

# Arduino 实验

## ——利用 WS2812b 灯带制作音乐节奏灯等装置

16307130308 张雨晴

### 摘要：

本次实验围绕 WS2812b LED 灯带利用 Arduino 进行了相关探索,而后基于 Scott Lawson 发布的 github 代码完成了音乐节奏灯的制作,使得灯光随着电脑播放的音乐而动态闪烁,实现了将听觉转换为视觉的音乐可视化。

### 一、引言：

音乐可视化,是指一种以视觉为核心,以音乐为载体,以大众为诉求对象,借助多种新媒体技术等传播媒介,通过画面、影像来诠释音乐内容的、视听结合的大众化传播方式。[1] 它能为理解、分析和比较音乐艺术作品形态的表现力和内外部结构提供的一种直观视觉呈现的技术。[2]

Arduino 是一款便捷灵活、方便上手的开源电子原型平台。包含硬件(各种型号的 Arduino 板)和软件(ArduinoIDE)。音乐节奏灯是利用 Arduino 将音乐节奏可视化出来的一种装置,使 LED 灯光颜色、亮度等随着音乐的节奏、音量而发生变化。本次实验围绕 WS2812b LED 灯带利用 Arduino 进行了相关探索,而后基于 Scott Lawson 发布的 github 代码完成了音乐节奏灯的制作。

### 二、实验原理：

#### 1. 基于 arduino ws2812b 简单的渐变代码

彩虹色 RGB 编码

	R	G	B
红	255	0	0
橙	255	152	0
黄	255	255	0
绿	0	255	0
青	0	255	255
蓝	0	0	255
紫	150	0	255

彩虹色渐变：

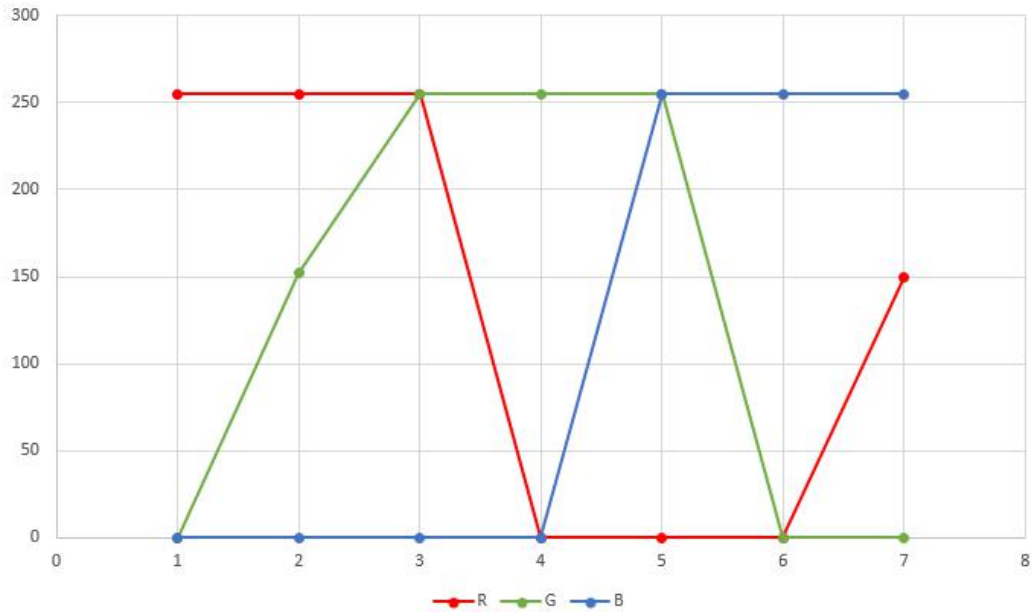


图 1 彩虹渐变

## 2. 音乐节奏灯说明

我们使用 Arduino UNO 作为主要的计算和处理模块，wifi 进行通信，在节奏灯的模式下，通过电脑上的 voicemeeter 软件来采集声音，通过得到的声音来控制灯带的颜色和闪烁。

