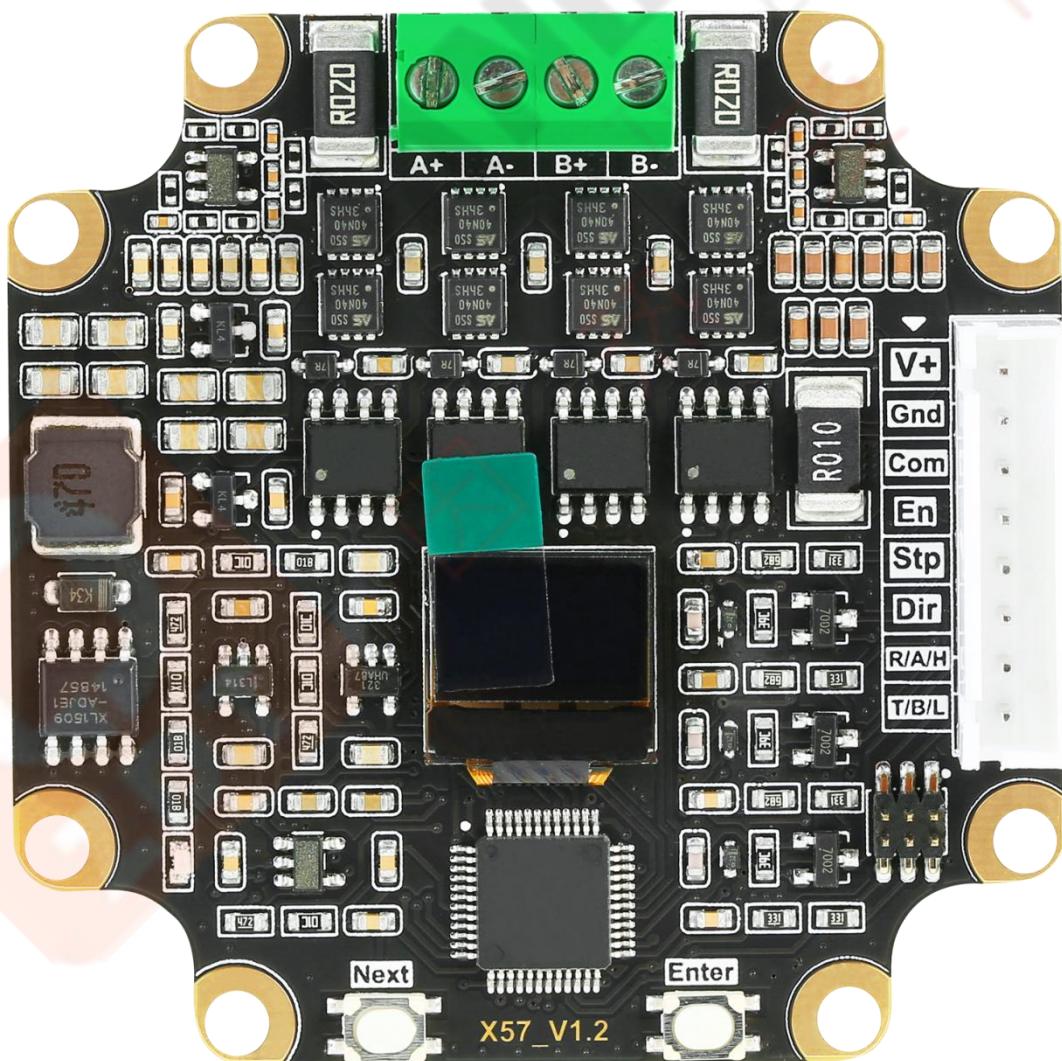




ZHANGDATOU
ZHANG DA TOU ELECTRONICS
因 热 爱 而 生

ZDT_X57_V2 步进闭环驱动说明书 Rev1.0 (X28/X35/X42/X57 通用)



目录

一、产品介绍.....	4
1.1 产品简介.....	4
1.3 产品特点.....	5
1.4 产品参数.....	6
1.5 产品尺寸.....	7
二、闭环 PCBA 安装.....	8
2.1 硬件清单.....	8
2.2 安装步骤.....	9
三、首次上电校准编码器.....	10
四、按键操作和 OLED 显示说明.....	12
五、OLED 菜单功能说明.....	13
六、脉冲控制.....	24
6.1 脉冲控制接线.....	24
6.1.1 3D 打印主板接线.....	24
6.1.2 STM32 脉冲控制接线.....	24
6.1.3 Arduino 脉冲控制接线.....	25
6.1.4 51 单片机脉冲控制接线.....	25
6.1.5 PLC 脉冲控制接线.....	26
6.1.6 EN+/EN-/PUL+/PUL-/DIR+/DIR-差分脉冲控制接线.....	26
6.2 脉冲控制说明.....	27

七、通讯控制.....	28
7.1 通讯控制接线.....	28
7.1.1 串口 TTL 通讯控制接线.....	28
7.1.2 RS232 通讯控制接线.....	29
7.1.3 RS485 通讯控制接线.....	30
7.1.4 CAN 通讯控制接线.....	31
7.2 通讯控制命令格式说明.....	32
7.2.1 串口 TTL/RS232/RS485 通讯控制命令格式说明.....	32
7.2.2 CAN 通讯控制命令格式说明.....	33
7.3 多机通讯及同步控制.....	35
7.4 通讯控制命令列表.....	36
7.4.1 控制动作命令列表.....	36
7.4.2 原点回零命令列表.....	39
7.4.3 触发动作命令列表.....	43
7.4.4 读取参数命令列表.....	44
7.4.5 修改参数命令列表.....	50
八、原点回零操作说明.....	52
8.1 单圈就近/方向回零操作说明.....	52
8.2 多圈无限位碰撞回零操作说明.....	52
8.3 多圈有限位开关回零操作说明.....	54
九、到位输出和报警输出.....	55

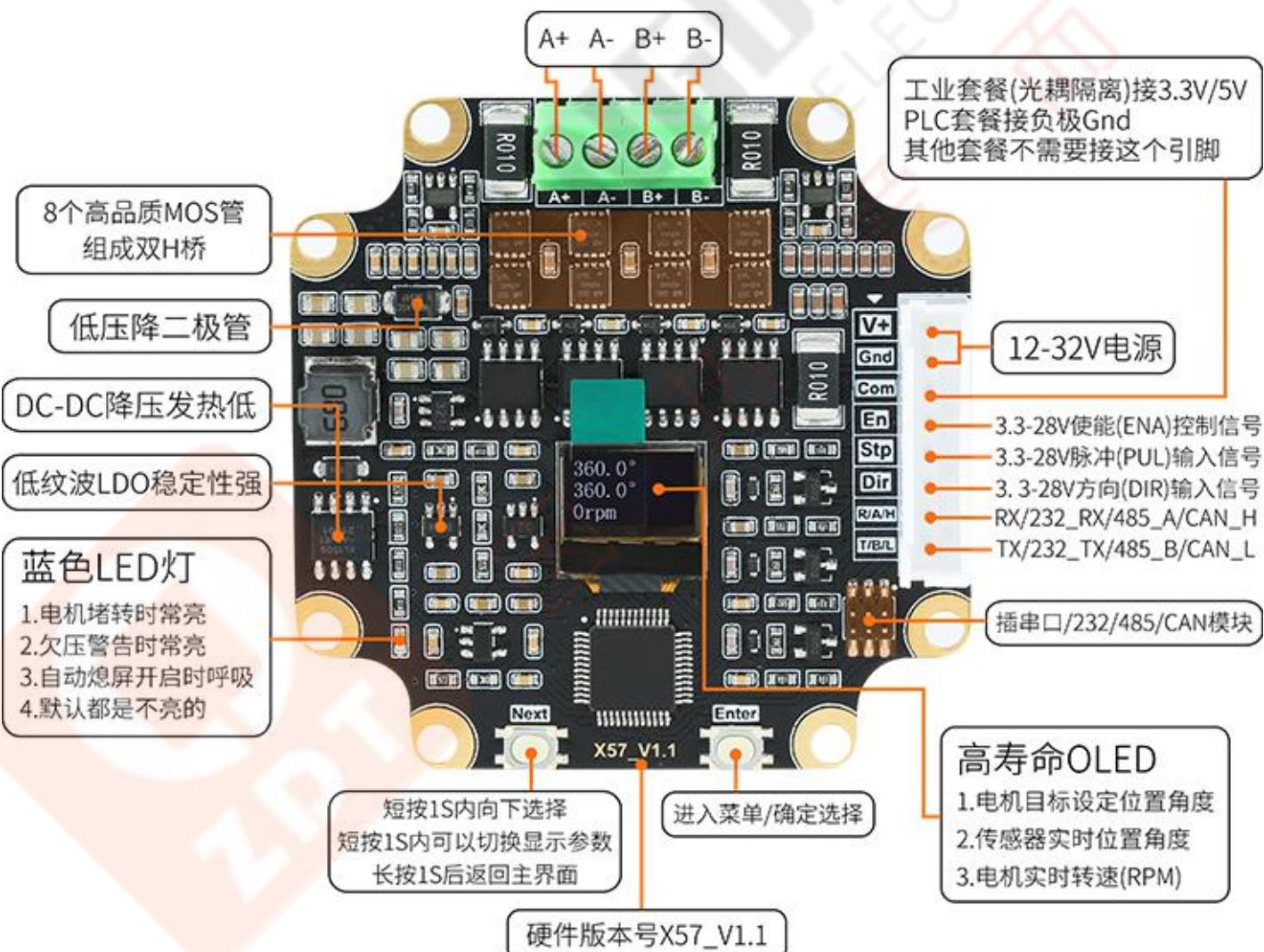
9.1 通讯控制到位输出/到位返回命令.....	55
9.2 脉冲控制模式报警输出和复位堵转保护.....	55
十、常见问题及注意事项.....	56
10.1 常见问题.....	56
10.2 注意事项.....	56
十一、技术支持及保证.....	57
十二、修改记录.....	58
十三、附录.....	59

一、产品介绍

1.1 产品简介

ZDT_X57_V2.0 是张大头闭环伺服，在 ZDT_X57_V1.2 的基础上，总结了广大用户的反馈信息，历经一年多的设计和改进升级而来，前后修改的软硬件版本多达几十上百次，经过长时间不断的测试和完善，性能稳定，可靠性高，欢迎大家使用。

1.2 硬件介绍



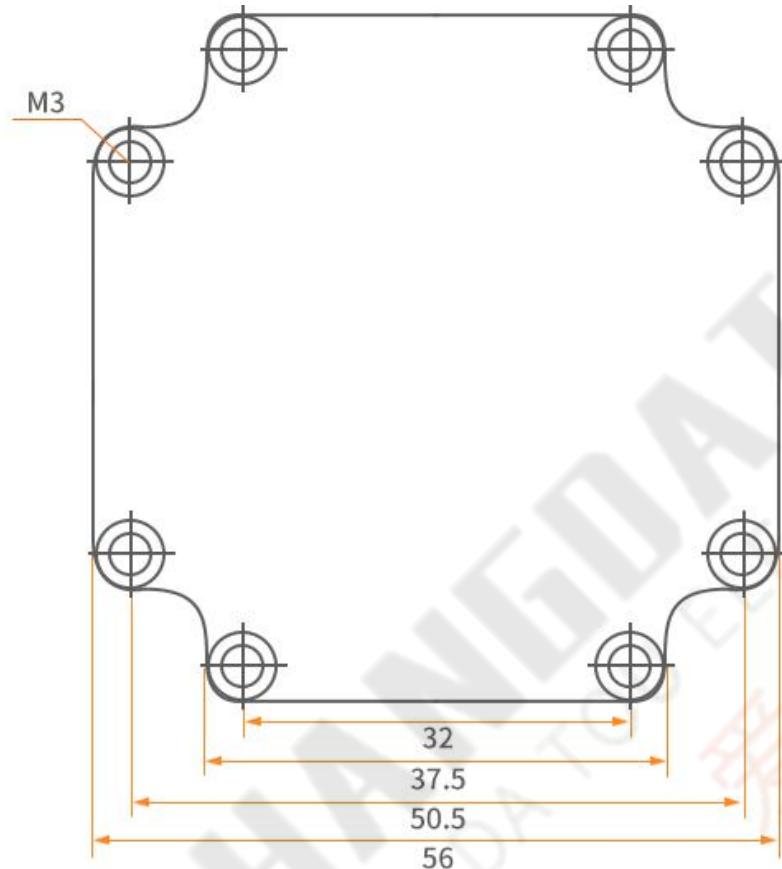
1.3 产品特点

- ZDT_X57_V2.0 板载工业级高精度 16384 线磁编码器、板载精密电流传感器；
- 采用优异的基于反馈编码器的 FOC 矢量闭环算法，力矩、速度、位置三闭环控制；
- 一键自动识别电机的相电阻和相电感等参数，并自动计算电流环 PI 参数；
- 最大电流 4000mA，转速范围 0.1RPM – 3000.0RPM+（电机参数不同最高转速不同）；
- 支持脉冲控制、支持串口/RS232/RS485/CAN 通讯控制，支持二者混合控制；
- 支格力矩模式、速度模式、位置模式控制，支持直通限速位置模式和梯形曲线加减速位置模式，每个位置指令都支持绝对位置和相对位置运动，所有的控制模式以及两条命令之间都可以连续无缝平滑过渡；
- 支持单圈就近回零、单圈方向回零、多圈无限位碰撞回零和多圈有限位开关回零，支持存储回零一组参数，上电自动触发回零功能；
- 支持自定义协议（支持 0x6B、XOR、CRC-8 校验）、支持 Modbus-RTU 协议；
- 支持多机通讯和多机同步控制，支持最多 255 个地址的电机进行通讯同步控制；
- 支持通讯控制到位输出/到位返回命令、支持脉冲控制堵转输出和复位堵转保护；
- 支持存储一组速度参数（方向、速度和加速度）上电自动运行，通过 En 脚控制启停；
- 具有堵转保护功能，支持发送命令或 I0 控制解除堵转保护，不需要重新上电；
- 可支持共阳/共阴信号输入，可支持 PLC 的 24V 信号输入，可选光耦隔离输入；
- 支持 1–256 任意细分，支持细分插补功能，电机运行更加平滑和静音；
- 板载小屏幕实时显示系统信息，板载按键一键修改参数，自动保存，立即生效；
- 电机转速和位置实时更新（使能不使能都可以），可发送命令清零位置；
- 具有编码器自校准、线序错误识别、自动熄屏、一键恢复出厂设置等众多功能；

1.4 产品参数

主板型号	ZDT_X57_V2.0
发布时间	2024 年 1 月
主控芯片	高性能 32 位 ARM 处理器
驱动电路	8 个高品质 MOS 管组成双 H 桥驱动
传感器	工业级高精度 16384 线磁编码器、精密电流传感器
供电电压	12-32V
工作电流	0-3000mA (X28/X35/X42)、0-4000mA (X57) 任意电流
信号输入	3.3V-28V 支持共阳/共阴输入、支持 PLC 的 NPN/PNP 的 24V 信号输入
闭环反馈频率	力矩环 20KHz+、速度环 20KHz+、位置环 20KHz+
最大脉冲频率	160KHz 以上
转速范围	0.1RPM - 3000.0RPM+ (每个电机参数不一样最高转速可能不同)
控制精度	小于 0.08°
细分支持	1-256 任意细分，支持细分插补功能
控制方式	脉冲/串口 TTL/RS232/RS485/CAN，支持多机通讯和多机同步控制
通讯控制模式	力矩模式、速度模式、位置模式（直通限速和梯形曲线位置模式）
保护措施	堵转保护、欠压警告、电流限制、转速限制等
静音/震动	低速超静音、超低震动，高速运行平稳，不抖动
制动/换向	支持高速 3000.0RPM+立即停止，立即切换方向
回零模式	单圈就近/方向回零、多圈无限位/有限位开关回零（3D 打印不支持）
输出类型支持	支持通讯控制到位输出/到位返回命令、支持脉冲控制报警输出

1.5 产品尺寸



单位: mm



二、闭环 PCBA 安装

2.1 硬件清单

序号	品名	数量
1	57 步进电机	1
2	ZDT_X57_V1.2 闭环 PCBA	1
3	M3*7mm 螺丝	4
4	5*3mm 径向磁铁	1
5	金属垫片	4
6	3M 胶水/超能胶水	1
8	螺丝刀	1
9	1 米通讯线	1

硬件清单如下图所示：



注意：

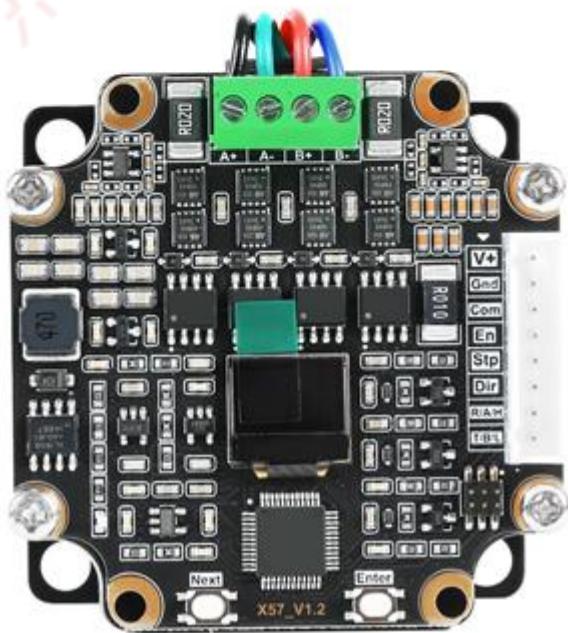
1. 胶水尽量选择 3M 胶水、超能胶水、AB 胶等有粘稠状、对金属和磁铁都有较强粘性的胶水，尽量不选择 502 等低粘稠状的胶水，此类胶水比较容易渗透到电机内部，造成电机的损坏。
2. 建议选用本店的 28/35/4257 磁柱步进电机，其他电机的电机参数不同，可能不适用。

