直流一体化力矩伺服

YZ-AIM_v2.56

一. 产品特性

1. 15 位绝对编码器，一圈脉冲高达 32768。
2. 多圈绝对值（需配电池）。脉冲模式：重新上电自动回断电位置。
   通信模式：可断电记录位置。
3. 多级 DD 马达结构，大扭力输出。
4. 一体化伺服，简化接线，体积超小。
5. 低噪音，低振动，高速定位，高可靠性。
6. FOC 场定向矢量控制，支持位置/速度闭环。
7. 可工作在零滞后给定脉冲状态，跟随零滞后。
8. 16 位电子齿轮功能。
9. modbus RTU 通信（19200，8,N,1）波特率可设置。
10. 位置模式，支持脉冲+方向信号
11. 具有堵转，过流保护，过压保护。

二. 参数表

型号参数		42AIM15	42AIM10
电源	电压	24VDC±10%	24VDC±10%
	电流	2.2A	1.6A
电机参数	扭矩	0.48NM	0.33NM
	额定转速	1000RPM	1000RPM
	最大转速	1500RPM	1500RPM
	功率	50W	35W
反馈信号		单圈 15 位磁电编码器（单圈 32768 脉冲）	
冷却方式		自然冷却	
重量			
位置控制模式	**最大输入脉冲频率**	500KHz	
	脉冲指令模式	脉冲+方向， A 相+B 相	
	电子齿轮比	设置范围 1~65535 比 1~65535	
	位置采样频率	2KHz	
保护功能		堵转报警	
通信接口		RS485（modbusRTU 19200,8,N,1）	
使用环境	**环境温度**	0~40°	
	电机允许最高温度	85°	
	湿度	5~95%	

型号参数		57AIM15	57AIM15H	57AIM30	57AIM30H
电源	电压	24~36VDC	24~36VDC	24~36VDC	24~36VDC
	电流	2.2A	2.2A	4.4A	4.4A
电机参数	扭矩	0.48NM	0.24NM	0.96NM	0.48NM
	额定转速	1000RPM	2500RPM	1000RPM	2500RPM
	最大转速	1500RPM	3000RPM	1500RPM	3000RPM
	功率	50W	50W	100W	100W
反馈信号		多圈绝对值编码器（单圈 32768 脉冲，单圈 15 位）			
冷却方式		自然冷却			
重量					
位置控制模式	**最大输入脉冲频率**	500KHz			
	脉冲指令模式	脉冲+方向， A 相+B 相			
	电子齿轮比	设置范围 1~65535 比 1~65535			
	位置采样频率	2KHz			
保护功能		堵转报警			
通信接口		RS485（modbusRTU 19200,8,N,1）			
使用环境	**环境温度**	0~40°			
	电机允许最高温度	85°			
	湿度	5~95%			

型号参数		60AIM25	60AIM25H
电源	电压	36VDC±10%	36VDC±10%
	电流	7A	7A
电机参数	扭矩	2NM	1NM
	额定转速	1000RPM	2500RPM
	最大转速	1500RPM	3000RPM
	功率	250W	250W
反馈信号		单圈 15 位磁电编码器（单圈 32768 脉冲）	
冷却方式		自然冷却	
重量			
位置控制模式	**最大输入脉冲频率**	500KHz	
	脉冲指令模式	脉冲+方向， A 相+B 相	
	电子齿轮比	设置范围 1~65535 比 1~65535	
	位置采样频率	2KHz	
保护功能		堵转报警	
通信接口		RS485（modbusRTU 19200,8,N,1）	
使用环境	**环境温度**	0~40°	
	电机允许最高温度	85°	
	湿度	5~95%	

二. 驱动器接口

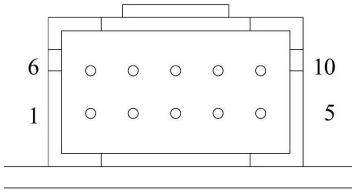
1. 电源与控制信号接口

端子序号	名称	功能
1	+24V	直流电源正极，+24V。 正负接反会直接短路电源，也可能损坏驱动器
2	GND	直流电源地。正负接反会直接短路电源，也可能损坏驱动器
3	PU+ (+5V)	脉冲控制信号：脉冲上升沿有效；PU-高电平时 3.3～5V，低电平时 0～0.5V。
4	PU- (PU)	为了可靠响应脉冲信号，脉冲宽度应大于 1.2μs。如采用+12V 或+24V 时需串电阻。
5	DIR+ (+5V)	方向信号：高/低电平信号，为保证电机可靠换向，方向信号应先于脉冲信号
6	DIR- (DIR)	至少 5μs 建立。DIR-高电平时 3.3～5V，低电平时 0～0.5V。

端子序号：面对端子，左边为第一。

AIM 系列采用差分式接口电路可适用差分信号，单端共阴及共阳等接口，内置高速光电耦合器，允许接收长线驱动器，集电极开路和 PNP 输出电路的信号。

2. 通信与输出接口



端子序号：面对端子，下排从左到右分别为 12345，上排从左到右分别为 6 7 8 9 10。

端子序号	名称	功能
1	NC	
2	485A	485 通信 正端
3	485B	485 通信 负端
4	EN+	使能信号正端。3.3~5V,如果电压为 24V 需要串一个 2K 电阻。
5	EN-	使能信号正端。
6	COM	输出信号与 485 电源公共地。
7	WR	报警信号输出，内部为光耦 NPN 输出。正常为高阻态，报警时与 COM 导通。
8	RDY/PF	伺服准备好信号/到位信号。上电自动工作以后有信号（导通），当跟随误差小于 0.5°为有信号（导通），跟随误差大于 0.5° 无信号（高阻态）。
9	ZO	编码器零点输出。有零点信号光耦 NPN 输出导通信号。
10	485_5V	485 通信 5V 电源，需要外部提供电源。（此电源通过控制器供电）

