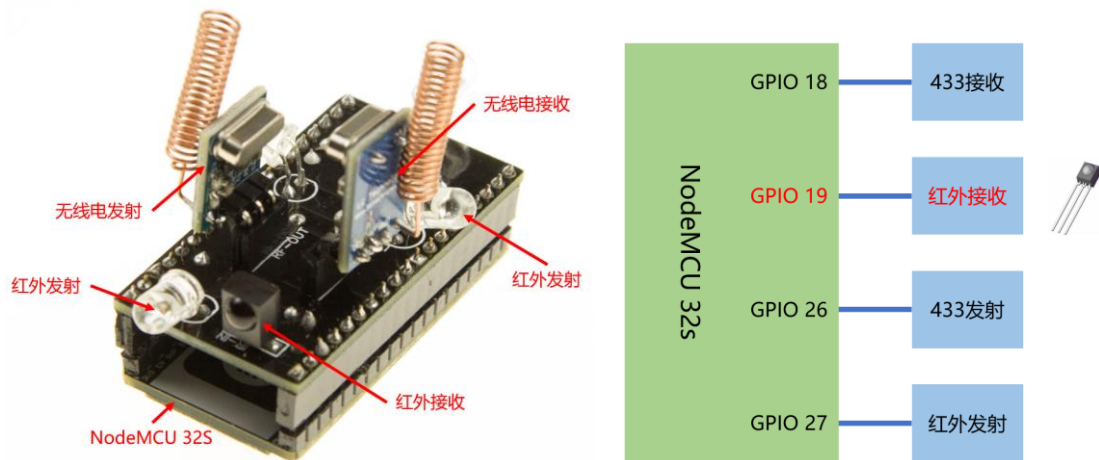


使用红外与无线模块 (1) —— 红外接收

【硬件】



【操作步骤】

1. 红外接收配置

```
remote_receiver:  
  - pin:  
    number: GPIO19  
    inverted: yes  
    dump: all  
    id: ir_receiver
```

2. 接收红外信号

3. 理解 raw 格式的红外编码

4. 配置开关型传感器 (红外接收)

```
binary_sensor:  
  - platform: remote_receiver  
    name: "Sony 电视开关按钮"  
    sony:  
      data: 0x00000A90  
      nbits: 12  
    filters:  
      delayed_off: 300ms  
  - platform: remote_receiver  
    name: "Sharp 电视开关按钮"  
    raw:  
      code: [334, -1765, ...]  
    filters:  
      delayed_off: 300ms
```

【参考】

- 编码图形查看
<https://test.sui.li/oszi/>
- sharp 遥控器编码协议
<https://www.sbprojects.net/knowledge/ir/sharp.php>
- esphome 中的 Remote Receiver
https://esphome.io/components/remote_receiver.html
- esphome 使用相关视频

[《ESPHome——不编程，集成 ESP8266》](#)

[《使用 NFC 识别不同的 ID 卡》](#)

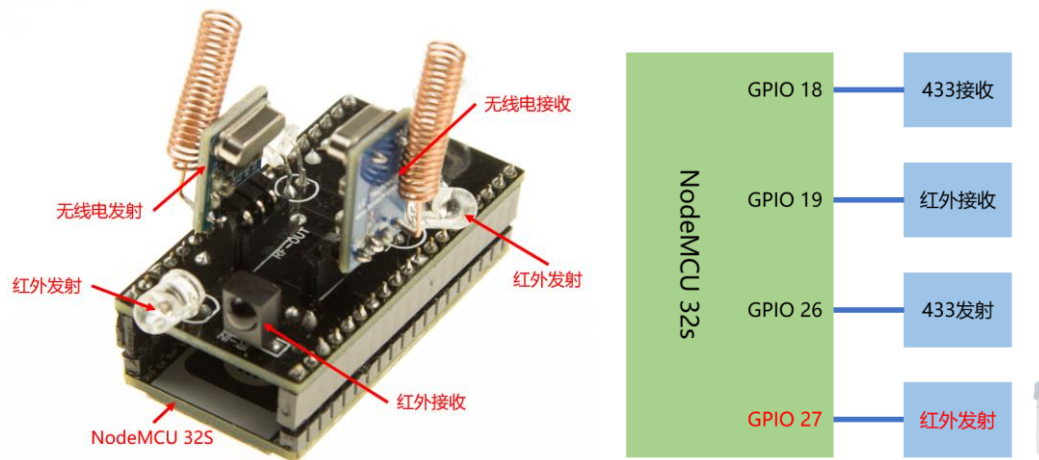
[《ESPHome 中的自动化基础》](#)