2.2 安装步骤



1.用胶水(推荐3M胶、超能胶), 在电机背部轴上滴一小滴胶水即可
注意不能太多，否则会损坏电机。

2.磁铁尽量粘在轴上居中，不能偏太多，也不要不水平、翘一边等
模块上电会自动检测位置，保证高精准。

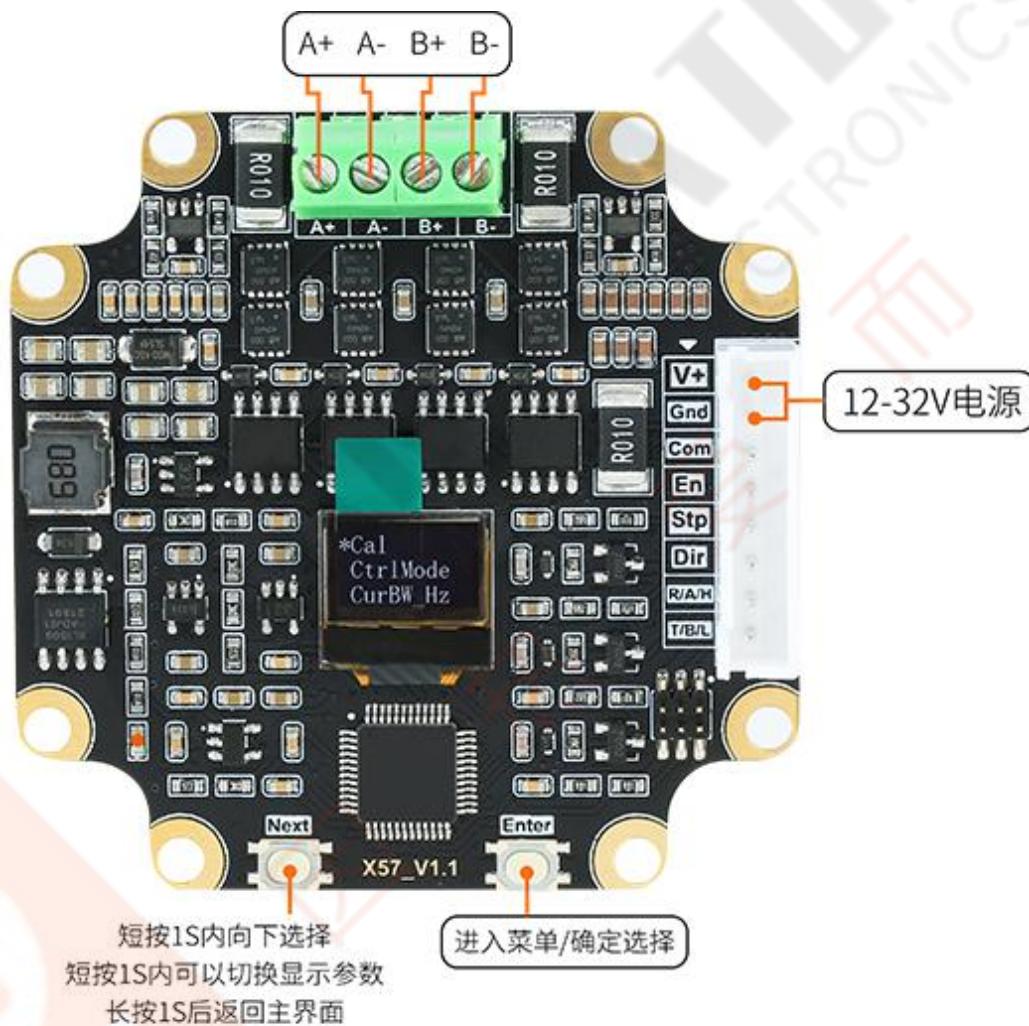


3.粘好磁铁后，放上金属垫片。

4.放好57闭环，拧上螺丝，即可使用。

三、首次上电校准编码器

按照“2.2 安装步骤”章节完成闭环 PCBA 安装到电机后，接入 12–32V 电源到闭环电机的 V+和 Gnd 进行供电，其他引脚可接可先不接。通电后，会弹出“Not Cal”两秒后进入菜单界面，如下图所示：



此时，按一下 Enter 键确定选择 Cal 进行编码器校准，闭环电机在“哔”的一声后会慢慢的正转一圈，然后反转一圈，最后高速运行几秒（高速运行仅在首次上电校准会执行，后续校准不会再执行）。运行结束后，编码器校准完成，闭环电机就可以正常控制使用了。如果出现了其他异常情况，请参考下面方法进行处理：

(1) 提示 “Phase A+ A- Error! ” 或 “Phase B+ B- Error! ” 或 “Phase AA BB Error! ”

- 错误类型：电机线序错误；
- 错误原因：A+A-或B+B-的电机线序不对；
- 操作步骤：57步进的电机线序一般为黑色和绿色是一组，红色和蓝色是一组，分别接到A+A-和B+B-，如果颜色分不出来，可以用万用表的蜂鸣档测一下电机线的任意两根，导通发出声响的两根线为一组，另外两根为一组。

(2) 提示 “Waiting V+ Power”

- 错误类型：电源供电电压太低；
- 错误原因：电源电压V+和Gnd的电压不足；
- 操作步骤：请提供12-32V的电压给驱动器进行供电，推荐24V供电，动力足；

(3) 提示 “Magnet Loss! Enter..”

- 错误类型：驱动器上的编码器没检测到磁铁；
- 错误原因：没粘磁铁到电机轴上，没把驱动板固定到电机背后；
- 操作步骤：按照2.2章节安装步骤进行磁铁的安装和驱动板的固定；

(4) 提示 “Ref Voltage Error!”

- 错误类型：基准电压芯片故障；
- 错误原因：基准电压芯片故障；
- 操作步骤：请联系客服进行处理；

四、按键操作和OLED显示说明

参数显示界面



按Enter键进入菜单设置界面

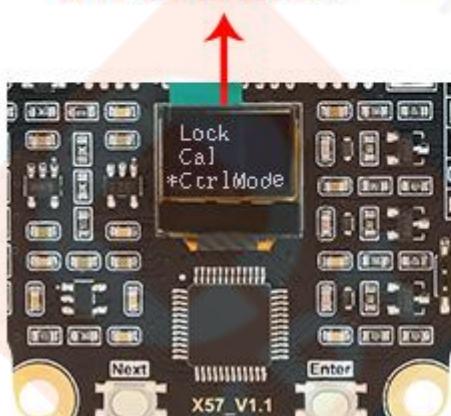
1. 长按Next键退出到参数显示界面
2. 选中Exit菜单，按Enter键退出菜单

菜单设置界面

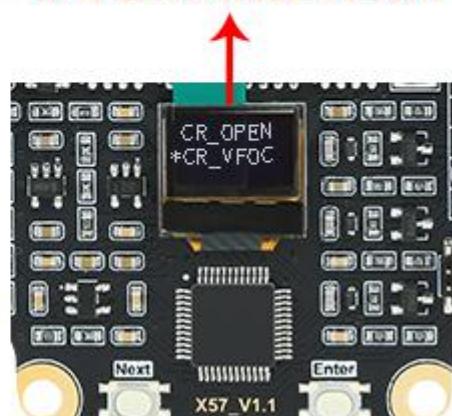
第1行. 锁定按键菜单
第2行. 触发编码器校准
第3行. 控制模式选择菜单

1. “*”号前就是当前选择的菜单
2. 按一下Next键向下选择其他菜单

第1行. 选择为开环模式运行
第2行. 选择为FOC矢量闭环模式运行



按Enter键确认选择退回菜单



五、OLED 菜单功能说明

➤ 菜单项：Lock

菜单功能：锁定按键

菜单选项：Disable、Enable

菜单作用：默认 Disable，按下 Enter 一下就会进入 OLED 菜单，如果设置为 Enable，则需要按住 Enter 按键 3 秒以上才能进入 OLED 菜单，防止按键误触；

➤ 菜单项：Cal

菜单功能：校准编码器

菜单选项：无

菜单作用：闭环模式下对编码器进行线性化插值和电角度对齐，可以提高编码器的线性精度，校准前请确保电机是空载校准；

➤ 菜单项：CtrlMode

菜单功能：控制模式选择

菜单选项：CR_OPEN、CR_VFOC

菜单作用：选择以开环模式运行，还是以 FOC 矢量闭环模式（最高 3000.0RPM+）；

➤ 菜单项：P_Pul

菜单功能：脉冲端口（En、Dir、Stop 引脚）复用功能选择

菜单选项：PUL_OFF、PUL_ENA、ESI_RCO

菜单作用：脉冲端口（En、Dir、Stop 引脚）复用功能选择：

- PUL_OFF：关闭脉冲端口，脉冲输入控制无效；
- PUL_ENA：使能脉冲端口，脉冲输入控制有效；
- ESI_RCO：将 En 引脚复用为多圈限位开关回零的限位开关输入引脚，将 Dir 引脚复用为电机到位输出高电平（如果 Com 不接，就输出 3.3V；如果 Com 接了，则输出 Com 电压）；

➤ 菜单项目： P_Serial

菜单功能：通讯端口（R/A/H、T/B/L 引脚）复用功能选择

菜单选项： RxTx_OFF、ESI_ALO、UART_FUN、CAN1_MAP

菜单作用：选择通讯端口复用为哪种通讯方式：

- RxTx_OFF：关闭通讯端口，通讯控制无效；
- ESI_ALO：将 R/A/H 引脚复用为上电自动触发多圈限位回零功能的限位开关输入引脚，上电后，如果闭环电机发生了堵转，可以给 R/A/H 引脚输入低电平（0V），可以复位堵转保护，将 T/B/L 端口复用为堵转输出功能，发生堵转时该引脚输出高电平 3.3V；
- UART_FUN：将 R/A/H 和 T/B/L 端口复用为串口 TTL/RS232/RS485 通讯控制功能；
- CAN1_MAP：将 R/A/H 和 T/B/L 端口复用为 CAN 通讯控制功能；

➤ 菜单项：En

菜单功能：设置 En 端口的有效电平

菜单选项： L、H、Hold

菜单作用：选择 En 端口的有效电平：

- L: En 端口低电平有效，即 En 端口控制到低电平电机使能，高电平不使能；
- H: En 端口高电平有效，即 En 端口控制到高电平电机使能，低电平不使能；
- Hold: En 端口一直有效，即无论 En 端口是高电平还是低电平，电机一直使能；

注意：En 端口在不接线情况下，En 端口默认是 Hold 模式，即一直使能状态；

➤ 菜单项目：Dir

菜单功能：设置电机旋转的正方向

菜单选项：CW、CCW

菜单作用：设置电机运动的正方向；

➤ 菜单项：MStep

菜单功能：设置细分步数

菜单选项：1、2、4、8、16、32、64、128、256（其他细分可用上位机进行修改）

菜单作用：设置闭环驱动脉冲输入的细分步数，此值要与你的主板设置的细分步数要保持一样，默认 16 细分。如果需要用到其他细分，可以通过串口发送命令进行修改；

➤ 菜单项：MPlyer

菜单功能：设置脉冲输入控制下，内部的细分插补功能

菜单选项：Disable、Enable

菜单作用：使能(Enable)该选项后，驱动器会将你设置的当前细分值，如 16 细分，内部自动插补到最高细分去运行，这将有效的减少电机低速运动时的震动和噪音；

➤ 菜单项：AutoSDD

菜单功能：设置自动熄屏功能

菜单选项：Disable、Enable

菜单作用：使能(Enable)该选项后，在7秒内无任何按键操作时，屏幕会自动关闭显示，需要点亮时，可以按任意按键进行重新点亮。

➤ 菜单项：LPFilter

菜单功能：设置采样电流的低通滤波器强度

菜单选项：Def、Low、Med、Hig、DefLIM、LowLIM、MedLIM、HigLIM

菜单作用：低通滤波器的计算公式为 $out = (ld - k * (ld - td))$ ，其中，ld 为上一次滤波输出值，td 为本次传感器采样值，建议保持默认值。

➤ 菜单项：Ma

菜单功能：设置开环模式的工作电流（电机相电流）

菜单选项：200、400、...、4000（其他电流可用上位机进行修改）

菜单作用：设置开环模式的工作电流。

➤ 菜单项：Ma_Limit

菜单功能：设置 FOC 矢量闭环控制模式下的最大电流（电机相电流）

菜单选项：200、400、...、4000（其他电流可用上位机进行修改）

菜单作用：设置 FOC 矢量闭环控制模式下的最大电流。

➤ 菜单项目: Vm_Limit

菜单功能: 设置 FOC 矢量闭环控制模式下的最大转速

菜单选项: 200、400、...、3000 (单位: RPM, 即转/每分钟)

菜单作用: 设置 FOC 矢量闭环控制模式下的最大转速, 默认是 3000RPM。

➤ 菜单项: CurBW_Hz

菜单功能: 设置 ZDT_X57_V2.0 闭环驱动的电流环带宽

菜单选项: 600、700、...、1500 (单位: rad/s)

菜单作用: 建议保持出厂默认值, 如果需要调整, 则在技术指导下进行调整。

➤ 菜单项: UartBaud

菜单功能: 设置串口/RS232/RS485 通讯的波特率

菜单选项: 9600、19200、25000、38400、57600、115200、256000、512000、921600

菜单作用: 设置串口/RS232/RS485 通讯的波特率, 默认是 115200;

➤ 菜单项: CAN_Baud

菜单功能: 设置 CAN 通讯的速率

菜单选项: 10000、20000、50000、83333、100000、125000、250000、500000、800000、

1000000

菜单作用: 设置 CAN 通讯的速率, 默认是 500000, 即 500KHz;

➤ 菜单项目：ID_Addr

菜单功能：设置串口 TTL/RS232/RS485/CAN 通讯时的电机地址，可设置范围 1-255

菜单选项：1、2、...、10（其他地址可用上位机进行修改）

菜单作用：设置串口 TTL/RS232/RS485/CAN 通讯时的电机地址，可设置范围 1-255，
0 为广播地址；

➤ 菜单项：Checksum

菜单功能：设置串口 TTL/RS232/RS485/CAN 通讯校验字节（每条命令最后一个字节）

菜单选项：0x6B、XOR、CRC-8、Modbus

菜单作用：默认为 0x6B，即通讯控制每条命令的最后一个字节都固定为 0x6B，XOR
和 CRC-8 校验的计算方式请查看“通讯控制”章节，Modbus 协议请查看“Modbus 指令
说明”文档；

➤ 菜单项：Response

菜单功能：设置串口 TTL/RS232/RS485/CAN 通讯控制时**控制动作命令列表**是否回复

菜单选项：None、Receive、Reached、Both、Other

菜单作用：设置串口/RS232/RS485/CAN 总线通讯控制时，主机发送**控制动作命令列**
表中的命令时，从机收到命令后，是否返回确认收到命令，以及是否返回到家命令；

- None：不返回确认收到命令，在发送位置模式命令时也不返回到家命令；
- Receive：只返回确认收到命令（默认值）；
- Reached：只在发送位置模式命令时返回到家命令（地址 + FB/FD + 9F + 6B）；

- Both: 既返回确认收到命令，也在发送位置模式命令时返回到位命令；
- Other: 位置模式下只返回到位命令，其他控制动作命令返回确认收到命令；

➤ 菜单项：S_PosTDP

菜单功能：将通讯控制位置模式的位置角度放大 100 倍进行发送，即精确到 0.01°。

菜单选项：Disable、Enable

菜单作用：使能该选项后，通讯控制位置模式的位置角度将放大 100 倍（保留 2 位小数）进行发送，如直通限速位置模式需要转 360.00°，则发送 01 FB 00 03 E8 00 00 8C A0 00 00 6B；如果该选项没使能，则发送 01 FA 01 03 E8 00 00 0E 10 00 00 6B。

➤ 菜单项：Clog_Pro

菜单功能：设置堵转保护功能

菜单选项：Disable、Enable

菜单作用：不管堵转保护功能是否开启，电机发生堵转时，蓝色 LED 都会点亮提示；如果使能(Enable)该选项，则堵转时电机将自动关闭闭环驱动器，可以保护电机和驱动器不受堵转大电流的损坏。如果 P_Serial 设置为 ESI_ALO 选项时，则堵转保护后 T/B/L 引脚将输出 3.3V；

(触发堵转条件：电机实时转速 < 设置的堵转检测转速阈值 + 电机实时相电流 > 设置的堵转检测相电流阈值)

(触发堵转保护条件：电机实时转速 < 设置的堵转检测转速阈值 + 电机实时相电流 > 设置的堵转检测相电流阈值 + 持续时间 > 设置的堵转检测时间阈值)

