家居氛围灯

设计文档

产品	WS2812 氛围灯		
日期	2022/5/20	设计人	陈星
单位		审阅人	

产品说明：

以 STC15 系列为主控 MCU，选用 WS2812 驱动芯片控制 35 个 RGB LED 灯，利用开关、按键和旋转编码器等外围设备实现全亮、全灭、七彩灯、呼吸灯、流水灯 5 种功能，其中利用自复位按键控制氛围灯的功能，使用旋转编码器进行调色调光。本产品可调节 50 余种颜色，四档亮度，使用简单方便，灯光稳定，适合用于各种家居环境与一些灯光亮度要求中等的日常活动场合。

目 录

1 设计需求.....4
2 总体方案设计.....4
2.1 设计思路.....4
2.2 系统方案设计.....4
3 硬件设计.....5
3.1 MCU 选型.....5
3.2 电源设计.....6
3.3 灯组级联电路.....7
3.4 MCU 外围电路.....8
3.4.1 按键电路.....9
3.4.2 旋转编码器电路.....9
4 软件设计.....11
4.1 设计思路.....11
4.2 主程序流程图.....12
4.3 初始化程序.....13
4.4 延时程序设计.....14
4.5 按键程序设计.....14
4.5.1 单次按键值获取.....14
4.5.2 定时器初始化以及中断程序.....15
4.6 灯组传输与控制程序设计.....16
4.6.1 WS2812B 数据协议与驱动方式.....16
4.6.2 WS2812B 传输程序.....18
4.6.3 WS2812B 控制程序.....21

（1）颜色布置 21
（2）全亮控制 21
（3）七彩灯控制 21
（4）呼吸灯控制 21
（5）流水灯控制 22
（6）调色程序 23
（7）调光程序 23
5 原理图 25

1 设计需求

主要应用场所针对家居领域，以单片机为主控芯片，LED 氛围灯需要实现的基本功能有调节亮度、切换颜色功能。

（1）合理选择氛围灯材料与型号以及符合所选应用场景的颜色和亮度范围；

（2）元器件的选型以及设计控制系统的各模块电路；

（3）单片机控制程序的编写；

（4）完成仿真调试直至实现产品功能；

（5）万能板上完成元器件焊接；

（6）制作成品并上电调试。

2 总体方案设计

2.1 设计思路

氛围灯主要是由单片机主控模块、编码器模块、按键模块、灯组控制模块以及 LED 灯组模块所组成（图 2.1）。其中由编码器模块、按键模块作为外围设备输入信号给单片机，单片机作为主控芯片，通过 I/O 口的分配来控制灯组控制模块，从而控制 LED 灯组发光。考虑到氛围灯的灯光应能营造用户所需氛围的功能，LED 灯组的数量选定为 35 个。编码器模块的功能是为了用户能够通过旋转编码器从而控制 LED 灯组的颜色，按键模块的功能是控制 LED 灯组的亮度、切换氛围灯的工作模式以及关灯操作。

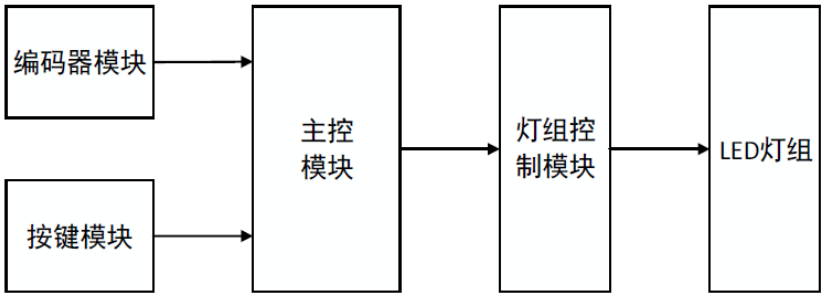


图 2.1 设计框图

2.2 系统方案设计

本设计方案的硬件主要是 STC15 系列单片机、EC11 旋转编码器、自复位按键、自锁按钮、开关、WS2812 芯片和 LED 灯组。系统方案是预先设定 LED 灯组的工作模式，分为全亮模式、功能模式和灭灯模式，通过开关可以切换 LED 灯组的工作模式，其中在全亮模式中用户通过开关 EC11 旋转编码器来切换 LED 灯组的颜色，通过按键调节 LED 灯组的亮度。考虑到营造的氛围应有不同，功能模式还分为七彩灯、呼吸灯以及流水灯，用户通过按键切换功能模式中的任一种氛围灯模式。此外，用户还可以随时通过灭灯按键将所有灯熄灭。为了满足上述功能，以信号传输过程进行此次设计，当用户进行外围操作(旋转编码器或打开/关闭开关或按下按键)，单片机检测到 I/O 状态从而发出一串控制信号给 WS2812 控制芯片，WS2812 接收到来自于单片机的信号后，锁存信号并对其进行整形放大，再传输给 LED 灯组，使其发光。系统方案框图如图 2.2 所示。

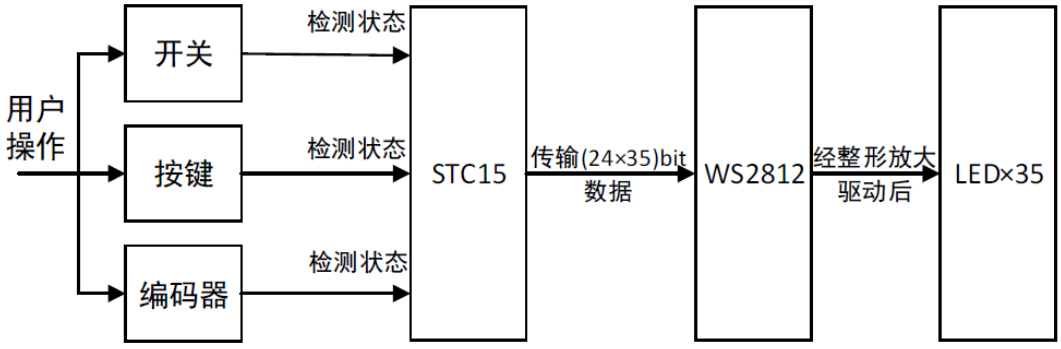


图 2.2 系统方案框图

3 硬件设计

3.1 MCU 选型

作为核心控制的单片机，需要拥有足够的速度、程序存储器容量和 I/O 引脚数量等基本参数。编程语言用 C 语言完成，编程环境选用 KEIL C 开发环境以及多次烧入程序的调试过程，单片机还应含有 FLASH 快擦写存储器，考虑到 WS2812 芯片的严格时序要求，本设计中单片机的内部时钟速度要求高，最好使用单时钟周期（1T）的单片机，因此传统 8051 单片机难以满足设计要求。本设计选用 STC15 系列单片机，可以满足所有要求。由于成本因素的考量，且封装

形式为直插式（DIP）的单片机价格较高，为了既满足使用 DIP 的封装形式又能在满足功能的前提下降低成本，本设计选用芯片价格较低的单片机型号 STC15W4K48S4-PDIP40。

STC15W4K48S430I-PDIP40 是增强型 8051 单片机，具有单时钟/机器周期(1T)，宽范围的电压、可靠性很高、功耗很低和抗干扰能力非常强的特点，与传统 8051 指令代码完全兼容，其内部还集成高精度 R/C 时钟和高可靠复位电路，外部昂贵的晶振和外部复位电路可完全摈弃，工作电压为 2.5V-5.5V，工作频率范围为 5MHz-30MHz，本设计选用 24MHz，无需编程器/仿真器，可直接使用 ISP 烧录软件进行程序烧入，总共有 20 个 I/O 口，每个 I/O 口驱动能力均可达到 20mA。

3.2 电源设计

氛围灯控制系统的供电电源主要是单片机供电和灯组控制模块供电，均可选用 5V 直流电源模块进行。设计思路是使用电源适配器将市电交流（AC）24V 转换为直流（DC）12V，再利用 LDO 将 DC12V 转换为 DC5V。