[《音乐灯带——接入 ESPHome》\(1\) \(2\)](#)

[《精彩众创-Bobo》ESPHome 系列](#)

使用红外与无线模块 (2) —— 红外发射

【硬件】



【操作步骤】

1. 配置红外开关 (红外发射)

```
remote_transmitter:  
  - pin: GPIO27  
    carrier_duty_percent: 50%  
    id: ir_sender
```

```
switch:
```

```
- platform: template  
  name: "Sony 电视开关"  
  turn_on_action:  
    - remote_transmitter.transmit_sony:  
      transmitter_id: ir_sender  
      data: 0x00000A90  
      nbits: 12  
- platform: template  
  name: "Sharp 电视开关"  
  turn_on_action:  
    - remote_transmitter.transmit_raw:  
      transmitter_id: ir_sender  
      carrier_frequency: 38kHz  
      code: [293, -1806, 292, -757, 291, -759, 292, -758, 317, -734, 292,  
-758, 291, -1686, 414, -1750, 373, -732, 293, -1706, 392, -757, 293, -757, 293,  
-757, 294, -1727, 373, -757, 292]  
    - delay: 40ms  
    - remote_transmitter.transmit_raw:  
      transmitter_id: ir_sender  
      carrier_frequency: 38kHz  
      code: [292, -1805, 305, -745, 317, -732, 293, -757, 293, -758, 293,  
-1805, 295, -756, 294, -755, 316, -1782, 294, -756, 294, -1807, 291, -1807, 292,  
-1807, 319, -730, 293, -1806, 318]
```

2. 配置 HomeAssistant 中的服务 (红外发射)

```
api:  
  services:  
    - service: toggle_sony_tv  
      then:  
        - remote_transmitter.transmit_sony:  
          transmitter_id: ir_sender  
          data: 0x00000A90
```

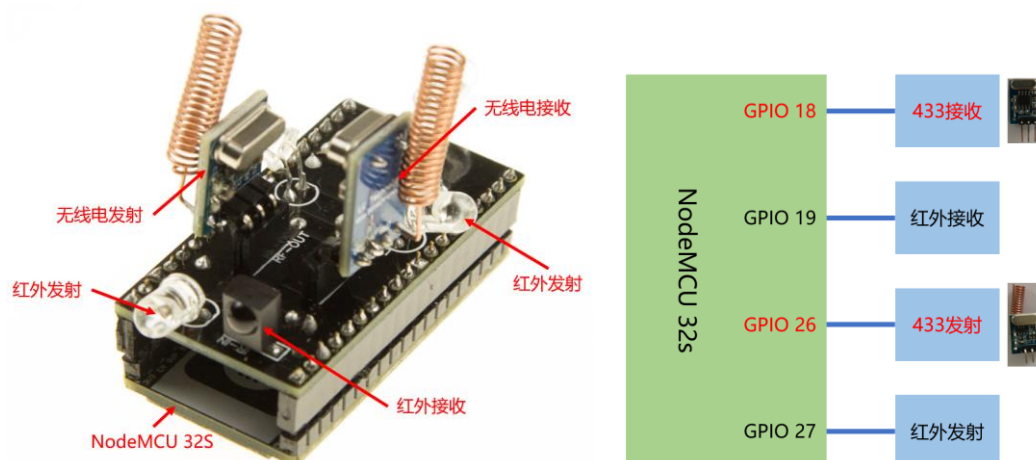
```
        nbits: 12
- service: toggle_sharp_tv
  then:
    - remote_transmitter.transmit_raw:
      transmitter_id: ir_sender
      carrier_frequency: 38kHz
      code: [293, -1806, 292, -757, 291, -759, 292, -758, 317, -734,
292, -758, 291, -1686, 414, -1750, 373, -732, 293, -1706, 392, -757, 293, -757,
293, -757, 294, -1727, 373, -757, 292]
    - delay: 40ms
    - remote_transmitter.transmit_raw:
      transmitter_id: ir_sender
      carrier_frequency: 38kHz
      code: [292, -1805, 305, -745, 317, -732, 293, -757, 293, -758,
293, -1805, 295, -756, 294, -755, 316, -1782, 294, -756, 294, -1807, 291, -1807,
292, -1807, 319, -730, 293, -1806, 318]
```