3. 状态指示与报警

开机后红灯绿灯都亮一次，用于检验 LED 是否工作正常。而后绿灯亮，红灯灭为正常状态。如果遇到报警状态，可以通过红色闪烁来判断原因，也可以通过 modbus 读取报警代码。

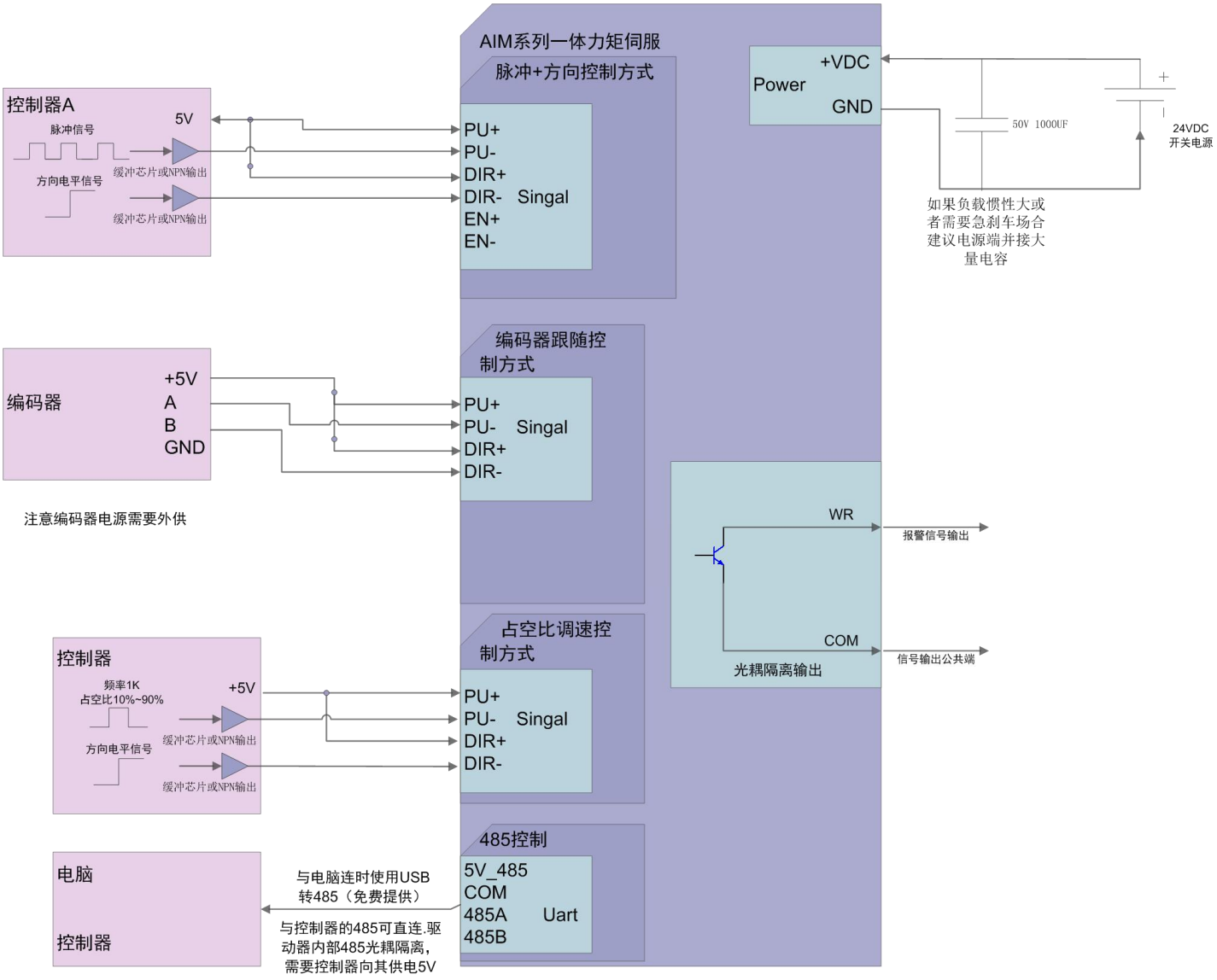
报警代码	红灯闪烁	报警原因	报警处理

0x10	一长闪	电池掉电报警	只提示，不停机
0x12	一长闪 2 短闪	过流报警	停机
0x14	一长闪 4 短闪	堵转报警	停机
0x15	一长闪 5 短闪	过压报警	停机，大惯性负载减速会发电，可能照成过压报警。需要在电源处加放电模块，或者电源处加大电容储能。

