

目录

- 1. 概述.....1
  - 1.1 题目名:简易电子琴.....1
  - 1.2 功能和技术指标要求.....1
  - 1.3 国内外相关情况概述.....1
- 2. 技术方案.....2
  - 2.1 基本原理.....2
  - 2.2 总体技术方案.....2
- 3. 硬件连接.....3
- 4. 软件设计.....3
  - 4.1 软件的功能要求.....3
  - 4.2 软件设计总体思路.....3
- 5. 性能测试.....5
  - 5.1 测试方法.....5
  - 5.2 测试结果.....5
- 6. 实习总结.....6
  - 6.1 收获.....6
- 7、参考文献.....7
- 附录：源程序.....8

# 1. 概述

## 1.1 题目名:简易电子琴

## 1.2 功能和技术指标要求

在屏幕上设计一个简易电子琴的琴键，通过点击触摸屏，发出对应的音调。

## 1.3 国内外相关情况概述

(1)电子琴是现代电子科技与音乐结合的产物，它在现代音乐扮演着重要角色。二十世纪八十年代，电子琴进入我国，得到了迅速发展，与它比较经济、有着丰富的表现力、功能比较强大的广泛应用特点分不开。在中国，1958年北京邮电学院研制了一台电子管单音电子琴。

(2)电子琴，就可以轻松演绎不同时期、地域和风格的音乐。如轻音乐，这是一种介于古典音乐和流行音乐之间的通俗音乐形式，其一般并没有深刻的内涵，而是重在给人们以轻松和美好的感觉。所以具有旋律流畅、结构简单、节奏。

(3)当前电子玩具在市场上供不应求，电子琴玩具就是一个非常好商机。电子琴经过上百年的发展，其设计生产技术已经趋于成熟。近几年儿童玩具的微型电子琴研究具有广阔的市场前景。系统运行过程中可以演奏不同乐曲以及模拟各种大自然音响，如生日快乐歌、两只老虎、雨滴声、叮咚门铃声等，取得了较好的控制效果。

(4)从1907年美国卡西尔发明第一台用电磁线圈产生音阶的电子琴算起，电子琴在国外的历史已有上百年的历史；如果从1959年日本YAMAHA公司成功地研制出世界上第一台晶体管双排键电子琴算起，(D—1电子风琴)那么其历史也有近50年。而当20世纪70年代，国外已有几百所大学把包括双排键电子琴在内的电子乐器教学列入正式课程的时候，电子琴在中国还鲜为人知。

## 2. 技术方案

### 2.1 基本原理

本次设计的电子琴基于 UCOSII 操作系统下实现任务调度，主控选择为 STM32F103RCT6，采用的是正点原子的 Mini 板，通过触摸屏上的按键，然后检测按键的输入，判断按键所在的位置，按键然后通过任务间通信来实现蜂鸣器任务，进而蜂鸣器就会发出低音或者高音的 Do、Re、Mi、Fa、SO、La、Si。

### 2.2 总体技术方案

本次设计的电子琴基于 UCOS 系统下，主要使用其任务调度和任务间的通信，每个任务相对独立，提高了处理的实时性，通过建立消息邮箱来实现任务间的通信。总体设计流程图如下：

