

SVD48 系列

伺服驱动器

使用说明书

版本：V1.00

版权所有 不得翻印
【使用前请仔细阅读本手册，以免损坏驱动器】

目录

第一章 产品简介及注意事项	4
1.1 产品简介	4
1.2 特点	4
1.3 应用领域	4
1.4 确认事项（确认包装及零配件）	4
1.5 运输和存储条件	5
1.6 技术要求	5
1.7 操作人员要求	5
1.8 环境要求	6
第二章 系统配置和型号说明	7
第三章 尺寸安装	8
第四章 系统接口与配线	12
4.1 驱动器各部分名称与功能	12
4.2 外部接线图	18
第五章 FULLING-tech 上位机使用指南	19
5.1 驱动器与上位机的连接	21
5.2 电机参数以及控制参数管理	22
5.3 简易调试	24
5.4 IO 口设置	26
5.5 触发式示波器使用	29
5.6 历史错误与报警	32
5.7 驱动器参数读写	34
5.8 固件下载	35
第六章 工作模式介绍	36
6.1 速度模式介绍	36
6.2 位置模式 (1)	40
6.3 力矩模式 (4)	44
6.4 脉冲模式介绍(-4)	46
6.5 原点模式 (6)	47
第七章 性能调节	59
7.1 速度环整定	59
7.2 位置环整定	63
7.3 综合调整	64
第八章 报警排除	65
8.1 驱动器 LED 灯报警排除	65
8.2 代码报警排除 (603F00)	65
8.3 错误字报警排除	68
第九章 常用对象列表	72
9.1 常用对象列表	72
9.2 单位转换	85
第十章 UART 通讯	87
10.1 UART 通讯格式	87

10.2	UART 通讯实例	88
第十一章	RS485 通讯	92
11.1	RS485 通讯硬件介绍	92
11.2	RS485 通讯格式	92
11.3	RS485 通讯实例	93
第十二章	CANopen 通讯	99
12.1	硬件说明	99
12.2	CAN 通讯	100
12.3	CAN 通讯实例	107
附录一	配置第三方电机	116
附录二	制动电阻使用	120
附录三	保险丝规格选择	121

说明书版本修订记录

日期	更新内容
2024. 10. 07	第一版

第一章 产品简介及注意事项

1.1 产品简介

SVD48 系列伺服驱动器是常州富兴机电有限公司自主研发的一款高性价比伺服电机驱动器。它采用 32 位基于 ARM 核心的带 512K 字节闪存微控制器，集成度高，拥有完善的保护措施。该系列驱动器采用新型的控制技术，具有体积小、性能好、稳定性高等特点。

1.2 特点

- 采用 ARM32 位 Cortex™-M3 内核芯片；
- 电压范围 20V-56V，支持宽电压输入；
- 支持增量式差分编码器 (5V)、多摩川协议通讯式编码器；
- 支持外接制动电阻；
- 支持串口通信、RS485 通信和 CAN 通信；
- 支持输入配置（驱动器使能，报警复位，紧急停止，正限位，负限位，多段速控制，多段位置控制等）；
- 支持输出配置（驱动器就绪，驱动器错误，电机零速，电机抱闸生效，限位生效等）；
- 具有过压保护、欠压保护、电机过热（I²T）保护、短路保护等驱动器保护；
- 超高性价比

1.3 应用领域

- 物流机器人：自动导航货运机器人，穿梭车，自动停车机器人等；
- 物流设备：全自动分拣线，立体仓库等；
- 医疗设备行业：小型系统；
- 其他需要高响应速度，高定位精度的场合。

1.4 确认事项（确认包装及零配件）

- 确认货物包装是否完好无损，无明显的破损、变形或湿润。
- 确认收到货物是否与您订购的产品型号、数量和规格相符。
- 确认驱动器和电机的外观是否完好无损，无明显的划痕、裂纹或锈蚀。
- 确认收到货物是否附有驱动器和电机的使用说明书、质保书、合格证等相关文件

表 1-1 SVD4812RC-AA 货物清单

货物清单			
名称	规格型号	图示	数量
驱动器	SVD4812RC-AA		1
插拔式连接器	UL0-TB13-15K/3.81-02P-400HA		1
插拔式连接器	UL0-TB27-2K/5.08-02P-4000C		1
插拔式连接器	UL0-TB27-2K/5.08-04P-4000C		1

插拔式连接器	15EDGKNH-3.5-7P		1
静电袋	140*200mm		1
飞机盒	17*9*4cm		1

表 1-1 SVD4822RC-AA 货物清单

货物清单			
名称	规格型号	图示	数量
驱动器	SVD4822RC-AA		1
插拔式连接器	15EDGKNH-3.5-7P		1
插拔式连接器	ULO-TB13-15K/3.81-02P-4000A		1
插拔式连接器	ULO-TB27-2K/5.08-02P-4000C		1
静电袋	180*260mm		1
飞机盒	21*12*4cm		1

1.5 运输和存储条件

- 运输前，应检查驱动器和电机的外观是否完好，无明显损坏或变形，电缆线是否牢固，无断裂或磨损。
- 运输时，应使用专用的包装箱或托盘，避免直接接触地面或其他物体，防止受到撞击或挤压。包装箱或托盘上应贴有明显的标识，如“易碎品”、“注意防潮”、“勿倒置”等。运输过程中，应轻拿轻放，避免剧烈震动或颠簸，尽量保持水平状态，不要倾斜或倒置。
- 运输后，应及时将驱动器和电机存放在干燥、通风、无腐蚀性气体的环境中，避免阳光直射或温度过高。存放时，应保持包装箱或托盘的完整性，不要堆放过高或过重，防止变形或压坏。

1.6 技术要求

为了正确和安全地使用本产品，您需要注意以下方面：

本产品的操作方法和注意事项，以及可能出现的风险和预防措施，都已经在本文档中提供了清晰的说明和警示信息。您必须仔细阅读并遵守这些信息，确保操作过程符合安全标准。如果您忽视这些信息，可能会造成人身伤害或财产损失。因此，在操作本产品之前，请务必认真学习并掌握本文档中的内容。

1.7 操作人员要求

安全意识： 操作人员必须具备良好的安全意识，遵循操作规程和安全准则，以确保自身和设备的安全。

技术知识： 对于复杂的驱动器和电机系统，操作人员需要具备一定的技术知识，能够理解和解决一些常见的故障和问题。

遵循操作指南： 操作人员应遵循生产商提供的使用说明书，正确设置和调整参数，以实现预期的运行效果。

避免操作错误： 操作人员应避免误操作或不当操作，如过载、逆转等，以免损坏设备或造成安

全风险。

紧急情况处理： 操作人员需要知道如何应对紧急情况，包括停止设备、切断电源等，以防止事故发生。

1.8 环境要求

表 1-2 环境要求

环境	条件
工作温度	0℃~40℃
工作湿度	5~95%RH（无凝露）
储藏温度	-10℃~70℃（不结冰）
储藏湿度	90%RH 以下（无凝露）
保护等级	IP20
安装场所	室内无日晒、无腐蚀性气体、无易燃性气体、无油气、无尘埃
安装方式	垂直安装或水平安装
大气压力	85kpa~105kpa
高度	额定工作海拔 1000 米以下，工作海拔在 1000 米以上时，每上升 100 米，需降额 1.5%使用，最大工作高度海拔 4000 米

第二章 系统配置和型号说明

SVD 4812 空/RC/EC - AA - XXX

① ② ③ ④ ⑤

① 驱动器系列名：SVD 系列

② 额定电压，额定电流
48V / 12A

③ 通讯类型
RC：RS-485 通讯+CAN 通讯
EC：EtherCAT 通讯+CAN 通讯
ER：EtherCAT 通讯+RS-485 通讯

④ 编码器类型
AA：UVW 差分+ABZ 差分、多摩川编码器、支持 UVW+ AB 差分
HD：UVW 差分+ABZ 差分
CT：多摩川编码器
HN：仅支持 UVW 信号
HE：支持 UVW+ AB 差分
CA：自定义通讯式磁编码器

⑤ 序列号

此代码作为驱动器出厂默认功能配置代码，包括电机参数，控制环参数，I/O 功能，通讯功能参数等配置。其中很多配方来自客户的现场应用需求，目的是方便客户使用，尽可能的做到客户拿到驱动器即可以满足其现场使用，而无需再配置相关参数。

第三章 尺寸安装

3.1 伺服驱动器安装尺寸

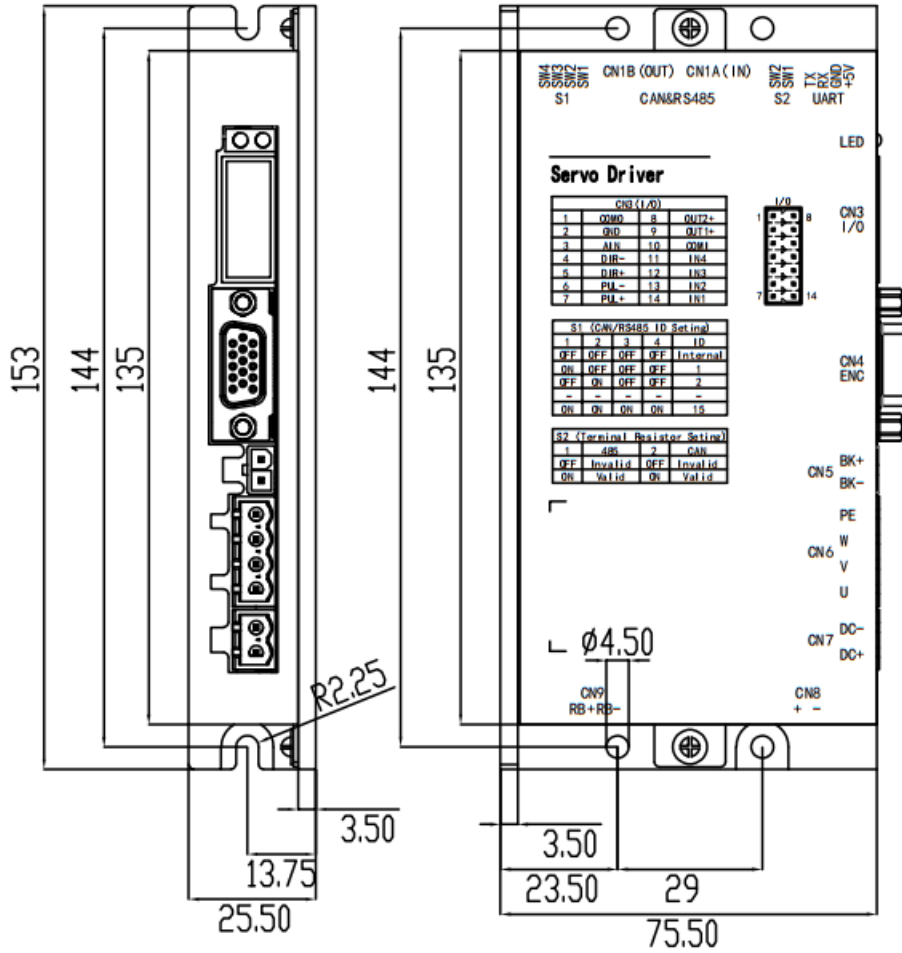


图 3-1 SVD4812RC-AA 驱动器安装尺寸

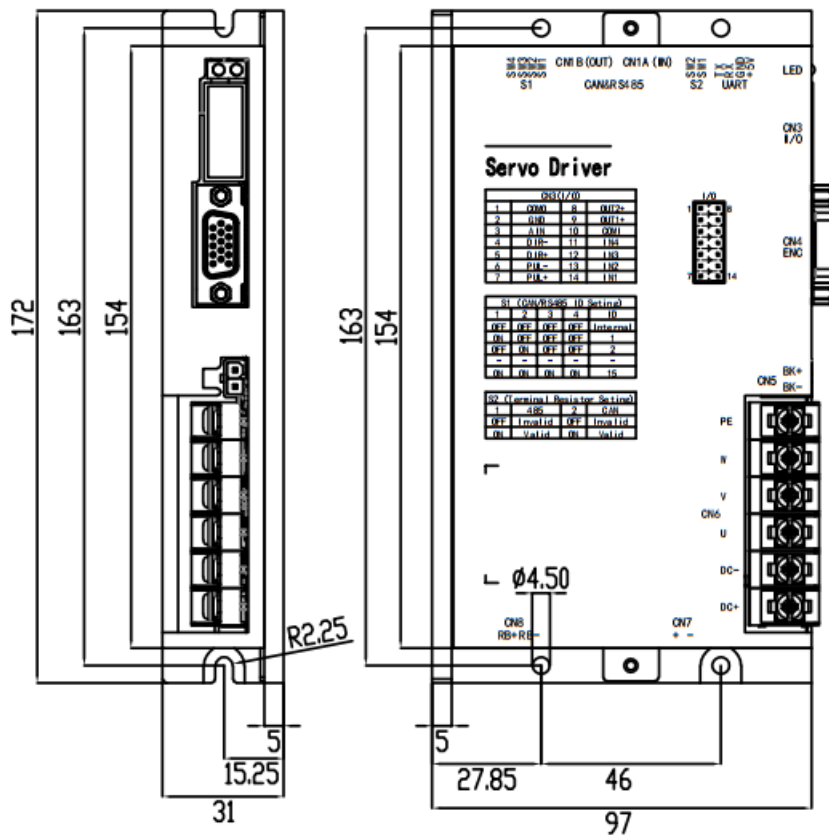


图 3-2 SVD4822RC-AA 驱动器安装尺寸

3.2 伺服驱动器安装注意事项

下面是安装驱动器的一些注意事项：

位置和安装： 驱动器应该安装在干燥、通风良好的环境中，远离潮湿、尘埃和腐蚀性气体。也要避免在高温、阳光直射、震动频繁的地方安装。

固定和连接： 确保将驱动器牢固地安装在稳定的支架或表面上。所有的螺丝和连接件都应该被适当地锁紧，以避免松动和振动。连接电缆时，确保电缆插头和插座牢固，避免拉扯电缆。

散热： 如果使用了发热性器件，如制动电阻，需要考虑散热问题。确保驱动器周围有足够的散热空间，不要堆积其他物体，以免影响散热效果。安装驱动器时，可以按照例图 3-2 所示距离进行安装

电气连接： 确保正确连接电源线和电机线。遵循正确的接线图或指导手册。确保电源符合要求，电压稳定。注意接地和绝缘问题，以避免电气问题。

保护措施： 安装驱动器时，应该避免异物进入驱动器内部，特别是金属屑、灰尘、油等导电或易燃物质。驱动器和电机都是精密设备，避免受到外力冲击和振动。

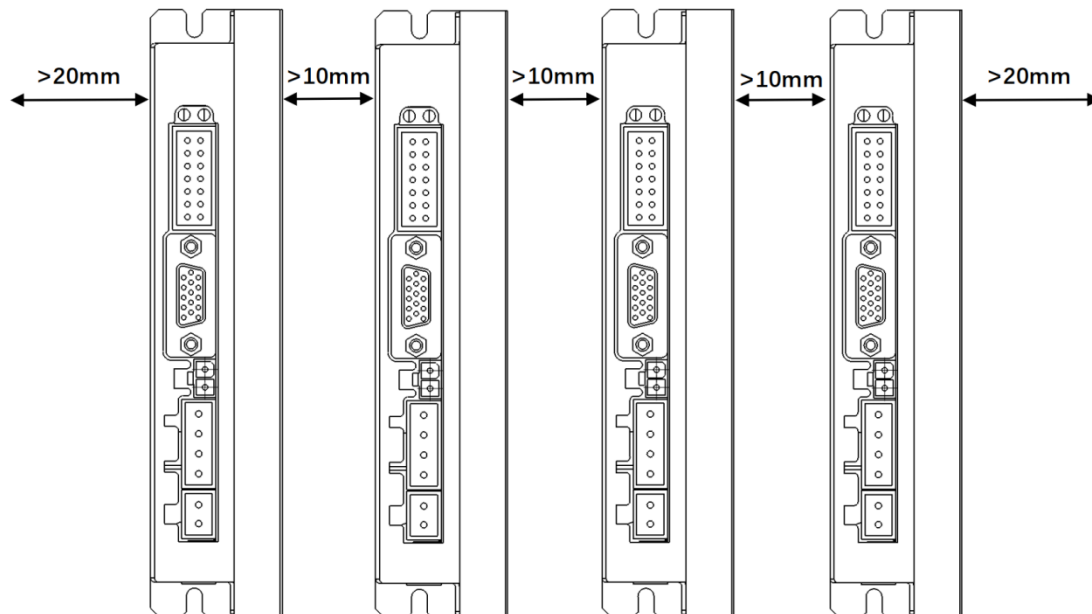
检查和测试： 在安装完成后，进行必要的检查和测试，确保驱动器的连接正确，功能正常，之后可以按照说明书进行对应的测试。

EMC 要求： 如果驱动器将用于工业电磁环境，需要遵循电磁兼容性（EMC）的要求。可能需要安装电源滤波器、屏蔽电缆等以减少干扰和电磁辐射。

安全和维护： 在安装完毕后，要确保有适当的安全措施，以防止未经授权的人员接近驱动器。

定期检查驱动器和连接，确保没有松动或损坏。

请在安装前详细阅读相关的驱动器安装手册和说明书，遵循制造商提供的指导和建议。如果有任何不确定的地方，建议咨询公司提供技术支持。



图片 3-3 推荐安装距离示例图

3.3 伺服驱动器相关参数

驱动器型号	SVD4812RC-AA	SVD4822RC-AA
支持电机类型	伺服电机	
输入电压范围	20~56V	
持续电流	12Arms (不加辅助散热) 15arms (加辅助散热)	22Arms (不加辅助散热) 30arms (加辅助散热)
峰值电流	45Ap	100Ap
反馈信号	增量式差分编码器 (5V)、多摩川协议通讯式编码器	
能耗制动	可外接制动电阻	
能耗制动电压	DC65V (默认值, 可设置)	
过压报警电压	DC70V	
欠压报警电压	DC18V	
冷却方式	自然冷却	
重量	0.34kg	0.6kg
通用功能	输入规格	4 路数字输入, 共 COMI 端, 高电平: 12VDC~30VDC, 低电平: 0~5VDC, 最大频率 1kHz, 输入阻抗 5kΩ
	输入功能	可配置功能如下: 驱动器使能, 报警复位, 紧急停止, 正限位, 负限位, 多段速控制, 多段位置控制等
	输出规格	2 路数字输出, 共 COMO 端
	输出功能	可配置功能如下: 驱动器就绪, 驱动器错误, 电机零速, 电机抱闸生效, 限位生效等

