改数字时，根据 `lcd_subact` 的数值可调整 `freq_temp[]` 对应的数字值，左右键可以修改 `lcd_subact` 的值以锁定数位或者确定键，在按下确定键的时候，检查此时的参数是否合规，若不合规则调到警告界面，并利用 2 秒的计时符号来返回数值修改页面。具体代码详见第 2.6.2 节。

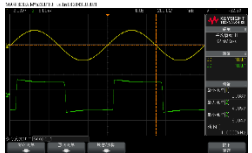
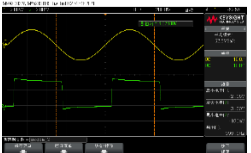
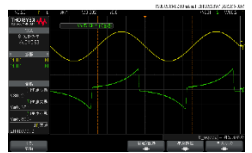
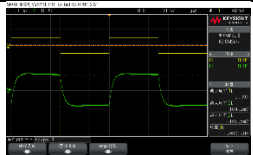
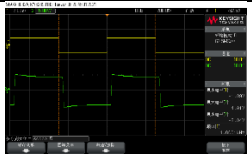
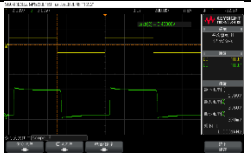
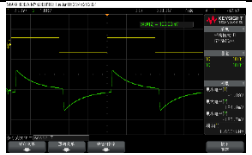
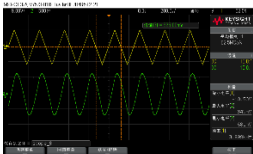
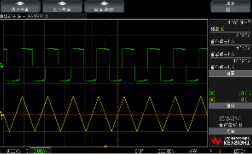
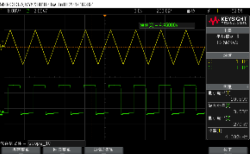
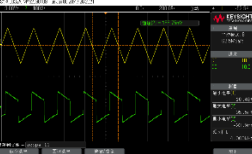
2.3.5 实验任务 5_8——联合实验

在实验任务 5_8 联调联试环节，我们采用相同工作频率（ 99.0MHz ），使两侧同时运作。测评现场显示，发射机能够有效传递温度数值信息并自动刷新；接收机能够实时接收利用右声道传来的指定传感器温度，延时较短，能够实现全部既定功能。

2.4 实验信号波形记录

该部分记录了实验任务 5_4 所要求的 4 个点位、3 种不同输入波形的示波器显示图像。

表 3 实验任务 5_4 所要求观测的波形记录

输入波形	TP1	TP2	TP3	P5
正弦波				
方波				
三角波				

2.5 实验现场考核记录

2021 年 06 月 18 日中午, 发射端、接收端在徐雄老师要求下进行了实验任务 5_8 的现场测评。
接收端的考核指标如图 3 所示, 亦可参考表 4 的记录。



图 3 2021 年 06 月 18 日现场考核记录

表 4 2021 年 06 月 18 日现场考核记录

注: 实验任务 5_8 全面完成则全组免测其余项。

实验任务 5_8	电压测量显示, 由评测官指定分压	基准电压: 3.1V 分压实测: 1.69V 显示: 1.7V
	FM 载频设置, 由评测官指定	指定载频: 99MHz
	FM 载频设置超范围, 是否有提示	√
	温度传送显示	发端: 32.6℃ 收端: 32.7℃

2.6 实验核心代码清单

2.6.1 实验任务 5_1

exp5_1.c

```
/*  
//  
// Copyright: 2020-2021,   
// File name: exp3_0.c  
// Description:  
//   
// 1.   
// 2.   
// 1)   
*/
```