➤ 菜单项目：Clog_Rpm

菜单功能：设置堵转检测的转速阈值

菜单选项：4、6、...、30（单位：RPM，即转/每分钟）

（如果需要设置为其他值，可以用上位机进行修改）

菜单作用：堵转保护检测判定条件之一：电机实时转速 < 堵转检测转速阈值。

➤ 菜单项：Clog_Ma

菜单功能：设置堵转检测的相电流阈值

菜单选项：2000、2200、...、4000（单位：Ma）

（如果需要设置为其他值，可以用上位机进行修改）

菜单作用：堵转保护检测判定条件之二：电机实时相电流 > 堵转检测相电流阈值。

➤ 菜单项：Clog_Ms

菜单功能：设置堵转检测的时间阈值

菜单选项：1000、2000、...、12000（单位：Ms）

（如果需要设置为其他值，可以用上位机进行修改）

菜单作用：堵转保护检测判定条件之三：持续时间 > 堵转检测时间阈值。

➤ 菜单项：O_Mode

菜单功能：设置上电自动触发回零功能的回零模式

菜单选项：Nearest、Dir、Senseless、EndStop

菜单作用：设置上电自动触发回零的回零模式，分别为单圈就近回零、单圈方向回零、多圈无限位碰撞回零、多圈有限位开关回零。原点回零操作说明请参考第八章节；

➤ 菜单项：0_Dir

菜单功能：设置回零方向

菜单选项：CW、CCW

菜单作用：设置单圈方向回零、多圈无限位碰撞回零、多圈有限位开关回零的方向。

➤ 菜单项：0_Vel

菜单功能：设置回零转速

菜单选项：30、60、...、300（单位：RPM）

菜单作用：设置回零的速度。

➤ 菜单项：0_Tmo_Ms

菜单功能：设置回零超时时间

菜单选项：2000、4000、...、20000（单位：毫秒）

菜单作用：设置回零超时时间，回零超过这个时间就自动中断并退出回零。

➤ 菜单项：0_Set

菜单功能：设置当前位置为单圈就近/方向回零的零点位置

菜单选项：Set 0、Clear 0、Exit

菜单作用：设置/清除当前位置为单圈就近/方向回零的零点位置。

(无限位碰撞回零触发条件: 电机实时转速 < 设置的碰撞回零转速阈值 + 电机实时相电流 > 设置的碰撞回零相电流阈值 + 持续时间 > 设置的碰撞回零时间阈值)

➤ 菜单项: 0_SL_Rpm

菜单功能: 设置无限位碰撞回零的检测转速

菜单选项: 30、60、...、300 (单位: RPM)

菜单作用: 碰撞回零检测判定条件之一: 电机实时转速 < 碰撞回零检测转速。

➤ 菜单项: 0_SL_Ma

菜单功能: 设置无限位碰撞回零的检测电流

菜单选项: 400、600、...、2200 (单位: Ma)

菜单作用: 碰撞回零检测判定条件之二: 电机实时相电流 > 碰撞回零检测电流。

➤ 菜单项: 0_SL_Ms

菜单功能: 设置无限位碰撞回零的检测时间

菜单选项: 20、40、...、200 (单位: 毫秒)

菜单作用: 碰撞回零检测判定条件之三: 持续时间 > 碰撞回零检测时间。

➤ 菜单项: 0_POT_En

菜单功能: 设置上电自动触发回零功能

菜单选项: Disable、Enable

菜单作用: 设置上电自动触发回零功能, 使能后, 每次上电都会自动执行回零功能。

➤ 菜单项目: Restore

菜单功能: 恢复出厂设置

菜单选项: Yes、No

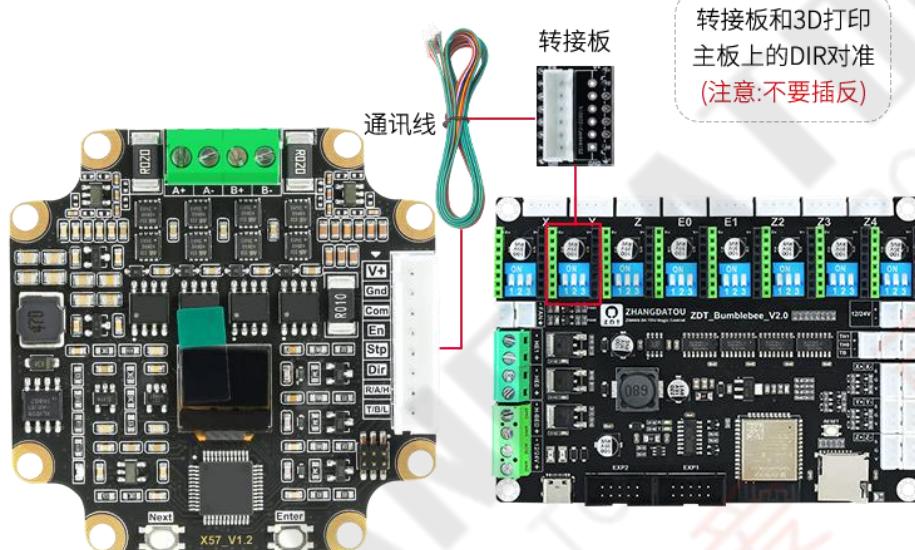
菜单作用: 恢复出厂设置, 需要断电重新上电, 重新空载校准编码器。

六、脉冲控制

6.1 脉冲控制接线

6.1.1 3D 打印主板接线

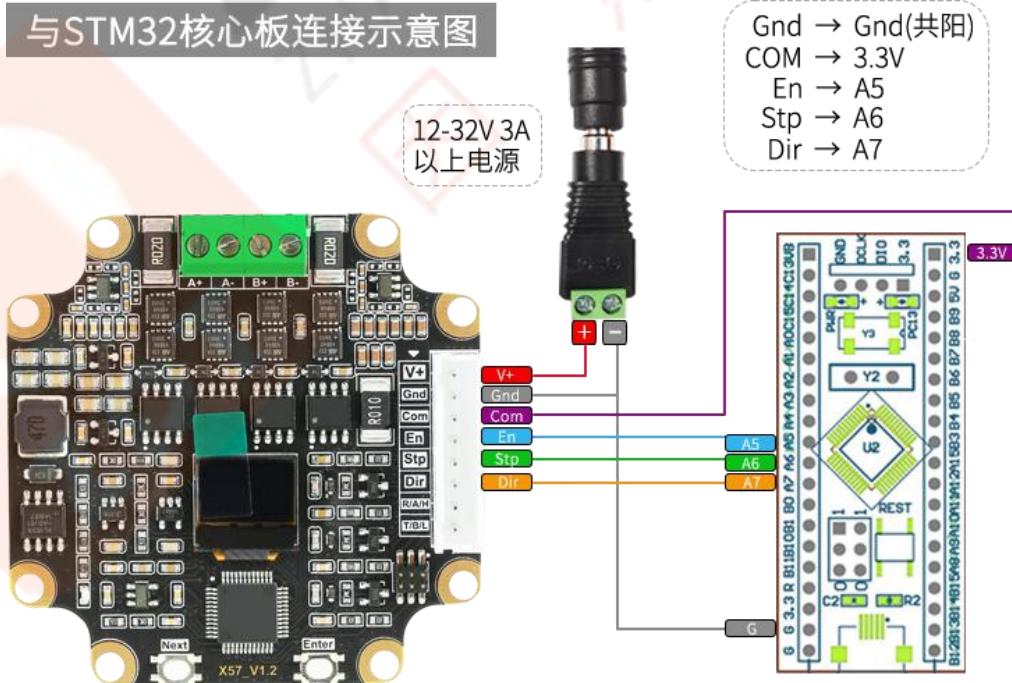
与大黄蜂3D打印主板连接示意图



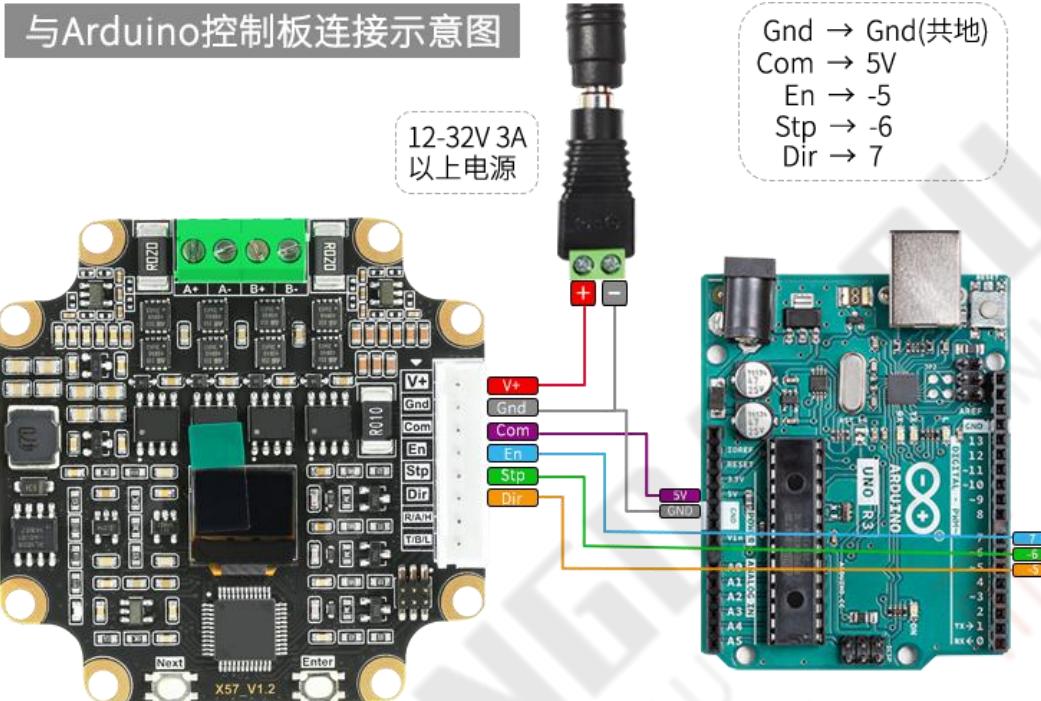
(注意：如果需要关闭电机，用手能够拧得动电机的功能，可以在 En 选项里设置为 L，默认是 Hold)

6.1.2 STM32 脉冲控制接线

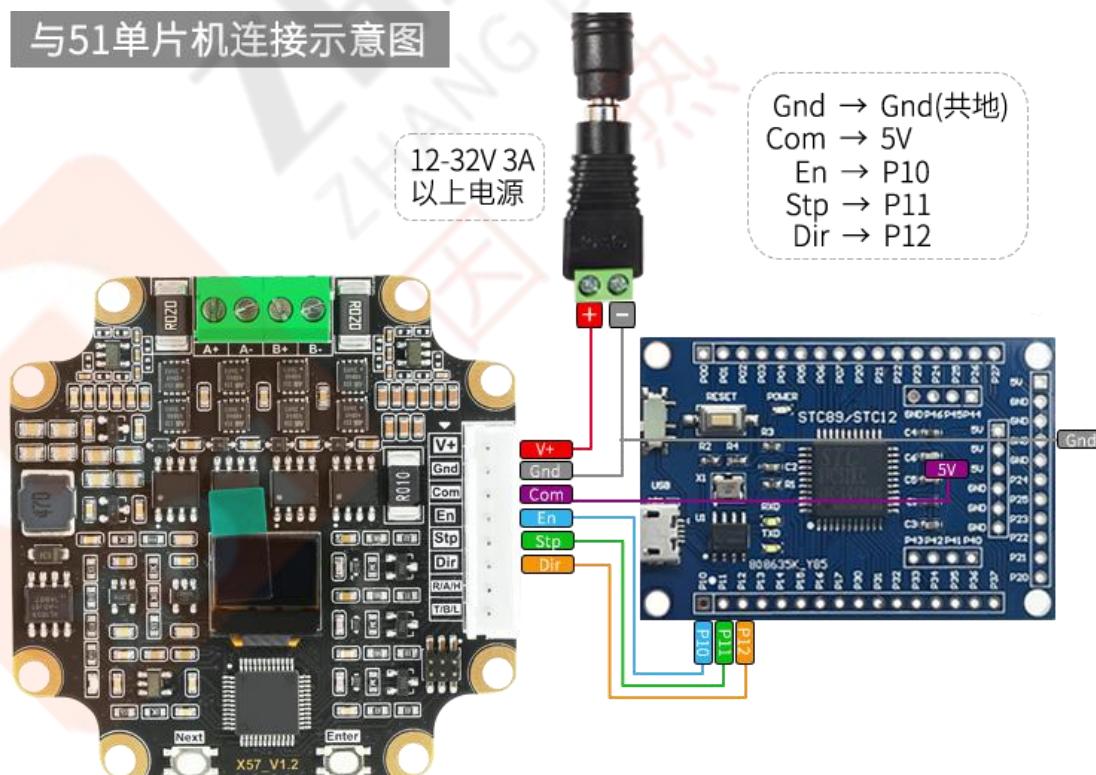
与STM32核心板连接示意图



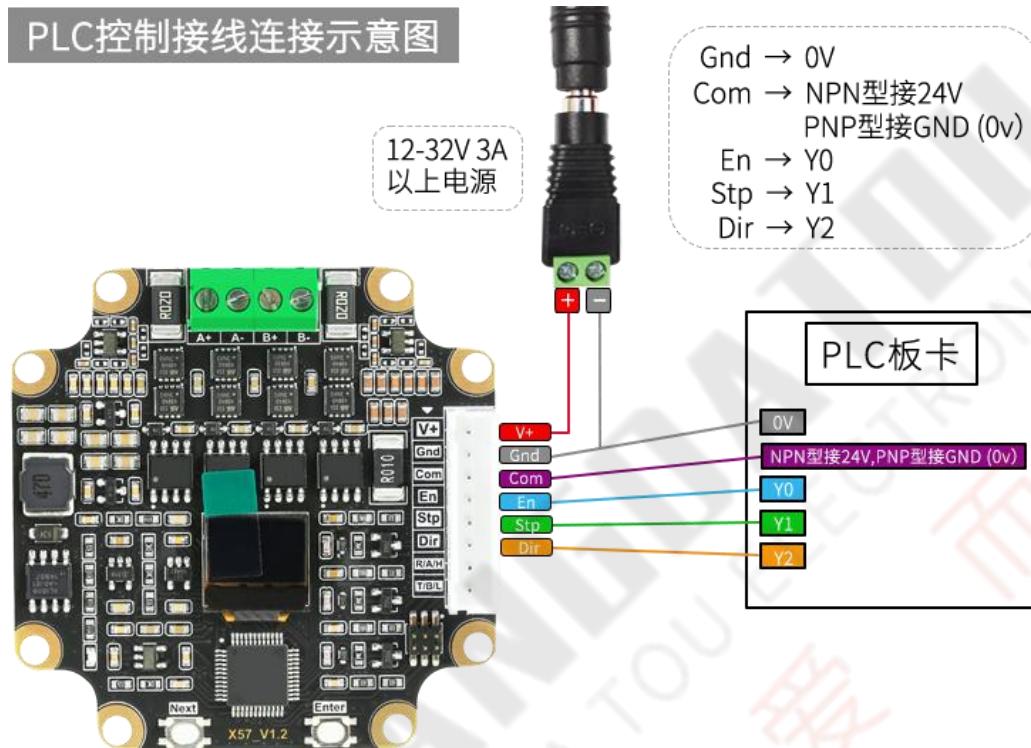
6.1.3 Arduino 脉冲控制接线



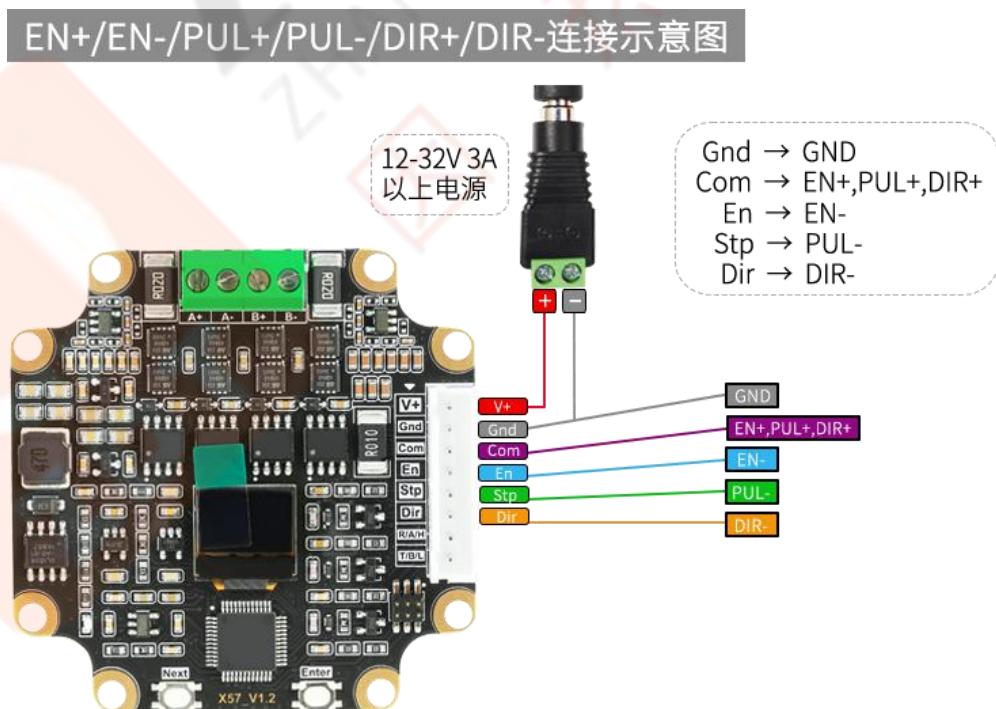
6.1.4 51 单片机脉冲控制接线



6.1.5 PLC 脉冲控制接线



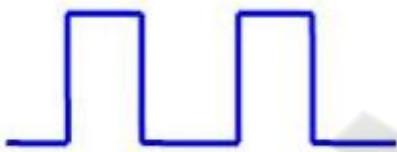
6.1.6 EN+/EN-/PUL+/PUL-/DIR+/DIR-差分脉冲控制接线