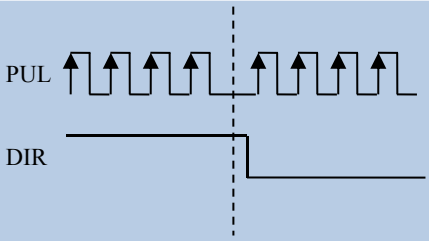
注：堵转报警，堵转时间可以设置，具体看寄存器说明。

三．驱动器接线图与控制方式

1. 驱动器典型接线图



2.指令脉冲+方向位置控制模式

脉冲指令形式	CCW	CW	参数设定值
脉冲列符号			指令脉冲+方向

如果需要 3200 脉冲一圈

电子齿轮设置为　32768（编码器一圈脉冲数） 比　3200（需要设置的一圈脉冲数）

约分后为：256 比 25

如果需要 8192 脉冲一圈（默认参数）

电子齿轮设置为　32768（编码器一圈脉冲数） 比　8192（需要设置的一圈脉冲数）

约分后为：32768 比 8192

约分后为：4 比 1

注意：能约分尽量约分，电子齿轮分子为 32768，数值太大，会影响跟随性能

指令脉冲频率 =(需要电机运行的转速/60) * 一圈的脉冲数

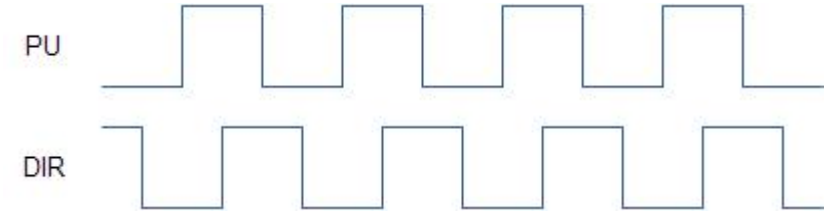
例如： 需要点击 1000RPM 一圈脉冲数为 8192

脉冲频率 = 1000/60 * 8192= 136533HZ

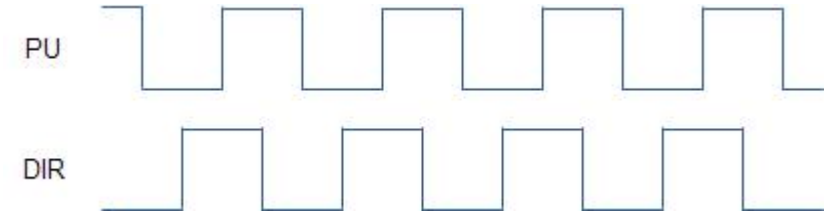
3. 正交指令脉冲位置控制模式

通过设置特殊功能（0x19 地址）为 2，重新上电后，即为编码器跟随模式。这种模式可以用于编码器跟随，如一个轴接了编码器，将编码器输出接到驱动器(接线方式如 驱动器典型接线图)，驱动器就能控制伺服电机，按输入编码器的信号，随动于控制的编码器。可以通过调节电子齿轮，来设置控制编码器和电机转动角度的比例。

正转脉冲：



反转脉冲：



电机转动的方向：PU 上升沿超前 DIR 上升沿 为正转。PU 上升沿滞后 DIR 上升沿 为反转。

四．参数调试

根据电机所接负载不同，参数需要调整才能达到最佳效果。

1. 内部加减速曲线

根据控制器输出信号的不同来选择是否使用内部加减速曲线。

使用内部加速曲线：

当电机加速度小于 60000 时，驱动器会使能内部加减速曲线，具体加速度的大小就和设置的值相同。

使用场合：使用内部加速曲线，会产生滞后脉冲的现象，一些不需要实时跟随的场合，可以使用内部加速曲线。有些控制器，脉冲直接给到对应速度的频率，没有加减速的情况，就使用内部加减速曲线，可以降低控制器编程难度。

禁止内部加速曲线：

当电机加速度大于等于 60000 时，驱动器根据外部脉冲的加减速允许，内部加速度无效。

使用场合：例如雕刻机，控制器输出的脉冲就是有加减速的，就不需要驱动器内部的加速曲线，如果这个时候使用，会滞后于实际的脉冲。

2. 丝杆负载

首先介绍下扭矩，先用 400W 电机，1.3NM。负载是 5mm 螺距的丝杆，就是电机轴转一圈负载移动 5mm，这样的话，

负载等效力臂 = 5mm / 3.14 = 1.592 mm

那电机能提供的推力就是

经过丝杆传动的推力 = 1.3NM / （1.592mm*0.001）= 816 N

那能推动负载的重量就大约是 80KG，这个是垂直的，平推可以稍微大些。

由于丝杆负载电机转动一圈移动的距离较短，所以驱动器的参数 （加速度可以较大，如 20000，位置环 KP 可以较大，如 3000）。伺服电机最适合此种负载。

3. 皮带轮负载

伺服电机其实不是很适合接这种负载。因为皮带轮一般直径比较大，例如直径 30mm。那电机转一圈，负载移动的距离就是 30mm*π = 94.2，比上面说的丝杆 5mm 大了很多倍。

那电机能提供的推力就是

经过皮带传动的推力 = 1.3NM / （30mm*0.001）= 43.3 N

那能推动负载的重量大约是 4.3KG. 所以伺服电机其实不适合接同步轮，因为同步轮转动一圈负载移动的距离太长，力臂长。如果这种场合要用伺服电机，可以选择直接尽量小的同步轮或通过电机轴接小同步轮，负载端接大同步轮，这样减速几倍，可以达到较好的效果。这种场合驱动器参数(加速度设置较小，如 5000，)，这样设置参数的目的是减小加速度和减速度，因为负载等效惯性大。

4. 圆盘负载

这种负载伺服无法直接带动，一般都需要接减速器。例如直径 200mm 重量 10KG 的圆盘。半径就是 100mm，重量等效半径就是 50mm。力臂很大。如果伺服要接此类负载，比较接减速器再接负载。

如果圆盘不是特别重，可以牺牲一些定位精度和刚性来控制。具体方法，电机加速度设置到比较小，例如 1000 左右。 速度 KI 设置到 2000，取消积分作用。位置 KP 改到 1000. 改这些参数一般的圆盘负载也能用。