上海交通大学 电子信息与电气工程学院

地 址: 东川路 800 号

邮 编: 200240


```

    UARTCharPut(UART6_BASE, '+');
    UARTCharPut(UART6_BASE, 'V');
    UARTCharPut(UART6_BASE, '0');
    UARTCharPut(UART6_BASE, 'L');
    UARTCharPut(UART6_BASE, '=');
    UARTCharPut(UART6_BASE, '1');
    UARTCharPut(UART6_BASE, '5');
}

//*****
//
//  void GPIOInit(void)
//  GPIO' PortK,PK4,PK5 PortM,PM0
//  PK4, TM1638, STB, PK5, TM1638, DIO, PM0, TM1638, CLK
//
//
//
//
//*****
void GPIOInit(void)
{
    //TM1638T
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK);          // ' K
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOK)) {}; // L. K
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOM);          // ' M
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOM)) {}; // L. M
    // K 4, 5, PK4, PK5 I PK4-STB PK5-DIO
    GPIOPinTypeGPIOOutput(GPIO_PORTK_BASE, GPIO_PIN_4|GPIO_PIN_5);
    // M 0, PM0 I PM0-CLK
    GPIOPinTypeGPIOOutput(GPIO_PORTM_BASE, GPIO_PIN_0);
}

void SysTickInit(void)
{
    SysTickPeriodSet(ui32SysClock/SYSTICK_FREQUENCY); // , 20ms
    SysTickEnable();          // SysTick'
    SysTickIntEnable();       // SysTick*
}

void UARTInit(void)
{
    // SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);          // ??UART0??
    // SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);          // ??? A
    // while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOA));    // ??? A???
    // GPIOPinConfigure(GPIO_PA0_U0RX);          // ??PA0?UART0 RX??
    // GPIOPinConfigure(GPIO_PA1_U0TX);          // ??PA1?UART0 TX??
    // // ??? A??0,1?(PA0,PA1)?UART??
    // GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
    // // ??????
    // UARTConfigSetExpClk(UART0_BASE,
    // ui32SysClock,
    // 115200,          // ??? :115200
    // (UART_CONFIG_WLEN_8 | // ??? :8
    // UART_CONFIG_STOP_ONE | // ??? :1
    // UART_CONFIG_PAR_NONE)); // ??? :?
    //
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART6);          // 'UART6

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);          // 'P
while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOA)); // P

GPIOPinConfigure(GPIO_PA0_U6RX);                      // PD4UART2 RX
GPIOPinConfigure(GPIO_PA1_U6TX);                      // PD5UART2 TX

// P 4,5 PA0,PA1 UART
GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

// '
UARTConfigSetExpClk(UART0_BASE,
                    ui32SysClock,
                    9600,                      // 115200
                    (UART_CONFIG_WLEN_8 |      // 8
                     UART_CONFIG_STOP_ONE |    // 1
                     UART_CONFIG_PAR_NONE)); // Y
UARTFIFOLevelSet(UART0_BASE,UART_FIFO_TX1_8,UART_FIFO_RX7_8);

IntEnable(INT_UART0); // UART0
UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT); // ' UART0 RX,RT

// 'JPC_ "Hello, 2A!"
//UARTStringPut(UART0_BASE, (const char *)"\r\nHello, 2A!\r\n");
}

void DevicesInit(void)
{
    // '25MHz PLL 20MHz
    ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ | SYSCTL_OSC_MAIN |
                                        SYSCTL_USE_PLL | SYSCTL_CFG_VCO_480),
                                        20000000);

    GPIOInit();          // GPIO
    UARTInit();          // UART
    SysTickInit();       // SysTick
    IntMasterEnable();   //

}

void SysTick_Handler(void) // 20ms
{
    if (++clock1000ms >= V_T1000ms)
    {
        clock1000ms_flag = 1; // 0.1 1
        clock1000ms = 0;
    }

    TM1638_RefreshDIGIandLED(digit, pnt, led);
}

void UARTStringPut(uint32_t ui32Base,const char *cMessage)
{
    while(*cMessage != '\0') //  FIFO
        UARTCharPut(ui32Base, *(cMessage++));
}

void UART0_Handler(void)
{

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```

int32_t uart6_int_status;
uint8_t uart_receive_char;
uint8_t i=0;

uart6_int_status = UARTIntStatus(UART6_BASE, true); // 清除中断标志

UARTIntClear(UART6_BASE, uart6_int_status); // 清除中断标志

while(UARTCharsAvail(UART6_BASE)) // 检查FIFO是否非空
{
    uart_receive_char=UARTCharGetNonBlocking(UART6_BASE);
}

}

```

2.6.2 实验任务 5_7 & 5_8

exp5_7.c

```

//*****
//
// Copyright: 2020-2021, 上海交大电子信息与电气工程学院
// File name: exp2_0.c
// Description:
// 1. 通过UART6接口实现LED的流水灯控制，当按下按键时，LED灯按照一定的顺序依次点亮。
// 2. 通过UART6接口实现LED的流水灯控制，当按下按键时，LED灯按照一定的顺序依次点亮。
// 3. 通过UART6接口实现LED的流水灯控制，当按下按键时，LED灯按照一定的顺序依次点亮。
// 作者： 上海交大电子信息与电气工程学院
// Version: 1.0.0.20201228
// Date: 2020-12-28
// History:
//
//*****

//*****
//
// 文件说明
//
//*****

#include <stdint.h>
#include <stdbool.h>
#include "define.h" // 定义NCPDT
#include "inc/hw_memmap.h" // 定义内存映射
#include "inc/hw_ints.h"
#include "inc/hw_types.h" // 定义寄存器类型
#include "driverlib/debug.h" // 调试函数
#include "driverlib/gpio.h" // GPIO库
#include "driverlib/pin_map.h" // TM4C661 MCU引脚映射
#include "driverlib/sysctl.h" // 系统控制
#include "driverlib/systick.h" // SysTick Driver
#include "driverlib/interrupt.h" // NVIC Interrupt Controller Driver
#include "driverlib/uart.h" // UART库
#include "driverlib/timer.h" // Timer库
#include "driverlib/pwm.h" // Timer库
#include "tm1638.h" // TM1638 7-segment display

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240


```
//LCD
uint8_t lcd_act = 0;
uint8_t lcd_subact = 0; //

bool if_refresh = true; //

//

uint32_t test_counter = 0;

// 8
// 8λ
// 8λ
uint8_t digit[8]={ ' ', ' ', ' ', ' ', 'H', ' ', ' ', ' ', ' '};

// 8λ
uint8_t pnt = 0x00;

// 8
// 8
// 8
uint8_t led[] = {1, 1, 1, 1, 1, 1, 1, 0};