6.2 脉冲控制说明

1. 什么是脉冲？

控制器控制某一个 IO 引脚输出一个 0（低电平）和一个 1（高电平），就是一个脉冲，相当于一个 IO 引脚取反两次，下图是控制器 IO 引脚输出 2 个脉冲，如图所示：



2. 如何发脉冲去控制 ZDT_X57_V2.0 闭环驱动器？

按照前面章节安装并校准好编码器后，用你的单片机的一个 IO 引脚向 ZDT_X57_V2.0 闭环驱动器的 Stp 引脚发送脉冲过去，就可以控制电机进行旋转。

3. 如何控制旋转的角度？

以 1.8° 步进电机、16 细分为例，就是发送 3200 个脉冲，电机就旋转一圈 360° ，如果是 32 细分，那就是发送 6400 个脉冲，电机旋转一圈，其他细分或角度可以类推。

4. 如何控制旋转的方向？

用控制器另外一个 IO 引脚控制 ZDT_X57_V2.0 闭环驱动器的 Dir 端口，输出 0 为一个方向旋转，输出 1 为另外一个方向旋转。

5. 如何控制旋转的速度？

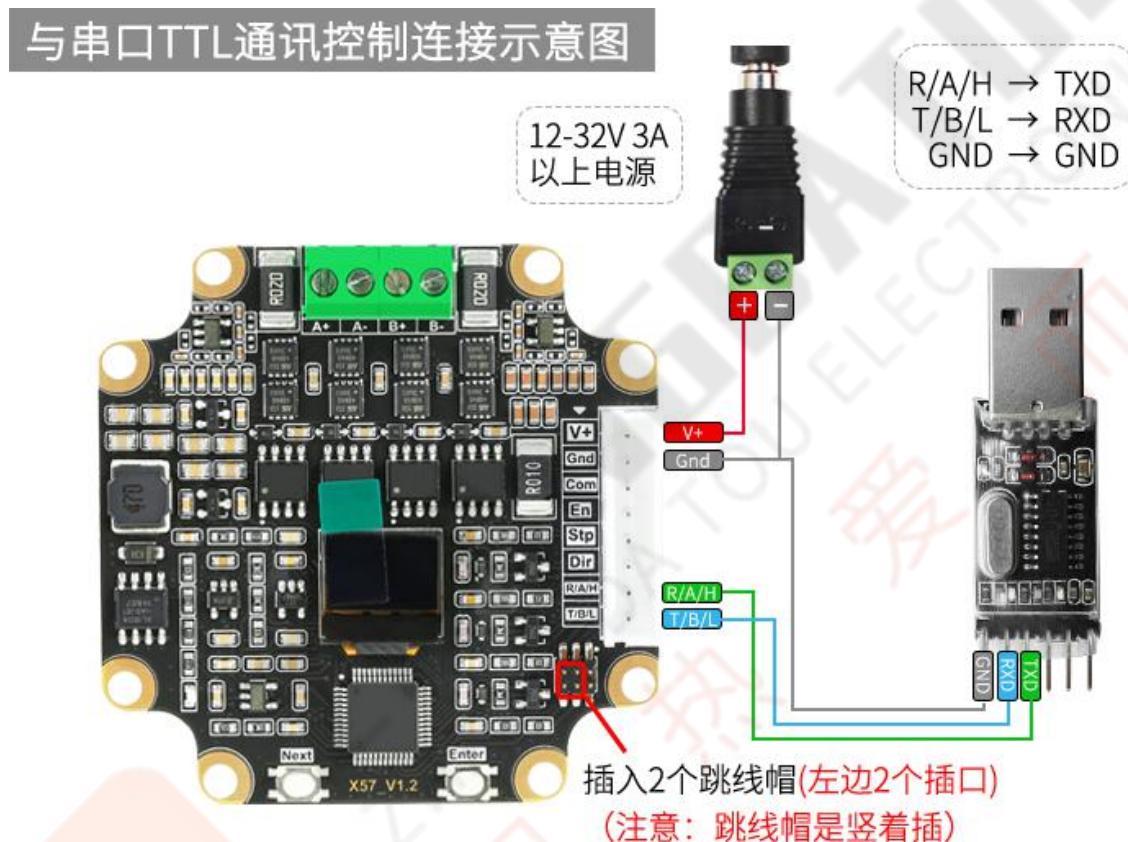
控制器输出 0 和 1 的时间加快，就能加快电机旋转的速度，也就是加快脉冲频率。

七、通讯控制

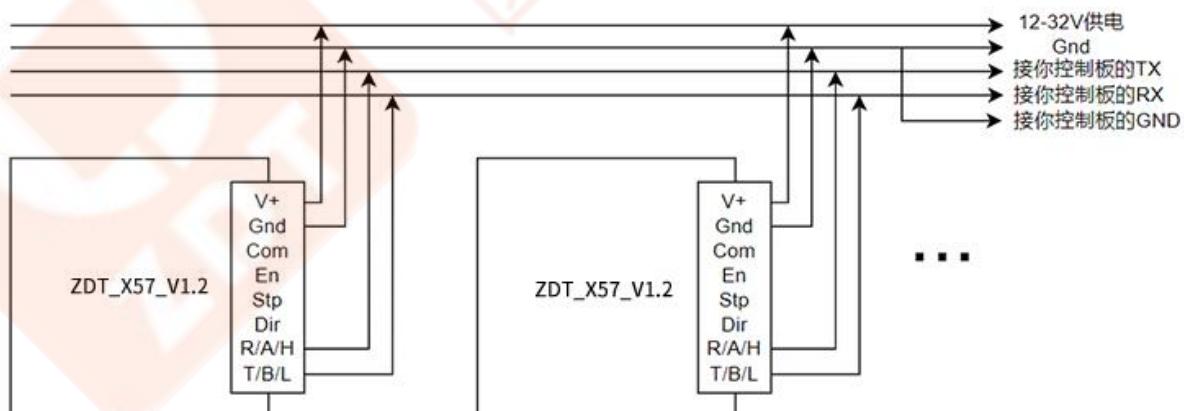
7.1 通讯控制接线

7.1.1 串口 TTL 通讯控制接线

串口 TTL 单机通讯控制接线：



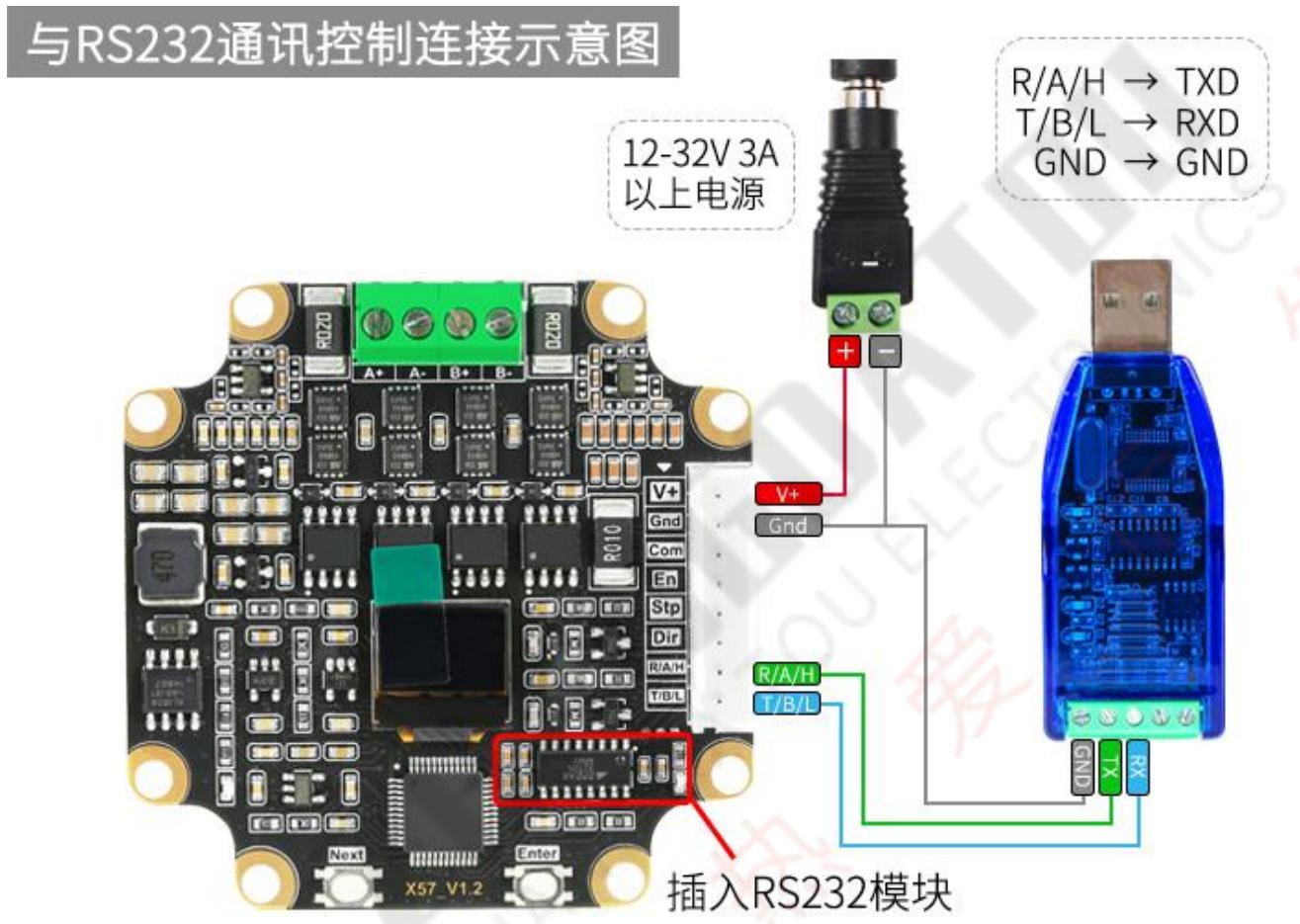
串口TTL多机通讯控制接线：



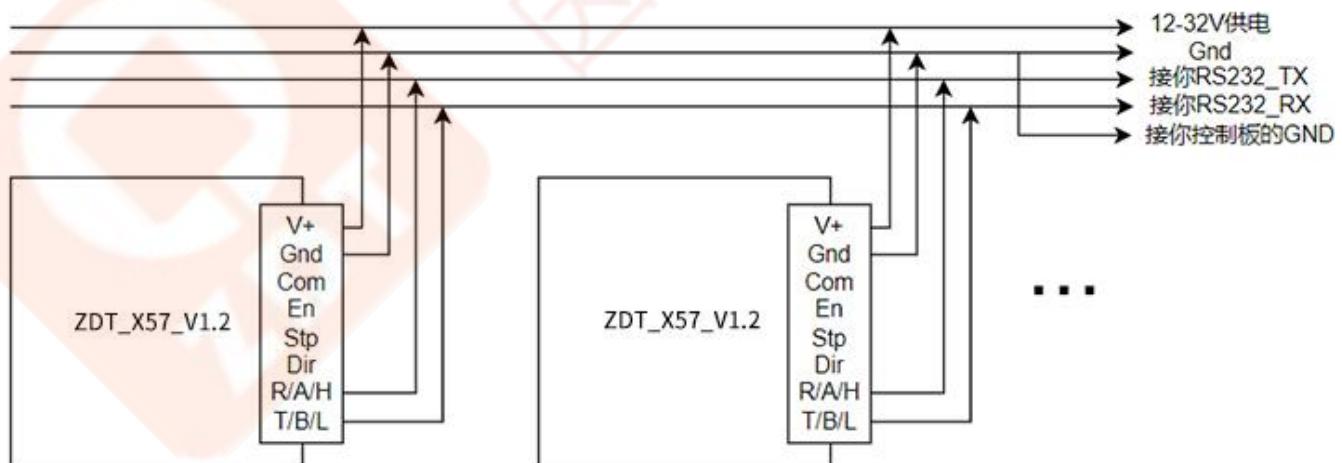
(注意：每个 ZDT_X57_V1.2 闭环驱动的插针上左边都需要插入两个跳线帽，是竖着插)

7.1.2 RS232 通讯控制接线

RS232 单机通讯控制接线：



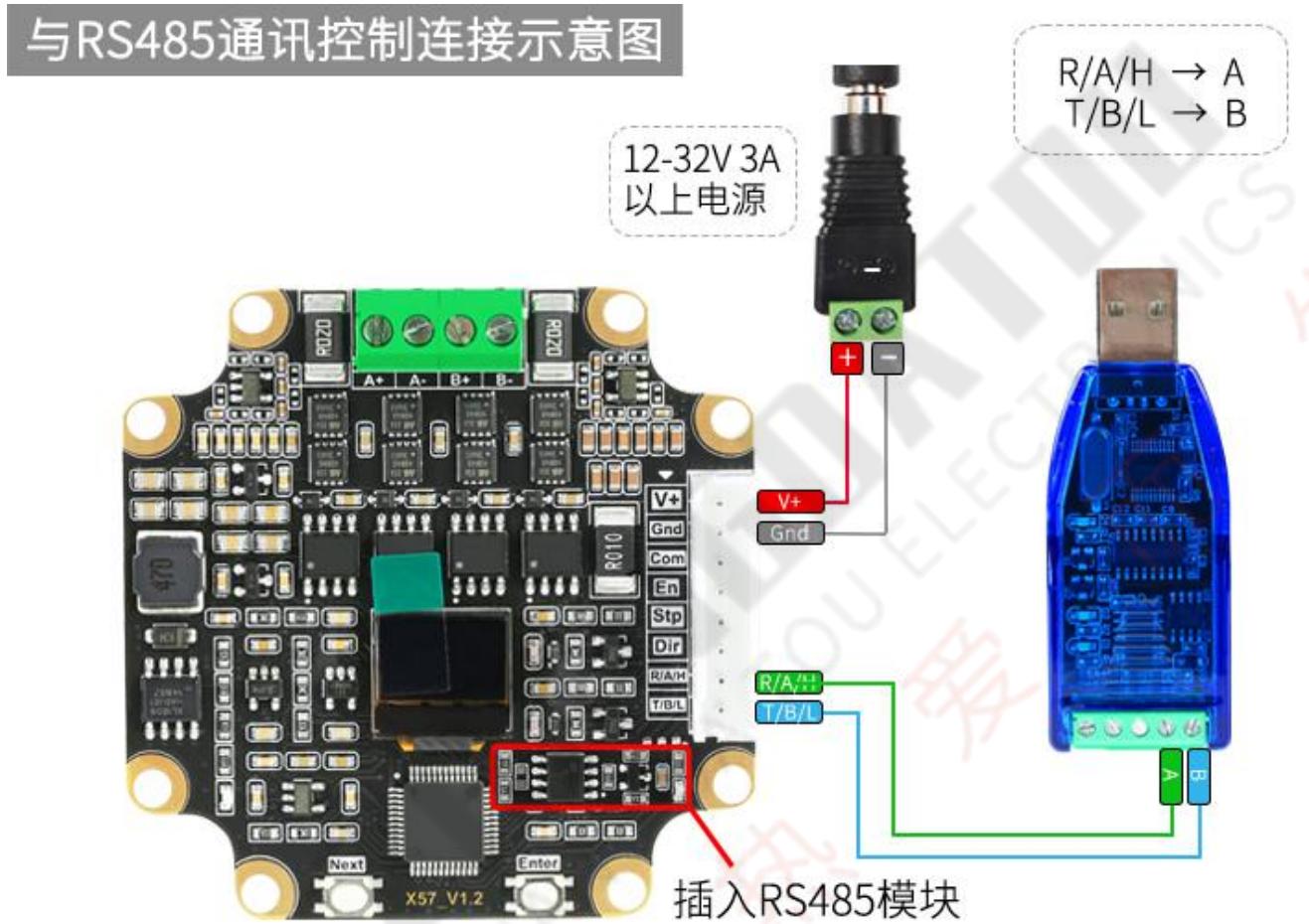
RS232多机通讯控制接线：



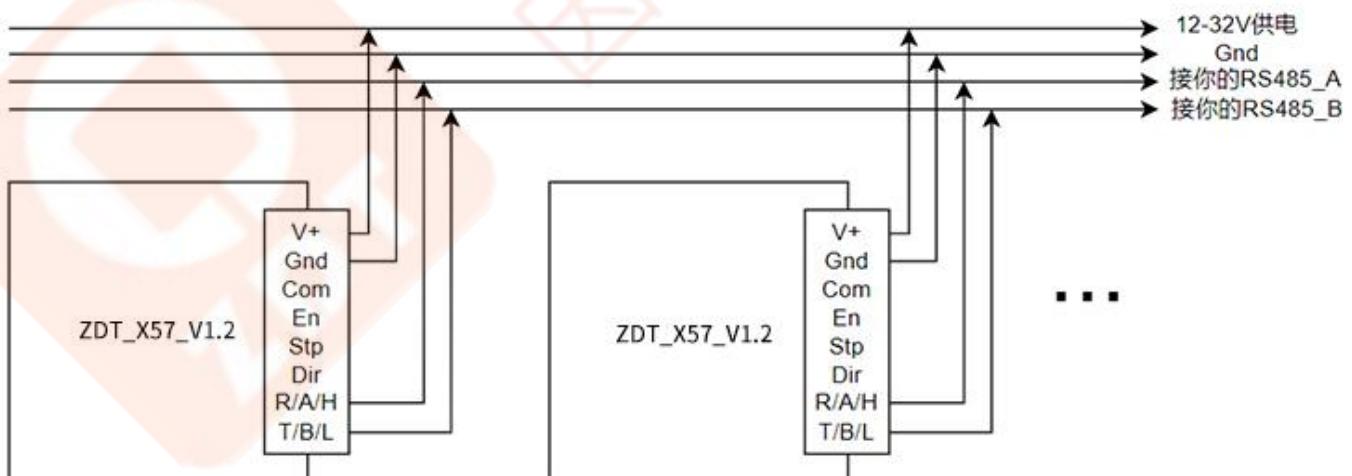
(注意：每个 ZDT_X57_V1.2 闭环驱动的插针上都需要插入一个 ZDT_RS232 小模块)

