



RPi Relay Board

用户手册

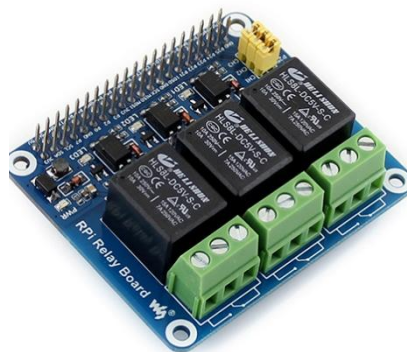
产品简介

RPi Relay Board 是适用于 Raspberry Pi A+/B+/2 代 B/3 代 B 的树莓派继电器扩展板。

主要用途:

树莓派 IO 输出的弱电流通过本产品可以控制强电流的通断,常用于智能家居等强电领域。结合树莓派的网络功能,可以实现对家电的远程控制。

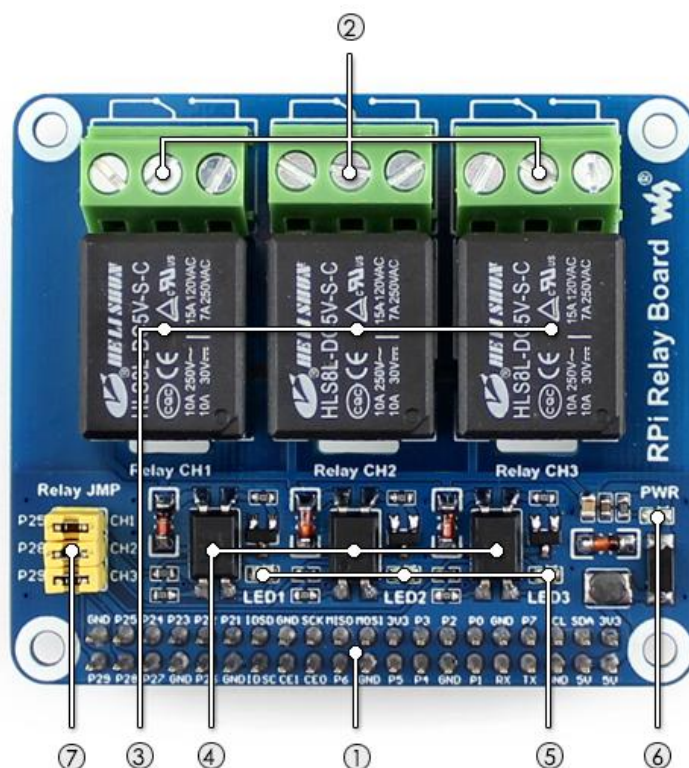
【注意】树莓派具有 40Pin 和 26Pin 两种接口,该产品只兼容 40Pin 的树莓派。



主要特性:

- 适用于树莓派 A+/B+/2B/3B
- 采用高质量的继电器
- 继电器允许接入高达 5A 250V AC 或 5A 30V DC
- 带光耦隔离,避免高电压电路干扰
- 带继电器指示灯,方便查看继电器的工作状态
- 带继电器选择跳线,方便选择树莓派其他引脚控制继电器
- 提供完善的配套资料手册(包括 wiringPi、WebioPi、shell、python 和 bcm2835 等例程)

资源简介



[接口简介]

1. **RPI 连接口**
方便接入树莓派
2. **继电器接线端子**
方便接外部被控制线

[跳线说明]

7. **继电器控制管脚选择跳线**
短接跳线：接入到示例程序指定的 I/O
断开跳线：可改为使用连接线接入自定义的 I/O

[器件简介]

3. **继电器**
4. **光耦**
PC817 光耦芯片
5. **继电器工作指示灯**
LED 亮，继电器常闭端断开，常开端闭合
LED 灭，继电器常闭端闭合，常开端断开
6. **电源指示灯**

接口说明

- 继电器通道和 Pi 引脚的对应关系：

通道标号	RPI 引脚号	wiringPi	BCM	描述
CH1	37	P25	26	通道 1
CH2	38	P28	20	通道 2
CH3	40	P29	21	通道 3