## 3. 基于 WiFi 模块的数据传递

使用 ESP32 一般是为了它的 WiFi 功能，其中最基本的就是建立/连接网络等基本操作，其它面向用户的网络应用都是建立在这基础上的。

ESP8266 芯片有三种工作模式，分别为：STA 模式，即 ESP8266 模块通过路由器连接互联网，手机或电脑通过互联网实现对设备的远程控制；AP 模式，即 ESP8266 模块作为热点，手机或电脑直接与模块连接，实现局域网无线控制；STA+AP 模式，即前两种模式的共存模式，即可以通过互联网控制可实现无缝切换，方便操作。[3]

## 三、实验装置与软件：

ESP8266 开发板、WS2812b LED 灯带（60 颗/1 米）、PVC 管、5V 的灯座、灯罩、杜邦线、数据线

开发环境：arduino2.0 Python3.7

其他软件：voicemeeter

## 四、实验过程

### （一）基于 arduino ws2812b LED 灯的可视化探究

#### 1. 连接电路

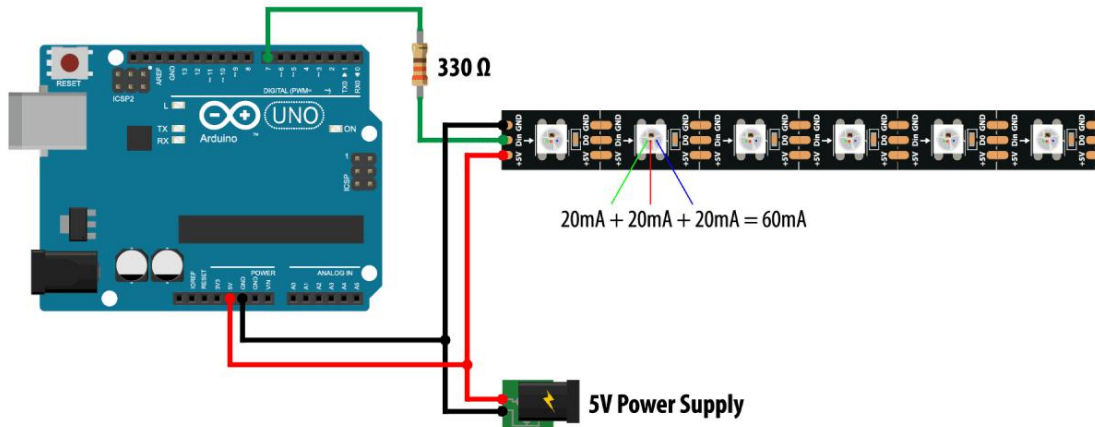


图 2 电路连接示意图

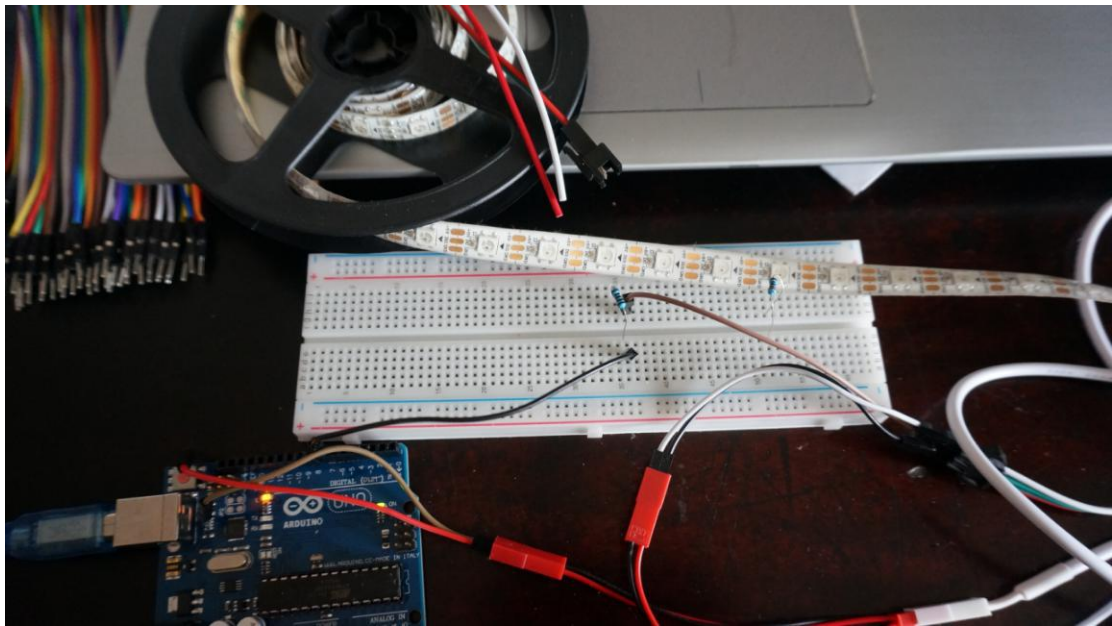


图 3 电路连接实物图

## 2. 导入代码

利用包 FastLED 完成点亮 LED 的操作

基础代码：

```
#include <FastLED.h>
#define LED_PIN 7
#define NUM_LEDS 60
CRGB leds[NUM_LEDS];
void setup() {
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
}
```

### (1) 灯带彩虹色点亮

每隔时间 500 ms 点亮一只 LED。

主要代码（由于空间有限这里只保留了前面几盏灯的代码）：

```
void loop() {
```

```

leds[0] = CRGB(255, 0, 0);
FastLED.show();
delay(500);
leds[1] = CRGB(0, 255, 0);
FastLED.show();
delay(500);
leds[2] = CRGB(0, 0, 255);
FastLED.show();
delay(500);
leds[3] = CRGB(150, 0, 255);
FastLED.show();
delay(500);
leds[4] = CRGB(255, 200, 20);
FastLED.show();
delay(500);
}

```

## (2) 灯带双色回旋

遍历灯带上的灯，先从头到尾用蓝色点亮，再从尾到头返回红色，达到炫酷效果。

主要代码：

```

void loop() {
  for (int i = 0; i <= 59; i++) {
