

本文档经常更新，需要查看本文档时候，请从 www.robomodule.net 进入下载专区重新下载。

常见疑问解答---关于电压问题

Q、驱动器供电电压问题

101 供电范围：15-33V

105 供电范围：7-26V

102、103、104、105+、106、106+、107、108、201 供电范围：7-33V

301 供电范围：10-55V

302 供电范围：10-45V

303、303+、401、402、403、405 供电范围：10-58V

Q、为什么会有 PWM 限制值这个参数

考虑到有可能出现一种情况，即驱动器的供电电压大于电机的额定电压。比如驱动器供电是 48V，而电机的额定电压是 24V 这种情况。有了 PWM 限制值这个参数，因为其满值是 5000 代表允许 100%电压输出，则直接限制为 2500，则可以直接限定驱动器的最大输出电压为 24V。这样就可以保证电机无论在何种工况下，都能不超出额定电压使用。

当驱动器供电电压与电机额定电压一致时候，此参数设置为 5000 即可，以后不需要再改动。

Q、使用 RMDS 系列驱动器，当供电电压为 12V，能不能使用 24V 的电机？

答案是可以的，而且不需要任何的注意事项。

就是有个缺点，24V 的电机，只有在 24V 的电压下，转速才能达到其最大额定转速。驱动器使用 12V 供电时，转速只能达到 24V 电机最大额定转速的一半。

不过好在，一般应用都不会用到电机的最大额定转速，所以也就没什么好纠结的。如果手头有 24V 的电机，尽管用吧。

如果转速达不到要求，参考如下解决方案：

解决方案 1：将驱动器的供电改为 24V。（RMDS-10x，20x，30x，40x 都支持的）

解决方案 2：购买额定电压为 12V 且转速满足要求的电机。

Q、使用 RMDS 系列驱动器，当供电电压为 24V，能不能使用 48V 的电机？

答案是可以的，而且不需要任何的注意事项。

就是有个缺点，48V 的电机，只有在 48V 的电压下，转速才能达到其最大额定转速。驱动器使用 24V 供电时，最大额定转速只能达到 48V 电机最大额定转速的一半。

不过好在，一般应用都不会用到电机的最大额定转速，所以也就没什么好纠结的。如果手头有 48V 的电机，尽管用吧。

如果转速达不到要求：

解决方案 1：购买支持 48V 电源输入的驱动器，比如 RMDS-303+、RMDS-401 以及后续版本，并且供电 48V。

解决方案 2：购买额定电压为 24V 且转速满足要求的电机。

Q、使用 RMDS 系列驱动器，当供电电压为 24V，能不能使用 12V 的电机？

答案是可以的，这个问题应该回答的是，24V 供电的驱动器，接入 12V 的电机，如何能更安全的使用它。

当使用开环模式，则把输出的 PWM 最大值，限制在 2500，因为满值是 5000，2500 是 5000 的一半，表示电机最大能得到的电压是 24V 的一半，即 12V。

当使用速度模式，则一样把 PWM 的最大值，限制在 2500，原理同开环模式。这样驱动器在 PID 调节之后输出的最大电压给到电机，也不会超过 12V。

当使用电流模式，则一样把 PWM 的最大值，限制在 2500，原理同开环模式。这样驱动器在 PID 调节之后输出的最大电压给到电机，也不会超过 12V。

其他模式类似，都是一样的。

当使用电流速度位置模式的时候，没有 PWM 的这个参数，则把电流设定为小于电机的最大带载电流即可。

Q、使用 RMDS 系列 301/303/303+/401/402 或其后续版本等支持 48V 的电压输入的驱动器，能不能使用 24V 的电机？

答案是可以的，这个问题应该回答的是，48V 供电的驱动器，接入 24V 的电机，如何能更安全的使用它。

当使用开环模式，则把输出的 PWM 最大值，限制在 2500，因为满值是 5000，2500 是 5000 的一半，表示电机最大能得到的电压是 48V 的一半，即 24V。

当使用速度模式，则一样把 PWM 的最大值，限制在 2500，原理同开环模式。这样驱动器在 PID 调节之后输出的最大电压给到电机，也不会超过 24V。

当使用电流模式，则一样把 PWM 的最大值，限制在 2500，原理同开环模式。这样驱动器在 PID 调节之后输出的最大电压给到电机，也不会超过 24V。

其他模式类似，都是一样的。

当使用电流速度位置模式的时候，没有 PWM 的这个参数，则把电流设定为小于电机的最大带载电流即可。

常见疑问解答——关于速度问题

Q、关于速度单位

速度单位，为 RPM，中文名：转每分钟。
也就是用一分钟能转多少圈来表示速度大小。

比如汽车的转速表也是 RPM 为单位，一般汽车发动机待机的速度为 800RPM 左右，正常行车时候，转速在 1000RPM 到 6000RPM。

再比如一般家用电风扇，转速在 2000RPM 左右。

Q、关于速度反馈的精度问题。

速度的单位是 RPM，意思是转/分钟。



这里涉及到一个 1ms 内速度反馈精度的问题，比如 500 线的编码器，1ms 的速度反馈精度为 30RPM，也就是说，驱动器对外输出的速度测量值，在这里都是 30 的倍数。比如速度为 1020RPM，速度为 990RPM，速度为 30RPM，速度为 60RPM 等等。为什么会这样？

还上面的 500 线编码器为例，对驱动器来说，驱动器是对单位时间内采集到的脉冲的个数来计算速度，这里的单位时间，驱动器内部设置的是 1ms。电机转动速度极低的时候，驱动器 1ms 内要么采集到 1 个脉冲，要么没采集到脉冲。对于没有采集到脉冲，速度测量结果为 0 RPM 无疑，对于采集到 1 个脉冲的情况，速度测量结果如何计算？一分钟有 60 秒，1 秒有 1000 个毫秒，那么按 1ms 刚好采集到一个脉冲的情况算，一分钟会采集到 60×1000 个脉冲。在此，编码器四倍频后一圈是 2000 个脉冲，那么一分钟电机转了多少圈就可以计算出来了，也就是 $60 \times 1000 / 2000 = 30$ 圈，也就是说电机一分钟转了 30 圈，专业的说，速度为 30RPM。也就是说，对驱动器而言，如果以 qc/ms 为单位，最低速度为 0qc/ms，或者 1qc/ms，但是要是以 RPM 为单位，最低可测速度则为 0RPM 或者 30RPM。

也就是说，上面的 30RPM，对于这个 500 线编码器，1ms 统计一次采集的脉冲数来计算速度而言，这个反馈精度 30RPM 是不可细分的。

对于 1000 线编码器，1ms 统计一次采集到的脉冲数的计算速度方式，反馈精度是 15RPM。

对于 2500 线编码器，1ms 统计一次采集到的脉冲数的计算速度方式，反馈精度是 6RPM。

对于 10 线编码器，1ms 统计一次采集到的脉冲数的计算速度方式，反馈精度是 1500RPM。
其他编码器计算方式同上。

如何获得更高的速度反馈精度？

对于 500 线编码器而言，最小反馈的非 0 速度为 30RPM，那么也就是说，统计 30 次的速度，就可以把显示精度变为 1 RPM 了。其他精度编码器计算方式也是一样的。

看到这里，有人可能有疑问，为什么不把采样周期延长以获得更高的速度测量精度，比如 1ms 统计一次脉冲数，变为 10ms 一次，100ms 一次呢？因为电机的机械响应速度在 1ms 左右，1ms 左右的控制周期，可以获得比较好的控制效果。