本设计使用的电源适配器型号为 ECVF+06120-0501，其输入为 AC100V~AC24V，50/60Hz，输出为 DC12V，0.5A 电流。将 DC12V 转换成 DC5V 即为稳压的过程，本设计选用 AMS1117-5.0V 实现稳压功能，其内部结构如图 3.1 所示。ASM1117 的工作原理是通过执行采样输出电压，然后反馈到调节电路去调节输出级调整管的阻抗，在输出电压偏低时调节输出级的阻抗变小从而减小调整管的压降，在输出电压偏高时调节输出级的阻抗变大从而增大调整管的压降，实现输出电压的稳定与维持其稳定性的效果。AMS1117 的稳压调整管是由一个 PNP 管驱动的 NPN 管组成的复合管。片内过热切断电路提供了过载和过热保护，以防环境温度造成过高的结温。本设计选用固定 5.0V 的版本，为了更为可靠的使用 AMS1117，输出需要连接一个与小于 22μF 的去耦电容，提高其稳定性，本

设计在此确定选用 1μF 的 MLCC。

为了提高电源的可用性，为供电电路增加自锁按钮以及 LED 指示灯，只有

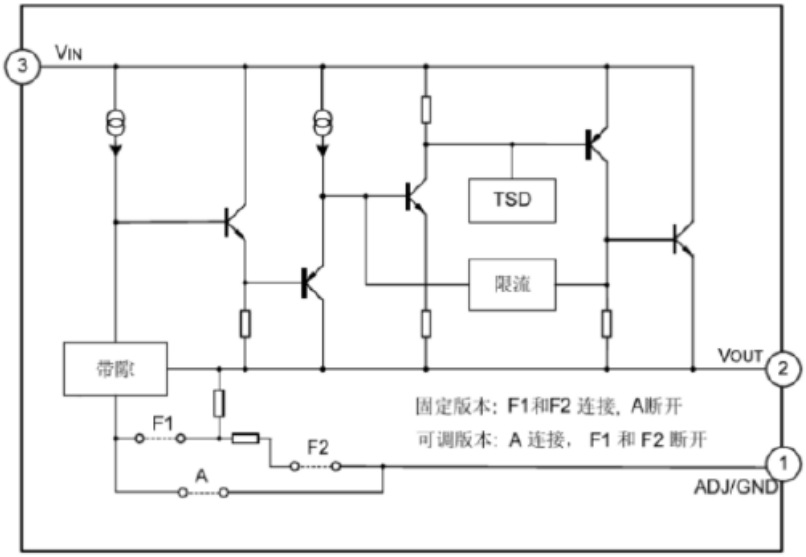


图 3.1 AMS1117 内部结构

按钮按下时才有直流电源 12V 输入，此时 LED 亮，限流电阻选用 2.2KΩ。供电电路如图 3.2 所示，其可以为单片机和灯组控制芯片进行供电。

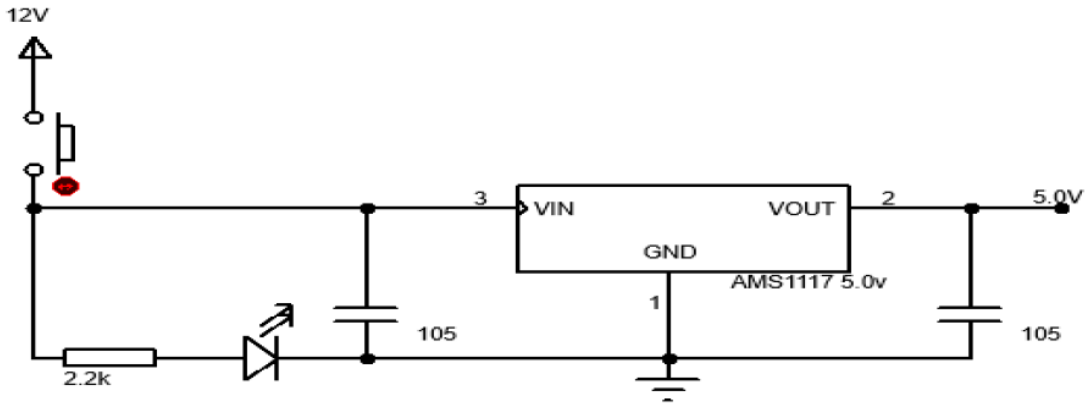


图 3.2 DC5V 供电电路

3.3 灯组级联电路

本设计选用 5050RGBLED 作为氛围灯，由于 WS2812B 已经将控制电路和 RGB 芯片集成在一个 5050 组件的封装中，构成一个完整的外控像素点，因此选用 WS2812B 作为氛围灯灯组与控制模块。

WS2812B 是一款内含智能控制芯片的 RGB LED 灯，每一个 WS2812B 可以控制其内部三个 LED 灯，分别为 Red LED、Green LED、Blue LED。WS2812B

可以进行串联使用，因此只需要一根数据线即可控制。通过主控芯片控制，彩灯组可以显示多种变换效果，例如流水灯、渐变色、彩虹流水等等。LED 具有驱动电压低、环保节能、亮度高、散射角大、一致性好、功耗低、寿命长等优点。WS2812B 内控制电路与 LED 点光源公用一个电源，内置信号整形电路，任何一个像素点收到信号后经过波形整形后输出，保证线路波形畸变不会累加，内置上电复位和掉电复位电路，每个像素点的三基色颜色可实现 256 级亮度显示，完成 16777216 种颜色的全真色彩显示，扫描频率不低于 400Hz/s，串行级联接口，能通过一根信号线完成数据的接收与解码，数据发送速度可达 800kbps。

WS2812B 的电源电压为+3.5~+5.3V，工作温度在-25~+80℃。其有四个管脚，

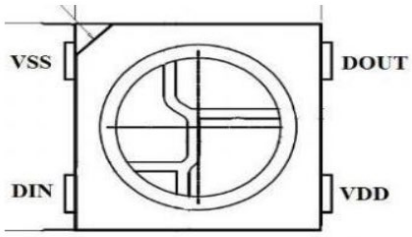


图 3.3 WS2812 管脚图[15]

分别为 VDD 电源端、VSS 接地端、DIN 数据输入端、DOUT 数据输出端，如图 3.3 所示。本设计选用 35 个 WS2812B LED，采用串行级联的方式，仅用单线完成数据传输，如图 3.4 所示。

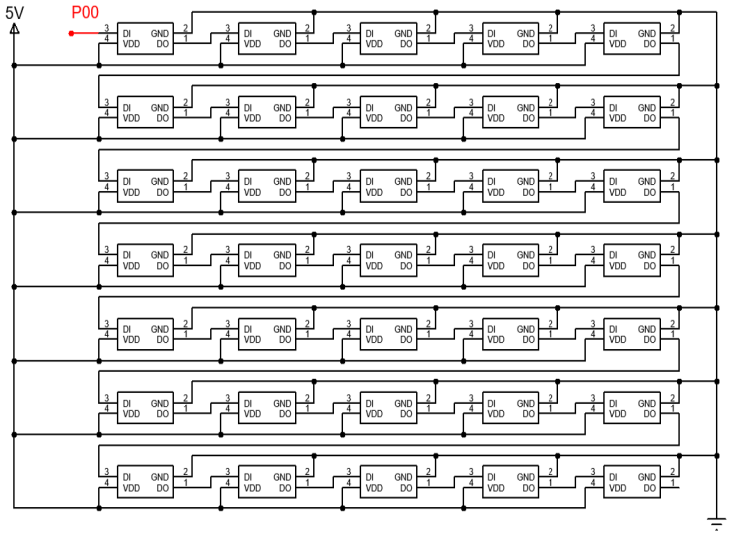


图 3.4 灯组及其控制模块

3.4 MCU 外围电路

3.4.1 按键电路

由于氛围灯需实现开关控制全亮模式与功能模式的切换，一个按键控制功能切换、一个按键控制亮度调节、一个按键控制灭灯模式，因此总共需要 3 个自复位按键与 1 个开关，各自分配一个 I/O 口，如图 3.5 所示。按键的消抖使用软件消抖