5. 自动找单圈原点功能

自动找原点功能通过改 寄存器地址 0x19（特殊功能）的参数来选择。如果需要上电自动找原点，设置方法如下：

modbus 使能 发送 1

特殊功能（地址 0x19）发送 10~32768（32768 对应电机的 360°）

参数保存发送 1

重新上电后就会自动找原点。由于是绝对值编码器，上电后可以自动找一圈中的任意位置。（DIR 极性是 1 或者 0 可以设置找原点的方向）

6. 自动找机械原点功能

自动找机械原点功能通过改 寄存器地址 0x19（特殊功能）的参数来选择。如果需要上电自动找机械原点，设置方法如下：

modbus 使能 发送 1

特殊功能（地址 0x19）发送 1 （此时立刻会自动找机械原点）

参数保存发送 1 （需要重新上电自动找机械原点可以通过保存此参数实现）

重新上电后就会自动反转到电机堵转，然后电机反转 36°作为原点。（DIR 极性是 1 或者 0 可以设置找原点的方向）

7. 自动找原点功能（原点开关）

自动找机械原点功能通过改 寄存器地址 0x19（特殊功能）的参数来选择。如果需要上电自动找原点，设置方法如下：

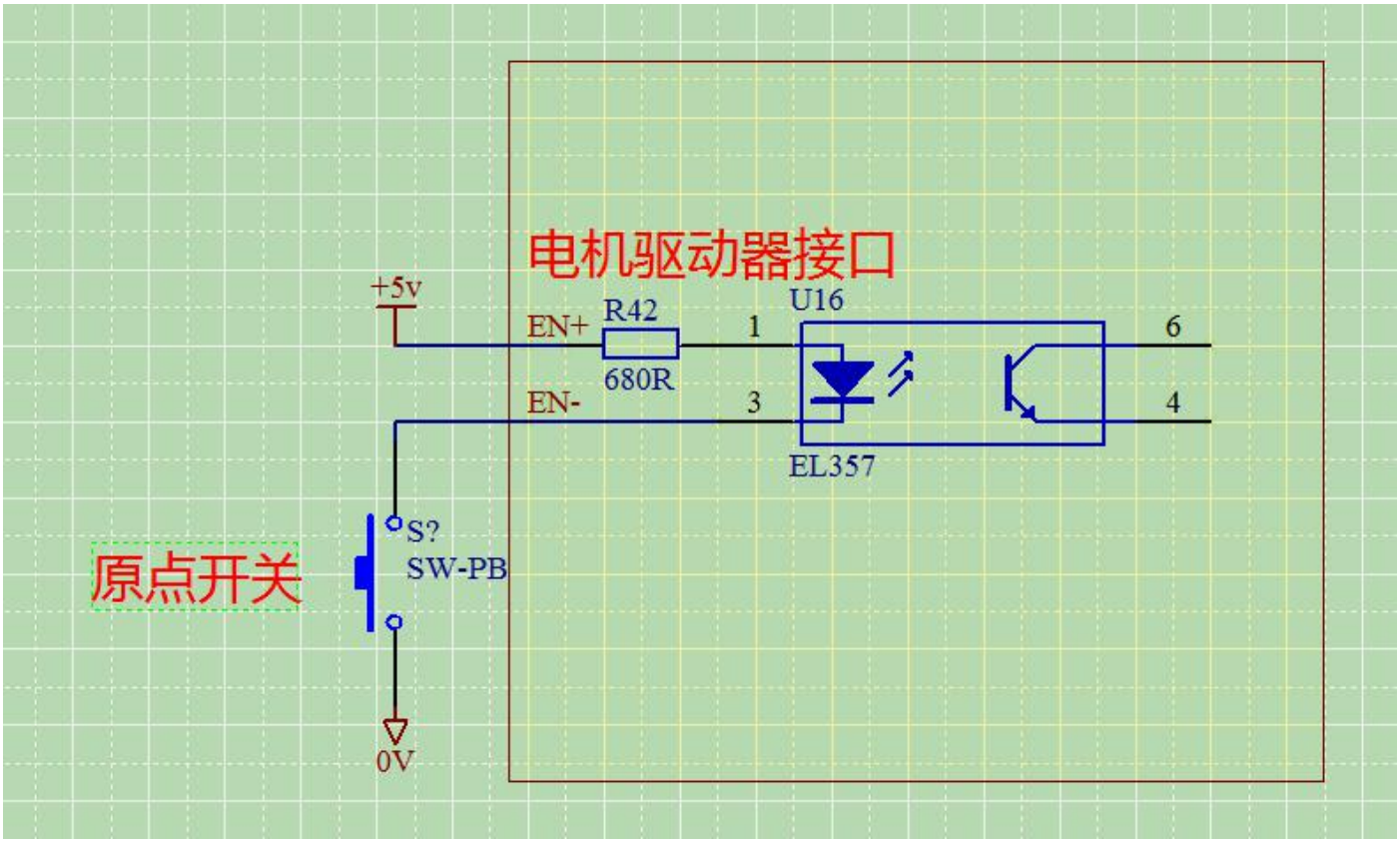
modbus 使能 发送 1

特殊功能（地址 0x19）发送 8 （此时立刻会自动找机械原点）

参数保存发送 1 （需要重新上电自动找机械原点可以通过保存此参数实现）

重新上电后就会自动反转到有 EN 信号，然后电机正转编码器原点作为原点。（DIR 极性是 1 或者 0 可以设置找原点的方向）

原点开关需要接到 EN 信号。接线图如下：



8. 通信方式清除位置

清除绝对位置：如果在运行过程中需要将绝对位置清 0，先电子齿轮分子发送 0（通信模式下电子齿轮无效，用于此特殊功能。如果通信控制可以直接电子齿轮分子保存成 0），然后绝对位置（0x16）发送 0，就直接给绝对位置清 0.