举个例子，汽车自动驾驶在 100km/h 的速度上，1s 的时间会移动 27.78 米，如果以 1s 的时间为控制周期，遇到前车、弯道、周边车的突然变道，根本就调节不过来。正常人的反应时间在 100ms 左右，自动驾驶的控制周期，当以这个值为界限，不得高于这个值，因为交规是按人与车的和谐统一的标准来指定的。同样对电机而言，控制周期当以电机的反应时间来设定最为合适，如果控制周期更短，同样的编码器下，测量的速度精度就更差了，所以按照电机的响应时间来设定控制周期，是最为合适的。

可能还有人会有这样的疑问，既然 500 线下编码器的最小测量的非 0 速度为 30RPM，那么速度模式下如何做到 1RPM 的可控速度？

你可以这样认为：比如输出 1RPM 速度，可能要先输出 29ms 的 0RPM，然后再输出 1ms 的 30RPM 的速度，此 30 个 ms 的周期平均下来的速度就是 1RPM 了。因为积分环节是不清零的，所以这个 1RPM 的长期统计是不会有误差的。比如可以拿不带减速器的电机，跟时钟的秒针对比，秒针 1 分钟转一圈，电机给定 1RPM 的速度，也必然是 1 分钟转 1 圈。

Q、RoboModule 的 RMDS 系列驱动器，速度取值范围是-32768~+32767，单位是 RPM。这个数值有多大？会不会出现超出这个速度的情况？

RoboModule 驱动器，支持最大的转速为 32767RPM，大部分的电机转速均在 10000RPM 以下，能到 2 万转每分钟的比较少见，超过 3 万转的就非常罕见了。

Q、速度模式下，无论如何调节速度值（RPM），速度都不变是怎么回事？

如果已经接了编码器，快速解决方案：将电机的 MT1、MT2 接线对调，然后重新测试即可。

如果上述解决不了，那就慢慢看下面这一大段：

有以下原因可能导致了上述情况

1. 没有接编码器
2. 接了编码器，但编码器线接错了，导致没有反馈信号
3. 接了编码器，但编码器线接触不良，导致只有一路的反馈信号或者两路反馈信号都没有
4. 接了编码器，但编码器方向与电机方向不匹配

电机有正反转两个方向，编码器的反馈也有正负值两个方向，只有电机方向和编码器方向匹配了，速度闭环的模式才会起作用。

如何知道电机和编码器方向匹配了还是不匹配？

调试软件上进入开环模式 2（反馈速度） 这个模式

先验证正方向：

PWM 给值给 500，或者 1000，让电机转起来，如果电机没转起来，继续加大这个 PWM 值，直到电机转起来。

观察右边的反馈窗口，如果全为 0，则说明编码器没有接，或者编码器接线接反了导致编码器不工作，或者编码器接触不良。

如果全为比较稳定的一个正数值，比如 1200，则表示电机和编码器的方向是匹配的。

如果全为比较稳定的一个负数值，比如-1200，则表示电机和编码器的方向是不匹配的。

上述是正方向的验证，下面再赘述一下负方向的验证。

PWM 给值给-500，或者-1000，让电机转起来，如果电机没转起来，则继续加大这个 PWM 值，直到电机转起来。

观察右边的反馈窗口，如果全为 0，则说明编码器没有接，或者编码器接线接反了导致编码器不工作，或者编码器接触不良。

如果全为比较稳定的一个负数值，比如-1200，则表示电机和编码器的方向是匹配的。

如果全为比较稳定的一个正数值，比如 1200，则表示电机和编码器的方向是不匹配的。

只有上述的正方向的验证和负方向的验证都通过了，才能使用速度位置闭环相关的模式：速度模式、电流速度模式、电流位置模式、速度位置模式、电流速度位置模式等。。。

如果上述的正方向和负方向的验证都不通过，且正转时候编码器反馈负数，反转时候编码器反馈正数，则将电机的 MT1 和 MT2 线对调即可。

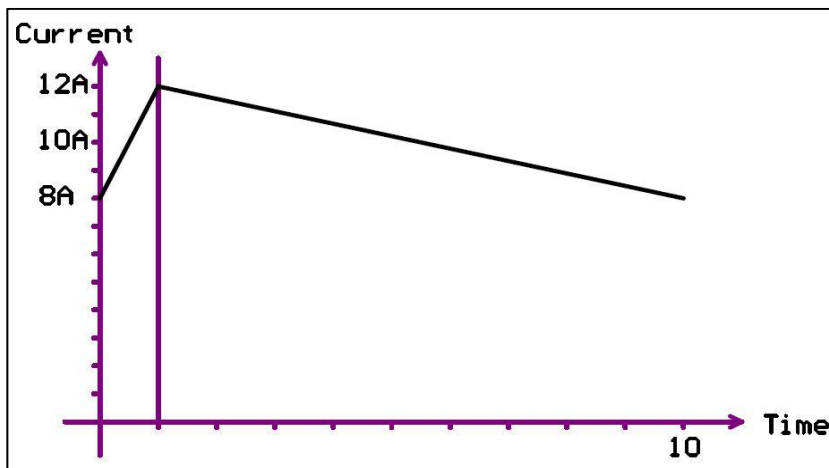
常见疑问解答——关于电流问题

Q、关于电流单位

电流单位，为 mA，中文名：毫安。

注意，所有运动模式，包括电流模式、电流速度模式、电流位置模式、电流速度位置模式，其设定电流值和反馈电流值的单位都是毫安，mA。

Q、电流模式，在电机完全堵转的时候，为什么供电电源显示的电流要比驱动器上软件上反馈的电流要小很多？



驱动器是以 PWM 的形式，将供电电源给的电压施加给电机。假设驱动器工作在电流模式下，设定的电流为 10A，由于电机的内阻比较小，稳态保持 10A 的电流输出的时候，占空比才刚刚达到 10%。

对于供电电源来说，仅有十分之一的时间里面，输出了 10A，其余十分之九的时间里面，是关闭输出的。因此对于供电电源来说，统计的电流是一段时间的平均电流，那么此时的平均电流为 $(10 \times 1 + 0 \times 9) / 10 = 1A$ 。

而对于电机来说，完全堵转的电机可以等效为一个电感，学过电路的人都知道，电感的电流不会发生突变，那么在此通电的十分之一的时间里面，电机的电流慢慢达到并超过了 10A，而接着的十分之九的时间里，电机的电流在缓慢下降，并不为零，而这整个变化的周期里面，平均电流是 10A。而驱动器的软件检测到的线圈上的平均电流也就是 10A 了。

因此明显可见，驱动器上反馈的电流是电机线圈上的真实的电流，而电源上的电流，仅仅只是输出的电流。

讲到这，有人会有疑问，这个是不是违背了能量守恒定律（比如想当然的认为电源功率 $VCC \times 1A$ ，电机功率 $VCC \times 10A$ ），实际上并没有违背能量守恒定律。电源输出的电压是 VCC，平均电流是 1A，功率是 $VCC \times 1A$ ，而电机上得到的平均电压，则是 $0.1 \times VCC$ ，平均电流为 10A，则电机上的功率为 $0.1 \times VCC \times 10A$ ，同样也是 $VCC \times 1A$ 。

