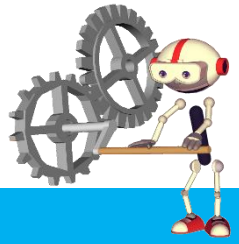


2021年第10届广东省创意机器人大赛培训

# 编程型机器人
# 执行器模块编程

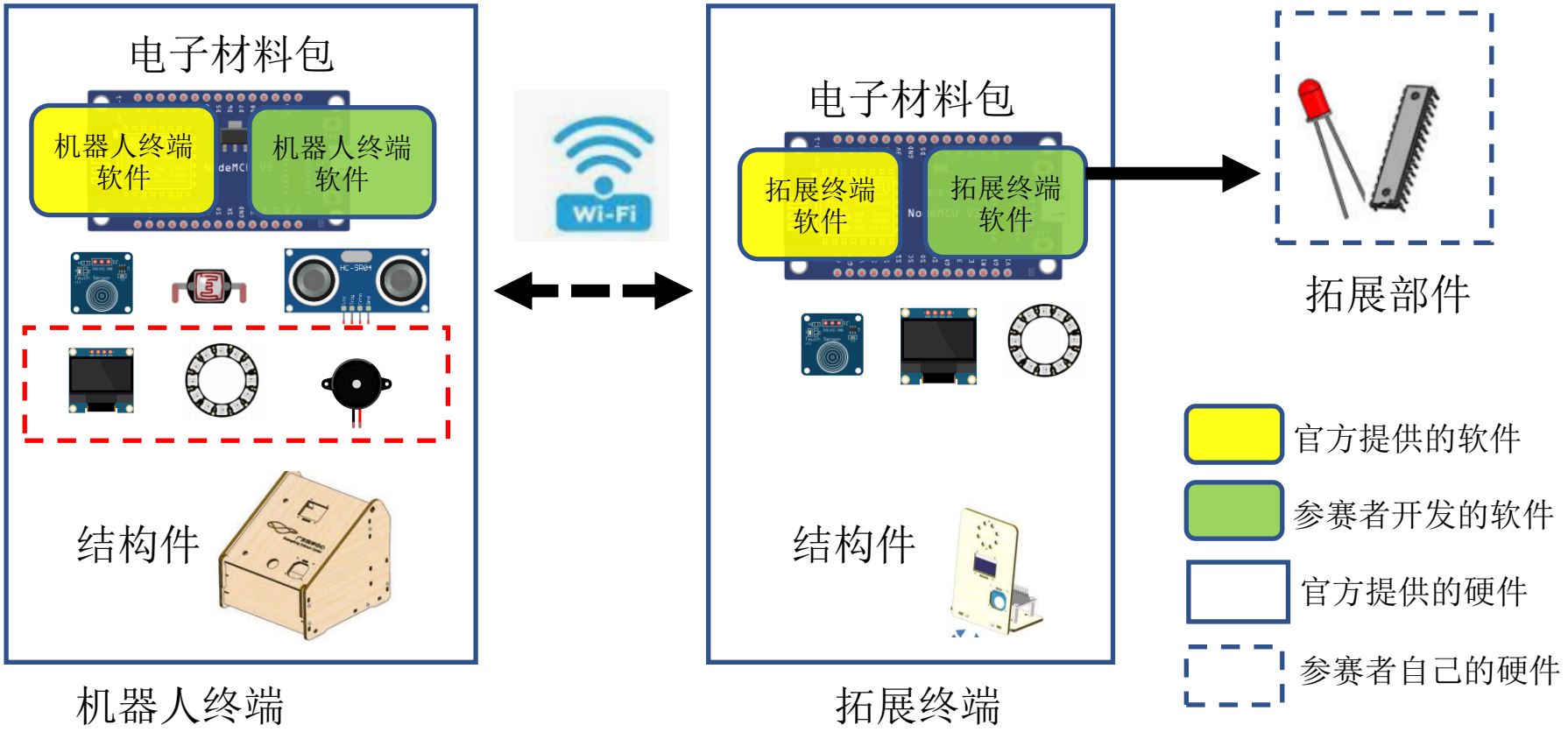
朱金辉

华南理工大学 软件学院

智能软件与机器人研究室

2021年7月

# 机器人的系统组成

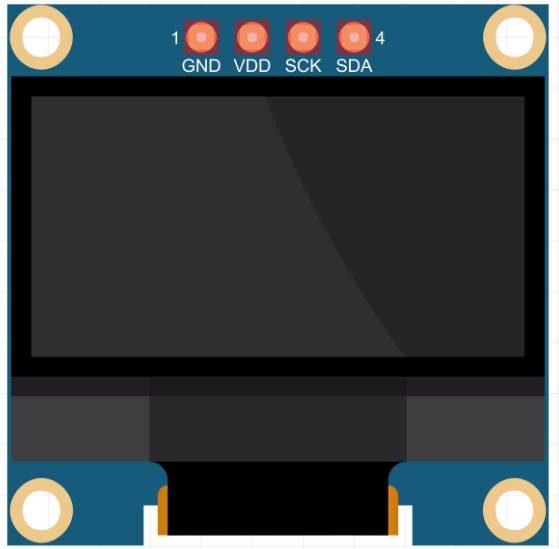

电子材料包
机器人终端软件
机器人终端软件
结构件
机器人终端
Wi-Fi
电子材料包
拓展终端软件
拓展终端软件
结构件
拓展终端
拓展部件
官方提供的软件
参赛者开发的软件
官方提供的硬件
参赛者自己的硬件

