

彩灯模块

1. 模块概述

彩灯模块是一种可编程控制的 LED 模块，可显示 2^{24} 种颜色，像素点内部包含了智能数字接口数据锁存信号整形放大驱动电路，还包含有高精度的内部振荡器和可编程定电流控制部分，有效保证了像素点光的颜色高度一致。280 μ s 以上的 RESET 时间，出现中断也不会引起误复位，可以支持更低频率、价格便宜的 MCU，因此广泛应用于嵌入式开发与教学实验中，同时还可应用于电子产品领域、LED 灯饰亮化领域和各种电器设备领域等。

2. 硬件介绍

2.1 引脚功能



| 引脚名称 | 功能说明 |
|------|-------|
| 5V | 5V 供电 |
| DIN | GPIO |
| DNP | 浮空 |
| GND | 接地 |

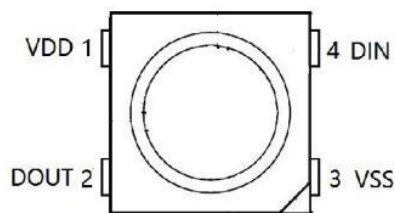
2.2 详细参数

| |
|--------------------------------|
| 工作电压：5V |
| 工作环境温度：-40℃ ~ +65℃ |
| LED 特性参数：工作电流 12mA（测试条件:DC=5V） |
| 定位孔：4*M3 |

2.3 WS2812 介绍

彩灯结构如下图所示

引出端排列



引脚功能

| 序号 | 符号 | 管脚名 | 功能描述 |
|----|------|------|-----------|
| 1 | VDD | 电源 | 供电管脚 |
| 2 | DOUT | 数据输出 | 控制数据信号输出 |
| 3 | VSS | 地 | 信号接地和电源接地 |
| 4 | DIN | 数据输入 | 控制数据信号输入 |

2.4 时序结构

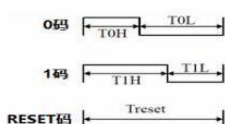
WS2812 的数据协议采用单线归零码的通讯方式，像素点在上电复位以后，

DIN 端接受从控制器传输过来的数据，首先送过来的 24bit 数据被第一个像素点提取后，送到像素点内部的数据锁存器，剩余的数据经过内部整形处理电路整形放大后通过 DO 端口开始转发输出给下一个级联的像素点，每经过一个像素点的传输，信号减少 24bit，以此类推，实现单总线控制多个小灯。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，仅受限信号传输速度要求。

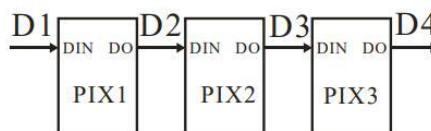
通讯时序如下：（注：不同厂家不同具体型号的 WS2812 时序可能不同，具体时序标准以使用的 WS2812 硬件手册为准）

时序波形图

输入码型:



连接方法:



数据传输时间

| | | |
|-----|------------|------------------|
| TOH | 0 码, 高电平时间 | 220ns~380ns |
| T1H | 1 码, 高电平时间 | 580ns~1μs |
| TOL | 0 码, 低电平时间 | 580ns~1μs |
| T1L | 1 码, 低电平时间 | 580ns~1μs |
| RES | 帧单位, 低电平时间 | 280μs 以上 |

一个 LED 灯需要 24bit 的颜色数据来控制，分别为 8 位绿色+8 位红色+8 位蓝色，高位在前，每个颜色可实现 256 级亮度显示，通过组合实现显示 2^{24} 种颜色。