// 8
char str[50] = {'\0'};

// 8j
uint8_t key_code = 0;

// 8Z
uint32_t ui32SysClock;

//
uint8_t* letter[3]={"A", "B", "C"};
uint8_t* number[10] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};
uint16_t voltage = 0;
uint8_t* v_passin[4] = {0};
uint16_t tempre = 0;
uint8_t* t_passin[3] = {0};
uint8_t freq[4] = {1, 0, 0, 0};
uint8_t freq_temp[4] = {0}; //
uint8_t* f_passin[4] = {0};

// 8
void (*func[3][5])(uint8_t* [], uint8_t* [], uint8_t* []) = {0};

// AIN2(PE1) ADC[0-4095]
uint32_t ui32ADC0Value;

// AIN2(λ0.01V) [0.00-3.30]
uint32_t ui32ADC0Voltage;

// 1s
volatile uint8_t g_ui8INTStatus = 0;

// 8hTIMER0
volatile uint32_t g_ui32TPreCount = 0;

// 8fTIMER0
volatile uint32_t g_ui32TCurCount = 0;

//*****
//
```

```
// *****
//
//*****
int main(void)
{
    uint8_t temp,i;
    uint32_t ui32Freq; // 0x00000000k0000000

    DevicesInit();          // MCU00000000'00

    while (clock100ms < 3); // 0x00000000>60ms,0x00000000TM1638000000

    TM1638_Init();          // 0x00'0000TM1638
    initial_lcd();

    clear_screen();

    //*****
    func[0][0] = mode0_0;
    func[1][0] = mode1_0;
    func[1][1] = mode1_1;
    func[1][2] = mode1_2;
    func[1][3] = mode1_3;
    func[1][4] = mode1_4;
    func[2][0] = mode2_0;

    // 0x0'00000000i000
    t_passin[0] = number[tempre / 100];
    t_passin[1] = number[(tempre / 10) % 10];
    t_passin[2] = number[tempre % 10];

    // 0x0'00000000i100.0MHz
    passFreq(freq);

    while (1)
    {
        // *****
        if (clock400ms_flag == 1)          // 0x0000400ms00000000
        {
            clock400ms_flag = 0;

            ui32ADC0Value = ADC_Sample(); // *****

            //          digit[4] = ui32ADC0Value / 1000;          // 0x0'0000000000000000
            //          digit[5] = ui32ADC0Value / 100 % 10;      // 0x0'0000000000000000
            //          digit[6] = ui32ADC0Value / 10 % 10;       // 0x0'0000000000000000
            //          digit[7] = ui32ADC0Value % 10;            // 0x0'0000000000000000

            ui32ADC0Voltage = ui32ADC0Value * 3300 / 4095;

            //          digit[0] = (ui32ADC0Voltage / 1000) % 10;
            //          digit[1] = (ui32ADC0Voltage / 100) % 10; // 0x0'0000000000000000
            //          digit[2] = (ui32ADC0Voltage / 10) % 10;  // 0x0'0000000000000000
            //          digit[3] = ui32ADC0Voltage % 10;          // 0x0'0000000000000000

            v_passin[0] = number[(ui32ADC0Voltage / 1000) % 10]; // 0x0'0000000000000000
            v_passin[1] = number[(ui32ADC0Voltage / 100) % 10]; // 0x0'0000000000000000
            v_passin[2] = number[(ui32ADC0Voltage / 10) % 10];  // 0x0'0000000000000000
            v_passin[3] = number[(ui32ADC0Voltage) % 10];        // 0x0'0000000000000000
            if (lcd_act == 0 && lcd_subact == 0) if_refresh = true;
        }
    }
}
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240


```

        if_refresh = false;

        for (i=0; i<4; ++i) f_passin[i] = number[freq_temp[i]]; // 0x00000000'Z
        TEST_H;

        func[lcd_act][lcd_subact](v_passin, t_passin, f_passin); // 0x00000000'g'00000000

        if (lcd_act == 2 && lcd_subact == 0) {if_count_2s = true; clock2s = 0; clock2s_flag = false;} // 0x00000000'

        else if_count_2s = false;
        TEST_L;
    }

}

}

//*****
//
// 0x00000000void GPIOInit(void)
// 0x00000000GPIO'0x00000000PortK0x00000000PK4,PK5I'0x00000000PortM0x00000000PM0I
// 0x00000000PK40x00000000TM16380x00000000STB0x00000000PK50x00000000TM16380x00000000DIO0x00000000PM00x00000000TM16380x00000000CLK
// 0x000000000x00000000
// 0x000000000x00000000
//
//*****
void GPIOInit(void)
{
    //0x00000000TM1638oT'
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK); // '0x00000000 K
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOK)){}; // 0x00000000. K0x00000000

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOM); // '0x00000000 M
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOM)){}; // 0x00000000. M0x00000000

    // 0x00000000 K0x000000004,5λ0x00000000PK4,PK50x00000000I'0x00000000 PK4-STB PK5-DIO
    GPIOPinTypeGPIOOutput(GPIO_PORTK_BASE, GPIO_PIN_4|GPIO_PIN_5);
    // 0x00000000 M0x000000000λ0x00000000PM00x00000000I'0x00000000 PM0-CLK
    GPIOPinTypeGPIOOutput(GPIO_PORTM_BASE, GPIO_PIN_0);
}

//*****
//
// 0x00000000SysTickInit(void)
// 0x00000000SysTick*
// 0x000000000x00000000
// 0x000000000x00000000
//
//*****
void SysTickInit(void)
{
    SysTickPeriodSet(ui32SysClock/SYSTICK_FREQUENCY); // 0x000000000x00000000,0x0000000020ms
    SysTickEnable(); // SysTick'
    SysTickIntEnable(); // SysTick*
}

//*****
//
// 0x00000000void DevicesInit(void)
// 0x00000000CU0x00000000'0x00000000T'0x00000000ä0x00000000GPIO'0x00000000SysTick*

