基于 STM32 的智能多彩台灯设计

学生姓名：	李筠颀	组长	学号：	2230506226
学生姓名：	李宪忠	组员	学号：	2230506248
学生姓名：	吴瀚东	组员	学号：	2230506225
专业班级：	**2022** 电子信息工程 **2** 班			
指导教师：	何永玲			
二级学院：	医药信息工程学院			

2023 年 5 月 17 日

基于 STM32 的智能多彩台灯

摘要

随着智能家居的不断发展，智能台灯也成为了人们生活中不可或缺的一部分。本文介绍了一款基于 STM32 单片机的智能多彩台灯，该台灯具备语音识别、RGB 氛围灯、定时功能等多种功能。

本文以智能多彩台灯为研究对象，探讨语音识别技术、自动控制技术等，通过单片机的处理器进行数据处理，完成对智能多彩台灯的控制。系统以 STM32F103C8T6 单片机作为主控制芯片，利用语音识别模块，定时器模块，OLED 显示。最终实现了台灯的语音识别，定时，调光，模式切换等的功能。

实验结果表明，多功能多彩台灯能按照命令要求完成操作。本设计性价比高，功能齐全，多功能多彩台灯能为日常生活、高效的工作服务，具有良好的应用价值。

[关键词]：多功能多彩台灯；语音控制

目　录

1. 前言

1.1 研究的目的与意义

基于 STM32 的智能多彩台灯的研究目的是通过整合先进技术，实现灵活、可控的 LED 光源智能化控制，并将其应用于生活场景中。具体来说，该研究的目的和意义主要包括以下几个方面：

实现智能控制：该研究可以实现智能控制功能，让用户能够通过语音指令、按键操作等方式进行操作，提高了用户的使用便利性和舒适度。

提高灯光质量：通过灯光颜色、亮度、渐变等多种方式进行调节，可以满足用户不同场景下的需求，提高了使用的效果和舒适感。

降低能耗：在定时开启和红外人体感应开关的作用下，可以有效地降低能耗，节约能源，同时也降低了对环境的污染。

综上所述，基于 STM32 的智能多彩台灯的研究目的和意义是为了实现更加智能化、环保和舒适的灯光控制系统，并推动开源硬件平台在生活场景中的应用。

2. 方案设计

2.1 设计思路

基于 STM32 的多功能多彩台灯控制系统设计包括硬件和软件设计 2 个部分，其中硬件部分采取拼接式结构，使用 Proteus 软件绘制电路，用万用板焊接控制板，使用塑料 、亚克力板、等制作底盘。软件部分就是对整个硬件主程序进行设计，利用 C 语言编写控制代码。包括、主程序代码、语音识别程序、菜单程序、调光程序等。

本设计是基于 STM32 的多功能多彩台灯控制系统，本设计硬件组成主要有单片机电路、LED 驱动电路、蜂鸣器电路、电源电路、OLED 电路、按键电路；其中单片机主要完成数据处理及运算主要对当前的工作状态进行控制；语音模块用于获取操作语音；LED 驱动电路来驱动多功能多彩台灯、蜂鸣器电路用于蜂鸣器的提示、按键电路用于识别按键状态、OLED 电路用于 0.96 寸的 OLED 屏的驱动、电源电路主要为系统提供适合的供电电压。本设计具备较高的安全性及稳定性，拥有较为广泛的应用前景。

另外，本次设计原计划是有包装和外观设计的，但最后由于时间不足，和手工技术的不足，没能实现此项目，以下是外观设计的图纸：

3. 硬件电路设计

3.1 STM32 单片机最小系统电路设计

3.1.1 STM32 单片机简介

根据前面的方案设计可知，本设计使用 STM32F103C8T6 单片机作为主控制器。完成系统传感器的识别，以及小车运动方向的控制。STM32F103C8T6 处理器的内核是 cortex-M3, 64KB 的 FLASH 内存容量以及 1 MB 的闪存，2 个 12 位，1us 模数转化器，可以映射到 16 个模数转化通道，3 个异步串行通信通道。STM32F103C8T6 单片机芯片原理图如下所示：