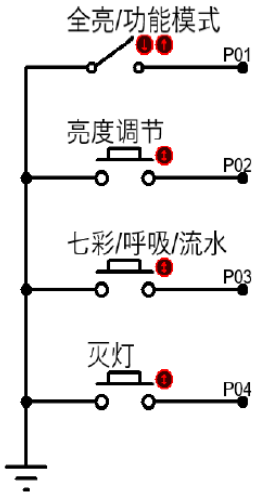


图 3.5 按键电路接线图

3.4.2 旋转编码器电路

选择旋转编码器的原因是：对于灯光的调色，站在用户的角度来看，如果使用按键来调色，当颜色种类数量多时用户的体验感会下降，因此需要旋钮使得用户可以快速切换到所需的颜色，而又由于如果选择普通的挡位旋钮，那么会出现旋钮有转到终点停止，若不旋转到起点则难以继续调色，不利于用户的使用，加上单片机只能接收数字信号，需要使用能够输出数字信号且能当作旋钮使用的器件，因此为了实现很简便地调色以及无限转动不停止的需求，旋转编码器提供了较为合理的作用。本设计采用常见的旋转编码器 EC11，其旋转一圈，能输出 20 个脉冲信号。

EC11 结构上由编码器部分和按键部分(带按键的型号)组合而成，编码器部分有 A，B，C 三个引脚，其中 C 是公共端。按键部分有 D，E 两个引脚属于微

动开关。引脚图如图 3.6 所示。

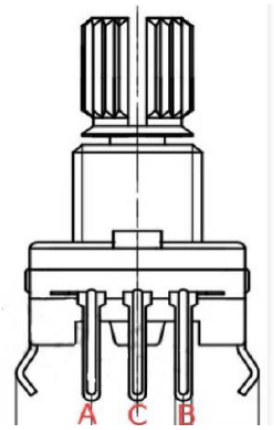


图 3.6 EC11 编码器引脚图

EC11 按旋转的输出动作可以分为两种。一种是转两格，A、B 对 C 端输出一个完整脉冲（转一格指由低电平跳到高电平或由高电平跳到低电平）；另一种就是转一格，A、B 对 C 端输出一个完整脉冲。本设计采取的是后者，当编码器输出一个脉冲的时候，A 相信号与 B 相信号如图 3.7 所示，若 A 信号的下降沿处 B 信号为高电平，则编码器旋转方向为顺时针，若 A 信号的下降沿处 B 信号为低电平，则编码器旋转方向为逆时针。

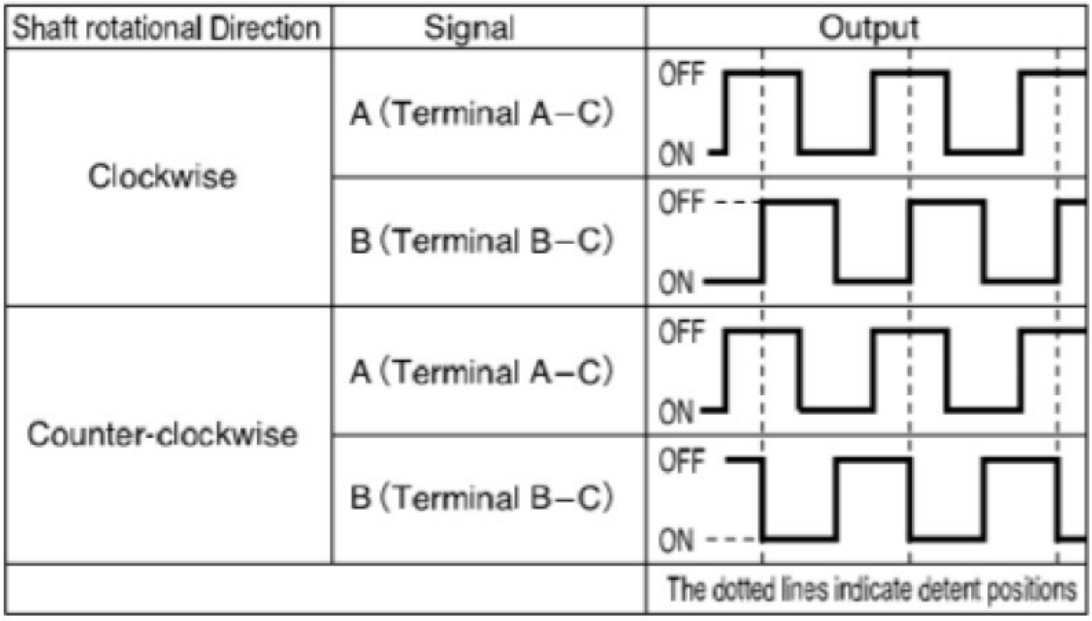


图 3.7 EC11 转向判断[16]

为了满足转一格切换一次颜色，本设计采用外部中断的方式来接收 EC11 编码器的信号 A，作为检测信号的 B 信号直接用单片机 I/O 口相连。此外，A，B

引脚需上拉 10KΩ 电阻再与电源相连。C 引脚属于公共端，与地相连。EC11 编码器电路接线图如图 3.8 所示。需要注意的是，编码器是通过机械接触的通断来输出脉冲，那便存在接触不良的问题，使得干扰信号必然存在。编码器在长时间使用后，可能出现接触点氧化等问题时，干扰信号的影响愈严重。最终结果是编码器旋转一格，结果输出两个或者多个脉冲，使得数据传输不准确，所以使用 EC11 编码器时需要排除干扰，本设计选用软件排除，即判断两个脉冲的间隔时间，若间隔时间小于 5ms，则认定为干扰信号，单片机不作为，即排除了干扰。

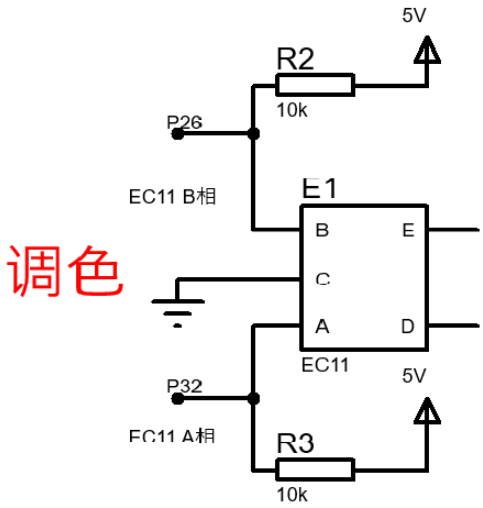


图 3.8 EC11 编码器电路接线图

4 软件设计

4.1 设计思路

本设计采取模块化的方式进行软件设计，利用 KEIL C 开发环境，使用 C 语言进行编程，分别将程序设计划分为：初始化程序设计、主程序设计、按键程序设计、灯组传输程序设计、灯组控制程序设计和延时程序设计。先依次将各个程序模块设计出后，再通过主程序将其整合并实现所需功能。整体的思路为单片机上电初始化，检测到相应 I/O 口的状态后传输数据给 WS2812B，使其实现调色调光的功能。重点出现在传输给 LED 灯组的数据时序，根据 WS2812B 数据协议来控制不同的时序即可对 R、G、B 值作赋值行为，其中 R、G、B 值的十进制数值范围为 0~255，控制 RGB 三基色的值的不同便能出现不同的颜色，即 RGB 比

例不同，则颜色不同。在 R、G、B 相应比例不变的情况下，距数值 255 越近，灯光则越亮，反之据数值 0 越近，灯光则越暗直至灭灯，可以看出灯组其实有 256 种亮度级别，但由于 256 级亮度中单位变化的亮度难以被人眼辨别，因此划分为四个亮度挡位，更好地体现调光的功能需求。下面分别介绍各模块程序的设计思路与方法。