```

```
// *****
// *****
//
//*****
void DevicesInit(void)
{
    // '*****#25MHz*****PLL*****zI20MHz
    ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |SYSCTL_OSC_MAIN |
        SYSCTL_USE_PLL |SYSCTL_CFG_VCO_480),
        20000000);

    GPIOInit();          // GPIO*****
    ADCInit();           // ADC*****
    Timer0Init();        // Timer0*****
    Timer1Init();
    UARTInit();          // UART*****
    SysTickInit();       // *****SysTick*****
    IntMasterEnable();   // *****

    IntPrioritySet(INT_TIMER1A,0x00);    // *****INT_TIMER0A*****
    IntPrioritySet(INT_TIMER0A,0x01);    // *****INT_TIMER0A*****
    IntPrioritySet(FAULT_SYSTICK,0xe0);  // *****SYSTICK*****INT_TIMER0A*****

}

//*****
//
// *****void SysTick_Handler(void)
// *****SysTick*****
// *****
// *****
//
//*****
void SysTick_Handler(void)    // *****I20ms
{
    uint8_t key_code_pre = key_code;    // *****jü"
    uint8_t i = 0;

    // 0.1*****
    if (++clock100ms >= V_T100ms)
    {
        clock100ms_flag = 1; // 0.1*****
        clock100ms = 0;
    }

    // 0.5*****
    if (++clock500ms >= V_T500ms)
    {
        clock500ms_flag = 1; // 0.5*****
        clock500ms = 0;
    }

    // 1.5*****
    if (++clock1500ms >= V_T1500ms)
    {
        clock1500ms_flag = 1; // 1.5*****
        clock1500ms = 0;
    }

    // 2*****
    if (++clock2s >= V_T2s)
```

```

{
    clock2s_flag = 1; // 2秒计时开始
    clock2s = 0;
}

// 5秒计时开始
if (++clock5s >= V_T5s){
    clock5s_flag = 1;
    clock5s = 0;
}

// 10秒计时开始
if (++clock10s >= V_T10s){
    clock10s_flag = 1;
    clock10s = 0;
}

if (++clock400ms >= V_T400ms)
{
    clock400ms_flag = 1; // 400ms计时开始
    clock400ms = 0;
}

// *0.5秒LED闪烁
TM1638_RefreshDIGandLED(digit, pnt, led);

// 按键扫描
key_code = TM1638_Readkeyboard();

// 按键处理
if (key_code_pre == 0 && key_code != 0 && lcd_act == 0 && lcd_subact == 0){
    lcd_act = 1;
    lcd_subact = 0;
    for (i = 0; i < 4; ++i){
        freq_temp[i] = freq[i];
    } // 初始化频率数组
    if_refresh = true;
    return;
}

if (key_code_pre != 8 && key_code == 8){
    if (lcd_act == 1 && lcd_subact == 0){
        freq_temp[0] = (freq_temp[0] + 9) % 10;
        if_refresh = true;
    } else if (lcd_act == 1 && lcd_subact == 1){
        freq_temp[1] = (freq_temp[1] + 9) % 10;
        if_refresh = true;
    } else if (lcd_act == 1 && lcd_subact == 2){
        freq_temp[2] = (freq_temp[2] + 9) % 10;
        if_refresh = true;
    } else if (lcd_act == 1 && lcd_subact == 3){
        freq_temp[3] = (freq_temp[3] + 9) % 10;
        if_refresh = true;
    }
}

if (key_code_pre != 2 && key_code == 2){
    if (lcd_act == 1 && lcd_subact == 0){
        freq_temp[0] = (freq_temp[0] + 1) % 10;
        if_refresh = true;
    } else if (lcd_act == 1 && lcd_subact == 1){
        freq_temp[1] = (freq_temp[1] + 1) % 10;
        if_refresh = true;
    } else if (lcd_act == 1 && lcd_subact == 2){
        freq_temp[2] = (freq_temp[2] + 1) % 10;
        if_refresh = true;
    } else if (lcd_act == 1 && lcd_subact == 3){
        freq_temp[3] = (freq_temp[3] + 1) % 10;
        if_refresh = true;
    }
}