3.1.2 晶振电路设计

从上可知 STM32F103C8T6 处理器的主频是 72MHz，因此需要外接一个晶振，产生一定的时钟频率。在处理器的 PD0 和 PD1 这两个引脚外接一个频率无诶 8MHz 的石英晶振，处理器内部有一个 9 倍频电路，把频率倍频为 72MHz。为了使晶振能正常起振，需要在石英晶振旁边并联 2 个容值相同的电容。

3.1.3 复位电路设计

基于 STM32 的多功能多彩台灯在运行的过程中如果入到程序有 bug,或者其他的异常，就有可能导致系统跑飞。此时就需要通过复位按键来控制处理器复位，使程序从头开始运行。STM32F103C8T6 的复位电路如下图所示：

STM32F103C8T6 处理器必须通过低电平触发复位，因此设计电路把复位引脚 RST 上拉一个电阻连接到 VCC3.3。然后并且把一个独立按键和一个电容并联连接到地之后，连接到复位引脚上。复位引脚连接到独立按键和上拉电阻中间。在正常情况下，此时电容充满电，电路断开，RST 引脚的电压等于 3.3 伏，系统不复位。需要复位时，按下独立按键，此时把 RST 引脚和 GND 导通，RST 引脚电压变为 0，此时系统内部复位。

3.1.4 电源电路

STM32F103C8T6 处理器的供电电压为 2 伏到 3.6 伏，正常供电电压为 3.3 伏，而常用的电源适配器通常是 5 伏输出，因此需要设计一个电压转化电路，把 5 伏电压转化为 3.3 伏电压。为此，这使用 12V/1A　5V/2A 1.5A 的电源适配器，和 5V，3.3V 稳压模块。其中包括 Ams1117-3.3 芯片，可将输入的正 12 伏电压转化为 3.3 伏电压，以及 LD1117 芯片，可将输入的正 12 伏电压转化为 5 伏电压。

3.2 蜂鸣器电路

利用 NPN 三极管实现开关作用，GPIO 引脚置高电平导通。二极管用于显示蜂鸣器工作状态。以下是电路图：

3.3 LED 驱动电路

此电路通过 12V 供电驱动大功率灯珠，PA0~PA3 引脚输出 PWM 波形，结合 NPN 三极管实现电路的通断，从而控制 LED 灯珠的亮度调节。RGB 灯珠共阴极，三个端口分别控制三种颜色的 PWM 波形。电路图如下：

3.4 语音识别模块 LD3320 电路

该模块通过 5V 电压驱动，STM32 中的 USART1 的发送端口 TXD 为 PA9，接 LD3320 中的接收端口 RXD；接收端口 RXD 为 PA10，接 LD3320 中的发送端口 TXD。因为此程序只有 LD3320 在发送数据，其实这里不需要要接 PA9。但是考虑到后期的进一步开发，这里还是接上。以下是电路图：

3.5 按键电路

按键按下，GPIO 端口电平拉低，GPIO 检测输入电流。以下是电路图：

3.6 OLED 电路

本次设计选用的是 I2C 通讯协议的 4 针 OLED 屏，电路图如下：

3.7 总电路图

以下是总电路图：

4. 软件系统设计

硬件上对于电路方面的各个设计已经完成，下一步就需要完成核心—程序设计。硬件上是基础，也是最重要的实物，软件是将这些硬件赋予灵魂，而两者的相互结合才能完成整个作品，所以，软件的设计在整个系统中是非常非常重要的，也是整个系统中比较有技术含量的设计。

4.1 系统软件介绍

单片机程序代码的开发与调试采用的是 KEIL 开发工具来实现，它是一种 C 汇编语言软件应用程序以及软件开发操作系统。Keil 这开发工具在我们使用过程中，编译功能极其强大，其中更多的能够直接体现为写出一种高级化的编程语言。下面为大家详细地简单介绍了这个 Keil 开发器管理系统各个组成部分的基本功能及其日常运行。

本设计中采用的中控制器是 STM32 单片机，程序编辑上采用的语言是 C 语言，这是一种通用的计算机编程设计语言，是广泛使用的编程语言之一。它的使用范围较广，不仅仅只是用于计算机系统程序的编写，还能够用于编写一般的应用程序。C 语言含有良好的可移植性，在使用时非常方便，且运用简单，除此之外，它还有非常好的硬件控制能力，以及非常强的表达、运算能力。

