

## 使用引脚复用减少 8 位单片机的引脚使用

Microchip technology Inc.

8 位单片机产品部

Nate Thompson 和 Alexandru Sabiuta

本文将介绍两个利用通用输入/输出（GPIO）引脚复用来减少项目引脚使用的示例。第一个示例使用了“查理复用算法”技术，第二个示例使用了定时外设和中断来快速切换引脚输入状态和输出状态，以在驱动 LED 的同时读取按键。

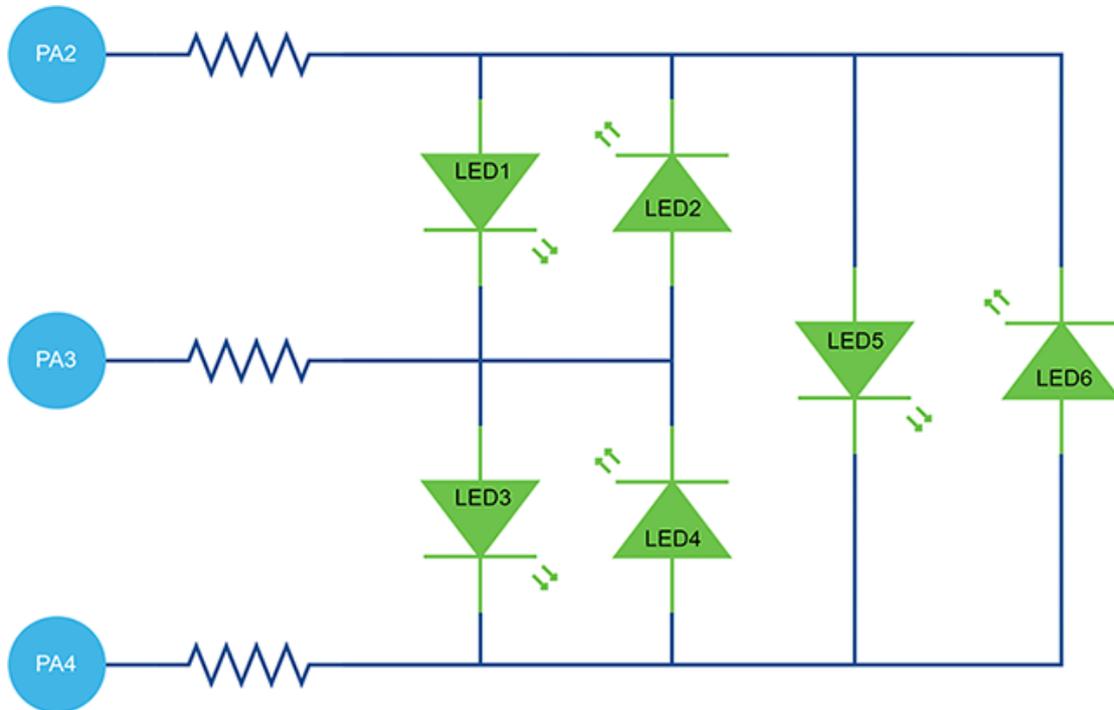
### PIC®和 AVR®单片机上的引脚复用

通用输入/输出（GPIO）引脚是单片机设计时的重要考量之一。您需要足够的引脚将单片机连接到设计中的所有其他组件；但是，超过所需的引脚可能会导致设计成本增加，并且占用电路板上更多空间。此外，减少设计中的所需引脚数量可能就可以选择另一种不同的封装。如果有一种方法可以从较小的封装中挤出更多的 I/O 引脚就好了！幸运的是，我们确实有方法可以在这种情况下释放引脚。

下面是 Microchip 应用团队使用 AVR DD 系列单片机创建的两个示例。首先，我们将探讨通过一种称为“查理复用算法”的技术复用 GPIO 引脚来控制 LED。接下来，我们将看一下如何使用单个引脚同时运行独立的按键和 LED。

