

# Amazon Echo 控制红外遥控

玩

September 8, 2017

## 目录

步骤一: 材料	4
安装 <b>Raspberry Pi</b>	5
步骤三: 学习 <b>IR</b> 编码	6
步骤四: 耳机插孔 (可选)	10
步骤五: 连接红外 <b>LED</b> (简易)	12
步骤六: 连接红外 <b>LED</b> (正确方法)	14
步骤七: 测试发送 <b>IR</b> 命令	15
步骤八: 安装 <b>Ha-bridge</b>	16
步骤九: 模拟飞利浦色调灯泡	17
步骤十: 连接到 <b>Amazon Echo</b>	18

玩点什么: <https://www.wandianshenme.com>

原文链接: <https://www.wandianshenme.com/play/amazon-echo-□□□□□□>

**Amazon Echo** 系统可以用于智能家居的方方面面, 但智能插座只能关闭和或打开。许多设备不会通过简单插入而立即打开, 需要额外的步骤, 例如按遥控器上的按钮或物理设备上电需获取所需设置。

在本指南中, **Raspberry Pi Zero W** 将被配置为充当由 **Amazon Echo** 控制的智能家居设备, 并在请求打开或关闭电源时将任何所需的 **IR** 命令发送到设备。

在这种情况下, **Pi** 将被配置为学习由红外线石英管 **ClassicFlame 23II310GRA 23** 发送的远程 **IR** 命令, 然后将使用红外 **LED** 发出需要的 **IR** 命令, 最后 **Pi** 配置为可控制 **Echo** 的 **Philips Hue** 模拟设备。



玩点什么:

步骤一：材料



所需材料:

- Raspberry Pi Zero Z
- 4G 或者更大的 Micro SDHC Class 10 存储卡
- MicroUSB 线
- 一个 IR LED
- MicroUSB 接口的电源适配器 (推荐 2.1A 或者更大电流)
- IR LED
- IR 接收器 VS1838B
- 100  $\Omega$  电阻
- 少量导线

推荐:

- 耳机插孔
- 1/8 英寸耳机/音频线
- 2N2222 NPN 晶体管
- 1000  $\Omega$  电阻

- Raspberry Pi Zero Case 开发板

要完成 Raspberry Pi Zero W 的初始配置，还需要其他几个的外围设备，但不会贯穿于整个完整项目。

- Mini HDMI 线: 用于连接 Pi Zero W 和 TV 或显示器，线稍长一些较好。
- USB OTG 线: 用于将微型 USB 转换为全尺寸 USB 端口，用于连接键盘和/或鼠标
- HDMI 线: 用于连接电视或显示器以及适配器到 mini HDMI。