Q、电流模式下，在电机旋转起来后，为什么反馈电流总是比设定电流要小很多？

比如设定电流为 1A，功率比较小的电机在不堵转的情况下，往往电流反馈只有 100mA 不到，这是为什么？

对于小功率电机，设定的电流为 1A 后，电机加速起来，直至达到最大速度，此时电流稳定的停在某个数值，比如 100mA。

对于电机而言，设定某电流 I ，产生的力为 F （为方便列公式，不考虑电机有摩擦），电机便获得一个加速度，对应物理学的 $F=ma$ ，从这个公式来看，此加速度 a 是一直存在的，则从这个公式可以认为，电机一直加速，无穷无尽。。。

但事实上，电机最大转速也仅能达到额定的空载转速，因为转动的电机同时会发电，从能量守恒定律上看，这个发电的方向，一定是相反于电源方向的，否则电机速度就无穷大了。

上述的 $F = kI$ ， I 是输出给电机的电流， k 是电流转换为力矩的比例系数，而 $I = U/R$ ，而叠加在电机线圈上的电压 $U=VDD-n*v$ ， v 是电机的转速， n 是电机发电电压与转速的比值， VDD 驱动器输出电压。也就是说，电机线圈上的电压，是驱动器输出的电压减去电机发电电压。

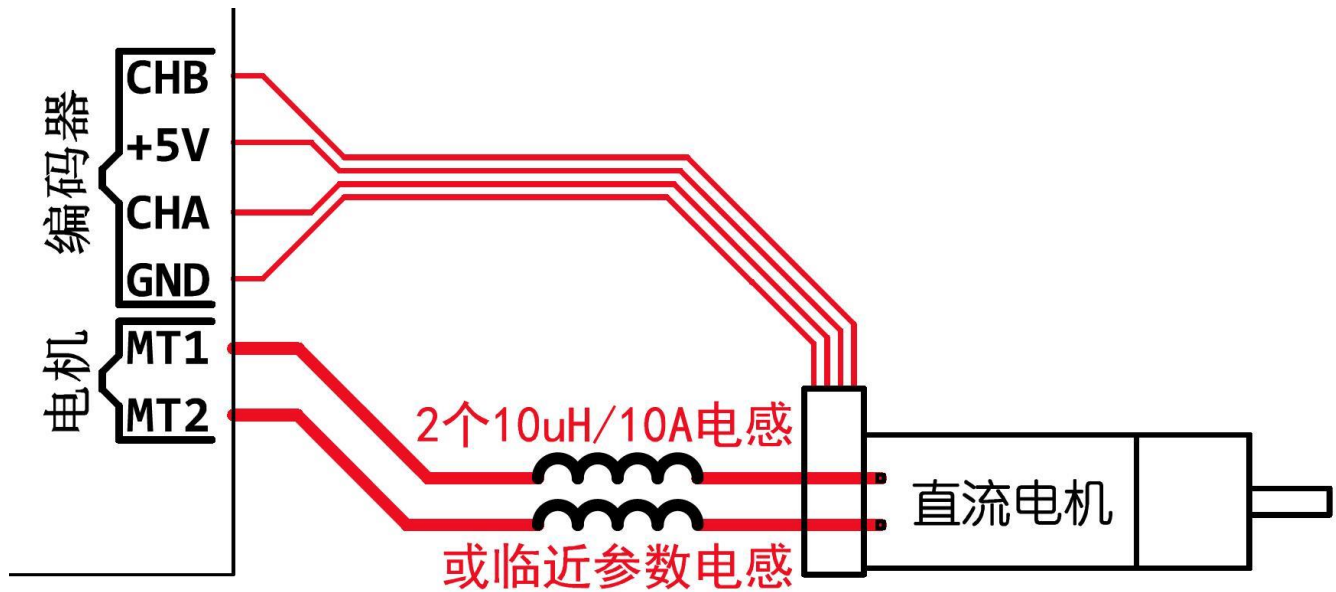
当电机转速 v 越来越大，则发电电压 $n*v$ 也越来越大，为了保持设定电流 I 的输出，驱动器输出电压 VDD 会一直增大，直到等于电源电压 VCC ，此时，电机的转速 v 不再升高（常识，给出了最大电压了，电机转速达到最大）。

此时， $VDD=n*v=VCC$ ，叠加在电机线圈上的电压为 $V=0$ ，电流 $I=0$ ，力 $F=0$ ，加速度 $a=0$ 。

因为存在摩擦力，所以需要有个微小的电流 i ，来产生一个力，去克服电机转动的摩擦力 f ，所以电流会稳定的停在某个很小的数值，此数值便是电机克服摩擦力所需要的电流。如果此数值较大，说明电机的阻力较大。

也就是说，上述的公式，实际为 $F-f = ma$ 。最后，输出的电流为 $i=f/k$ 。

Q、驱动器带动电机，正常运转下，电机发热怎么办？



RMDS-10x、20x、30x 等，采用上图解决方案，可以降低正常运转时的电机电流，且不损失功率。电感的作用是将脉动直流电转换为稳压直流电。对于 40x 要使用能过 30A 电流的电感才可以。

理论推导：

理想电感不消耗能量，所以无论是否有电感，电机的理论转速相同。

电机线圈上得到的电压 = 驱动器输出的电压 - 电机转动产生的反电动势，
因为转速稳定且不再增加，所以此时的反电动势值也是稳定的。

假设为占空比为 50%，此时线圈上得到的平均电压设为 U_a 。

当不加电感时，线圈在 50%的时间里面，得到了 $2U_a$ 的电压，在剩下 50%的时间里面，得到了 0 电压。

此时的线圈热功率： $P=2U_a*2U_a/R*0.5+0=2U_a*U_a/R$

加了电感之后，线圈在 100%的时间里面，得到的都是 U_a 的电压，此时的线圈热功率： $P=U_a*U_a/R$

可见：占空比为 50%时，线圈的热功率，在有电感的情况下，可以降低到原来的二分之一。

假设为占空比为 25%，此时线圈上得到的平均电压设为 U_b 。

当不加电感时，线圈在 25%的时间里面，得到了 $4U_b$ 的电压，在剩下 75%的时间里面，得到了 0 电压。

此时的线圈热功率： $P=4U_b*4U_b/R*0.25+0=4U_b*U_b/R$

加了电感之后，线圈在 100%的时间里面，得到的都是 U_b 的电压，此时的线圈热功率： $P=U_b*U_b/R$

可见：占空比为 25%时，线圈的热功率，在有电感的情况下，可以降低到原来的四分之一。

如果使用带电流显示的电源设备供电，也可以明显看出，加了电感之后，同样转速下，电源设备输出的电流明显减小。这个现象很好理解：因为加了电感之后，虽然电机线圈上得到的等效输出电压是一样的，但此时电机线圈少消耗了一些热功率，因为电源电压是不变的，热功率减小后，电源设备输出的电流也会减小。

正常情况下，如果电机长期使用发热量可以接受，就无需增加额外电感来增加走线负担了。在空载下，上述的 U_a , U_b 的值实际上是很小的，因为电机转速不变，所以加速度为 0，所以 U_a , U_b 产生的电流，电流产生的力矩，仅仅只是刚好抵消了摩擦力矩而已了。

