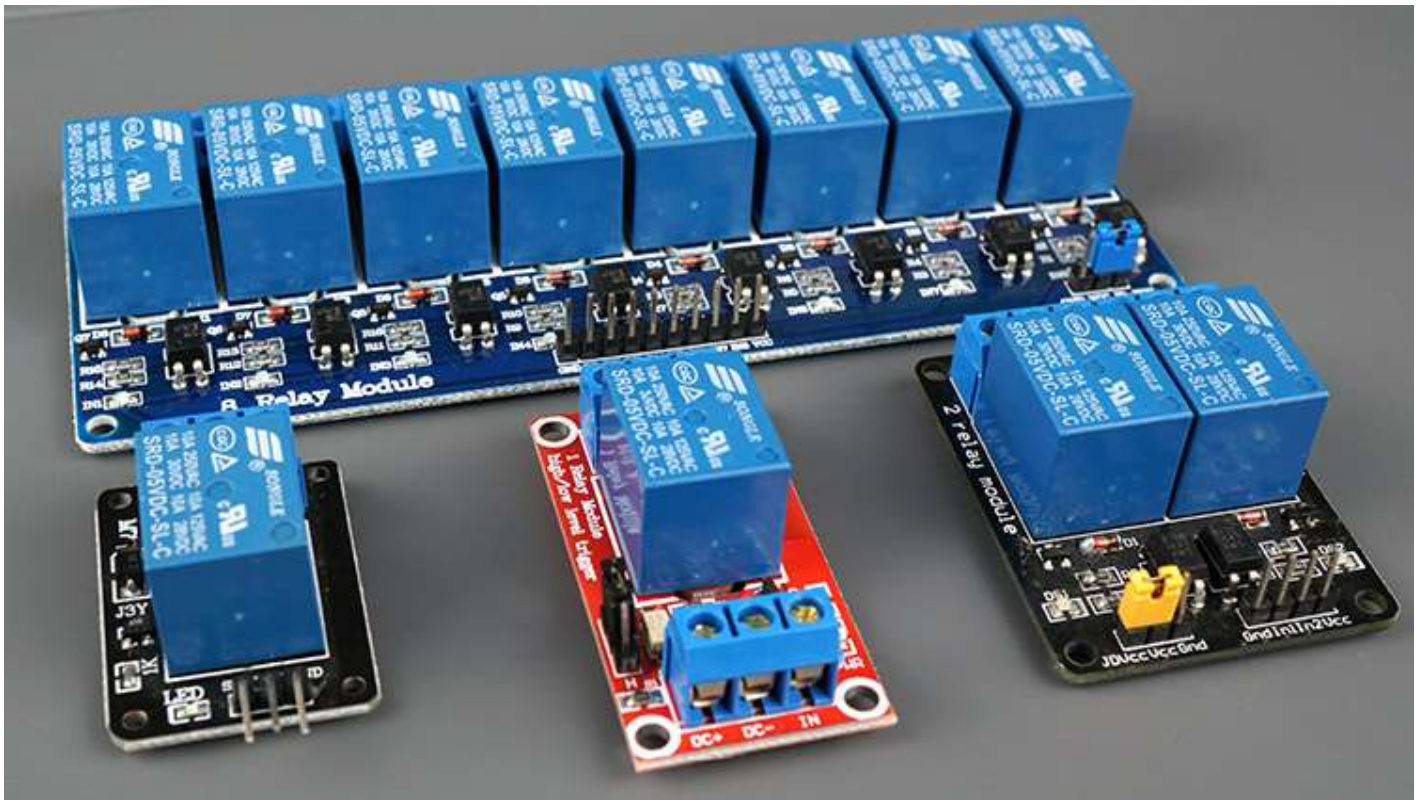
介绍继电器

继电器是一种电动开关，与任何其它开关一样，它可以打开或关闭，让电流通过或不通过。它可以用低压进行控制，例如 ESP32 GPIO 提供的 3.3V，并允许我们控制高压，例如 12V、24V 或电源电压（中国家用电为220V）。