【注意】PCB 上的丝印标示对应 wiringPi 编码。

- Relay_JMP (6P 跳线) 用途：

Relay_JMP 是继电器控制管脚选择跳线。当连接跳线帽时，即可使用树莓派控制继电器。

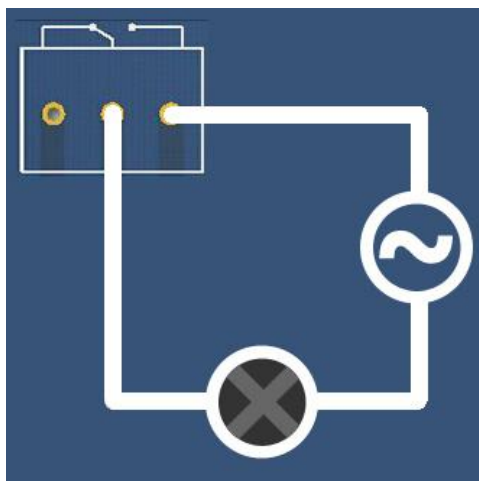
- 接线端子说明：

本扩展板带有三个接线端子，每个接线端子上分别有三个脚，用来连接外部电路。

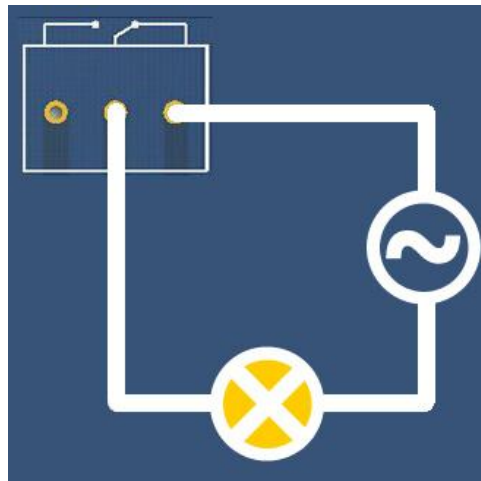
三个端子都是低电平使能。当树莓派 IO 口输出低电平时，则对应通道的 LED 灯点亮。同时继电器常开触点闭合，常闭触点断开，使得外部电路通断状态发生改变。

(PS: 不要忘记连接 Relay_JMP 跳线帽哦！)

继电器动作前：



继电器动作后：



软件例程

该模块的例程包括 Shell、bcm2835、WiringPi、python 和 Webiopi 例程。

开发环境

在使用本扩展板之前，我们需要给树莓派安装 bcm2835, wiringPi 和 python 函数库，树莓派的库函数安装和配置过程请参见[为树莓派安装必要的函数库](#)。

安装好函数库之后，还需要下载产品的示例程序。把示例程序复制并释放到系统中，例如执行 `tar -zxvf RPi_Relay_Board.tar.gz ~`，释放到/home/pi 目录下。部分示例程序拥有不同的实现方式（例如 bcm2835, wiringPi, python, Shell, Webiopi 等方式），即使实现同一个功能，也可能依赖不同的函数库。因此使用之前请务必安装好 bcm2835, wiringPi, python 和 Webiopi 函数库。

如果安装完函数库之后依然无法执行以下的示例程序，那么您可以尝试使用 `chmod +x filename` 命令给程序赋予可执行权限。

实验操作和现象

1) Shell 程序

进入 Linux 终端，在终端执行以下命令。

执行程序：

```
pi@raspberrypi ~/RPi_Relay_Board/shell $ sudo ./Relay.sh CH1 ON
```

预期结果：继电器通道 1 的 LED 被点亮，同时听到继电器接合的声音。命令中后面两个参数可改变，例如运行如下命令分别为继电器 2 接合，继电器 3 断开。

```
pi@raspberrypi ~/RPi_Relay_Board/shell $ sudo ./Relay.sh CH2 ON
```

```
pi@raspberrypi ~/RPi_Relay_Board/shell $ sudo ./Relay.sh CH3 OFF
```

2) Bcm2835 程序

编译程序：

```
pi@raspberrypi ~/RPi_Relay_Board/bcm2835 $ make
```

终端执行：

```
pi@raspberrypi ~/RPi_Relay_Board/bcm2835 $ sudo ./Relay_Module
```

预期结果: 可以看到 3 个 LED 依次点亮, 继电器依次在常开触点和常闭触点之间来回切换。同时终端会显示目前继电器在哪个触点。

按键盘 **Ctrl + C** 结束实验。

3) wiringPi 程序

编译程序:

```
pi@raspberrypi ~/RPi_Relay_Board/wiringPi $ make
```

终端执行:

```
pi@raspberrypi ~/RPi_Relay_Board/wiringPi $ sudo ./Relay_Module
```

预期结果: 可以看到 3 个 LED 依次点亮, 继电器依次在常开触点和常闭触点之间来回切换。同时终端会显示目前继电器在哪个触点。

按键盘 **Ctrl + C** 结束实验。

4) python 程序

终端执行:

```
pi@raspberrypi ~/RPi_Relay_Board/python $ sudo python Relay_Module.py
```

预期结果: 可以看到 3 个 LED 依次被点亮, 继电器依次在常开触点和常闭触点之间来回切换。同时终端会显示目前继电器在哪个触点。

按键盘 **Ctrl + C** 结束实验。

5) Webiopi 程序

本例程的网页控制是基于 Webiopi 的软件框架来控制继电器的。

● 环境安装:

