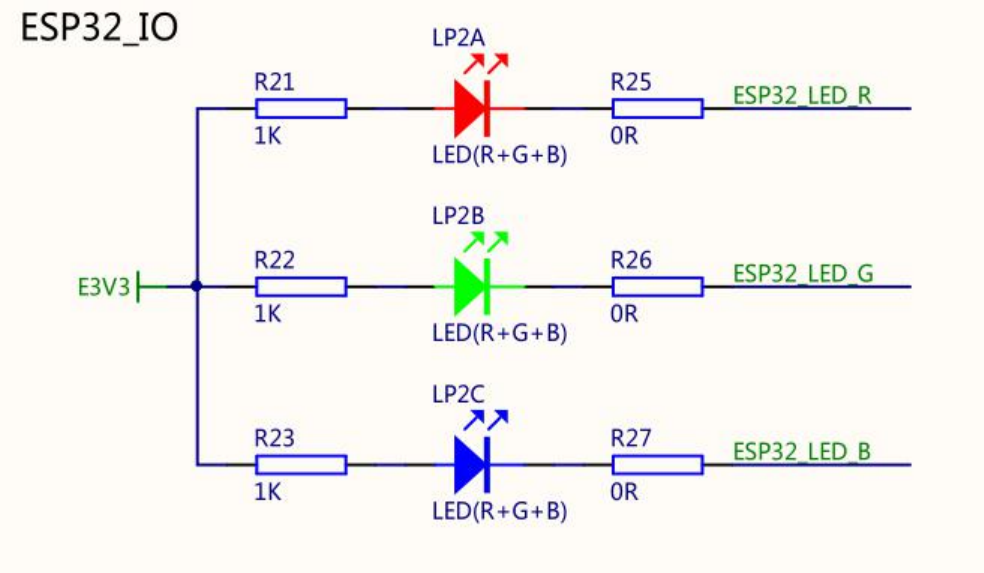
第一章 驱动 LED 灯

1. 学习目的及目标

- 掌握 LED 灯电路设计：控制方式
- 掌握 ESP32 库函数对 IO 配置的相关参数设置
- 掌握 ESP32 库函数对 IO 控制的操作
- 编写 LED 闪烁和流水灯程序

2. 硬件设计及原理

本实验板连接了一个 RGB 彩灯，RGB 彩灯实际上由三盏分别为红色、绿色、蓝色的 LED 灯组成，通过控制 RGB 颜色强度的组合，可以混合出各种色彩，此章只学习如何开关，调色放在 PWM 章学习。



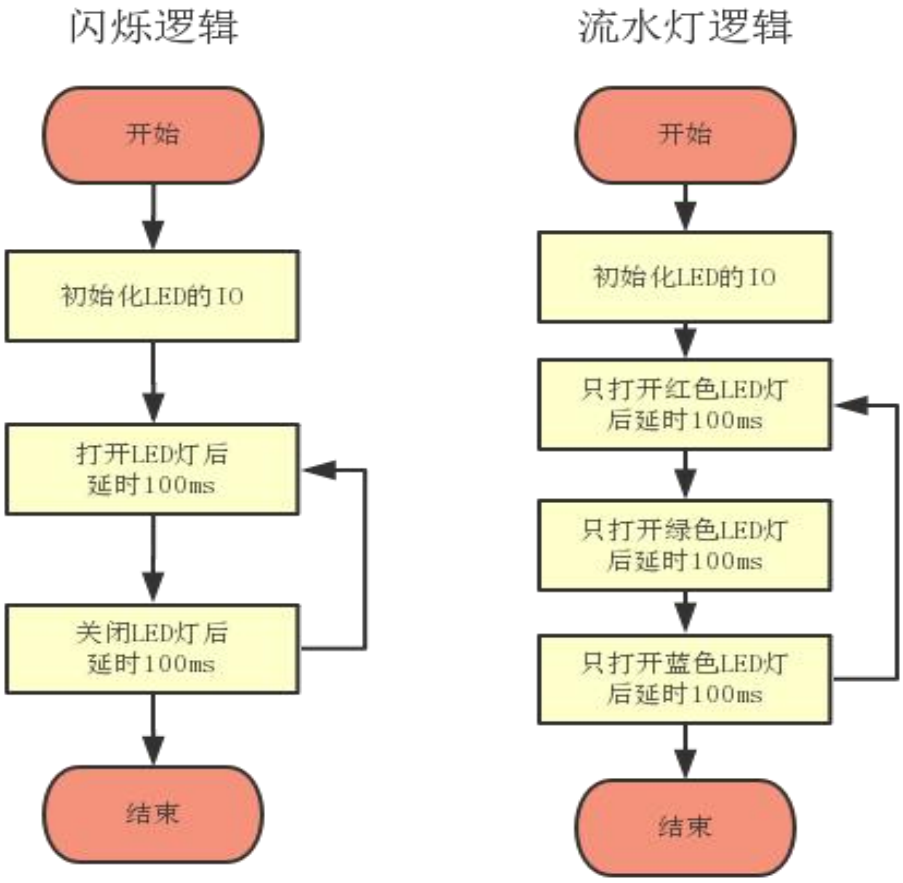
这些 LED 灯的阴极都是通过 0 欧姆电阻连接到 ESP32 的 GPIO 引脚，只要我们控制 GPIO 引脚的电平输出状态，即可控制 LED 灯的亮灭。图中去掉 0 欧姆电阻，可以切断和单片机的连接，释放这个 GPIO。3 个 LED 灯占用 ESP32 的引脚如下：