	抱闸输出	默认 24VDC 抱闸输出，可配置电压为 0VDC~输入电压，可配置抱闸占空比
	脉冲控制	脉冲+方向、CCW+CW、A 相+B 相（5V-24V）
	TTL232	默认波特率 115200，最大支持 115200，可用 FULLING 上位机连接，也可使用自定义协议与控制器通讯
	保护功能	过压保护、欠压保护、电机过热（I ² T）保护、短路保护驱动器过热保护等
UART 波特率		115200bps（默认值，可修改）
RS485 波特率		115200bps（默认值，可修改）
CAN 波特率		500kbps（默认值，可修改）

第四章 系统接口与配线

4.1 驱动器各部分名称与功能

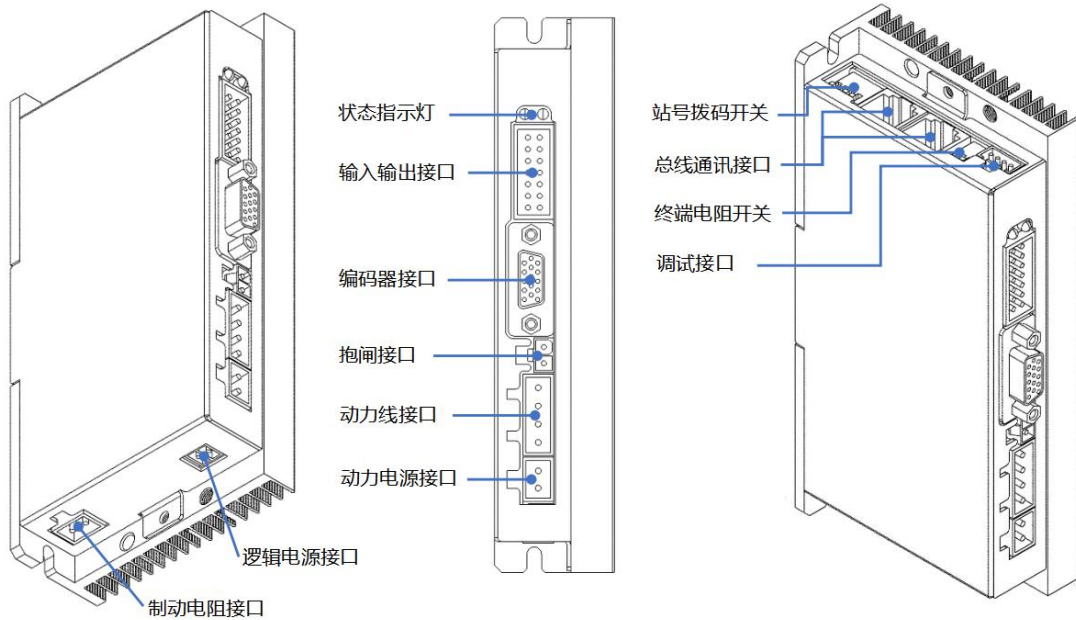


图 4-1 SVD4812RC-AA 各部分名称与功能

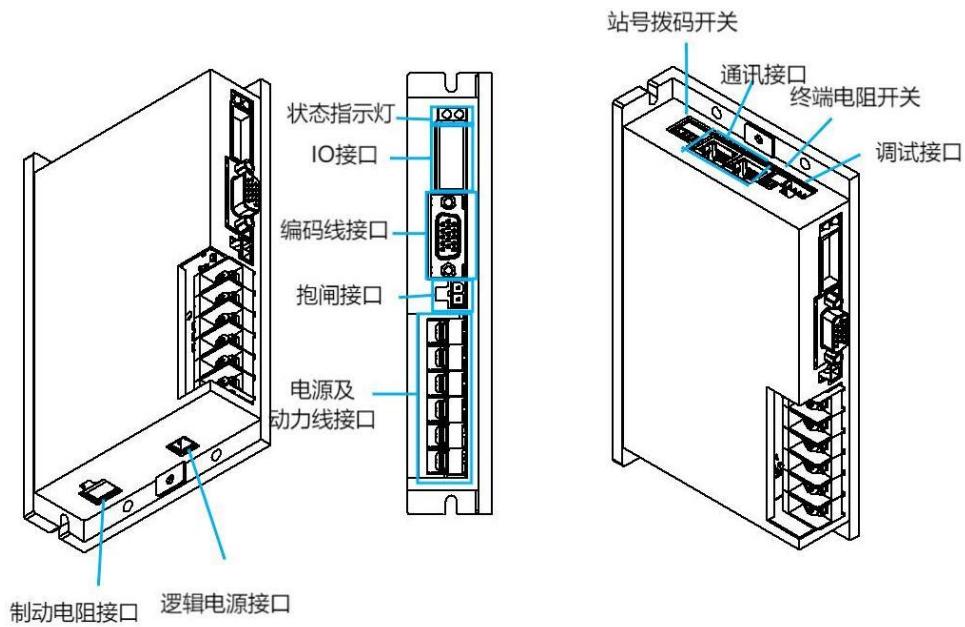
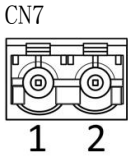



图 4-2 SVD4822RC-AA 各部分名称与功能

4.1.1 动力电源接口 (CN7)

图示	引脚编号及名称	功能描述
SVD4812RC-AA 	PIN1:DC+	电源输入正极
	PIN2:DC-	电源输入负极
SVD4822RC-AA 	PIN1:DC+	电源输入正极
	PIN2:DC-	电源输入负极
	PIN3:U	电机动力 U 相
	PIN4:V	电机动力 V 相
	PIN5:W	电机动力 W 相
	PIN6:PE	电机输入接地端

4.1.2 动力线接口 (CN6)

图示	引脚编号及名称	功能描述
CN6 	PIN1:U	电机动力 U 相
	PIN2:V	电机动力 V 相
	PIN3:W	电机动力 W 相
	PIN4:PE	电机输入接地端

4.1.3 抱闸接口 (CN5)

图示	引脚编号及名称	功能描述
CN5 	PIN1:BK-	抱闸输出负极
	PIN2:BK+	抱闸输出正极

4.1.4 编码器接口 (CN4)

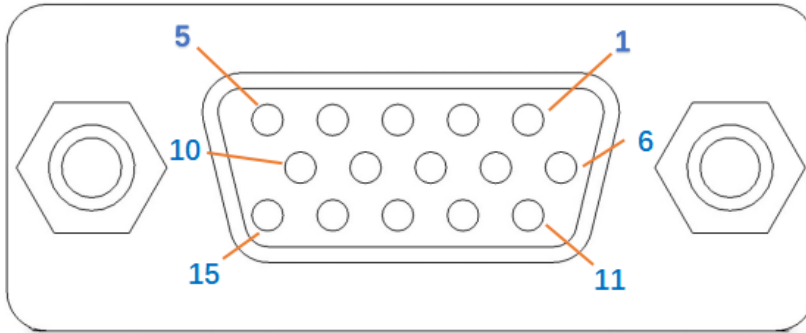


图 4-3 编码器电机的引脚定义

增量式编码器电机的引脚定义

接口名称	引脚编号及名称	功能描述
CN4 ENCODER	PIN13: A -	编码器 A - 输入端
	PIN12: B -	编码器 B - 输入端
	PIN11: Z -	编码器 Z - 输入端
	PIN5: U -	编码器 U - 输入端
	PIN15: V -	编码器 V - 输入端
	PIN14: W -	编码器 W - 输入端
	PIN8: A +	编码器 A + 输入端
	PIN7: B +	编码器 B + 输入端
	PIN6: Z +	编码器 Z + 输入端
	PIN4: U +	编码器 U + 输入端
	PIN10: V +	编码器 V + 输入端
	PIN9: W +	编码器 W + 输入端
	PIN1: 5V+	5V 电源电压输出端
	PIN2: GND	编码器信号接地端

通讯式单圈/多圈编码器电机的引脚定义

CN4 ENCODER	PIN1:ENC5V	编码器 5V 输出
	PIN2:GND	编码器信号接地端
	PIN9:SD+	数据信号正端
	PIN14:SD-	数据信号负端

4.1.5 输入输出接口 (CN3)

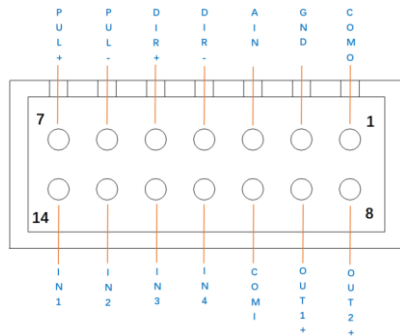


图 4-4 IO 口示意图

接口名称	引脚编号及名称	功能描述
CN3	PIN1:COMO	输出公共端
	PIN2:GND	接地端
	PIN3:AIN	模拟输入端
	PIN4:DIR-	脉冲输入端 输入电压：3.3V~24V 最大输入频率：500KHz
	PIN5:DIR+	
	PIN6:PUL-	
	PIN7:PUL+	
	PIN8:OUT2+	输出信号 2 正极，最大输出电流：50mA
	PIN9:OUT1+	输出信号 1 正极，最大输出电流：50mA
	PIN10:COMI	输入公共端
	PIN11:IN4	数字输入口（DIN 口），在 DIN 模式下，通过四个接口的数字逻辑输入，可以选择对应的不同的状态。（具体详情可以查看对应章节内容） 高电平：输入电压为 12VDC~30VDC，输入电流为输入电流 4~20mA 低电平：输入电压 0VDC~5VDC，输入频率：<1KHz
	PIN12:IN3	
	PIN13:IN2	
	PIN14:IN1	

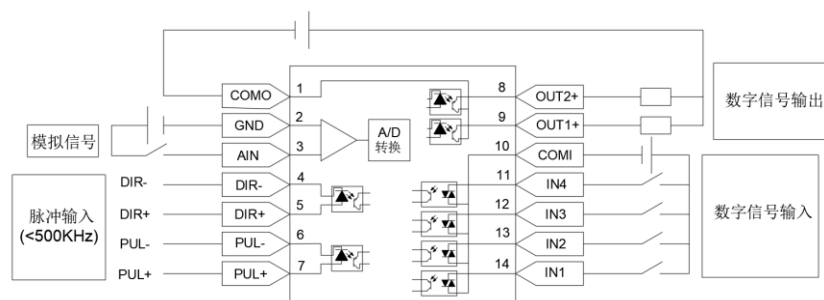
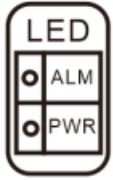


图 4-5 输入输出接口接线图

4.1.6 状态指示灯 (LED)

图示	引脚编号及名称	功能描述
	电机故障与运行指示灯	红色 LED 闪亮代表本机报警，仅绿色 LED 灯亮表示系统运行正常。请查看“第八章 报警与排除”章节

4.1.7 调试接口 (CN2-UART)

主要用于 UART 通讯，使用上位机时，通过该接口与驱动器连接

图示	引脚编号及名称	功能描述
	PIN1: +5V	UART +5V 输出端
	PIN2: GND	UART 模拟量 GND
	PIN3: RX	UART 数据接收端
	PIN4: TX	UART 数据发送端

4.1.8 终端电阻开关 (S2)

图示	引脚编号及名称	功能描述
	SW1	终端电阻匹配选择 (485) OFF: 断开 120 Ω 终端电阻 ON: 接入 120 Ω 终端电阻
	SW2	终端电阻匹配选择 (CAN) OFF: 断开 120 Ω 终端电阻 ON: 接入 120 Ω 终端电阻

4.1.9 总线通讯接口 (CN1A)

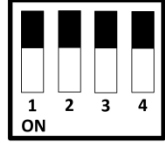
主要用于 CAN 通讯以及 RS485 通讯

图示	引脚编号及名称	功能描述
	PIN1: CAN_H	CAN 总线 H 端
	PIN2: CAN_L	CAN 总线 L 端
	PIN3: CAN/485 GND	RS485/CAN 总线公共地
	PIN4: 485-A	RS485 总线 A 端
	PIN5: 485-B	RS485 总线 B 端
	PIN6-PIN8: 保留	

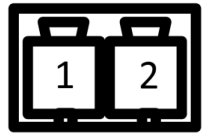
4.1.10 总线通讯接口 (CN1B)

图示	引脚编号及名称	功能描述
	PIN1:CAN_H	CAN 总线 H 端
	PIN2:CAN_L	CAN 总线 L 端
	PIN3:CAN/485 GDN	RS485/CAN 总线公共地
	PIN4:485-A	RS485 总线 A 端
	PIN5:485-B	RS485 总线 B 端
	PIN6-PIN8:保留	

4.1.11 站号拨码开关 (S1)

图示	引脚编号及名称	功能描述																																																																																					
	SW1	注 1:当站号大于 15 时, 请用内部存储方式来设置站号 注 2:此拨码开关状态只在上电时被识别一次 拨码开关所对应的站号对于所有通讯方式都适用 <table border="1" data-bbox="758 907 1348 1624"> <thead> <tr> <th>SW4</th> <th>SW3</th> <th>SW2</th> <th>SW1</th> <th>站号</th> </tr> </thead> <tbody> <tr> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>内部站号</td> </tr> <tr> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>ON</td> <td>1</td> </tr> <tr> <td>OFF</td> <td>OFF</td> <td>ON</td> <td>OFF</td> <td>2</td> </tr> <tr> <td>OFF</td> <td>OFF</td> <td>ON</td> <td>ON</td> <td>3</td> </tr> <tr> <td>OFF</td> <td>ON</td> <td>OFF</td> <td>OFF</td> <td>4</td> </tr> <tr> <td>OFF</td> <td>ON</td> <td>OFF</td> <td>ON</td> <td>5</td> </tr> <tr> <td>OFF</td> <td>ON</td> <td>ON</td> <td>OFF</td> <td>6</td> </tr> <tr> <td>OFF</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>7</td> </tr> <tr> <td>ON</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>8</td> </tr> <tr> <td>ON</td> <td>OFF</td> <td>OFF</td> <td>ON</td> <td>9</td> </tr> <tr> <td>ON</td> <td>OFF</td> <td>ON</td> <td>OFF</td> <td>10</td> </tr> <tr> <td>ON</td> <td>OFF</td> <td>ON</td> <td>ON</td> <td>11</td> </tr> <tr> <td>ON</td> <td>ON</td> <td>OFF</td> <td>OFF</td> <td>12</td> </tr> <tr> <td>ON</td> <td>ON</td> <td>OFF</td> <td>ON</td> <td>13</td> </tr> <tr> <td>ON</td> <td>ON</td> <td>ON</td> <td>OFF</td> <td>14</td> </tr> <tr> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>15</td> </tr> </tbody> </table>	SW4	SW3	SW2	SW1	站号	OFF	OFF	OFF	OFF	内部站号	OFF	OFF	OFF	ON	1	OFF	OFF	ON	OFF	2	OFF	OFF	ON	ON	3	OFF	ON	OFF	OFF	4	OFF	ON	OFF	ON	5	OFF	ON	ON	OFF	6	OFF	ON	ON	ON	7	ON	OFF	OFF	OFF	8	ON	OFF	OFF	ON	9	ON	OFF	ON	OFF	10	ON	OFF	ON	ON	11	ON	ON	OFF	OFF	12	ON	ON	OFF	ON	13	ON	ON	ON	OFF	14	ON	ON	ON	ON	15
	SW4		SW3	SW2	SW1	站号																																																																																	
	OFF		OFF	OFF	OFF	内部站号																																																																																	
	OFF		OFF	OFF	ON	1																																																																																	
	OFF		OFF	ON	OFF	2																																																																																	
	OFF		OFF	ON	ON	3																																																																																	
	OFF		ON	OFF	OFF	4																																																																																	
	OFF		ON	OFF	ON	5																																																																																	
	OFF		ON	ON	OFF	6																																																																																	
	OFF		ON	ON	ON	7																																																																																	
	ON		OFF	OFF	OFF	8																																																																																	
	ON		OFF	OFF	ON	9																																																																																	
	ON		OFF	ON	OFF	10																																																																																	
	ON		OFF	ON	ON	11																																																																																	
	ON		ON	OFF	OFF	12																																																																																	
ON	ON	OFF	ON	13																																																																																			
ON	ON	ON	OFF	14																																																																																			
ON	ON	ON	ON	15																																																																																			

4.1.12 逻辑电源接口 (CN8)

图示	引脚编号及名称	功能描述
	PIN1:+	逻辑电源输入端 输入电压: 24V 最大输入电流: 1A
	PIN2:-	

4.1.13 制动电阻接口 (CN9)