1、2、4、8、16 通道继电器模块

不同的继电器模块具有不同的通道数。您可以找到具有 1 个、2 个、4 个、8 个甚至 16 个通道的继电器模块。通道的数量决定了我们能够控制的输出数量。



有继电器模块，其电磁铁可以由 5V 和 3.3V 供电。两者都可以与 ESP32 一起使用——您可以使用 VIN 引脚（提供 5V）或 3.3V 引脚。

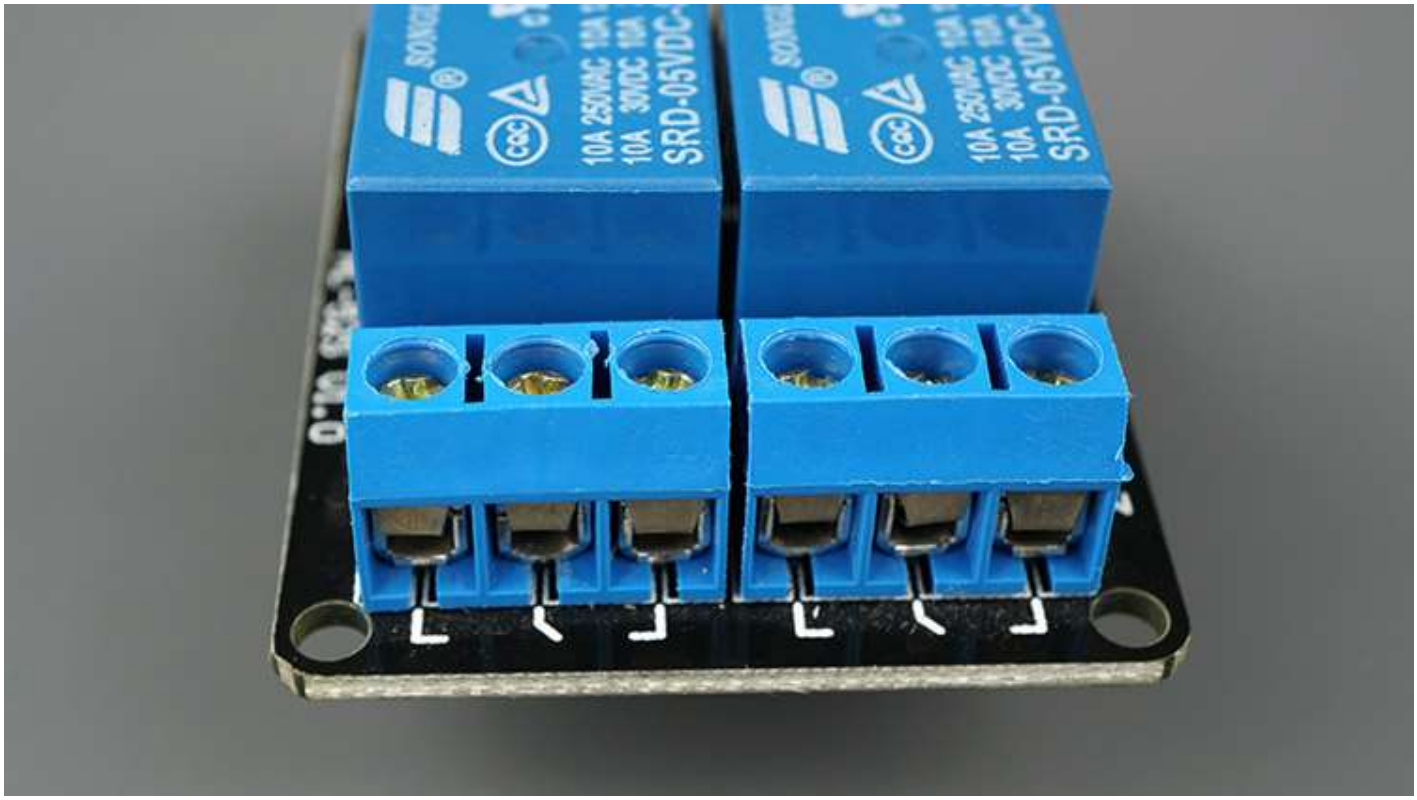
此外，有些还带有内置光耦合器，增加了额外的“保护层”，将 ESP32 与继电器电路光学隔离。