【参考】

- esphome 中的 Remote Transmitter
https://esphome.io/components/remote_transmitter.html
- esphome 中的 Api Service
<https://esphome.io/components/api.html#api-services>
- sharp 遥控器编码协议
<https://www.sbprojects.net/knowledge/ir/sharp.php>

使用红外与无线模块 (3) ——无线收发

【硬件】



【操作步骤】

1. 无线控制信号编码范式

- 最常用的编码方式: https://github.com/sui77/rc-switch/wiki/KnowHow_LineCoding
- esphome 中的定义代码:
https://github.com/esphome/esphome/blob/dev/esphome/components/remote_base/rc_switch_protocol.cpp

2. 无线发射配置

```
remote_transmitter:  
  - pin: GPIO26  
    carrier_duty_percent: 100%  
    id: rf_sender  
  
switch:  
  - platform: template  
    name: "test1"  
    turn_on_action:  
      - remote_transmitter.transmit_rc_switch_raw:  
          transmitter_id: rf_sender  
          code: '01010101111'  
          protocol: 1  
          repeat:  
            times: 3  
            wait_time: 50ms  
  - platform: template  
    name: "test2"  
    turn_on_action:  
      - remote_transmitter.transmit_rc_switch_raw:  
          transmitter_id: rf_sender  
          code: '01010101111'  
          protocol:  
            pulse_length: 350  
            sync: [1, 31]  
            zero: [1, 3]  
            one: [3, 1]  
          repeat:  
            times: 3
```

```
wait_time: 50ms
```

3. 无线接收

```
remote_receiver:  
  - pin:  
    number: GPIO18  
    inverted: no  
    mode: INPUT_PULLDOWN  
    dump: rc_switch  
    id: rf_receiver
```

4. 使用原始 (raw) 信号

为什么要使用 raw 信号：在接收解析中，rc_switch 仅解析协议 1-7，一些设备的信号并不完全符合这 7 个协议

难度：杂波与干扰

解决：将 raw 信号画成图形，人工判断协议与传输的信息

硬件制作（1）——画电路图

【操作步骤】

1. EDA 软件

Altium Designer: <https://www.altium.com/altium-designer/>

EasyEDA: <https://easyeda.com/>

EasyEDA 中文: <https://lceda.cn>

2. 登录、创建新的项目

<https://easyeda.com/zhujiasheng/hong-wai-yu-wu-xian-mu-kuai>

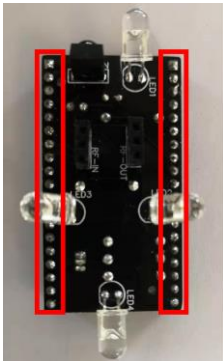
或: <https://easyeda.com/zhujiasheng/test>

3. 创建 Schematic (电路图)

