大连海事大学

单片机应用实践报告

实验内容：电子琴

专业班级：电子信息工程4班

姓 名：李金格

学 号：2220211507

1. 摘要

随着科技的进步，人们越来越追求生活的多元化和娱乐化。传统的电子琴虽然功能丰富，但成本较高，而且不便于携带。因此，基于STM32的电子琴应运而生，它具有成本低、易于携带、易于扩展等特点，能满足大众对音乐娱乐的需求。主要内容上，STM32电子琴的设计和实现主要包括硬件设计和软件设计两个部分。硬件设计主要选择合适的STM32芯片，设计电路板，包括音频输入输出、按键输入、显示屏等模块。软件设计则主要编写控制程序，实现按键扫描、音频处理、显示屏控制等功能。此外，还需要对系统进行测试，以确保系统的稳定性和可靠性。要实现STM32电子琴设计，首先需要对STM32芯片和其周边硬件有深入的了解和掌握，能够根据需求进行硬件选择和电路设计。其次，需要熟悉嵌入式系统的编程，能够编写出实现按键扫描、音频处理、显示屏控制等功能的程序。此外，还需要对音频处理有一定的了解，能够根据按键输入产生对应的音频输出。最后，需要对系统进行全面的测试，以确保系统的稳定性和可靠性。总的来说，STM32电子琴的设计和实现是一个涵盖了硬件设计、软件编程、音频处理等多个领域的综合性项目，是对嵌入式系统技术和音乐技术的完美结合。

二、设计的任务和要求

一、需求分析

电子琴应能够识别用户的按键输入，并根据输入播放相应的音符。

二、硬件设计

1. 选择合适的STM32系列芯片，根据需求确定所需的IO口数量、时钟频率等参数。

2.设计按键输入电路，确保按键的稳定性和灵敏度。

3.设计音频输出电路，支持高质量的音频输出

4.加入外设接口，通过ST-LINK实现软硬件连接。

5.利用OLED提供直观的音符显示功能，使得操作和设计简洁明了

三、软件编程

1.编写STM32的初始化代码，包括IO口配置时钟配置等。

2.实现按键扫描与识别功能，确保按键的实时响应。

3.编写音符播放代码，根据按键输入播放相应的音符。

四、测试与优化

1.对电子琴进行全面测试，确保各项功能正常运行。

2.针对测试中发现的问题进行优化改进，提升设备性能。

综上所述，STM32电子琴设计涉及多个方面，需要综合考虑硬件设计、软件编程以及测试与优化等任务与要求。通过合理的设计与实现，可以打造出一款功能强大、性能稳定的电子琴产品。

三、研究的目的和意义

研究目的:

本课程设计的电子琴是一个利用STM32微控制器设计和实现的音乐设备。该项目基于STM32F103C8T6单片机控制器，并结合了数码管模块、按键以及蜂鸣器等硬件组件。STM32电子琴的设计不仅是对嵌入式系统硬件和软件设计的实践，同时也是对音乐和技术结合的探索。

意义:

STM32电子琴的设计具有实际应用价值。它可以作为一个独立的音乐工具，用于创作和表演。此外，它还可以作为其他更复杂音乐系统或音频处理应用的基础。通过STM32电子琴的设计和实现，我接触和了解嵌入式系统和音频处理技术，激发了我对这些领域的兴趣。

STM32电子琴设计提供一个将理论知识应用于实际项目的机会，加深我对嵌入式系统、STM32微控制器等的理解和运用。研究STM32电子琴不仅有助于推动技术和艺术的结合，还具有实际应用价值、技术普及和推广以及科研的推动作用。

1. 系统方案的论证和选择

方案一:

采用 MCS-51系列单片机来实现设计要求的功能。MCS-51系列单片机中的基本型产品是 8051,8031和8751。8位CPU拥有片震荡器与时钟电路:32根IO线;外部存储器 ROM 和 RAM 寻址围各 64KB;2个 16 位的定时器/计数器;5 个中断源，2个中断优先级;全双工串行口;8051的中央处理器 CPU 由运算器和控制逻辑构成51 单片机是一款比较基础的单片机。I/0(输入/输出)引脚系统结构紧凑，功能简单，低成本。可以实现各种丰富的应用。通过控制I/0口的输出电平的翻转频率来实现对蜂鸣器发音音调的控制。

