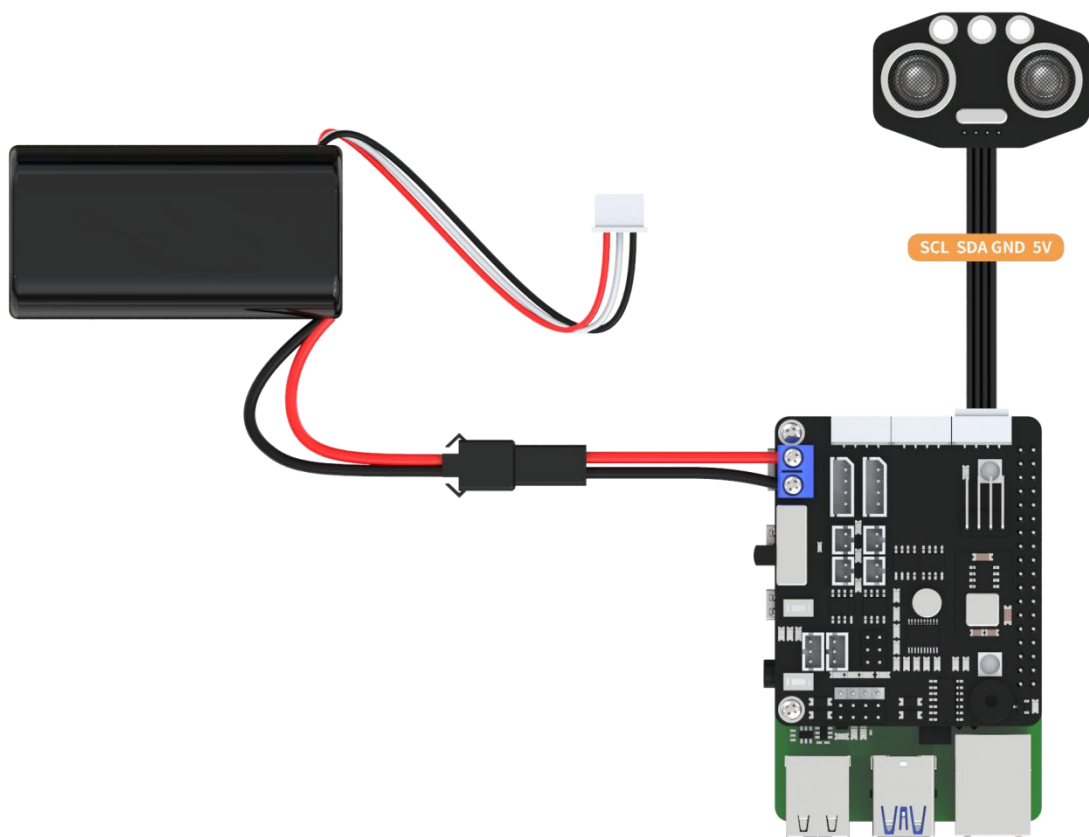


# 第 6 课 控制超声波传感器颜色

## 1. 硬件连接

将超声波传感器通过 4pin 线连接至树莓派扩展板的任意一个 IIC 接口，如下图所示：



## 2. 实现原理

发光超声波除了拥有测距的功能外，还自带 RGB 彩灯。我们通过高低电平的变化控制 RGB 亮灭，改变各颜色分量数值控制显示的灯光颜色，便可以实现对它的控制。

该程序的源代码位于：5.附录->2.源码->EXAMPLE->sonar\_rgb.py

```
90 while True:
91     print('Color wipe animations.')
92     colorWipe(Board.PixelColor(255, 0, 0)) # Red
93     wipe
94     colorWipe(Board.PixelColor(0, 255, 0)) #
95     Blue wipe
96     time.sleep(1)
97     colorWipe(Board.PixelColor(0, 0, 255)) #
98     Green wipe
99     time.sleep(1)
00     print('Theater chase animations.')
01     theaterChase(Board.PixelColor(127, 127, 127))
02     # White theater chase
03     time.sleep(1)
04     theaterChase(Board.PixelColor(127, 0, 0)) #
05     Red theater chase
06     time.sleep(1)
07     theaterChase(Board.PixelColor(0, 0, 127)) #
```

本节玩法主要调用了 Board 库中的 PixelColor()函数和传感器获取的数据来实现控制超声波 RGB 的颜色，并通过 colorWipe()函数来实现转换颜色的效果。