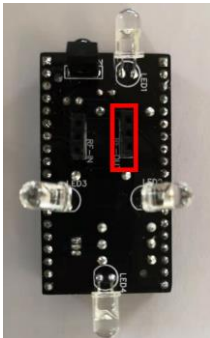
4. 画电路图

注意: 每个元器件可以选择错误的型号与参数, 但必须选择正确的尺寸

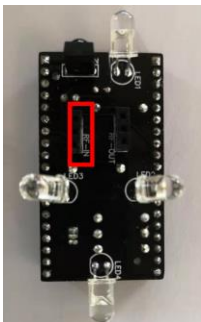
- 设计 NodeMcu32S 的对插连接器



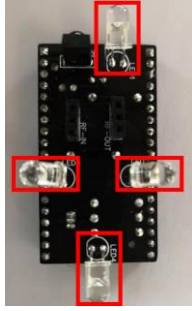
- 设计无线发射模块的连接器



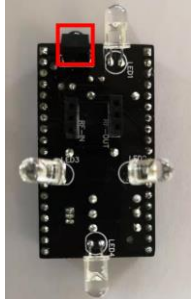
- 设计无线接收模块的连接器



- 设计红外发射管电路



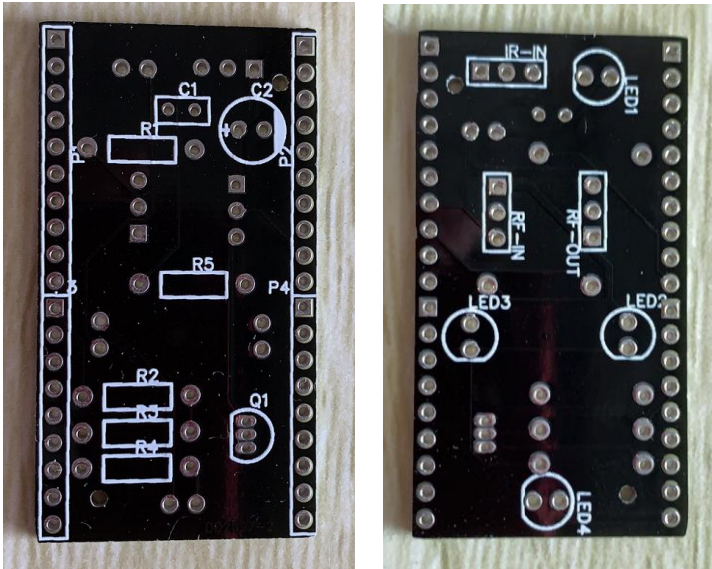
■ 设计红外接收管电路



硬件制作 (2) ——画 PCB 板

【操作步骤】

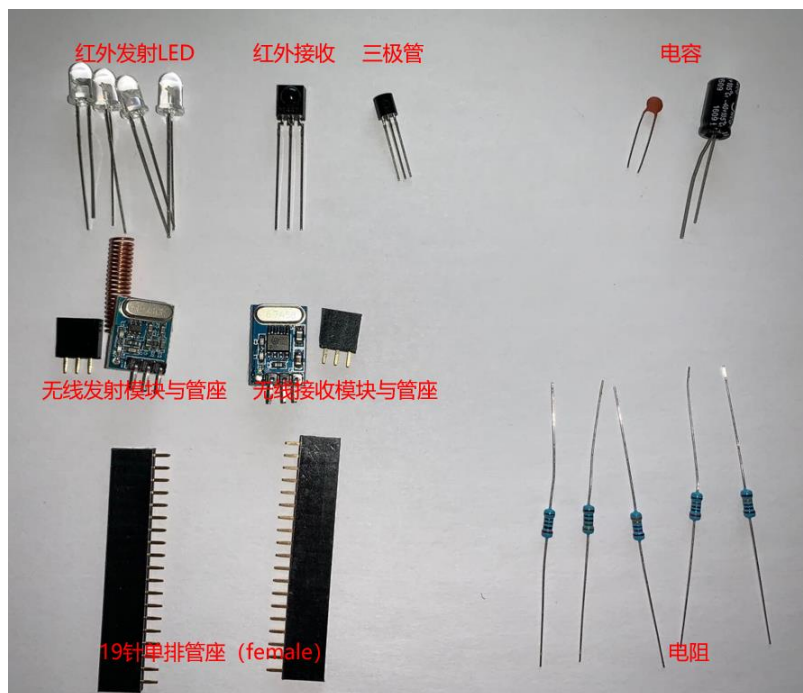
1. 将电路图转换成 PCB 布线图
设置 PCB 板的形状、大小、层数
2. 元器件部署
与 esp32 的连接器 (正面)
红外发射 led、红外接收器 (反面)
无线发射与无线接收连接器 (反面)
部署电阻、电容、三极管等元器件 (正面)
3. 布线
4. 印刷文字
5. 导出文件
6. 寻找加工厂



