# Arduino的步进马达控制

# ****学习使用Arduino的马达扩展板去控制步进马达****

ChelseaBack



# ****什么是步进马达?****

步进马达是一种透过用离散的方式来移动的电磁装置，它是由几个线圈来组成不同的” 相”。当每个阶段能依次通电后，步进马达就能实现十分准确的定位或速度的控制，这亦是步进马达常被运用于高精度的应用如打印机的原因。

# ****单极和双极****

步进马达具有两种不同的类型，分别是单极和双极。它们之间最大的区别就是两种不同类型都有各自的卷绕装置，而且它们各自的卷绕装置是会影响它们的步进马达的控制方式。

## ****单极步进马达****

单极步进马达是由具备中心轴头的单卷绕组所组成的。卷绕组的每个部份都是根据磁场的方向而被接通，因为它内里的磁极是能够在不转变电流的方向的情况下反转。试举例说明一下，虽然在双相单极步进器上有6个接点，每相3个，但有两个公共点是能够在内部连接起来，这意味着只有5个接点。

## ****双极步进马达****

此类型的马达与单极步进马达不同，它每相只有一个卷绕组，而且它没有轴头。所以为了能够转变磁极，卷绕组中的电流需要被反转。这意味着双极步进器运行时比较复杂。虽然如此，但它亦有其好处，纵然双极步进马达和单极步进马达是相同重量，但是它具备较好使用的卷绕组，因此双极步进马达比单极步进马达更强大。而且单极步进马达在相同空间中需要达两倍的导线量，并且在任何时间下亦只能使用其一半，换言之单极步进马达只有50%的效率。

## ****对于单极性和双极性，它们各自拥有几种不同的卷绕装置，如下图所示****

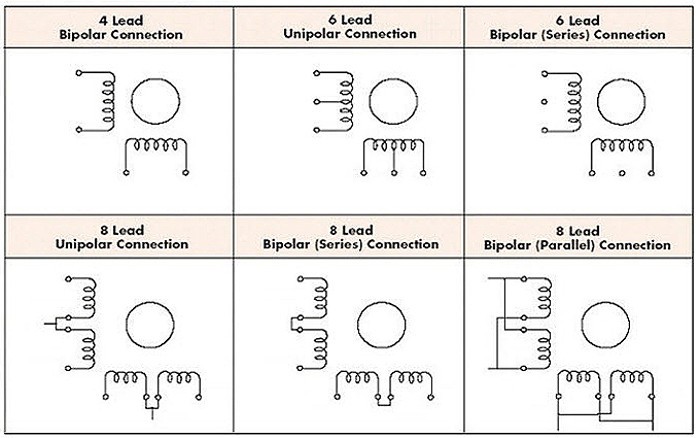
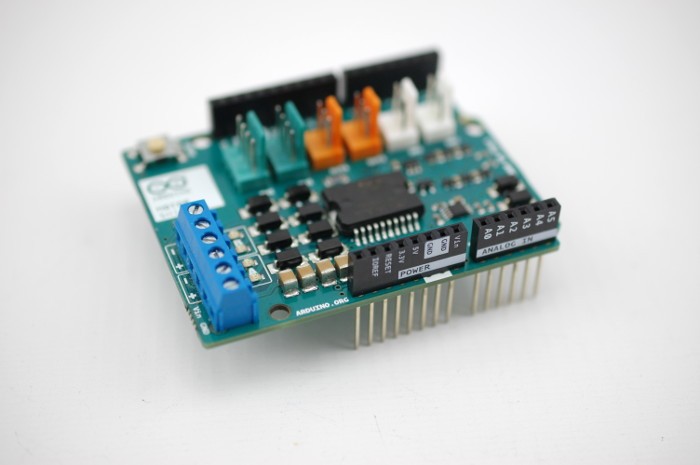


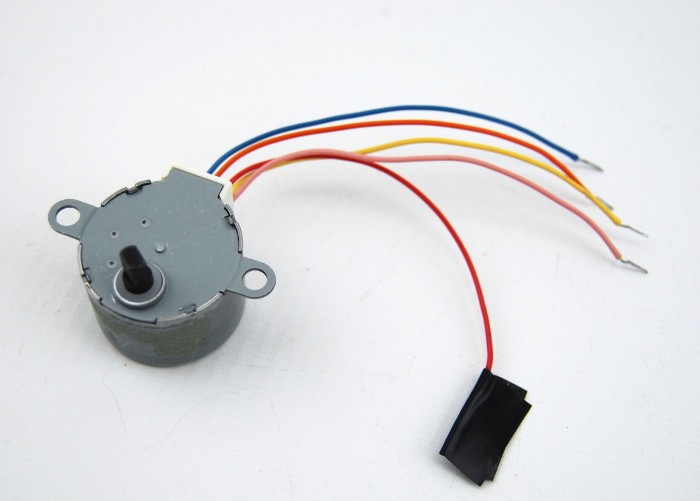
Image source: Osmtec.com

## ****Arduino的马达扩展板****

此马达扩展板是基于L298驱动器，它能驱动一个步进马达或者是两个直流电的马达。如具备了此扩展板，你能够独立地控制一个或更多个马达的速度和方向。当你使用扩展板时能直接将马达连接到Arduino板，而不需要面包板或任何使用Arduino时通常需要的额外电路。