```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240


```

GPIOPinConfigure(GPIO_PP1_U6TX); // 配置PP1UART6 TX引脚

// 配置PA0, PA1为UART引脚
GPIOPinTypeUART(GPIO_PORTP_BASE, GPIO_PIN_0 | GPIO_PIN_1);

// 配置UART6
UARTConfigSetExpClk(UART6_BASE,
                    ui32SysClock,
                    9600, // 波特率9600
                    (UART_CONFIG_WLEN_8 | // 8位数据长度
                     UART_CONFIG_STOP_ONE | // 1位停止位
                     UART_CONFIG_PAR_NONE)); // 无校验位

IntEnable(INT_UART6); // 使能UART6中断
UARTIntEnable(UART6_BASE, UART_INT_RX | UART_INT_RT); // 使能UART6接收和接收完成中断

// 配置GPIO
SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0); // 使能UART0
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA); // 使能GPIOA
while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOA)); // 等待GPIOA就绪

GPIOPinConfigure(GPIO_PA0_U0RX); // 配置PA0为UART0 RX引脚
GPIOPinConfigure(GPIO_PA1_U0TX); // 配置PA1为UART0 TX引脚

// 配置UART0
UARTConfigSetExpClk(UART0_BASE,
                    ui32SysClock,
                    115200, // 波特率115200
                    (UART_CONFIG_WLEN_8 | // 8位数据长度
                     UART_CONFIG_STOP_ONE | // 1位停止位
                     UART_CONFIG_PAR_NONE)); // 无校验位

// 发送字符串
UARTStringPut(UART0_BASE, (const char *)"Hello, 2A!");

IntEnable(INT_UART0); // 使能UART0中断
UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT); // 使能UART0接收和接收完成中断
}

void UART6_Handler(void)
{
    uint32_t uart6_int_status;

    uart6_int_status = UARTIntStatus(UART6_BASE, true); // 获取UART6中断状态
    UARTIntClear(UART6_BASE, uart6_int_status); // 清除UART6中断状态

    UARTStringGet(UART6_BASE, str);
    UARTStringPut(UART0_BASE, str);
}

// *****
//
// 配置UART0
// 配置UART0

```

```
//          cMessage
//
//
//*****
void UARTStringPut(uint32_t ui32Base,const char *cMessage)
{
    while(*cMessage != '\0') {
        UARTCharPut(ui32Base, *(cMessage++));
        SysCtlDelay(1 * ( ui32SysClock / 3000)); //1ms
    }
    //  UARTCharPut(ui32Base, '\r');
    //  UARTCharPut(ui32Base, '\n');
}

//*****
//
//  void UARTStringGet(uint32_t ui32Base,const char *cMessage)
//  UARTG
//  ui32BaseUARTG
//          str
//
//
//*****
void UARTStringGet(uint32_t ui32Base, char* s) {
    //  char* temp = s;
    while (UARTCharsAvail(ui32Base)) //  FIF0
    {
        *s = (char) UARTCharGetNonBlocking(ui32Base); //
        ++s;
        SysCtlDelay(1 * ( ui32SysClock / 3000)); //1ms
    }
    *s = '\0';
    //  UARTStringPut(UART0_BASE, (const char *) "Get String:");
    //  UARTStringPut(UART0_BASE, temp);
}

void passFreq(uint8_t* f){
    UARTStringPut(UART6_BASE, "AT+FREQ=");
    if (f[0] == 0){
        UARTCharPut(UART6_BASE,f[1]+'0');
        UARTCharPut(UART6_BASE,f[2]+'0');
        UARTCharPut(UART6_BASE,f[3]+'0');
    }
    else {
        UARTCharPut(UART6_BASE,f[0]+'0');
        UARTCharPut(UART6_BASE,f[1]+'0');
        UARTCharPut(UART6_BASE,f[2]+'0');
        UARTCharPut(UART6_BASE,f[3]+'0');
    }
}

// Z
//*****
//
//  void TIMER1A_Handler(void)
//  Timer1AxB
//
//
//
//
//*****
void TIMER1A_Handler(void)
{
    TimerIntClear(TIMER1_BASE, TIMER_TIMA_TIMEOUT); //
}
```

```

g_ui32TPreCount = g_ui32TCurCount; // 0x00000000 TIMER0 0x00000000.

g_ui32TCurCount = TimerValueGet(TIMER0_BASE, TIMER_A); // 0x00000000 TIMER0 0x00000000.

TimerDisable(TIMER0_BASE, TIMER_A); // TIMER0 0x00000000.

g_ui8INTStatus = 1; // 1s 0x00000000
}

//*****
//
// 0x00000000 void PWMInit(uint32_t ui32Freq_Hz)
// 0x00000000 0x00000000 Z 0x00000000 I ui32Freq_Hz 0x00000000 k 0x00000000 (n 0x00000000 I 50% 0x00000000 PWM) 0x00000000 0x00000000 I M0 PWM4 (PG0)
// 0x00000000 0x00000000 I 0x00000000 0x00000000 0x00000000 z 0x00000000 0x00000000 h 0x00000000 0x00000000 z 0x00000000 0x00000000 d 0x00000000 g
// 0x00000000 0x00000000 ui32Freq_Hz 0x00000000 C 0x00000000 0x00000000 k 0x00000000 0x00000000 Z
// 0x00000000 0x00000000
//
//*****
void PWMInit(uint32_t ui32Freq_Hz)
{
    SysCtlPWMClockSet(SYSCTL_PWMDIV_1); // 0x00000000 Z
    SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0); // PWM0' 0x00000000
    PWMOutputState(PWM0_BASE, PWM_OUT_4_BIT, true); // ' 0x00000000 (0x00000000) PWM0_4 0x00000000
    PWMGenEnable(PWM0_BASE, PWM_GEN_2); // ' 0x00000000 PWM0 g 0x00000000 2 0x00000000 y 0x00000000 (0x00000000 I 4 0x00000000 PWM 0x00000000 2 0x00000000 y 0x00000000 0x00000000)
    PWMGenPeriodSet(PWM0_BASE, PWM_GEN_2, ui32SysClock / ui32Freq_Hz); // 0x00000000 Freq_Hz 0x00000000 PWM 0x00000000

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOG); // ' 0x00000000 GPIOG
    GPIOPinConfigure(GPIO_PG0_M0PWM4); // 0x00000000 Y 0x00000000
    GPIOPinTypePWM(GPIO_PORTG_BASE, GPIO_PIN_0); // 0x00000000 y 0x00000000

    PWMGenConfigure(PWM0_BASE, PWM_GEN_2, PWM_GEN_MODE_DOWN | PWM_GEN_MODE_NO_SYNC); // 0x00000000 PWM 0x00000000
    PWMPulseWidthSet(PWM0_BASE, PWM_OUT_4, (PWMGenPeriodGet(PWM0_BASE, PWM_GEN_2) / 2)); // 0x00000000 n 0x00000000 I 50%
}

//*****
//
// 0x00000000 void Timer1Init(void)
// 0x00000000 0x00000000 Timer1 I h 0x00000000 2 0x00000000 h 0x00000000 I 1s
// 0x00000000 0x00000000
// 0x00000000 0x00000000
//
//*****
void Timer1Init(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER1); // TIMER1 ' 0x00000000

    TimerConfigure(TIMER1_BASE, TIMER_CFG_ONE_SHOT); // 0x00000000 I 32 λ 1 0x00000000 h 0x00000000
    TimerLoadSet(TIMER1_BASE, TIMER_A, ui32SysClock); // TIMER1 A 1 0x00000000 ns
}

//*****
//
// 0x00000000 void Timer0Init(void)
// 0x00000000 0x00000000 Timer0 I 0x00000000 i 0x00000000 i 0x00000000 g ' 0x00000000 T0 CCP0 (PL4) I 0x00000000
// 0x00000000 0x00000000
// 0x00000000 0x00000000
//
//*****