硬件制作 (3) —— 元器件与焊接

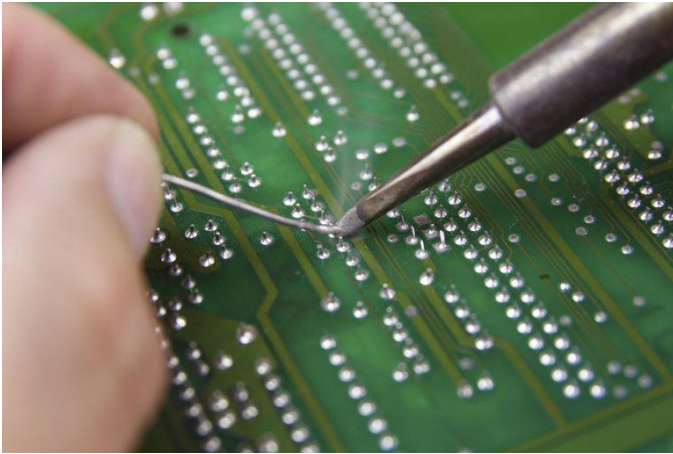
【操作步骤】

1. BOM 表



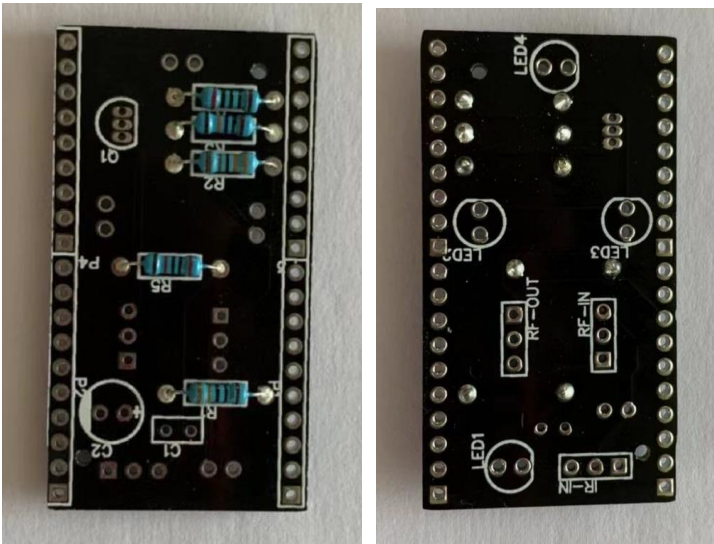
类型	设计文件中名称	元器件	数量	备注
Header 管座	J1、J2、P1、P2	Header-Female-2.54_1x19	2	19 针 2.54 间距排针母座
	RF-OUT、RF-IN	Header-Female-2.54_1x3	2	3 针 2.54 间距排针母座
红外 模块	LED1-4	IR-Infrared-LED-5mm	4	5MM 的红外发射管
	IR-IN	IRM-56384	1	红外接收管
无线 模块	--	SYN115	1	射频发射模块 (433M 或 315M)
	--	SYN480R	1	射频接收模块 (433M 或 315M)
三极管	Q1	SS8050	1	NPN 型三极管
电容	C1	0.1uF-瓷片电容	1	瓷片电容
	C2	100uF-25V-电解电容	1	电解电容
电阻	R1	330Ω-1/4W-金属膜电阻	1	射频发射 DATA 端限流电阻
	R2	3.3kΩ-1/4W-金属膜电阻	1	三极管基极限流电阻
	R3	100kΩ-1/4W-金属膜电阻	1	三极管基极下拉电阻
	R4	180Ω-1/4W-金属膜电阻	1	并联红外发射管限流电阻
	R5	20kΩ-1/4W-金属膜电阻	1	红外接收 DATA 端上拉电阻

