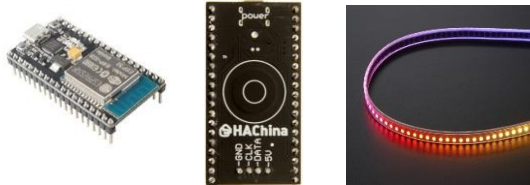


# 音乐灯带硬件连接与基本使用

## 【必要硬件】

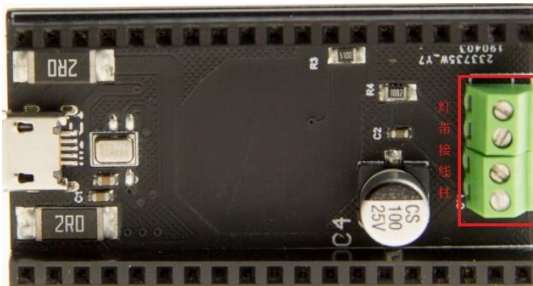
产品地址: <https://lw.hachina.io/>

1. nodemcu-32s
2. 麦克风扩展板
3. led 灯带 (neopixel/ws2812b/apa102/...)

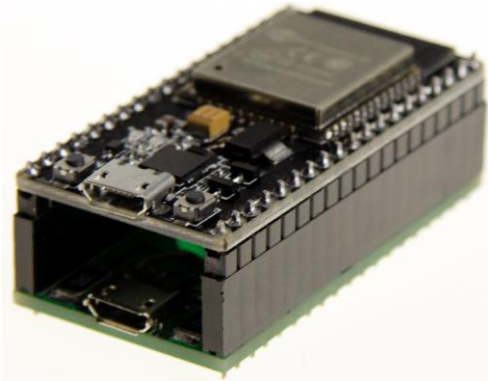


## 【操作步骤】

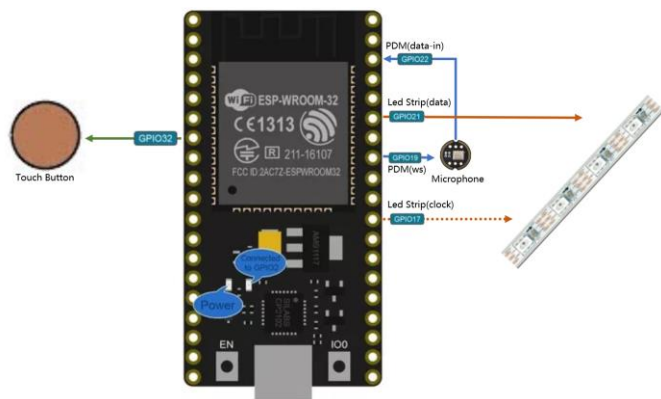
1. 硬件连接
  - a) 灯带接线



- b) 对接麦克风扩展板与 nodemcu-32s



- c) 实际物理连接图



## 2. 电源供电模式

电源	说明
Nodemcu32s 上的 mini usb	首先保证 esp32 的电力供应，剩余供给灯带（具有 usb 通讯功能）
麦克风板上的 mini usb	首先给灯带供电，剩余供给 esp32（无 usb 通讯功能）
灯带上的电源线	首先给灯带供电，剩余供给 esp32

注：

- 可以接一个或多个电源；
- mini usb 口一般不超过 2 安培；
- 在实际工程中，一般只要接通灯带上的电源线，给与充足的电力就可以了

## 3. 缺省程序的使用

### 【参考】

- 项目 github 地址  
<https://github.com/zhujisheng/audio-reactive-led-strip>

## 音乐灯带接入 ESPHome (1) ——常规设备

### 【操作步骤】

1. 连接 nodemcu 的 USB 口到 esphome 运行设备上
2. 构建新的 esp32 配置
3. 在基础配置上，增加对应硬件信息
4. UPLOAD 固件与 USB 功率不够的问题
5. 在 HomeAssistant 中配置 esphome
6. 使用演示

### 【参考】

- ESPHome 操作视频
  - 《ESPHome——不编程，集成 ESP8266》
  - 《使用 NFC 识别不同的 ID 卡》

- 参考配置

```
esphome:
  name: esp32_music
  platform: ESP32
  board: nodemcu-32s

wifi:
  .....
logger:

api:
  password: 'hachina'

ota:
  password: 'hachina'

esp32_touch:
binary_sensor:
  - platform: esp32_touch
    name: "Touch Pad on ESP32"
    pin: GPIO32
    threshold: 1000
    on_press:
      then:
        - light.toggle: LedsStrip

output:
  - platform: ledc
    id: output_gpio2
    pin: GPIO2

light:
  - platform: monochromatic
    name: "Led on ESP32 Board"
    output: output_gpio2
  - platform: fastled_clockless
    id: LedsStrip
    chipset: NEOPIXEL
    pin: GPIO21
    num_leds: 60
    name: "MUSIC LEDS"
    effects:
      - addressable_rainbow:
      - addressable_color_wipe:
      - addressable_scan:
      - addressable_twinkle:
      - addressable_random_twinkle:
      - addressable_fireworks:
      - addressable_flicker:
  # - platform: fastled_spi
```

```
# id: LedsStrip
# chipset: APA102
# data_pin: GPIO21
# clock_pin: GPIO17
# num_leds: 60
# rgb_order: BGR
# name: "MUSIC LEDS"
# effects:
#   - addressable_rainbow:
#   - addressable_color_wipe:
#   - addressable_scan:
#   - addressable_twinkle:
#   - addressable_random_twinkle:
#   - addressable_fireworks:
#   - addressable_flicker:
```