常见疑问解答——关于位置问题

Q、关于位置的单位。

RoboModule 用户手册中，位置的单位，默认情况下是 qc，即四倍频后线数。

假设编码器参数为 500 线，在驱动器看来，经过四倍频后，这个编码器一圈是 2000qc。假设不存在减速器，那么在位置模式下，让电机转动一圈，就是给 2000qc 的值就行了。假设存在减速箱，还要乘上减速器的减速比得到的线数才是一圈，比如减速器是 16 倍力矩放大的，则一圈是 32000qc。

Q、RoboModule 的 RMDS 系列驱动器，位置取值范围是-2147483648~+2147483647，单位是 qc (四倍频后的线数)。这个数值有多大？会不会轻易溢出？

常见的电机，假设一分钟转速为 6000 转，编码器为 500 线，按这个电机来计算，假设电机全速运行，则一分钟可以跑 $6000 \times 500 \times 4 = 12000000$ qc，那么 2147483647qc 可以跑多少分钟？

$2147483647 / 12000000 = 179$ 分钟，约三个小时。这个范围对于 99% 以上的应用场合是足够了。

以上计算是对于单方向运动的场合，一般应用场合都是双向来回运动的，这样的场景，连续跑好几年都不会达到溢出的界限。

Q、位置模式下，给多少的 qc，电机刚好转过一圈？

假设电机编码器为 500 线，减速比为 16:1，则 $500 \times 4 \times 16 = 32000$ 为一圈。4 从何而来，双通道编码器都是四倍频的，下面解释四倍频：

编码器为 500 线的，则说明，编码器的钢盘上刻了 500 个孔，同时有 500 个遮挡，则对于 1 通道的编码器来说，产生了 1000 个上升沿+下降沿。对于双通道的编码器，错开 90 度相位摆放，则产生了不重合的 1000 个上升沿和下降沿。则一共有 2000 个上升沿和下降沿。也就是说，对于单片机来说，500 线的编码器，实际上可以获得一圈 2000 个上升沿和下降沿，即角度精度为 $1/2000 \times 360^\circ$ 。

Q、驱动器工作在位置相关的模式，还没到位的时候想要中途停下来，怎么做？

直接将 PWM 给 0 即可。如需恢复运动，将 PWM 恢复至原来的值即可。

Q、驱动器内没有直接的相对位置功能，那么如何发送相对位置？

先在程序中设置一个变量，比如 32bit 有符号的 history_position，用来保存历史位置。

```
int history_position = 0;
```

比如需要运动+500qc，那么发送位置时候，就发送： history_position + 500

然后执行 history_position += 500;

然后下次假如要再运动+1000qc，发送位置的时候，就发送： history_position + 1000

然后执行 history_position += 1000;

再比如要再运动-2000qc，发送位置的时候，就发送： history_position - 2000

然后执行 history_position += (-2000);

这样看上去的总体效果，就是相对位置了。

Q、如何找零位？

驱动器内没有寻零模式。但也有其他方式找零。比如，在外部有机械限位的情况下，让电机使用开环模式或者电流速度模式，让电机慢速小电流运动到机械卡位，同时让 can 总线反馈电流信息。当到达机械卡位时候，返回的电流绝对值会变大，且稳定。此时，复位驱动器即可。

然后重新进入位置相关的模式，即相当于从零点开始运动了。

常见疑问解答——关于十进制和十六进制问题

Q、十进制数如何换算成 16 进制数？（手工算法）

驱动器的 CAN 发送、RS232 发送、RS485 发送，比如 PWM 限制值、电流信息、速度信息、位置信息、电压信息。从上述的通讯线发送时，都要将人类认知的 10 进制数，转化为 16 进制数发送。

大前提：PWM 限制值、电流信息、速度信息等，皆为 16 位有符号整数类型。位置信息值为 32 位有符号整数类型。

下面列几个例子：

速度+1，发送的时候，实际发送的是 00 01

速度-1，发送的时候，实际发送的是 FF FF

速度+10，发送的时候，实际发送的是 00 0A

速度-10，发送的时候，实际发送的是 FF F6

速度+100，发送的时候，实际发送的是 00 64

速度-100，发送的时候，实际发送的是 FF 9C

速度+1000，发送的时候，实际发送的是 03 E8

速度-1000，发送的时候，实际发送的是 FC 18

速度+5000，发送的时候，实际发送的是 13 88

速度-5000，发送的时候，实际发送的是 EC 78

位置+1，发送的时候，实际发送的是 00 00 00 01

位置-1，发送的时候，实际发送的是 FF FF FF FF

位置+10，发送的时候，实际发送的是 00 00 00 0A

位置-10，发送的时候，实际发送的是 FF FF FF F6

位置+30000，发送的时候，实际发送的是 00 00 75 30

位置-30000，发送的时候，实际发送的是 FF FF 8A D0

位置+5 亿，发送的时候，实际发送的是 1D CD 65 00

位置-5 亿，发送的时候，实际发送的是 E2 32 9B 00

位置+21 亿，发送的时候，实际发送的是 7D 2B 75 00

位置-21 亿，发送的时候，实际发送的是 82 D4 8B 00

熟悉 C 语言的人一般都知道，上面的举例，与整型数在内存存放的方式完全一致。正数的存放方式很好理解，即直接将 10 进制转换为 16 进制即可搞定。

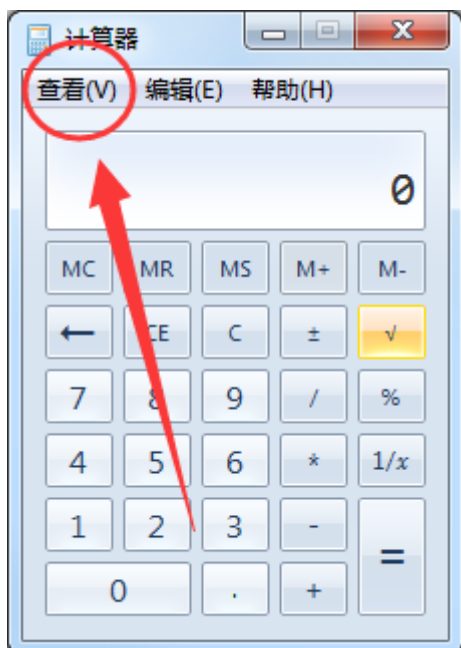
而负数的存放方式稍微麻烦一些，比如-1000 的速度，首先其反码为 1000，转换为二进制，得到 0000 0011 1110 1000，然后求补码，即 1111 1100 0001 0111，然后将这个补码加 1，即 1111 1100 0001 1000，把这个数据整理为十六进制，得到 FC 18，对照上表，这个结果是正确的。

简单的一句话说：反码的补码加一，即可求得负数在内存中的存放。

上面列举的，可以直接使用 Windows 自带的计算器算出来，以 win7 为例，假设要计算+1000 的十六进制值数。

Q、16 位正整数的十进制数如何换算成 16 进制数？（使用 win7 电脑计算器）

1. 首先打开计算器，点击左上角的查看(V)，选择程序员(P)，进入程序员模式。



2. 此处举例计算速度，(因 PWM 限制值、电流值与速度值都是 16 位有符号整型，所以此处操作完全相同)，比如求 1000 的十六进制数，则首先如图设置，设置为“十进制”，“字”，可以看到，右边二进制显示区为默认显示了 16 个 0。