图示	引脚编号及名称	功能描述
CN9 	PIN1:RB+	制动电阻
	PIN2:RB-	

4.2 外部接线图

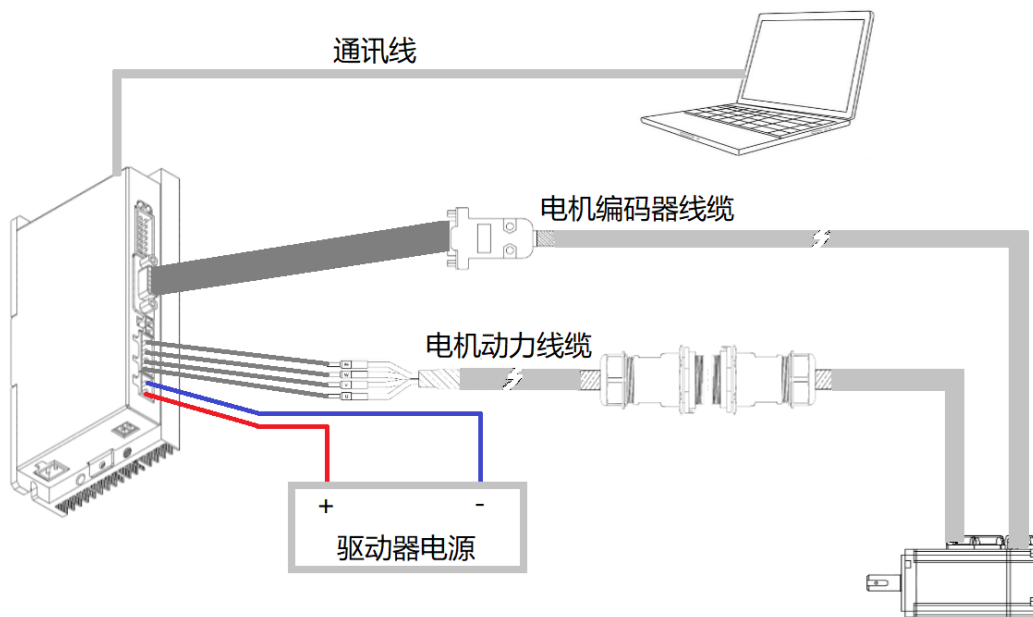


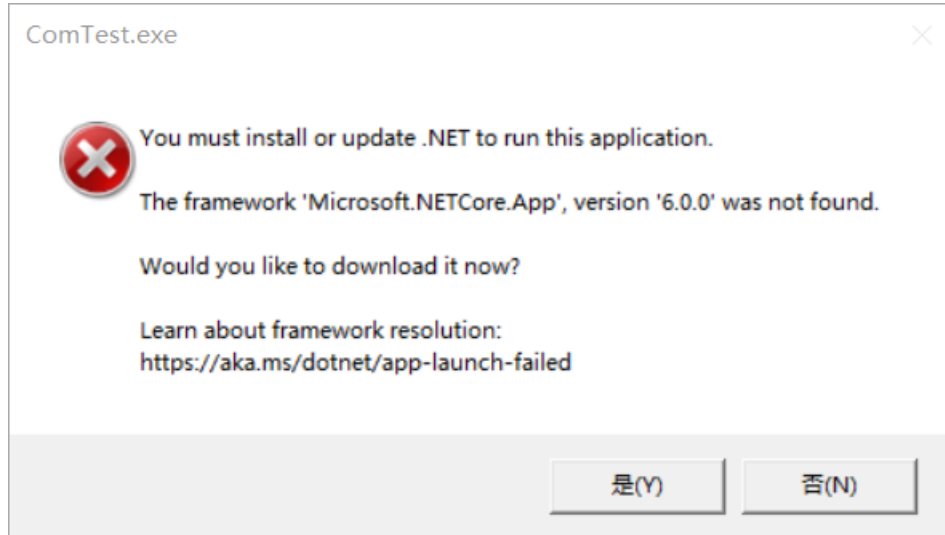
图 4-6 外部接线图

第五章 FULLING-tech 上位机使用指南

使用前必看（该上位机通过 UART 接口转 USB 同电脑连接）：

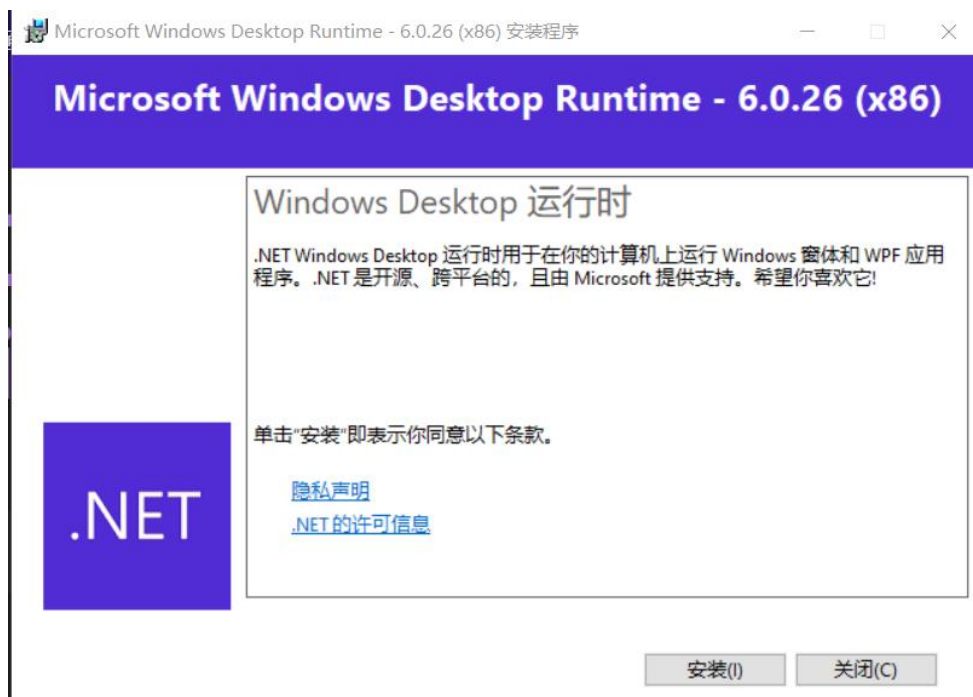
1. 插件问题

部分电脑打开 FULLING 上位机时会弹出如下窗口



这是由于电脑缺失软件运行所需插件，这里我们点“是”，页面会自动跳转下载插件（需连接网络）

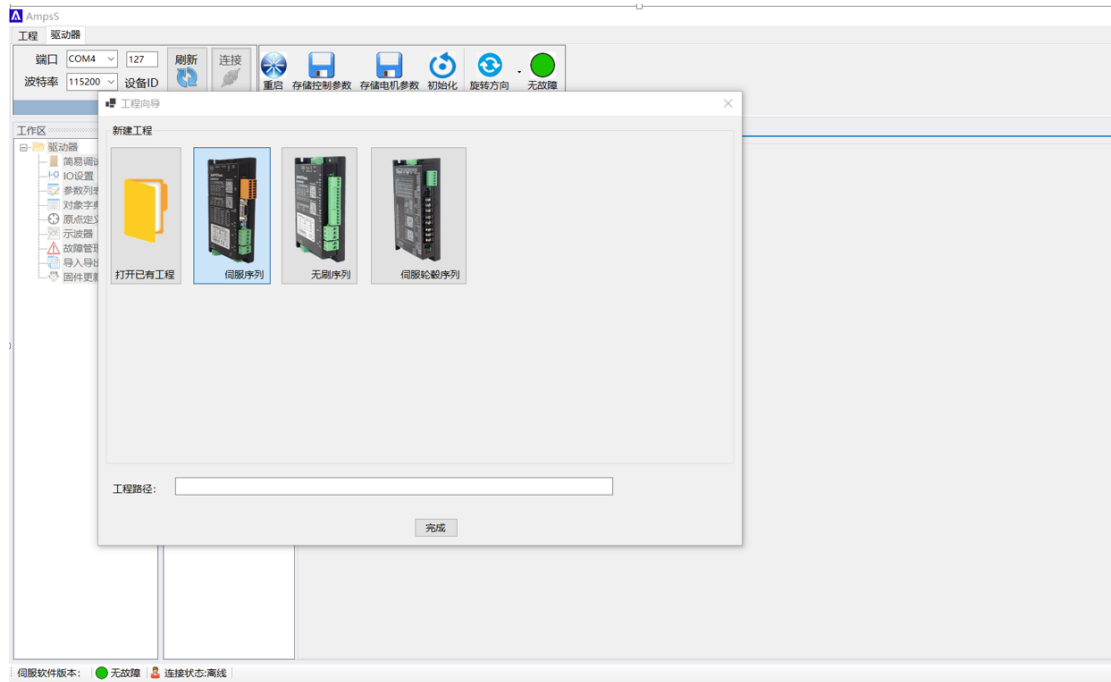
下载完成后，打开安装包，点击“安装”，待安装完毕点击“关闭”即可



之后便可正常打开 FULLING 上位机软件。

2. 基本页面介绍

打开上位机后，弹出如下页面



根据所使用驱动器选择相应系列进入软件，这里我们选择 ASDB 伺服序列。

进入后界面大致分为五个区域

串口设置：用于电脑与驱动器连接

电机控制：用于电机相关参数管理

状态：显示驱动器以及电机的状态

工作区：用户在工作区选择不同的对象进行对应的操作

弹窗：在工作区点击对象后，弹出对应的窗口，上位机登录后默认为工作区里参数列表。



图 5-1 上位机基本页面

关于弹窗新增行：

点击弹窗都有默认参数，如需再添加一些参数，可以在弹窗空白处按下鼠标右键，然后点击“添

加”，则会出现如下页面

在搜索框输入参数名称->点击搜索->点击 ->点击“确定”



图 5-2 新增参数

3. 对象字典的使用

在工作区可以找到“对象字典”，点击“对象字典”，会出现如下弹窗



图 5-3 对象字典页面

通过搜索栏搜索，我们可以快速找到相关参数的信息，如索引号、数据类型等。

5.1 驱动器与上位机的连接

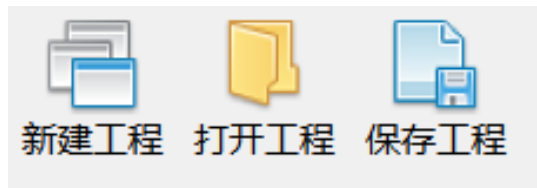
5.1.1 工程文件管理

工程文件用于保存配置和参数的设定值，通过保存工程，用户可以在后续使用时快速加载这些设定值，避免重复设置，节省时间和劳力，可以轻松地切换到不同的场景，以便在不同的工作环境

下进行实验和测试。

具体使用方法如下：

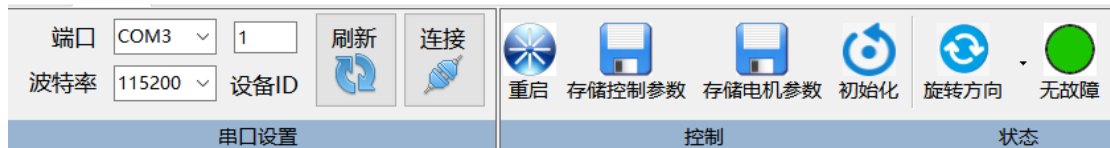
打开上位机，点击左上角的”工程”，即出现如下菜单栏



- “新建工程”用于创建新的工程
- “打开工程”用于打开现有的工程文件
- “保存工程”用于将当前工程保存为.kpjt 格式的文件

5.1.2 驱动器的连接

点击左上角的”驱动器”，即出现如下菜单栏



选择对应的端口号、波特率以及设备 ID，点击”连接”即可

- 任何通讯方式的设备 ID 都由拨码开关决定，详情请看第四章系统接口及配线。（驱动器万能拨号站为 127，即任何通讯方式都可以将设备 ID 设置为 127，进而连接驱动器）

点击参数列表->基本参数，里面的实际电流、状态字等出现数值时表示连接成功。

N	索引	数据类型	名称	设定值	当前值	单位
1	606100	Integer8	有效工作模式	---	0	DEC
2	604100	Unsigned16	状态字	---	231	HEX
3	606300	Integer32	实际位置	---	1	inc
4	606C00	Integer32	实际速度	---	0	rpm
5	607800	Integer16	实际电流q	---	-0.087	Ap
6	606000	Integer8	工作模式		3	DEC
7	604000	Unsigned16	控制字	6	6	HEX
8	607A00	Integer32	目标位置		0	inc
9	608100	Unsigned32	轮廓运行速度		0	rpm
10	608300	Unsigned32	轮廓加速度		50	rps/s
11	608400	Unsigned32	轮廓减速度		50	rps/s
12	60FF00	Integer32	目标速度	0	0	rpm
13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		43.945	Ap

图 5-4 基本参数

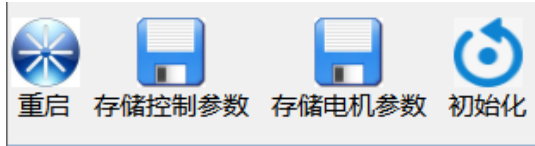
注意：

设备 ID、波特率进行设置之后需要存储控制参数，然后重新启动驱动器后才能生效。

5.2 电机参数以及控制参数管理

5.2.1 电机配置基本信息

- 电机配置：通过“参数列表”->“电机配置”
- 驱动器连接完成后，可通过如下图四个按键进行电机参数以及控制参数管理



- 完成电机相关参数设置后，依次点击
存储电机参数->重启->初始化->存储控制参数

注意：

电机初始化只需一次，后续如无更换电机或者更换驱动器等特殊情况，无需再进行配置。

5.2.2 驱动器配套电机具体操作如下：

(1) 通讯式

对于使用通讯式编码器的电机可以在电机型号处写入“???”四个问号，然后点击“存储电机参数”，重启驱动器后即可自动读取编码器中电机的型号与参数。

(2) 增量式

- 打开上位机基本参数里面的“电机配置”，填入对应电机型号
- 点击“存储电机参数”，然后点击“重启”
- 重启之后点击“初始化”，然后点击“存储控制参数”

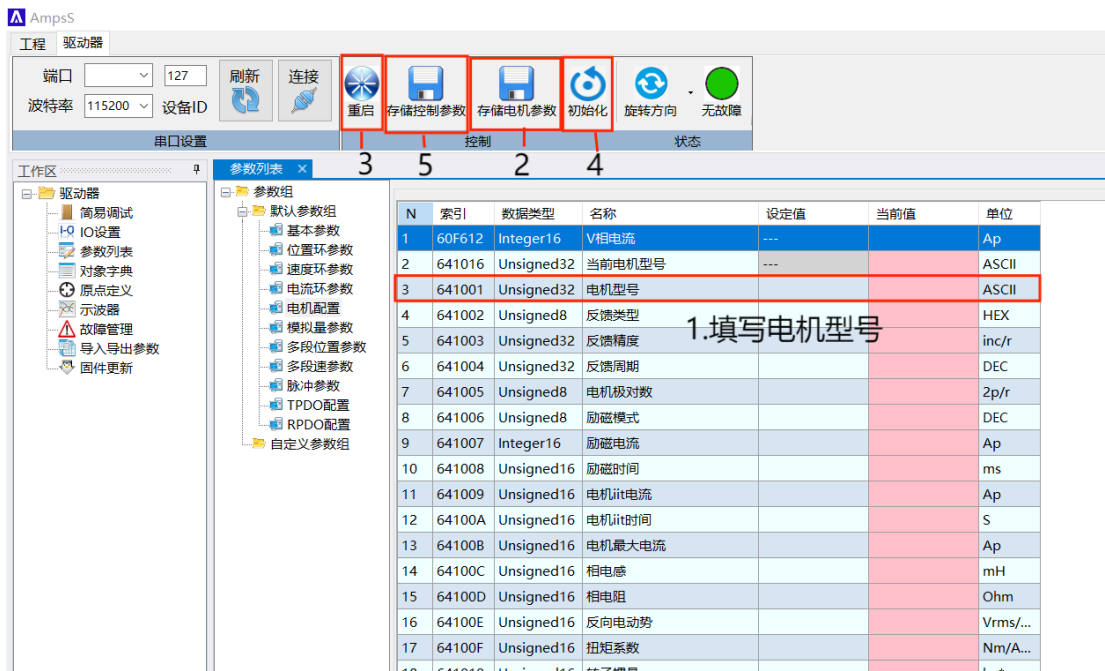


图 5-5 电机配置

以上介绍了配套电机的相关设置，若使用第三方电机，请看附录一 配置第三方电机

(3) 反馈类型

在“电机配置”栏中有“反馈类型”这一参数，根据编码器反馈信号类型进行填写，不同信号类型可以根据下表参考填入相应数值

反馈类型 bit位 编码器信号	bit3: 通讯式编 码器	bit2: UVW 单端 输出	bit1: UVW 接线检查（默认 差分信号输出）	bit0: ABZ 接线检查（默认 差分信号输出）	输入值 (HEX)
-----------------------	---------------------	-----------------------	---------------------------------	---------------------------------	--------------

ABZ 差分 无 UVW	0	0	0	1	1
ABZ 差分 UVW 差分	0	0	1	1	3
ABZ 差分 UVW 非差分	0	1	0	1	5
通讯式编码器	1	0	0	0	8

5.2.3 绝对值编码器相关参数

(1) 绝对值编码器复位命令

15	300303	Unsigned8	绝对值编码器复位命令	0	0	HEX
----	--------	-----------	------------	---	---	-----

0xBA: 复位故障状态

0x62: 清除多圈数据

0xC2: 清零单圈数据

(2) ALMC (编码器故障信息)

21	300302	Unsigned8	ALMC	---	0	HEX
----	--------	-----------	------	-----	---	-----

bit0: 速度异常

bit2: 单圈信息解算故障

bit5: 多圈数据丢失

bit6: 电池电压过低故障

bit7: 电池电压过低警告

(3) 编码器数据

18	300304	Integer32	单圈数据	---	116871	DEC
19	300305	Integer32	多圈数据	---	11	DEC
20	300306	Integer32	绝对值编码器位置	---	1558664	DEC