对于单片机系统来说，系统上电复位初始化操作是开始工作的第一步，也是非常重要的一步，能够有效防止 CPU 在发出错误的指令，导致执行错误操作，也能够提高单片机系统的性能。初始化复位操作，能够影响到整个系统的工作可靠性，初始化操作不单单只是上电可操作，也可进行使用复位电路操作，恢复到最初的状态，若是在程序出现错误，导致出现“死机”、“卡死”等现象时，能够通过初始化操作进行恢复。

4.2 系统主程序软件设计

部分代码：

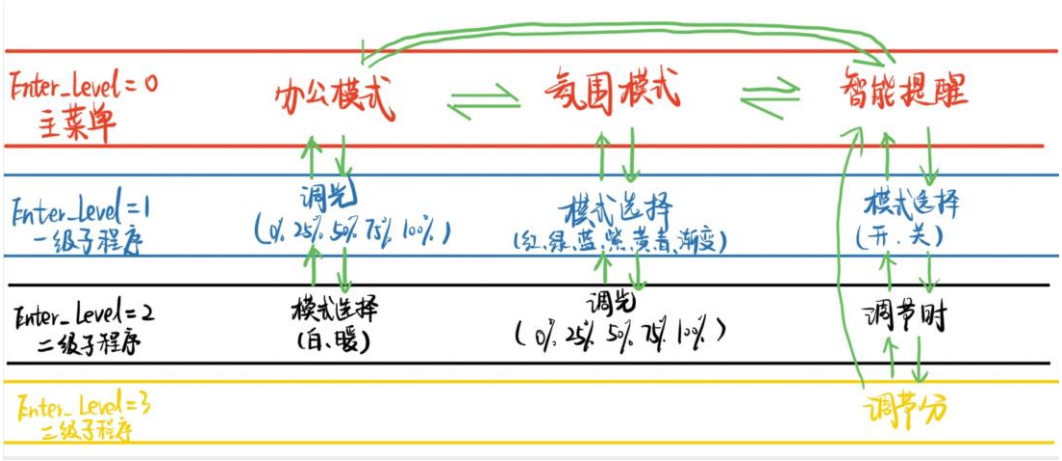
```
while(1)
    {
        Key_Clear ();
        Key_Scan ();
        if(VoiceFlag == 1)   //语音模式
        {
            Voice ();
            Delay_ms (1000);
            VoiceFlag=0;
            mr=0;
```

```
            Clear=1;
        }
        else   //手动模式
        {

            if(mr==1)   //默认标志位
            {
                Dimming();
            }
            else
            {
                if(Enter_Level==0)
                    Menu ();   //主程序
                else
                    Menu_Enter ();   //子程序
            }
        }
    }
```

4.3 多级子程序软件设计

此处程序设计利用状态机的原理，通过设立主次程序标志位，来确定当前的系统程序状态，使得多级程序间能够准确无误地进行切换，以下是主程序以及子程序软件设计流程图：



部分代码：

//** 模式判断

```
void Menu_Mode(void)
{
    if(Flag_Key_Left==1)
    {
        Clear=1;
        Menu_Mode_Num--;
    }
    else if(Flag_Key_Right==1)
    {
        Clear=1;
        Menu_Mode_Num++;
    }
    else if(Menu_Mode_Num==4)
    {
        Menu_Mode_Num=1;
    }
    else if(Menu_Mode_Num==0)
```

