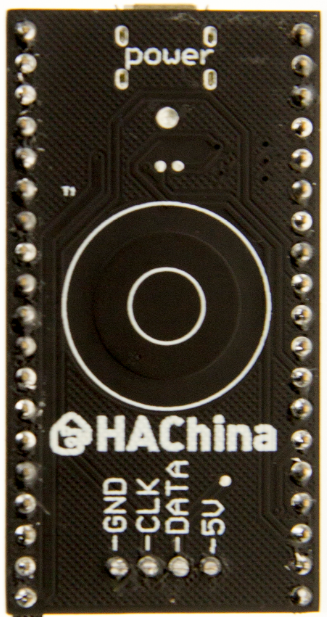
音乐灯带硬件连接与基本使用

【必要硬件】

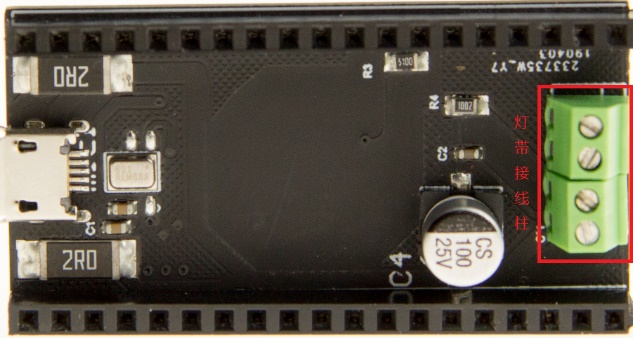
产品地址：<https://lw.hachina.io/>

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

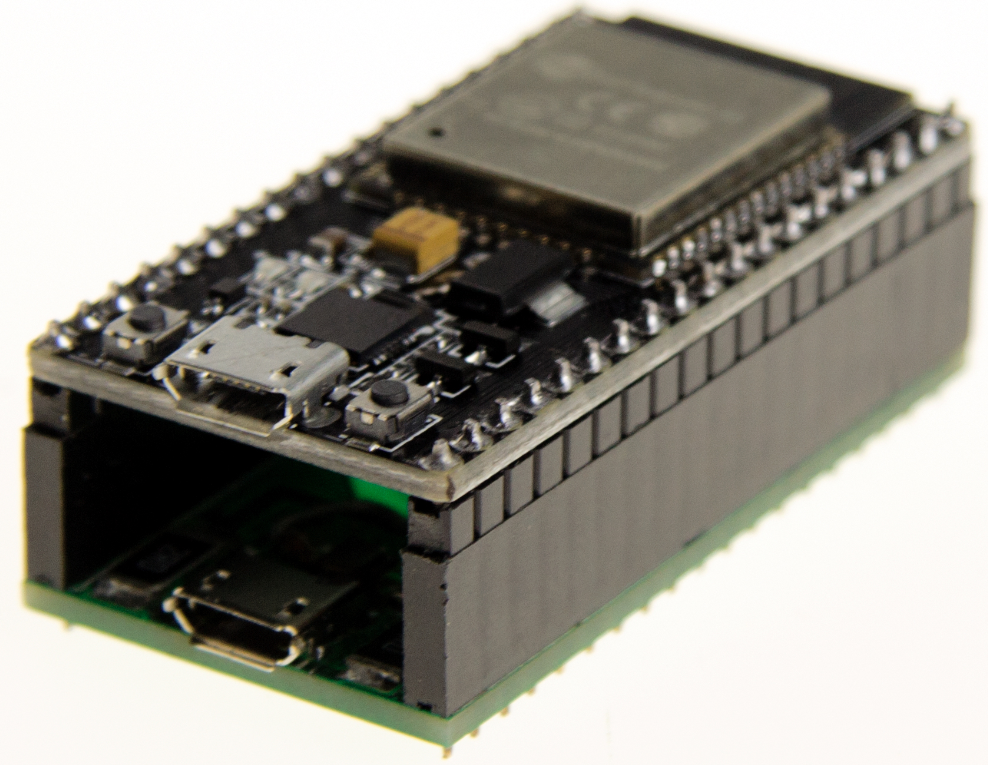
  

【操作步骤】

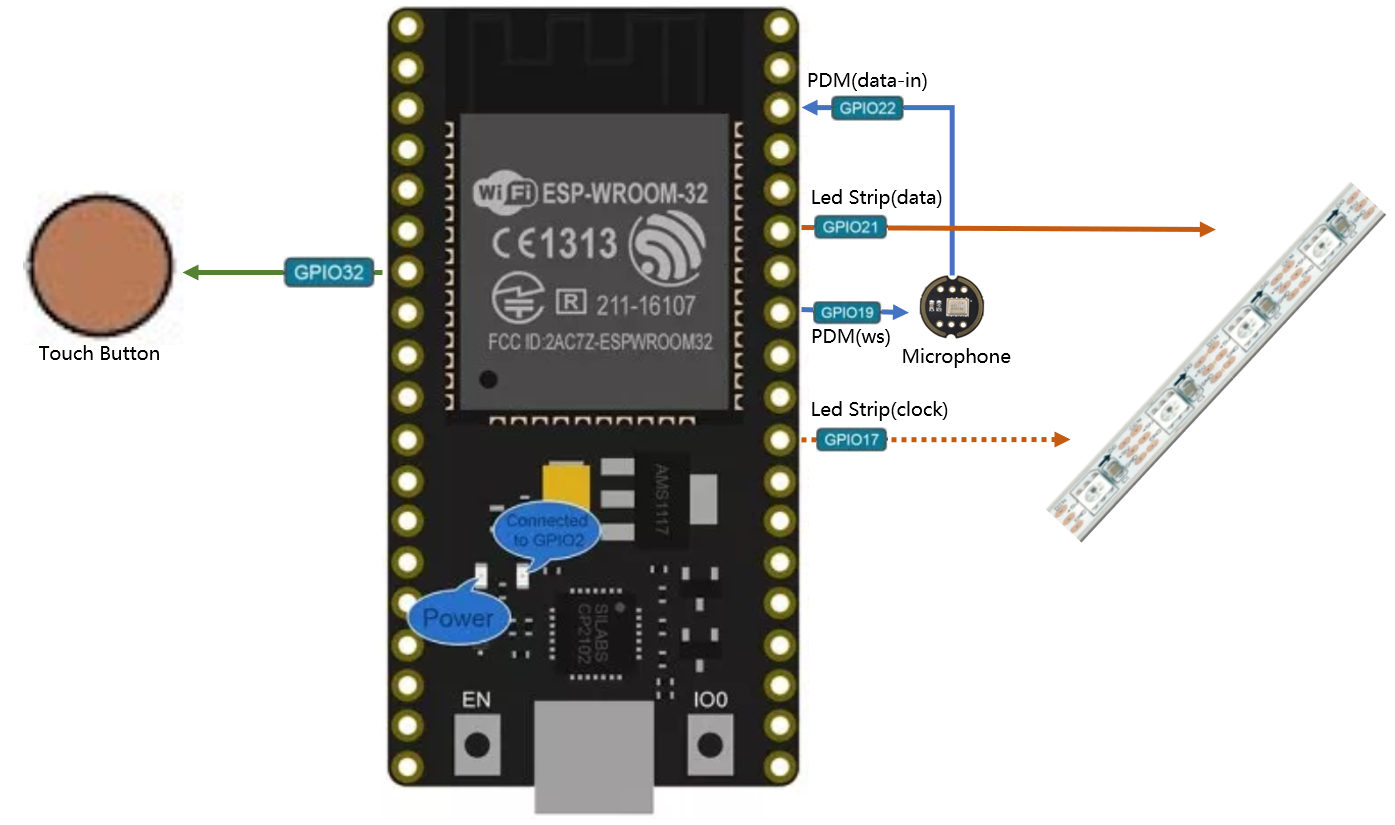
1. 硬件连接
   1. 灯带接线



* 1. 对接麦克风扩展板与nodemcu-32s



* 1. 实际物理连接图



1. 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ubs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：

* 1. 可以接一个或多个电源；
  2. mini usb口一般不超过2安培；
  3. 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

1. 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>

* 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
   1. 下载必要的源代码
   2. 修改配置文件