在我的项目中，我决定用Cliff Electronic的单极双极,单极步进马达，它是具备了5个接点的步进马达。但是当这个公共点被忽视的时侯，不论是它的性能或是整体都可当成双极步进马达。

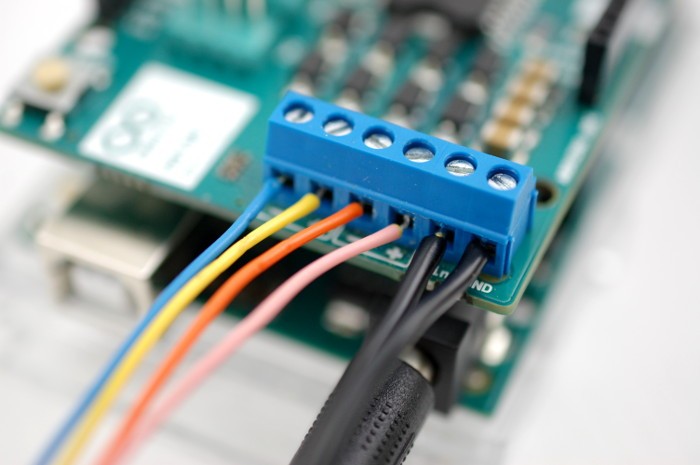


由于此马达需要一个12伏特的电压，因此我必须把扩展板里面的电线和ArduinoUNO分开，以避免任何可能的损害(详情在Arduino 马达扩展板的网页中提供)。亦因为这个原因，我无法直接使用12伏特的电压直接插入ArduinoUNO板，所以我把它们改装，令它们能同样被充电。



## ****找出在马达中的线圈****

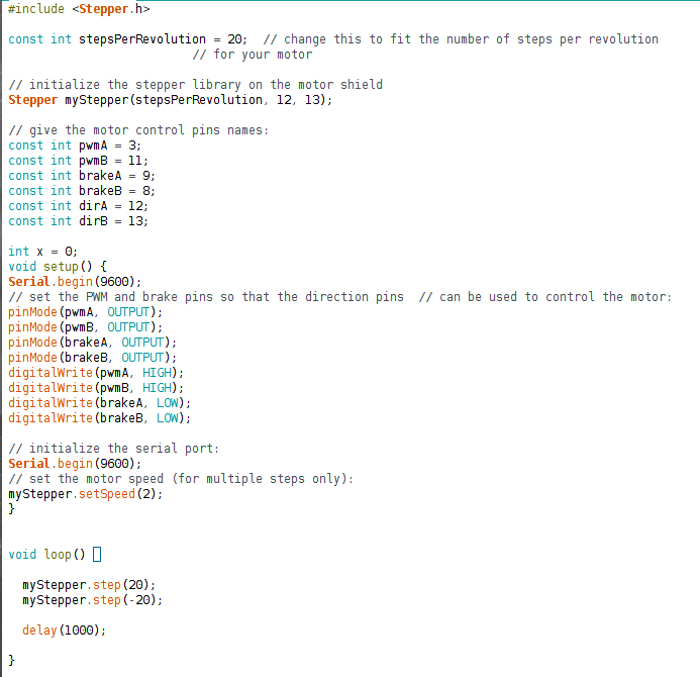
厂商提供的马达资料表中能清楚显示出每一条电线是代表什么，所以我知道蓝色和黄色是一组线圈，而粉红色和橙色则是另一组线圈，而红色是两者的公共点。举例来说: 如果我把该马达看成是单极步进的马达，我会将一组线圈连接到通道A，而把另一组的线圈连接到通道B，并同时保持红色的电线不会连接到任何一组线圈，因为这是一个单极的步进马达，而红色的电线是它们的公共接点。



如果资料表是没有说明每一条电线是代表什么，自己也十分容易确定。首先把万用表设置为欧姆，线圈的某部分将显示出电阻的数值，线圈的轴头的数值会是电阻的一半。所以当轴头的数值是0的时侯，则说明了这两个是独立的线圈。

