

第五课 RGB 灯的使用



一． 本节要点

城市是各色灯光的舞台，我们在欣赏这美丽的风景时，也不妨自己做一个变色灯吧！

- ◆ 介绍全彩 LED 灯光效果的生产原理
- ◆ 学习库文件的使用，通过软件设置小灯的各种颜色
- ◆ 使用灯光显示小车底部巡线传感器的数据情况

二． 程序下载

STEP1： 把库文件安装在 arduino IDE 路径的 libraries 文件夹下。

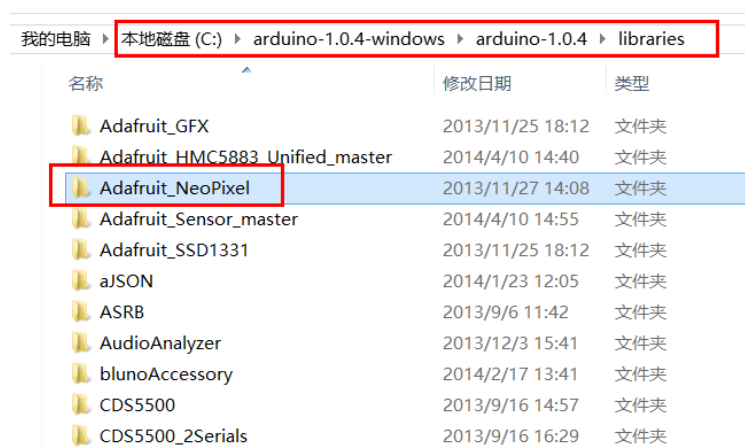
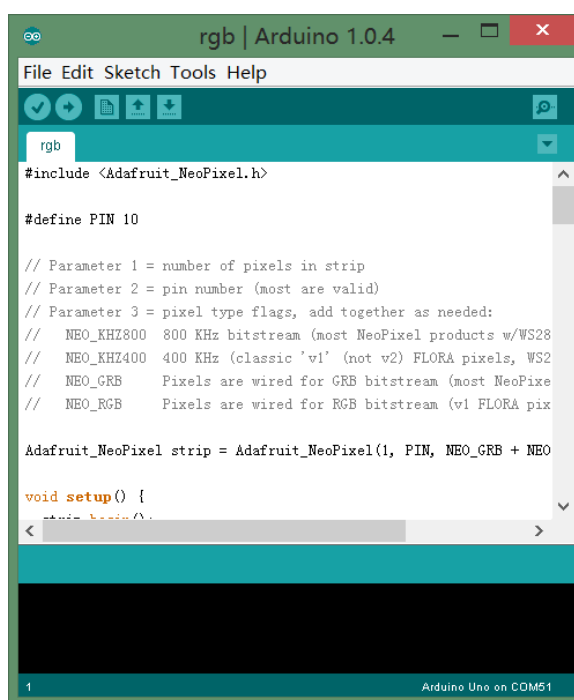


图 1 库文件存放位置示例

我的电脑 > 本地磁盘 (C:) > arduino-1.0.4-windows > arduino-1.0.4 > libraries > Adafruit_NeoPixel			
名称	修改日期	类型	大小
examples	2013/11/27 14:08	文件夹	
Adafruit_NeoPixel.cpp	2013/7/14 10:26	CPP 文件	34 KB
Adafruit_NeoPixel.h	2013/7/14 10:26	H 文件	3 KB
COPYING	2013/7/14 10:26	文件	40 KB
README.md	2013/7/14 10:26	MD 文件	1 KB

图 2 安装好的库文件路径

STEP2： 打开 rgb.ino，将小车接在电脑上，打开小车电源，下载程序。



```

rgb | Arduino 1.0.4
File Edit Sketch Tools Help
rgb
#include <Adafruit_NeoPixel.h>

#define PIN 10

// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS28
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixe
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pix

Adafruit_NeoPixel strip = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO

void setup() {
  strip.begin();
}

```

图 3 程序下载截图

好了，现在观察你的小灯吧。

三． 软件部分解析

(1) 添加库文件

```
#include <Adafruit_NeoPixel.h>
```

(2) 定义类

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(1, PIN, NEO_GRB +  
NEO_KHZ800);
```

(3) 初始化 LED 灯

```
strip.begin();
```

(4) 显示设定效果

```
strip.begin();
```

(5) 灯光变化效果

```
rainbow(20);
```

(6) 设定第 i 个灯光颜色为 Wheel ((i+i) &255))