4.2 主程序流程图

主程序流程如图 4.1 所示。按照流程图，需要执行以下几个流程：

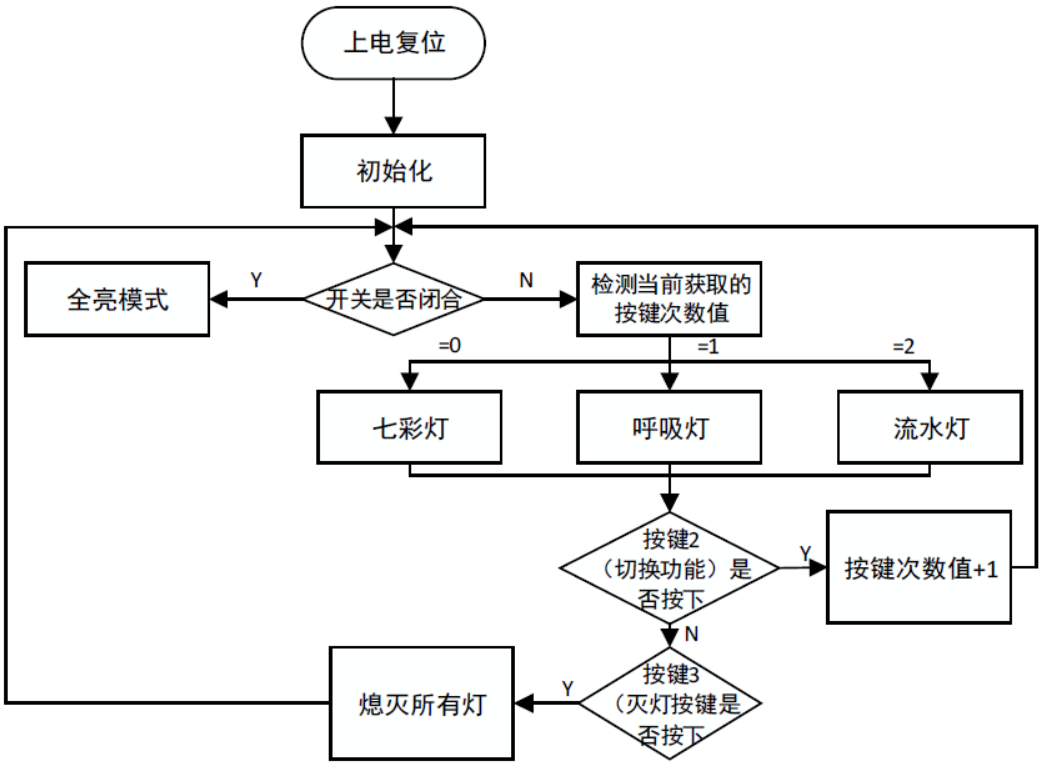


图 4.1 主程序流程图

（1）初始化

在硬件系统加电后准备开始执行软件程序前，必须先对系统内所有芯片进行系统初始化设置以解决系统内部各个硬件可能存在的数据不确定问题。

（2）按键模块

开关用来判断氛围灯是工作在全亮模式还是功能模式。若为功能模式则需要检测状态参数从而确定该工作在哪一种功能状态，随之继续判断功能按键是否被

按下，即切换功能状态。在整个流程中，均需检测是否按下关灯按键，按下后所有灯都将熄灭。

（3）编码器模块

在全亮模式下，随着编码器转动，检测两个脉冲是否小于 5ms，若小于则视为干扰单片机不作为，若不小于 5ms，则进入外部中断，在中断程序中，按预先设定好的灯组传输数据中 RGB 三基色的值，从而切换颜色。

（4）灯组传输与控制模块

灯组无论工作在全亮模式还是功能模式，单片机都一直在调用灯组传输函数。

4.3 初始化程序

初始化程序包含设置 I/O 口模式、打开总中断允许、打开外部中断 0、外部中断 0 设置为下降沿触发、定时器 1 初始化、设置灯组数据传输端为低电平、数据传输复位、全亮模式颜色从颜色表第一个开始、所有灯亮白色。程序流程图如 4.2 所示。

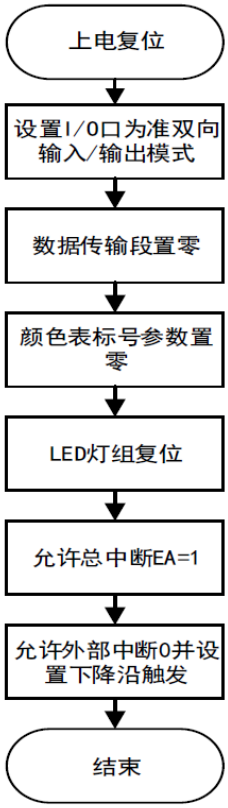


图 4.2 初始化程序流程图

需注意的是，STC15 系列 I/O 口具有 4 种工作模式，在使用 I/O 口前需要设置好工作模式后才能正常使用，本设计将所有使用到的 I/O 口均设置为准双向模式。

4.4 延时程序设计

本设定选用单片机内部时钟 24MHz，1T 模式，即一个时钟周期为一个机器周期，不分频。那么一个空指令操作_nop_()的时间为

$$t_{nop} = \frac{1}{24 \times 10^6} = 0.04167us$$

对于延时程序的使用场合，分别有按键消抖需要延时 10ms，编码器旋转程序为了排除干扰也需要延时 5ms，单片机给灯组一帧数据后需要延时大于 50μs 以进行灯组复位。由此本设计需要三个延时程序，分别是 10ms 延时、5ms 延时、30μs 延时，延时程序均采用软件延时。

4.5 按键程序设计

4.5.1 单次按键值获取

此程序是为了判断功能模式和灭灯模式，思路是如果功能模式按键被按下，则返回按键状态参数 1；若灭灯按键被按下，则返回按键状态参数 2；若两个按键都没按下，则返回按键状态参数 0。返回的按键状态参数是为了判断氛围灯所处工作状态以及指明下一个工作状态。流程图如图 4.3 所示。

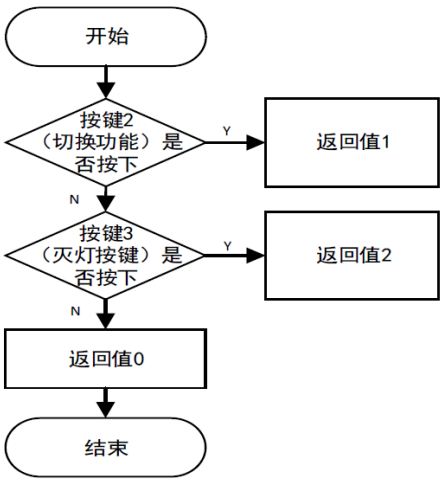


图 4.3 获取按键值

4.5.2 定时器初始化以及中断程序

在主程序中需要一直检测功能模式按键或者灭灯按键是否按下，本设计选用 STC15 系列单片机内部定时器 1 以每 2ms 检测一次的频率进行采样，并在中断程序中记录当前按键值，上一次按键值，按键次数。程序流程图如图 4.4 所示。