2. 锡焊直插件的标准姿势

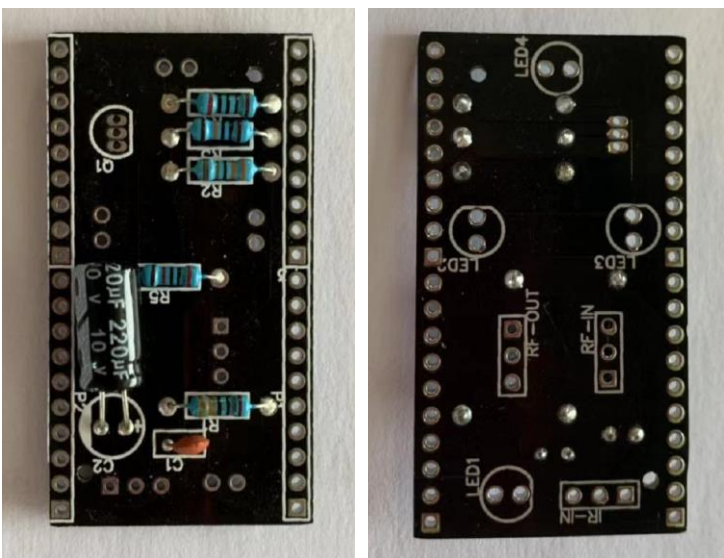


3. 焊接元器件

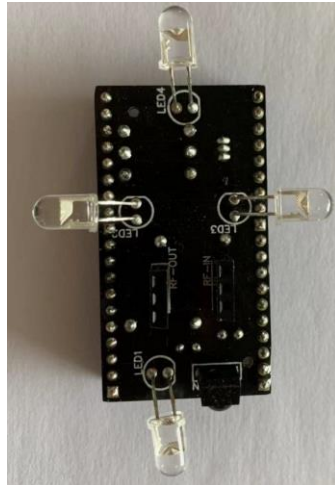
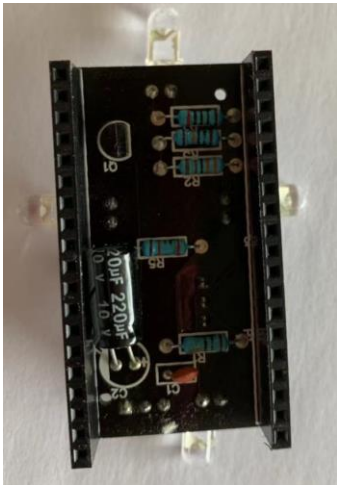
■ 电阻



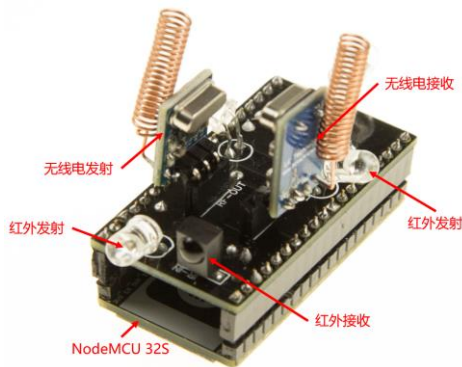
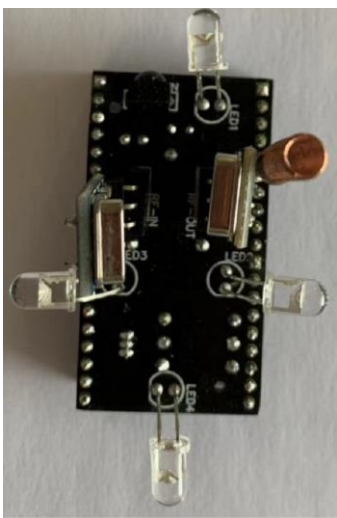
■ 电容



■ 三极管与红外接收管



■ 成品



【参考】

- 焊接那点事儿
<http://www.shaoguoji.cn/2017/03/14/welding-skill/>
- 元器件的各种封装