绝对值编码器位置 = (多圈数据 × 反馈精度) + 单圈数据

5.3 简易调试

简易调试是指在上位机软件中提供简单且易于操作的调试功能,以帮助用户快速检查设备或系统的工作状态和性能。这种调试功能通常不需要复杂的参数设置或专业知识,适用于快速排查问题和基本性能测试。

简易调试相关使用如下:



点击工作区“简易调试”,即会弹出如下页面






图 5-6 简易调试页面

该页面分为三大部分：

(1) **速度点动**：在速度点动模式下，用户可以设定设备的目标速度和加速度，设备会根据这些设定值进行平滑的加速和减速，达到目标速度后持续运动，直到停止命令被发送或设备停止。

- 设置完对应的点动速度、点动加速度以及点动减速度后，点击“on”，当“on”对应的图标由变为表示驱动器使能，电机锁轴。电机锁轴后长按“长按正转”或者“长按反转”即可使得电机旋转，我们也可以通过点击“全速运行”，勾选上该选项后，点击一下“长按正转”或者“长按反转”即可使得电机旋转。

(2) **位置点动**：用户可以输入目标位置的坐标信息，设备会根据这些坐标信息，按照设定的速度和加速度进行运动，从当前位置平滑地移动到目标位置。

- 设置完对应的转速、加速度以及减速度后，再设置电机所需要旋转到的对应位置，击“on”，当“on”对应的图标由变为表示驱动器使能，电机锁轴。点击即可使得电机旋转；设置剩余运行次数、间隔时间后勾选自动运行，电机将自动在设置的两个位置之间移动。

(3) **自动运行**：自动运行模式可以选择不控制、位置控制、速度控制、时间控制以及自动位置控制。速度控制、位置控制与上述的速度点动、位置点动概念相同，这不做解释。下面是时间控制和自动位置控制的解释。

时间控制：可以设置正转和反转的时间，电机将会不断地进行正转和反转。

如图所示，通过时间控制，设正转运行时间为 2000ms，反转运行时间为 2000ms，运行速度为 1000rpm，加速度和减速度都为 100rps/s。

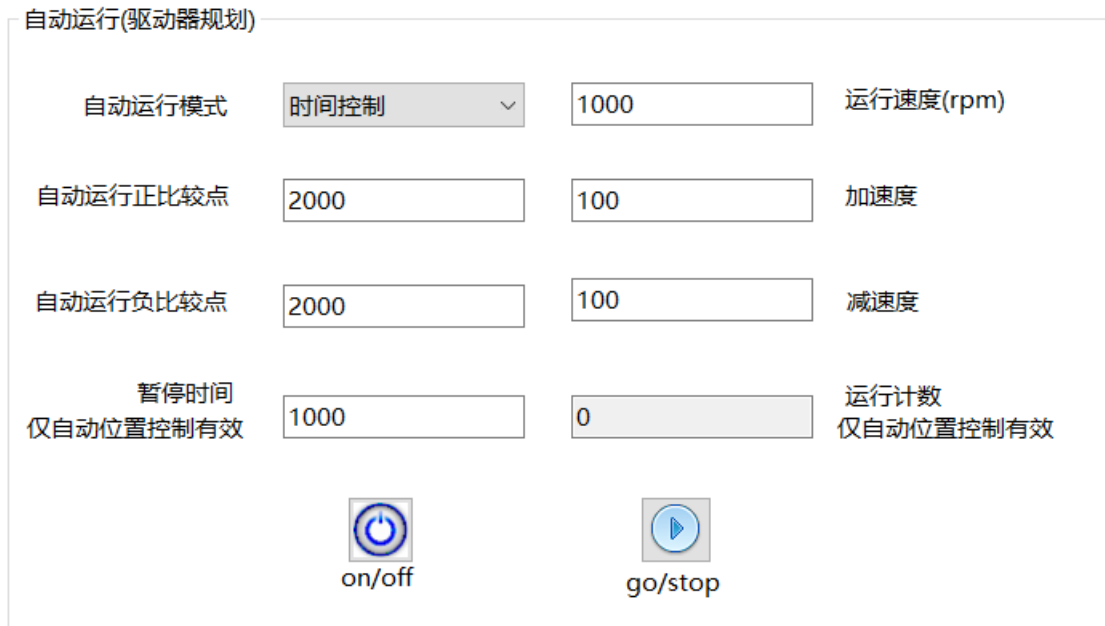


图 5-7 自动运行

自动位置控制：通过设置自动运行正比较点以及自动运行负比较点，使得电机可以在这两个位置之间来回运行，设置暂停时间来设置停留在某一位置的时间。（该模式与点位置模式下勾选自动运行类似）

注意：

若需进行模式切换或者参数修改，需要先点击 暂停，修改完参数及模式后点击 继续。

5.4 IO 口设置

在工作区点击 IO 设置，即可进入数字输入、数字输出模块



图 5-8 IO 口设置

5.4.1 输入模式


功能：点击 区域块进行功能的选择


仿真： 表示仿真通电情况。通电时为 ，断电时为

实际输入：表示实际输入的数字情况

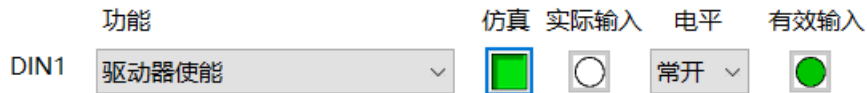
电平：（1）常开模式——高电平导通，低电平不导通

（2）常闭模式——高电平不导通，低电平导通

有效输入：由仿真（或者实际输入）和电平属性共同作用。表示数字逻辑 0 时为 ，表示数字

逻辑 1 时为 。

输入实例：



此时表示将 DIN1 设为驱动器使能端，同时表示使能了驱动器。

表 5-1 输入功能模块列表及说明

功能名称	功能描述
紧急停止	用于紧急情况下，停止系统
驱动器使能	用于使能驱动器，若使用 I0 口进行使能那么状态字将无法写入 1: 给 604000 写 0x2F 0: 给 604000 写 0x06
报警复位	用于关断报警
预使能	用于判断整个系统是否已经准备好 1: 驱动器可以使能 0: 驱动器不可以使能
Kvi 关闭	关闭速度环积分增益
正限位	配合硬件限位开关使用
负限位	
原点信号	原点开关信号，只能用作找原点
速度指令反向	在速度和力矩模式下，可将速度指令反向
多段速度控制 0	用于 Din 速度模式下的 Din 速度索引
多段速度控制 1	
多段速度控制 2	
外部输入故障	外部输入故障，如：外接温控开关，温度超限时输入 I0 信号让驱动器报错停机
找原点	用于找原点信号
多段位置控制 0	在 DIN 位置模式下，用于选择多段位置通过 BCD 码表示，如多段位置控制 2、多段位置控制 1、多段位置控制 0 分别为 011，则表示选择多段位置 3。 (多段位置都是表示绝对位置)
多段位置控制 1	
多段位置控制 2	
电子齿轮选择 0	在 DIN 位置模式下，用于选择电子齿轮通过 BCD 码表示，如电子齿轮选择 2、电子齿轮选择 1、电子齿轮选择 0 分别为 001，则表示选择电子齿轮 1。
电子齿轮选择 1	
电子齿轮选择 2	

注意：

在使用多段位置时，应当对多段速度给值。多段速度 0 表示：从当前位置运行到多段位置控制 0 的速度。

1	608100	Unsigned32	轮廓运行速度		49.999	rpm
2	608300	Unsigned32	轮廓加速度		49.997	rps/s
3	608400	Unsigned32	轮廓减速度		49.997	rps/s
4	202001	Integer32	多段位置控制0		10000	DEC
5	202002	Integer32	多段位置控制1		20000	DEC
6	202003	Integer32	多段位置控制2		30000	DEC
7	202004	Integer32	多段位置控制3		40000	DEC
8	202005	Integer32	多段位置控制4		50000	DEC
9	202006	Integer32	多段位置控制5		60000	DEC
10	202007	Integer32	多段位置控制6		70000	DEC
11	202008	Integer32	多段位置控制7		80000	DEC
12	202009	Integer32	多段速控制0	1000	999.999	rpm
13	20200A	Integer32	多段速控制1	1000	999.999	rpm
14	20200B	Integer32	多段速控制2	1000	999.999	rpm
15	20200C	Integer32	多段速控制3	1000	999.999	rpm
16	20200D	Integer32	多段速控制4	1000	999.999	rpm
17	20200E	Integer32	多段速控制5		0	rpm
18	20200F	Integer32	多段速控制6		0	rpm
19	202010	Integer32	多段速控制7		0	rpm

图 5-9 多段速控制

5.4.2 输出模式

功能：点击 区域块进行功能的选择

仿真： 表示仿真通电情况。通电时为 ，断电时为

实际输出：表示实际输出的数字情况

电平：（1）常开模式——高电平导通，低电平断开

（2）常闭模式——高电平断开，低电平导通

有效输出：由仿真（或者实际输出）和电平属性共同作用。表示数字逻辑 0 时为 ，表示数字

逻辑 1 时为 。

表 5-2 输出功能模块列表及说明

功能名称	功能描述
驱动器就绪	驱动器就绪，可以使能
驱动器错误	驱动器出现错误
电机位置到	位置小于位置到窗口
电机零速	速度小于速度到窗口
电机抱闸生效	指示抱闸已生效
电机速度到	电机达到速度

索引信号出现	索引信号出现
达到速度限制	在力矩模式下，实际速度达到最大速度限制
电机锁轴	驱动器使能，电机通电锁轴
限位中	启动位置限位开关
已找到原点	在零点模式下，已找到原点
目标扭矩到达	目标扭矩到达

5.5 触发式示波器使用

触发式示波器是一种特殊类型的示波器，用于在波形图中捕捉和显示特定的事件或信号。它通过触发信号来控制示波器采集数据的时机，以确保在特定条件下才显示波形图，从而使得波形图更容易被观察和分析。

在工作区点击”示波器”，即会弹出如下页面

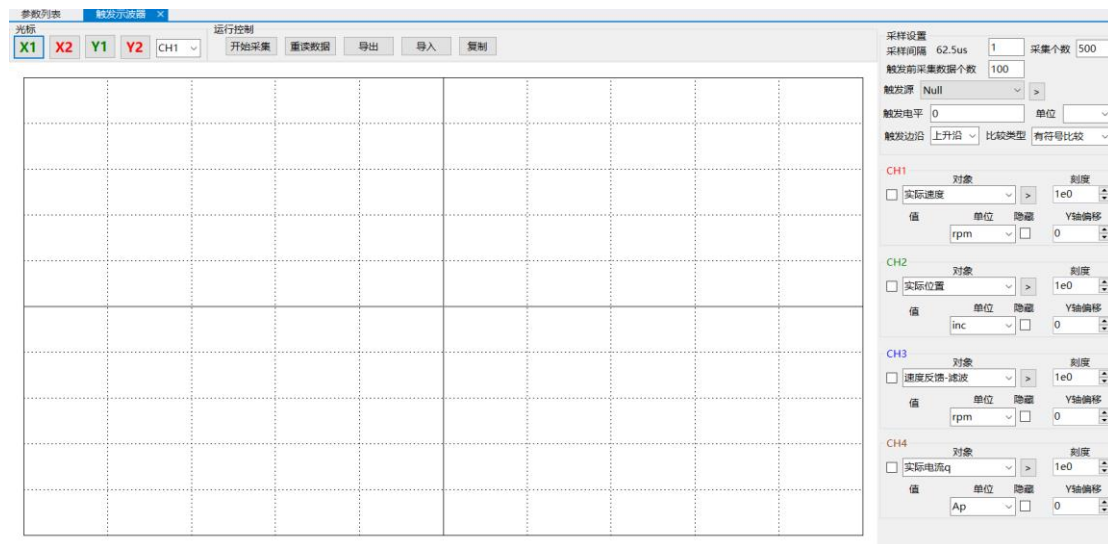


图 5-10 示波器页面

下面是对于示波器相关参数的一些解释：

- **采样周期：**采集数据的周期，设置为 1 表示每隔 62.5us 采集一个数据。
- **采集个数：**表示此次采样共采多少个数据，设置为 500 表示采集 500 个数据。
- **触发前采集数据个数：**触发源被触发前的采样个数，设置为 100 表示触发前采样 100 个。
- **触发源以及触发电平：**可以通过此设置决定什么时候开始采样，条件由自己设置。
- **触发边沿：**点击可更改为上升沿触发、下降沿触发或上下沿触发。
- **对象：**同时采样的 4 个对象数据长度之和最大为 64 位，例如，2 个 32 位的对象，或 4 个 16 位的对象。
- **光标：**通过点击按钮 可以选择相应光标，光标将会在示波器上显示出来，并在光标右边选择需要观察的通道。
- **复制：**将采样的数据复制到粘贴板。
- **导出：**将采样的数据导出到 .scope 文件。
- **导入：**将 .scope 文件导入并显示示波图。
- **重读数据：**将最近采集的数据从驱动器中读出并显示示波图。

假设通道默认对象列表里面没有所需要的参数，可以点击 ，跳转到“对象字典”后，在搜索栏里面输入参数名称，然后双击，参数便添加完成了。



图 5-11 添加参数

下面是有关于示波器读取实际位置与实际电流的实例：

(1) 先将工作模式设置为“3”模式，然后将轮廓加速度和轮廓减速度都设置为 50，目标速度设置为 1000。(这一步是为了有明显的参数可以读取)

1	606100	Integer8	有效工作模式	---	3	DEC
2	604100	Unsigned16	状态字	---	637	HEX
3	606300	Integer32	实际位置	---	156475512	inc
4	606C00	Integer32	实际速度	---	999.140	rpm
5	607800	Integer16	实际电流q	---	0.043	Ap
6	606000	Integer8	工作模式	3	3	DEC
7	604000	Unsigned16	控制字	f	f	HEX
8	607A00	Integer32	目标位置		0	inc
9	608100	Unsigned32	轮廓运行速度		0	rpm
10	608300	Unsigned32	轮廓加速度		50	rps/s
11	608400	Unsigned32	轮廓减速度		50	rps/s
12	60FF00	Integer32	目标速度	1000	999.999	rpm
13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		43.945	Ap

图 5-12 修改参数

(2) 打开示波器，按照如下步骤进行操作

第一步：在右侧 CH1，将对象设置为实际位置

第二步：触发源设置为 NULL，即点击“开始采集”就会立即开始采集

第三步：点击“开始采集”，等到进度条完成即采集成功

第四步：将第一步中的“实际位置”切换为“实际电流”，再点击“开始采集”，即可将图像由实际位置图像切换为实际电流的图像

注意：

若想设置采样触发条件，如将触发条件设置为 有效目标速度达到 500rpm 时，需要跟修改控制字一样，通过键盘上的“Enter”确认。

每进行一次对象切换或者通道切换都需要出现点击一次“开始采集”

- 我们可以通过右侧的“刻度”对图像进行缩放；
- 此外长按鼠标中键同样可以对波形进行缩放。
- 通过“Y轴偏移”图像进行上下移动；
- 点击光标附近的“X1”和“Y1”，屏幕上出现两条分别垂于与X轴、Y轴的虚线，图像中右上角会显示两条直线的交叉点坐标。X2、Y2使用方法相同。dX、dY表示X1与X2、Y1与Y2的差值。