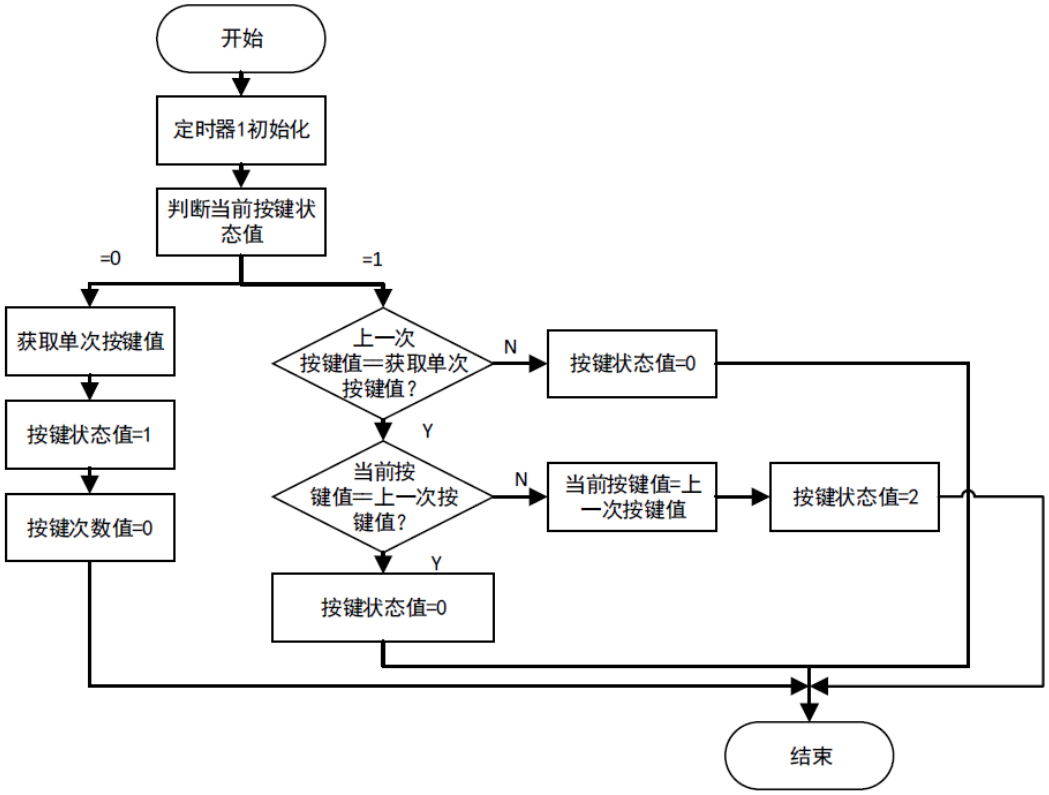


图 4.4 定时器中断程序

定时器初始化包含设定定时器时钟模式为 1T，不分频，设置定时器模式，设置定时初值，清除标志位，打开定时器 1 中断，运行定时器 1。定时器初值计算过程如下：

由于定时器晶振为 24MHz,，且时钟模式为 1T，不分频，一个机器周期为一个时钟周期。所以

$$1s = 24 \times 10^6 \text{个机器周期}$$

则若需定时 2ms，则需要

$$2ms = 48000 \text{ 个机器周期}$$

所以定时器初值为

$$65536 - 48000 = 17536$$

转换成十六进制数后，定时器 1 的高八位与低八位初值分别为

$$TH1 = 0x44$$

$$TL1 = 0x80$$

4.6 灯组传输与控制程序设计

4.6.1 WS2812B 数据协议与驱动方式

WS2812B 数据协议采用单线归零码的通讯方式，像素点在上电复位以后，DIN 端接受从单片机传输过来的数据，由于芯片内部的锁存器和整形电路，数据每经过一级 LED 灯，就会相应的减少 24 位，这样的逐级递减从而达到每个灯都能够有数据的传递过来[18]。像素点采用自动整形转发技术，使得该像素点的级联个数不受信号传送的限制，仅仅受限信号传输速度要求。

WS2812B 级联采用串行的方式，如图 4.5 所示，即单片机传输给灯组的数据必须遵守 WS2812B 的数据协议才能实现氛围灯的控制。

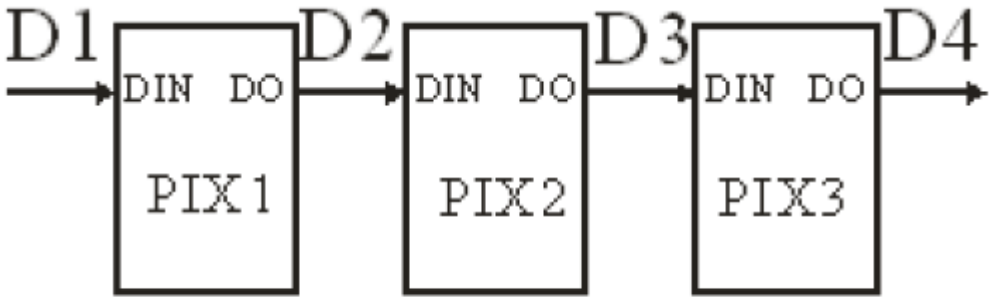


图 4.5 串行级联[15]

因为 WS2812B 的发光源本质上是 RGB 三种颜色的 LED 灯 即调光调色是通过三基色混光原理实现的。WS2812B 数据协议包含判断写 1 码和判断写 0 码的时序协议，首先 WS2812B 中 R、G、B 各自为 8bit 数据，那么 1 个 WS2812B 需要给 24bit 数据，其数据传输结构如表 4.1 所示。由表 4.1 可知，数据需遵循 G、R、B 的顺序进行传输，且高位作为优先位进行数据发送。

表 4.1 数据传输结构[15]

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

其次由表 4.2，图 4.6 可看出，写 1 码和写 0 码的高低电平持续时间基本相反，则可使用同样的延时时间，以方便程序编写。此时序表明 WS2812B 数据协议的严格性，其高低电平持续时间精确到纳秒级别，因此本设计不使用定时器延时或编写延时函数的方法，而是采用空操作指令_nop_()来精确控制高低电平的持续时间，值得注意的是每次发完一帧数据就应复位（大于 50μs 的低电平时间），然而在下一章调试时，复位时间的位置一定要正确，否则数据传输不会进入第二个像素点，表现的现象是仅第一个像素点发光，其余不发光。

（TH+TL=1.25μs±600ns）

表 4.2 数据传输时间

类别	描述	要求持续时间
T0H	写0码的高电平持续时间	0.25~0.55μs
T1H	写1码的高电平持续时间	0.65~0.95 μs
T0L	写0码的低电平持续时间	0.7~1.0μs
T1L	写1码的低电平持续时间	0.3~0.6 μs
RES	一帧数据后的低电平复位时间	50μs以上

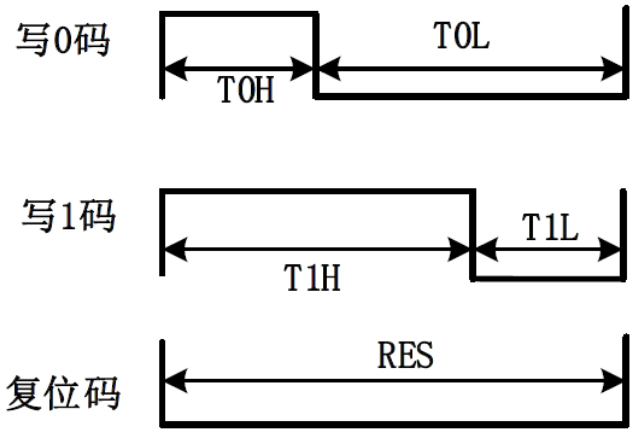


图 4.6 时序输入波形图[15]

最后，如图 4.7 所示，当第一个像素点得到第一个 24bit 数据后，将剩下的数据继续传输给下一个像素点，直至传到最后一个像素点，在发送完一帧数据（例如 35×24bit=840bit）后，进行复位。

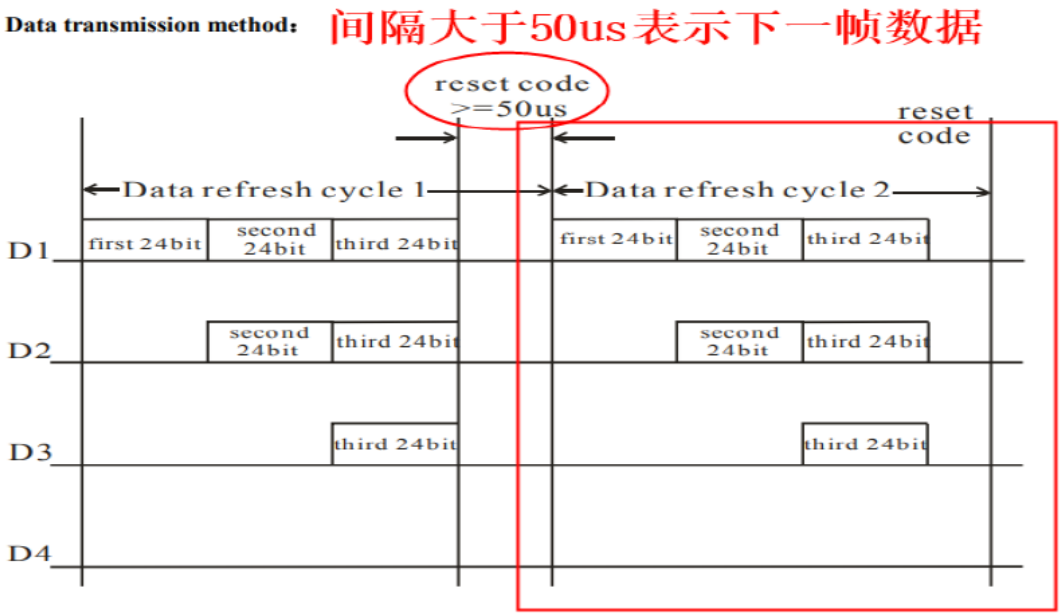


图 4.7 数据传输方法

4.6.2 WS2812B 传输程序

本设计将传输程序细分为 8bit 传输，RGB 取值，RGB 传输、复位模式四个细分结构。

（1）8bit 传输程序是对于单个 R、B、G 值的写 1 码和写 0 码，系统时钟为 24MHz，则写 1 码的时序_nop_()个数 NUM1H、NUM1L 应为