# 提纲

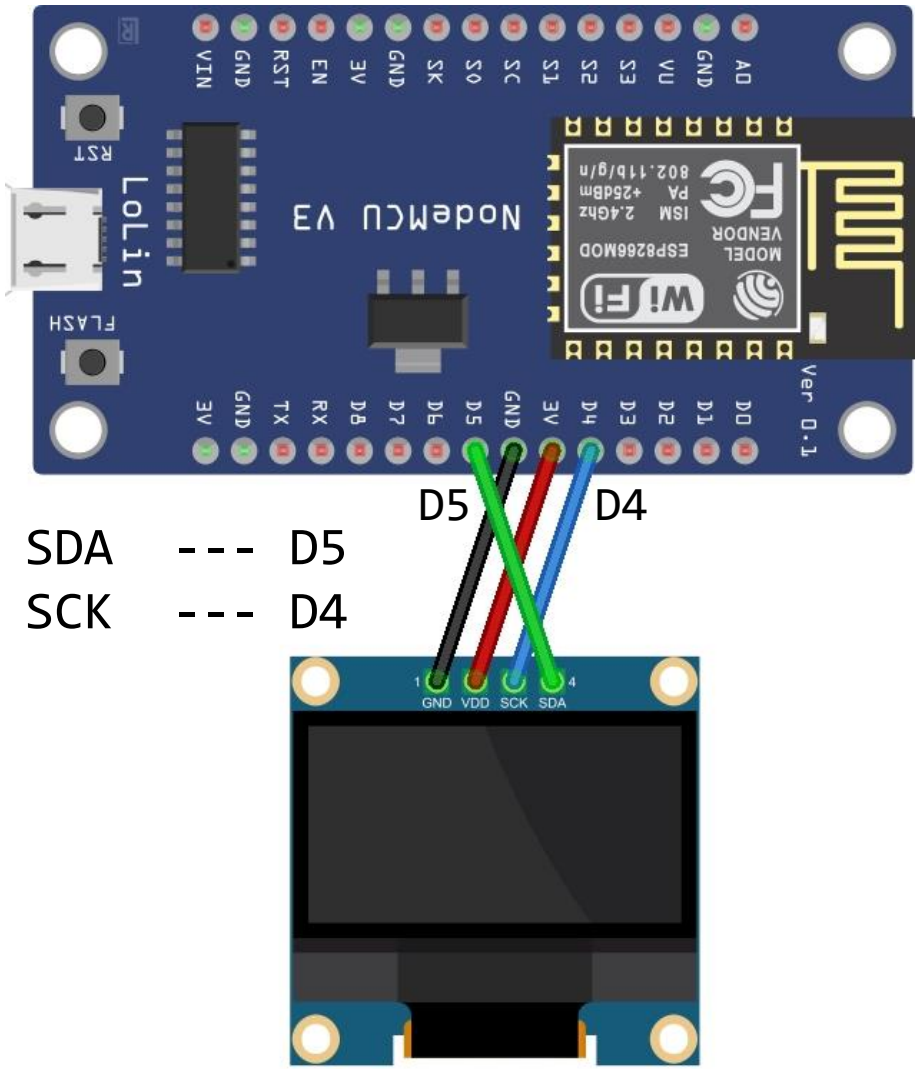
1. OLED显示屏
2. LED灯环
3. 蜂鸣器

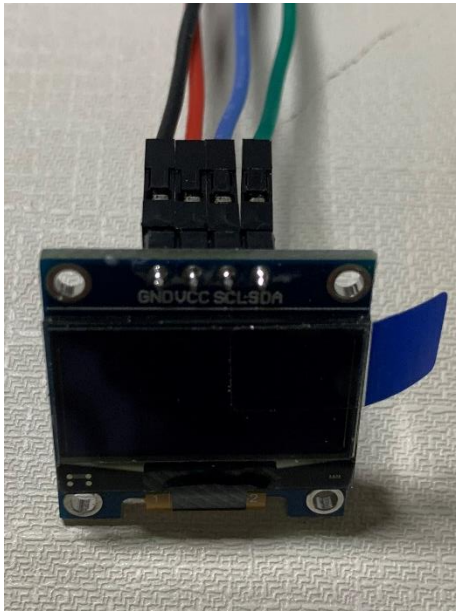
# 1. OLED显示屏

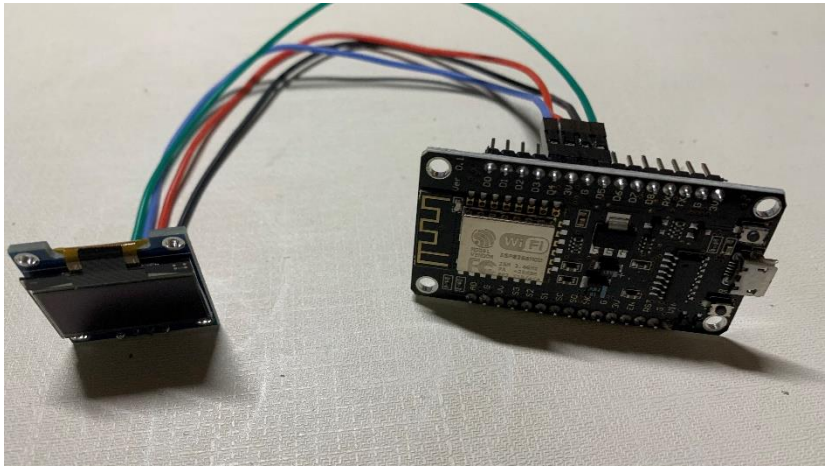
- OLED （Organic Light-Emitting Diode）即有机发光二极管。
- OLED显示技术与传统的LCD显示方式不同，无需背光灯，采用非常薄的有机材料涂层和玻璃基板(或柔性有机基板)，当有电流通过时，这些有机材料就会发光。
- 功能：显示信息



# 电路连接

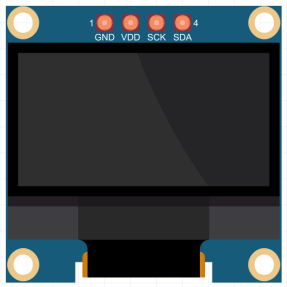

VIN GND RST EN 3V GND SK S0 SC S1 S2 S3 VU GND A0
RST
FLASH
NodeMCU V3
Lolin
3V GND TX RX D8 D7 D6 D5 GND 3V D4 D3 D2 D1 D0
Ver 0.1
D5
D4
SDA --- D5
SCK --- D4
GND VDD SCK SDA