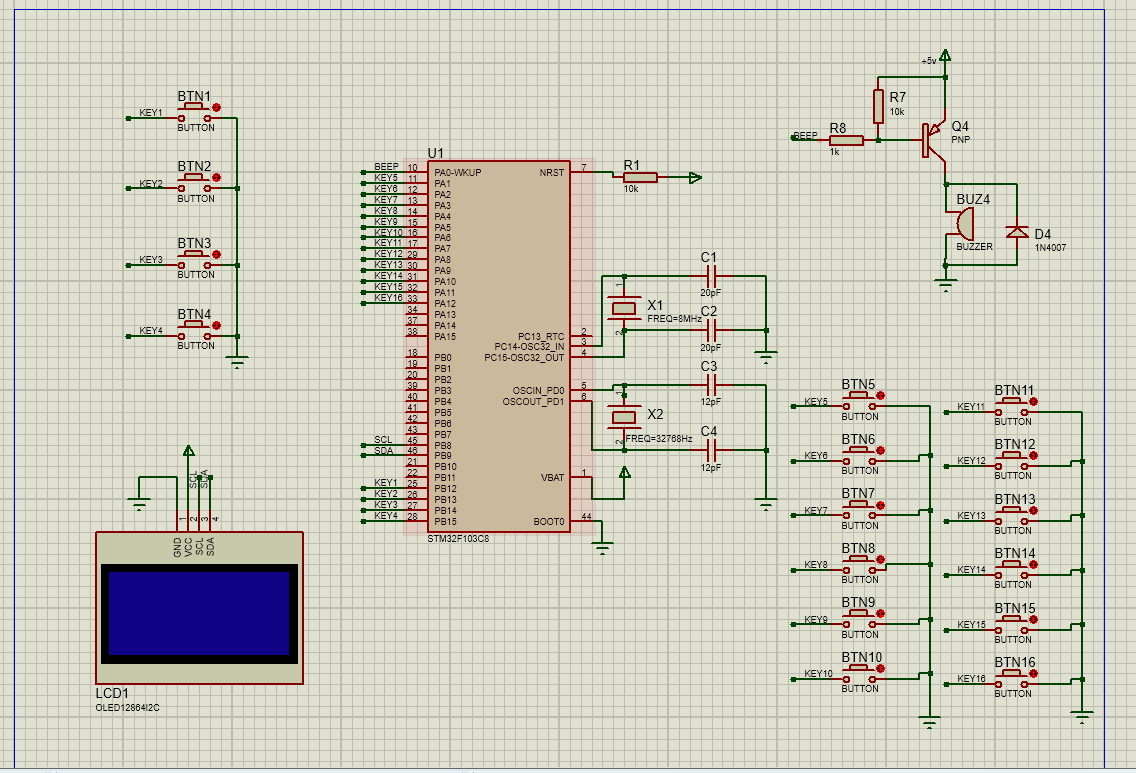
方案二:

采用 STM32f03C8T6 来实现此次设计，最高72MHz工作频率。它有多个外部中断,八个定时器,有 PWM 输出模式,其中2个高级定时器，两个基本定时器，其他的是通用定时器,共 48 个引脚,除基本的I/O功能之外还包含有复用功能，其外设功能强大，是一款能够很方便使用的功能强大的芯片，同时可以直接对相关寄存器进行操作。

方案选择：

选择上述两种方案相比，51单片机虽然价格便宜，但是其功能简单，1/0口少，并且仅仅只有5个中断源,而 STM32f03C8T6 共 48个引脚,并且其定时器有 PWM 输出模式，可以更加方便的控制蜂鸣器的发音。

1. 系统硬件设计



元器件清单:

STM32F103C8T6微处理器、12个独立按键、蜂鸣器、0.97寸OLED显示屏、导线若干

（1）蜂鸣器：将其连接在PA0输出，利用Buzzer\_Init函数进行初始化。通过控制GPIO引脚的电平使蜂鸣器发声：当GPIO引脚输出高电平（或特定的使能信号）时，蜂鸣器开始振动并发出声音；当输出低电平或撤销使能信号时，蜂鸣器停止振动。

（2）控制按键：控制12个音的按键的是PA1-PA12，调节音长音调的按钮接在控制按钮：PB12-PB15。首先初始化GPIOA的12个引脚和GPIOB的4个引脚，再定义一个函数Key\_GetNum来检测按键是否被按下，并返回相应的键码值。

（3）显示屏：STM32F10x微控制器与OLED屏幕之间的通信使用了I2C通信协议，引脚分别为PB8和PB9，通过OLED\_ShowString函数显示静态字符串。