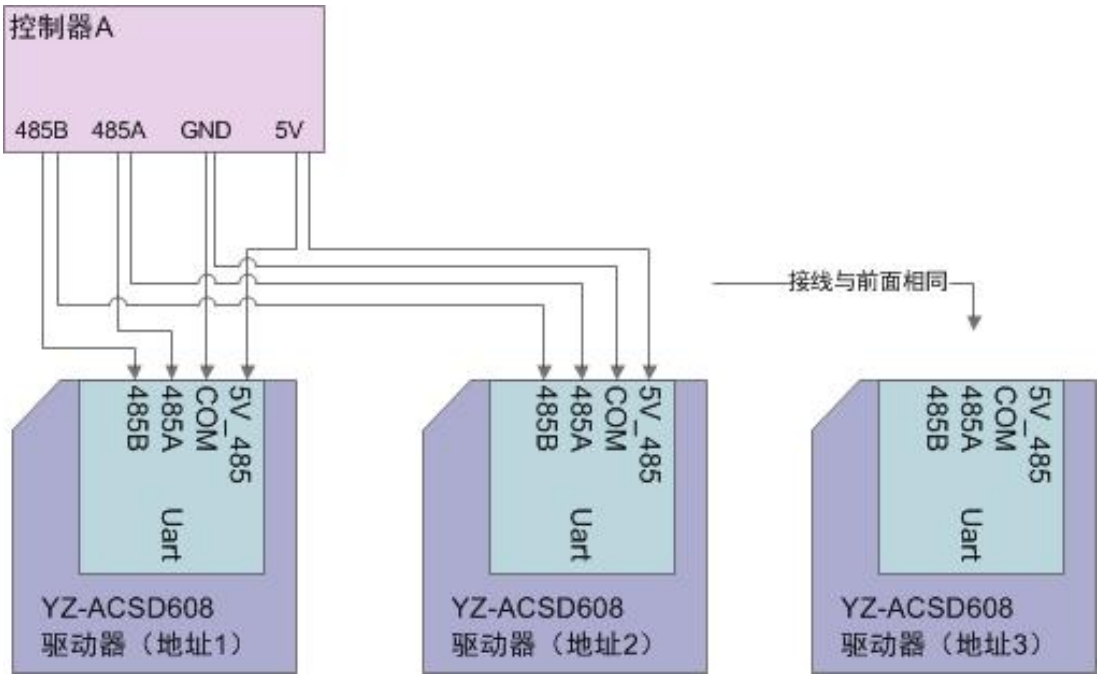
急停：在通信模式下，如果剩余了很多脉冲需要走，需要急停的情况。先电子齿轮分子发送 0（通信模式下电子齿轮无效，用于此特殊功能。如果通信控制可以直接电子齿轮分子保存成 0，再目标位置（0x0C）发送 0，就可以急停。急停也有少量减速距离，减速距离长短通过位置环 KP 控制。

9. 上电默认通信控制

只要设置电子齿轮分子为 0，保存以后，重新上电，modbus 使能 默认 是 1.

五． Modbus 控制方式

1. 硬件连接



驱动器内部 485 都通过光耦隔离，解决了一台主机连接多台从机容易被干扰和损坏的问题。

2. 寄存器说明

驱动器可以通过 modbus（RTU 模式）来控制驱动器。主机可以通过 modbus 的读写寄存器功能来设置驱动器参数和控制运行。驱动器支持的功能码为 0x3（读寄存器）、0x6（写寄存器）、0x78（写目标位置）、0x7a（修改设备地址）。

寄存器列表如下：

地址	参数名称	只读/读写	参数范围	参数说明
0x00	Modbus 使能	读写	0~1	0：modbus 禁止

				1：modbus 使能
0x01	驱动器输出使能	读写	0~1	0：驱动器输出禁止 1：驱动器输出使能
0x02	电机目标速度	读写	0~3000 r/min	速度模式时，目标速度 位置模式时，最大速度
0x03	电机加速度	读写	0~60098 (r/min)/s	参数小于 60000 时，驱动器内部产生加减速曲线，参数等于 60000 时，没有加速过程，减速大小根据位置 KP 决定. 大玉 60000 到 60098，个位和十位 0~98 对应位置前馈 0%~98%。位置前馈越大，跟随脉冲滞后越小。
0x04	弱磁角度	读写	0~306 r/min	内部参数不需要另外设置
0x05	速度环比例系数	读写	0~10000	代表 0.0~10.0 数值越大刚性越强 个位为偶数：脉冲输入极性为断开时刻有效 个位为奇数：脉冲输入极性为导通时刻有效
0x06	速度环积分时间	读写	2~2000 ms	积分时间 2~2000ms 数值越小刚性越强
0x07	位置环比例系数	读写	60~30000	位置 KP，数值越大，刚性越强 个位为偶数：报警输出常开（正常为常开，报警常闭） 个位为奇数：报警输出常闭（正常为常闭，报警常开）
0x08	速度前馈	读写	0~12.0V/KRP M	327 代表 1V/KRPM，不需要自行设置
0x09	DIR 极性	读写	0~1	0：外部 DIR 不导通顺时针旋转 1：外部 DIR 导通顺时针旋转
0x0A	电子齿轮分子	读写	0~65535	16 位电子齿轮分子 如果电子齿轮分子为 0，可以实现特殊功能具体看前文介绍
0x0B	电子齿轮分母	读写	1~65535	16 位电子齿轮分母
0x0C	目标位置低 16 位	只读		需要走步数的高 16 位
0x0D	目标位置高 16 位	只读		需要走步数的低 16 位
0x0E	报警代码	只读		
0x0F	系统电流	只读	0~32767	实际电流为 x/2000(A)
0x10	电机当前速度	只读	-30000~30000 r/min	实际电机转速=电机当前速度/10
0x11	系统电压	只读	0~32767	实际电压为 x/327（V）
0x12	系统温度	只读	0~100	摄氏度
0x13	系统输出的 PWM	只读	-32768~32767	代表-100%~100%
0x14	参数保存标志	读写	0~1	0：参数未保存 1：保存参数中 2：保存完毕
0x15	设备地址	只读	0~255	设备地址
0x16	绝对位置低 16 位	读写		走过步数的高 16 位
0x17	绝对位置高 16 位	读写		走过步数的低 16 位
0x18	静止最大允许输出	读写	0~609	0~609 对应允许最大输出 0~60.9% 个位 1~9 对应堵转报警时间。个位 0 堵转不报警,堵转 3S 后自动输出降低为原来的 1/2。
0x19	特殊功能	读写	0~100	0：脉冲+方向模式