```
strip.setPixelColor(i, Wheel((i+j) & 255));
```

四． RGB 变色原理

RGB 色彩模式是工业界的一种颜色标准，是通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色的，RGB 即是代表红、绿、蓝三个通道的颜色，这个标准几乎包括了人类视力所能感知的所有颜色，是目前运用最广的颜色系统之一。一般的 RGB 灯有 4 个引脚，R、G、B 三个引脚连接到 LED 灯的一端，还有一个引脚是共用的正极（阳）或者共用的阴极（负）。我们这里选用的是共阴 RGB。看下图 5-2，就是明白了，展示了三个 LED 如何华丽蜕变为一个 RGB 的过程，R、G、B 其实就是三个 LED 的正极，把它们的负极拉到一个公共引脚上了，它们公共引脚是负极，所以称之为共阴 RGB。

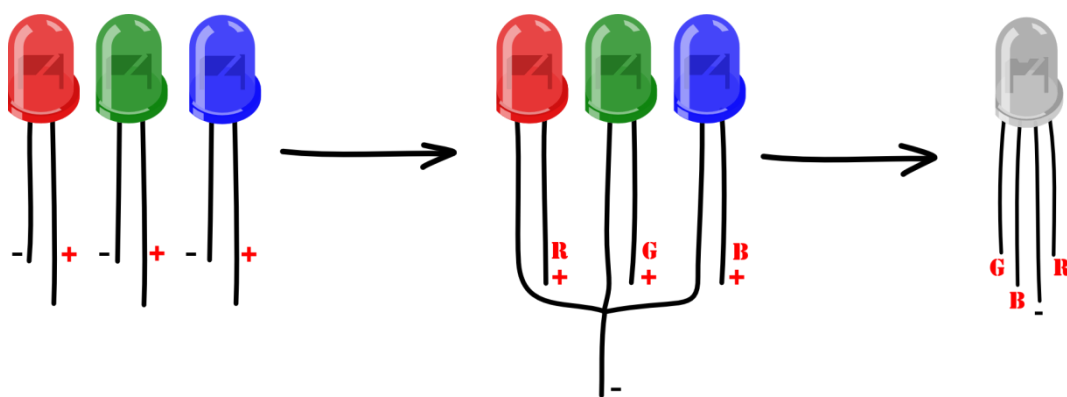


图 4 3 个 LED 蜕变为 1 个 RGB 的过程

RGB 灯如何使用呢？如何实现变色呢？

RGB 只是简单的把三个颜色的 LED 灯封装在一个 LED 中。只要当做三个灯使用就可以了。我们都知道红色、绿色、蓝色是三原色，小车上的 RGB 灯自带了一个处理芯片，只要按一定的要求发送各个颜色的值，就能让 LED 调出任何你想要的颜色。

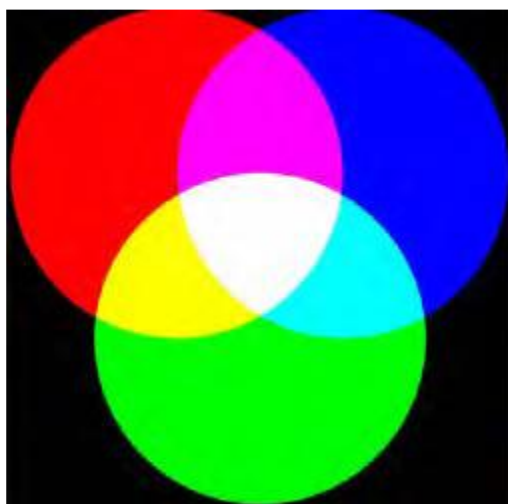


图 5 混合 R、G、B 获得不同的颜色

表 1 不同 LED 的 PWM 值所组合产生的颜色

红色	绿色	蓝色	颜色
255	0	0	红色
0	255	0	绿色
0	0	255	蓝色
255	255	0	黄色
0	255	255	蓝绿色
255	0	255	紫红色
255	255	255	白色

表中只是罗列了几种典型的颜色，可调的色彩远多于上表所示的，使用 PWM 可以产生 0~255 之间的全部颜色，共 16777216 种颜色（ $256 \times 256 \times 256$ ）。

小车上使用的 LED 灯也是这样配色的。比如我们需要红色，就在程序里使 LED 的红色打开，其它两种颜色的值为 0 即可，比如需要青色，我们需要把红色的值为 0，绿和蓝色的值最大就可以了。需要白色的话就把所有的颜色都打开就可以了。

RGB LED

这里我们认识下两种 RGB 灯的使用，第一种是单独引脚控制单种色彩，比如这种：



图 6 普通 RGB LED 灯

这种灯通常是共用正极（共阳）或负极（共阴）的，要实现某种颜色需要同时对 3 中颜色进行控制，也就是要使用到 3 个引脚。通常情况下如果一个系统使用的 器件不是很多的 ,3 个引脚是不会什么大的问题的 ,但是器件多起来就会出现引脚紧张或是不够用的情况。如果想控制两个 LED 发出不同情况，那样需要 6 个引脚才能控制，这样对引脚数量的要求就会很大。