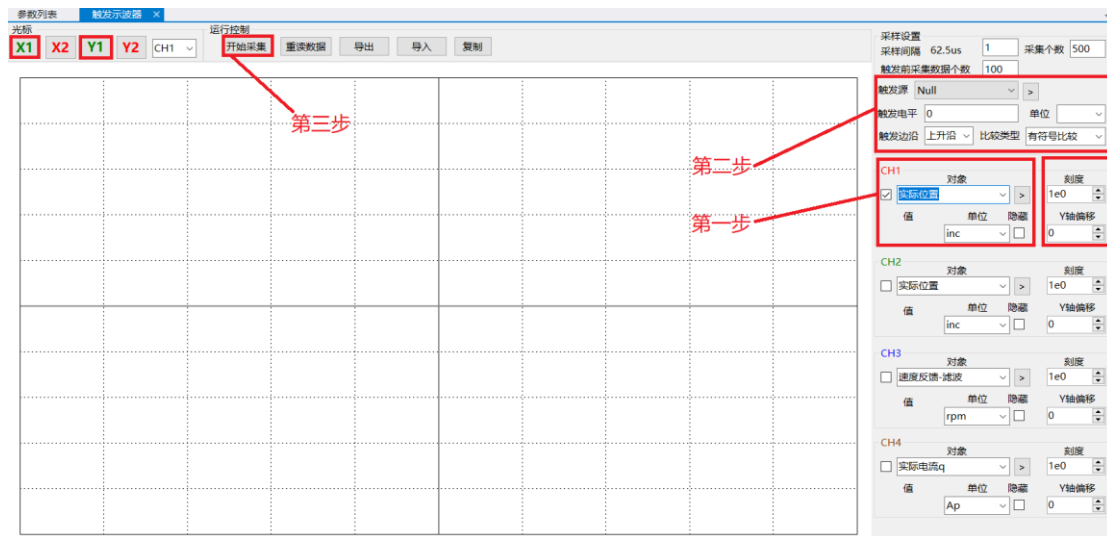



图 5-13 读取实际位置的步骤

以上列举了读取实际位置的步骤，读取实际速度等参数方法基本同上。

点击“导出”，即可在指定路径（自定义）生成一个 .wave 的文件 ，再次点击“导入”，即可在示波器上，展示这个波形了。

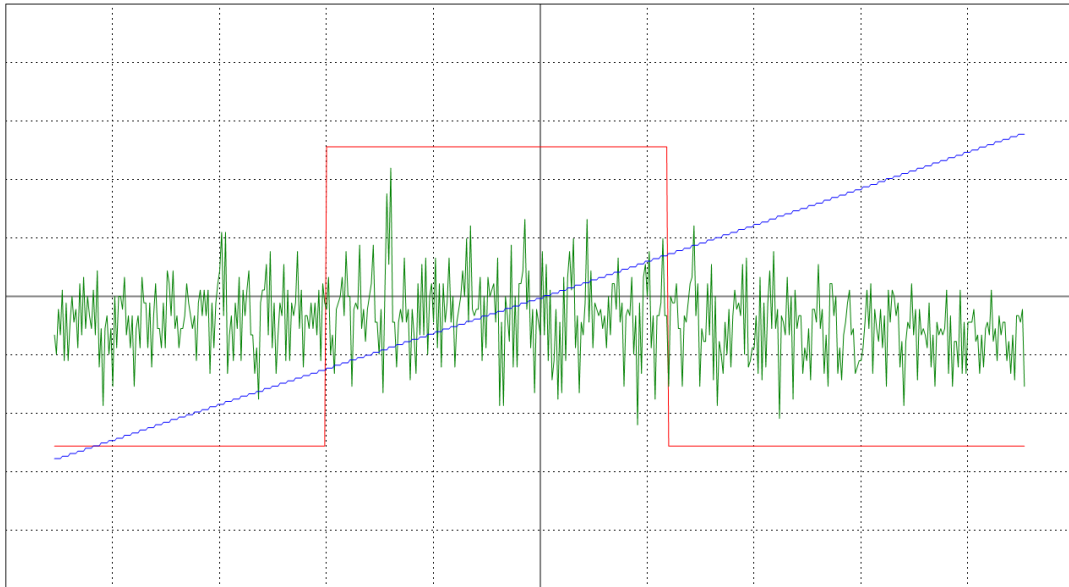


图 5-14 test.wave 的文件导入后的波形

此外，我们还有可以点击“复制”，将数据复制到 Excle 表格。

5.6 历史错误与报警



当上位机的图标由 无故障 变为 有故障 表示有故障出现

N	索引	数据类型	名称	设定值	当前值	单位
1	606100	Integer8	有效工作模式	---	3	DEC
2	604100	Unsigned16	状态字	---	5437	HEX
3	606300	Integer32	实际位置	---	0	inc
4	606C00	Integer32	实际速度	---	0	rpm
5	607800	Integer16	实际电流q	---	0	Ap
6	606000	Integer8	工作模式		3	DEC
7	604000	Unsigned16	控制字		2f	HEX
8	607A00	Integer32	目标位置		0	inc
9	608100	Unsigned32	轮廓运行速度		0	rpm
10	608300	Unsigned32	轮廓加速度		50	rps/s
11	608400	Unsigned32	轮廓减速度		50	rps/s
12	60FF00	Integer32	目标速度		0	rpm
13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		43.945	Ap

图 5-15 故障例图



直接点击 有故障 可以跳转到故障管理，或者通过点击工作区的故障管理

故障管理页面如下



图 5-16 故障管理

通过上位机页面我们可以直接得到故障原因以及解决方案。此外点击历史故障可以查询到最近 8 次错误信息，包括故障代码、总线电压、速度、电流、温度、工作模式、时间以及 PWM 状态。第一行为最新一次错误信息，以此类推往下推。下面是常见错误代码表，如果问题无法解决，请跳转至第八章 报警排除 查询解决问题的办法。

表 5-3 错误代码表

错误代码	错误名称	错误描述
0x1001	MCU 内部错误	检测到 MCU 型号错误
0x2214	软件过流	软件过流
0x2320	过流报警	驱动器功率管或电机短路
0x3130	电机缺相	电机动力线 UVW 中某相未连接
0x3210	高压报警	总线电压过高
0x3220	低压报警	总线电压过低
0x4210	温度报警	散热器温度过高
0x4310	电机温度过高	电机温度传感器报警
0x5112	逻辑低压报警	逻辑电源电压过低
0x5210	电流传感器	电流传感器信号偏移或波纹过大
0x5441	负限位错误	负限位报错仅在限位功能定义为 1 时才会产生故障报警
0x5442	正限位错误	正限位报错仅在限位功能定义为 1 时才会产生故障报警
0x5443	预使能错误	输入口定义预使能，在驱动器使能或将要使能预使能输入口无有效输入
0x5530	Eeprom 报警	EEPROM 数据校验错误
0x6320	电机配置	EEPROM 中没有电机配置信息/电机从未配置
0x7110	吸收电阻报警	制动电阻过载
0x7122	电机励磁报警	电机 UVW 相序不正确或者编码器未连接
0x7305	编码器计数报警	编码器受到干扰
0x7306	主编码器计数错误	主编码器计数错误

0x7310	速度超差	设定速度与实际速度偏差过大
0x7382	主编码器连接错误	主编码器连接错误
0x7500	总线掉线错误	总线通信发生异常
0x8611	位置误差过大报警	实际跟随误差超过设置的最大跟随误差
0x8613	找原点错误	在执行原点搜索或归零操作时出现问题或失败的情况
0x8A81	全闭环下编码器计数方向错误	全闭环工作状态下,主编码器计数方向与电机编码器计数方向相反
0xFF10	用户 iit	电机或驱动器功率管 IIt 故障
0xFF21	保留	
0xFF22	保留	
0xFF30	脉冲频率过高	脉冲输入频率过高
0xFF40	编码器 ABZ 连接报警	编码器 ABZ 连接错误或者未连接
0xFF41	编码器 UVW 连接报警	编码器 UVW 连接错误或者未连接
0xFF42	保留	

5.7 驱动器参数读写

通过驱动器参数读写,可以将驱动器配置保存在上位机或控制器中,可以方便后续的设备复现和配置。这在设备更换或维护时特别有用,可以快速将之前的配置应用到新的设备上,节省调试时间和成本。

5.7.1 读驱动器配置

点击工作区里面的“导入导出参数”->“读驱动器参数”->“打开列表”(选择后缀为.oparam的文件)->“读数据”->“保存文件”



图 5-17 读驱动器配置

5.7.2 写驱动器配置

点击工作区里面的“导入导出参数”->“写驱动器参数”->“打开文件”(选择后缀为.iparam的文件)->“写数据”->“存储写入的参数”

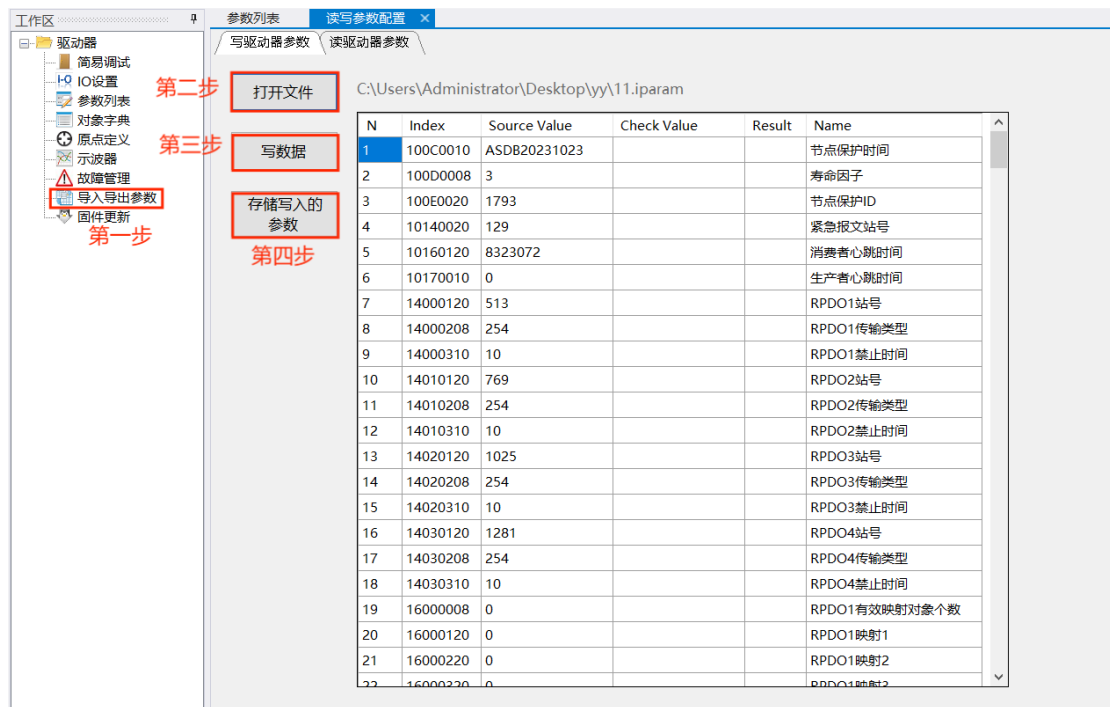


图 5-18 写驱动器配置

5.8 固件下载

如因一些特殊原因，需要重新烧录固件则可按照以下步骤：

点击“固件更新”->选择文件->点击“烧录”

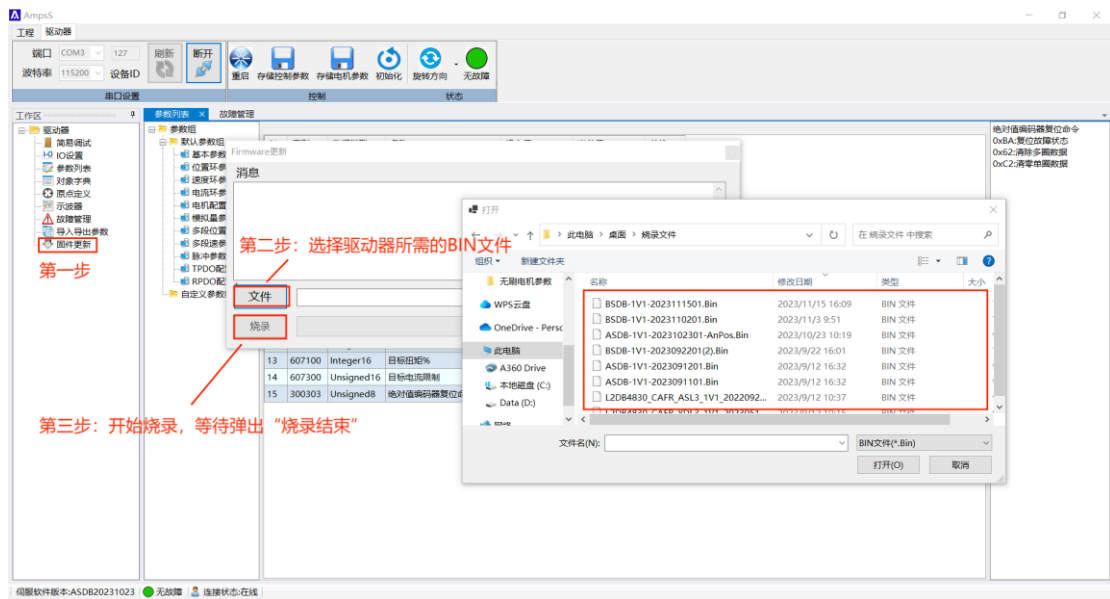


图 5-19 固件下载

第六章 工作模式介绍

6.1 速度模式介绍

速度模式分为立即速度模式（-3）和带加速度的速度模式（3），速度模式的速度可以由内部指令、外部 I/O 口（DIN 速度模式）、模拟输入三种方式进行设置。

6.1.1 内部指令设置带加速度的速度（3）

表 6-1 速度模式相关参数

索引	数据类型	参数名称	参数描述	设置值
606000	Integer8	工作模式	用于设置电机工作模式 3: 带加速度的速度模式	3
604000	Unsigned16	控制字	用于设置电机状态 F: 电机锁轴 6: 电机松轴	F 和 6
60FF00	Integer32	目标速度	用户预设的速度值	由用户自行设定
608300	Unsigned32	轮廓加速度	带加速度的速度模式（3）所设定的加速度	由用户自行设定
608400	Unsigned32	轮廓减速度	带加速度的速度模式（3）所设定的减速度	由用户自行设定

若采用上位机进行调试，则可以按照以下步骤：

工作区“参数列表”->“基本参数”->设置工作模式、控制字等参数

注意：

输入完数据后按下“Enter”键，然后查看当前值是否修改



图 6-1 上位机修改速度

6.1.2 内部指令设置立即速度（-3）

表 6-2 立即速度模式相关参数

索引	数据类型	参数名称	参数描述	设置值
606000	Integer8	工作模式	用于设置电机工作模式 -3: 立即速度模式	-3
604000	Unsigned16	控制字	用于设置电机状态 F: 电机锁轴 6: 电机松轴	F 和 6
60FF00	Integer32	目标速度	用户预设的速度值	由用户自行设定

若使用上位机进行操作，则设置工作模式、控制字以及目标速度就可以了

6.1.3 外部 I/O 口设置速度（DIN 速度模式）

表 6-3 外部 I/O 口设置速度模式

索引	数据类型	参数名称	参数描述	设定值	单位
202009	Integer32	多段速控制 0	由 DIN 口设置的多段速度控制 2、多段速度控制 1、多段速度控制 0 进行控制选择。 若多段速度控制 2、1、0 输入的数字逻辑分别 010，则表示选用多段速控制 2；输入的数字逻辑分别 110，则表示选用多段速控制 6。	用户自行设定	rpm
20200A	Integer32	多段速控制 1			
20200B	Integer32	多段速控制 2			
20200C	Integer32	多段速控制 3			
20200D	Integer32	多段速控制 4			
20200E	Integer32	多段速控制 5			
20200F	Integer32	多段速控制 6			
202010	Integer32	多段速控制 7			
608300	Unsigned32	轮廓加速度	处于 3 模式下的加速度	用户自行设定	rps/s
608400	Unsigned32	轮廓减速度	处于 3 模式下的减速度		
606000	Integer8	工作模式	用于设置电机工作模式 3: 带加速度的速度模式	3	606000
604000	Unsigned16	驱动器使能		6->2F	HEX

若采用上位机进行调试，则可以按照以下步骤：

(1) 工作区“参数列表”->“多段速参数”->设定对应的值



图 6-2 多段速参数设置

(2) 点击“IO 设置”->选择 DIN 口，将其设置为多端速度控制

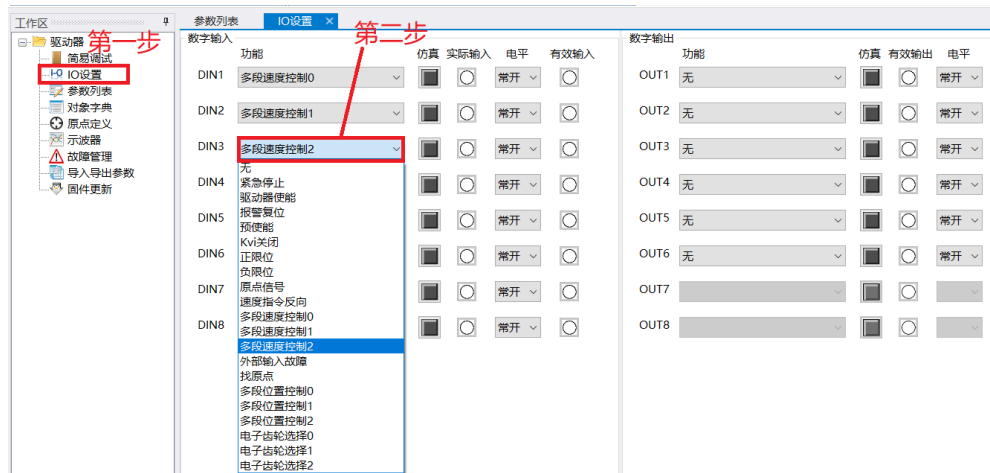


图 6-3 IO 口设置

(3) 进入“基本参数”，将工作模式设为 3，控制字设为 F，这样我们就可以通过 IO 口的输入控制不同的速度了。

N	索引	数据类型	名称	设定值	当前值	单位
1	606100	Integer8	有效工作模式	---	3	DEC
2	604100	Unsigned16	状态字	---	437	HEX
3	606300	Integer32	实际位置	---	10292548	inc
4	606C00	Integer32	实际速度	---	54.539	rpm
5	607800	Integer16	实际电流q	---	0.087	Ap
6	606000	Integer8	工作模式	3	3	DEC
7	604000	Unsigned16	控制字	f	f	HEX
8	607A00	Integer32	目标位置		0	inc
9	608100	Unsigned32	轮廓运行速度		0	rpm
10	608300	Unsigned32	轮廓加速度		50	rps/s
11	608400	Unsigned32	轮廓减速度		50	rps/s
12	60FF00	Integer32	目标速度		0	rpm
13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		43.945	Ap

图 6-4 基本参数设置

以上情况为例，设 DIN1 为多段速度控制 0，DIN2 为多段速度控制 1，DIN3 为多段速度控制 2，假设 DIN3-DIN2-DIN1 输入为 110，则电机以 500rpm（多段速度控制 6）的速度旋转。

注意：

若使用 I0 口进行速度设置，那么通过其他模式对于速度的输入会失效

6.1.4 模拟输入设置速度

表 6-4 模拟输入设置速度相关参数

索引	数据类型	参数名称	参数描述	设定值	单位
250106	Unsigned16	模拟量原始值	模拟量原始值	只读	DEC
250206	Integer16	模拟输入有效数据	模拟量输入信号滤波后的值	用户自定义	V
250201	Integer16	模拟输入校正增益	对模拟输入信号进行校准的比例因子	用户自定义	DEC
250202	Integer16	模拟输入校正偏移	用于校准模拟输入信号的偏移量	用户自定义	DEC
250203	Unsigned8	模拟输入滤波系数	模拟量输入信号滤波参数	用户自定义	DEC
250204	Integer16	模拟输入偏移	模拟输入偏移电压		V
250205	Integer16	模拟输入死区	模拟输入死区电压		V
250207	Unsigned8	模拟-速度控制	模拟量输入信号控制 1：表示模拟输入速度模式 2：表示模拟输入位置模式	1	DEC
25020A	Integer32	模拟-速度因数	将模拟输入信号转换为电机速度的比例因子	用户自定义	rpm/v
606000	Integer8	工作模式	用于设置电机工作状态 3：速度模式 -3：立即速度模式	3、-3	DEC
604000	Unsigned16	控制字	用于设置电机状态 F：电机锁轴 6：电机松轴	F 和 6	HEX