继电器引脚

出于演示目的，让我们看一下 2 通道继电器模块的引脚排列。使用具有不同数量通道的继电器模块是类似的。

左侧有两组三个插座，用于连接高压，右侧（低压）的引脚连接到 ESP32 GPIO。

电源电压连接



上图所示的继电器模块有两个连接器，每个都有三个插座：普通 (通讯), 常闭 (数控) 和常开 (不）。

- COM： 连接您要控制的电流（市电电压）。
- NC（常闭）： 当您希望继电器默认关闭时使用常闭配置。NC 是 COM 引脚已连接，这意味着电流正在流动，除非您从 ESP32 向继电器模块发送信号以打开电路并停止电流流动。

- NO（常开）： 常开配置相反：NO 和 COM 引脚之间没有连接，因此除非您从 ESP32 发送信号关闭电路，否则电路会断开。

控制引脚

低压侧有一组四个引脚和一组三个引脚。第一组包括VCC和接地为模块加电，并输入 1 (IN1) 和输入 2 (IN2) 分别控制底部和顶部继电器。

如果您的继电器模块只有一个通道，那么您将只有一个 IN 引脚。如果您有四个通道，您将有四个 IN 引脚，依此类推。

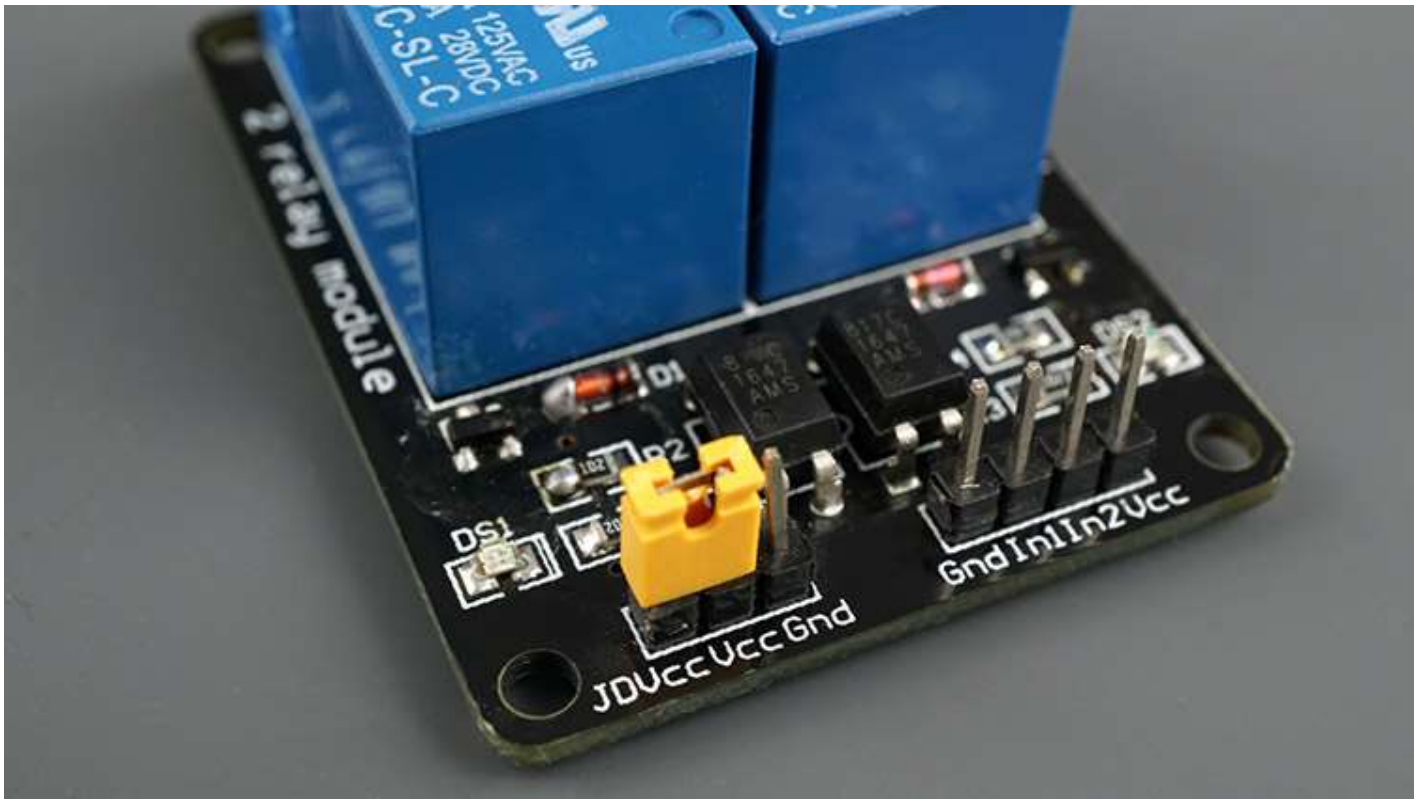
您发送到 IN 引脚的信号决定了继电器是否处于活动状态。当输入低于约 2V 时触发继电器。这意味着您将遇到以下情况：