进入 Webiopi 目录下解压库文件并安装

```
pi@raspberrypi ~/RPi_Relay_Board/Webiopi $ tar -zxvf WebIOPi-0.7.1-raspi2.tar.gz
```

```
pi@raspberrypi ~/RPi_Relay_Board/Webiopi $ cd WebIOPi-0.7.1-raspi2
```

```
pi@raspberrypi ~/RPi_Relay_Board/Webiopi/WebIOPi-0.7.1-raspi2 $ sudo ./setup.sh
```

运行 `webiopi -h` 如果出现以下界面, 则库安装成功。

```
pi@raspberrypi ~ $ webiopi -h
WebIOPi command-line usage
webiopi [-h] [-c config] [-l log] [-s script] [-d] [port]

Options:
-h, --help            Display this help
-c, --config file     Load config from file
-l, --log file        Log to file
-s, --script file     Load script from file
-d, --debug           Enable DEBUG

Arguments:
port                  Port to bind the HTTP Server
pi@raspberrypi ~ $
```

终端执行：

```
pi@raspberrypi ~ $ sudo webiopi -d -c /etc/webiopi/config
```

然后在电脑端或者手机端打开网页浏览器，在地址栏内输入树莓派 ip 地址，端口号 8000。例如：<http://192.168.1.16:8000>（根据实际情况填入）。在登录网站时，需要输入账号和密码，默认的账号和密码分别是：webiopi 和 raspberry。如果进入 WebIOPi Main Menu 页面，说明环境配置成功了：

WebIOPi Main Menu

[GPIO Header](#)

Control and Debug the Raspberry Pi GPIO with a display which looks like the physical header.

[GPIO List](#)

Control and Debug the Raspberry Pi GPIO ordered in a single column.

[Serial Monitor](#)

Use the browser to play with Serial interfaces configured in WebIOPi.

[Devices Monitor](#)

Control and Debug devices and circuits wired to your Pi and configured in WebIOPi.

- **实验操作和现象：**

回到终端。按下 **Ctrl + C** 结束上面的进程。然后执行：

```
cd ~/RPI_Relay_Board/Webiopi
```

```
pi@raspberrypi ~/RPI_Relay_Board/Webiopi $ sudo webiopi -d -c config
```

打开网页浏览器，在地址栏内重新输入树莓派 ip 地址，端口号 8000，或者刷新网页：



如果没有出现这个页面，建议使用 Chrome 或者 Firefox 浏览器。此外还要确保树莓派的当前目录为~/RPI_Relay_Board/Webiopi，这个目录下有一个 index.html 文件，然后再执行 `sudo webiopi -d -c config`。这样客户端的浏览器才能进入 index.html，否则会进入 WebIOPi Main Menu 页面。

点击网页上的不同继电器通道的控件可以控制继电器。

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Single Board Computers](#) category:

Click to view products by [Waveshare](#) manufacturer:

Other Similar products are found below :

[MANO882VPGGA-H81](#) [SSD3200W-S-SLC-INN](#) [AmITX-SL-G-Q170](#) [IB100](#) [MVME61006E-2173R](#) [20-101-0738](#) [PCE-4128G2-00A1E](#)
[RSB-4220CS-MCA1E](#) [SHB230DGGA-RC](#) [IB909AF-5650](#) [AmITX-BT-I-E3815](#) [PICO841VGA-E3827](#) [IMB210VGGA](#) [MI981AF](#) [RSB-4221CS-MCA1E](#) [PCE-9228G2I-00A1E](#) [IB915F-3955](#) [IB909F-5010](#) [MI958F-16C](#) [UPS-P-8G-64GB-PACK](#) [S2600WFT](#) [IB915AF-6300](#)
[S2600STB](#) [BBS2600BPS](#) [IB915F-6100](#) [Nit6QP_MAX](#) [MI990VF-X28-E](#) [MI990VF-6820](#) [MI991AF-C236](#) [94AC6636](#) [BANANA PI BPI-M4](#)
[BLKNUC7I3DNHNC1978015](#) [BLKNUC7I5DNK1E 960791](#) [IOT-LS1012A-OXALIS](#) [NITX-300-ET-DVI](#) [94AC6633](#) [A33-OLINUXINO-N8G](#) [A64-OLINUXINO-1GE16GW](#) [A20-SOM-E16GS16M](#) [A20-SOM204-1G-M](#) [EMB-APL1-A10-3350-F1-LV](#) [PICO-APL1-A10-F001](#)
[PICO-APL4-A10-F003](#) [ODYSSEY - STM32MP157C BOARD WITH SOM](#) [BEAGLEBONE GREEN GATEWAY DEV BOARD](#) [ODYSSEY - X86J4105864 8GB RAM 64GB EMMC](#) [ODYSSEY -X86J4105864 8GB/64GB ENTERPRISE](#) [VISIONDK-STM32MP1 V.1.0](#) [VISIONDK-6ULL V.2.0](#) [VISIONDK-8MMINI V.1.0](#)