7.1.3 RS485 通讯控制接线

RS485 单机通讯控制接线：



RS485多机通讯控制接线：

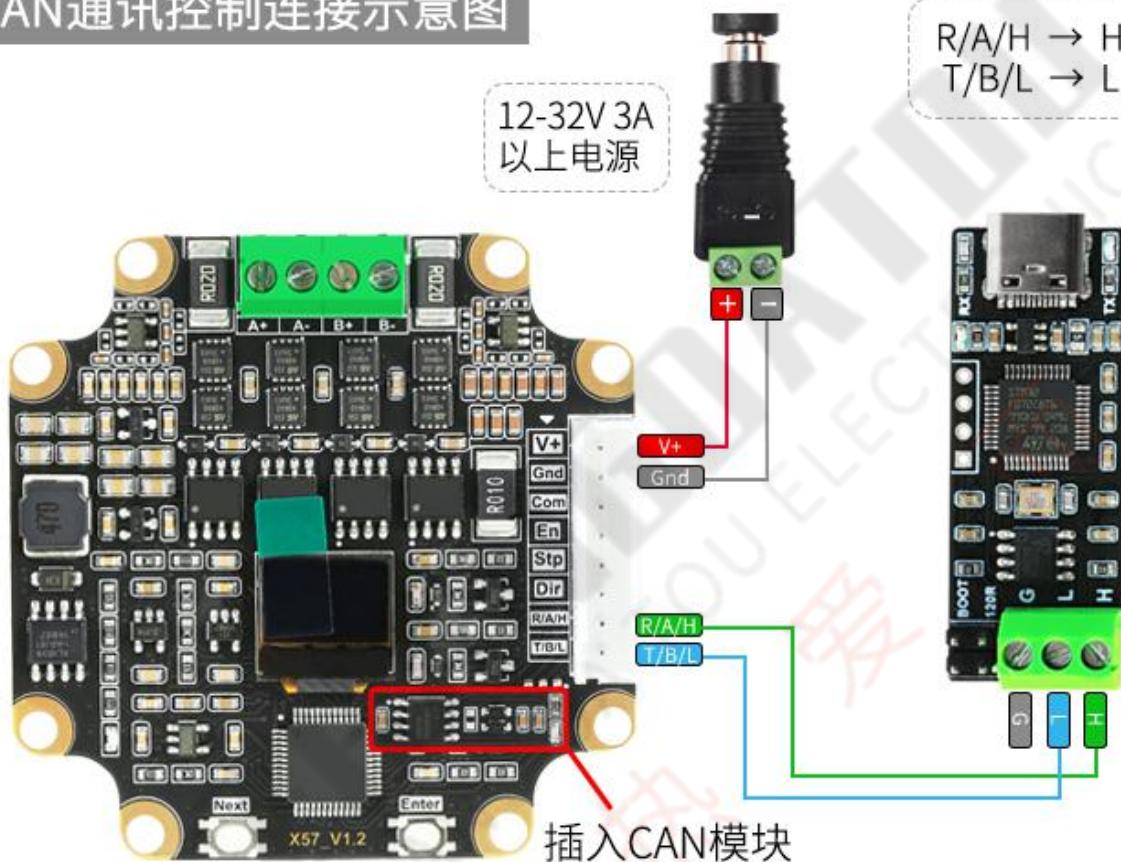


(注意：每个 ZDT_X57_V1.2 闭环驱动的插针上都需要插入一个 ZDT_RS485 小模块)

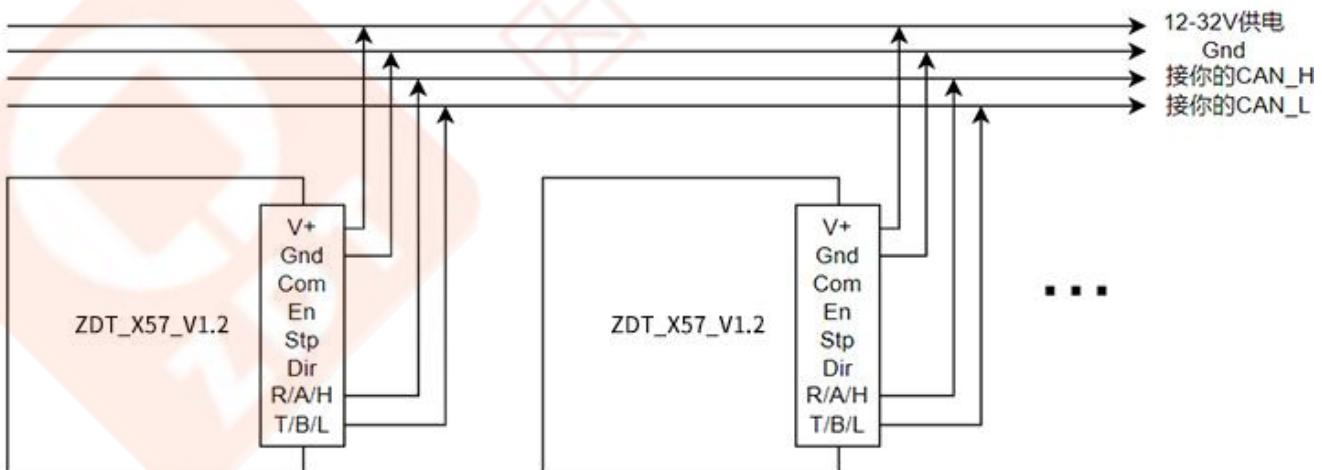
7.1.4 CAN 通讯控制接线

CAN 单机通讯控制接线：

与CAN通讯控制连接示意图



CAN多机通讯控制接线：



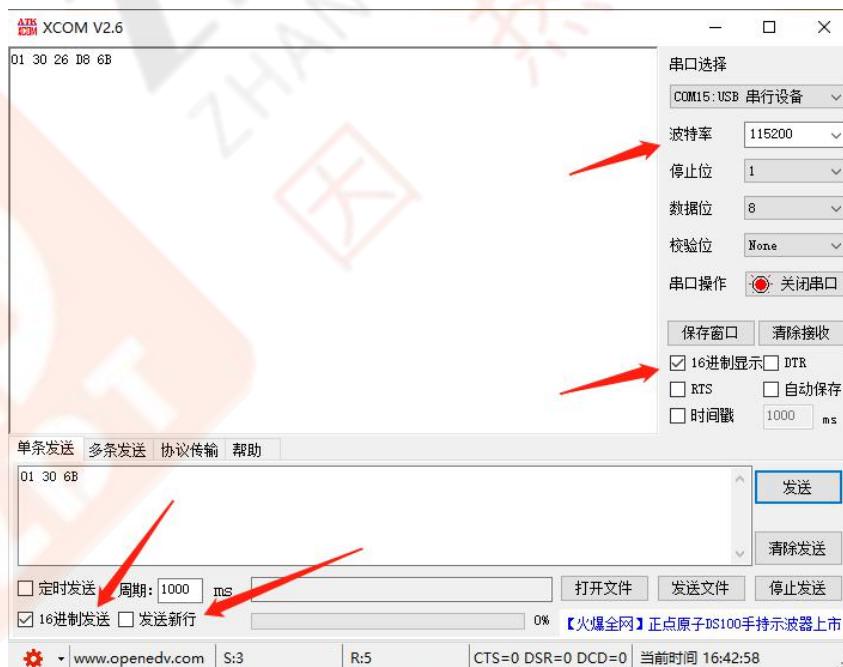
(注意：每个 ZDT_X57_V1.2 闭环驱动的插针上都需要插入一个 ZDT_CAN 小模块)

7.2 通讯控制命令格式说明

7.2.1 串口 TTL/RS232/RS485 通讯控制命令格式说明

电机地址	功能码	(辅助码)	指令数据			校验字节
ID_Addr	Fun_Code	(Aux_Code)				CheckSum

1. 电机地址默认为 1，可设置范围为 1-255，0 为广播地址；
2. 以广播地址 0 发送命令，则所有的闭环电机都收到并执行这条命令，其中，只有地址为 1 的电机会回复这条命令（注意：Modbus 协议不回复）；
3. Checksum 默认为 0x6B，即每条命令的最后一个字节都是 0x6B 作为校验字节，比如 01 45 42 **6B**；如果设置 XOR，则为 01 06 45 **42**，其中 $0x42 = 0x01 \wedge 0x06 \wedge 0x45$ ；如果设置 CRC-8，则请参考文末附录中的 CRC-8 计算方法进行计算最后一个字节；
4. 串口 TTL 通讯负载能力较低，一个串口 TTL 建议挂载 4 个或以下的闭环；
5. RS485 总线的负载能力理论上最多能够挂载 256 个从机；
6. 串口助手设置如下图所示：



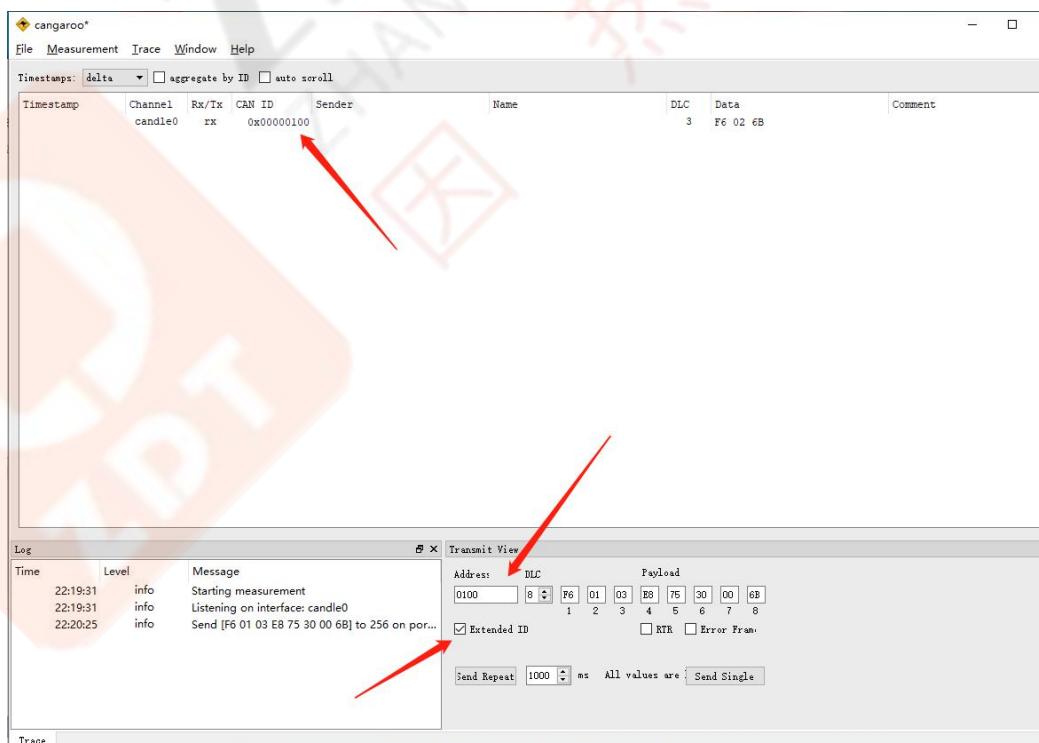
(勾选 16 进制发送和 16 进制显示，取消勾选发送新行，波特率默认是 115200)

7.2.2 CAN 通讯控制命令格式说明

功能码	(辅助码)	指令数据			校验字节
Fun_Code	(Aux_Code)				CheckSum

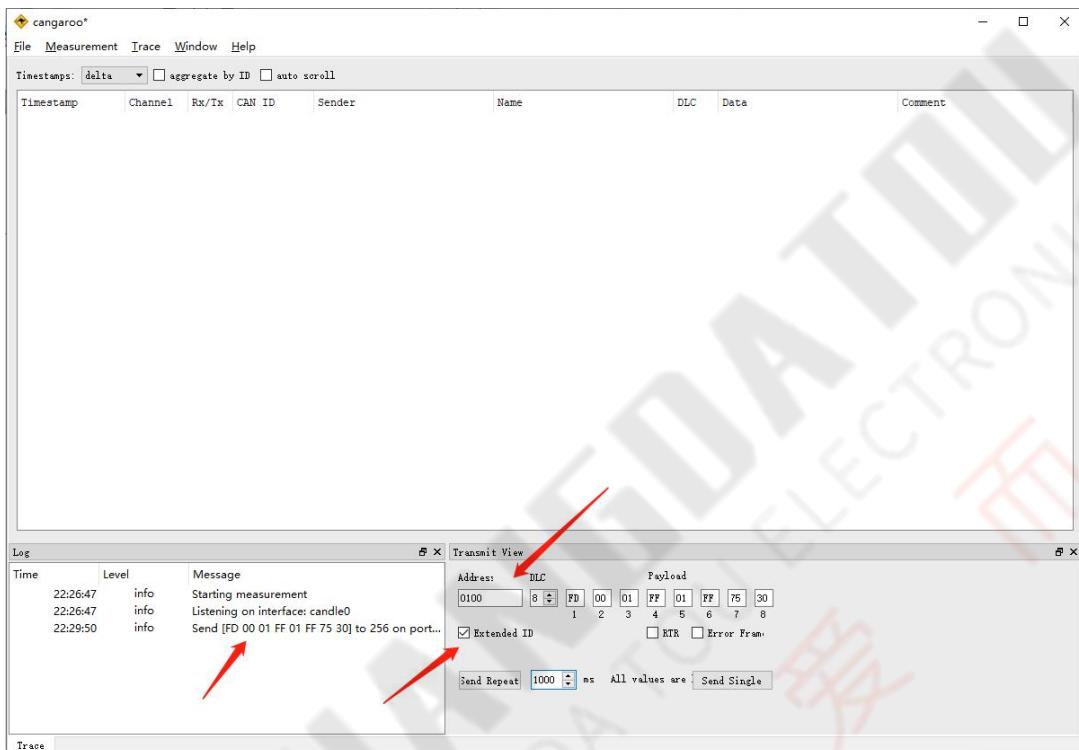
1. 将闭环 P_Serial 菜单设置为 CAN1_MAP，固定为扩展帧，默认速率为 500K；
 2. **对于小于 8 字节的命令**，则主机的帧 ID 设置为($ID_Addr \ll 8$)，比如，将帧 ID 设置为 0x0100，然后发送 45 42 6B，等同于串口通讯发送命令 01 45 42 6B；
 3. **对于大于 8 字节的命令**，则需要将命令进行拆包和分包发送，则主机的帧 ID 设置为(($ID_Addr \ll 8$) + 第几包数据)，并且每包数据的首个字节都为功能码，比如，梯形曲线加减速位置模式命令：01 FD 00 01 FF 01 FF 75 30 00 36 EE 80 00 00 6B，则
 - (1) 先设置帧 ID 为 0100，然后发送 FD 00 01 FF 01 FF 75 30
 - (2) 再设置帧 ID 为 0101，然后发送 FD 00 36 EE 80 00 00 6B
4. 下面提供上位机 cangaroo 的参考示例：

① 小于 8 字节命令（速度模式：01 F6 01 03 E8 75 30 00 6B）

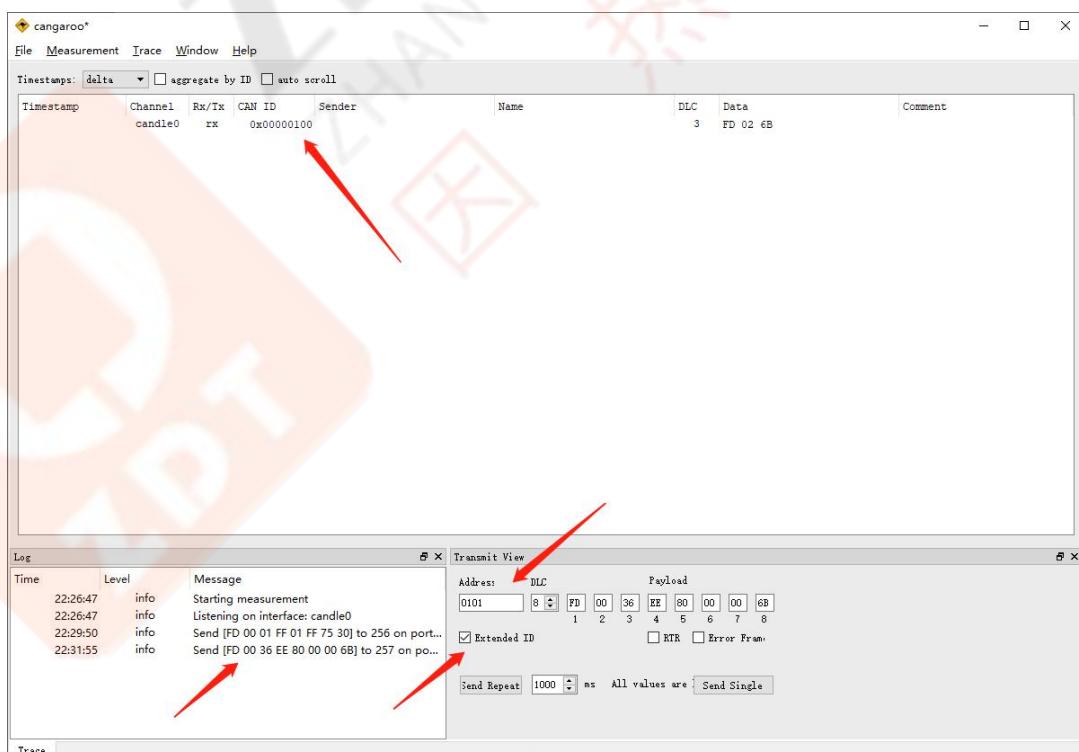


② 大于 8 字节命令 (梯形位置: 01 FD 00 01 FF 01 FF 75 30 00 36 EE 80 00 00 6B)