假设输入的模拟信号为 V_{in} ，则经过数据处理后得到的为 V_{in_r} ，那么有：

$V_{in_r} = V_{in} - \text{模拟输入偏移电压}$

如果 V_{in_r} 的绝对值大于死区（即 $|V_{in_r}| > \text{死区}$ ），则模拟输入有效数据为：模拟输入有效数据 = $|V_{in_r}| - \text{死区}$

否则，模拟输入有效数据为 0。

当前速度=模拟输入有效数据*模拟-速度因数

若使用上位机进行操作，可按照以下步骤

- (1) 在 CN3 的 GND、AIN 接入模拟电压
- (2) 上位机连接完成后，点击“参数列表”->“模拟量参数”，进入如下页面

(3) 设置完数值后，修改工作模式(3 或者-3)，然后进行使能 (F)

N	索引	数据类型	名称	设定值	当前值	单位
1	250106	Unsigned16	模拟量原始值	---	889	DEC
2	250206	Integer16	模拟输入有效数据		5.344	V
3	250201	Integer16	模拟输入校正增益		9010	DEC
4	250202	Integer16	模拟输入校正偏移		7548	DEC
5	250203	Unsigned8	模拟输入滤波系数		5	DEC
6	250204	Integer16	模拟输入偏移		0	V
7	250205	Integer16	模拟输入死区		0	V
8	250207	Unsigned8	模拟-速度控制		0	DEC
9	25020A	Integer32	模拟-速度因数		75	rpm/v
10	25020B	Unsigned8	模拟-力矩控制		0	DEC
11	25020C	Integer16	模拟-力矩因数		1.098	Ap/v
12	25020D	Integer16	模拟-最大力矩因数		2.25	Ap/v

图 6-5 模拟输入相关参数

6.2 位置模式 (1)

位置模式分为相对位置模式、绝对位置模式、DIN 位置模式

关于绝对位置与相对位置：

- 绝对位置：在绝对位置中，每个位置都有唯一的编码值与之对应，通过读取编码值可以准确确定电机的绝对位置。
- 相对位置：相对位置是指电机或运动装置相对于起始位置的位移量，相对位置通常需要一个初始位置的参考点，以便在运动过程中准确地累积位移量。

我们可以通过下图的表格来进一步理解绝对位置与相对位置

实际位置	执行操作	执行后的实际位置
500	绝对位置 600	600
500	相对位置 600	1100

表 6-5 位置模式常用对象列表

索引	数据类型	参数名称	参数描述	设定值	单位
606000	Integer8	工作模式	用于设置电机工作模式 1: 位置模式	1	DEC
604000	Unsigned16	控制字	绝对位置: 0x2F->0x3F(单次) 0x103F(多次) 相对位置: 0x4F->0x5F DIN 位置: 0x2F	根据需求 选择	HEX
607A00	Integer32	目标位置	用户指定的位置	用户 自定义	inc
608100	Unsigned32	轮廓运行速度	处于位置模式下,将以轮廓运行速度向目标位置移动		rpm

注：0x3F 仅能跑到第一次识别的目标位置，若到达目标位置后要再次运动到其他位置，需先将

控制字写入 6，再写入 3F，电机才会往下一个位置；
0x103F 则无需改变控制字，电机会运动目标位置的当前数值。

6.2.1 内部指令设置位置

内部指令设置位置模式与 6.1 内部指令设置速度模式操作基本相同，这里不做过多叙述。

6.2.2 外部 I/O 口设置位置（DIN 位置模式）

表 6-6 DIN 位置模式常用对象列表

索引	数据类型	参数名称	参数描述	设定值	单位
608100	Unsigned32	轮廓运行速度	用户将以此速度运行至指定位置	用户根据需求自行定义	rpm
202001	Integer32	多段位置控制 0	用户需要电机所到达的位置，最多可以设置 8 个不同位置，根据 I/O 口输入的多段位置控制 BCD 码进行选择		DEC
202002	Integer32	多段位置控制 1			DEC
202003	Integer32	多段位置控制 2			DEC
202004	Integer32	多段位置控制 3			DEC
202005	Integer32	多段位置控制 4			DEC
202006	Integer32	多段位置控制 5			DEC
202007	Integer32	多段位置控制 6			DEC
202008	Integer32	多段位置控制 7			DEC

假设 I/O 口下图进行设置

数字输入

	功能	仿真	实际输入	电平	有效输入
DIN1	多段位置控制0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	常开	<input type="checkbox"/>
DIN2	多段位置控制1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	常开	<input type="checkbox"/>
DIN3	多段位置控制2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	常开	<input type="checkbox"/>

图 6-6 多段位置-I/O 口设置

那么当 DIN3-DIN2-DIN1 的数字逻辑输入为 010 时，则表示选用多段位置控制 2；
当 DIN3-DIN2-DIN1 的数字逻辑输入为 011 时，则表示选用多段位置控制 3，电机将根据设定的值运行到指定位置。

6.2.3 模拟输入设置位置

使用该模式需先在模拟量参数界面添加“模拟输入最大值”“模拟输入最小值”“模拟量位置最大值”和“模拟量位置最小值”。



表 6-7 模拟输入设置速度相关参数

索引	数据类型	参数名称	参数描述	设定值	单位
250106	Unsigned16	模拟量原始值	模拟量原始值	只读	DEC
250206	Integer16	模拟输入有效数据	模拟量输入信号滤波后的值	用户自定义	V
250201	Integer16	模拟输入校正增益	对模拟输入信号进行校准的比例因子	用户自定义	DEC
250202	Integer16	模拟输入校正偏移	用于校准模拟输入信号的偏移量	用户自定义	DEC
250203	Unsigned8	模拟输入滤波系数	模拟量输入信号滤波参数	用户自定义	DEC
250204	Integer16	模拟输入偏移	模拟输入偏移电压		V
250205	Integer16	模拟输入死区	模拟输入死区电压		V
250207	Unsigned8	模拟-速度控制	模拟量输入信号控制 1:表示模拟输入速度模式 2:表示模拟输入位置模式	2	DEC
606000	Integer8	工作模式	用于设置电机工作状态 1:位置模式	1	DEC
604000	Unsigned16	控制字	用于设置电机状态	103F:模拟量位置控制使能 6: 松轴	HEX

25020E	Integer16	模拟输入最大值	用于做位置计算的模拟量输入最大值	用户自定义	DEC
25020F	Integer16	模拟输入最小值	用于做位置计算的模拟量输入最小值	用户自定义	DEC
250210	Integer32	模拟量位置最大值	电机在模拟量位置模式下的位置最大值	用户自定义	inc
250211	Integer32	模拟量位置最小值	电机在模拟量位置模式下的位置最小值	用户自定义	inc

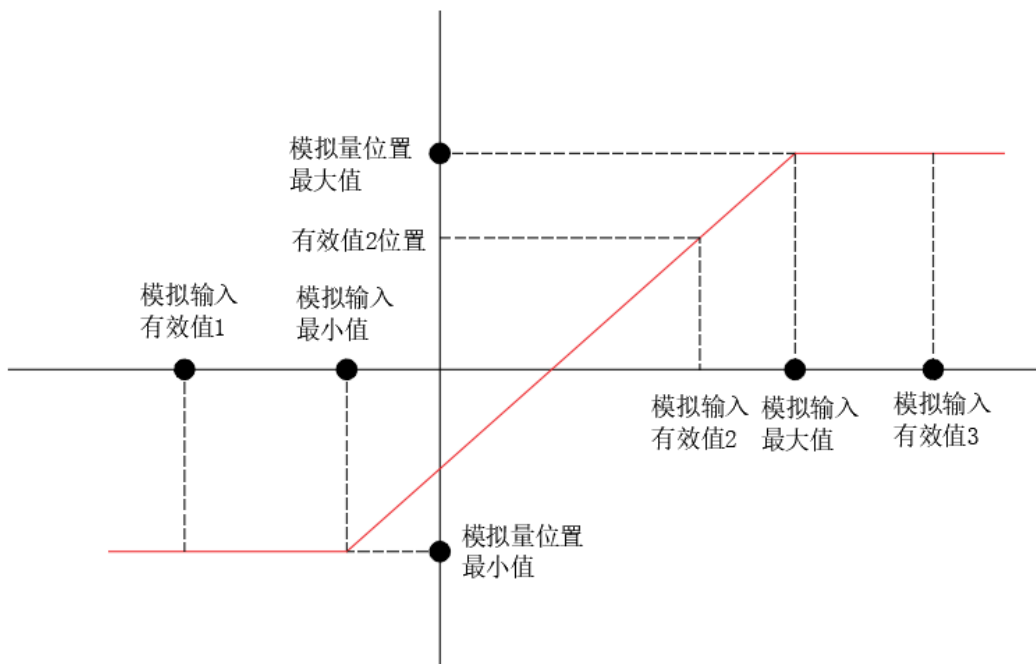


图 6-7 模拟输入与位置关系

如上图，在该模式下，

当 模拟输入有效值 < 模拟输入最小值，电机所在位置为“模拟量位置最小值”；

当 模拟输入最小值 < 模拟输入有效值 < 模拟输入最大值，电机位置在模拟量位置函数直线上；

当 模拟输入有效值 > 模拟输入最大值，电机所在位置为“模拟量位置最大值”。

例：各参数设置如下

13	25020E	Integer16	模拟输入最大值	3	2.999	V
14	25020F	Integer16	模拟输入最小值	1	0.999	V
15	250210	Integer32	模拟量位置最大值	30000	30000	DEC
16	250211	Integer32	模拟量位置最小值	10000	10000	DEC

1. 当模拟输入有效值位 0.491 V 时，小于模拟输入最小值，电机将运行值模拟量位置最小值，即 10000 inc 处。

1	250106	Unsigned16	模拟量原始值	---	1796	DEC
2	250206	Integer16	模拟输入有效数据	---	0.491	V
3	250201	Inteaer16	模拟输入校正增益		9010	DEC
17	606300	Integer32	实际位置	---	10000	inc

2. 当模拟输入有效值位 2.410 V 时，处于模拟输入最大值和最小值之间，电机将按其函数比例运动至 24000 inc 处。

1	250106	Unsigned16	模拟量原始值	---	1436	DEC
2	250206	Integer16	模拟输入有效数据	---	2.410	V
3	250201	Integer16	模拟输入校正增益		9010	DEC
17	606300	Integer32	实际位置	---	24083	inc

3. 当模拟输入有效值位 3.627 V 时，大于模拟输入最大值，电机将运行至模拟量位置最大值，即 30000 inc 处。

1	250106	Unsigned16	模拟量原始值	---	1209	DEC
2	250206	Integer16	模拟输入有效数据	---	3.627	V
3	250201	Inteaer16	模拟输入校正增益		9010	DEC
17	606300	Integer32	实际位置	---	29999	inc

6.3 力矩模式 (4)

力矩模式即电机的输出力矩被设定为目标值，并通过控制系统来调整电机的输出以使其尽可能地接近或保持在设定的目标力矩上。

6.3.1 内部指令设置力矩

表 6-8 力矩模式常用对象列表

索引	数据类型	参数名称	参数描述	设定值	单位
606000	Integer8	工作模式	用于设置电机工作模式 4: 力矩模式	4	DEC
604000	Unsigned16	控制字	使能驱动器	0x0F	HEX
607100	Integer16	目标扭矩%	目标扭矩/额定扭矩*100%	用户 自定义	%
608000	Unsigned32	最大速度限制 rpm	力矩模式下可执行的最大速度	用户 自定义	rpm

若使用上位机进行操作，可按照以下步骤

“参数列表” -> “基本参数” -> 对控制字、工作模式等参数进行设置



图 6-7 力矩模式相关参数

6.3.2 模拟输入设置力矩

表 6-9 模拟输入设置力矩相关参数

索引	数据类型	参数名称	参数描述	设定值	单位
250106	Unsigned16	模拟量原始值	模拟量原始值	只读	DEC
250206	Integer16	模拟输入有效数据	模拟量输入信号滤波后的值	用户自定义	V
250201	Integer16	模拟输入校正增益	对模拟输入信号进行校准的比例因子		DEC
250202	Integer16	模拟输入校正偏移	用于校准模拟输入信号的偏移量		DEC
250203	Unsigned8	模拟输入滤波系数	模拟量输入信号滤波参数		DEC
250204	Integer16	模拟输入偏移	模拟输入偏移电压		V
250205	Integer16	模拟输入死区	模拟输入死区电压		V
25020B	Unsigned8	模拟-力矩控制	模拟量输入信号控制最大力矩 0:关闭 1:Ain1 控制最大力矩		DEC
25020C	Integer16	模拟-力矩因数	将模拟输入信号转换为力矩的比例因子		Ap/v
25020D	Integer16	模拟-最大力矩因数	模拟电压最大力矩系数		Ap/v

假设输入的模拟信号为 V_{in} ，则经过数据处理后得到的为 V_{in_r} ，那么有：

$V_{in_r} = V_{in} - \text{模拟输入偏移电压}$

如果 V_{in_r} 的绝对值大于死区（即 $|V_{in_r}| > \text{死区}$ ），则模拟输入有效数据为：

模拟输入有效数据 = $|V_{in_r}| - \text{死区}$

否则，模拟输入有效数据为 0。

驱动器的目标扭矩=模拟输入有效数据 * 模拟速度因数

模拟输入设置力矩的大小可以参考 6.1.3 模拟输入设置速度的大小章节。

6.4 脉冲模式介绍(-4)

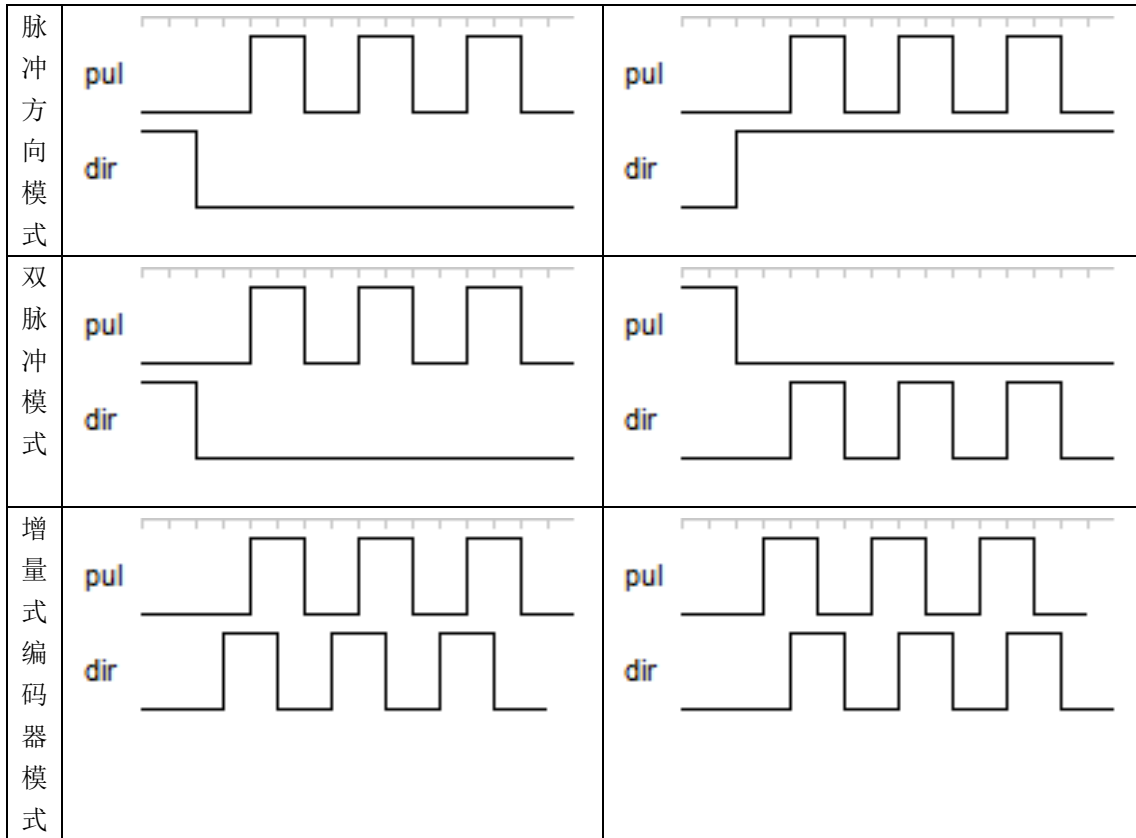
表 6-10 脉冲模式相关参数

索引	数据类型	参数名称	参数描述	设定值	单位
606000	Integer8	工作模式	用于设置电机工作模式 -4: 脉冲模式	-4	DEC
604000	Unsigned16	控制字	使能驱动器	0x2F	HEX
250806	Integer16	原始脉冲频率	从 I/O 口接入的 脉冲频率	只读	kHz
250807	Integer16	电子齿轮换算 脉冲频率	经过电子齿轮 换算的频率		kHz
250804	Integer32	原始脉冲数据	从 I/O 口接入的 脉冲计数值	用户 定义	DEC
250805	Integer32	电子齿轮换算 脉冲数据	经过电子齿轮换算的计 数值 电子齿轮换算脉冲数据= 原始脉冲数据*电子齿轮 比		DEC
250801	Integer16	电子齿轮分子 0	电子齿轮比 = 电子齿轮 分子 / 电子齿轮分母		DEC
250802	Unsigned16	电子齿轮分母 0			DEC
250803	Unsigned8	脉冲模式	0: 脉冲方向 (P/D) 模式 1: 双脉冲 (CW/CCW) 模 式 2: 增量式编码器模式	0、1、2	DEC
250808	Unsigned16	脉冲滤波系数	主编码器口脉冲输入滤 波参数	用户 定义	DEC