1. 软件设计

main函数解析：

#include "stm32f10x.h" // Device header

#include "Delay.h"

#include "OLED.h"

#include "Timer.h"

#include "LED.h"

#include "Key.h" //这些是包含所需头文件的指令。stm32f10x.h 是用于指定特定型号的STM32微控制器的头文件，其他头文件包含了延时函数、OLED显示屏控制函数、定时器控制函数、LED控制函数和按键控制函数的声明。

extern uint8\_t KeyNum;

int Count=0; //用来实现控制声音长度的

int i=0;

int num[22]={0,190,170,151,143,127,113,100,95,85,75,71,63,56,50,47,42,37,35,31,28,25};

int flag=0; //音调标志位(范围是-3~~6)

int length=2500; //声音长度预设值

//定义了一些全局变量，包括用于声音长度控制的 Count，音调标志位 flag，声音长度 length，以及用于定时器中断的计数器 i 和存储频率值的数组 num。

int main(void)

{

/\*模块初始化\*/

OLED\_Init(); //OLED初始化

Timer\_Init(); //定时中断初始化

Key\_Init\_A(); //PA口初始化

Key\_Init\_B(); //PB口初始化

Buzzer\_Init(); //蜂鸣器

/\*显示静态字符串\*/

OLED\_ShowString(1, 1, " -----Music---- "); //1行1列显示字符串Num:

OLED\_ShowString(2, 1, "Count:"); //显示音调

OLED\_ShowString(3, 1, "Tone:"); //现实计数值

OLED\_ShowString(4, 1, "length:"); //显示预设值,计数值达到预设值键码变为0不发声

while (1)

{

KeyNum=Key\_GetNum();

OLED\_ShowNum(2, 7, num[KeyNum+3+flag], 5); //音调（频率值）

OLED\_ShowSignedNum(3, 7, flag, 5); //音调标志位

OLED\_ShowNum(4, 8, length, 5); //响声长度

}

}

void TIM2\_IRQHandler(void)

{

if (TIM\_GetITStatus(TIM2, TIM\_IT\_Update) == SET) //判断是否是TIM2的更新事件触发的中断

{

if(KeyNum!=0){

i++;

if(Count==length){Count=0;KeyNum=0;GPIO\_SetBits(GPIOA, GPIO\_Pin\_0);} //达到预定计数值键码为0不发声，检查是否达到预设的声音长度，如果是，则停止发声，并将按键值重置为0。

Count++;

if(KeyNum==14){length+=100;KeyNum=0;} //调节声音长度实现声音长短变化 PB12

if(KeyNum==15){length-=100;KeyNum=0;} // PB13

if(KeyNum==16){flag+=1;KeyNum=0;} //调节音调 PB14

if(KeyNum==17){flag-=1;KeyNum=0;} // PB15

if(flag<=-3){flag=-3;} //音调调节范围（-3~~6）

if(flag>=6){flag=6;}

if(length>=9000){flag=9000;} //长度调节范围（-100~~9000）

if(length<=100){length=100;}

if(KeyNum==1){

TIM\_SetAutoreload(TIM2,num[KeyNum+3+flag]); //每个键码有个单独的重装值 PA1-PA12

if(i%2==0){GPIO\_SetBits(GPIOA, GPIO\_Pin\_0);} //翻转电频实现发出相应的音调

//如果按键值为1，则设置定时器重装值，并根据计数器翻转电平以发出相应的音调。

//如果计数器为偶数，则设置GPIOA的引脚0为高电平。如果计数器为奇数，则设置GPIOA的引脚0为低电平。

...

...