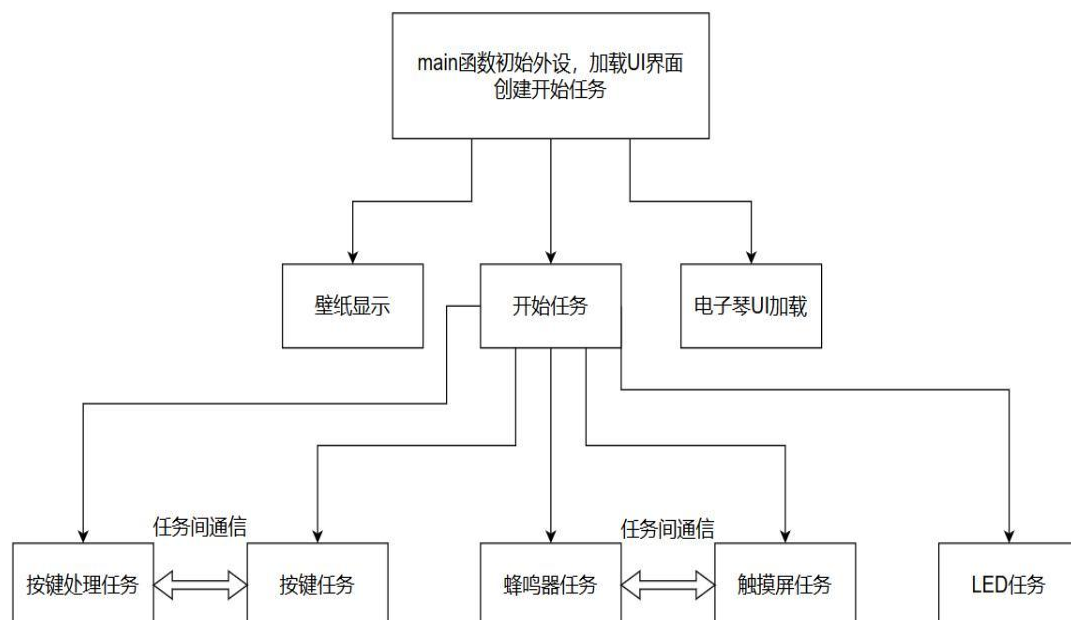


图 2.2 总体技术方案图

## 3. 硬件连接

本实验采用的蜂鸣器是无源蜂鸣器，可以发出“多来米发索拉西”的效果。它有三个引脚，VCC, I/O, GND。该蜂鸣器模块采用 9012PNP 三极管驱动，工作电压 3.3-5V。VCC 接到开发板的 5V, GND 接到开发板的 GND, I/O 接到单片机管脚的 PA8，用来输出音频。

## 4. 软件设计

### 4.1 软件的功能要求

- (1) 采用 UCOSII 操作系统实现任务调度以及任务间的通信。
- (2) 实现个性化电子琴壁纸的显示。
- (3) 在 LCD 上实现电子琴 UI 界面的显示。
- (4) 触摸屏的触摸检测，触摸位置的检测以及触摸位置的填充来显示触摸效果。
- (5) 对 PWM 的配置来实现蜂鸣器的震动频率可变。

### 4.2 软件设计总体思路

#### (1) UCOSII 的任务创建

最先肯定是要调用 CPU\_Init() 初始化 UCOSIII。创建任务，一般我们在 main() 函数中只创建一个 start\_task 任务，其他任务都在 start\_task 任务中创建。在调用 OSTaskCreate() 函数创建任务的时候一定要调用 OS\_CRITICAL\_ENTER() 函数进入临界区，任务创建完以后调用 OS\_CRITICAL\_EXIT() 函数退出临界区。

#### (2) UCOSII 的任务间通信

消息的传递，也称之为任务间通信，在 UCOSIII 中消息可以通过消息队列作为中介发布给任务，也可以直接发布给任务。在 UCOSII 中有消息邮箱和消息队列。在触摸屏任务 touch\_task() 中向消息邮箱发送一条消息给蜂鸣器任务。而在蜂鸣器任务 buzzer\_task() 中等待消息的到来。

#### (3) 个性化电子琴壁纸的显示

由于实习所用的 LCD 屏幕是 320x240 的，所以在显示图片是要对图片进行裁剪处理。因为是使用的内部 flash, 为了减少内存的浪费，同时对图片进行压缩，然后在使用 image2lcd 软件生成的任意 16 位真彩色图片的数组。本功能的设计是为了给用户一个赏心悦目的可视开机界面。在屏幕上画图，编写扫描函数，选择从左到右，从上到下或者从上到下，从左到右的扫描方式。

(4)电子琴 UI 界面的设计

设置 STM32 与 TFTLCD 模块相连接的 IO。这一步，先将我们与 TFTLCD 模块相连的 IO 口进行初始化，以便驱动 LCD。初始化 TFTLCD 模块。这里我们没有硬复位 LCD，因为 MiniSTM32 开发板的 LCD 接口，将 TFTLCD 的 RST 同 STM32 的 RESET 连接在一起了，只要按下开发板的 RESET 键，就会对 LCD 进行硬复位。初始化序列，就是向 LCD 控制器写入一系列的设置值。在初始化之后，LCD 才可以正常使用。通过函数将字符和数字显示到 TFTLCD 模块上。设置坐标写 GRAM 指令写 GRAM 来实现，但是这个步骤，只是一个点的处理，我们要显示字符/数字，就必须多次使用这个步骤，从而达到显示字符/数字的目标，所以需要设计一个函数来实现数字/字符的显示，之后调用该函数，就可以实现数字/字符的显示了。

由于要设计的界面比较简单，故采用的是在 LCD 上画图的形式来实现 UI 界面的设计。为了更好地利用屏幕资源，所以将 LCD 的设置改为横屏，同时对屏幕建立坐标系，方便绘制电子琴的 UI 界面。