    leds[i] = CRGB ( 0, 0, 255);
    FastLED.show();
    delay(40);
  }
  for (int i = 59; i >= 0; i--) {
    leds[i] = CRGB ( 255, 0, 0);
    FastLED.show();
    delay(40);
  }
}

```

## (3) 灯带彩虹色渐变

根据彩虹色渐变原理，缓慢完成由红——紫的彩虹渐变。

主要代码（由于空间有限这里只保留了红——橙——黄三个步骤的代码）：

```

void loop() {
  GRE1=0;for(int o=0;o<255;o++){
    RDE1=255;
    GRE1=GRE1+1;
    BLE1=0;
    for(int l=0;l<NUMPIXELS;l++){
      pixels.setPixelColor(l, pixels.Color(RDE1,GRE1,BLE1));
    }
    pixels.show();
  }
}

```

```

}
    delay(5); //让渐变慢些
}
RDE1=255;
for(int p=0;p<255;p++){
    RDE1=RDE1-1;
    GRE1=255;
    BLE1=0;
    for(int l=0;l<NUMPIXELS;l++){
        pixels.setPixelColor(l, pixels.Color(RDE1,GRE1,BLE1));
        pixels.show();
    }
    delay(5);
}
BLE1=0;
for(int k=0;k<255;k++){
    RDE1=0;
    GRE1=255;
    BLE1=BLE1+1;
    for(int l=0;l<NUMPIXELS;l++){
        pixels.setPixelColor(l, pixels.Color(RDE1,GRE1,BLE1));
        pixels.show();
    }
    delay(5);
}
}
}

```

## (二) 音乐节奏灯

### 1. 安装 Python 及所需的包

主要需要 numpy、scipy 来进行 python 的编程，pyqtgraph (PYQT5) 来进行 user interface 交互式页面的制作，同时需要安装 pyaudio 来进行 python 与声音信号的交互。

### 2. 安装 Arduino 所需要的库

采用 ESP8266 作为开发板，因此需要在【首选项】中设置附加开发板管理器网址为 ([http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json))。