先发送 FD 00 01 FF 01 FF 75 30 (注意: 帧 ID 为 0100, 表示第 0 包数据)



再发送 FD 00 36 EE 80 00 00 6B (注意: 帧 ID 为 0101, 表示第 1 包数据)



7.3 多机通讯及同步控制

多机通讯:

1. 多机通讯的接线见 7.1 章节“通讯控制接线”。
2. 修改每个电机的 ID 地址不同，可在菜单 ID_Addr 中进行修改。
3. 按电机 ID 地址发送命令，控制电机运动，比如有地址 1 和地址 2 两个电机：
 - 发送 01 F6 01 03 E8 75 30 00 6B，则 1 地址电机执行这条命令运动；
 - 发送 02 F6 01 03 E8 75 30 00 6B，则 2 地址电机执行这条命令运动；
 - 发送 00 F6 01 03 E8 75 30 00 6B，则 1 和 2 地址电机都执行这条命令；

同步控制:

比如有地址 1、地址 2 两个电机，假设需要：

- ◆ 1 地址电机用直通限速位置模式运动 -3600.0°；
- ◆ 2 地址电机用梯形曲线位置模式运动 7200.0°；
- ◆ 1 和 2 地址电机要同步运动（同时开始运动）；

则，可以按照以下操作进行两者的同步控制：

1. 先发送直通限速位置模式命令 01 FB 01 03 E8 00 00 8C A0 00 01 6B，正确返回 01 FB 02 6B，此时 1 地址电机先不会运动，因为多机同步标志为 01；
2. 再发送梯形位置模式命令 02 FD 01 01 FF 01 FF 27 10 00 01 19 40 00 01 6B，正确返回 02 FD 02 6B，此时 2 地址电机先不会运动，因为多机同步标志为 01；
3. 再用广播地址发送多机同步运动命令 00 FF 66 6B，此时 1 和 2 地址电机同时开始运动；

7.4 通讯控制命令列表

7.4.1 控制动作命令列表

➤ 命令功能：电机使能控制

命令格式：地址 + 0xF3 + 0xAB + 使能状态 + 多机同步标志 + 校验字节

命令返回：地址 + 0xF3 + 命令状态 + 校验字节

命令示例：发送 01 F3 AB 01 00 6B，正确返回 01 F3 02 6B，错误命令返回 01 00 EE 6B

数据解析：控制电机使能发送 01 F3 AB 01 00 6B；不使能发送 01 F3 AB 00 00 6B

➤ 命令功能：力矩模式控制

命令格式：地址 + 0xF5 + 符号 + 电流斜率 + 电流 + 多机同步标志 + 校验字节

命令返回：地址 + 0xF5 + 命令状态 + 校验字节

命令示例：发送 01 F5 00 03 E8 03 20 00 6B，正确返回 01 F5 02 6B，条件不满足返回 01 F5 E2 6B，错误命令返回 01 00 EE 6B

数据解析：00 表示正电流（01 表示负），03 E8 表示 $0x03E8 = 1000\text{Ma/s}$ ，03 20 表示 $0x0320 = 800\text{Ma}$ ，00 表示多机同步标志，多机同步控制原理请参考“7.3 多机通讯及同步控制”章节；（条件不满足情况：触发了堵转保护、电机没使能）

总结：这条命令表示以 1000Ma/s 的电流斜率增加/减少电机的相电流到 800Ma 。

➤ 命令功能：速度模式控制

命令格式：地址 + 0xF6 + 符号 + 速度斜率 + 速度 + 多机同步标志 + 校验字节

命令返回: 地址 + 0xF6 + 命令状态 + 校验字节

命令示例: 发送 01 F6 01 03 E8 4E 20 00 6B, 命令正确返回 01 F6 02 6B, 条件不满足返回 01 F6 E2 6B, 错误命令返回 01 00 EE 6B

数据解析: 01 表示负转速 (00 表示正), 03 E8 表示 $0x03E8 = 1000\text{RPM}/\text{s}$, 4E 20 表示 $0x4E20 = 20000 = 2000.0\text{RPM}$, 00 表示多机同步标志, 多机同步控制原理请参考“7.3 多机通讯及同步控制”章节; (条件不满足情况: 触发了堵转保护、电机没使能)

总结: 这条命令表示以 $1000\text{RPM}/\text{s}$ 的加速度增加/减少电机的转速到 2000.0RPM 。

➤ 命令功能: 直通限速位置模式控制

命令格式: 地址 + 0xFB + 符号 (方向) + 速度 + 位置角度 + 相对位置/绝对位置标志 + 多机同步标志 + 校验字节

命令返回: 地址 + 0xFB + 命令状态 + 校验字节

命令示例: 发送 01 FB 01 4E 20 00 00 8C A0 00 00 6B, 正确返回 01 FB 02 6B, 条件不满足返回 01 FB E2 6B, 错误命令返回 01 00 EE 6B

数据解析: 01 表示负位置 (00 表示正), 4E 20 表示 $0x4E20 = 20000 = 2000.0\text{RPM}$, 00 00 8C A0 表示 $0x00008CA0 = 36000 = 3600.0^\circ$, 00 表示相对位置运动 (01 表示绝对), 00 表示多机同步标志, 多机同步控制原理请参考“7.3 多机通讯及同步控制”章节; (条件不满足情况: 触发了堵转保护、电机没使能)

总结: 这条命令表示以 2000.0RPM 的速度运动到相对当前位置角度的 -3600.0° 的位置上, 就是在当前位置的基础上以 2000.0RPM 的转速再转 10 圈。

(注意: 如果是绝对位置模式 01, 则是直接运动到 -3600.0° 的位置上)

➤ 命令功能：梯形曲线位置模式控制

命令格式：地址 + 0xFD + 符号（方向）+ 加速加速度 + 减速加速度 + 最大速度
+ 位置角度 + 相对位置/绝对位置标志 + 多机同步标志 + 校验字节

命令返回：地址 + 0xFD + 命令状态 + 校验字节

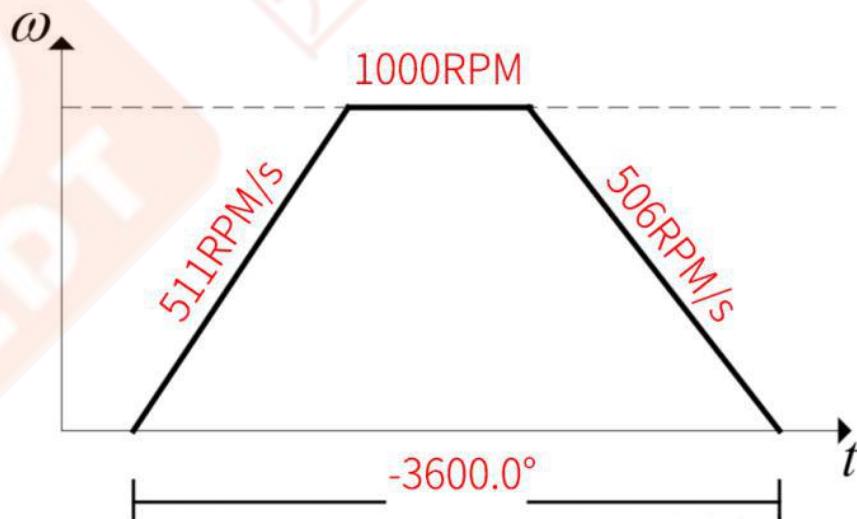
命令示例：发送 01 FD 01 01 FF 01 FA 27 10 00 00 8C A0 00 00 6B，正确返回
01 FD 02 6B，条件不满足返回 01 FD E2 6B，错误命令返回 01 00 EE 6B

(条件不满足情况：触发了堵转保护、电机没使能)

数据解析：01 表示负位置（00 表示正），01 FF 表示 $0x01FF = 511\text{RPM/s}$ ，01 FA 表示 $0x01FA = 506\text{RPM/s}$ ，27 10 表示 $0x2710 = 10000 = 1000.0\text{RPM}$ ，00 00 8C A0 表示 $0x00008CA0 = 36000 = 3600.0^\circ$ ，00 表示相对位置运动（01 表示绝对），00 表示多机同步标志，多机同步控制原理请参考“7.3 多机通讯及同步控制”章节；

总结：这条命令表示以 511RPM/s 的加速加速度， 506RPM/s 的减速加速度，最大转速 1000.0RPM 的梯形曲线，运动到相对当前位置角度的 -3600.0° 的位置上，就是在当前位置的基础上以 2000.0RPM 的转速再转 10 圈。

(注意：如果是绝对位置模式 01，则是直接运动到 -3600.0° 的位置上)



➤ 命令功能：立即停止

命令格式：地址 + 0xFE + 0x98 + 多机同步标志 + 校验字节

命令返回：地址 + 0xFE + 命令状态 + 校验字节

命令示例：发送 01 FE 98 00 6B，正确返回 01 FE 02 6B，条件不满足返回 01 FE E2 6B，错误命令返回 01 00 EE 6B（**条件不满足情况：**触发了堵转保护、正在回零）

数据解析：让电机立即停止运动，适用于力矩模式、速度模式、位置模式都可以。

➤ 命令功能：多机同步运动

命令格式：地址 + 0xFF + 0x66 + 校验字节

命令返回：地址 + 0xFF + 命令状态 + 校验字节

命令示例：发送 01 FF 66 6B，正确返回 01 FF 02 6B，条件不满足返回 01 FF E2 6B，错误命令返回 01 00 EE 6B（**条件不满足情况：**触发了堵转保护、电机没使能）

数据解析：让多个电机同时开始运动，请参考“7.3 多机通讯及同步控制”章节。

7.4.2 原点回零命令列表

➤ 命令功能：设置单圈回零的零点位置

命令格式：地址 + 0x93 + 0x88 + 是否存储标志 + 校验字节

命令返回：地址 + 0x93 + 命令状态 + 校验字节

命令示例：发送 01 93 88 01 6B，正确返回 01 93 02 6B，错误命令返回 01 00 EE 6B

数据解析：可以让电机运动到想要的位置，然后发送命令设置单圈回零的零点位置。

➤ 命令功能：触发回零

命令格式：地址 + 0x9A + 回零模式 + 多机同步标志 + 校验字节

命令返回：地址 + 0x9A + 命令状态 + 校验字节

命令示例：发送 01 9A 00 00 6B，正确返回 01 9A 02 6B，条件不满足返回 01 9A E2 6B，错误命令返回 01 00 EE 6B

(条件不满足的情况有：触发了堵转保护、单圈回零的零点位置值无效)

数据解析：设置完原点回零参数后，可以发送该命令触发原点回零功能。

➤ 命令功能：强制中断并退出回零操作

命令格式：地址 + 0x9C + 0x48 + 校验字节

命令返回：地址 + 0x9C + 命令状态 + 校验字节

命令示例：发送 01 9C 0x48 6B，正确返回 01 9C 02 6B，条件不满足返回 01 9C E2 6B，错误命令返回 01 00 EE 6B

(条件不满足的情况有：当前没有触发回零操作)

数据解析：正在回零的过程中，可以使用该命令强制中断并退出回零操作。

➤ 命令功能：读取原点回零参数

命令格式：地址 + 0x22 + 校验字节

命令返回：地址 + 0x22 + 原点回零参数 + 校验字节

命令示例：发送 01 22 6B，正确返回 01 22 00 00 00 1E 00 00 27 10 0F A0 03 20 00 3C 00 6B，错误命令返回 01 00 EE 6B

数据解析: 返回的原点回零参数, 它将按照下面的数据格式进行排列和数据的转换:

回零模式	= 0x00 = Nearest = 单圈就近回零模式
回零方向	= 0x00 = CW
回零转速	= 0x001E = 30 (RPM)
回零超时时间	= 0x00002710 = 10000 (ms)
无限位碰撞回零检测转速	= 0x0FA0 = 4000 (RPM)
无限位碰撞回零检测电流	= 0x0320 = 800 (mA)
无限位碰撞回零检测时间	= 0x003C = 60 (ms)
是否使能上电自动触发回零功能	= 0x00 = 不使能

(无限位碰撞回零检测判定条件: 电机实时转速 < 碰撞回零检测转速 + 电机实时相电流 > 碰撞回零检测电流 + 持续时间 > 碰撞回零检测时间)

➤ 命令功能: 修改原点回零参数

命令格式: 地址 + 0x4C + 0xAE + 是否存储标志 + 原点回零参数 + 校验字节

命令返回: 地址 + 0x4C + 命令状态 + 校验字节

命令示例: 发送 01 4C AE 01 原点回零参数 6B, 正确返回 01 4C 02 6B,

错误命令返回 01 00 EE 6B

数据解析: 修改原点回零参数需要按照下面的数据格式进行排列和数据的转换:

比如, 发送如下数据对原点回零参数进行修改:

01 4C AE 01 00 00 00 1E 00 00 27 10 OF A0 03 20 00 3C 00 6B

0x01 = 保存本次修改的配置参数;

0x00 = 回零模式为 Nearest (单圈就近回零模式) ;

0x00 = 回零方向为 CW;

0x001E = 回零速度为 30 (RPM);

0x00002710 = 回零超时时间为 10000 (ms);

0x0FA0 = 无限位碰撞回零检测转速为 4000 (RPM);

0x0320 = 无限位碰撞回零检测电流为 800 (mA);

0x003C = 无限位碰撞回零检测时间为 60 (ms);

0x00 = 不使能上电自动触发回零功能;

➤ 命令功能：读取回零状态标志位

命令格式：地址 + 0x3B + 校验字节

命令返回：地址 + 0x3B + 回零状态标志+ 校验字节

命令示例：发送 01 3B 6B，正确返回 01 3B 回零状态标志 6B，错误命令返回 01 00 EE 6B

数据解析：返回的回零状态标志字节的每一位都代表一种状态，比如，返回的回零状态标志字节为 0x03，它将按照下面的数据格式进行排列和数据的转换：

编码器就绪状态标志位 = 0x03 & 0x01 = TRUE

校准表就绪状态标志位 = 0x03 & 0x02 = TRUE

正在回零标志位 = 0x03 & 0x04 = FALSE

回零失败标志位 = 0x03 & 0x08 = FALSE

通讯控制位置角度精度选项 (S_PosTDP 菜单) = 0x03 & 0x80 = FALSE

7.4.3 触发动作命令列表

➤ 命令功能：触发编码器校准

命令格式：地址 + 0x06 + 0x45 + 校验字节

命令返回：地址 + 0x06 + 状态码 + 校验字节

命令示例：发送 01 06 45 6B，正确返回 01 06 02 6B，条件不满足返回 01 06 E2 6B，
错误命令返回 01 00 EE 6B（条件不满足的情况：当前是开环模式、触发了堵转保护）

➤ 命令功能：将当前位置角度清零

命令格式：地址 + 0x0A + 0x6D + 校验字节

命令返回：地址 + 0x0A + 状态码 + 校验字节

命令示例：发送 01 0A 6D 6B，正确返回 01 0A 02 6B，错误命令返回 01 00 EE 6B

➤ 命令功能：解除堵转保护

命令格式：地址 + 0x0E + 0x52 + 校验字节

命令返回：地址 + 0x0E + 状态码 + 校验字节

命令示例：发送 01 0E 52 6B，正确返回 01 0E 02 6B，错误命令返回 01 00 EE 6B

➤ 命令功能：恢复出厂设置

命令格式：地址 + 0x0F + 0x5F + 校验字节

命令返回：地址 + 0x0F + 状态码 + 校验字节

命令示例：发送 01 0F 5F 6B，正确返回 01 0F 02 6B，错误命令返回 01 00 EE 6B，
触发恢复出厂设置后，蓝灯亮起，需要断电重新上电校准编码器。

7.4.4 读取参数命令列表

- 命令功能：读取固件版本和硬件版本

命令格式：地址 + 0x1F + 校验字节

命令返回：地址 + 0x1F + 固件版本号 + 硬件版本号 + 校验字节

命令示例：发送 01 1F 6B，正确返回 01 1F 00 C9 00 78 6B，错误命令返回 01 00 EE 6B

数据解析：固件版本 = 0xC9 = 201 = ZDT_X57_V2.0.1，硬件版本 = 0x78 = 120 = ZDT_X57_V1.2

- 命令功能：读取相电阻和相电感

命令格式：地址 + 0x20 + 校验字节

命令返回：地址 + 0x20 + 相电阻 + 相电感 + 校验字节

命令示例：发送 01 20 6B，正确返回 01 20 04 7A 0D 28 6B，错误命令返回 01 00 EE 6B

数据解析：相电阻 = 0x047A = 1146mΩ，相电感 = 0x0D28 = 3368uH（注意单位）

- 命令功能：读取速度环和位置环 PID 参数

命令格式：地址 + 0x21 + 校验字节

命令返回：地址 + 0x21 + 速度环和位置环 PID 参数 + 校验字节

命令示例：发送 01 21 6B，正确返回 01 21 00 01 EE B0 00 07 1E D0 00 00 3C F0 00 00 00 1A 6B，错误命令返回 01 00 EE 6B