3. 直接输入 1000，然后点击“十六进制”，完成转换。



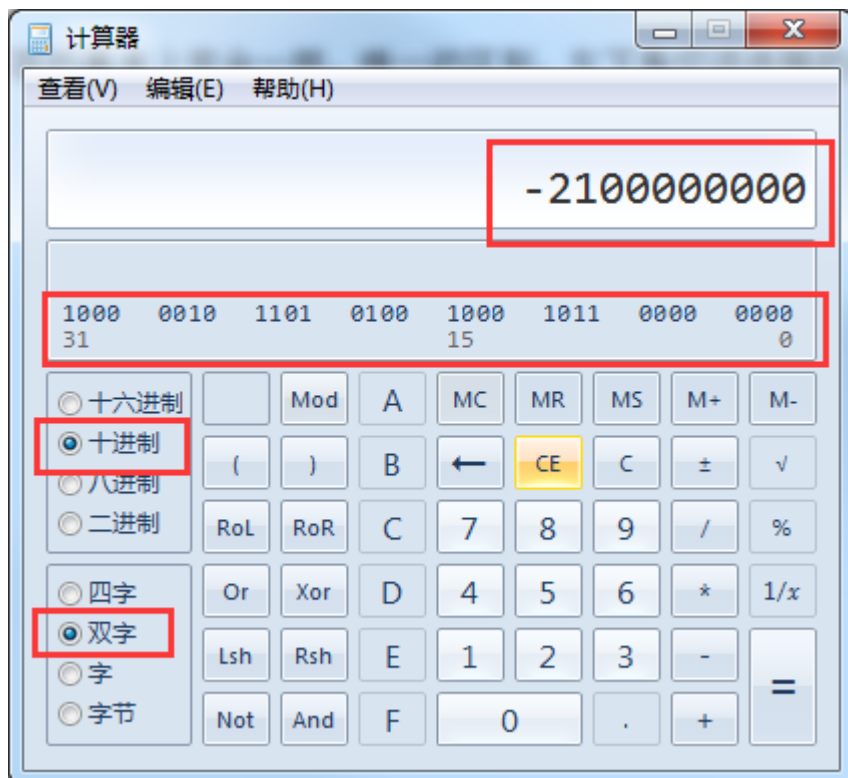
Q、16 位负整数的十进制数如何换算成 16 进制数？（使用 win7 电脑计算器）

大致方式与上条步骤完全一致。例如-1000 的速度，输入的时候，首先输入 0，再按一下“-”号，然后填入 1000，然后按一下“=”号，即可得到-1000。然后直接点十六进制的按钮即可得到。



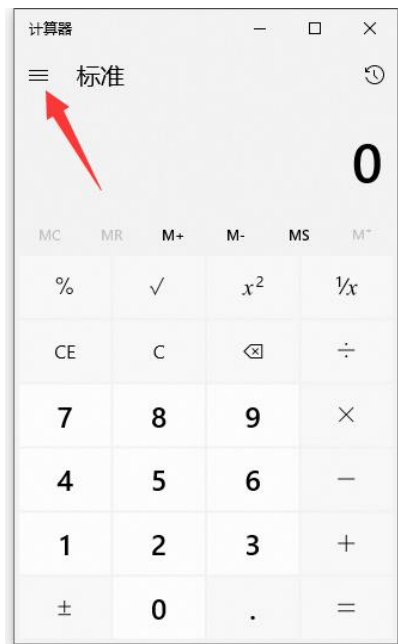
Q、32 位整数的十进制如何换算成 16 进制数？（使用 win7 电脑计算器）

换算步骤与上述两条解答的基本上完全一样，唯一的区别，左下角应该选择的是“双字”，举例：位置值为-21 亿。

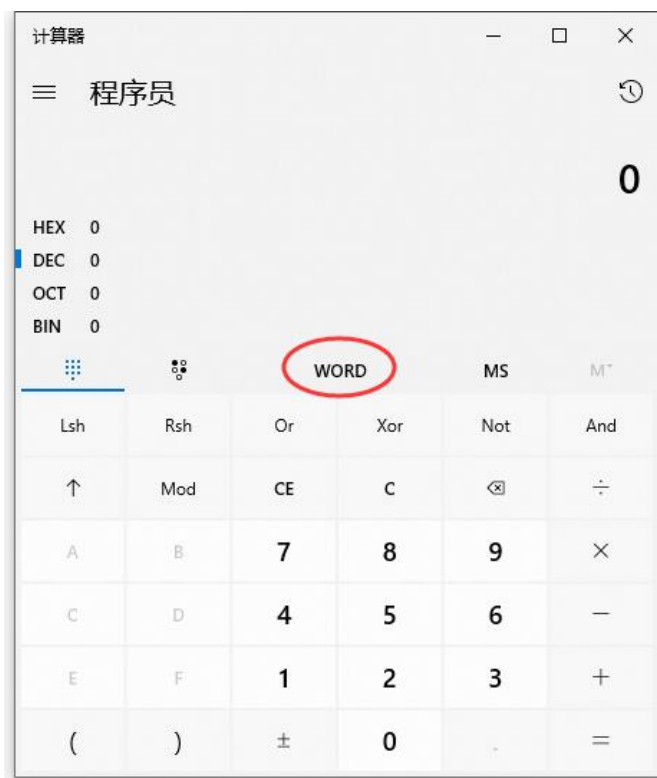


Q、16 位正整数的十进制数如何换算成 16 进制数？（使用 win10 电脑计算器）

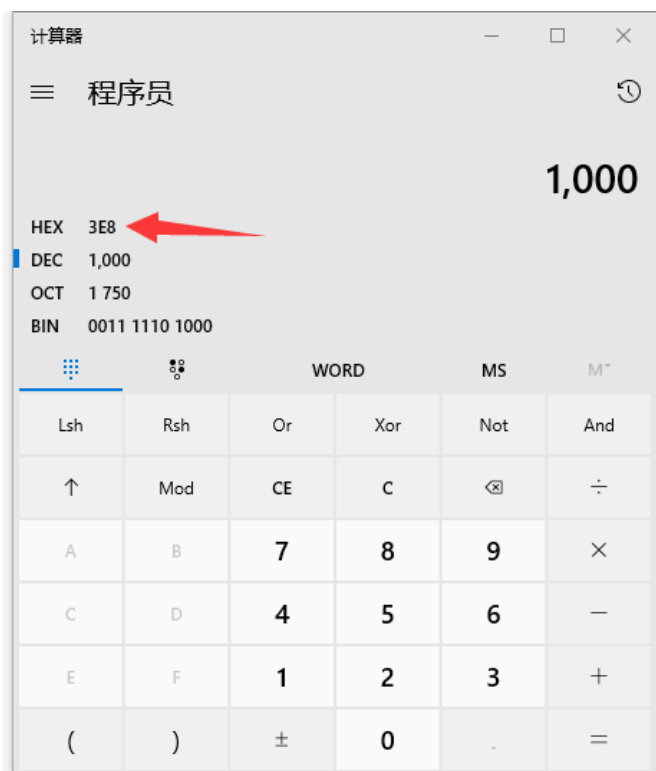
1. 首先打开计算器，点击左上角的“三”，选择程序员，进入程序员模式。



2. 此处举例计算速度，（因 PWM 限制值、电流值与速度值都是 16 位有符号整型，所以此处操作完全相同），比如求 1000 的十六进制数，则首先如图设置，设置为“WORD”，此处 WORD 表示 16bit。