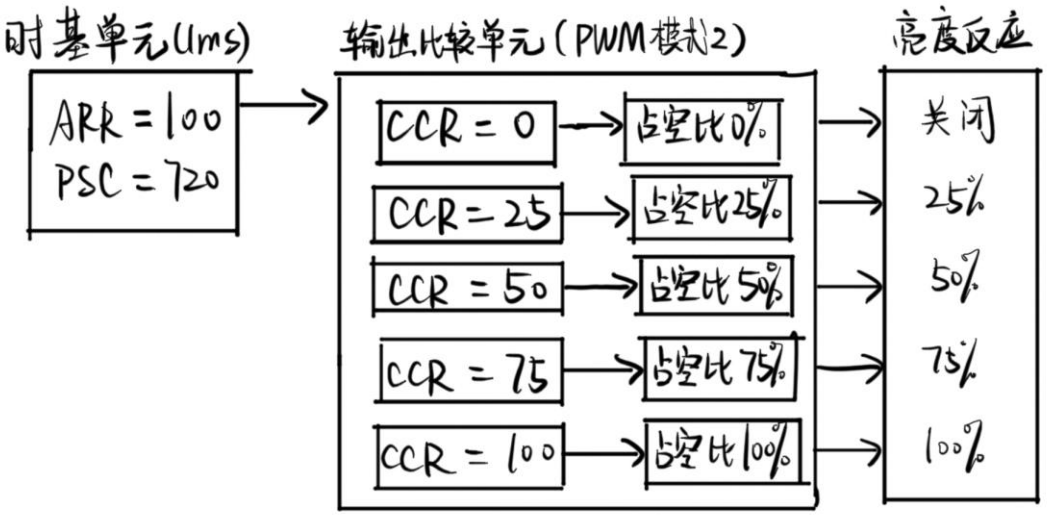
```
    {
        Menu_Mode_Num=3;
    }
}

//**********************************************************模式进入
void Menu_Enter(void)
{
    if(Menu_Mode_Num==1)   //办公模式
    {
        Dimming();
    }
    else if(Menu_Mode_Num==2)   //氛围模式
    {
        if(Enter_Level==1 || Enter_Level==3)
            RGB();
        if(Enter_Level==2)
            Dimming_RGB();
    }
    else if(Menu_Mode_Num==3)   //定时模式
    {
        Regular();
    }
}
```