$$NUM1H = 0.8 \div t_{nop} \approx 20$$

$$NUM1L = 0.4 \div t_{nop} = 10$$

写 0 码的时序_nop_()个数 NUM0H、NUM0L 应为

$$NUM0H = 0.45 \div t_{nop} \approx 11$$

$$NUM0L = 0.85 \div t_{nop} \approx 21$$

因此只需在高低电平的延时时间给到相应数量的 NOP 指令即可满足传输时序。程序设计框图如 4.8 所示。

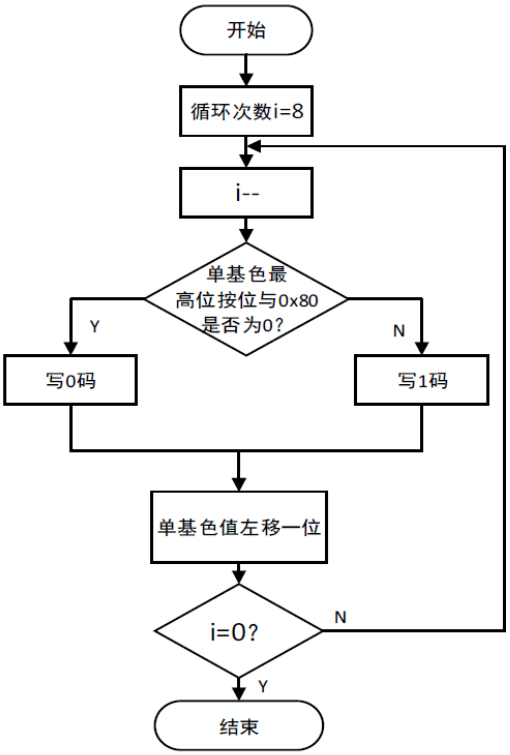


图 4.8 8bit 传输程序框图

（2）RGB 取值的本质是从一个十六进制数即 RGB 中取高二位 R，中间二位 G，低二位 B，便于调色调光的基色值改变。实现方法是按位与再赋值。程序流程图如 4.9 所示。

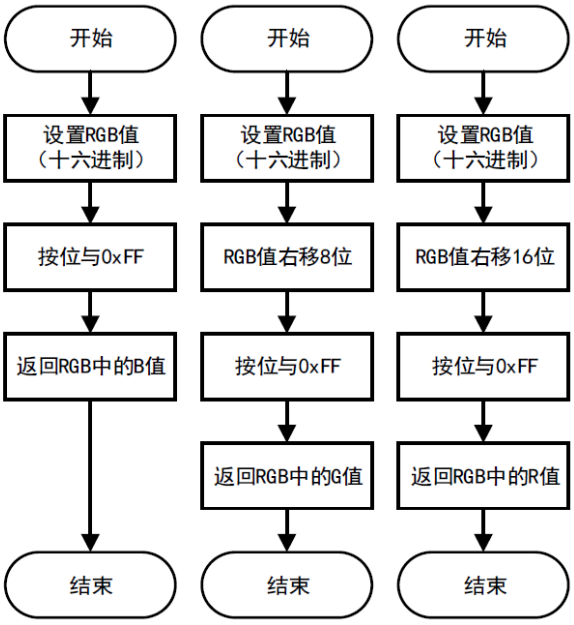


图 4.9 RGB 取值

（3）RGB 传输函数是按照 GRB 的顺序来调用 8bit 传输程序，循环三次，程序流程图如 4.10 所示。

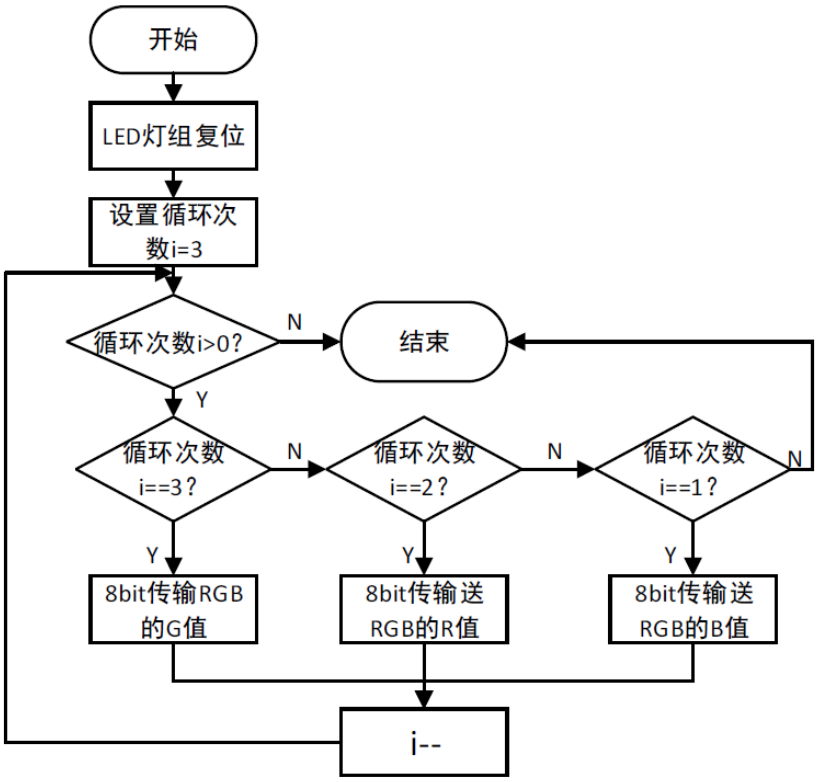


图 4.10 传输 RGB24bit

（4）复位模式即让数据传输端口保持低电平 50μs，程序设计即可直接令端口为低电平并延时大于 50μs 即可，程序流程图如 4.11 所示。

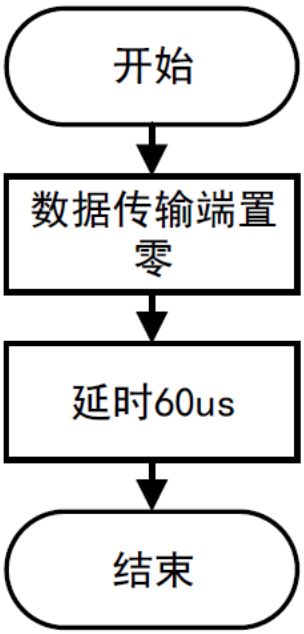


图 4.11 灯组复位

4.6.3 WS2812B 控制程序

WS2812B 控制程序细分为颜色布置、全亮控制、七彩控制、呼吸灯控制、流水灯控制、调色程序、调光程序。

（1）颜色布置

设置一个颜色表数组，便于后续准确地调色控制[22]。

（2）全亮控制

使用循环函数，循环 35 次（35 个像素点），循环中执行 RGB 传输函数。程序流程图如 4.12 所示。

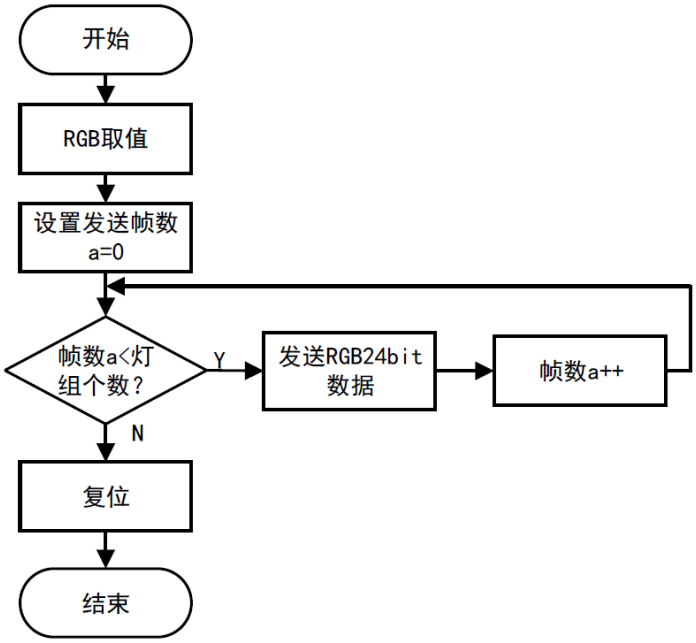


图 4.12 全亮模式

（3）七彩灯控制

七彩灯实际是显示出红、橙、黄、绿、青、蓝、紫七种颜色，按彩虹般布置，即每连续 5 个灯亮一种颜色，程序上只需要根据对应 RGB 值写多次循环函数即可。

（4）呼吸灯控制

呼吸灯是指灯光缓慢变暗缓慢变亮，似乎灯光在呼吸，故得此名。本设计欲实现呼吸灯呼吸一次，变色一次，其中包含了调色调光的方法。首先进入函数先

设定亮度增减量，判断目前亮度情况，先让所有灯亮度调至最低，之后开始呼吸模式，缓慢变亮后再缓慢变暗，属于一次呼吸，呼吸速率由延时程序所表示。判断亮度情况的方法是将当前 RGB 值分别与色温设定值作逻辑与运算，根据逻辑计算结果选择变暗或是变亮。程序设计框图如图 4.13 所示。