前两个步骤和一个案例包括各种 Pi Zero 入门套件，例如: [MakerSpot Mega Kit](#)。

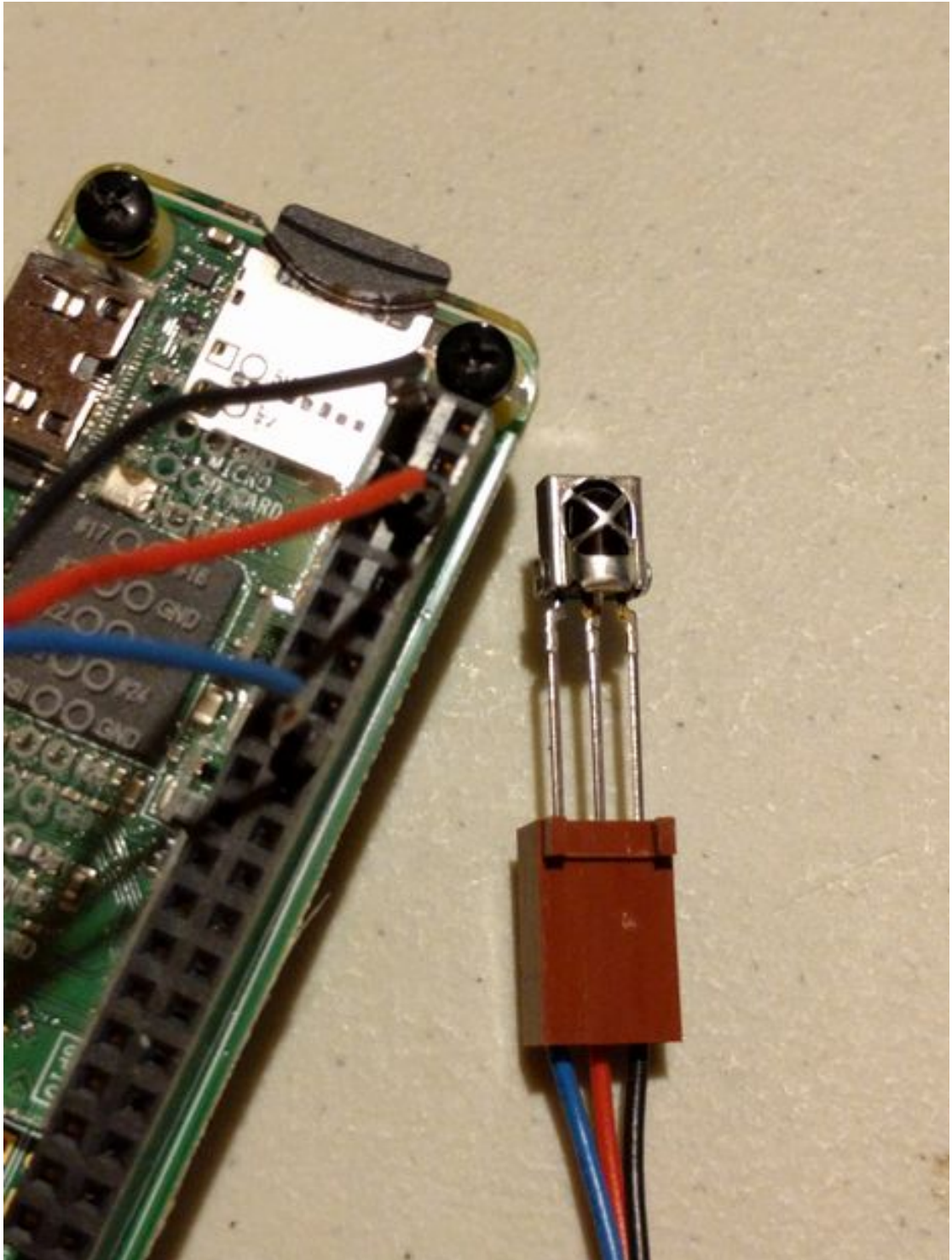
## 安装 Raspberry Pi

Raspberry Pi 网站关于在 Raspberry Pi 上安装 Raspbian 操作系统有一个很好的指导。如果您希望将来有更多的操作系统选项，或者更简单的安装，请按照 NOOBS 的说明进行操作。本指南是基于 Raspbian，它包含在 NOOBS 中，<https://www.raspberrypi.org/learning/software-guide/>

一旦 Raspbian 开始运行，启用 SSH 允许远程连接到设备，而不需要直接连接到 Pi 的监视器/键盘/鼠标。如果您希望可以远程访问 GUI，还可以启用 VNC 访问<https://www.raspberrypi.org/documentation/remote-access/ssh/>

此外还强烈建议您在无线网络配置上设置静态 IP，以便不随时间改变。如果新设备未定期连接到无线网络，但将其配置为静态 IP 可能会导致 IP 不会改变，否则将无法确定，<http://www.circuitbasics.com/how-to-set-up-a-static-ip-on-the-raspberry-pi/>

步骤三: 学习 IR 编码



玩点



以下步骤基于以下优秀指南: <http://alexba.in/blog/2013/01/06/setting-up-lirc-o...>(链接不可用, 可参考国内相关资料)。

Linux 红外遥控器 (LIRC) 库用于通过接收器模块处理接收到的 IR 命令, 将其保存到文件中, 然后通过 IR LED 发送命令。

第一步是使用红外线接收器记录来自现有遥控器的红外信号, 并将其保存到文件中。红外线接收器只需要初始学习 IR 信号, 然后可以关掉, 因此可以使用临时连接。

将 IR 接收器连接到 Raspberry Pi。使用附图标识 VCC, GND 和 Signal 引脚。使用面包板, 连接线, 或创意弯曲的插针完成以下连接:

- VCC 连接到 5V 引脚
- GND 连接到地
- Signal 连接到 23 号脚

通过打开本地设备上的终端或使用诸如 Putty 的程序创建 SSH 连接, 打开并连接到 Raspberry Pi。剩下的步骤将用命令行界面执行。

初始化 LIRC。

```
1 sudo apt-get install lirc
```

将需要的信息添加到模块文件中。

```
1 sudo nano /etc/modules
```

将以下内容添加到文件的末尾。

```
1 lirc_dev
2 lirc_rpi gpio_in_pin=23 gpio_out_pin=22
```

完成之后, 按 **CTRL+X**, 然后, **Y**, 保存并退出。

将以下代码加入到 **hardware.conf** 文件。

```
1 LIRCD_ARGS="--uinput
2 # Run "lircd --driver=help" for a list of supported drivers.
3 DRIVER="default"
4 # usually /dev/lirc0 is the correct setting for systems using udev
5 DEVICE="/dev/lirc0"
6
7 MODULES="lirc_rpi"
```