注：切换脉冲模式需存储控制参数后重启才能生效

表 6-11 脉冲模式示意图

模 式 名 称	电机正转	电机反转
------------------	------	------



若使用上位机进行操作，可按照以下步骤：

- (1) 在驱动器的 CN3 的 DIR+、DIR-、PUL+、PUL-接入脉冲
- (2) 将驱动器与上位机连接
- (3) 点击“参数列表”->“基本参数”->对相应的参数进行设置
- (4) 修改工作模式、使能驱动

这样就能够通过外部脉冲来控制电机了。

6.5 原点模式 (6)

在电机控制系统中，原点模式是一种运动控制模式，用于确定电机的初始位置，也称为原点位置或零点位置。原点通常是电机旋转的起始点或参考点。

原点模式工作流程如下：

进入原点模式->寻找原点->确定原点->回到原点

表 6-12 原点模式相关参数

索引	数据类型	参数名称	参数描述	设定值	单位
606000	Integer8	工作模式	用于设置电机工作模式 6: 原点模式	6	DEC
604000	Unsigned16	控制字	使能驱动器	F->1F	HEX
609800	Integer8	原点模式	寻找原点的方式选择	用户定义	DEC
609900	Unsigned8	原点速度组	用于执行原点搜索过程的一组参数设置		
609901	Unsigned32	原点转折信号速度	寻找原点开关、限位开关信号时的速度	用户定义	rpm

609902	Unsigned32	原点信号速度	寻找原点信号速度	用户定义	rpm
609903	Unsigned8	上电找原点	每次重新上电后执行一次找原点的功能	0、1	DEC
609904	Integer16	寻找原点最大电流	寻找原点最大电流	用户定义	Arms
609905	Unsigned8	原点偏移模式	原点偏移模式控制 0: 运行到原点偏移 1: 运行到原点事件触发点, 结束后实际位置将变为-原点偏移	0、1	DEC
609906	Unsigned8	原点索引信号盲区	原点索引信号盲区	0、1	DEC
609A00	Unsigned32	原点加速度	寻找原点的加速度		rps/s
607C	Integer32	原点偏移	最终定位距离原点位置的偏移位置设置	用户定义	DEC

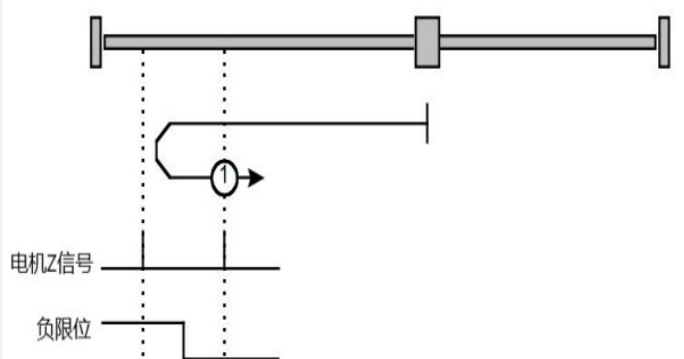
关于原点索引信号盲区：

原点索引信号盲区是在使用原点模式时，为避免相同机器归位结果不同的问题而设置的。它用于归位信号和索引信号非常接近的情况。通过在原点回归前将盲区参数设置为 1，驱动器会自动找到一个合适的盲窗口。这样，在归位过程中，当找到归位信号后，在盲窗口内部的索引信号将被忽略。默认情况下，原点索引信号盲区为 0，但将其设置为 1 会根据索引信号位置自动更改为 0 或 2。如果之后机械设计有变化，只需重新设置此参数为 1 即可。这样做可以确保每次找原点的结果都是相同的，避免由于信号接近导致的非一致性问题。

在寻找原点的过程中有着不同的寻找方式即上述所提到的原点模式，下表为部分原点模式的描述与示意图。

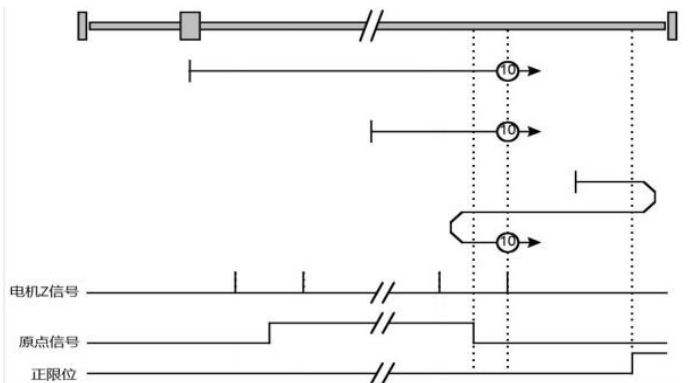
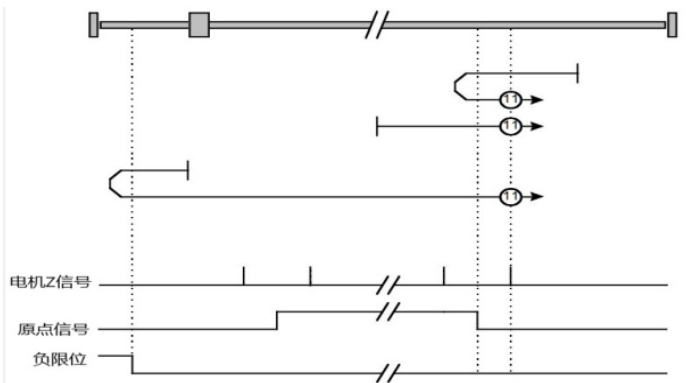
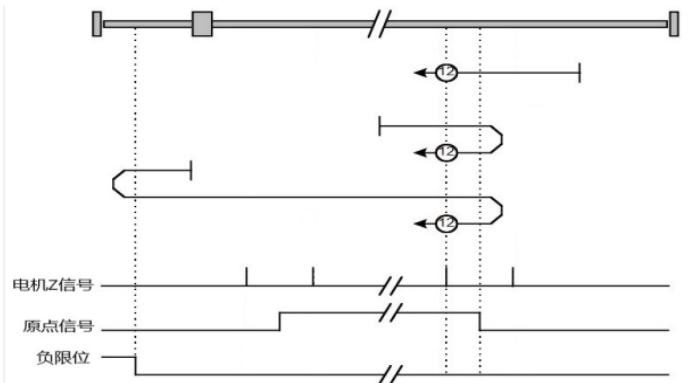
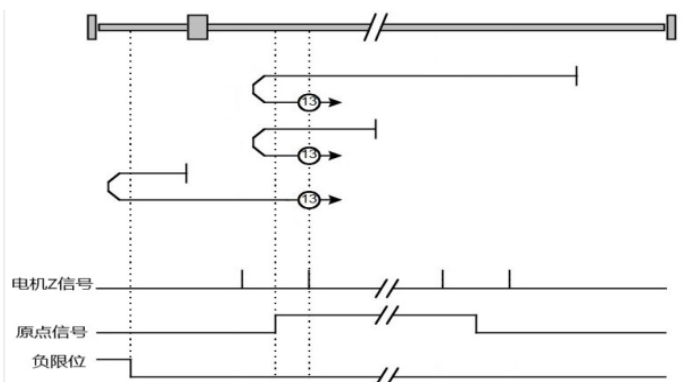
注意：

- (1) 以下列表中的高电平均为有效电平、低电平为无效电平
- (2) 在配置模式时应当注意所有参数的时序而不是只看一条参数线

原点模式	模式描述	示意图
1	负限位为高电平时，遇到索引信号反转；负限位为低电平时，遇到索引信号时为原点	 <p>The diagram illustrates the operation of mode 1. It shows a motor shaft with a limit switch and an index signal. The limit switch is high when the motor is at the negative limit. The index signal is a pulse that occurs when the motor reaches the limit. The diagram shows the motor signal and the limit signal waveforms.</p>

2	<p>正限位为高电平时，遇到索引信号反转；正限位为低电平时，遇到索引信号时为原点</p>	<p>电机Z信号</p> <p>正限位</p>
3	<p>初始方向为正。若原点信号为低电平，则当原点信号变为高电平且遇到索引信号时，电机反转。若原点信号为高电平，则当原点信号变为低电平且遇到索引信号时，标记为原点位置。</p>	<p>电机Z信号</p> <p>原点信号</p>
4	<p>初始方向为正。若原点信号为高电平，则当原点信号变为低电平且遇到索引信号时，电机反转。若原点信号为低电平，则当原点信号变为高电平且遇到索引信号时，标记为原点位置。</p>	<p>电机Z信号</p> <p>原点信号</p>
5	<p>初始方向为负。若原点信号为低电平，则当原点信号变为高电平且遇到索引信号时，电机反转。若原点信号为高电平，则当原点信号变为低电平且遇到索引信号时，标记为原点位置。</p>	<p>电机Z信号</p> <p>原点信号</p>

6	<p>初始方向为负。若原点信号为高电平，则当原点信号变为低电平且遇到索引信号时，电机反转。若原点信号为低电平，则当原点信号变为高电平且遇到索引信号时，标记为原点位置。</p>	
7	<p>电机Z信号与原点信号时序配合参考模式3；若正限位产生上升沿，电机反转，当原点信号产生一个脉冲（010）后遇到索引信号，标记为原点位置</p>	
8	<p>电机Z信号与原点信号时序配合参考模式4；若正限位产生上升沿，电机反转，当原点信号产生一个上升沿后，再产生一个下降沿时电机反转，在反转后遇到索引时标记为原点位置</p>	
9	<p>初始方向为正 ①正限位遇到上升沿电机反转，原点信号产生上升沿后，遇到索引信号标记为原点 ②原点信号产生下降沿时电机反转，原点信号再次产生上升沿后，遇到索引信号标记为原点。</p>	

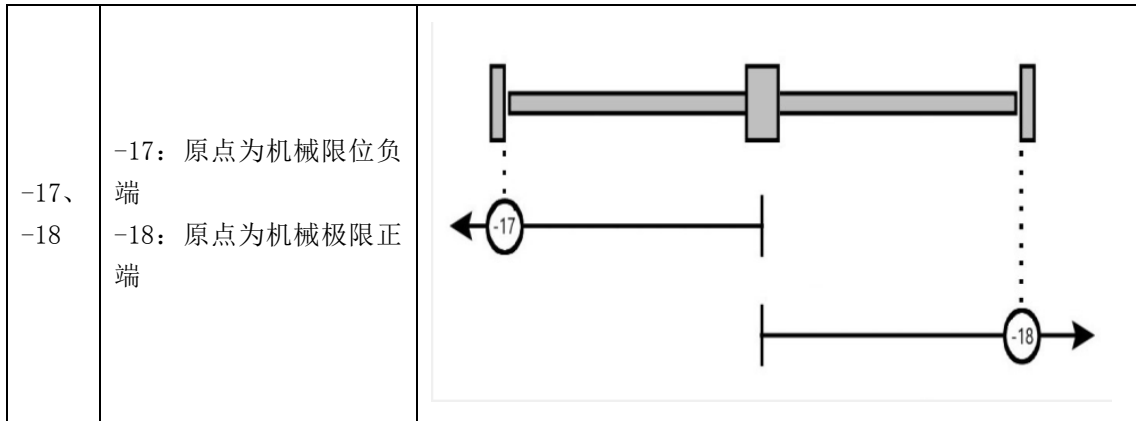
10	<p>初始方向为正</p> <p>①正限位产生上升沿电机反转，原点信号产生上升沿后电机再反转，原点信号再产生下降沿后，遇到索引信号标记为原点</p> <p>②原点信号产生下降沿后，遇到索引信号标记为原点</p>	
11	<p>初始方向为负</p> <p>①负限位产生下降沿后，电机反转，原点信号产生一个脉冲（010）后遇到索引时标记为原点位置；</p> <p>②若原点信号由低电平转为高电平，电机反转，若原点信号再由高电平变为低电平，遇到索引时标记为原点位置</p>	
12	<p>初始方向为负</p> <p>①负限位产生下降沿后，电机反转，原点信号产生一个脉冲（010）后电机反转，若原点信号再由低电平转为高电平，遇到索引标记为原点</p> <p>②若原点信号由高电平转为低电平，电机反转，若原点信号再由低电平变为高电平，遇到索引时标记为原点位置</p>	
13	<p>初始方向为负</p> <p>①负限位产生下降沿后，电机反转，若原点信号由低电平转为高电平，遇到索引标记为原点</p> <p>②若原点信号从高电平变为低电平，电机反转，若原点信号再从高电平变为低电平，遇到索引标记为原点</p>	

<p>14</p>	<p>初始方向为负 ①负限位产生下降沿后，电机反转，若原点信号由低电平转为高电平，电机再次反转，当原点信号从高电平转为低电平时，遇到索引标记为原点 ②若原点信号从高电平变为低电平，电机反转，若原点信号再从高电平变为低电平，遇到索引标记为原点</p>	
<p>17</p>	<p>负限位产生上升沿时，电机反转；负限位产生下降沿时为原点</p>	
<p>18</p>	<p>正限位产生上升沿时，电机反转；正限位产生下降沿时为原点</p>	
<p>19</p>	<p>初始方向为正方向，原点信号产生上升沿时，电机反转；原点信号产生下降沿时为原点</p>	

<p>20</p>	<p>初始方向为正方向，原点信号产生下降沿时，电机反转；原点信号产生上升沿时为原点</p>	
<p>21</p>	<p>初始方向为负方向，原点信号产生上升沿时，电机反转；原点信号产生下降沿时为原点</p>	
<p>22</p>	<p>初始方向为负方向，原点信号产生下降沿时，电机反转；原点信号产生上升沿时为原点</p>	
<p>23</p>	<p>初始方向为正 ①正限位产生上升沿时，电机反转，原点信号产生一个脉冲(010)，将下降沿处标记为原点 ②原点信号产生上升沿电机反转，原点信号再次产生下降沿标记为原点</p>	

<p>24</p>	<p>初始方向为正 ①正限位产生上升沿时，电机反转，原点信号产生一个脉冲(010)电机再次反转，原点信号从低电平转为高电平时标记为原点 ②原点信号产生下降沿，电机反转，再次产生上升沿时，标记为原点</p>	
<p>25</p>	<p>初始方向为正 ①正限位产生上升沿时，电机反转，原点信号从低电平处标记为原点（在上升沿处） ②原点信号产生下降沿电机反转，原点信号再产生上升沿时标记为原点</p>	
<p>26</p>	<p>初始方向为正 ①正限位产生上升沿时，电机反转，原点信号从低电平转为高电平时，电机再次反转，若原点信号再从高电平变为低电平，将下降处标为原点 ②原点信号产生下降处标记为原点</p>	
<p>27</p>	<p>初始方向为负 ①负限位产生上升沿时电机反转，原点信号产生一个脉冲(010)时，再下降沿处标为原点 ②原点信号由低电平转为高电平，电机反转，当原点信号再次由高电平变为低电平时，标记为原点</p>	

<p>28</p>	<p>初始方向为负 ①负限位产生上升沿时电机反转，原点信号产生一个脉冲后再次反转，原点信号由低电平转为高电平时，标记为原点 ②原点信号由高电平转为低电平，电机反转，当原点信号再由低电平转为高电平时，标记为原点</p>	
<p>29</p>	<p>初始方向为负 ①负限位产生上升沿时，电机反转，原点信号产生上升沿时为原点 ②原点信号产生下降沿后电机反转，原点信号再次产生上升沿时标记为原点。</p>	
<p>30</p>	<p>初始方向为负 ①负限位产生上升沿时，电机反转，原点信号产生上升沿时，电机再次反转，原点信号产生下降沿时标记为原点 ②原点信号产生下降沿时标记为原点</p>	
<p>33、 34</p>	<p>电机Z信号的下一个脉冲</p>	
<p>35</p>	<p>以电机当前位置为参考原点</p>	



若采用上位机进行调试则有两种方法：
第一种方法：通过配置回原曲线进行选择：
(1) 点击工作区里的“原点定义”

假设我们需要 原点信号——原点开关信号
使用限制开关
电机停止位置——负方向原点信号下降沿处
回原参数设置为默认值
那么我们可以参照以下步骤
第一步：设置回原参数
第二步：按照需求对回原曲线进行配置
第三步：点击“写入”

回原曲线配置

原点信号

原点转折信号

初始搜寻方向 使用限位开关

电机停止位置

实际原点模式

预设原点模式

回原参数设置

原点偏移(DEC) 搜索原点转折速度(rpm) 找原点电流(Ap) 上电自动找原点

原点加速度(rps/s) 低速搜索原点速度(rpm) 原点偏移模式

描述:
原点模式17-30: 不使用索引信号找原点
原点模式17-30跟原点模式1-14相似, 差别在原点位置不以索引信号决定, 而仅仅依靠原点开关以及限位开关。例如, 模式19和20跟模式3和4对应。

回原参数可自行配置

按照需要配置回原曲线
点击“写入”