数据解析：

梯形曲线位置模式位置环 $K_p = \text{0x0001EEB0} = 126640;$

直通限速位置模式位置环 $K_p = \text{0x00071ED0} = 466640;$

速度环 $K_p = \text{0x00003CF0} = 15600;$

速度环 $K_i = \text{0x0000001A} = 26;$

➤ 命令功能：读取总线电压

命令格式：地址 + 0x24 + 校验字节

命令返回：地址 + 0x24 + 总线电压 + 校验字节

命令示例：发送 01 24 6B，正确返回 01 24 5C 6A 6B，错误命令返回 01 00 EE 6B

数据解析：总线电压 = 0x5C6A = 23658mV（输入电压经过反接二极管后会有压降）

➤ 命令功能：读取总线平均电流

命令格式：地址 + 0x26 + 校验字节

命令返回：地址 + 0x26 + 总线平均电流 + 校验字节

命令示例：发送 01 26 6B，正确返回 01 26 00 80 6B，错误命令返回 01 00 EE 6B

数据解析：总线平均电流 = 0x0080 = 128mA

➤ 命令功能：读取相电流

命令格式：地址 + 0x27 + 校验字节

命令返回：地址 + 0x27 + 总线相电流 + 校验字节

命令示例：发送 01 27 6B，正确返回 01 27 02 73 6B，错误命令返回 01 00 EE 6B

数据解析：相电流 = 0x0273 = 627mA

➤ 命令功能：读取编码器原始值

命令格式：地址 + 0x29 + 校验字节

命令返回：地址 + 0x29 + 编码器原始值 + 校验字节

命令示例：发送 01 29 6B，正确返回 01 29 26 72 6B，错误命令返回 01 00 EE 6B

数据解析：编码器原始值 = 0x2672 = 9842

(注：编码器原始值一圈的数值是 0-16383，即 0-16383 表示 0-360°)

➤ 命令功能：读取实时脉冲数

命令格式：地址 + 0x30 + 校验字节

命令返回：地址 + 0x30 + 符号 + 电机实时位置 + 校验字节

命令示例：发送 01 30 6B，正确返回 01 30 01 00 00 0C 80 6B，错误命令返回 01 00 EE 6B

数据解析：电机实时脉冲数 = -0x00000C80 = -3200 个 (注：00 为正，01 为负)

➤ 命令功能：读取经过线性化校准后的编码器值

命令格式：地址 + 0x31 + 校验字节

命令返回：地址 + 0x31 + 读取经过线性化校准后的编码器值 + 校验字节

命令示例：发送 01 31 6B，正确返回 01 31 8D 9E 6B，错误返回 01 00 EE 6B

数据解析：经过线性化校准后的编码器值 = 0x8D9E = 36254

(注：经过线性化校准后，内部对编码器值进行了 4 倍频，一圈的数值是 0-65535)

➤ 命令功能：读取输入脉冲数

命令格式：地址 + 0x32 + 校验字节

命令返回：地址 + 0x32 + 符号 + 电机实时位置 + 校验字节

命令示例：发送 01 32 6B，正确返回 01 32 01 00 00 0C 80 6B，错误命令返回 01 00 EE 6B

数据解析：输入脉冲数 = -0x00000C80 = -3200 个（注：00 为正，01 为负）

➤ 命令功能：读取电机目标位置

命令格式：地址 + 0x33 + 校验字节

命令返回：地址 + 0x33 + 符号 + 电机目标位置 + 校验字节

命令示例：发送 01 33 6B，正确返回 01 33 01 00 00 0E 10 6B，错误命令返回 01 00 EE 6B

数据解析：电机目标位置 = -0x00000E10 = -3600 = -360.0°

（注：00 为正，01 为负，并且返回的位置角度经过了 10 倍放大，所以是-360.0°）

➤ 命令功能：读取电机实时设定的目标位置

命令格式：地址 + 0x34 + 校验字节

命令返回：地址 + 0x34 + 符号 + 电机目标位置 + 校验字节

命令示例：发送 01 34 6B，正确返回 01 34 01 00 00 0E 10 6B，错误命令返回 01 00 EE 6B

数据解析：电机实时设定的目标位置 = -0x00000E10 = -3600 = -360.0°

（注：00 为正，01 为负，并且返回的位置角度经过了 10 倍放大，所以是-360.0°）

➤ 命令功能：读取电机实时转速

命令格式：地址 + 0x35 + 校验字节

命令返回：地址 + 0x35 + 符号 + 电机实时转速 + 校验字节

命令示例：发送 01 35 6B，正确返回 01 35 01 4E 20 6B，错误命令返回 01 00 EE 6B

数据解析：电机实时转速 = -0x4E20 = -20000 = -2000.0RPM (转/每分钟)

(注：00 为正，01 为负，并且返回的速度经过了 10 倍放大，所以是-2000.0RPM)

➤ 命令功能：读取电机实时位置

命令格式：地址 + 0x36 + 校验字节

命令返回：地址 + 0x36 + 符号 + 电机实时位置 + 校验字节

命令示例：发送 01 36 6B，正确返回 01 36 01 00 00 1C 19 6B，错误命令返回 01 00 EE 6B

数据解析：电机实时位置 = -0x00001C19 = -7193 = -719.3°

(注：00 为正，01 为负，并且返回的位置角度经过了 10 倍放大，所以是-719.3°)

➤ 命令功能：读取电机位置误差

命令格式：地址 + 0x37 + 校验字节

命令返回：地址 + 0x37 + 符号 + 电机位置误差 + 校验字节

命令示例：发送 01 37 6B，正确返回 01 37 01 00 00 00 08 6B，错误命令返回 01 00 EE 6B

数据解析: 电机位置误差 = $-0x00000008 = -8 = -0.08^\circ$

(注: 00 为正, 01 为负, 并且返回的位置角度经过了 100 倍放大, 所以是 -0.08°)

➤ **命令功能:** 读取驱动器实时温度

命令格式: 地址 + 0x39 + 校验字节

命令返回: 地址 + 0x39 + 符号 + 电机实时温度 + 校验字节

命令示例: 发送 01 39 6B, 正确返回 01 39 00 21 6B, 错误命令返回 01 00 EE 6B

数据解析: 驱动器实时温度 = 0x21 = 33 = 33° (注: 00 为正, 01 为负)

➤ **命令功能:** 读取电机状态标志位

命令格式: 地址 + 0x3A + 校验字节

命令返回: 地址 + 0x3A + 电机状态标志+ 校验字节

命令示例: 发送 01 3A 6B, 正确返回 01 3A 电机状态标志 6B, 错误命令返回 01 00 EE 6B

数据解析: 返回的电机状态标志字节的每一位都代表一种状态, 比如, 返回的电机状态标志字节为 0x03, 它将按照下面的数据格式进行排列和数据的转换:

电机使能状态标志位 = $0x03 \& 0x01 = 0x01 = \text{TRUE}$

电机到位标志位 = $0x03 \& 0x02 = 0x02 = \text{TRUE}$

电机堵转标志位 = $0x03 \& 0x04 = 0x00 = \text{FALSE}$

电机堵转保护标志 = $0x03 \& 0x08 = 0x00 = \text{FALSE}$

(注: 某个标志如果不为 0 表示为 TRUE, 等于 0 则表示 FALSE)

7.4.5 修改参数命令列表

➤ 命令功能：修改细分

命令格式：地址 + 0x84 + 0x8A + 是否存储标志 + 细分值 + 校验字节

命令返回：地址 + 0x84 + 命令状态 + 校验字节

命令示例：发送 01 84 8A 01 07 6B，正确返回 01 84 02 6B，错误命令返回 01 00 EE 6B

数据解析：修改细分为 $0x07 = 7$ 细分，并存储到芯片上；

(注：00 表示 256 细分，如果要修改为 256 细分，则发送 01 84 8A 01 00 6B)

➤ 命令功能：修改 ID 地址

命令格式：地址 + 0xAE + 0x4B + 是否存储标志 + ID 地址 + 校验字节

命令返回：地址 + 0xAE + 命令状态 + 校验字节

命令示例：发送 01 AE 4B 01 10 6B，正确返回 01 AE 02 6B，错误命令返回 01 00 EE 6B

数据解析：修改 ID 地址为 $0x10 = 16$ ，并存储到芯片上；

➤ 命令功能：修改速度环和位置环 PID 参数

命令格式：地址 + 0x4A + 0xC3 + 是否存储标志 + 位置环速和度环 PID 参数 + 校验字节

命令返回：地址 + 0x4A + 命令状态 + 校验字节

命令示例：发送 01 4A C3 01 速度环和位置环 PID 参数 6B，正确返回 01 4A 02 6B，错误命令返回 01 00 EE 6B

数据解析：修改速度环和位置环 PID 参数需要按照下面的数据格式进行排列和数据的转换：

比如，发送如下数据对速度环和位置环的 PID 参数进行修改：

01 4A C3 00 00 01 EE B0 00 07 1E D0 00 00 3C F0 00 00 00 1A 6B

0x00 = 不保存本次修改的配置参数；

0x0001EEB0 = 修改梯形曲线位置模式位置环 Kp 为 126640；

0x00071ED0 = 修改直通位置限速模式位置环 Kp 为 466640；

0x00003CF0 = 修改速度环 Kp 为 15600；

0x0000001A = 修改速度环 Ki 为 26；

八、原点回零操作说明

8.1 单圈就近/方向回零操作说明

单圈回零模式有两种：单圈就近回零和单圈方向回零，顾名思义，单圈就近回零就是往靠近零点位置的方向回零，而单圈方向回零就是根据 0_Dir 选项设定的方向回零，

单圈就近/方向回零操作步骤如下：

1. 断电状态将电机轴移动到指定位置、或上电状态下将电机轴运动到指定位置；
2. 点击菜单 0_Set 中的 Set 0 选项，或者发送单圈零点设置命令 01 93 88 01 6B 设置单圈回零的零点位置；
3. 设置完成后，根据自身需求，可以选择使能上电自动触发回零功能（0_POT_En 菜单设置为 Enable），也可以通讯发送“触发回零”命令进行回零。



8.2 多圈无限位碰撞回零操作说明

原理：利用电机运动过程中，碰撞到机器结构边缘时，电流会瞬间变大，转速会瞬间变小，这一特点，就是堵转检测的原理，简而言之，就是卡一下电机轴进行回零。

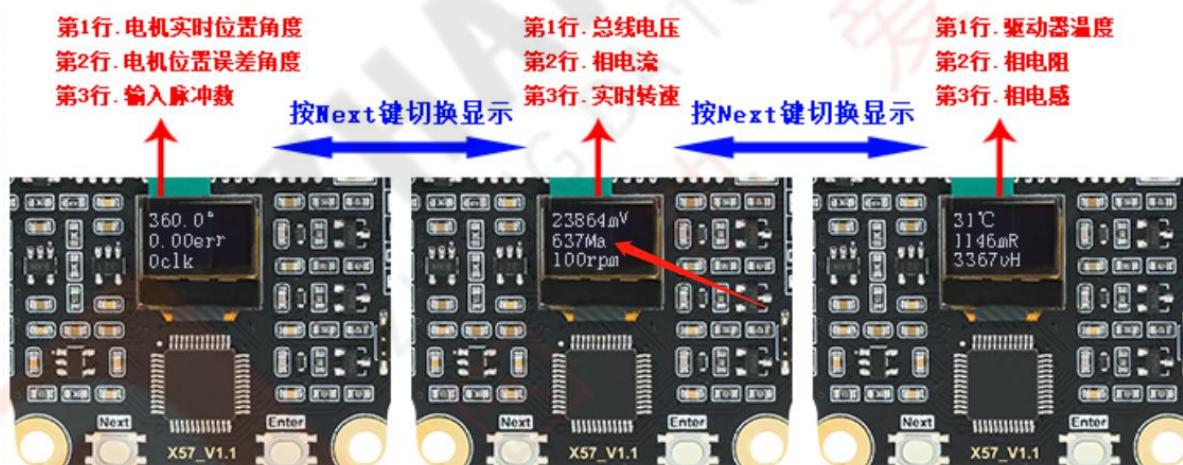
前提条件：电机带的负载固定；

检测判定条件: 电机实时转速 < 碰撞回零检测转速 + 电机实时相电流 > 碰撞回零

检测电流 + 持续时间 > 碰撞回零检测时间 (注: 检测阈值可在上位机中进行修改)

多圈无限位碰撞回零操作步骤如下:

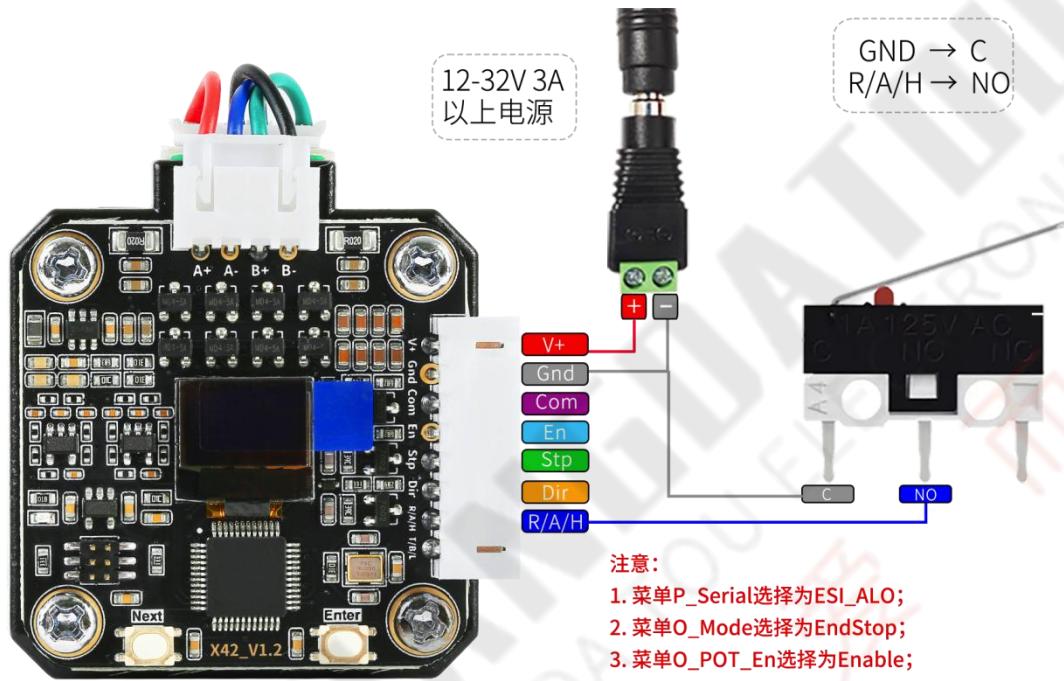
- (1) 先用上位机用直通限速位置模式控制电机带着“固定负载”，以“回零速度”正常运动，并观察小屏幕此时的相电流大概是多少，比如大约 637Ma；
- (2) 修改“无限位碰撞回零检测电流”比 637Ma 略大一些，比如设置为 800Ma；
- (3) 修改“无限位碰撞回零检测转速”可以设置为 300RPM；
- (4) 修改“无限位碰撞回零检测时间”可以设置为 60ms；
- (5) 根据自身需求，可以将“上电自动触发回零功能”进行使能（Enable）；
- (6) 如果上电自动触发无限位回零的方向不对，可在菜单 0_Dir 修改回零方向；



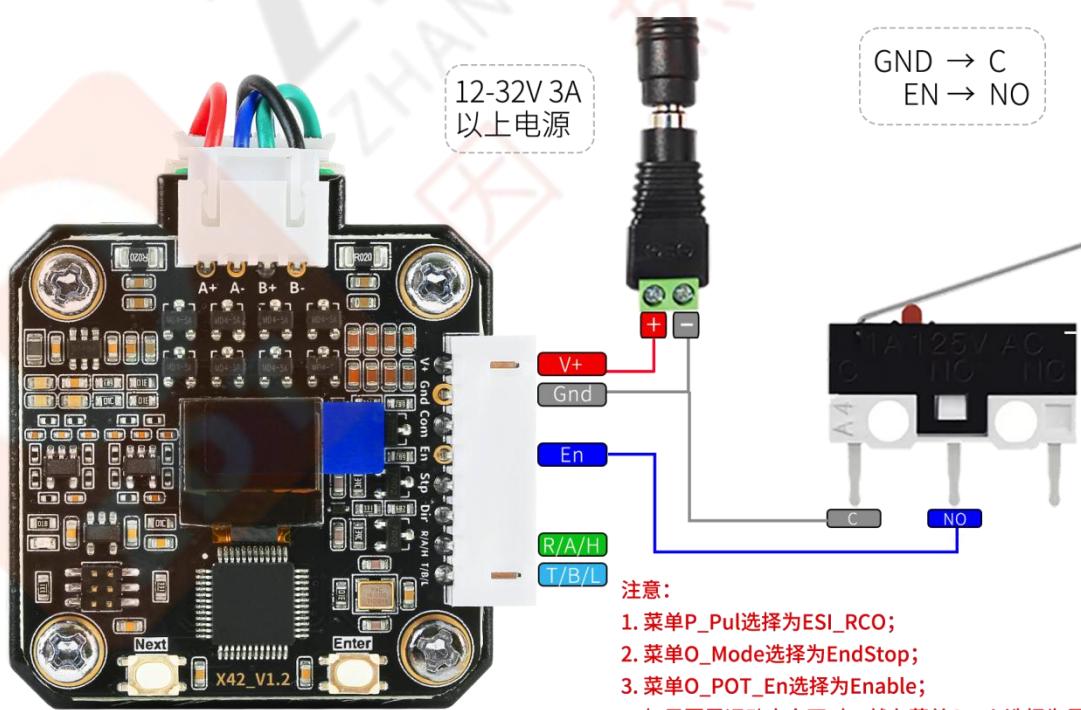
8.3 多圈有限位开关回零操作说明

多圈有限位开关根据自身使用的控制模式按照下图进行接线和设置：

脉冲模式控制-上电自动回零限位开关接线



串口/232/485/CAN 通讯控制-上电自动回零限位开关接线



九、到位输出和报警输出

9.1 通讯控制到位输出/到位返回命令

到位输出

菜单 P_Pul 选择为 ESI_RCO，当电机到位，即输入位置角度 - 电机实时位置角度 < 位置到达窗口（默认 0.3° ，可在上位机上修改）时，Dir 引脚会输出高电平。