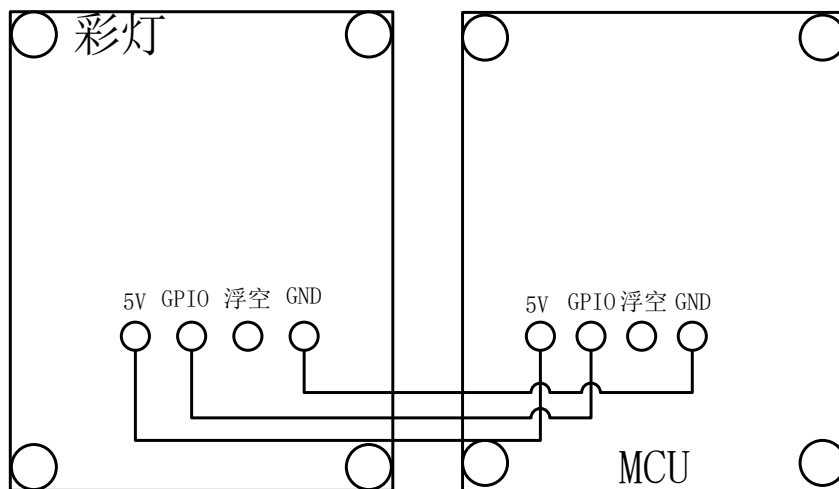
24bit 数据结构

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

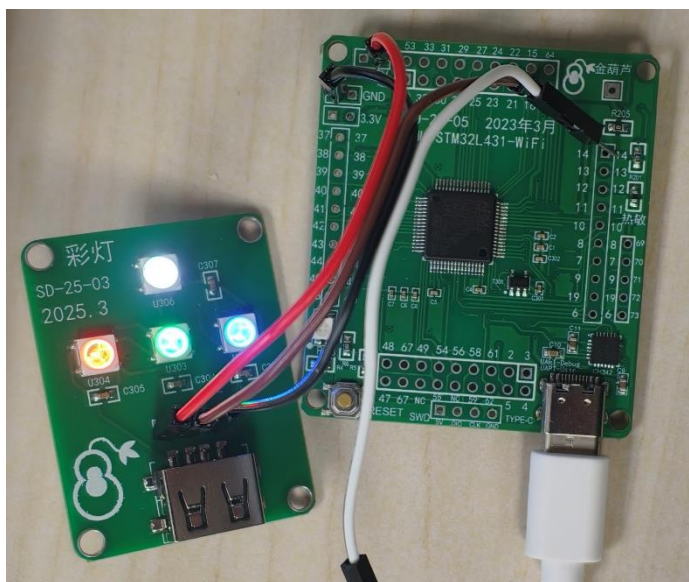
注：高位先发，按照 GRB 的顺序发送数据。

3. 硬件连接

原理连接图



实物连接图



红线：5V

黑线：GND

棕线：GPIO-LED (默认程序连接 PA6, GEC16)

4. 驱动程序功能

4.1 模块初始化

```
void WS_Init(uint16_t port_pin)
```



苏州析羽信息科技有限公司
Suzhou Xiyu Information Technology Co.,Ltd.

函数名称: ws_Init

函数参数: port_pin: 控制 ws2812 的 GPIO

函数返回: 无

功能概要: 初始化 ws2812 的 gpio 口

4.2 设置彩灯颜色

```
void WS_SendOnePix(uint16_t port_pin, uint8_t *ptr, uint8_t num)
```