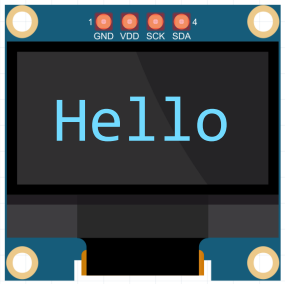

GND VCC SCL SDA

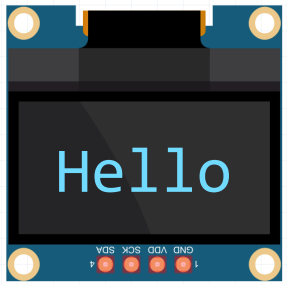


# 初始化

```
1  import time
2  import ssd1306
3  from machine import I2C, Pin
4
5  # display
6  i2c = I2C(scl=Pin(2), sda=Pin(14)) # D4 D5
7  display = ssd1306.SSD1306_I2C(128, 64, i2c)
8  # flip screen
9  display.write_cmd(0xA0) # SEGREMAP
10 display.write_cmd(0xC0) # COMSCANINC
11
12 display.fill(0)
13 display.show()
```



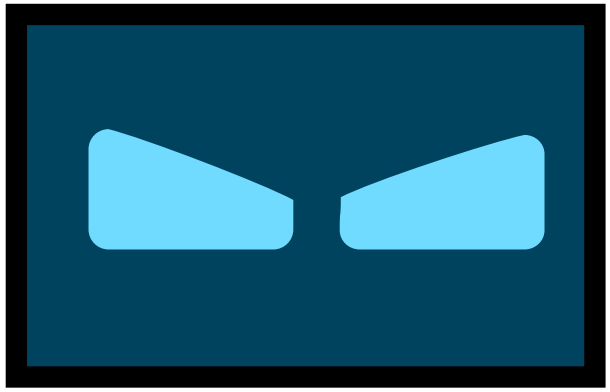



# 显示文本


```
1 display.fill(0)
2 display.text("Hello World",16,30,1)
3 display.show()
4
5
6
7 display.fill(0)
8 display.text("Hello World",16,56,1)
9 display.show()
```


# 显示图片



```
1 display.fill(0)
2 drawImageFile("img/02.pb",0,0)
3 display.show()
```

# drawImageFile函数

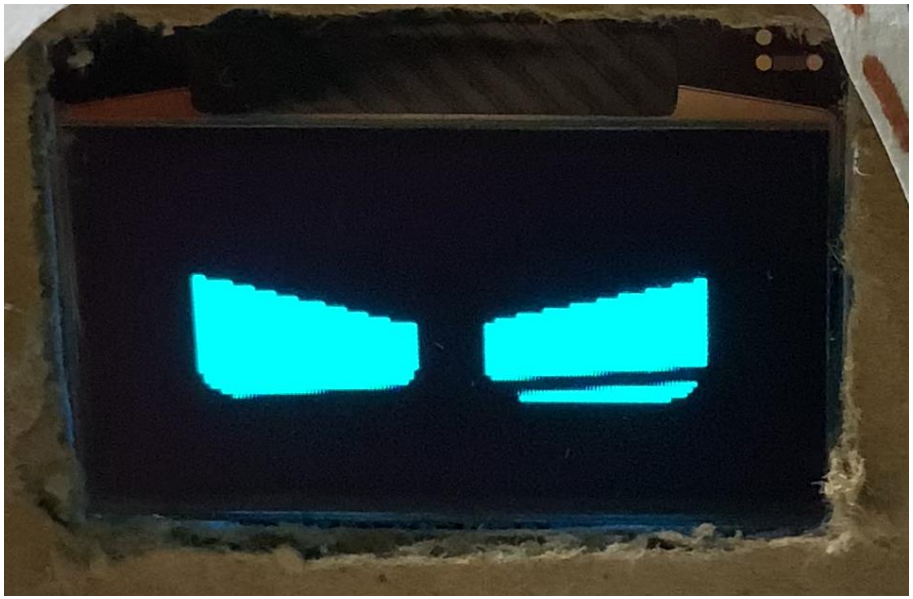
```
# draw an image from buf
def drawImage(rows, cols, buf, offset_col, offset_row):
    buf_index = 0
    for y in range(rows):
        Y = y + offset_row
        BUF_index = 128*Y + offset_col
        if BUF_index > 1023:
            return
        for x in range(cols):
            display.buffer[BUF_index + x] = buf[buf_index]
            buf_index += 1

# draw a picture from a img file
def drawImageFile(imgFile,offset_col, offset_row):
    try:
        f = open(imgFile,'rb')
        data = f.read(1)
        rows=int.from_bytes(data, "big")
        data = f.read(1)
        cols=int.from_bytes(data, "big")
        buf=f.read(rows*cols)
        f.close()
        drawImage(rows,cols,buf,offset_col, offset_row)
        del(buf)
    except OSError as er:
        print(er)
```