图 7 WS2812 效果图

小车上使用的是 WS2812 型号的 RGB 灯珠。WS2812 是一个集控制电路与发光电路于一体的智能外控 LED 光源，只需要一个控制引脚，就可以做出上图的效果。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。LED 具有低电压驱动，环保节能，亮度高，散射角度大，一致性好，超低功率，超长寿命等优点。将控制电路集成于 LED 上面，电路变得更加简单，体积小，安装更加简便。因此，学习使用 ws2812 可以满足更多的场景需求。

五． 电路分析

小车上 RGB 灯的电路比较简单，只接通电源和控制脚就行了，如下图：

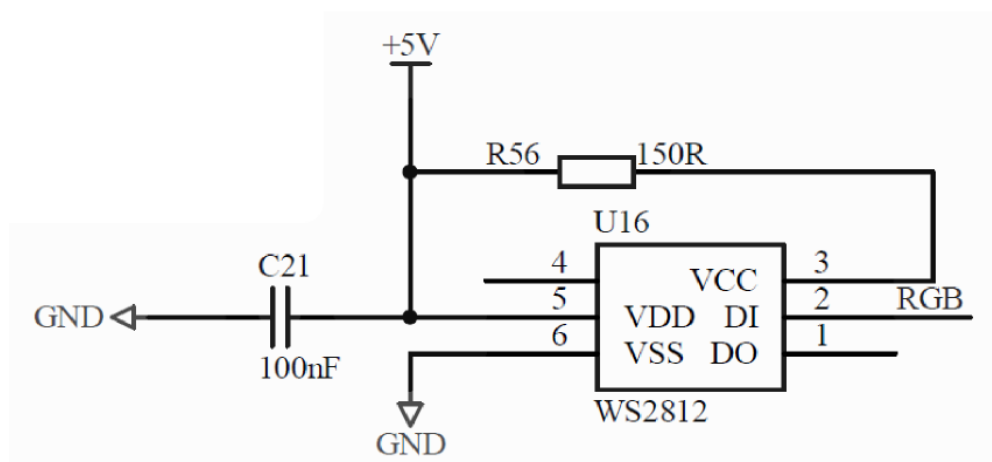


图 8 小车上使用的 RGB 灯电路图

其中标有“RGB”的引脚连 Arduino 控制器用于接收信号。DO 为信号输出，是将接收到的信号传递给下一个 ws2812 灯珠的。

六． 充电库

好了，大家一定在诧异如何在一根线上传递这么多的信息。最简单的回答就是上面多了个芯片。那么我们来一起学习下单线数据传输相关的知识吧。

首先，我们要明确的一点就是为什么需要一根线？答案很简单，可以节省成本，节约控制脚和安装方便。既然这个有这么多的优点，也就使我们的学习变得有意义起来。

我们都知道，芯片间传递信息都是靠引脚电压的变化传递的。比如你发个 1，就把引脚变为 5V，另一台收到高电平（5V）就当作收到 1，收到低电平就当作收到 0，于是这个时候接收这个信号的芯片就好像在做一个电报接收员一样把一连串的 010101 信号对应着一个信号表翻译成具体的指令。等等，说到这，喜欢思考的读者应该已经发现，之前我所说的

高电平，低电平发送数据有个致命的毛病，如果我要连续发送 2 个 0 信号怎么发？我发送 2ms（实际可能要快得多）的低电平，到底是想发多少个 0？

这时候，各位聪明的人类很快就想出了 2 个办法：

- （1）限定每个信号的时间，比如规定高低电平持续时间为 1ms，那么 2ms 低电平就代表 2 个 0。（UART 基本是用的这个原理）
- （2）多添加一个信号线，当这个信号线上的电平发生变化时读入之前信号线上的信号（后来这个新加的信号线就叫时钟线，大部分协议都是用的这种方式）

这两种方式各有利弊，第一种只需要 1 根线，同样由于只有一根线，那么为了减少误传输只能降低传输速度，第二个需要 2 根线，但可以以很快的速度进行数据传输。

我们在这里选择了一条线的方式，就只要在通讯数据的格式做出规定就行了，不如那几个数是控制红灯的，哪几个是蓝灯的。这样，我们就可以很顺利地传递我们需要的数据了。