函数名称: ws_sendOnePix

函数参数: port_pin: 控制 ws2812 的 GPIO; ptr: 代表灯珠颜色的字节数组; num: 灯珠数量

函数返回: 无

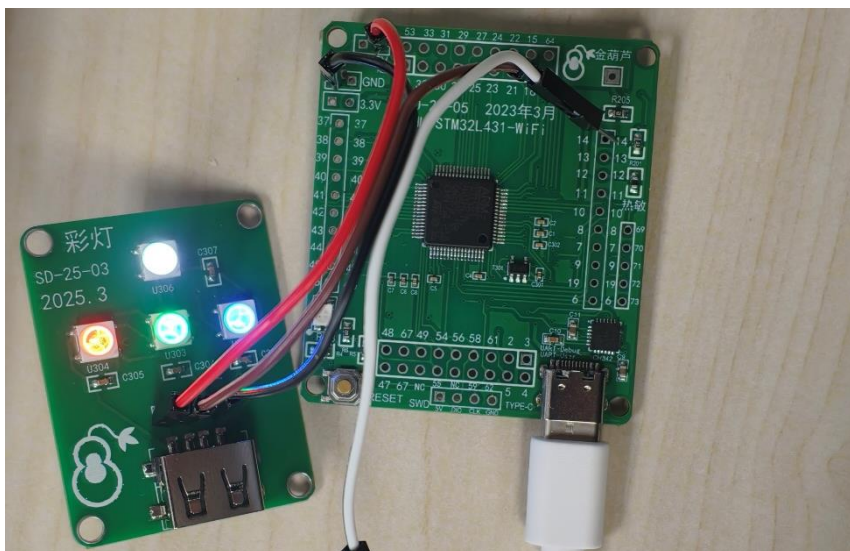
功能概要: 设置 ws2812 灯珠的颜色

4.3 示例实验

测试彩灯传感器函数，彩灯模块的 GPIO-LED 默认程序连接 PA6 (GEC16)，如果想要修改 GPIO 引脚，只需要到 user.h 中修改 COLORLIGHT 的引脚即可。本实验使用 AHL-STM32L431 开发板。

```
void Test_Lantern()
{
    // (1) 测试彩灯
    // 一种彩灯颜色占3个字节, 按grb顺序
    uint8_t grbw[12] = {0xFF, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF};
    uint8_t rwb[12] = {0x00, 0xFF, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0xFF};
    uint8_t black[12] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
    static char OutputCtl=0;
    OutputCtl++;
    if (OutputCtl>=4) OutputCtl=1;
    // (1) 彩灯测试数据
    switch (OutputCtl)
    {
        case 1:
            WS_SendOnePix (COLORLIGHT, grbw, 4);
            printf ("点亮彩灯\r\n\r\n");
            break;
        case 2:
            WS_SendOnePix (COLORLIGHT, black, 4);
            printf ("熄灭彩灯\r\n\r\n");
            break;
        case 3:
            WS_SendOnePix (COLORLIGHT, rwb, 4);
            printf ("改变彩灯颜色\r\n\r\n");
            break;
        default:
            break;
    }
}
```

实验现象如下图所示:




5. 附录

5.1 快速测试

本例程的测试需要将测试固件 hex 文件烧录进实验板中，需到网站上下载 AHL-GEC-IDE。下载地址：<http://www.xiyu-info.cn/ahl-gec-ide/>

下载安装完成后打开 AHL-GEC-IDE，打开如下图所示的下载->串口更新。



在出现的界面中选择自动搜索->连接 GEC，如果无法连接选择 B 端口 COM11（不同电脑、不同实验板端口号会不同），如下图。寻找 B 口可右键点击 windows 图标  ->设备管理器->端口（COM 和 LPT）->USB-SERIAL-B CH342 (COM39)。该串口号即为 A 端口。



若无法连接，只显示端口号，请通过任务管理器关闭 AHL-GEC-IDE，多次按下 RESET 复位按钮，10 次以上，然后重新步骤操作到这一步。连接成功过后，

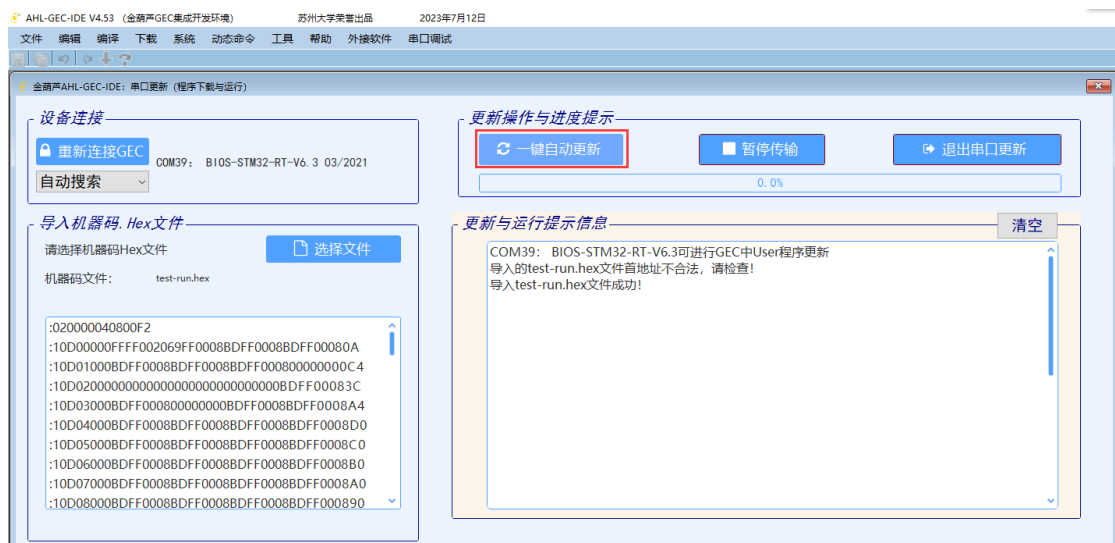
点击选择文件，如下图。



选择如下图所示的固件 hex。



点击一键自动更新，如下图所示。



更新后即可看见测试成功样例。