电子琴 UI 界面：

1	2	3	4	5	6	7	嵌 入 式 电 子 琴
. 1	. 2	. 3	. 4	. 5	. 6	. 7	

图 4.2 电子琴 UI 界面

以左上角为坐标原点(0,0)，得出各个点的坐标，忽略屏幕的四周，所以电子琴的 UI 设计就只需要画一条横线和七条竖线就可以完成了。然后通过调用函数在固定的地方显示 1-7 的数字，点和汉字。

为了显示汉字，可以通过 PCtoLCD2002 软件进行汉字取 24x24 汉字点阵，保存在二维数组中，实际上汉字也是通过在 LCD 上根据点阵数据画点来得出来的。通过调用汉字显示的函数来实现在屏幕右方显示“嵌入式电子琴”的字样。

(5)触摸屏的触摸检测

Mini 板的 TFTLCD 模块自带的触摸屏采用的是电容式触摸屏，只支持最多 5 点触摸。T2001A 持最多 5 点触摸，所以只有 5 个位用来表示对应点坐标是否有效，其余位为保留位（读为 0），通过读取该寄存器，我们可以知道哪些点有数据，哪些点无数据，如果读到的全是 0，则说明没有任何触摸。在本次电子琴设

计中我们一般只用到触点的 x, y 坐标，然后判断是哪一块区域被按下，然后 LCD 进行填充，如果检测到的区域和上次不一样，就出去上一次填充的区域，进行下一次填充。

#### (6) 蜂鸣器任务

定义一个一维数组存放频率值，然后配置 PWM，在单片机的 PA8 连接蜂鸣器，建立消息邮箱等待消息的到来，从而发出一定频率的声音。

## 5. 性能测试

### 5.1 测试方法

通过给开发板上电，观察是否出现开机壁纸，然后过几秒，一次按下电子琴上的按键低音和高音 1-7，观察是否出现低音或者高音的“Do、Re、Mi、Fa、S0、La、Si”。如果上述测试都符合，说明符合测试要求

### 5.2 测试结果

经多次验证，测试符合预期要求，实现了本实习所需要的要求，可以实现开发板上电显示壁纸图片，然后显示电子琴的界面，通过依次测试，可以发出“Do、Re、Mi、Fa、S0、La、Si”的音调。



图 5.2 测试结果

## 6. 实习总结

### 6.1 收获

通过本次嵌入式系统课程设计实习，我对 UCOS 的系统有了进一步的了解。我开始对任务的相关 API 有了更进一步的了解。简易电子琴是我刚刚开始做的有关于嵌入式的第一个项目，前期工作主要是通过上网查阅关于电子琴的相关文献资料，主要是学习电子琴的设计思路，以及想着怎么来实现电子琴和我们所学的 UCOS 操作系统相联系。在跟着正点原子的 UCOS 例程来重新复习了一遍，终于慢慢学会了任务调度以及利用消息邮箱来实现消息的传递。利用操作系统，很好地实现任务调度和任务间的通信，具有很好的实时性。通过本次实习，让我们在实践中验证我们之前网课所学的知识点。而且这次实习，让我重新复习了一下 STM32 的其他外设，例如 LCD, 电容触摸屏等的基本用法。在中期调试完成代码，实现了基本的简易电子琴的基础上，我又想着给电子琴加上一个开机壁纸，壁纸选择的是一幅钢琴的动漫图片，让它显示 3 秒，然后在显示出所设计的电子琴。然后按照小星星的乐谱弹了一下，体验到完成设计的快乐。在实习中，和队友分工合作，我主要负责的是总体方案的设计和 UI 界面的设计，主要是把按照屏幕大小建立坐标轴画图，以及按键按下时对按下的区域进行填充，实现按键的动感，开发板上电后显示的壁纸，也是我后面临时起意加上去的。在实习中，我深深感受到 C 语言对于嵌入式开发真的很重要，嵌入式比较偏向软件的设计，所以在大学剩下的日子里还是得学好 C 语言。在实习中，我觉得不足的是，没有把所学过的 EmWin 用来设计界面，而是采用了传统的方法在 LCD 上绘图的方式实现界面的设置。

## 7、参考文献

- [1]董开茂,李红岭,杜志维,张月星,王家祺.基于单片机的电子琴设计[J].数字技术与应用,2019,37(04):181-182.
- [2]路曦.简易电子琴的设计与实现[J].科技经济导刊,2018,26(05):41.
- [3]顾严,邢岩.基于单片机的微型电子琴建模[J].科技资讯,2019,17(02):112+116.
- [4]纪虹.双排键电子琴在我国的发展现状[J].林区教学,2007(03):119-120.