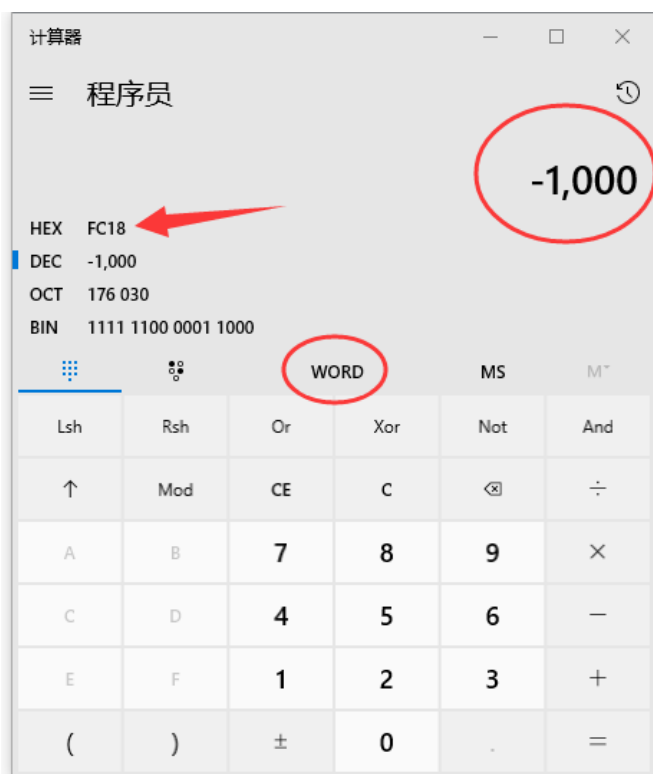


3. 直接输入 1000，就可以看到 HEX 显示为 3E8, 也就是说 1000 的十进制数，用十六进制表示为 03 E8



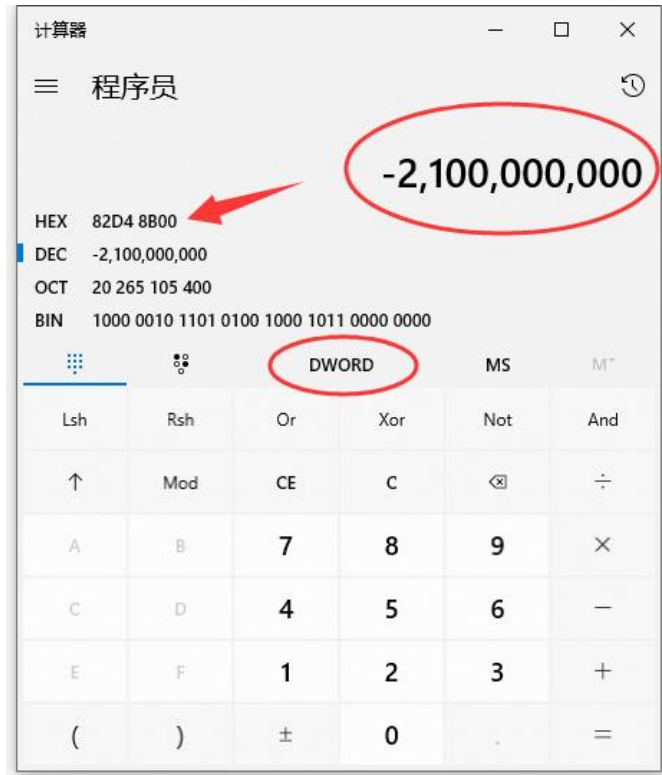
Q、16 位的负整数的十进制数如何换算成 16 进制数？（使用 win10 电脑计算器）

大致方式与上条步骤完全一致。例如-1000 的速度，输入的时候，首先输入“-”，也就是负号/减号，然后填入 1000，再按一下“=”号，即可得到-1000，可以看到，16bit 下，十进制的-1000 表示为 FC 18



Q、32 位整数的十进制如何换算成 16 进制数？（使用 win10 电脑计算器）

换算步骤与上述两条解答的基本上完全一样，唯一的区别为，这里应该选择“DWORD”，举例：位置值为 -21 亿。



可以看到十进制数的-2100000000，在 32bit 的十六进制中表示为 82 D4 8B 00

常见疑问解答——关于 485 和 CAN 问题

Q、如何判断驱动器的 CAN 收发器损坏？

对于 101、102、103、104、106、301、302 等早期驱动器，CAN 收发器共模电压较低，仅有 16V，在异常操作下有可能损坏。如果常用 CAN 总线，对于上述型号的驱动器，CAN 收发器有损坏的可能。如何判断 CAN 收发器损坏？首先需要一个电流显示精度 10mA 或者 1mA 的电源，1mA 精度最佳。以上驱动器在仅接通电源的情况下（编码器要断开），电流一般都在 10-25mA，如果电流大于 30mA，则可以认为驱动器的 CAN 收发器损坏。

对于 106+、107、108、201、303、303+、401、402 等后续型号驱动器，吸取了早期经验之后，使用的 CAN 收发器共模电压达到 58V，CAN 收发器损坏率低于万分之一，非恶意操作下，基本不可能损坏。

Q、如何判断驱动器的 485 收发器损坏？

对于 101、103、106、106+、302、303、401 等型号驱动器，485 收发器共模电压较低，且无内置 TVS/ESD 二极管，如果常用 485 总线，对于上述型号的驱动器，在异常操作下有可能损坏。判断是否损坏的方式同上一条解答。由于无法找到共模电压更高的 485 收发器，在上述型号除外的后续型号中，比如 108、303+、402 等，如带有 485 功能，则会增加额外的 12V 和 -7V 的 ESD 保护，因电路空间有限，不够做更多的保护措施，依旧无法避免高压直连的情况下损坏，仅能保证正常使用情况下损坏率降低到一个很低的值。