LED 标号	LED 颜色	接至 ESP32 的引脚
LP2A	红色	IO2
LP2B	绿色	IO18
LP2C	蓝色	IO19

若您使用的实验板 LED 灯的连接方式或引脚不一样，只需根据我们的工程修改引脚即可，程序的控制原理相同。

3. 软件设计

3.1. 代码逻辑



3. 2. ESP32 的 GPIO 接口介绍

➢ 设置 IO 模式函数：gpio_set_pull_mode();

函数原型	esp_err_t gpio_set_pull_mode (gpio_num_t gpio_num, gpio_pull_mode_t pull)
函数功能	设置 IO 模式
参数	[in]gpio_num:引脚编号，0~34（存在部分） [in]pull:IO 模式，可以设置： GPIO_MODE_DISABLE：不输入不输出 GPIO_MODE_INPUT：输入模式 GPIO_MODE_OUTPUT：输出模式 GPIO_MODE_OUTPUT_OD：开漏输出模式 GPIO_MODE_INPUT_OUTPUT_OD：开漏输入输出模式 GPIO_MODE_INPUT_OUTPUT：输入输出模式
返回值	ESP_OK：成功 ESP_ERR_INVALID_ARG ： 参数错误

➢ 设置 IO 输出值函数：gpio_set_level();

函数原型	esp_err_t gpio_set_level (gpio_num_t gpio_num, uint32_t level)
函数功能	设置 IO 输出值
参数	[in]gpio_num:引脚编号，0~34（存在部分） [in]pull:IO 模式，可以设置： 0：输出低 1：输出高
返回值	ESP_OK：成功 ESP_ERR_INVALID_ARG ： 参数错误

更多更详细接口请参考官方指南。

3.3. 代码编写

➢ 闪灯代码

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/gpio.h"
#include "sdkconfig.h"
#define LED_R_IO          2
#define LED_G_IO          18
#define LED_B_IO          19
void app_main()
{
    //选择 IO
    gpio_pad_select_gpio(LED_R_IO);
    //设置 IO 为输出
    gpio_set_direction(LED_R_IO, GPIO_MODE_OUTPUT);
    while(1) {
        //红灯亮
        gpio_set_level(LED_R_IO, 0);
        vTaskDelay(100 / portTICK_PERIOD_MS);
        //红灯灭
        gpio_set_level(LED_R_IO, 1);
        vTaskDelay(100 / portTICK_PERIOD_MS);
    }
}
```

28

➢ 流水灯的源码

```
void app_main()
{
 //选择 IO
   gpio_pad_select_gpio(LED_R_IO);
   gpio_pad_select_gpio(LED_G_IO);
   gpio_pad_select_gpio(LED_B_IO);
   //设置 IO 为输出
   gpio_set_direction(LED_R_IO, GPIO_MODE_OUTPUT);
   gpio_set_direction(LED_G_IO, GPIO_MODE_OUTPUT);
   gpio_set_direction(LED_B_IO, GPIO_MODE_OUTPUT);
   while(1) {
       //只点亮红灯
       gpio_set_level(LED_R_IO, 0);
       gpio_set_level(LED_G_IO, 1);
       gpio_set_level(LED_B_IO, 1);
       vTaskDelay(100 / portTICK_PERIOD_MS);
       //只点亮绿灯
       gpio_set_level(LED_R_IO, 1);
       gpio_set_level(LED_G_IO, 0);
       gpio_set_level(LED_B_IO, 1);
       vTaskDelay(100 / portTICK_PERIOD_MS);
       //只点亮蓝灯
       gpio_set_level(LED_R_IO, 1);
       gpio_set_level(LED_G_IO, 1);
       gpio_set_level(LED_B_IO, 0);
       vTaskDelay(100 / portTICK_PERIOD_MS);
   }
}
```

3.4. 硬件连接

红旭开发板默认已经连接好 LED，下载程序即可，使用其他开发板需要修改程序或者修改硬件连接皆可。

3.5. 效果展示

➢ 闪灯：红灯 100ms 取反一次

➢ 流水灯：红绿蓝 100ms 循环点亮

4. 驱动 LED 灯总结

上面的方法是使用库函数配置 GPIO，也可以使用结构体进行配置，先设置结构体参数，再使用结构体配置 IO，实现功能一样。

源码地址：https://github.com/xiaolongba/wireless-tech