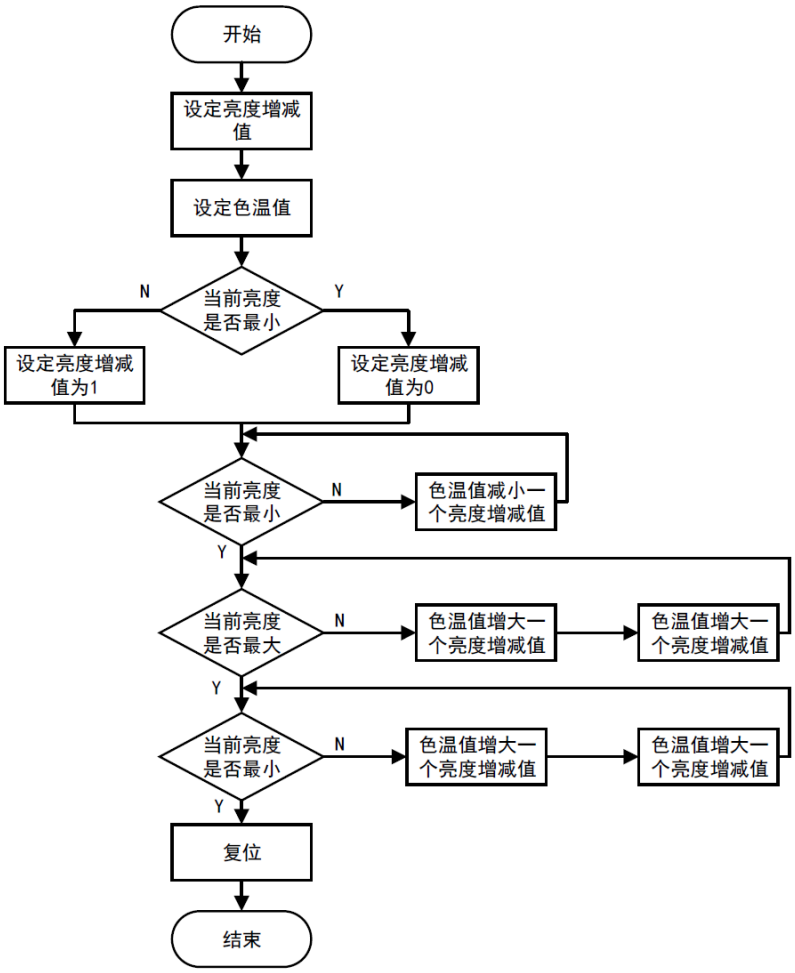


图 4.13 呼吸灯程序流程图

（5）流水灯控制

流水灯是指灯光按布置的依次顺序和个数进行亮与暗。从外观上看灯光行如流水，实际上程序的设计即是每次亮灯预设的个数，但不同的是起始亮灯序数逐次递增，这样在短时间（微秒级别）人眼观察到的结果便是发光灯如流水般移动。具体程序的编写只需通过循环函数以及传输相应 RGB 值即可。考虑到在氛围灯流水一轮后灯光进行颜色改变以提高趣味性，则通过 RGB 值数据逐轮改变的方式即可实现。流水灯程序流程图如图 4.14 所示。

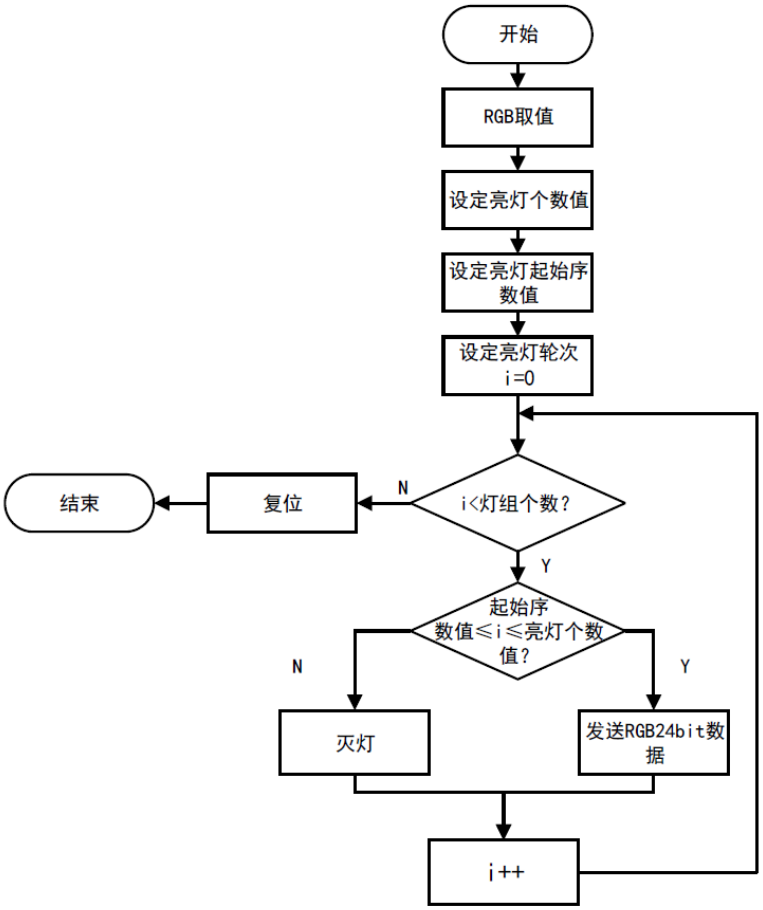


图 4.14 流水灯程序流程图

（6）调色程序

此调色程序主要针对全亮模式下的任意调色，外设使用 EC11 旋转编码器，通过旋转 EC11，进入外部中断 0，中断程序中改变颜色布置表数组中的下标，使得选取不同的颜色。程序流程图如 4.15 所示。

（7）调光程序

此调光程序旨在实现四个挡位的亮度切换，包含最亮，较亮，较暗，最暗四种亮度。设计思路为在全亮模式下当切换亮度挡位的按键按下后，先读取当前 RGB 值，是否为最亮或最暗，若 RGB 值反映氛围灯工作在最亮状态，则 RGB 值均减少原来的 1/4，若 RGB 值反映氛围灯工作在最暗状态，则 RGB 值均增加原来的 3/4，若 RGB 值反映氛围灯工作并非处于最亮/最暗状态，则 RGB 值均减少原来的 1/4。RGB 值改变后数据传输循环 35 次，即可得到所需亮度。程序