				1: 自动找机械原点并正转 36°（上电自动反转到机械零点，并正向走 36°停下） 2：编码器跟随模式 3: 速度模式，占空比调速（10%~90%对应 0~1000RPM） 10~32768：上电自动转到的角度， 算法为： X*360°/32768 4 :自动找机械原点并正转到编码器零点（上电自动反转到机械零点，并正向走到编码器零点停下） 5: Z_OUT 模拟原点信号模式 :上电给电机脉冲控制电机慢速反转直至堵转，Zout 输出导通信号，然后控制器给电机脉冲正转到 ZOUT 信号消失即为系统原点（Zout 在出现一次编码器零点后消失，这样可以保证原点的准确） 8: 自动找原点（原点接入到 EN 信号）。上电自动反转到 EN 有信号，再正转到 EN 无信号停止。

3. Modbus 通信格式

a．modbus 主机读取数据及从机应答格式 （功能码 03)

主机读取数据 格式							
设备地址	功能码	第一个寄存器的高位地址	第一个寄存器的低位地址	寄存器个数高位	寄存器个数低位	CRC 高位	CRC 低位
0x01	0x03	0x00	0x00	0x00	0x01	0x84	0x0a

从机应答						
设备地址	功能码	数据长度	第一个数据高字节	第一个数据低字节	CRC 高位	CRC 低位
0x01	0x03	0x02	0x00	0x01	0x79	0x84

串口接收到的数据都是无符号数，如果寄存器是有符号数，发送的则是二进制补码的格式，转换成有符号数的算法如下（VB 代码）：

```
If modbus.data(11) > 32767 Then
    disp_modbus_data.PU = (modbus.data(11) - 32768) * 65536 + modbus.data(10)
    disp_modbus_data.PU = -((&H7FFFFFFF - disp_modbus_data.PU) + 1)
Else
    disp_modbus_data.PU = dmodbus.data(11) * 65536 + modbus.data(10)
End If
```

注: modbus.data(11)为目标位置高 16 位 modbus.data(10)为目标位置低 16 位

b．modbus 主机写数据及从机应答格式 (功能码 06)

主机写数据 格式							
设备地址	功能码	第一个寄存器的高位地址	第一个寄存器的低位地址	数据高位	数据低位	CRC 高位	CRC 低位
0x01	0x06	0x00	0x00	0x00	0x01	0x48	0x0a

从机应答 格式

设备地址	功能码	第一个寄存器的高位地址	第一个寄存器的低位地址	数据高位	数据低位	CRC 高位	CRC 低位
0x01	0x06	0x00	0x00	0x00	0x01	0x48	0x0a

c．modbus 主机写脉冲数 （功能码 0x10）

主机写双字节数据 （写 PU 脉冲数）						
设备地址	功能码	第一个寄存器的高位地址	第一个寄存器的低位地址	寄存器个数高位	寄存器个数低位	数据长度
0x01	0x10	0x00	0x0c	0x00	0x02	0x04
PU:8~15 位	PU:0~7 位	PU:24~31 位	PU:16~23 位	CRC 高位	CRC 低位	
0x27	0x10	0x00	0x00	0xf8	0x8b	

脉冲数是有符号数，一个负数(假设此数为 X)转换成 32 位 16 进制数的算法如下(vb 代码)：

```
If X < 0 Then
  X = &H7FFFFFFF + (X + 1)
  PU24_31 = Fix(X / (256 * 65536)) + &H80
Else
  PU24_31 = Fix(X / (256 * 65536))
End If
  PU16_23 = Fix(X / 65536) mod 256
  PU8_15 = Fix(X / 256) mod 256
  PU0_7 = X mod 256
```

注：fix() 为取整函数

从机应答 格式							
设备地址	功能码	第一个寄存器的高位地址	第一个寄存器的低位地址	寄存器个数高位	寄存器个数低位	CRC 高位	CRC 低位
0x01	0x10	0x00	0x0c	0x00	0x02	0x81	0xcb

d．modbus 主机写增量脉冲数 （特殊功能码 0x78）

主机特殊功能码 0x78 格式 （写 PU 脉冲数）							
设备地址	功能码	PU:24~31 位	PU:16~23 位	PU:8~15 位	PU:0~7 位	CRC 高位	CRC 低位
0x01	0x78	0x00	0x00	0x27	0x10	0xbb	0xfc

从机应答 格式							
设备地址	功能码	PU:8~15 位	PU:0~7 位	PU:24~31 位	PU:16~23 位	CRC 高位	CRC 低位
0x01	0x78	0x27	0x0e	0x00	0x00	0xca	0xb7

e．modbus 主机写绝对位置 （特殊功能码 0x7b）

主机特殊功能码 0x78 格式 （写 PU 脉冲数）							
设备地址	功能码	PU:24~31 位	PU:16~23 位	PU:8~15 位	PU:0~7 位	CRC 高位	CRC 低位
0x01	0x7b	0x00	0x00	0x27	0x10	0xff	0xfc

从机应答　格式							
设备地址	功能码	PU:8~15 位	PU:0~7 位	PU:24~31 位	PU:16~23 位	CRC 高位	CRC 低位
0x01	0x7b	0x27	0x10	0x00	0x00	0xee	0xb1