完成之后, 按 **CTRL+X**, 然后, **Y**, 保存并退出。

修改 **config.txt**, 以便引导启动时加载 **LIRC** 内核模块。添加以下内容到文件末尾。

```
1 sudo nano /boot/config.txt
2 dtoverlay=lirc-rpi,gpio_in_pin=23,gpio_out_pin=22
```

完成之后, 按 **CTRL+X**, 然后, **Y**, 保存并退出。

重启 **Raspberry Pi**。

```
1 sudo shutdown -r now
```

通过操作一次 **LIRC** 然后监视输入来测试 **IR** 接收器, 第一次 **LIRC** 一旦结束, **LIRC** 设备就能监测到数据。

```
1 sudo /etc/init.d/lirc stop
2 mode2 -d /dev/lirc0
```

在接收器上瞄准红外遥控器, 然后按一个按钮, 确保数据出现在屏幕上。

按 **CTRL+C** 停止。

此时, **LIRC** 程序已安装, 我们可以查看 **IR** 信息。现在, 可以从 **LIRC** 网站下载远程配置文件, 也可以使用自己的遥控器创建自定义配置文件。



在此过程中, 您将输入要记录的每个红外遥控器的按钮的名称, 必须是有效的名称, 因此运行以下命令查看所有可用的名称。

```
1 irrecord --list-namespace
```

示例: 当我记录遥控器上的 **Power** 按钮时使用名称 **KEY\_POWER**, 记录 **Timer** 按钮时使用 **KEY\_TIME** 名称。

如果已经记录了几个按键, 建议您将记录的名称和按键建立一一对应关系, 因为记录的按键可能没有完美的名称。这将使之更容易参考。

使用以下代码生成远程配置文件。

```
1 # Stop lirc to free up /dev/lirc0
2 sudo /etc/init.d/lirc stop
3 # Create a new remote control configuration file (using /dev/lirc0) and
   save the output to ~/lircd.conf
4 irrecord -d /dev/lirc0 ~/lircd.conf
```

按照屏幕上的说明进行操作。记录初始化完成后, 输入要使用的按键名, 然后按遥控器上的按钮并将其指向接收器, 直到出现几个点。对要记录的遥控器上的每个按钮并重复此步骤。

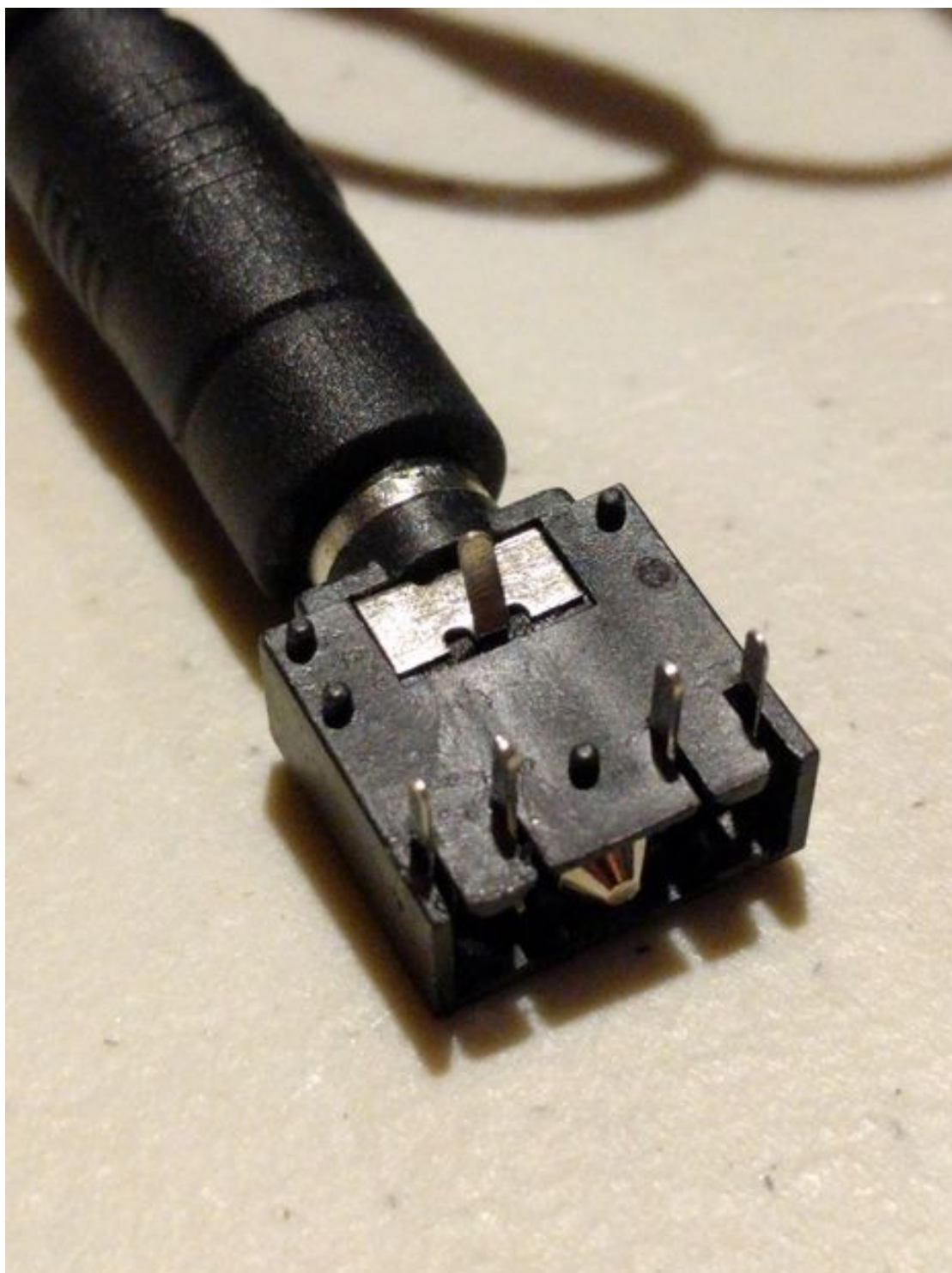
```
1 #Rename the remote in the configuration file
2 sudo nano ~/lircd.conf
```