- 常闭配置（NC）：
  高信号 – 电流正在流动
  低信号 – 电流不流动
- 常开配置（NO）：
  高信号 – 电流不流动
  低信号 – 电流流动

当电流应该在大部分时间流动时，您应该使用常闭配置，并且您只想偶尔停止它。

如果您希望电流偶尔流动（例如，偶尔打开灯），请使用常开配置。

电源选择



第二组引脚包括GND,VCC， 和JD-VCC引脚。这JD-VCC引脚为继电器的电磁铁供电。请注意，该模块有一个连接 VCC 和 JD-VCC 引脚的连接线帽；这里显示的是黄色的，但你的可能是不同的颜色。

随着连接线帽，VCC和JD-VCC引脚连接。这意味着继电器电磁铁直接由 ESP32 电源引脚供电，因此继电器模块和 ESP32 电路之间没有物理隔离。

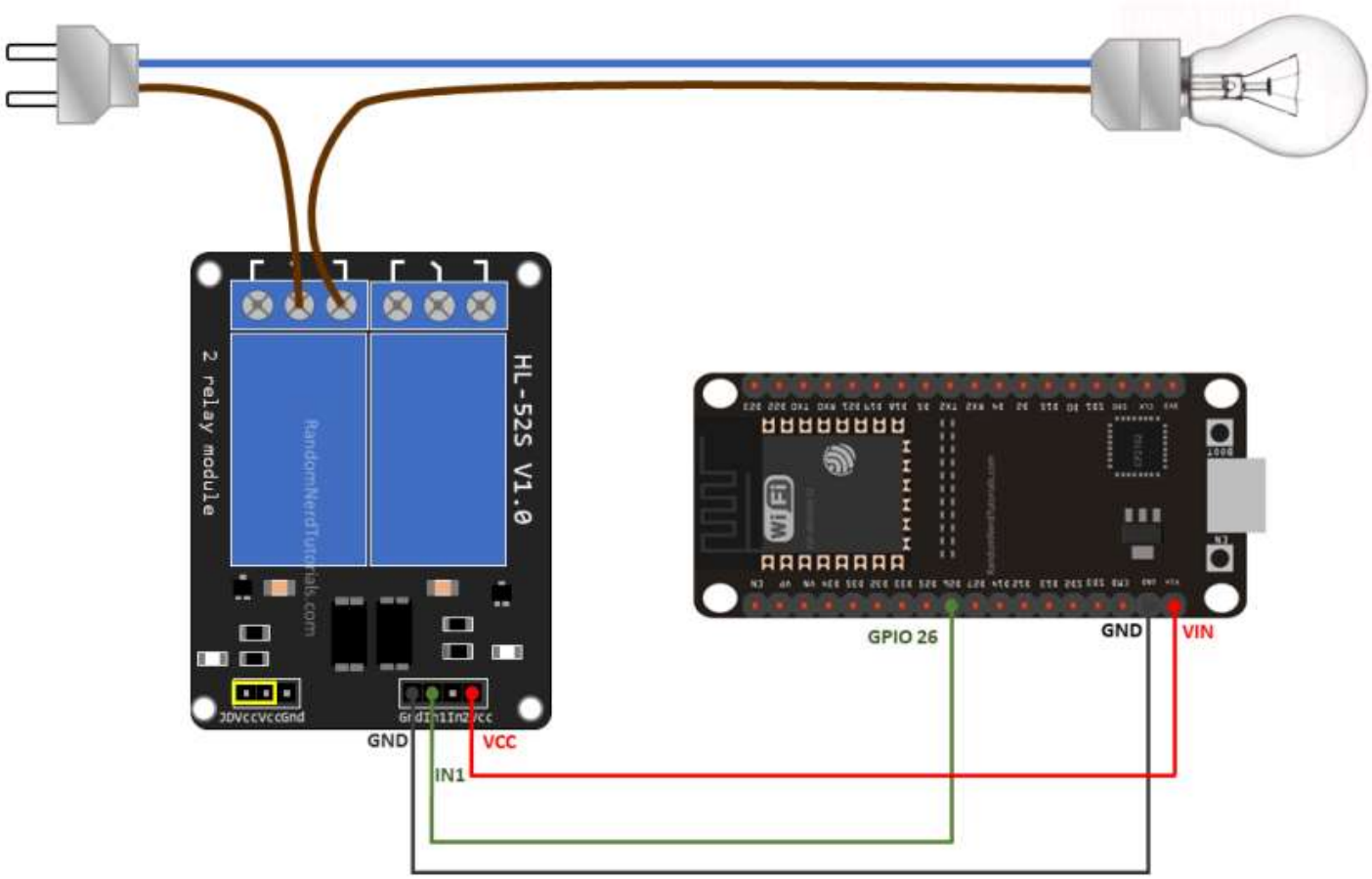
如果没有连接线帽，则需要提供一个独立的电源，通过连接线给继电器的电磁铁供电京东-VCC引脚。该配置使用模块的内置光耦合器将继电器与 ESP32 物理隔离，从而防止在出现电尖峰时损坏 ESP32。

将继电器模块连接到 ESP32

如下图所示将继电器模块连接到 ESP32。该图显示了 2 通道继电器模块的接线，不同通道数的接线类似。

警告：在这个例子中，我们正在处理电源电压。误用会导致严重伤害。如果您不熟悉电源电压，可以查一下相关资料。在对 ESP 进行编程或为电路接线时，请确保一切都与电源电压断开。