4. Modbus 同步控制多台电机

通过 modbus 同步控制多台伺服电机，最多 100 台。通讯格式如下：

Modbus-rtu 通讯写指令

主机（电脑或者其他主机）写入：

指令格式									
1BYTE	1 BYTE	2BYTE		2BYTE		1BYTE	寄存器个数*8 BYTE	2BYRE	
设备地址	功能码	寄存器地址		寄存器个数		数据长度	数据内容	CRC	
0x00	0x10	0x00	0x16	0x00	0x10	0xXX	见下表	HI	LO

设备地址：默认是 0，这样发给所有地址电机都能收到。

功能码：0x10 兼容标准功能码，支持尽量多的标准设备

寄存器地址：默认 0x16，此地址为绝对位置的地址。

寄存器个数：如果控制 1 台电机个位 0x4，如果 2 台 0x8，以此类推。

数据长度：可以发任意数。

数据内容													
4byte				2byte		2byte		4byte		2byte		2byte	……
1 轴位置				1 轴速度		1 轴加速度		2 轴位置		2 轴速度		2 轴加速度	……
32 位数				16 位数		16 位数		32 位数		16 位数		16 位数	……
[15~8]	[7~0]	[31~24]	[23~16]	[15~8]	[7~0]	[15~8]	[7~0]						

数据内容：每个电机会根据自己地址不同来截取自己对应的数据，如果是 1 号地址的电机，就是前面 8 个字节，内容如上表所示。

1 轴位置：地址 1 电机的绝对位置。

1 轴速度：地址 1 电机的目标速度。

1 轴加速度：地址 1 电机的加速度。

5. CRC 校验示例代码

```
unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg ; /* 要进行 CRC 校验的消息 */
unsigned short usDataLen ; /* 消息中字节数 */
{
unsigned char uchCRCHi = 0xFF ; /* 高 CRC 字节初始化 */
unsigned char uchCRCLo = 0xFF ; /* 低 CRC 字节初始化 */
unsigned uIndex ; /* CRC 循环中的索引 */
while (usDataLen--) /* 传输消息缓冲区 */
```

```
{
uIndex = uchCRCHi ^ *puchMsgg++ ; /* 计算 CRC */
uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex} ;
uchCRCLo = auchCRCLo[uIndex] ;
}
return (uchCRCHi << 8 | uchCRCLo) ;
}
/* CRC 高位字节值表 */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;
/* CRC 低位字节值表*/
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA,
0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15,
0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34,
0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9,
0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76,
0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D,
0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D,
0x4C, 0x8C, 0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 }
```

6. modbus 方式主机控制过程

a：位置模式

通过拨码开关 SW1 打 OFF 再上电即为位置模式。

先上电可以通过我们提供上位机软件设置如下参数：

1. Modbus 使能 发送 1（只有 modbus 使能为 1 才能改其他参数，且外部脉冲信号无效。）

   HEX 源码命令 ： 01 06 00 00 00 01 48 0A

2. 电机加速度 发送 5000 (根据实际需要设置加速度，不设置即使用默认参数 20000)

   HEX 源码命令 ： 01 06 00 03 13 88 74 9C

3. 目标转速 发送 1500 （根据实际运行需要设置运行的速度，不设置即使用默认参数 2800）

   HEX 源码命令 ： 01 06 00 02 05 DC 2A C3

4. 电子齿轮分子 发送 0 （电子齿轮分子保存为 0 后，下次上电 mdobus 使能默认是 1）

   HEX 源码命令 ：01 06 00 0A 00 00 A9 C8

5. 参数保存标志 发送 1 （发此参数后，前面设置的参数保存到内部）

   HEX 源码命令 ：01 06 00 14 00 01 08 0E

6. 重新上电，看参数是否已经正确保存。以上设置只需要用提供的上位机设置即可，HEX 源码不需要自己通 过串口发送。

参数设置完以后，就可以通过 PLC 或者单片机，或者自己设计的上位机软件发位置命令。发位置命令只需要过 0x10 命令发送需要走的位置就行。

1. 发增量位置（增量位置的含义是，发送的数据即为电机需要向前或者向后走的位置）

   例如需要向前走一圈（假设电机编码器为 1000 线编码器，一圈脉冲数即为 4000）

   HEX 源码命令：01 10 00 0C 00 02 04 0F A0 00 00 F0 CC

   例如需要向前后一圈（假设电机编码器为 1000 线编码器，一圈脉冲数即为-4000）-4000 的二进制计算方法如下：4000 的二进制为 00 00 0F A0 。(注：0= FF FF FF FF +1)

   -4000 即为 0 - 00 00 0F A0 =FF FF FF FF - 00 00 0F A0 +1=FF FF F0 5F +1 = FF FF F0 60

   HEX 源码命令：01 10 00 0C 00 02 04 F0 60 FF FF C1 54

2. 发绝对位置（绝对位置的含义是，刚刚上电或者绝对位置清 0 或者自动找原点后的时候定义位置为 0，绝对位置就是走到新发的位置，如第一次发 4000 为走一圈，第二次发已经走到了 4000 的位置，再发相同命令电机不走）

例如需要电机走到 2 圈位置（假设电机编码器为 1000 线编码器，2 圈脉冲数即为 8000）

HEX 源码命令：01 10 00 16 00 02 04 1F 40 00 00 74 89

例如需要电机走回原点（当电子齿轮分子为 0 的时候，发送 0 为清除当前位置，所以走回原点发送 1，此时一个脉冲并不会影响精度）

HEX 源码命令：01 10 00 16 00 02 04 00 01 00 00 23 49

注：控制电机只需要先发送需要的位置（尽量用绝对位置指令，因为可以重复发多次，依然是走到相同位置），然后可以通过读取绝对位置对比是否走到设置位置，来判断是否执行下一条指令（注意判断的时候需要允许+-2 的误差）。或者可以通过接 PF 信号，走到位后，驱动器会给出一个光耦输出的开关量信号。