然后点击【工具】-【开发板】-【开发板管理器】-安装 ESP8266。

接着点击【项目】-【加载库】-【管理库】-安装 websocket。

3. 利用 WIFI 传输的功能编写 arduino 代码，编写 python 代码采集电脑音频、制作 UI 页面（见附录）

4. 将 arduino 代码上传到 ESP8266 上

5. 运行 MusicLED.py，看到由 PYQT5 制作的 UI 页面

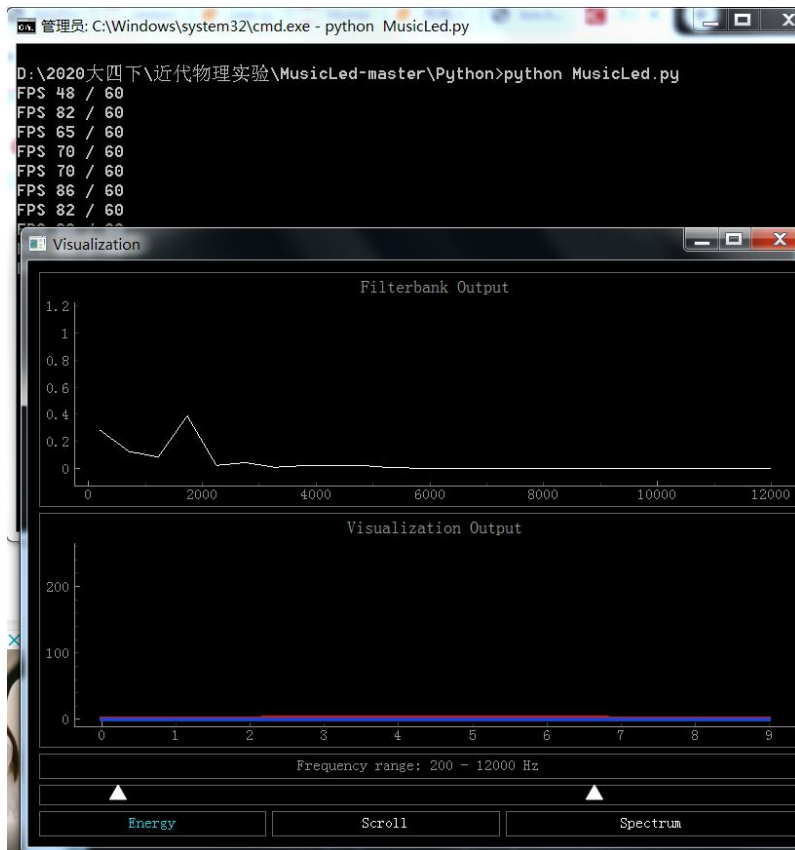


图 4 UI 界面-音乐可视化

## 6. 组装节奏灯

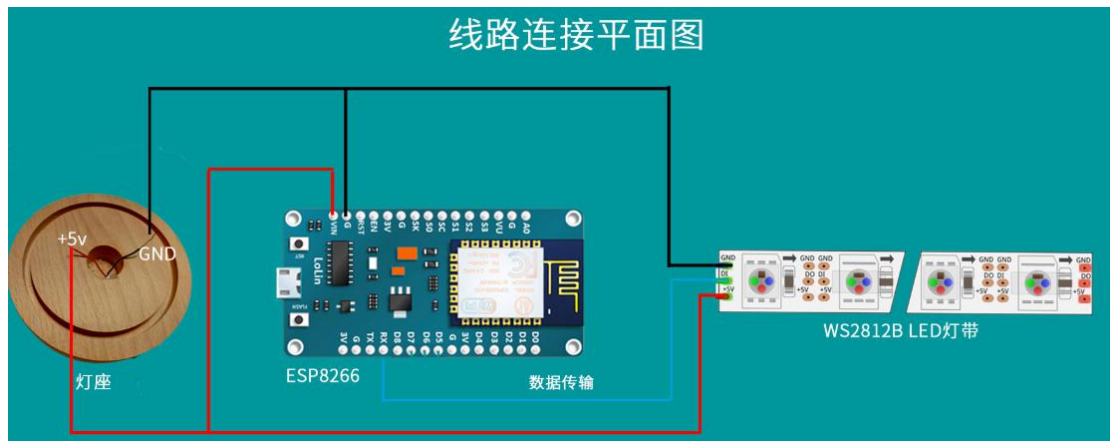


图 5 电路连接示意图

- ① ESP8266 RX 口接灯带 DI、DO 口。
- ② 灯座正极接两根线，一根接到 ESP8266 的 VIN 口，一根接到灯带的+5V。
- ③ 灯座负极接两根线，一根接到 ESP8266 的 G 口，一根接到灯带的 GND。
- ④ 每只灯泡亮起需要 60mA，因此为了给灯带供电，使用强力的电源（如连接快充充电器）。
- ⑤ WS2812b 灯带标有方向，要用起始段连接灯座和 ESP8266。
- ⑥ 灯座正负极各要连接两根线，选用 JSP 一分二公母插头。