（注：如果 Com 不接，则到位后 Dir 引脚到位输出 3.3V；如果 Com 接了，则到位后 Dir 引脚到位输出 Com 电压；到位输出仅适用于 PLC 套餐，也就是不带光耦隔离的套餐。）

到位返回命令（地址 + FB/FD + 9F + 6B）

菜单 Response 可设置串口/RS232/RS485/CAN 总线通讯控制时，主机发送**控制动作命令列表**中的命令时，从机收到命令后，是否返回确认收到命令，以及是否返回到位命令；

None: 不返回确认收到命令，在发送位置模式命令时也不返回到位命令；

Receive: 只返回确认收到命令（默认值）；

Reached: 只在发送位置模式命令时返回到位命令（地址 + FB/FD + 9F + 6B）；

Both: 既返回确认收到命令，也在发送位置模式命令时返回到位命令；

Other: 位置模式下只返回到位命令，其他控制动作命令返回确认收到命令；

9.2 脉冲控制模式报警输出和复位堵转保护

菜单 P_Serial 选择为 ESI_AL0，当闭环电机发生堵转、欠压等异常情况时，T/B/L 引脚会输出 3.3V 高电平，正常情况下默认是 0V 低电平；发生堵转后，将 R/A/H 引脚输入低电平 0V 可以复位堵转保护，不需要重新上电；

十、常见问题及注意事项

10.1 常见问题

序号	问题	解决方法
1	Waiting V+ Power!	供电电压不足, V+和 Gnd 需要 12-32V 进行供电
2	Not Cal	未校准编码器, 可以点击 Cal 进行编码器校准
3	Phase AABB Error!	电机线序 A+ A- 和 B+ B- 错误, 请检查电机线序
4	Magnet Loss! Enter..	检测不到磁铁, 未安装磁铁或驱动到电机上
5	Encoder Error! Enter..	磁编码器通讯失败, 可能编码器芯片已经损坏
6	Ref Voltage Error!	基准电压错误, 可能基准电压芯片已经损坏
7	Bus Current Error!	上电瞬间总线电流过大, 请检查是否短路
8	Origin Set Done!	单圈回零的零点位置设置成功
9	Origin Set Fail!	单圈回零的零点位置设置失败, 检查是否校准编码器
10	Going to Origin..	正在执行回零操作
11	Go to Origin Fail!	上电自动回零失败, 请检查电机力矩是否足够
12	Restore Done! Reboot!	恢复出厂设置成功, 请重新上电校准编码器

10.2 注意事项

1. 请勿带电拔插供电接口线和电机线, 以免造成电气损坏;
2. 本驱动不一定适用所有的 57 电机。
3. 建议搭配本店 57 磁柱步进电机;
4. 到店一起购买 57 步进电机, 会安装并校准好整套发货;

十一、技术支持及保证

1. 发货前都会做通电测试，保证可以正式使用才发货。
2. 欢迎各位朋友加入张大头闭环伺服交流群：262438510
3. 欢迎光临 CSDN 博客交流：<http://blog.csdn.net/zhangdatou666>
4. 有问题可联系我们的客服或技术支持人员，我们将竭诚为您服务。

QQ 交流群：



技术支持 QQ：

424116966

十二、修改记录

日期	版本	内容
2024/1/15	Rev1.0	初版

十三、附录

13.1 checksum 菜单选择 XOR 和 CRC-8 校验的计算方式

XOR 校验：

```
unsigned char command_calXOR(unsigned char *p, unsigned char len)
{
    unsigned char i = 0, xorV = p[0];

    for(i=1; i < len; i++)
    {
        xorV ^= (*p + i);
    }

    return xorV;
}
```

比如，读取实时位置命令，则发送：01 36 37，其中 $0x37 = 0x01 \wedge 0x36$ ；

XCRC-8 校验：

```
const unsigned char crc8Table[256] = {
    0x00, 0x5E, 0xBC, 0xE2, 0x61, 0x3F, 0xDD, 0x83,
    0xC2, 0x9C, 0x7E, 0x20, 0xA3, 0xFD, 0x1F, 0x41,
    0x9D, 0xC3, 0x21, 0x7F, 0xFC, 0xA2, 0x40, 0x1E,
    0x5F, 0x01, 0xE3, 0xBD, 0x3E, 0x60, 0x82, 0xDC,
    0x23, 0x7D, 0x9F, 0xC1, 0x42, 0x1C, 0xFE, 0xA0,
    0xE1, 0xBF, 0x5D, 0x03, 0x80, 0xDE, 0x3C, 0x62,
    0xBE, 0xE0, 0x02, 0x5C, 0xDF, 0x81, 0x63, 0x3D,
    0x7C, 0x22, 0xC0, 0x9E, 0x1D, 0x43, 0xA1, 0xFF,
    0x46, 0x18, 0xFA, 0xA4, 0x27, 0x79, 0x9B, 0xC5,
```

```
0x84, 0xDA, 0x38, 0x66, 0xE5, 0xBB, 0x59, 0x07,
0xDB, 0x85, 0x67, 0x39, 0xBA, 0xE4, 0x06, 0x58,
0x19, 0x47, 0xA5, 0xFB, 0x78, 0x26, 0xC4, 0x9A,
0x65, 0x3B, 0xD9, 0x87, 0x04, 0x5A, 0xB8, 0xE6,
0xA7, 0xF9, 0x1B, 0x45, 0xC6, 0x98, 0x7A, 0x24,
0xF8, 0xA6, 0x44, 0x1A, 0x99, 0xC7, 0x25, 0x7B,
0x3A, 0x64, 0x86, 0xD8, 0x5B, 0x05, 0xE7, 0xB9,
0x8C, 0xD2, 0x30, 0x6E, 0xED, 0xB3, 0x51, 0x0F,
0x4E, 0x10, 0xF2, 0xAC, 0x2F, 0x71, 0x93, 0xCD,
0x11, 0x4F, 0xAD, 0xF3, 0x70, 0x2E, 0xCC, 0x92,
0xD3, 0x8D, 0x6F, 0x31, 0xB2, 0xEC, 0x0E, 0x50,
0xAF, 0xF1, 0x13, 0x4D, 0xCE, 0x90, 0x72, 0x2C,
0x6D, 0x33, 0xD1, 0x8F, 0x0C, 0x52, 0xB0, 0xEE,
0x32, 0x6C, 0x8E, 0xD0, 0x53, 0x0D, 0xEF, 0xB1,
0xF0, 0xAE, 0x4C, 0x12, 0x91, 0xCF, 0x2D, 0x73,
0xCA, 0x94, 0x76, 0x28, 0xAB, 0xF5, 0x17, 0x49,
0x08, 0x56, 0xB4, 0xEA, 0x69, 0x37, 0xD5, 0x8B,
0x57, 0x09, 0xEB, 0xB5, 0x36, 0x68, 0x8A, 0xD4,
0x95, 0xCB, 0x29, 0x77, 0xF4, 0xAA, 0x48, 0x16,
0xE9, 0xB7, 0x55, 0x0B, 0x88, 0xD6, 0x34, 0x6A,
0x2B, 0x75, 0x97, 0xC9, 0x4A, 0x14, 0xF6, 0xA8,
0x74, 0x2A, 0xC8, 0x96, 0x15, 0x4B, 0xA9, 0xF7,
0xB6, 0xE8, 0x0A, 0x54, 0xD7, 0x89, 0x6B, 0x35
};

unsigned char command_calCRC8(unsigned char *p, unsigned char len)
{
    unsigned char i = 0, crc8 = p[0];

    for(i = 1; i < len; i++)

```

```
{  
    crc8 = crc8Table[crc8 ^ (*(p + i))];  
  
}  
  
return crc8;  
}
```

13.2 修改驱动参数命令

➤ 命令功能：修改驱动参数

命令格式：地址 + 0x48 + 0xD1 + 是否存储标志 + 驱动参数 + 校验字节

命令返回：地址 + 0x48 + 命令状态 + 校验字节

命令示例：发送 01 48 D1 01 驱动参数 6B，正确返回 01 48 02 6B，错误命令返回 01 00 EE 6B

数据解析：修改驱动参数需要按照下面的数据格式进行排列和数据的转换：

比如，发送如下数据对驱动参数进行修改：

01 48 D1 01 00 01 01 02 02 00 10 01 00 00 04 B0 08 98 0B B8 03 E8 05 07 00
01 00 01 00 08 07 D0 07 D0 00 03 6B

0x01 = 保存本次修改的配置参数；

0x00 = 修改锁定按键菜单 Lock 为 Disable (0x01 为 Enable)；

0x01 = 修改控制模式菜单 Ctrl_Mode 为 CR_VFOC，即 FOC 矢量闭环控制模式；

0x01 = 修改脉冲端口复用功能菜单 P_PUL 为 PUL_ENA，即使能脉冲输入控制；

0x02 = 修改通讯端口复用功能菜单 P_Serial 为 UART_FUN，即使能串口通讯；

0x02 = 修改 En 引脚的有效电平菜单 En 为 Hold，即一直有效；

0x00 = 修改电机旋转正方向菜单 Dir 为 CW，即顺时针方向；

- 0x10 = 修改细分菜单 MStep 为 16 细分；(注：256 细分用 00 表示)；
- 0x01 = 修改细分插补功能菜单 MPlyer 为 Enable，即使能细分插补；
- 0x00 = 修改自动熄屏功能菜单 AutoSDD 为 Disable，即关闭自动熄屏功能；
- 0x00 = 修改采样电流低通滤波器强度菜单 LPFilter 为 Def，即默认强度。
- 0x04B0 = 修改开环模式工作电流菜单 Ma 为 1200Ma；
- 0x0898 = 修改闭环模式最大电流菜单 Ma_Limit 为 2200Ma；
- 0x0BB8 = 修改闭环模式最大转速菜单 Vm_Limit 为 3000RPM (转/每分钟)；
- 0x03E8 = 修改电流环带宽菜单 CurBW_Hz 为 1000rad/s；
- 0x05 = 修改串口波特率菜单 UartBaud 为 115200；(对应小屏幕选项顺序)
- 0x07 = 修改 CAN 通讯速率菜单 CAN_Baud 为 500000；(对应小屏幕选项顺序)
- 0x00 = 修改通讯校验方式菜单 Checksum 为 0x6B；
- 0x01 = 修改控制命令应答菜单 Response 为 Receive，即只返回确认收到命令；
- 0x00 = 修改通讯控制输入角度精确度选项菜单 S_PosTDP 为 Disable；
- 0x01 = 修改堵转保护功能菜单 Clog_Pro 为 Enable，即使能堵转保护；
- 0x0008 = 修改堵转保护转速阈值菜单 Clog_Rpm 为 8RPM (转/每分钟)；
- 0x07D0 = 修改堵转保护电流阈值菜单 Clog_Ma 为 2000Ma；
- 0x07D0 = 修改堵转保护检测时间阈值菜单 Clog_Ms 为 2000ms；
- 0x0003 = 修改位置到达窗口为 0.3°；
(注：表示当目标位置角度与传感器实时位置角度相差小于 0.3° 时，认为电机已经到达设定的位置，将置位位置到达标志位)

13.3 读取驱动参数命令

➤ 命令功能：读取驱动参数

命令格式：地址 + 0x42 + 0x6C + 校验字节

命令返回：地址 + 0x42 + 驱动参数 + 校验字节

命令示例：发送 01 42 6C 6B，正确返回 01 42 驱动参数 6B，错误命令返回 01 00 EE 6B

数据解析：命令正确返回 01 42 后面的字节就是驱动参数，它将按照下面的数据格式进行排列和数据的转换：

比如，发送读取驱动参数 01 42 6C 6B 命令后，返回如下数据：

01 42 25 18 00 01 01 02 02 00 10 01 00 00 04 B0 08 98 0B B8 03 E8 05 07 00
01 00 01 00 08 07 D0 07 D0 00 03 6B

0x25 = 返回命令所包含的字节数 = 37 个；

0x18 = 返回命令的配置参数个数 = 24 个；

0x00 = 锁定按键菜单 Lock 为 Disable (0x01 为 Enable)；

0x01 = 控制模式菜单 Ctrl_Mode 为 CR_VFOC，即 FOC 矢量闭环控制模式；

0x01 = 脉冲端口复用功能菜单 P_PUL 为 PUL_ENA，即使能脉冲输入控制；

0x02 = 通讯端口复用功能菜单 P_Serial 为 UART_FUN，即使能串口通讯；

0x02 = En 引脚的有效电平菜单 En 为 Hold，即一直有效；

0x00 = 电机旋转正方向菜单 Dir 为 CW，即顺时针方向；

0x10 = 细分菜单 MStep 为 16 细分；(注：256 细分用 00 表示)；

0x01 = 细分插补功能菜单 MPlyer 为 Enable，即使能细分插补；

0x00 = 自动熄屏功能菜单 AutoSDD 为 Disable，即关闭自动熄屏功能；

0x00 = 采样电流低通滤波器强度菜单 LPFilter 为 Def；

0x04B0 = 开环模式工作电流菜单 Ma 为 120 0Ma；

0x0898 = 闭环模式最大电流菜单 Ma_Limit 为 2200Ma；

0x0BB8 = 闭环模式最大转速菜单 Vm_Limit 为 3000RPM（转/每分钟）；

0x03E8 = 电流环带宽菜单 CurBW_Hz 为 1000rad/s；

0x05 = 串口波特率菜单 UartBaud 为 115200；（对应小屏幕选项顺序）

0x07 = CAN 通讯速率菜单 CAN_Baud 为 500000；（对应小屏幕选项顺序）

0x00 = 通讯校验方式菜单 Checksum 为 0x6B；

0x01 = 控制命令应答菜单 Response 为 Receive，即只返回确认收到命令；

0x00 = 通讯控制输入角度精确度选项菜单 S_PosTDP 为 Disable；

0x01 = 堵转保护功能菜单 Clog_Pro 为 Enable，即使能堵转保护；

0x0008 = 堵转保护转速阈值菜单 Clog_Rpm 为 8RPM（转/每分钟）；

0x07D0 = 堵转保护电流阈值菜单 Clog_Ma 为 2000Ma；

0x07D0 = 堵转保护检测时间阈值菜单 Clog_Ms 为 2000ms；

0x0003 = 位置到达窗口为 0.3°；

（注：表示当目标位置角度与传感器实时位置角度相差小于 0.3° 时，认为电机已经到达设定的位置，将置位位置到达标志位）

13.4 读取系统状态参数命令

➤ 命令功能：读取系统状态参数

命令格式：地址 + 0x43 + 0x7A + 校验字节

命令返回：地址 + 0x43 + 系统状态参数 + 校验字节

命令示例：发送 01 43 7A 6B，正确返回 01 43 系统状态参数 6B，错误命令返回 01 00 EE 6B

数据解析：命令正确返回 01 43 后面的字节就是系统的状态参数，它将按照下面的数据格式进行排列和数据的转换：

比如，发送读取系统状态参数 01 43 7A 6B 命令后，返回如下数据：

01 43 27 0C 5C 67 00 00 00 03 30 B5 43 EB 01 00 00 34 2E 00 00 00 01 00 00
34 2E 01 00 00 00 01 00 22 03 03 6B

返回命令所包含的字节数 = 0x25 = 37 个；

返回命令的配置参数个数 = 0x0C = 12 个；

总线电压 = 0x5C67 = 23655mV；

总线电流 = 0x0000 = 0mA；

电机相电流 = 0x0003 = 3mA；

编码器原始值 = 0x30B5 = 12469；

校准后编码器值 = 0x43EB = 17387；

电机目标位置 = -0x0000342E = -13358 = -1335.8°；（位置值前面的 01 表示符号）

电机实时转速 = 0x0000 = 0.0RPM（转/每分钟）；（转速值前面的 01 表示符号）

电机实时位置 = -0x0000342E = -13358 = -1335.8°；（位置值前面的 01 表示符号）

电机位置误差 = `-0x00000001` = -1 = -0.01° ; (位置误差值前面的 01 表示符号)

电机实时温度 = `0x22` = 34°C ; (温度值前面的 00 表示符号)

就绪状态标志 = `0x03`;

- 编码器就绪状态标志位 = `0x03 & 0x01` = TRUE = 就绪;
- 校准表就绪状态标志位 = `0x03 & 0x02` = TRUE = 就绪;
- 正在回零标志位 = `0x03 & 0x04` = FALSE = 当前没有回零;
- 回零失败标志位 = `0x03 & 0x08` = FALSE = 没有回零失败;
- 通讯控制位置角度精度选项(S_PosTDP 菜单) = `0x03 & 0x80` = Disable

电机状态标志 = `0x03`;

- ◆ 使能状态标志位 = `0x03 & 0x01` = TRUE = 电机处于使能状态;
- ◆ 电机到位标志位 = `0x03 & 0x02` = TRUE = 电机已经到位;
- ◆ 电机堵转标志位 = `0x03 & 0x04` = FALSE = 没触发堵转;
- ◆ 堵转保护标志位 = `0x03 & 0x08` = FALSE = 没触发堵转保护;

(注: 某个标志如果不为 0 表示为 TRUE, 等于 0 则表示 FALSE)