读取绝对位置指令如下：01 03 00 16 00 02 25 CF

7. 修改波特率

修改波特率可以通过我们提供的上位机软件发送。具体要按如下步骤发送：

A. Modbus 使能（地址 0） 发送 1

B. 电机加速度(地址 3) 发送 803 （备注 803:115200 802:38400 801:19200 800:9600）

C. 弱磁角度（地址 4） 发送 129

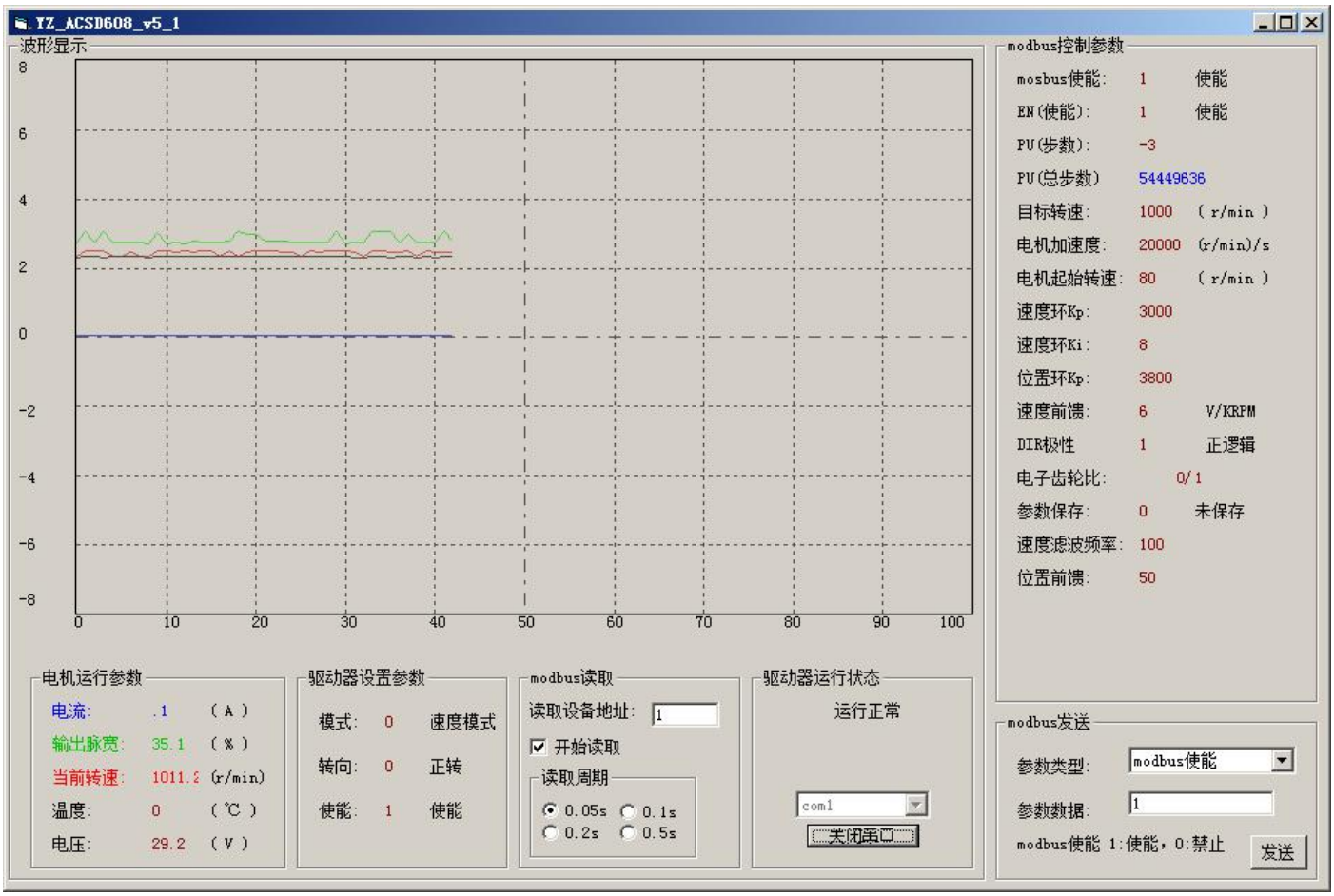
D. Modbus 使能（地址 0） 发送 506

重新上电后生效

注意： 不需要发参数保存，因为这个是改内部参数。只需要严格按上面步骤发送。

六. 上位机软件使用说明

本驱动器提供一个上位机软件，用于监测和测试驱动器。可以通过软件查看和设置驱动器内部参数。



如上图所示，软件分为波形显示，电机运行参数等几个部分。下面介绍一下各个部分的功能和作用。

波形显示：一共有 4 个通道，分别用 4 种颜色表示。颜色和 电机运行参数 内的字体颜、色相同。即：蓝表示电流，绿表示输出的脉宽，红表示当前转速，黑表示电压。

电机运行参数：表示电机运行的实时数据。

驱动器设置参数：显示驱动器的拨码开关，和方向使能设置。如果是 modbus 模式，此栏无效。

驱动器运行状态：此栏会显示驱动器的报警状态，如果没有报警会显示运行正常。

Modbus 控制参数：此栏内的参数是驱动器内部的参数，如果要修改这些参数，必须先对 modbus 使能写 1。具体的参数含义参考 寄存器说明。

Modbus 读取：此栏可设定驱动器的地址，读取驱动器数据的周期，和是否读取。

Modbus 发送：此栏用于修改驱动器参数，首先选定参数类型，再设定好参数数据，然后点发送即可。