```



```
void Timer0Init(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0); // '♦♦TIMER0

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOL); // '♦♦GPIOL

    GPIOPinConfigure(GPIO_PL4_T0CCP0); // ♦♦♦♦♦♦♦♦♦♦
    GPIOPinTypeTimer(GPIO_PORTL_BASE, GPIO_PIN_4); // ♦♦♦♦♦♦♦♦
    GPIOPadConfigSet(GPIO_PORTL_BASE, GPIO_PIN_4, GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD_WPU); // ♦♦♦♦♦♦♦♦♦♦♦♦

    TimerConfigure(TIMER0_BASE, TIMER_CFG_SPLIT_PAIR | TIMER_CFG_A_CAP_COUNT_UP); // ♦♦♦♦♦♦♦♦♦♦♦♦♦♦
    //TimerPrescaleSet(TIMER0_BASE, TIMER_A, 255); // ♦♦♦♦♦♦♦♦♦♦
    TimerControlEvent(TIMER0_BASE, TIMER_A, TIMER_EVENT_POS_EDGE); // ♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦
}
```

modes.c

```
#include "JLX12864_2.c"

// g'0_0
void mode0_0(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    // clear_screen();
    display_GB2312_string(1, 1, "♦♦0:", 0);
    display_GB2312_string(1, 41, v_passin[0], 0);
    display_GB2312_string(1, 49, ".", 0);
    display_GB2312_string(1, 57, v_passin[1], 0);
    display_GB2312_string(1, 65, v_passin[2], 0);
    display_GB2312_string(1, 73, v_passin[3], 0);
    display_GB2312_string(1, 81, "V", 0);
    display_GB2312_string(3, 1, "♦♦1:", 0);
    display_GB2312_string(3, 81, t_passin[0], 0);
    display_GB2312_string(3, 89, t_passin[1], 0);
    display_GB2312_string(3, 97, ".", 0);
    display_GB2312_string(3, 105, t_passin[2], 0);
    display_GB2312_string(3, 113, "♦♦", 0);
    display_GB2312_string(3, 121, "C", 0);
}

// g'1_0
void mode1_0(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    clear_screen();
    display_GB2312_string(1, 1, "♦♦Z:", 0);
    display_GB2312_string(1, 41, f_passin[0], 1);
    display_GB2312_string(1, 49, f_passin[1], 0);
    display_GB2312_string(1, 57, f_passin[2], 0);
    display_GB2312_string(1, 65, ".", 0);
    display_GB2312_string(1, 73, f_passin[3], 0);
    display_GB2312_string(1, 81, "M", 0);
    display_GB2312_string(1, 89, "H", 0);
    display_GB2312_string(1, 97, "z", 0);
    display_GB2312_string(7, 97, "j♦♦", 0);
}

// g'1_1
void mode1_1(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    clear_screen();
    display_GB2312_string(1, 1, "♦♦Z:", 0);
    display_GB2312_string(1, 41, f_passin[0], 0);
    display_GB2312_string(1, 49, f_passin[1], 1);
    display_GB2312_string(1, 57, f_passin[2], 0);
}
```