   3. 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/zhujisheng/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

……:

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>

1. 增加esp8266与esp32开发库
2. 连接ESP硬件，选择对应的硬件平台，选择COM口
3. arduino编程语言，setup和loop函数
4. 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

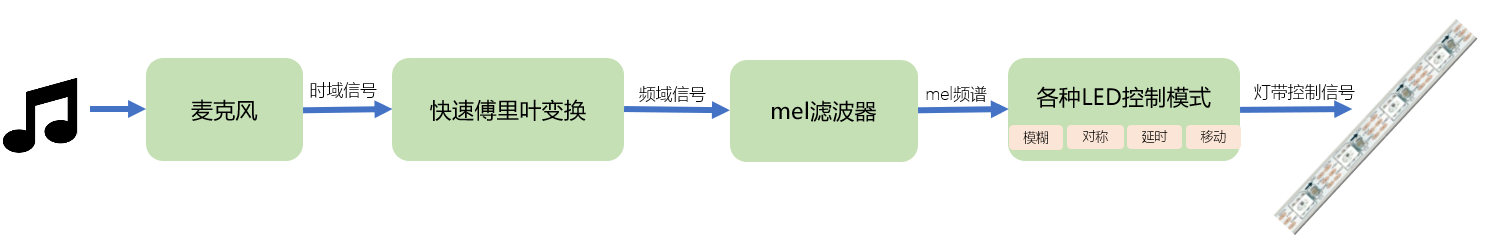
* ESP32开发板管理器配置

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



1. 安装fastled库
2. 在arduino中编译音乐灯带程序

|  |  |
| --- | --- |
| 程序 | 功能 |
| audio-reactive-led-strip.ino | Arduino平台主程序 |
| includes/ExpFilter.h | 平滑过滤器（一般用于时间维度） |
| includes/FFT.h | 快速傅里叶变换、mel频谱转换 |
| includes/gaussian\_filter1d.h | 一维高斯模糊（一般用于空间维度） |
| includes/VisualEffect.h | 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>