将每行的开始修改为能够影射您正在记录的设备名字, 本例中, 它被重名为 **Spectrafire**。结束之后, 按 **CTRL+X** 退出并保存。

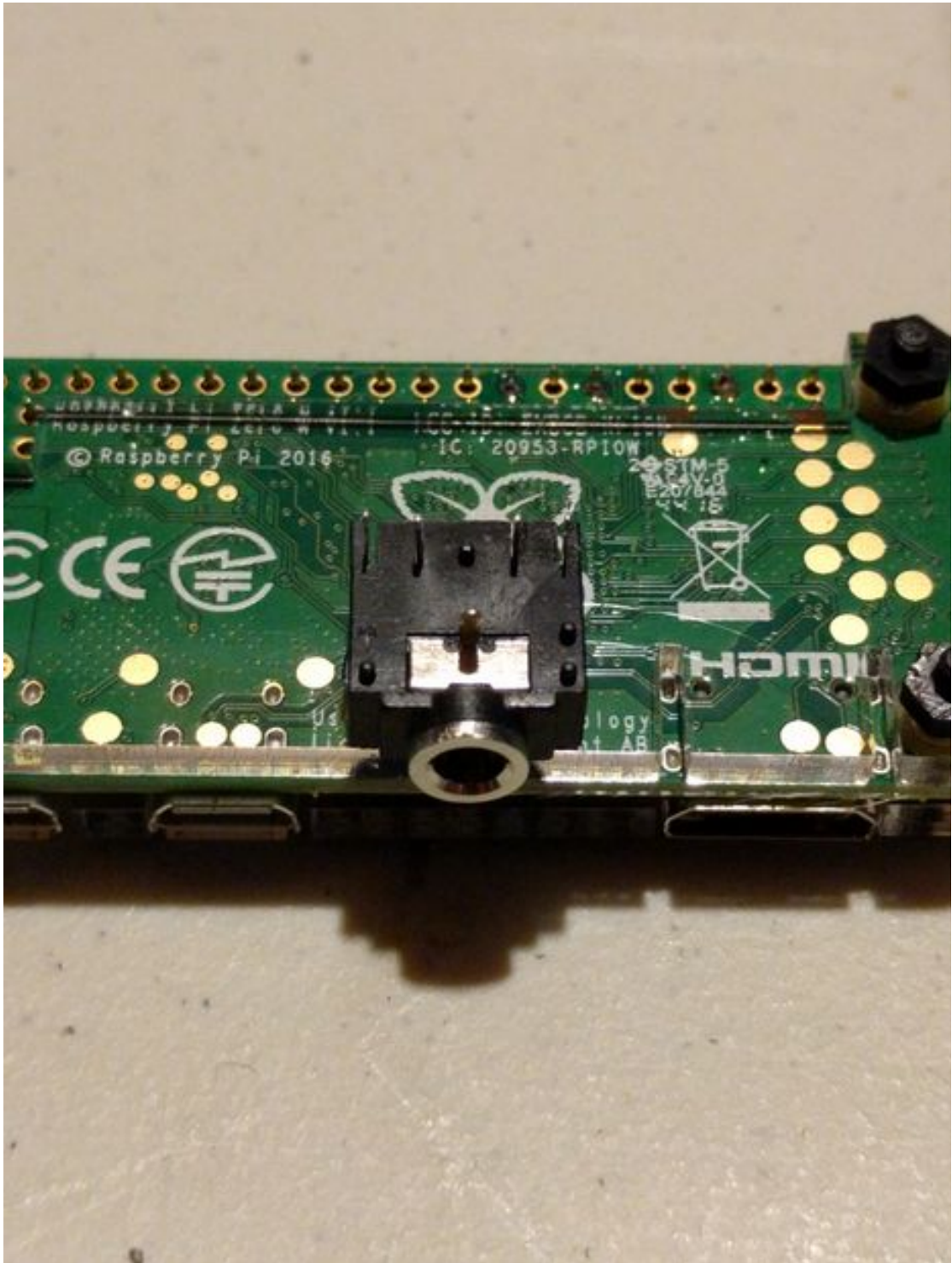
```
1 # Make a backup of the original lircd.conf file
2 sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd_original.conf
3 # Copy over your new configuration file
4 sudo cp ~/lircd.conf /etc/lirc/lircd.conf
5 # Start up lirc again
6 sudo /etc/init.d/lirc start
```