如使用 STM32 单片机当主控，请尽量使用 CAN 总线，可以减少不少麻烦。具体请参考下一条。

Q、CAN 总线相对于 485 总线优势在哪里？

CAN 总线相比 485，无论哪方面，都是优势。

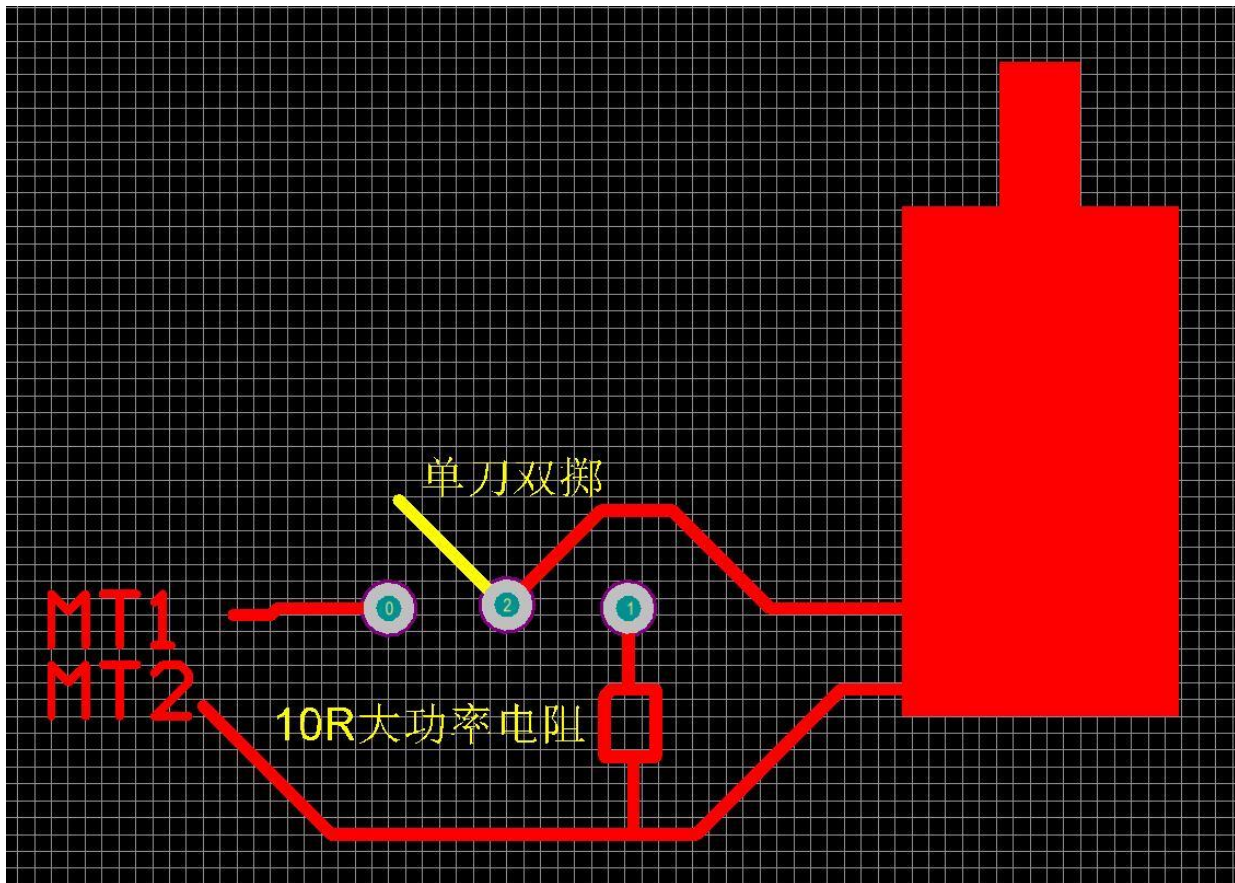
1. CAN 支持的通信速率可高达 1M；485 通信速率高了极不稳定。
2. CAN 数据包由硬件打包，不可能产生接收错位；485 数据包只能软件处理，产生错位可能性很大。
3. CAN 总线共模耐压高，为 58V，很难损坏；485 总线共模耐压只有 12V，容易坏。
4. CAN 总线为全双工，支持多主多从，硬件自动仲裁各设备的数据包优先级；485 是半双工，只能存在一个主机，其他设备都只能是从机，从机不能主动对外发数据。
5. CAN 的发送接收由硬件自动无缝切换，不会出错；485 需要软件干预切换，时序要求严格，否则一定出错。
6. CAN 数据包有硬件校验，出错率极低；485 无校验，走线好坏对出错率影响很大。
7. CAN 带有硬件过滤器，可以将不属于自己的数据包从硬件上过滤掉，不需要软件干预。对于多个驱动器一起使用的情况下，不会被别的驱动器的数据包打断并占用 CPU 资源。485 总线没有硬件过滤器。
8. CAN 可支持 120 个驱动器和多个主控。485 最多只能接 15 个驱动器，以及最多只能有 1 个主控。

常见疑问解答——关于电机抱死与急停的问题

Q、如何安装急停开关比较安全？

当程序异常导致电机不受控时，需要紧急切断电源。但如果直接切换总电源的话，如果电机还在疯狂发电，则可能会损坏系统电路。

下面是推荐的切断电机电源的连线方式：



Q、阻尼抱死和闭环抱死的区别？

驱动器可以设置上电抱死或者上电不抱死。

此功能仅对不启用 CTL1、CTL2、ASIN、DSIN 的功能的情况下有效。

此处的上电抱死，仅仅只是阻尼抱死，相当于将电机的两根线直接短接产生的阻尼，此时，电机还是可以被转动的，但是转动的阻尼比较大。

闭环抱死的功能存在于：速度模式，位置模式、速度位置模式。进入上述模式后，在给了第一个数据命令之后，驱动器抱死生效。此时无论使用多大的外力，都无法转动电机。

为什么要等到给了第一个数据命令后，抱死才生效。因为上述模式都有一个传入参数，即限制 PWM 值，刚进入上述模式时候，PWM 限制值默认为 0，所以在给了 PWM 值之后，抱死才会生效。

常见疑问解答——关于固件升级失败的问题

Q、使用 RMDS-40x 系列驱动器，如何擦除固件？

10x, 20x, 30x 系列驱动器，如果下载了错误的固件造成无法启动，则需要返厂维修。40x 系列可以自行擦除固件，然后自己重新升级固件即可。

40x 系列驱动器，预留了 SOS 固件一键擦除端口。在用户下载了错误的固件升级包造成驱动器无法启动的时候。可以用此 SOS 端口将错误的固件擦除，重新下载。

方法：

1. 断电。
2. 将 SOS 端口与旁边的 GND 端口用线连一起。
3. 通电，等待驱动器响一声。
4. 断电。
5. SOS 端口与 GND 端口断开连接。至此，固件擦除成功。接着重新升级固件然后恢复出厂设置即可恢复正常使用。

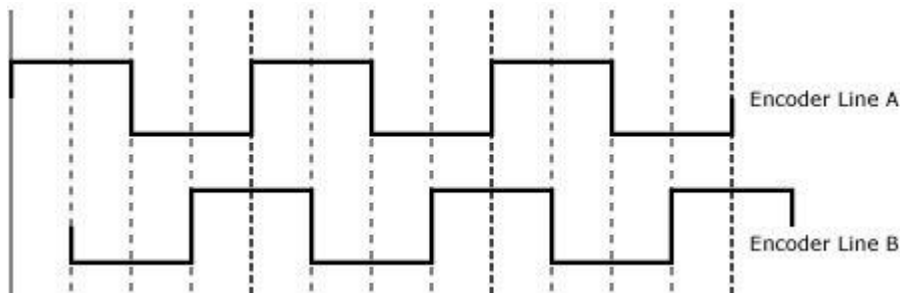
常见疑问解答——关于编码器问题

Q、如何用示波器测量编码器信号质量

需要满足以下，才能开始测试：

1. 需要一台双通道示波器
2. 编码器的 5V 和 GND 需要有供电。如果是 NPN/PNP 集电极开路输出的编码器，对 CHA 和 CHB 需要用 10k 电阻上拉到 5V 才可以测量的到信号。
3. 让电机带动编码器达到一个稳定的转速。可以使用驱动器，进入开环模式 2，给一个 1000 的 PWM，让电机转起来。也可以直接将电机接电源，调节电压到电机能稳定转起来的值。

以上条件满足后，开始测量，将示波器的 GND 接到编码器的 GND，然后两个探头打在 CHA 和 CHB 上。出现如下波形为正常。



正常的波形，满足 CHA 波形超前或者滞后 CHB 波形 90 度相位。

异常波形通常有如下情况（实践中经验总结）：

1. 其中一个通道为方波，另一个通道为三角波。
2. 正常波形的 CHA 超前或者滞后 CHB 约 90 度相位，异常波形的相位差不稳定，有时会接近 0。

Q、不带减速器的电机，当不知道编码器精度时候，如何测量得到编码器精度？

进调试软件的开环模式 3（反馈位置），用手电机拧过一圈，假如窗口反馈的值为 2000，则说明编码器的实际精度是 $2000/4=500$ 线

但这样验证精度不够，无法辨别编码器是 500 线还是 512 线，可以将电机用手慢慢拧 10 圈，假如窗口反馈的值为 20000，则说明编码器的实际精度是 $20000/10/4=500$ 线