或者，您可以使用 12V 电源来控制 12V 电器。

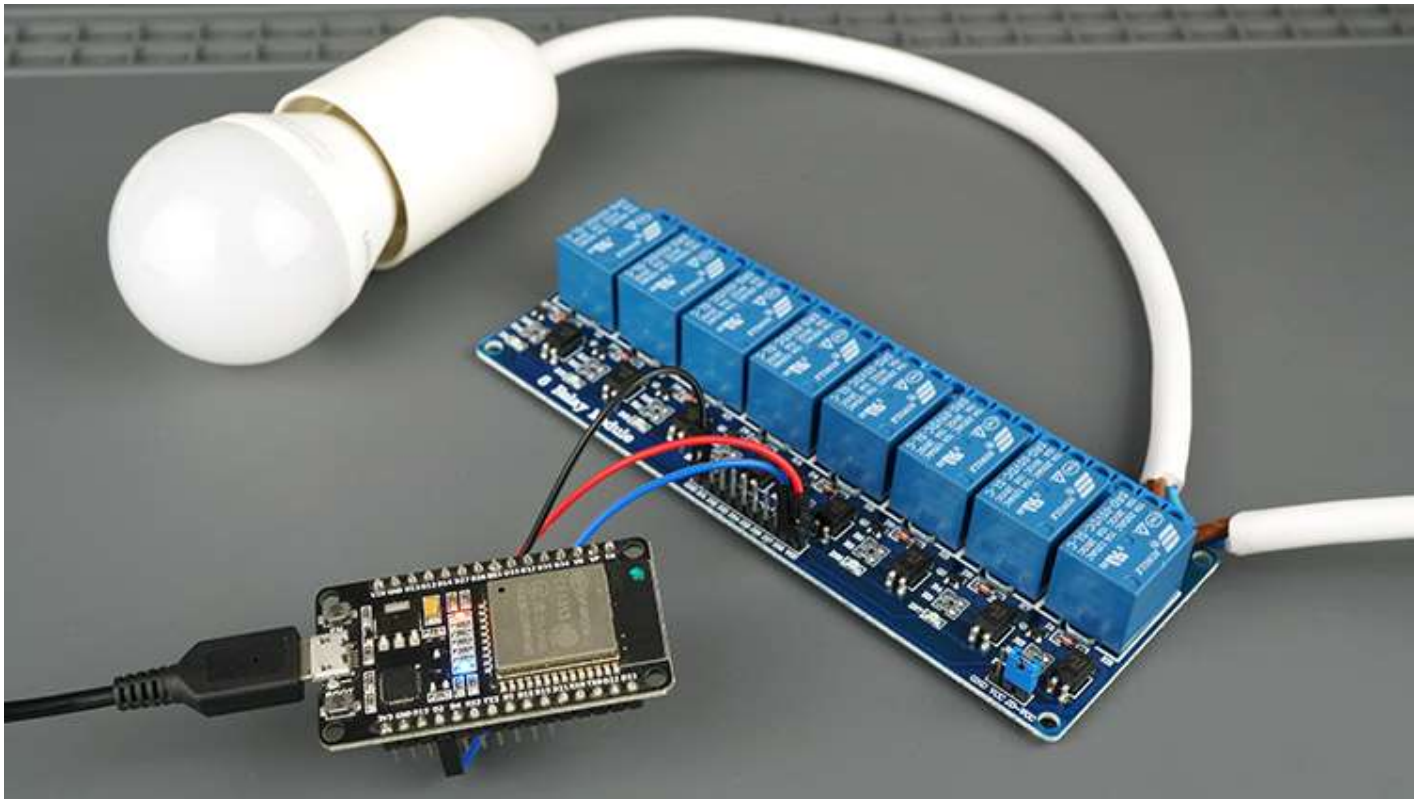


在这个例子中，我们正在控制一盏灯。我们只是想偶尔点亮灯，所以最好使用常开配置。

我们将 IN1 引脚连接到通用输入输出接口 26，您可以使用任何其它合适的 GPIO。

使用 ESP32 控制继电器模块 – Arduino Sketch

使用 ESP32 控制继电器的代码就像控制 LED 或任何其它输出一样简单。在此示例中，由于我们使用的是常开配置，我们需要发送一个 LOW 信号让电流流动，并发送一个 HIGH 信号来停止电流流动。



以下代码将点亮您的灯 10 秒，然后再将其关闭 10 秒。

```
const int relay = 26;

void setup() {
  Serial.begin(115200);
  pinMode(relay, OUTPUT);
}

void loop() {
  // Normally Open configuration, send LOW signal to let current flow
  // (if you're usong Normally Closed configuration send HIGH signal)
  digitalWrite(relay, LOW);
  Serial.println("Current Flowing");
  delay(5000);

  // Normally Open configuration, send HIGH signal stop current flow
  // (if you're usong Normally Closed configuration send LOW signal)
  digitalWrite(relay, HIGH);
  Serial.println("Current not Flowing");
  delay(5000);
}
```

代码如何运作

定义继电器 IN 引脚连接的引脚。

```
const int relay = 26;
```

在 setup() 里面，将继电器定义为输出。

```
pinMode(relay, OUTPUT);
```

在 loop() 里面， 设置一个低的信号让电流流动并点亮灯。

```
digitalWrite(relay, LOW);
```

如果您使用的是常闭配置，请发送高的信号灯点亮。然后，等待 5 秒钟。

```
delay(5000);
```

通过发送一个停止当前流高的信号到继电器引脚。如果您使用的是常闭配置，请发送低的信号停止电流。

```
digitalWrite(relay, HIGH);
```

使用 ESP32 Web 服务器控制多个继电器