此时, 解码值已经记录在一个文件中。

步骤四: 耳机插孔 (可选)



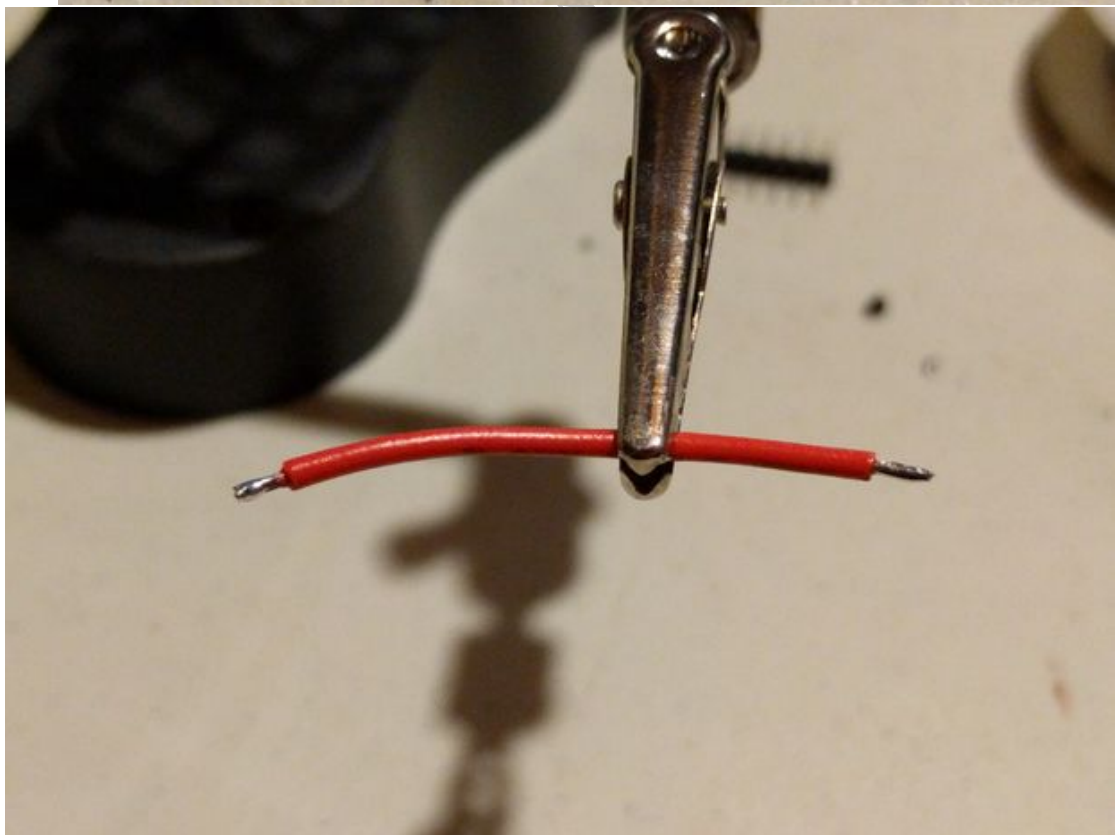
玩点

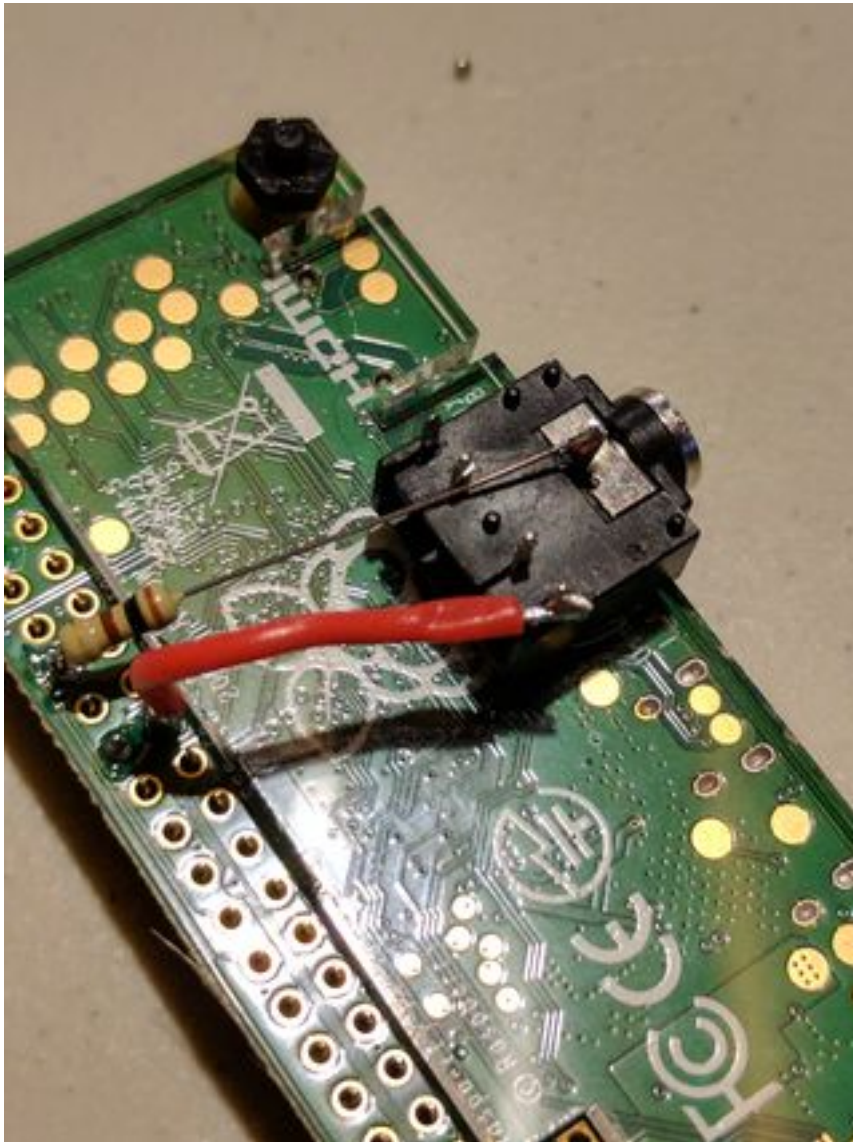


为确保使用的导线和修改 **Raspberry Pi** 更加模块化，我热粘了一个 1/8 英寸耳机音频插孔到外壳并连接电线到插孔。使用带有匹配插头的耳机线连接红外 **LED**，因此该导线可以连接到任何其他的位置，最后指向我想要连接的红外接收器设备，但是可以很容易地从 **Pi** 拔下，而不需要移除所有的电线。

此步骤可省略，但我们已开始使用。

步骤五: 连接红外 LED(简易)





将红外发光 LED 连接到 Raspberry Pi 可以通过多种方式完成。这一步我使用了快速连接方式，我后来发现可能超出当前 Pi 引脚的电流门限。到目前为止，我没有遇到任何问题，但更为理想的连接方式将在下一步进行描述。