- ⑦ 将然后将灯带缠绕到 PVC 管上。
- ⑧ 通电，查看效果



图 6 JSP 一分二公母插头



图 7 将灯带缠绕到 PVC 管

## 7. 下载并打开 Voicemeeter，实现电脑的混音功能

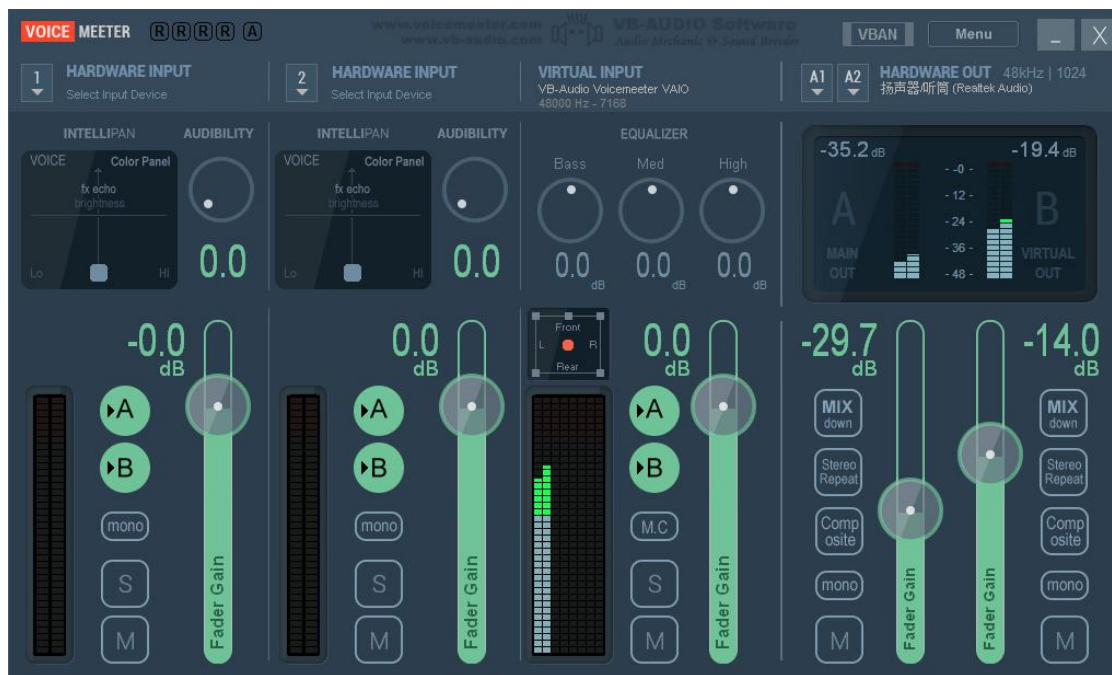


图 8 Voicemeeter 页面

Voicemeeter 显示的音量大小变化与 python UI 界面显示的波形变化应当一致。