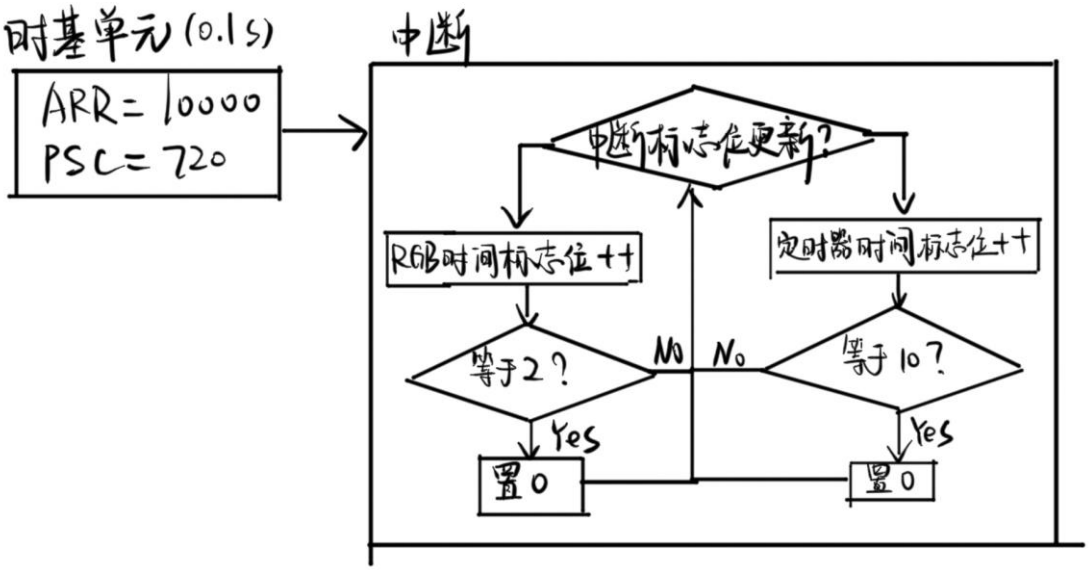
4.4 定时器以及中断程序软件设计

STM32 的定时器功能丰富，分为基本定时器，通用定时器以及高级定时器，此程序主要利用了 STM32 中通用定时器 2 和定时器 3。

以下是定时器 2 的软件设计流程图：



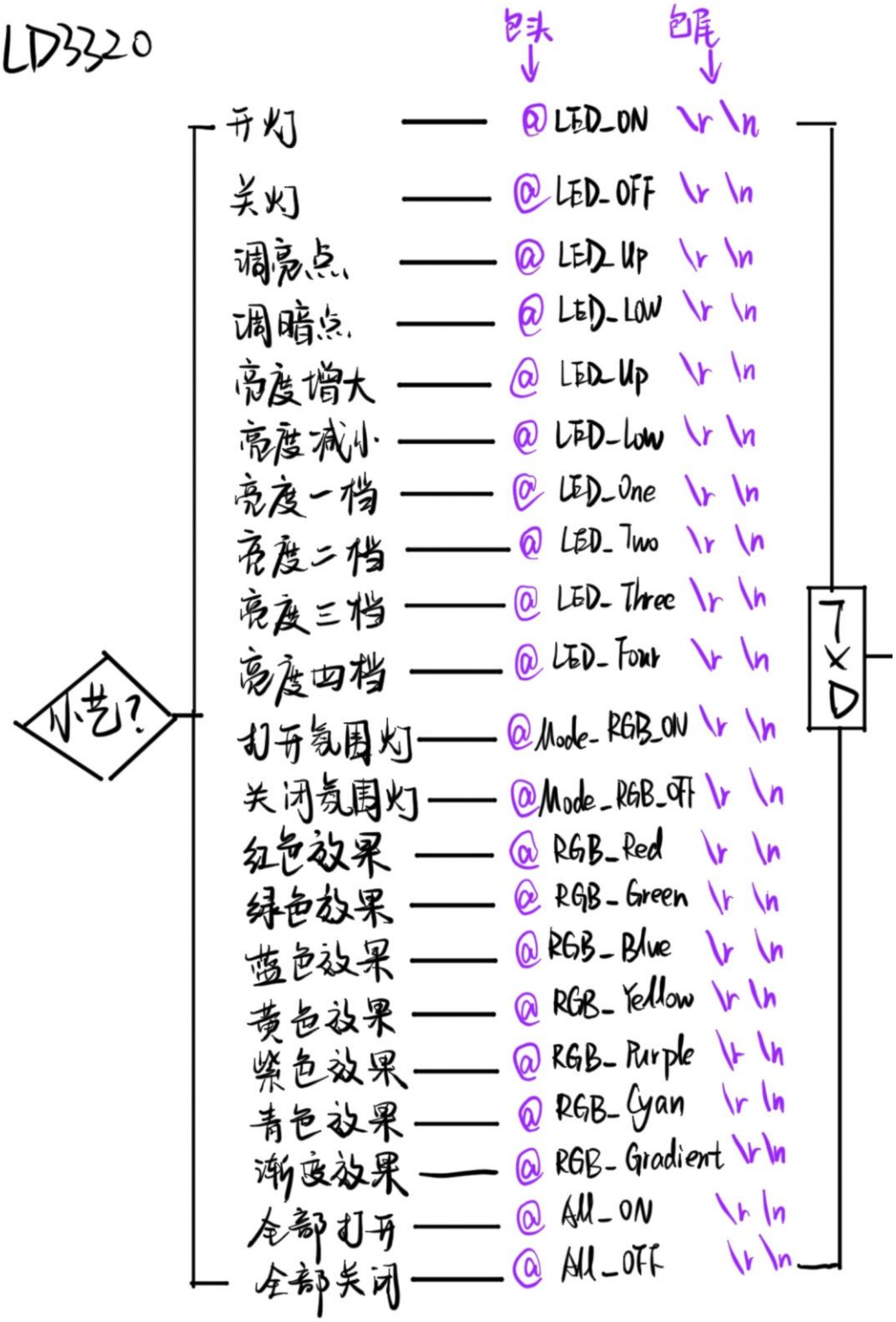
以下是定时器 3 的软件设计流程图：



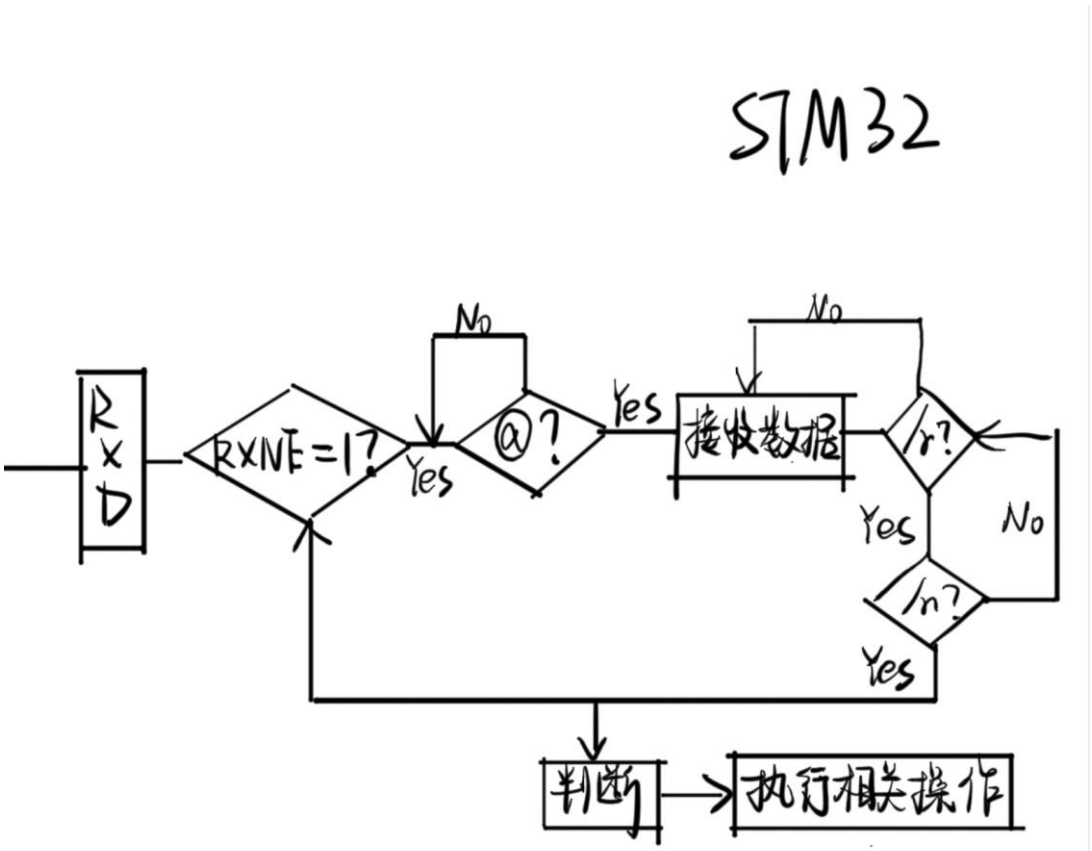
4.5 语音识别程序软件设计

此模块利用 USART 串口协议使 STM32 与语音识别模块 LD3320 实现串口通

信。并且设定了数据包的规则，以’@’为包头，’\r\n’为包尾，对数据进行打包。减少数据传输过程中的差错。具体软件设计流程图如下：



接上图：



部分代码：

```
//*************************************************************中断
void USART1_IRQHandler(void )
{
    static u8 RxState=0;    //只置一次 0
    static u8 i=0;

    if(USART_GetFlagStatus(USART1 ,USART_FLAG_RXNE) == SET)
    {

        RxData = USART_ReceiveData(USART1 );
```

```
if(RxState == 0)
{
    if(RxData == '@' && RxFlag == 0)
    {
        RxState = 1;
        i=0;
    }