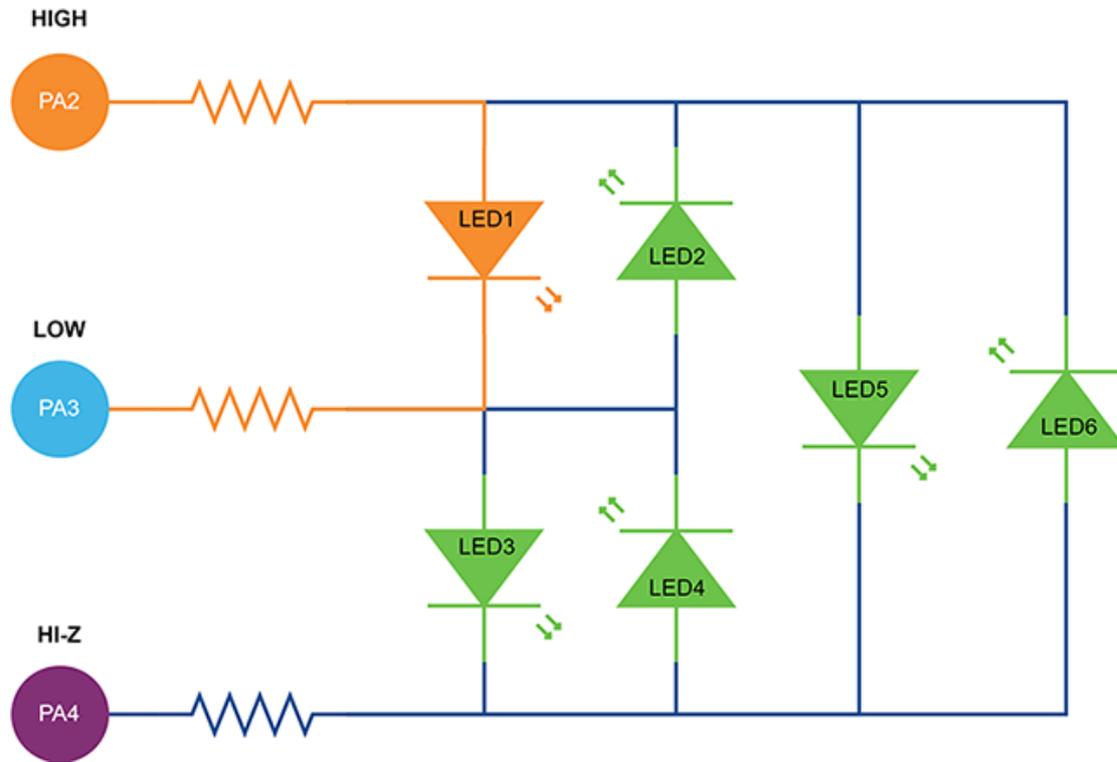
查理复用算法的核心思想是利用单片机引脚可以处于的三种状态：数字高电平、数字低电平和高阻抗数字输入状态（也称为高阻态或三态）。通过使用这三种状态而不仅仅是典型的高电平和低电平，用户可以使用  $n$  个引脚驱动最多  $(n^2-n)$  个独立的 LED。在此示例中，三个指定的引脚可以驱动六个独立的 LED。

那么查理复用算法的工作原理到底是什么呢？为了理解这一点，我们需要看一个示例。

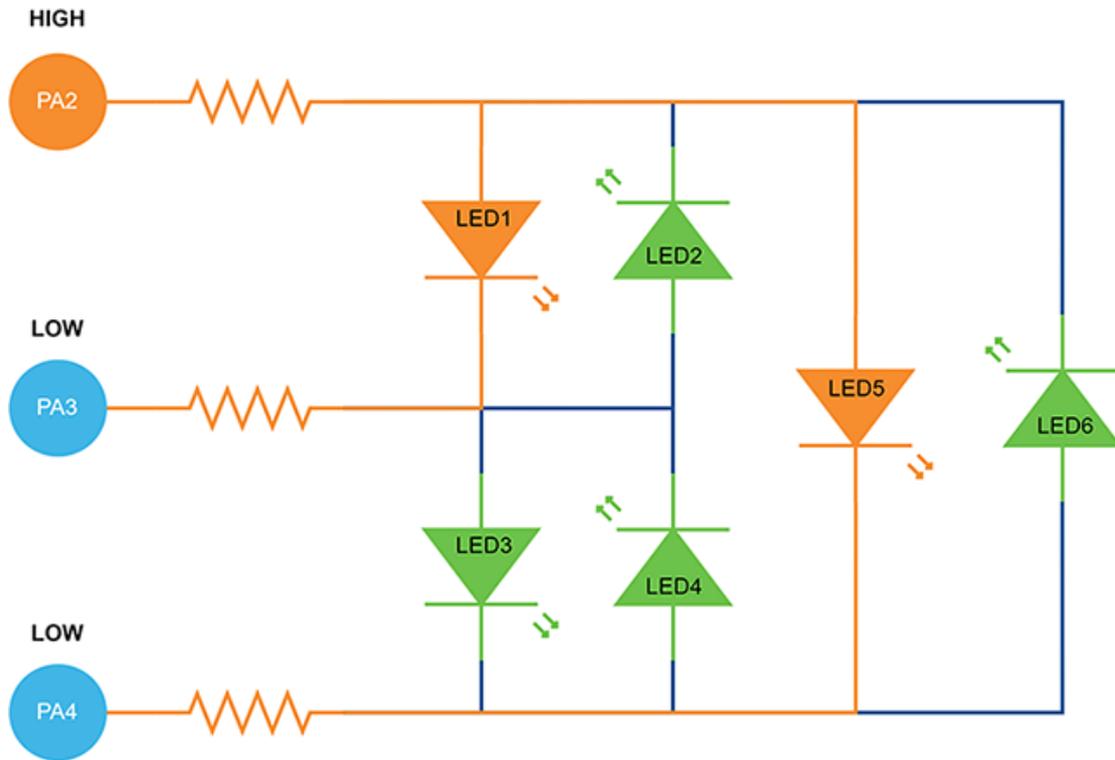


这里我们用三个 I/O 引脚驱动六个 LED。

如果我们只想导通 LED1，则需要将 PA2 设为高电平，PA3 设为低电平，PA4 设为高阻态。电流将流过 LED1 并使其导通，同时使所有其他 LED 保持关断状态。