## 五、实验效果

### (一) 基于 arduino ws2812b LED 灯的可视化探究

#### 1. 灯带彩虹色点亮

如图为点亮后。

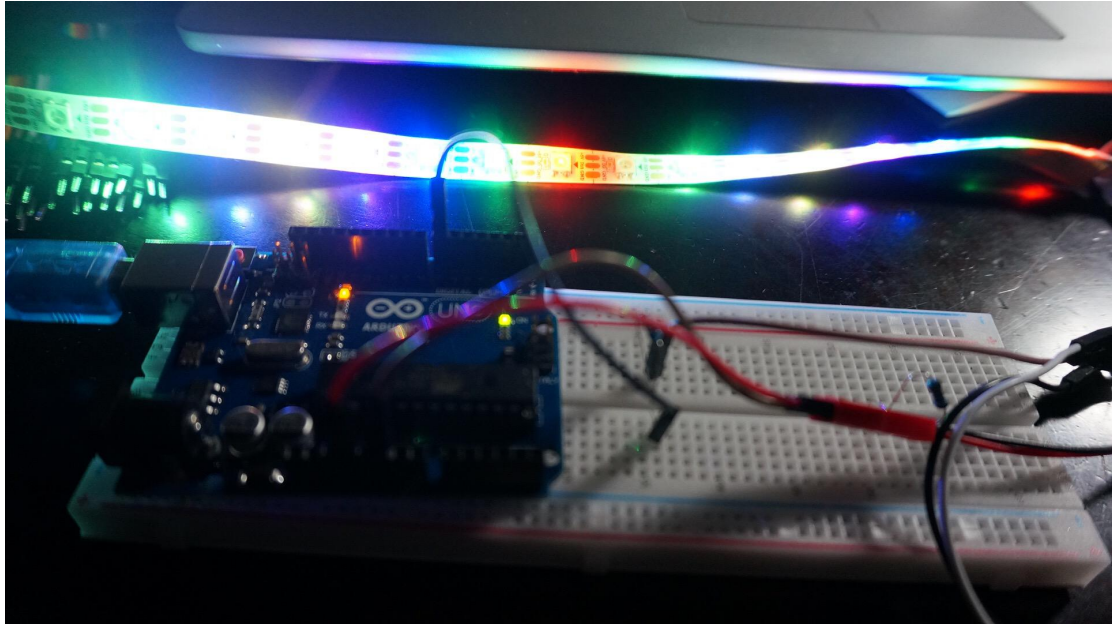


图 9 灯带彩虹色点亮

## 2. 灯带双色回旋

如图为某一帧的中间过程。



图 10 灯带双色回旋

## 3. 灯带彩虹色渐变

如图为黄色灯光。



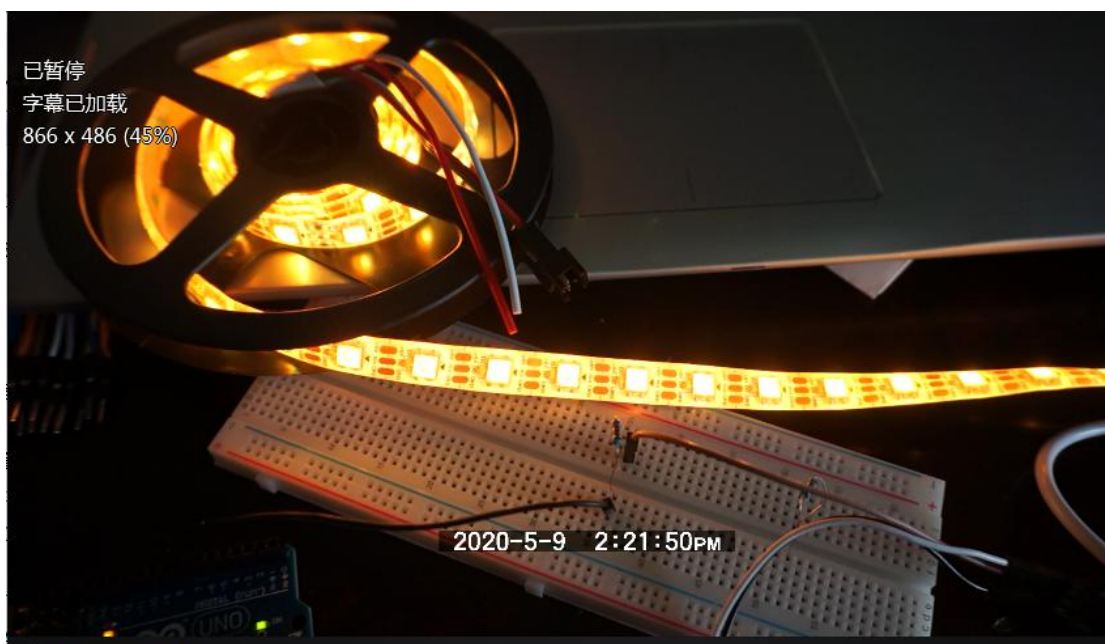


图 11 彩虹色渐变 (黄色)