- ESPHome 中配置 TouchPad  
[https://esphome.io/components/binary\\_sensor/esp32\\_touch.html](https://esphome.io/components/binary_sensor/esp32_touch.html)
- ESPHome 中配置 FastLED Light  
<https://esphome.io/components/light/fastled.html>

## 音乐灯带接入 ESPHome (2)

### ——增加音乐效果以及定制化

#### 【操作步骤】

1. 增加灯带的音乐效果
  - a) 下载必要的源代码
  - b) 修改配置文件
  - c) UPLOAD
2. 接入不通规格的灯带(APA102 30led 灯带)
3. 通过服务调用控制智能灯的显示模式

#### 【参考】

- 下载命令参考

```
git clone https://github.com/zhujiasheng/audio-reactive-led-strip
cp audio-reactive-led-strip/music_leds_esphome.h ~/esphome_config/
cp -r audio-reactive-led-strip/includes/ ~/esphome_config/
```

- ESPHome 中配置 FastLED Light

<https://esphome.io/components/light/fastled.html>

- 参考配置

```
esphome:
  name: esp32_music
  platform: ESP32
  board: nodemcu-32s
  includes:
    - music_leds_esphome.h
    - includes/FFT.h
    - includes/ExpFilter.h
    - includes/VisualEffect.h
    - includes/gaussian_filterId.h

.....;

light:
  - platform: fastled_clockless
    id: LedsStrip
    chipset: NEOPIXEL
    pin: GPIO21
    num_leds: 60
    name: "MUSIC LEDS"
    effects:
      - addressable_rainbow:
      - addressable_color_wipe:
      - addressable_scan:
      - addressable_twinkle:
      - addressable_random_twinkle:
      - addressable_fireworks:
      - addressable_flicker:
      - addressable_lambda:
          name: Scroll with Music
          update_interval: 0s
          lambda: |-
            music_leds.ShowFrame(MODE_SCROLL, &it);
      - addressable_lambda:
          name: Energy with Music
          update_interval: 0s
          lambda: |-
            music_leds.ShowFrame(MODE_ENERGY, &it);
      - addressable_lambda:
          name: Spectrum with Music
          update_interval: 0s
          lambda: |-
            music_leds.ShowFrame(MODE_SPECTRUM, &it);
```

# Arduino 与 ESP 硬件

## 【操作步骤】

1. 下载安装 Arduino IDE  
<https://www.arduino.cc/en/Main/Software>
2. 增加 esp8266 与 esp32 开发库
3. 连接 ESP 硬件，选择对应的硬件平台，选择 COM 口
4. arduino 编程语言，setup 和 loop 函数
5. 编写一个每秒闪烁一次 led 的固件程序

```
void setup() {  
    pinMode(2, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(2, HIGH);  
    delay(500);  
    digitalWrite(2, LOW);  
    delay(500);  
}
```

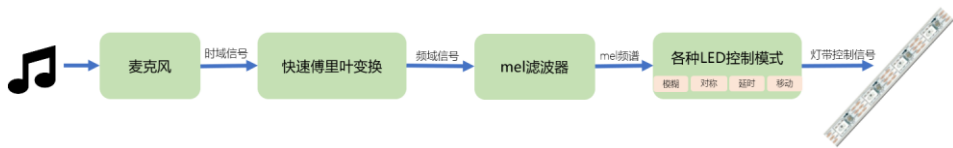
## 【参考】

- ESP8266 开发板管理器配置  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
- ESP32 开发板管理器配置  
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

# 音乐灯带音频处理过程与 Arduino 平台编译

## 【操作步骤】

### 1. 音频处理过程



### 2. 安装 fastled 库

### 3. 在 arduino 中编译音乐灯带程序

程序	功能
<code>audio-reactive-led-strip.ino</code>	Arduino 平台主程序
<code>includes/ExpFilter.h</code>	平滑过滤器（一般用于时间维度）
<code>includes/FFT.h</code>	快速傅里叶变换、mel 频谱转换
<code>includes/gaussian_filter1d.h</code>	一维高斯模糊（一般用于空间维度）
<code>includes/VisualEffect.h</code>	灯带虚拟效果的实现

## 【参考】

- Arduino 的 FastLED 库  
<http://fastled.io/>
- 音乐灯带的 github 地址  
<https://github.com/zhujisheng/audio-reactive-led-strip>