## ****运行中的马达****

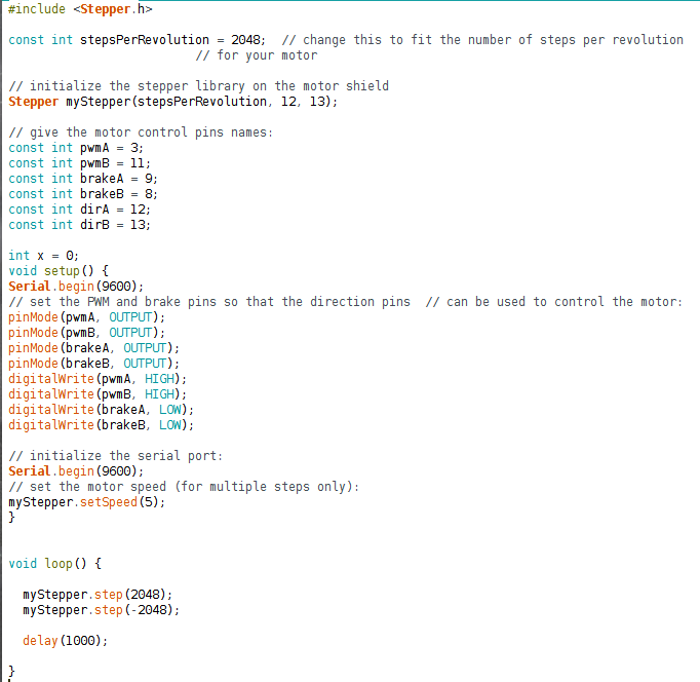
因为我的马达已被连接，我需要做的下一步就是找到代码，所以我有一个粗略的想法。我在Instructables中的Arduino马达扩展板教程中找到了一些代码，这使我能够驱动我的马达。我真正想做的是在所使用的sketch中包括Stepper.h库，但这个例子中没有使用这个库。最好就是使用官方的资料库，所以我在一个Arduino论坛找了另一个sketch，在这个sketch中，马达能在一个方向旋转360度，然后在相反方向也可以旋转360度。



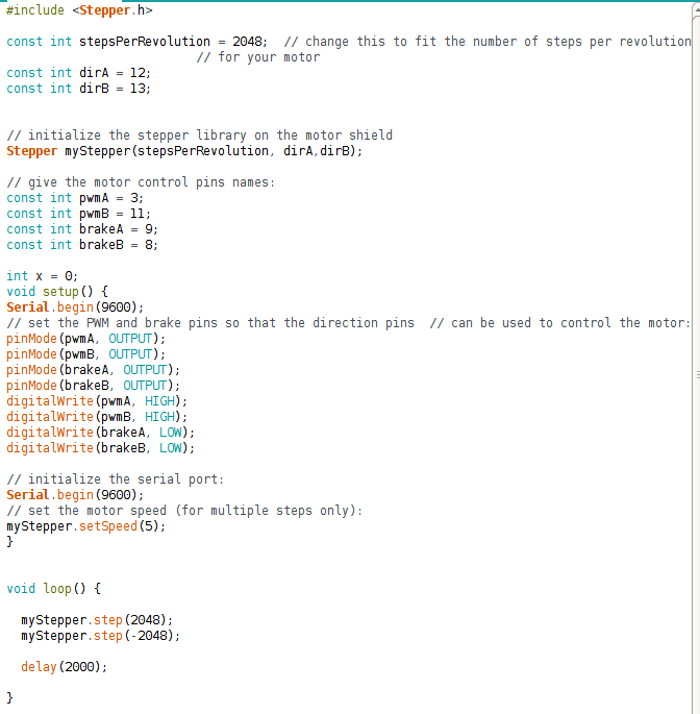
在这个sketch中，我需要指定我使用的马达每一转的步数，但我必须先计算出来，厂商的资料表规定了马达是5.625和11.25度的增加量。因此我用它来确定每一转的步数，所以我们需要将360除以角度。由于这个马达也是依赖齿轮来转动的，而且齿轮比为1:64，因此我们需要将转数乘以齿轮比。像这样：

360 / 11.25×64 = 2048 360 / 5.625×64 = 4096

一开始我在sketch中使用每转4096步，假设马达的角度为5.625度。但当我这样做时，它逆时针旋转了两次，然后顺时针旋转了两次。很明显这里我假设了错误的角度，因为它做了两个旋转，而且5.625是11的一半，当在分区中使用时它会提供两倍的答案。我把每转的步数改为2048，这时马达顺时针旋转一圈，然后逆时针旋转一圈。由于把马达的速度设置为二似乎移动得太慢，因此我把速度从二改为五。



直到现时，我看着我使用中的代码，有一些代码似乎并不是完全有意义。我不明白为何dirA和dirB的位置都没有被使用或设定过，为什么要把它们也列出来。我曾讨论这两行，并再次上传sketch，我也认为这些代码不是必要的，所以我删除它们。接下来，我再次添加我这些行，但上面的步进库初始化并将其用作定义。



然后我看了一下Stepper库的源代码，看到那里并没有任何一组代码来控制不同的插头的位置。原因是控制插头的位置的线路全被定义为输出和被设定为低逻辑，变成是防止其浮动的预防措施。

我觉得我已经学到了很多关于步进马达的知识，它非常有趣，而且我了解到它们是如何运作，以及各种卷绕组的安排是如何影响他们的操作。我真的很兴奋及希望能够在未来的项目中再次使用它们。