## (二) 音乐节奏灯

给灯座通上电，运行 python，打开 Voicemeeter，打开音乐播放器播放音乐/直接对着电脑讲话，可以看到灯光随着音乐节奏变化。

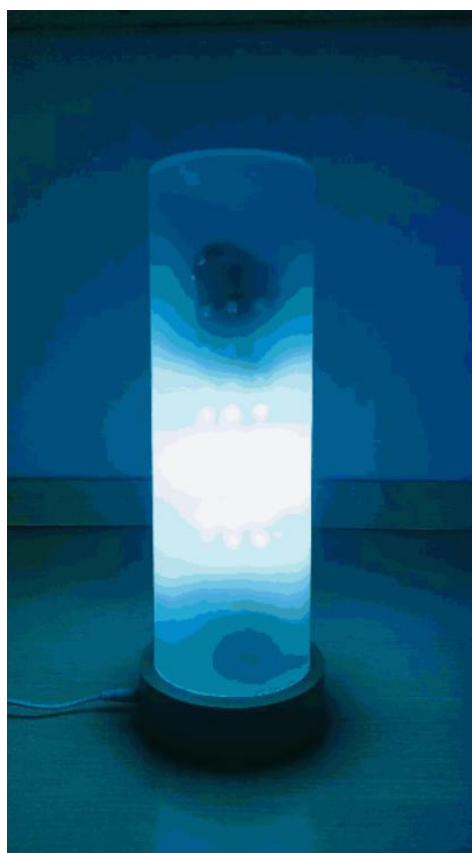
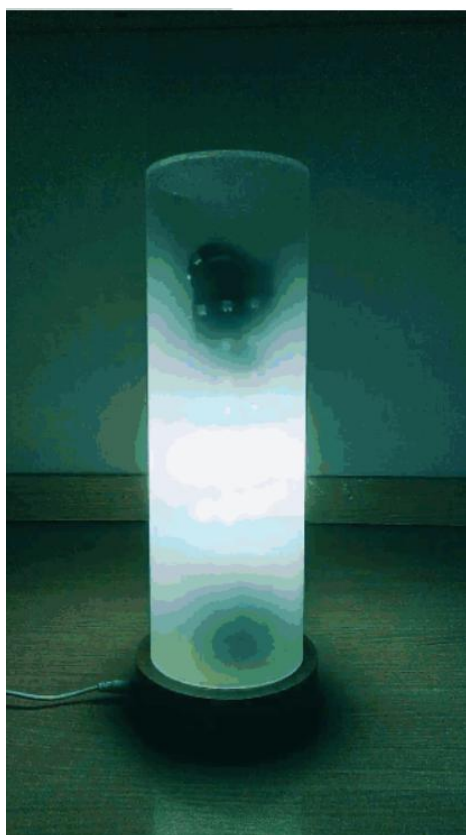


图 11 音乐节奏灯

## 六、实验结论：

本实验学习了 Arduino 的安装和使用方法，运行了三个示例程序来熟悉 ws2812b 灯带的点亮流程操作，而后基于 Scott Lawson 发布的 github 代码完成了音乐节奏灯的制作，实现了灯光随音乐节奏变化的功能。

[1] 文化月刊·动漫游戏 2011 年第 11 期

[2] 文化月刊·动漫游戏 2011 年第 12 期

[3] ESP3266 与 wifi 通信：

[https://www.csdn.net/gather\\_28/MtTaggzsmTEOMSlibG9n.html](https://www.csdn.net/gather_28/MtTaggzsmTEOMSlibG9n.html)

