

A4988的引脚图及运用

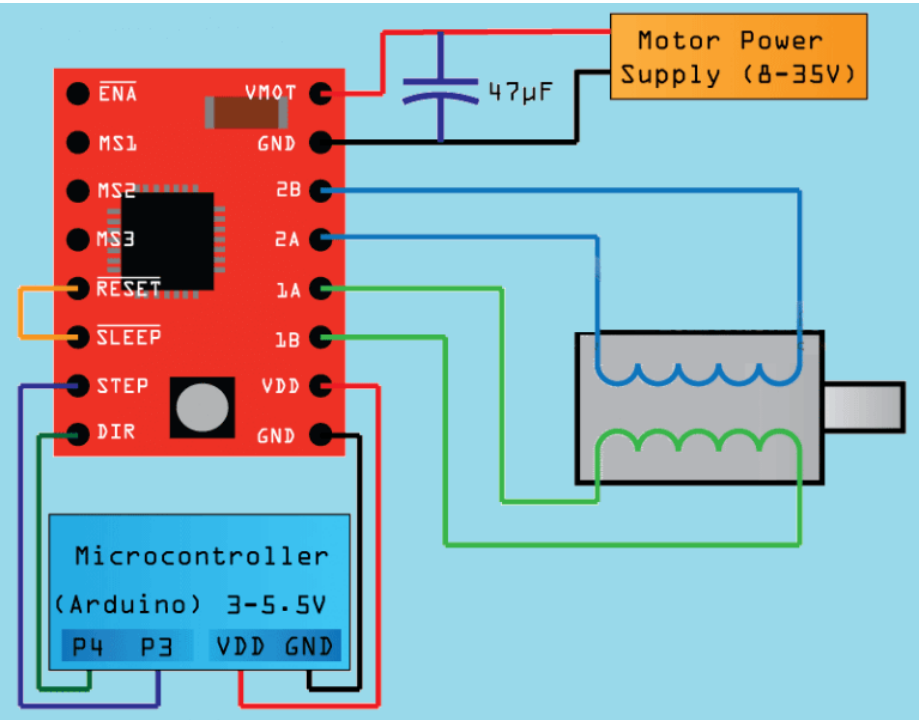
在本文中，我们将学习如何使用A4988 **步进电机** 驱动模块来控制步进电机。A4988是控制双极步进电机的驱动模块，因为它非常便宜同时便于操作，因此在很多Arduino制作项目中A4988得到了广泛的应用。



因为A4988内置了译码器，我们可以可以通过控制器的2个引脚来控制步进电机，一个控制旋转方向，另一个控制步数。A4988提供了五种不同的微步控制：全步（full-step），半步（haft-step），四分一步（quarter-step），八分之一步（eight-step）和十六分之一步（sixteenth-step）。此外，它还配备了一个调节电流输出、过热保护和过流保护的电位计。A4988的逻辑电压范围是：3~5.5V，如果配备较好的散热条件每相最大电流可达2A，在没有配备散热器的情况下，每相连续电流最好控制在1A范围内。

Minimum Logic Voltage: (最小逻辑电压)	3V
Maximum Logic Voltage: (最大逻辑电压)	5.5 V
Continuous current per phase: (连续电流)	1 A
Maximum current per phase: (最大电流)	2 A
Minimum Operating Voltage: (最小工作电压)	8 V
Maximum Operating Voltage: (最大工作电压)	35 V

A4988驱动模块的引脚定义图



现在，让我们看一下A4988的引脚图，然后将其与步进电机和 **Arduino** 控制器连接起来。A4988驱动模块右上侧的两个引脚（VMOT、GND）是外部供电引脚，目的是能给电机提供足够的动力输出，**供电范围：0-35V**，此处使用一个47uf的电解电容来保护驱动板免受瞬时电压的冲击；右下侧的VDD和GND我们需要将它连接到Arduino控制板的5V电源和GND引脚上；1A和1B引脚连接到步进电机的一相，2A和2B引脚连接到步进电机的另一相；左下角的STEP和DIR连接至Arduino控制板的P3和P4引脚，这两个针脚主要用于控制电机的运动。Direction引脚控制转动方向，STEP针脚用于控制电机旋转的步数控制。

下一个是SLEEP针脚，低电平使能使模块处于休眠模式，当电机不工作时，它可以最大程度地降低功耗，默认为高电平。RESET针脚，如果这个针脚的输入是低电平，那么所有的微步设置都将被忽略掉。**因此上图将SLEEP和RESET针脚连接起来，目的是将RESET针脚设置为高电平，以便模块可控。**

MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Halft Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth step
HIGH	HIGH	HIGH	Sixteenth Step

MS1, MS2 和 MS3针脚用于微步设置，如上图所示，需要进行设置。ENABLE针脚用于打开和关闭场效应管的输出，低电平打开，高电平关闭。

完成本例的费用测算

1. [Arduino Mega](#)
2. [A4988驱动板](#)
3. [NEMA17步进电机](#)
4. [12V2A电源](#)
5. [面包板](#)

Arduino使用A4988控制电机代码

```
const int stepPin = 3;
const int dirPin = 4;

void setup() {
  // Sets the two pins as Outputs
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
}
void loop() {
  digitalWrite(dirPin,HIGH); // Enables the motor to move in a particular direction
  // Makes 200 pulses for making one full cycle rotation
  for(int x = 0; x < 200; x++) {
    digitalWrite(stepPin,HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin,LOW);
    delayMicroseconds(500);
  }
  delay(1000); // One second delay

  digitalWrite(dirPin,LOW); //Changes the rotations direction
  // Makes 400 pulses for making two full cycle rotation
  for(int x = 0; x < 400; x++) {
    digitalWrite(stepPin,HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin,LOW);
    delayMicroseconds(500);
  }
  delay(1000);
}
```