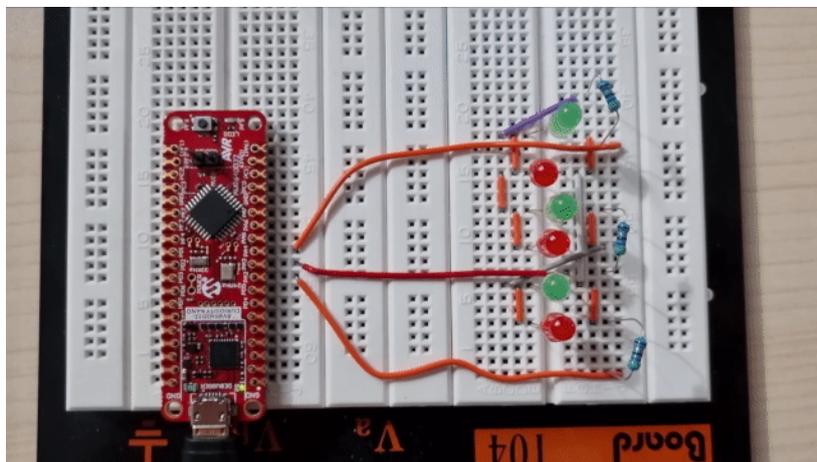


这里的三态十分重要，因为如果 PA4 为“低电平”，它也会意外导通 LED5。  
如下面所示：



高阻抗逻辑状态确保电流只流过 LED1。利用 I/O 引脚状态的相应组合（要获得预期行为，用户必须将一个引脚设为高电平、一个引脚设为低电平，其余引脚设为高阻抗），可以为电路中的任何 LED 重复此过程。

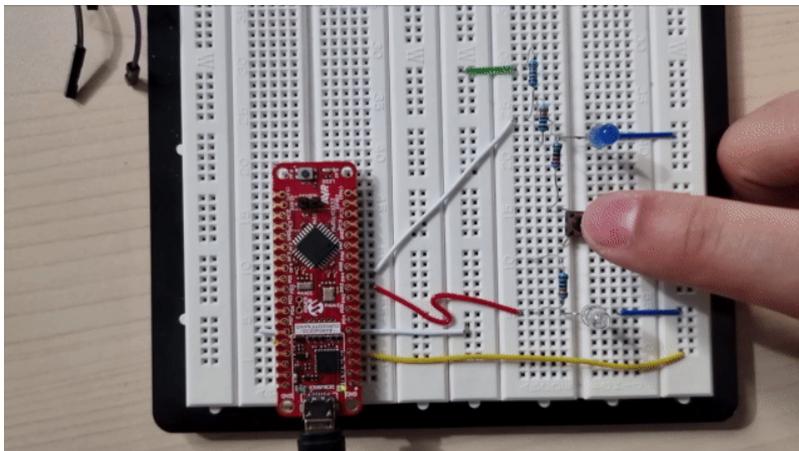
虽然在任意给定时刻这种方法只会点亮一个 LED，但是快速改变 I/O 状态可创造出点亮多个 LED 的错觉。（LED 调光的工作原理是通过在单个 LED 上使用脉冲宽度调制，原理与此非常相似）。可以在下面看到：（点击图片查看动态效果）



第二个示例涉及到在单个引脚上改动一个 LED 和一个按键。

从有利的角度看，可利用中断和定时器来实现此目的。单片机引脚大部分时间都在驱动 LED。然而，一个定时外设会定期在单片机内部触发中断，将引脚从输出快速切换为输入，然后检查按键的状态。虽然这会暂时停止驱动 LED，但如果这个过程可以做得足够快，那么在检查按钮状态的时候，对 LED 造成的影响几乎不可见。

我们在下面的示例中使用了这种方法。单片机不停地驱动一个闪烁的 LED，同时定期检查按键，如果按键被按下，就会点亮一个单独的 LED。（点击图片查看动态效果）



尽管本文章只涵盖了这两种技术的基础知识，但我们的 **Microchip** 应用团队发布的相应 [GitHub 页面](#) 包含了图表、示例代码和更详细的说明。请访问该页面或我们的 [AVR DD 产品系列页面](#) 来了解更多信息。