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

```
display_GB2312_string(1, 65, ".", 0);
display_GB2312_string(1, 73, f_passin[3], 0);
display_GB2312_string(1, 81, "M", 0);
display_GB2312_string(1, 89, "H", 0);
display_GB2312_string(1, 97, "z", 0);
display_GB2312_string(7, 97, "j", 0);
}

// g'1_2
void mode1_2(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    clear_screen();
    display_GB2312_string(1, 1, "Z:", 0);
    display_GB2312_string(1, 41, f_passin[0], 0);
    display_GB2312_string(1, 49, f_passin[1], 0);
    display_GB2312_string(1, 57, f_passin[2], 1);
    display_GB2312_string(1, 65, ".", 0);
    display_GB2312_string(1, 73, f_passin[3], 0);
    display_GB2312_string(1, 81, "M", 0);
    display_GB2312_string(1, 89, "H", 0);
    display_GB2312_string(1, 97, "z", 0);
    display_GB2312_string(7, 97, "j", 0);
}

// g'1_3
void mode1_3(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    clear_screen();
    display_GB2312_string(1, 1, "Z:", 0);
    display_GB2312_string(1, 41, f_passin[0], 0);
    display_GB2312_string(1, 49, f_passin[1], 0);
    display_GB2312_string(1, 57, f_passin[2], 0);
    display_GB2312_string(1, 65, ".", 0);
    display_GB2312_string(1, 73, f_passin[3], 1);
    display_GB2312_string(1, 81, "M", 0);
    display_GB2312_string(1, 89, "H", 0);
    display_GB2312_string(1, 97, "z", 0);
    display_GB2312_string(7, 97, "j", 0);
}

// g'1_4
void mode1_4(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    clear_screen();
    display_GB2312_string(1, 1, "Z:", 0);
    display_GB2312_string(1, 41, f_passin[0], 0);
    display_GB2312_string(1, 49, f_passin[1], 0);
    display_GB2312_string(1, 57, f_passin[2], 0);
    display_GB2312_string(1, 65, ".", 0);
    display_GB2312_string(1, 73, f_passin[3], 0);
    display_GB2312_string(1, 81, "M", 0);
    display_GB2312_string(1, 89, "H", 0);
    display_GB2312_string(1, 97, "z", 0);
    display_GB2312_string(7, 97, "j", 1);
}

// g'2_0
void mode2_0(uint8_t* v_passin[], uint8_t* t_passin[], uint8_t* f_passin[]){
    clear_screen();
    display_GB2312_string(1, 1, "ZXXXXXXXX", 0);
}
```

3. 学习心得和意见建议

《工程实践与科技创新 II-A》拓展实验，“物”如其名，难度大、任务重、时间紧。相较于基础实验而言，实验四和实验五的工程要求全方位提升了一个档次，对学生综合素质与实践能力提出了更为严苛的要求。经过了近一个月的奋战，我们见贤思齐、查缺补漏，总体而言，组员的综合素质与实践能力得到了一定程度的提升，团队整体水准有了不小的进步。

3.1 实验四学习心得

实验四是《工程实践与科技创新II-A》拓展部分的开山之作，是实验五的基础。实验四整体来讲难度略低，但对于我们拓宽视野、初涉团队协作有着一定的启迪、引领作用。

在实验四中，最让人眼前一亮的就是有限状态机(FSM, Finite State Machine)逻辑的学习与使用。在之前的基础实验部分，我们已经对该程序设计思想与方法有了一定的了解，但对该思想的真正落地转化与应用，还是在实验四的代码编写环节。笔者认为，FSM的精髓在于简洁易读、通用性强，在未来的科研、工作中，该思路有着一定的借鉴意义。这就类似于好的代码写作风格，可以有效提升团队协作效率，这点在团队结构化、模块化协作时更能凸显其意义。

3.2 实验五学习心得

实验五较之前实验的最大区别在于以下两点：电路板的焊接与组员间的协作。笔者将从这两个角度来重点谈谈我们的感悟。

3.2.1 电路板的焊接

其中，电路板的焊接对于大多数电院同学来说可能并不陌生，在上一第二学期的工科平台课程《工程实践》及大二第一学期的电院平台课程《工程实践与科技创新 I》中，我们已经尝试使用过电烙铁焊接收音机、无线话筒与多用电表了。不过与之前相比，本次实验对电路板的焊接要求更高，一方面需要满足电路板本身相关参数的精度要求；另一方面，电路板的功效与本实验其他电路模块息息相关，决定整个实验任务的成败，在实验五中起着先导性与突破性的重要作用。