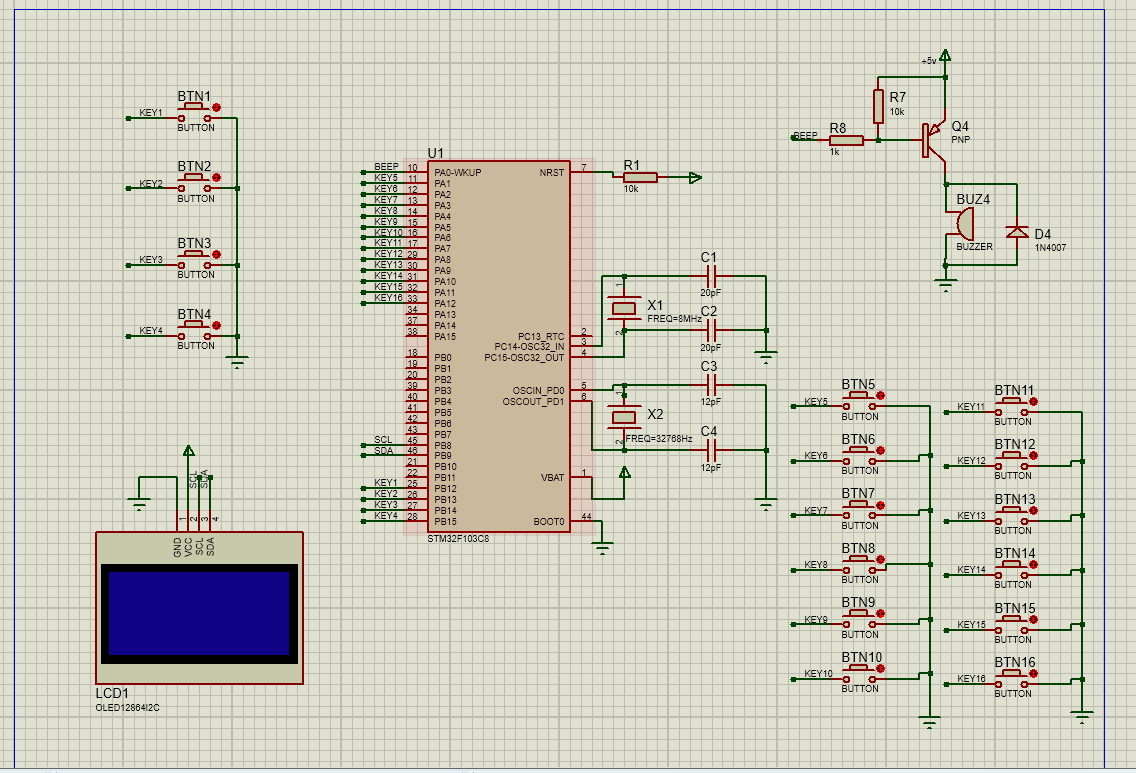
TIM\_ClearITPendingBit(TIM2, TIM\_IT\_Update); //清除TIM2更新事件的中断标志位

//中断标志位必须清除,否则中断将连续不断地触发，导致主程序卡死

}

}

七、系统仿真调试



八、系统功能测试

实现的功能：

通过按下琴键来播放音符。设计的电子琴包括12个琴键，每个琴键对应一个特定的音符/音调，且从左到右按键按下会呈依次升调，有规律，可以实现任意琴谱的演奏。琴键的状态(按下或松开)会被实时检测，并转化为相应的音符播放出来。另外有4个按键分别控制音调变高/变低、单个按键按下时发声持续时长变长/变短，同时OLED显示屏上会显示当下音调和音长的数据。按下setting复位键可使调后的音调、音长恢复到初设值，显示屏对应显示count：00151，tone：＋00000，length：02500

测试数据记录：

|  |  |  |
| --- | --- | --- |
| 初始值 |  |  |
| 按下第一个播放按键，count从151变为143，在发音结束后变回151 |  |  |
| 按下第二个播放按键，count从151变为127，在发音结束后变回151  剩余10个按键同理 |  |  |
| 按下右侧第一个按键（音长增加），length由02500变为02600，此时按下任一播放按键，可听到发声持续时间变长 |  |  |
| 按两次右侧第三个按键（音调增加），length由0变为2，此时依次按下播放按键，可听到一排按键相比初始统一音调变高了 |  |  |

十、总结

在本次STM32电子琴课程设计中，我们成功地实现了一个基于STM32微控制器的电子琴系统，但功能比较单一，实时演奏，不能进行录制播放。在实物搭建过程中，我们一开始选择了4×4矩阵按键组，但考虑到不够简洁直观，遂换成16个单个按键，排列成一排，使得操作更加简便，按键功能一目了然，也更符合生活常识。实验中我们出现了音调不准、蜂鸣器长鸣的情况，排查了代码故障后，又重连实物图，换导线、按键和器件，调试后使得电子琴运作正常。

通过本次STM32电子琴课程设计，我们深入了解了STM32的开发流程、电路设计以及软件编程，同时也提升了我们的动手能力和团队合作精神，我深刻体会到了团队合作的重要性。在项目实施过程中，我们互相学习、互相支持，共同解决了许多技术难题。此外，我也认识到了实践经验的宝贵性，只有将理论知识与实际项目相结合，才能真正掌握技能。