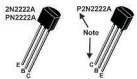
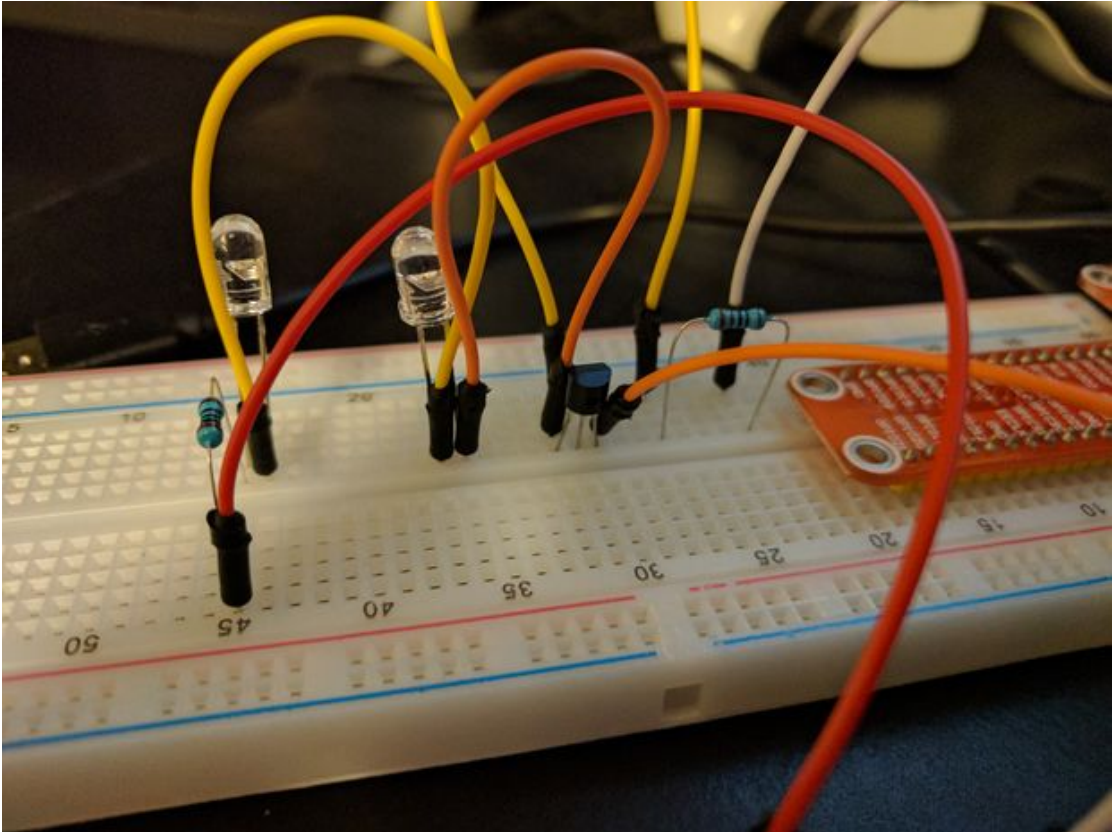
计算红外 LED 所需的电阻。如果您有 LED 的所有规格，<http://ledcalculator.net/>可以帮助确定正确的电阻值。在这种情况下，22 号引脚的电压为 3.3V，LED 压降为 1.2V，额定电流为 20mA，使用 1 个 LED，因此需要 110Ω 电阻值。我使用了一个 100Ω 的电阻。

注意：后来我注意到，任何时刻所有引脚的最大电流为 16mA，所以这种配置可能会超过此门限值。在下一步中将介绍一种包含晶体管和 5V 电源的方案，但使用此方案运几周后，我还没有遇到任何问题。

Raspberry Pi 上的 22 号脚连接到 IR LED 的阳极，阳极是 IR LED 默认较长的一端。

LED 的较短的引脚连接电阻，然后接地。我剪断了电阻上的大部分电线，并将其直接焊接到接地引脚和 LED 的地线。

### 步骤六: 连接红外 LED(正确方法)



为了正确连接 LED 且不超过 Raspberry Pi 的参数门限，请使用合适的电阻将 LED 连接到 5V 电源，将阴极引脚连接到 2N2222 的集电极引脚，将晶体管的发射极引脚连接到地，并将 Pi 的 22 号引脚连接 1K 欧姆电阻到晶体管的基极引脚。这使 22 号引脚非常小的电流流过 LED 到地，并且不超过 Pi 的限制。

在我的例子中，我连接了 2 个红外 LED，所以我可以控制环境照明以及电壁炉。

步骤七: 测试发送 **IR** 命令



要发送 **IR** 命令，使用程序 `irsend`。

语法:`irsend`

例如:`irsend SEND_ONCE Spectrafire KEY_POWER`

这将从 **Spectrafire** 远程发送一次电源按钮命令。重复记录文件时使用其他键名。

在这个阶段, 您可以发送以前使用连接到 **Raspberry Pi** 的 **IR LED** 记录的任何命令。

## 步骤八: 安装 **Ha-bridge**

为了让 **Echo** 能够控制我们的设备, 我们将使用 **Ha-Hridge** 模拟飞利浦灯泡。一旦配置完成, **Echo** 将能够检测到该设备并发送电源开/关命令。