流程图如 4.16 所示。

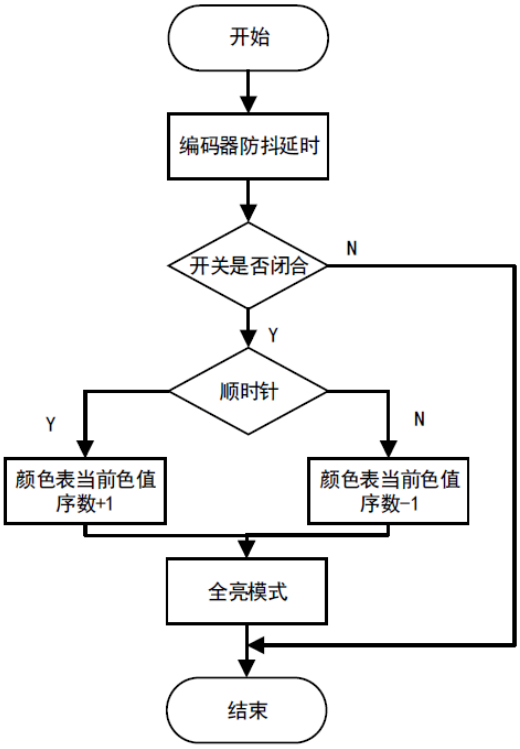


图 4.15 调色

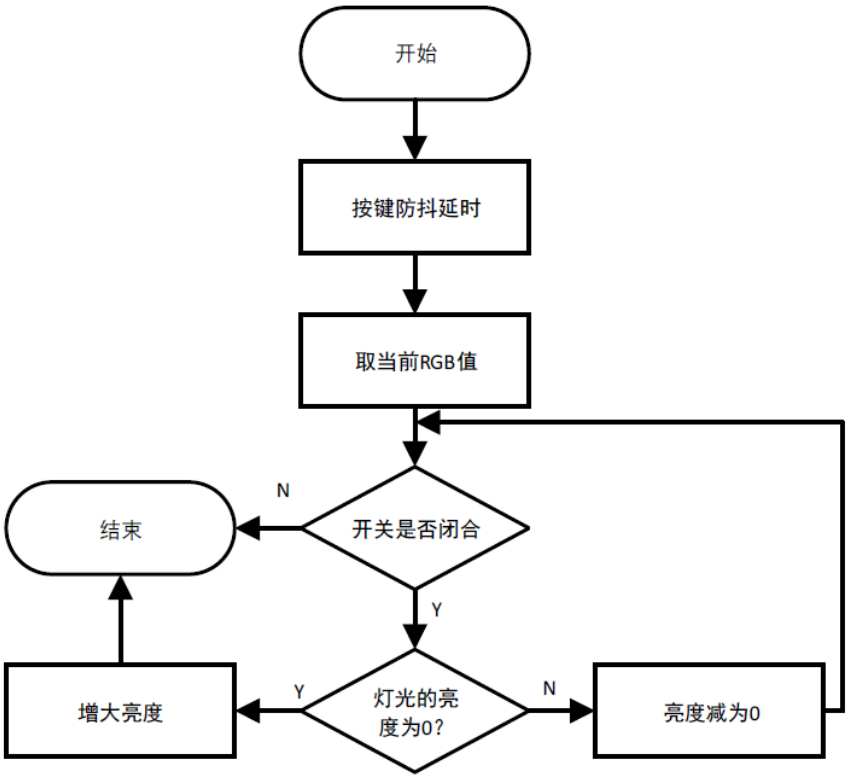
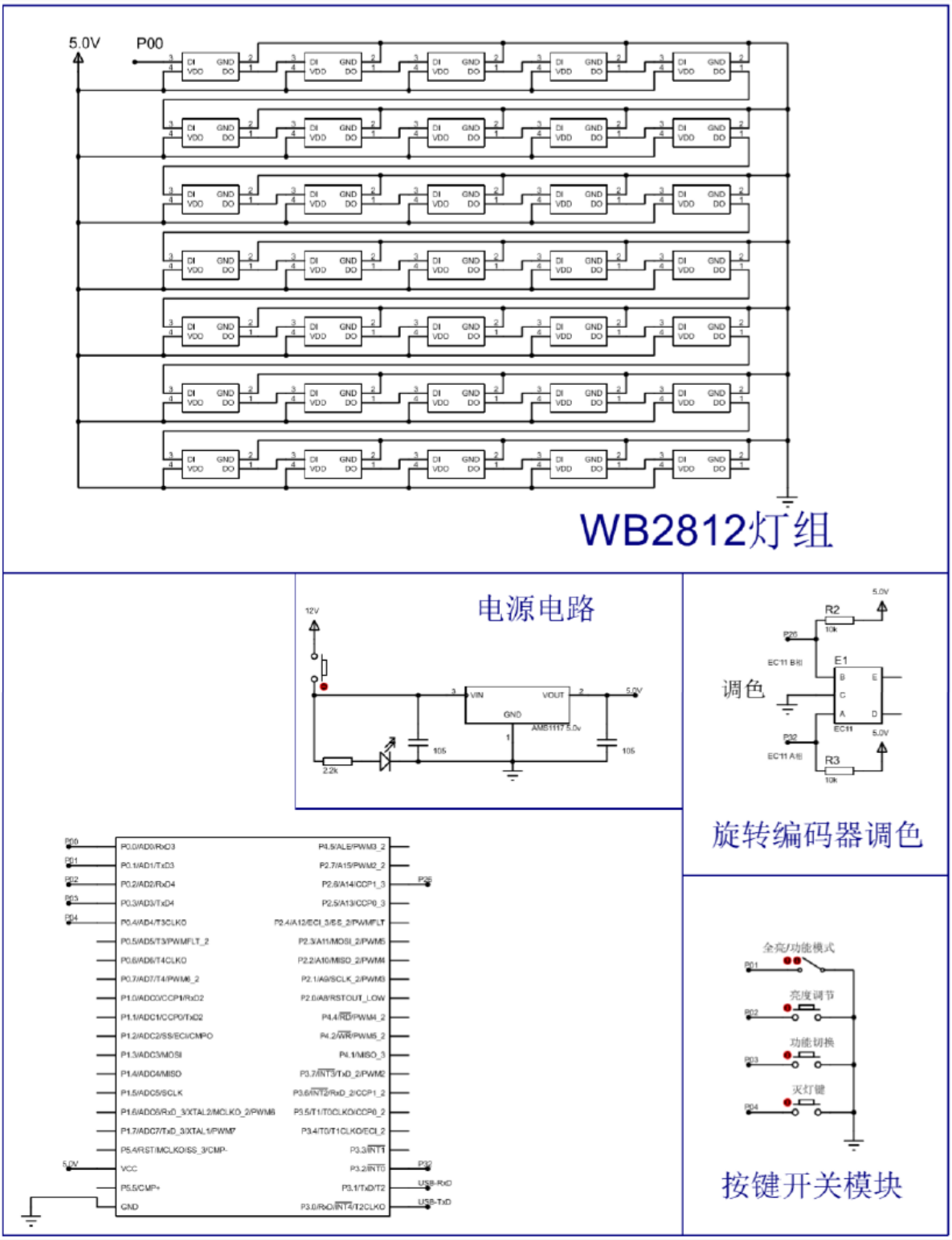


图 4.16 调光

5 原理图


5.0V
P00
WB2812灯组
电源电路
12V
VIN
VOUT
GND
AMS1117 5.0v
5.0V
105
2.2k
调色
R2
10k
P26
EC11 B相
E1
EC11
P32
EC11 A相
R3
旋转编码器调色
全亮/功能模式
P01
亮度调节
P02
功能切换
P03
灭灯键
P04
按键开关模块
P00
P01
P02
P03
P04
P0.0/AD0/RxD3
P0.1/AD1/TxD3
P0.2/AD2/RxD4
P0.3/AD3/TxD4
P0.4/AD4/T3CLKO
P0.5/AD5/T3/PWMFLT_2
P0.6/AD6/T4CLKO
P0.7/AD7/T4/PWM6_2
P1.0/ADC0/CCP1/RxD2
P1.1/ADC1/CCP0/TxD2
P1.2/ADC2/SS/ECI/CMPO
P1.3/ADC3/MOSI
P1.4/ADC4/MISO
P1.5/ADC5/SCLK
P1.6/ADC6/RxD_3/XTAL2/MCLKO_2/PWM6
P1.7/ADC7/TxD_3/XTAL1/PWM7
P5.4/RST/MCLKO/SS_3/CMP-
VCC
P5.5/CMP+
GND
P4.5/ALE/PWM3_2
P2.7/A15/PWM2_2
P2.6/A14/CCP1_3
P2.5/A13/CCP0_3
P2.4/A12/ECI_3/SS_2/PWMFLT
P2.3/A11/MOSI_2/PWM5
P2.2/A10/MISO_2/PWM4
P2.1/A9/SCLK_2/PWM3
P2.0/A8/RSTOUT_LOW
P4.4/RD/PWM4_2
P4.2/WR/PWM5_2
P4.1/MISO_3
P3.7/INT3/TxD_2/PWM2
P3.6/INT2/RxD_2/CCP1_2
P5.5/T1/T0CLKO/CCP0_2
P3.4/T0/T1CLKO/ECI_2
P3.3/INT1
P3.2/INT0
P3.1/TxD/T2
P3.0/RxD/INT4/T2CLKO
P25
P32
USB-RxD
USB-TxD
5.0V