# 实验1：显示

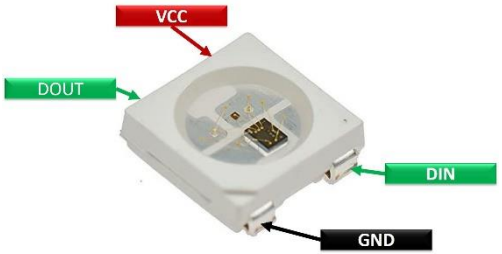
```
while True:
    for i in range(1,4):
        display.fill(0)
        drawImageFile("img/0"+str(i)+".pb",0,0)
        display.show()
        time.sleep(1)
```

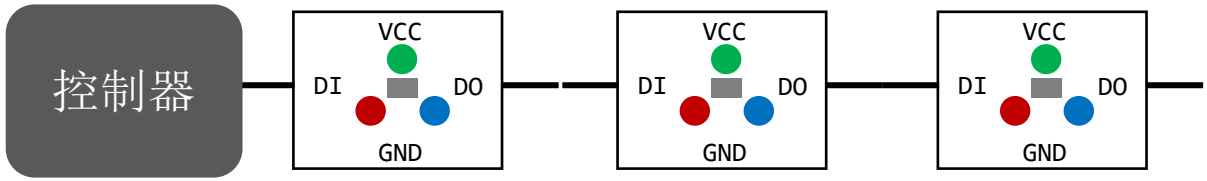
视频



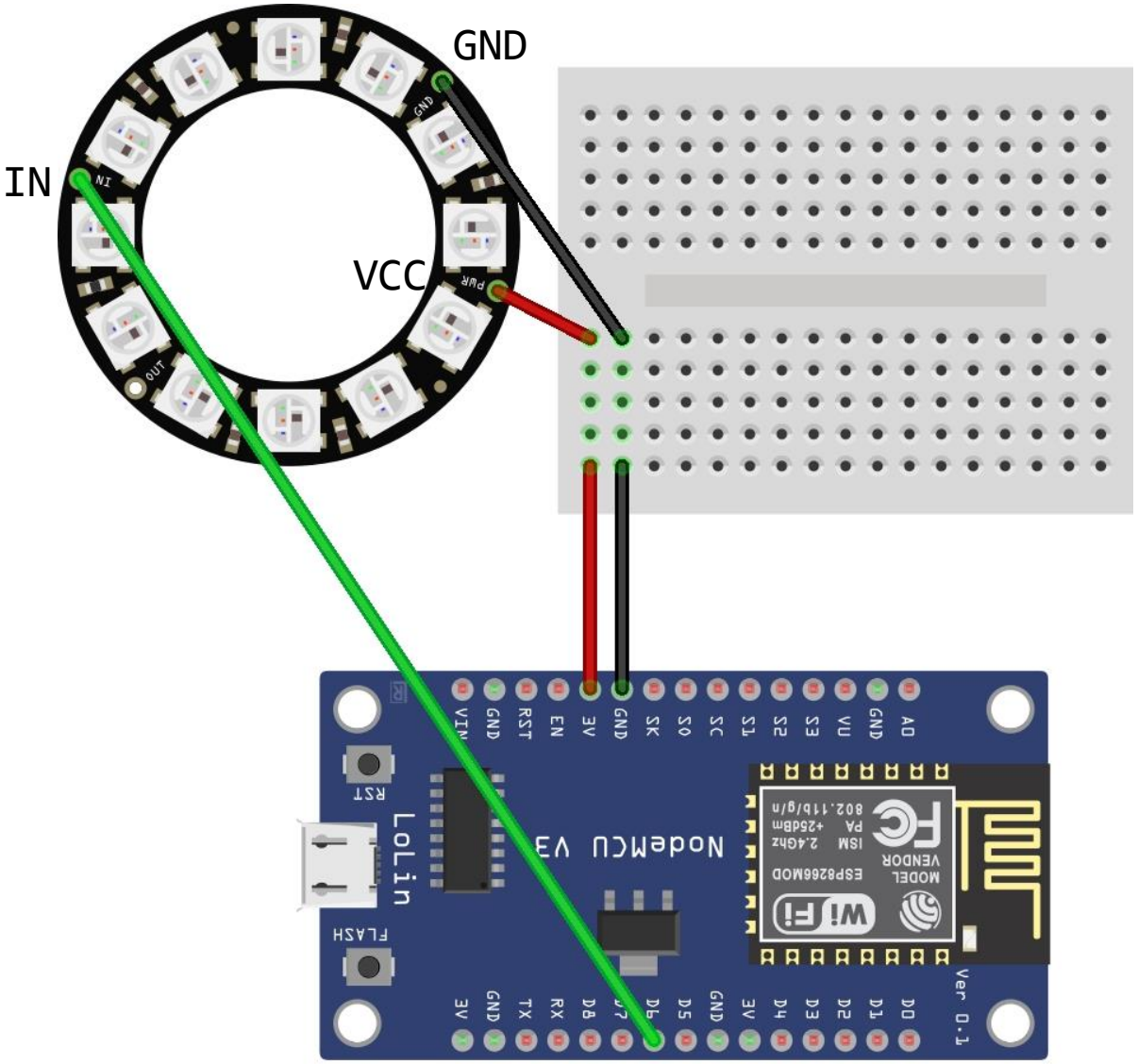
# 2. RGBLed环

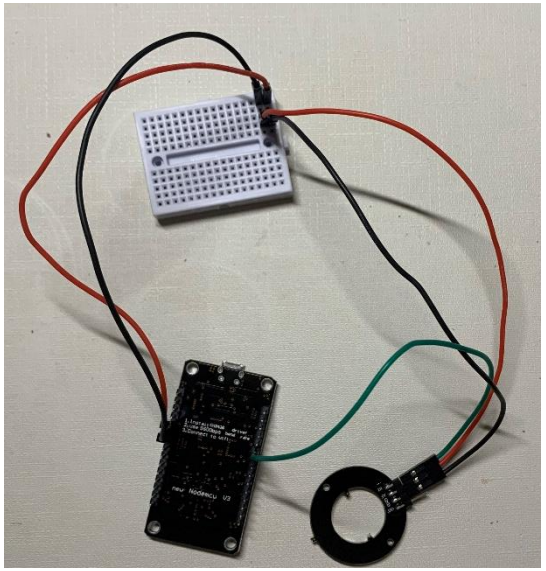
- 发光二极管（light-emitting diode，缩写为LED）是能发光的半导体电子器件。
- 当电流流过时，电子与空穴在其内复合而发出单色光，这叫电致发光效应。