## 附录：源程序

(1) 电子琴 UI 界面加载函数

```
void ucos_load_main_ui(void)
{
    LCD_Clear(WHITE); //清屏
    LCD_Display_Dir(1);
    POINT_COLOR=BLACK; //设置字体为红色
    //纵向
    LCD_DrawLine(0, 0, 0, lcddev.height);
    LCD_DrawLine(40, 0, 40, lcddev.height);
    LCD_DrawLine(80, 0, 80, lcddev.height);
    LCD_DrawLine(120, 0, 120, lcddev.height);
    LCD_DrawLine(160, 0, 160, lcddev.height);
    LCD_DrawLine(200, 0, 200, lcddev.height);
    LCD_DrawLine(240, 0, 240, lcddev.height);
    LCD_DrawLine(280, 0, 280, lcddev.height);
    //横向
    LCD_DrawLine(0, lcddev.height/2, 280, lcddev.height/2);
    //mark
    LCD_ShowChar(10, 10, '1', 24, 0);
    LCD_ShowChar(50, 10, '2', 24, 0);
    LCD_ShowChar(90, 10, '3', 24, 0);
    LCD_ShowChar(130, 10, '4', 24, 0);
    LCD_ShowChar(170, 10, '5', 24, 0);
    LCD_ShowChar(210, 10, '6', 24, 0);
    LCD_ShowChar(250, 10, '7', 24, 0);

    LCD_ShowChar(10, 130, '1', 24, 0);
    LCD_ShowChar(50, 130, '2', 24, 0);
    LCD_ShowChar(90, 130, '3', 24, 0);
    LCD_ShowChar(130, 130, '4', 24, 0);
    LCD_ShowChar(170, 130, '5', 24, 0);
    LCD_ShowChar(210, 130, '6', 24, 0);
    LCD_ShowChar(250, 130, '7', 24, 0);

    TP_Draw_Big_Point(15, 127, BLACK);
    TP_Draw_Big_Point(55, 127, BLACK);
    TP_Draw_Big_Point(95, 127, BLACK);
    TP_Draw_Big_Point(135, 127, BLACK);
    TP_Draw_Big_Point(175, 127, BLACK);
    TP_Draw_Big_Point(215, 127, BLACK);
}
```

```

TP_Draw_Big_Point(255, 127, BLACK);

POINT_COLOR=BLUE;//设置字体为蓝色

OLED_Show_Font(288, 12, 3);           //嵌
OLED_Show_Font(288, 48, 4);           //入
OLED_Show_Font(288, 84, 5);           //式

OLED_Show_Font(288, 132, 6);           //电
OLED_Show_Font(288, 168, 7);           //子
OLED_Show_Font(288, 204, 8);           //琴

}

```

## (2) 开始任务

```

void start_task(void *pdata)
{
    OS_CPU_SR cpu_sr=0;
    pdata = pdata;
    msg_key=OSMboxCreate((void*)0); //创建消息邮箱
    msg_fre=OSMboxCreate((void*)1); //创建消息邮箱
    sem_led1=OSSemCreate(0);        //创建信号量
    OSStatInit();                    //初始化统计任务. 这里会延时 1 秒钟左右
    OS_ENTER_CRITICAL();              //进入临界区(无法被中断打断)
    OSTaskCreate(buzzer_task, (void
*)0, (OS_STK*)&BUZZER_TASK_STK[BUZZER_STK_SIZE-1], BUZZER_TASK_PRIO);
    //创建蜂鸣器任务
    OSTaskCreate(touch_task, (void
*)0, (OS_STK*)&TOUCH_TASK_STK[TOUCH_STK_SIZE-1], TOUCH_TASK_PRIO);

    //OSTaskCreate(led1_task, (void
*)0, (OS_STK*)&LED1_TASK_STK[LED1_STK_SIZE-1], LED1_TASK_PRIO);

    OSTaskCreate(main_task, (void
*)0, (OS_STK*)&MAIN_TASK_STK[MAIN_STK_SIZE-1], MAIN_TASK_PRIO);

    OSTaskCreate(key_task, (void
*)0, (OS_STK*)&KEY_TASK_STK[KEY_STK_SIZE-1], KEY_TASK_PRIO);

    OSTaskSuspend(START_TASK_PRIO); //挂起起始任务.
    OS_EXIT_CRITICAL();               //退出临界区(可以被中断打断)
}

```