}
else if(RxState == 1)
{
    if (RxData == '\r')    //回车符
    {
        RxState = 2;
    }
    else
    {
        Serial_RxPacket[i] = RxData;
        i ++;
    }
}
else if(RxState == 2)
{
    if (RxData == '\n')
    {
        RxState = 0;
        Serial_RxPacket[i] = '\0';
        RxFlag = 1;    //数据接收完毕
        VoiceFlag=1;    //进入语音模式标志位
```

```
            }
        }
    }
    USART_ClearITPendingBit (USART1 ,USART_IT_RXNE );   //清除标志位
}
```

6 总结

由于时间紧迫，本次设计目前只完成了部分功能并取得一定成果。同时也由于工作量大、缺乏知识量和经验，成品存在很多问题，后续还需要时间完善和实现一些存在的问题和功能。本设计在设计的过程中过多地侧重功能实现，在硬件上没有充足的时间设计和调试，导致在截止日期前没能在硬件上完美实现软件的功能。且因为其过程中的各种因素，多次使得工程停滞不前。因此在今后的设计过程中应该多学习电路基本知识，多动手，积累经验。

通过本次的设计，学到了很多，包括环境的创建，程序的开发，硬件的设计、焊接与制作流程。各方面都得到了锻炼。本学期的课程多也重要，平时也是忙里偷闲来学习单片机，一边学习一边制作。虽然这段时间很累，目前也没能打到预定目标，但是这是一定是一段很好的锻炼经历。