音乐节奏灯参考资料: <https://github.com/scottlawsonbc/audio-reactive-led-strip>

附录：

参考资料: <https://github.com/scottlawsonbc/audio-reactive-led-strip>

修改以下代码：

Arduino 代码

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <WebSocketsServer.h>
#include <Hash.h>
#include <WiFiUdp.h>
#include "ws2812_i2s.h"

#define NUM_LEDS 60 //LED 灯数量

#define BUFFER_LEN 1024

#define PRINT_FPS 1

const char* ssid = "zzzzzzzccccccc"; //WIFI 名字
const char* password = "zcp5313.."; //WIFI 密码
unsigned int localPort = 7777;
char packetBuffer[BUFFER_LEN];

static WS2812 ledstrip;
static Pixel_t pixels[NUM_LEDS];
WiFiUDP port;

//可以使用Windows+R 键输入 cmd, 然后输入命令 ipconfig 查看自己网络情况。
```

```
IPAddress ip(192, 168, 1, 150); //给 ESP8266 设置的 IP 地址, 注意, 这个 IP 不可以跟其他设备  
(手机电脑灯) 重复。
```

```
IPAddress gateway(192, 168, 1, 1); //网络的网关, 这个跟电脑的网关一样
```

```
IPAddress subnet(255, 255, 255, 0); //子网掩码, 跟电脑的一样, 一般不需要修改。
```

```
void setup() {  
    Serial.begin(115200);  
    WiFi.config(ip, gateway, subnet);  
    WiFi.begin(ssid, password);  
    Serial.println("");  
    // Connect to wifi and print the IP address over serial  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
    Serial.println("");  
    Serial.print("Connected to ");  
    Serial.println(ssid);  
    Serial.print("IP address: ");  
    Serial.println(WiFi.localIP());  
    port.begin(localPort);  
    ledstrip.init(NUM_LEDS);  
}
```

```
uint8_t N = 0;  
#if PRINT_FPS  
    uint16_t fpsCounter = 0;  
    uint32_t secondTimer = 0;  
#endif
```

```
void loop() {  
    // Read data over socket  
    int packetSize = port.parsePacket();  
    // If packets have been received, interpret the command  
    if (packetSize) {  
        int len = port.read(packetBuffer, BUFFER_LEN);  
        for(int i = 0; i < len; i+=4) {  
            packetBuffer[len] = 0;  
            N = packetBuffer[i];  
            pixels[N].R = (uint8_t)packetBuffer[i+1];  
            pixels[N].G = (uint8_t)packetBuffer[i+2];  
            pixels[N].B = (uint8_t)packetBuffer[i+3];  
        }  
    }  
}
```

```

    }
    ledstrip.show(pixels);
    #if PRINT_FPS
        fpsCounter++;
    #endif
}
#if PRINT_FPS
    if (millis() - secondTimer >= 1000U) {
        secondTimer = millis();
        Serial.printf("FPS: %d\n", fpsCounter);
        fpsCounter = 0;
    }
#endif
}
}

```

## Config.py

```

#coding = utf8
from __future__ import print_function
from __future__ import division
import os

DEVICE = 'esp8266'

if DEVICE == 'esp8266':
    # 下面是刚才在 Arduino 中为 ESP8266 设置的 IP 地址。
    UDP_IP = '192.168.1.150'
    UDP_PORT = 7777
    SOFTWARE_GAMMA_CORRECTION = False

USE_GUI = True

DISPLAY_FPS = True
# 下面是 LED 灯的数量
N_PIXELS = 60

GAMMA_TABLE_PATH = os.path.join(os.path.dirname(__file__), 'gamma_table.npy')
MIC_RATE = 44100

FPS = 60
_max_led_FPS = int(((N_PIXELS * 30e-6) + 50e-6)**-1.0)
assert FPS <= _max_led_FPS, 'FPS must be <= {}'.format(_max_led_FPS)

MIN_FREQUENCY = 200
MAX_FREQUENCY = 12000

```

```
N_FFT_BINS = 24
```

```
N_ROLLING_HISTORY = 2
```

```
MIN_VOLUME_THRESHOLD = 1e-7
```