以代码“colorWipe(Board.PixelColor(255, 0, 0))”为例，其中：

第一个参数“255”代表“R”通道（红色分量）的数值，R 值越大，红色越深；

第二个参数“0”代表“G”通道(绿色分量)的数值，G 值越大，绿色越深；

第三个参数“0”代表“B”通道（蓝色分量）的数值，B 值越大，蓝色越深。

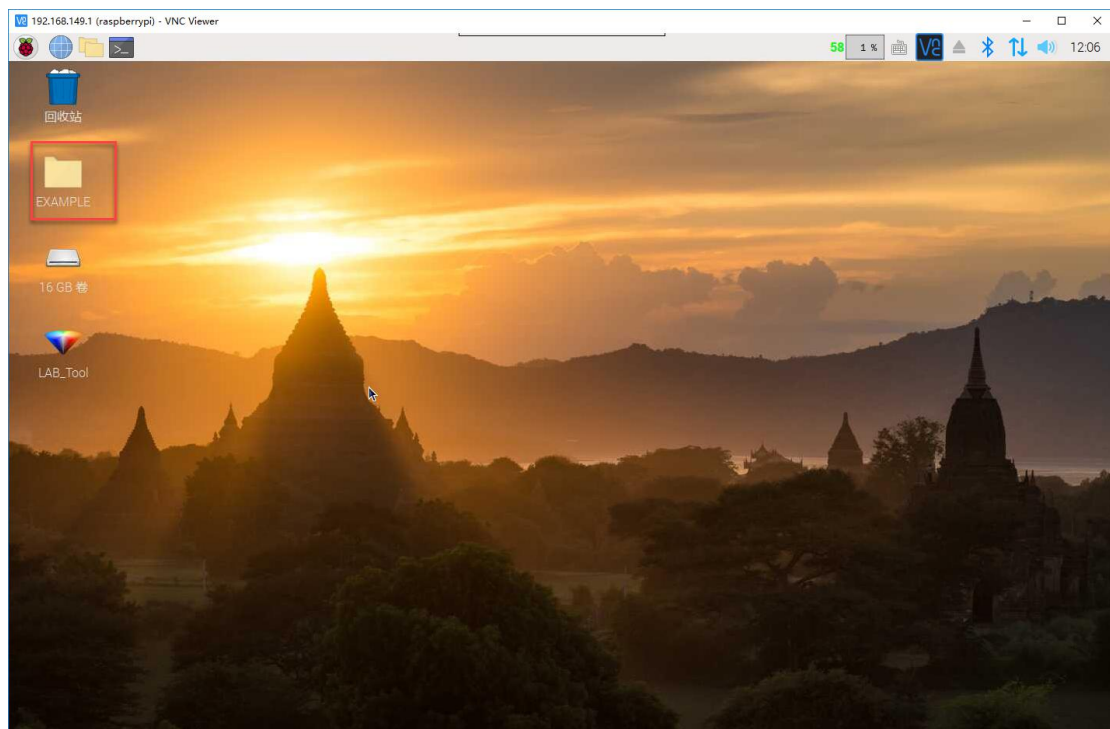
### 3.实验流程

---

本节程序必须与 HiwonderSDK 文件夹（底层文件）放在同一目录下才可启动！

---

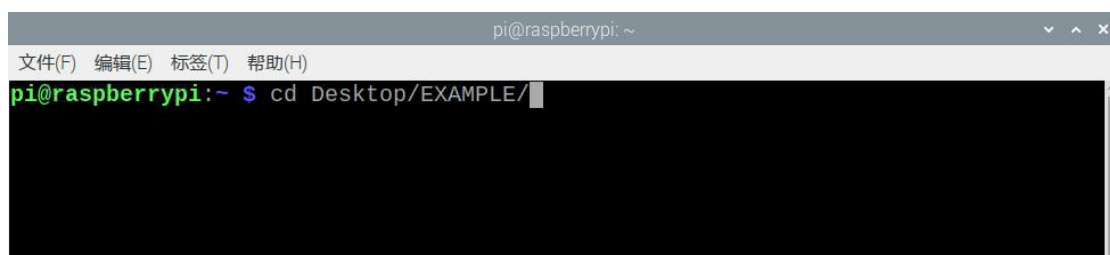
1) 将资料内的“5.附录->2.源码->EXAMPLE”内的 EXAMPLE 文件夹通过 U 盘或者远程传输的方式导入至树莓派，然后放置在用户所需要存储的位置。这里为了便于展示，以放置在系统桌面为例：



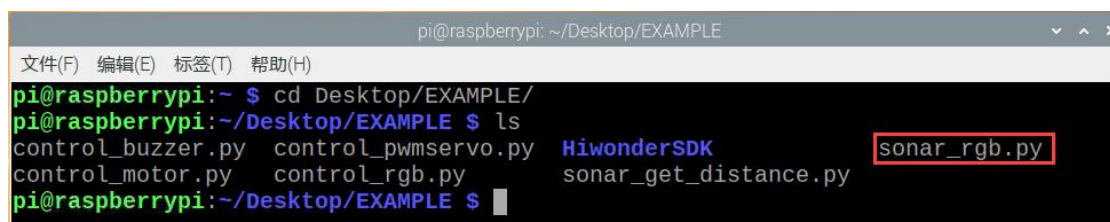
2) 点击下方图示位置进入终端命令行。



3) 本节程序是放置在桌面的 EXAMPLE 文件夹,输入命令“**cd Desktop/EXAMPLE/**”,然后按下回车进行切换。



4) 我们可用 ls 命令进行查看程序名称,输入命令“**ls**”,按下回车。如下图所示,这个“**sonar\_rgb.py**”即本节控制超声波传感器颜色的程序。



- 5) 输入命令“**sudo python3 sonar\_rgb.py**”即可直接启动本节程序。

## 4.功能实现

程序运行后，超声波传感器的 RGB 灯将进行不同颜色的闪烁。