通过 RS232 不能正常连接电脑调试软件的原因总结：

1. 驱动器没接电源，或者电源电压不在要求的范围。正常接上电源的驱动器，刚上电时候，蜂鸣器会响一声。如果驱动器刚上电，蜂鸣器没有响，下面的内容就不用看了。

供电范围：

101 供电范围：15-33V

105 供电范围：7-26V

102、103、104、105+、106、106+、107、108、201、202 供电范围：7-33V

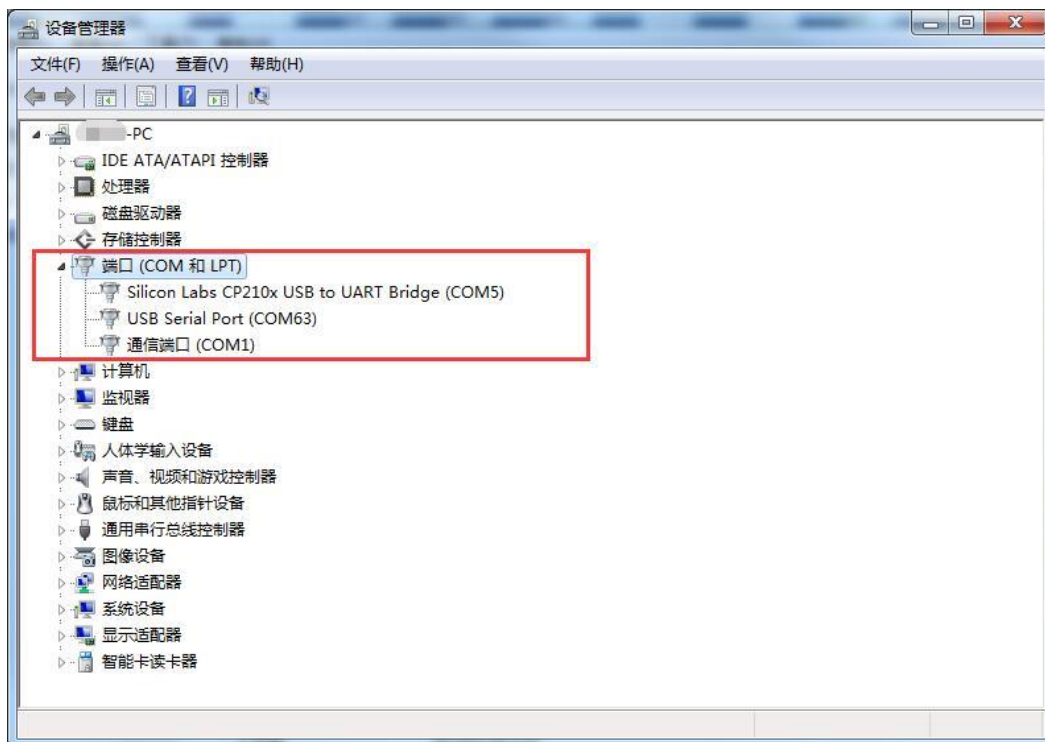
301 供电范围：10-55V

302 供电范围：10-45V

303、303+、401、402 供电范围：10-58V

2. **串口线的驱动器没有安装或者没有安装正确。**USB 转 RS232 的串口线，一般都需要驱动程序，如果没有安装驱动程序，电脑就无法识别 USB 转 RS232 串口线，也就无法生成正确的串口号。安装完串口驱动程序后，在电脑的设备管理器中可以看到串口的描述和串口号。

正确安装后，在设备管理器中显示如下：



其中，比如 FT232 的串口线会生成 USB Serial Port 的串口描述，CP2102 的串口线会生成 Silicon Labs CP210x USB to UART Bridge 的串口描述。

此处再次强调，不要使用 CH340 和 PL2303 的串口线。

3. **串口号选择错误。**进驱动器的调试软件，在首页上，要选择自己接入驱动器的串口线所对应的串口号才可以正确进入。

4. **波特率选择错误**。进驱动器的调试软件，在首页上，要选择正确的波特率。所谓正确的波特率，出厂默认为 115200，如果自己后期改过了，则要选择改过之后的波特率才可以进入。
5. **串口电平错误**。驱动器的串口电平是 RS232 电平，并非 TTL 电平。用户使用的串口线，如果是 USB 转 TTL 则不能使用。必须是 USB 转 RS232。
6. **从 RS232 的 DB9 串口接头出来的，到驱动器 232 端口之间的接线，接触不良或者断线**。如使用普通杜邦线，断线概率很高。
7. **CAN/485 线连着，且已经进入 CAN/485 的通信模式**。RS232、CAN、RS485 等通信接口不可同时使用，先入为主。抢占后不可被打断。如出现此情况，关闭主控的电源即可。
8. **驱动器的 232 端口收发器芯片损坏**。如果有带电插拔驱动器上的 232 端口的习惯，一定要改掉。带电插拔或者带电接线 232 端口，有很大的概率会损坏 232 收发器芯片。
9. **USB 转 RS232 线的 RS232 收发器芯片损坏**。这种情况也很常见。测试方法：短接 TX 和 RX，使用第三方串口助手自发自收即可测试。（如不懂如何操作此测试，基本上可以告别电子行业了）

RS232 通信数据出错原因总结：

1. 从 DB9 出来的，到驱动器 232 端口的线，长度不能过长。驱动器默认使用的波特率为 115200，为通用 RS232 的最高波特率，因此，对线的要求比较高，一般不能大于 0.2 米，如果使用符合技术要求的屏蔽线，则可以走的更长。
2. 从 DB9 出来的，到驱动器 232 端口的线，中间不能含有转接板。驱动器默认使用的波特率比较高，当有转接板的时候，信号会遭到严重破坏。以致于通信错误率极高。此处转接板包含：使用手工焊接的万用板来转接，印制电路板转接，鳄鱼夹转接，通通都不行。
3. 从 DB9 出来的，到驱动器 232 端口的线，太粗也不行。当频率比较高时，走线越细越好。粗的线是用来走大电流的，不是用来走信号的。一般用 24awg、26awg、28awg 标号的线最佳。
4. 使用了 CH340、PL2303 的 USB 转 RS232 线。CH340 和 PL2303，说明书中已多次强调不能使用，重要的事情再说一遍。CH340 和 PL2303 两个都不稳定，特别是不能用它们来升级固件，否则驱动器可能成为板砖。PL2303 和 CH340 两者都属于国产廉价芯片，成本不到 1 块钱。推荐的 FT232 和 CP2102 芯片，前者成本 15 元一片，后者也要 6 元多一片。贵有贵的道理，这两个 USB 转串口芯片是业界公认最稳定的 USB 转串口芯片，无需置疑。明知地沟油不能吃，还因为贪便宜买了地沟油的，最终只会害了自己。PS：一些无良奸商更过分的，直接将 TTL 反向就声称是 RS232 了，这种线，无论是什么芯片为核心的都是不能用的。