在电路板的第一次焊接过程中，我们的拖锡操作极不熟练，看着鼓鼓囊囊的锡包，我们不禁怀疑存在着大量虚焊情况。果不其然，在自主测试时，整型电路部分根本没有任何响应。鉴于电路调试非常浪费时间，我们当机立断从头焊起。第二次焊接的过程虽然枯燥且乏味，但是我们熟练了很

多，最终很顺利地满足任务要求。尤其是 P1 处电压值的测量，最终结果显示为 3.10000V，精确度出乎意料，让我们体验到了成功后的喜悦与获得感。

整体而言，虽然我们接受组的焊板子过程不算完满，但是我们认为，成长往往比成功更重要。这次焊板子给我们留下了太多难以磨灭的印象，感受到了一些前所未有的情感体验。其中，最让我们深受触动的就是耐心与抗压的重要性。经历了比他人更长时间的鏖战，感受了从头再来的绝望，眼睁睁地看到其他组别同学成功之后的喜悦，我们依旧只能静下心来一个个地仔细焊。在这个过程中，我们接受组的压力是巨大的，不仅有来自自己对未知的恐慌，更有与他人相互竞争下的挫败感。但我们没有退缩，想要解决困难只能迎难而上，最终的结果我们也就能坦然接受了。

虽然焊板子的过程很痛苦，但这次动手体验依旧让我们再次领略了电子元器件及电子系统的魅力，对电子类产生了兴趣。其实，我们在基电、数电、模电，嵌入式课堂上所学到的知识大多都是太过理想化、理论化的演绎，若想要真切感受电路的魅力，还是要亲自到实验室来动动手。

3.2.2 组员间的协作

实验五作为本次拓展实验的集大成者与《工程实践与科技创新II-A》课程的整体凝练，将“引导学生在解决工程实际问题中提升测量技巧、工程设计能力和高效率团队协作能力”的课程整体目标体现的淋漓尽致，充分发挥了立德树人、为国育才的导向作用，是电子工程系“厚基础、重实践、严要求”优良传统的发扬光大与凝练升华。

其中，“团队协作能力”是本实验的核心。作为25组的组长，我一方面需要一心一意、保质保量地完成分内任务，起到率先垂范的示范引领作用；另一方面还需要协调组间、组内同学任务，确保按时精确地完成工程整体目标，万无一失地准备最终考核的到来。

在实验五的任务实施与自主评估阶段，25组四位同学目标明确、积极向上，为早日完成实验五全部既定任务而努力奋斗。秦同学本学期选修了全部要求的交叉课程，课业压力繁重，但仍然抽出时间一下午完成了发射组的硬件焊接任务；朱同学选修了安泰工商管理辅修专业，晚课众多，依旧完成了实验五发射组所有代码的书写与修改；张同学身为校航模队主力成员与学生创新中心助管，依然抽出时间来重新焊接接收机的洞洞板；我也身兼数职，实现了接收机的硬件、软件的各项性能……凡此种种，我就不再一一列举。

正是在组员们的通力协作下，我们一步一个脚印，一步步实现了工程全部既定目标。在06月06日，发射组率先完成全部工程任务；随后，经历了重新焊板子的曲折，接收组在06月09日完成了全部工程任务；06月11日，发射组-接受组初样联调联试试验取得圆满成功；06月18日，全组顺利通过了徐雄老师的考核，评测各项指标正常，参评器件性能良好，为《工程实践与科技创新II-A》课程画上了圆满的句号。

3.3 《工程实践与科技创新 II-A》拓展实验部分的意见建议

- i. 理论学时安排与实践环节安排相脱节，建议本课程改为一周一课模式一以贯之；
- ii. 课程教学学时与学生接受能力有冲突，建议本课程从头至尾由浅入深循序渐进；
- iii. 组内贡献组长评价维度标准过于主观，建议课程组给出明确细致的量化测评表；
- iv. 课程科技感新鲜感生活感有些许欠缺，建议根据专业前沿与特色加入时代气息；
- v. 工科创系列课程前后体系衔接有漏洞，建议全面革新科创 1 的教学内容与形式。

4. 课程代码开源链接

2020-2021-2 EI227 《工程实践与科技创新 II-A》接收组所有代码均已开源，详情请见 25 组组长戴天杰的 GitHub 网址：[Elfenreigen/SJTU-EI227: EI227 Engineering Practice and Technological Innovation II-A For EE in SJTU \(github.com\)](https://github.com/Elfenreigen/SJTU-EI227: EI227 Engineering Practice and Technological Innovation II-A For EE in SJTU)

Semester 2020-2021-2: All codes of EI227 Engineering Practice and Technological Innovation II-A Receiving Group have been open source. For details, please refer to GitHub website of Group Leader Tianjie Dai's GitHub account: [Elfenreigen/SJTU-EI227: EI227 Engineering Practice and Technological Innovation II-A For EE in SJTU \(github.com\)](https://github.com/Elfenreigen/SJTU-EI227: EI227 Engineering Practice and Technological Innovation II-A For EE in SJTU)

5. 致谢

- 本次实验，由于以下单位及个人的帮助而顺利进行，在此特表感谢！
- 感谢上海交通大学电子信息与电气工程学院电子工程系实验中心提供的设备、场地；
- 感谢指导老师李安琪老师、徐雄老师及各位助教辛勤授课与指导答疑。

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240