- 而光线的波长、颜色跟其所采用的半导体物料种类与故意掺入的元素杂质有关。
- WS2812B芯片使用的是单一的数据线和一个时间具体（timing-specific）协议。



# 电路连接



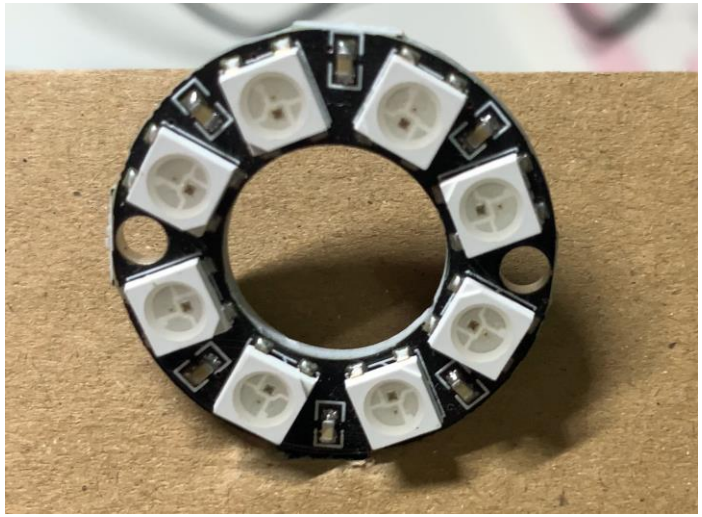



# 代码

```
1 import neopixel
2 from machine import Pin
3
4 # D6 GPIO12
5 led = neopixel.NeoPixel(Pin(12), 8) # D6
6 led[0] = (255, 0, 0) # set to red, full brightness
7 led[1] = (0, 128, 0) # set to green, half brightness
8 led[2] = (0, 0, 64)  # set to blue, quarter brightness
9 led.write()
```

