



RGB LED HAT

用户手册

产品概述

本产品是采用 WS2812B 智能外控 LED，共有 4*8 个 LED，它们构成了 32 个像素点，且每个点可单独寻址，在底部有 4 个可选择的电阻，分别对应的树莓派的 4 个 PWM 管脚，默认为 P18，通过此管脚可以控制全部 LED，模块直接从树莓派上获取电源，不需要另接电源。

规格

工作电压	5V
产品尺寸	65mm×30.2mm
固定孔尺寸	3.0mm

接口说明

1. 若使用在树莓派上，只需将模块直接插入树莓派的接口上即可；
2. 若使用在其他主控，建议将底部的 5V,GND,DIN,DOUT 四个焊盘焊上排针，以方便接线。

标识	管脚描述
5V	5V 电源
GND	电源地
DIN	控制信号输入
DOUT	控制信号输出

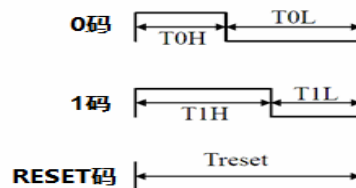
工作原理

1. 器件介绍

WS2812B 是一个集控制电路与发光电路于一体的智能外控 LED 光源。其外型与一个 5050LED 灯珠相同，每个元件即为一个像素点。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 12V 高压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。

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。像素点采用自动整形转发技术（见：数据传输协议），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。

2. 时序波形图

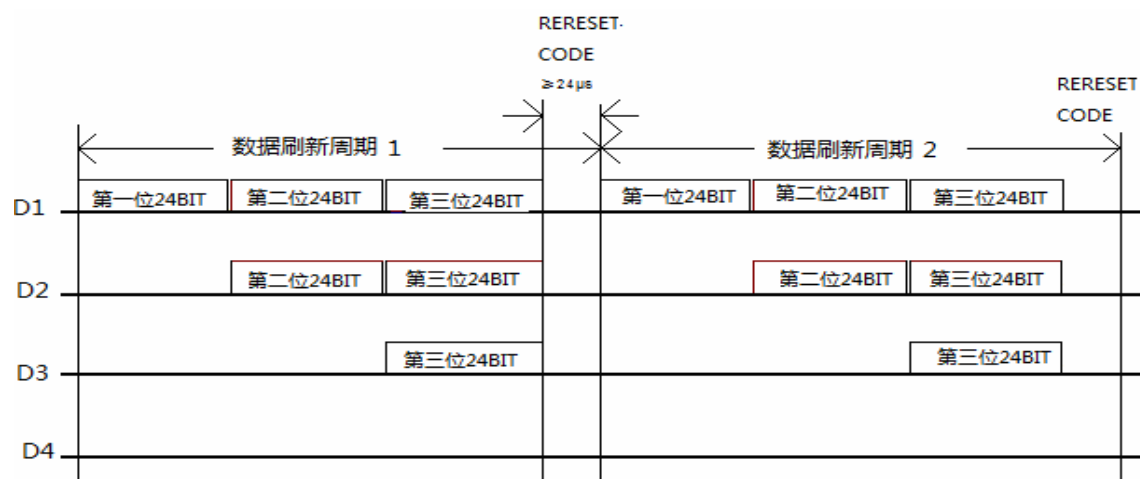


数据传输时间($T_H+T_L=1.25\mu s \pm 600ns$)

T _{0H}	0 码， 高电平时间	0.4 μs	$\pm 150ns$
T _{1H}	1 码， 高电平时间	0.8 μs	$\pm 150ns$
T _{0L}	0 码， 低电平时间	0.85 μs	$\pm 150ns$
T _{1L}	1 码， 低电平时间	0.45 μs	$\pm 150ns$
RES	帧单位，低电平时间	50 μs 以上	

3. 数据传输协议

知道了时序的波形定义，那么就可以遵循通信的协议，从而完成数据的传输。



注：其中 D1 为 MCU 端发送的数据，D2、D3、D4 为级联电路自动整形转发的数据。

4. 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。

操作现象

以上是 WS2812B 的工作原理。如果使用树莓派控制该模块，那么不需理会其工作原理，只需调用库即可。将示例程序复制到树莓派并解压。

1. 安装库

运行如下命令：

```
sudo apt-get install python-pip
```

```
sudo pip install rpi_ws281x
```

把程序下载到树莓派上，运行：

```
cd RGB_LED_HAT
```

```
sudo python ws2812.py
```

运行这个示例程序，RGB LED 会有颜色渐变的效果。

如果显示颜色不对，请尝试在 `/boot/config.txt` 文件中添加如下两个语句，重启生效。

由于 RGB LED 采用 DMA 控制，占用树莓派上的音频输出 DMA 通道，因此添加这两个语句会导致耳机接口不能使用。

```
hdmi_force_hotplug=1
```

```
hdmi_force_edid_audio=1
```

2. 代码分析

```
from rpi_ws281x import Adafruit_NeoPixel, Color

# LED strip configuration:
LED_COUNT      = 32      # Number of LED pixels.
LED_PIN        = 18      # GPIO pin connected to the pixels (must support PWM!).
LED_FREQ_HZ    = 800000  # LED signal frequency in hertz (usually 800khz)
LED_DMA        = 5        # DMA channel to use for generating signal (try 5)
LED_BRIGHTNESS = 10      # Set to 0 for darkest and 255 for brightest
LED_INVERT     = False    # True to invert the signal (when using NPN transistor level shift)

# Create NeoPixel object with appropriate configuration.
strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, LED_FREQ_HZ, LED_DMA, LED_INVERT, LED_BRIGHTNESS)
# Initialize the library (must be called once before other functions).

while 1:
    strip.begin()
    #order
    for i in range(0,strip.numPixels()):
        strip.setPixelColor(i, Color(0,0,255))
        strip.setPixelColor(i-1, Color(0,0,200))
        strip.setPixelColor(i-2, Color(0,0,150))
        strip.setPixelColor(i-3, Color(0,0,100))
        strip.setPixelColor(i-4, Color(0,0,0))
    strip.show()
    time.sleep(0.1)
```

1. `from rpi_ws281x import Adafruit_NeoPixel, Color` 导入 `rpi_ws281x` 库。
2. `Adafruit_NeoPixel` 创建一个对象 `strip`，设置 LED 为 32 个，管脚为 18，频率为 800000。
其中 `LED_BRIGHTNESS` 为 LED 的亮度。
3. `begin()` 初始化 RGB LED 函数。
4. `setPixelColor(pos, color)` 函数为设置像素点，`pos` 为 LED 位置。`color` 为 RGB 颜色。
5. `color(r, g, b)` 函数将 RGB 的值转换为一个 24 位的颜色。
6. `show()` 函数为传输数据，显示 LED 设置的颜色。
7. 更多函数可以查看 `rpi_ws281x-master /python/neopixel.py` 文件。

3. 通过 Bottle 实现 web 网页控制。

Bottle 是一个简单高效、遵循 WSGI 的微型 python Web 框架。通过 Bottle 可以快速实现 web 控制。

安装库：

```
sudo apt-get install python-bottle
```

通过 Bottle 实现 web 控制 RGB LED。

```
cd ~/RGB LED HAT/web-RGB
```

```
sudo python main
```

在浏览器地址栏内输入树莓派 ip 地址，端口号 8000。通过点击调色盘不同的位置，RGB LED 会显示不同的颜色。“static”、“breath”、“flash” 分别对应三种显示模式。

注意：此 web-RGB 程序支持手机端浏览器，电脑端浏览器不能控制。