**Ha-Bridge** 的网站清楚地概述了当前版本的功能, 强烈建议您查看。 <https://github.com/bwssystems/ha-bridge> 安装 **HA-Bridge**:

```
1 sudo apt-get install oracle-java8-jdk
2
3 mkdir /home/pi/habridge
4
5 cd /home/pi/habridge
6
7 wget
   https://github.com/bwssystems/ha-bridge/releases/download/v4.3.1/ha-bridge-4.3.1.jar
```

运行 **Ha-Brige**:

```
1 sudo java -jar /home/pi/habridge/ha-bridge-4.3.1.jar
```

设置 **Ha-Brige** 开机启动运行:

```
1 cd /etc/systemd/system
2
3 sudo nano habridge.service
```

复制粘贴以下内容:

```
1 [Unit]
2
3 Description=HA Bridge
4
5 Wants=network.target
6
7 After=network.target
8
```



```

9 [Service]
10 Type=simple
11 WorkingDirectory=/home/pi/habridge
12
13 ExecStart=/usr/bin/java -jar
    -Dconfig.file=/home/pi/habridge/data/habridge.config
    /home/pi/habridge/ha-bridge-4.3.1.jar
14
15 [Install]
16
17 WantedBy=multi-user.target
    
```

CTRL+X,Y 保存并退出。

重载系统控制:

```
1 sudo systemctl daemon-reload
```

开启服务:

```
1 sudo systemctl start habridge.service
```

配置服务以开机启动:

```
1 sudo systemctl enable habridge.service
```

## 步骤九: 模拟飞利浦色调灯泡

The screenshot shows the HA Bridge web interface. At the top, there is a navigation bar with 'HA Bridge', 'Home', 'My Echo', 'Help', and 'About'. Below this is a secondary navigation bar with 'Bridge Devices', 'Bridge Control', 'Logs', and 'Add/Edit'. The main content area shows 'Current devices (1)' and a 'Renumber Devices' button. Below this is a table with the following data:

Row	ID	Name	Desc	Type	Targ	Inacti	No St	Actions
1	1	Bedroom Fireplace		exec		false	false	Test ON Test Dim Test OFF Edit/Copy Delete

At the top of the screenshot, there is a table with columns: On Items, Type, Target Item, Desc, Cour, Filter, Http Verb, Http, Http, Content Type, and Manage. It contains three rows of configuration items for Spectralre KEY\_POWER, Spectralre KEY\_TIME, and The Cat.

当 *ha-bridge* 开始运行, 打开 Web 浏览器并输入 Raspberry Pi 的 IP 地址, 应该会显示 *ha-bridge* 的界面。

点击页面顶端的 ‘Add/Edit’ 链接。

**Name:** 输入你想使用的语音命令名。

‘On Items’ 选项设置类型 ‘Execute Command/Script/Program’ 并在 **Target Item** 工具箱输入命令, 如果需要多个命令, 请单击添加按钮保存当前行并输入另一个命令。也可以设置延迟并重复一个命令一定次数。在这种情况下, 首先按下电源按钮, 然后按下定时器按钮 3 次, 将自动关机定时器设置为 3 小时。

对 ‘Off Items’ 区域重复相同的做法, 完成后单击 **Add**。

在页面顶部, 单击 ‘Add Bridge Device’ 将其保存为新主题, 或 ‘Update Bridge Device’ (如果修改现有主题)。

将 IR LED 对准设备。在 ‘Bridge Devices’ 页面上, 单击 ‘Test ON’ 或 ‘Test OFF’ 按钮, 以验证其是否符合要求。

## 步骤十: 连接到 **Amazon Echo**



最后一步是让 Amazon Echo 与该设备进行通信。注意: 两台设备必须处于相同的网络。

选项 1: 说:“Alexa, discover smart home devices”。

选项 2: 打开 Alexa App, 选择 Menu > Smart Home 并点击 ‘Discover Devices’ 链接。

几分钟后, 设备应该能被识别到。

说: “Alexa, turn the bedroom fireplace on” 并确保设备已按设定运行。

在 ha-bridge 中用任意您想要的名字来替换 Bedroom Fireplace (卧室壁炉)。

如果还没有安装红外 LED, 请找到一个不显眼的地方安装, 同时允许它指向设备中红外线接收器的大体方向。您可能需要移动它来指向不同的区域以找到最佳位置。

原文链接: <https://www.wandianshenme.com/play/amazon-echo-□□□□□□>