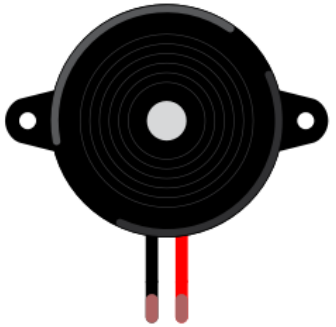
# 实验2：灯环

```
1  import neopixel
2  from machine import Pin
3  import time
4
5  # D6 GPIO12
6  led = neopixel.NeoPixel(Pin(12), 8) # D6
7
8  ledCount = 0
9  while True:
10     if ledCount < 8:
11         led[ledCount] = (0,0,255)
12         ledCount += 1
13     else:
14         ledCount = 0
15         for i in range(8):
16             led[i] = (0,0,0)
17     led.write()
18     time.sleep(1)
```

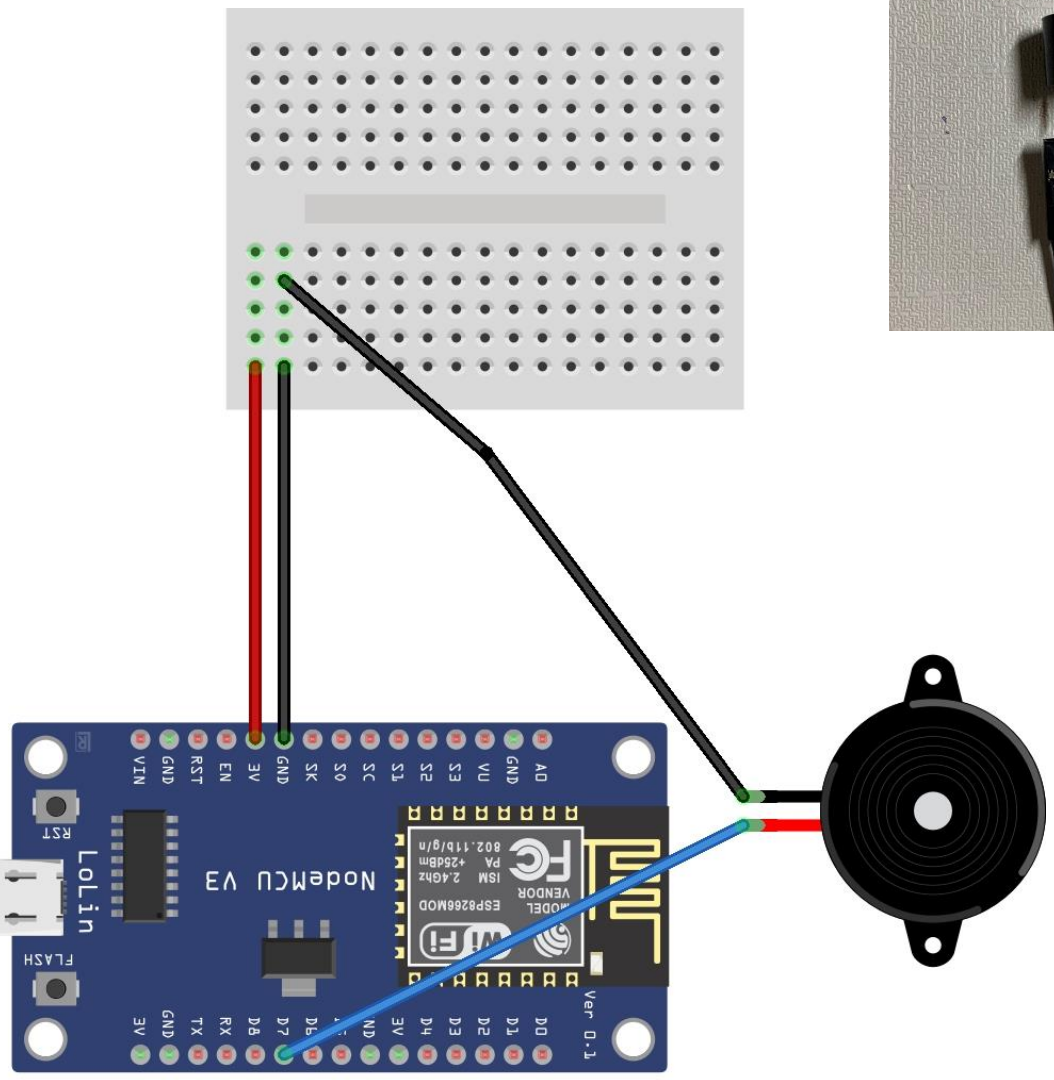


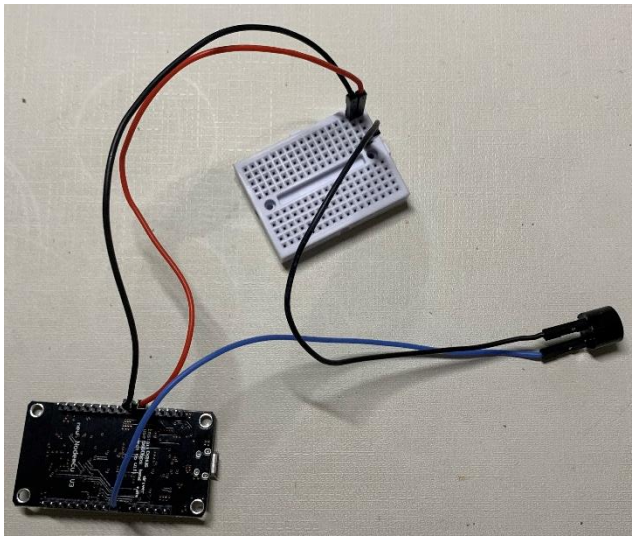
# 3. 蜂鸣器

- 蜂鸣器（Buzzer）是产生声音的信号装置。
- 小型蜂鸣器大都是压电式蜂鸣器，主要依靠压电效应来产生振动，发出声音。
- 不同频率的方波电信号输入谐振装置转换为不同的声音信号输出。
- 功能：发声



# 电路连接图



# 简单发音

```
1 from machine import Pin, PWM
2 import time
3
4 # beeper D7 GPIO13
5 beeper = PWM(Pin(13, Pin.OUT), freq=600, duty=0)
6
7 beeper.duty(512)
8 time.sleep_ms(2)
9 beeper.duty(0)
```

# 连续发音

```
from machine import Pin, PWM
import time

beeper = PWM(Pin(13, Pin.OUT), freq=440, duty=512) # D7 GPIO13

tempo = 5
tones = {
    'c': 262,
    'd': 294,
    'e': 330,
    'f': 349,
    'g': 392,
    'a': 440,
    'b': 494,
    'C': 523,
    ' ': 0,
}
melody = 'cdefgabC'
rhythm = [8, 8, 8, 8, 8, 8, 8, 8]

for tone, length in zip(melody, rhythm):
    beeper.freq(tones[tone])
    time.sleep(tempo/length)

beeper.deinit()
```

# 实验3：发音

```
from machine import Pin, PWM
import time

beeper = PWM(Pin(13, Pin.OUT), freq=440, duty=512) # D7 GPIO13

tempo = 5
tones = {
    'c': 262,
    'd': 294,
    'e': 330,
    'f': 349,
    'g': 392,
    'a': 440,
    'b': 494,
    'C': 523,
    ' ': 0,
}
melody = 'cdefgabC'
rhythm = [8, 8, 8, 8, 8, 8, 8, 8]

for tone, length in zip(melody, rhythm):
    beeper.freq(tones[tone])
    time.sleep(tempo/length)

beeper.deinit()
```

# 谢谢！