图 6-8 原点模式配置

(2) 点击“IO 设置”->将 DIN1、DIN2 分别设置为原点信号和正限位（实例中选择电平属性为常开，记得给先通电）



图 6-9 IO 口设置

(3) 在基本参数里面将工作模式设置为 6，控制字体从 0x0F->0x1F



图 6-10 基本参数设置

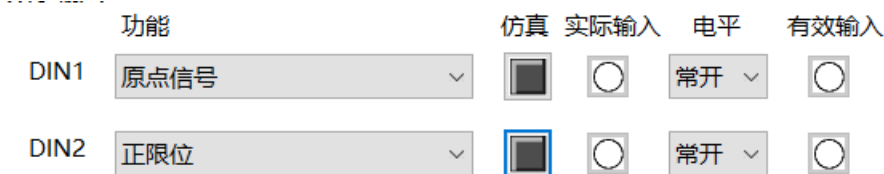
(4) 通过修改 IO 口输入的值就可以寻找原点了。

注意：

若使用仿真，限位信号需要取反（原点信号不需要取反），如模式 24 里面正限位产生一个上升沿便可以使得电机反转，那么我们需要将 IO 口从如下状态



修改为



以此产生一个上升沿将电机反转，其余模式以此类推。

第二种方法：直接写入模式进行选择

第二种方法与第一种方法并没有太大差别，第二种方法适合于比较熟悉原点模式的人操作，进行直接写入省去了部分时间。

回原曲线配置

原点信号

原点转折信号

初始搜寻方向 使用限位开关

电机停止位置

实际原点模式

预设原点模式

回原参数设置

原点偏移(DEC) 搜索原点转折速度(rpm) 找原点电流(Ap) 上电自动找原点

原点加速度(rps/s) 低速搜索原点速度(rpm) 原点偏移模式

此方法跳过回原曲线的配置,
直接进行写入

描述:
 原点模式17-30: 不使用索引信号找原点
 原点模式17-30跟原点模式1-14相似, 差别在原点位置不以索引信号决定, 而仅仅依靠原点开关以及限位开关。例如, 模式19和20跟模式3和4对应。

图 6-11 直接写入原点模式

第七章 性能调节

7.1 速度环整定

表 7-1 速度环整定相关参数

索引	数据类型	参数名称	参数描述	单位
60F901	Unsigned16	速度环比例增益 0	用于设定速度环的跟随速度	DEC
60F902	Integer32	速度环积分增益 0	用来消除速度控制中的静态误差的参数	DEC
60F907	Integer32	速度环 Kvb 阈值	速度环退积分电流阈值	Ap
60F908	Integer32	速度环积分限制	速度环积分限制	Ap
60F915	Unsigned8	目标速度滤波	目标速度滤波	rpm
60F903	Unsigned8	陷波滤波器	速度环的陷波滤波频率设定 BW=Notch_N*10+100[Hz]	Hz
60F904	Unsigned8	陷波滤波器控制	速度环的陷波滤波器控制	DEC
60F905	Unsigned8	速度反馈滤波	速度环的速度反馈滤波 BW=Speed_Fb_N*20+100[Hz]	Hz
60F906	Unsigned8	速度反馈模式	速度反馈模式选择 位 0: 直接反馈 位 1: 观测器反馈 位 2: 低通反馈阶数, 0 = 2 阶, 1 = 1 阶 位 4: 速度指令滤波器, 0 = 不开启, 1 = 开启 默认值: 0, 即 2 阶低通滤波	HEX
60F90A	Integer32	速度到窗门	速度误差窗口	inc/16.38 DEC rpm
60F91C	Unsigned16	失速判断时间	失速判断时间	ms
201018	Unsigned16	输出口状态	bit0: Dout1 bit1: Dout2 bit2: Dout3 ...	HEX
60F914	Integer16	陷波滤波器-输入	陷波滤波器的输入	DEC

在进行速度环整定时，我们主要用到两个参数——速度环比例增益和速度环积分增益。它们共同影响着系统的响应特性和稳定性。它们如何配合使用取决于系统的具体需求和特点。

速度环比例增益：比例增益控制了系统对于速度误差的响应强度，即控制了速度环的动态特性。增大比例增益会增加系统的响应速度，但也可能导致超调量的增加和系统不稳定。

速度环积分增益：积分增益主要用于消除速度环中的静态误差，确保系统在稳态下达到准确的速度跟踪。增大积分增益可以减小静态误差，但也可能导致系统的超调和振荡。

在配合使用这两个参数时，可以遵循以下原则：

先调节速度环比例增益： 从一个相对较小的速度环比例增益开始，逐渐增大比例增益，以观察系统的响应。增大比例增益可以提高系统的响应速度，但要注意当增益过大时，可能会引发超调和不稳定。

再调节速度环积分增益： 一旦达到较为满意的速度响应，可以开始逐步增大速度环的积分增益，以减小静态误差。但要小心过大的积分增益可能导致系统振荡。

观察超调和稳定性： 在调节的过程中，需要观察系统的超调量和稳定性。如果超调量过大或者系统产生振荡，可能需要适当降低速度环比例增益和积分增益。

平衡响应和稳定性： 调节时要在响应速度和稳定性之间寻找平衡。如果需要更快的响应速度，可以适当增大比例增益；如果需要更低的静态误差，可以适当增大积分增益。但要避免过大的参数设置导致系统不稳定。

总之，速度环比例增益和速度环积分增益需要在实际应用中进行逐步调整，并观察系统的响应和稳定性。根据系统的需求，平衡响应速度和稳定性，找到最佳的参数设置。

下面是在速度模式下，电机从 0rpm 以 50rps/s 的加速度加速到 100rpm 时，不同 KP 和 KI 对速度的影响。

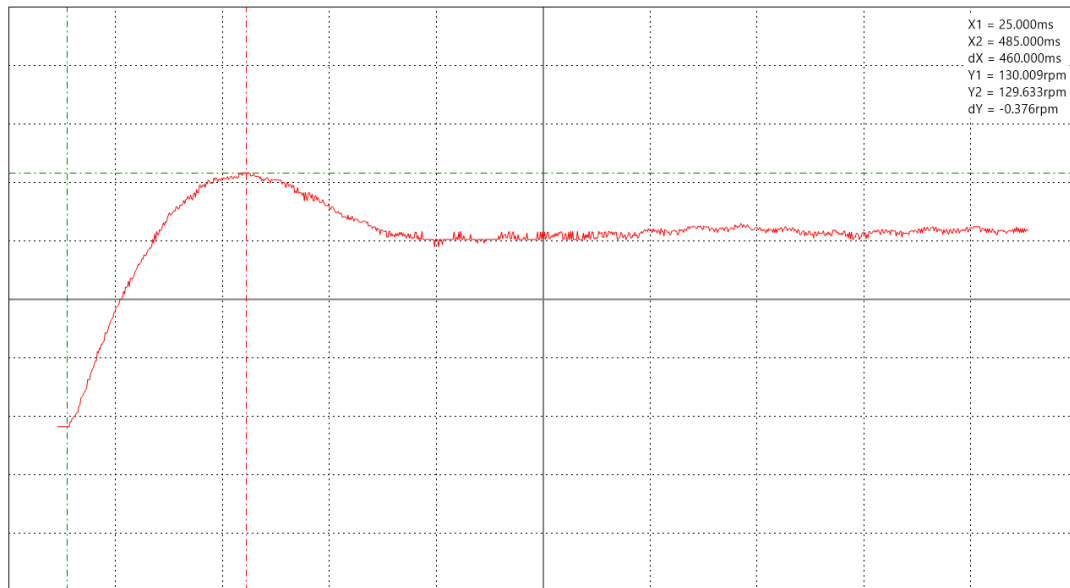


图 7-1 KP=100 KI=0

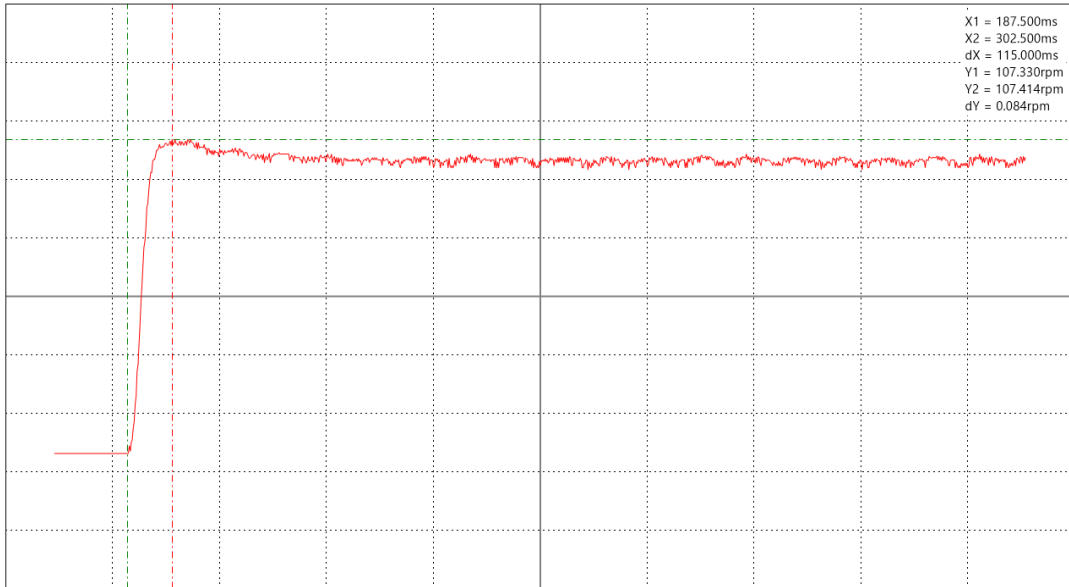


图 7-2 KP=1000 KI=0

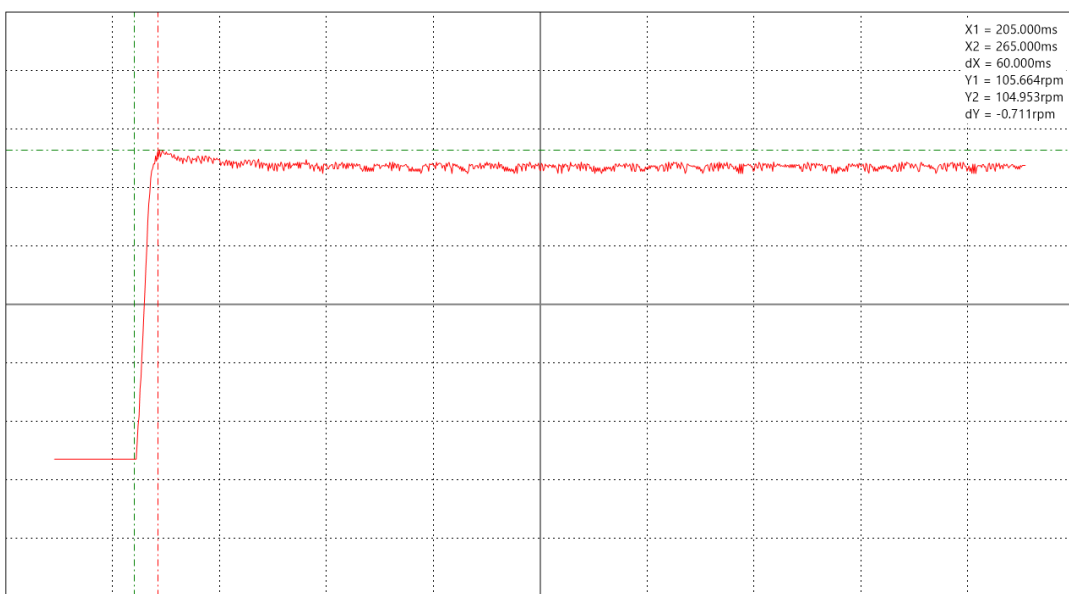


图 7-3 KP=2000 KI=0

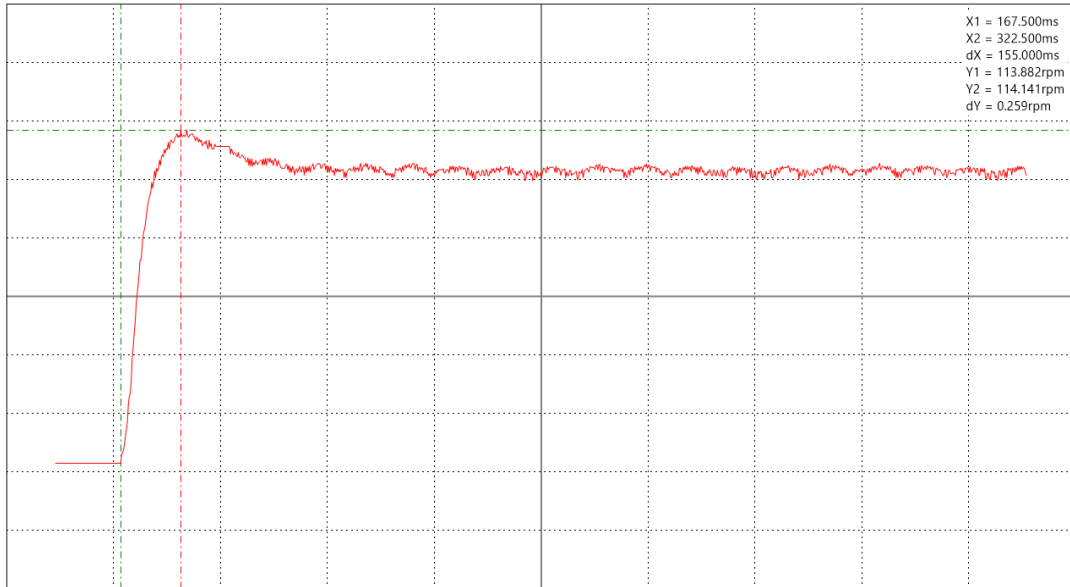


图 7-4 KP=500 KI=0

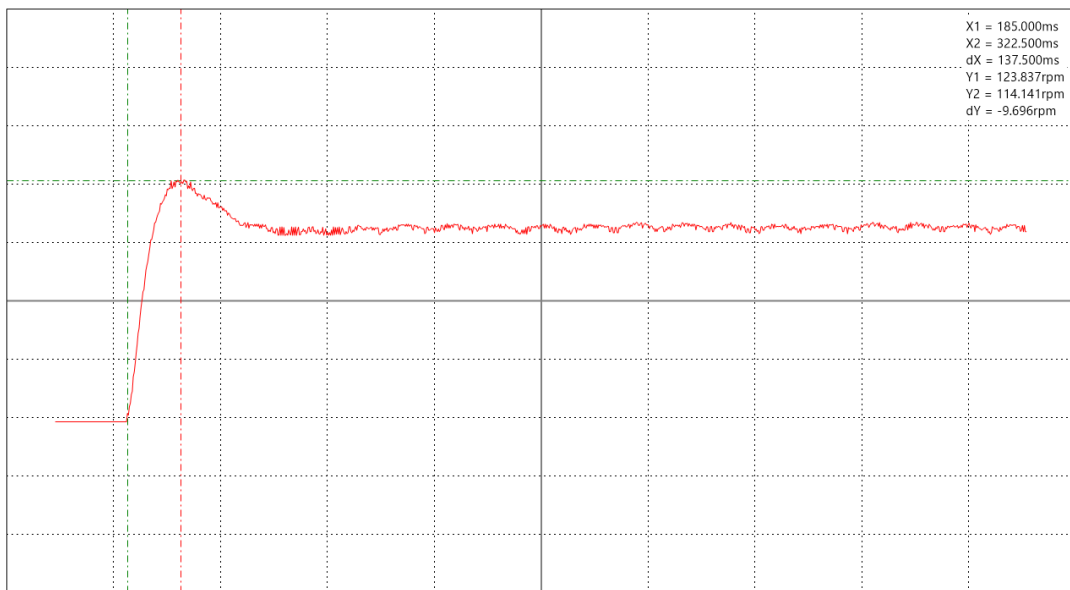


图 7-4 KP=500 KI=20

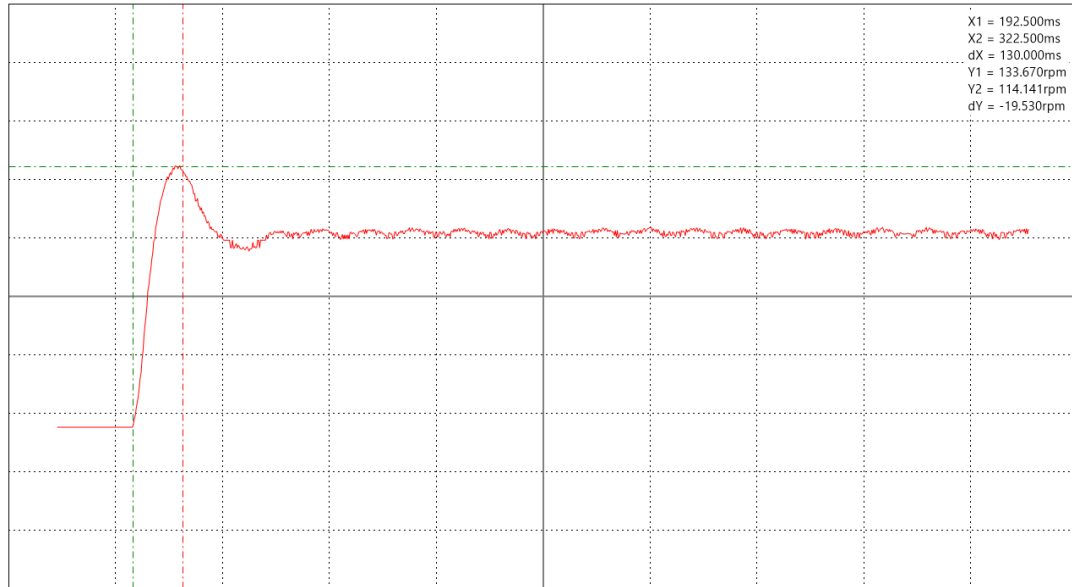


图 7-5 KP=500 KI=60

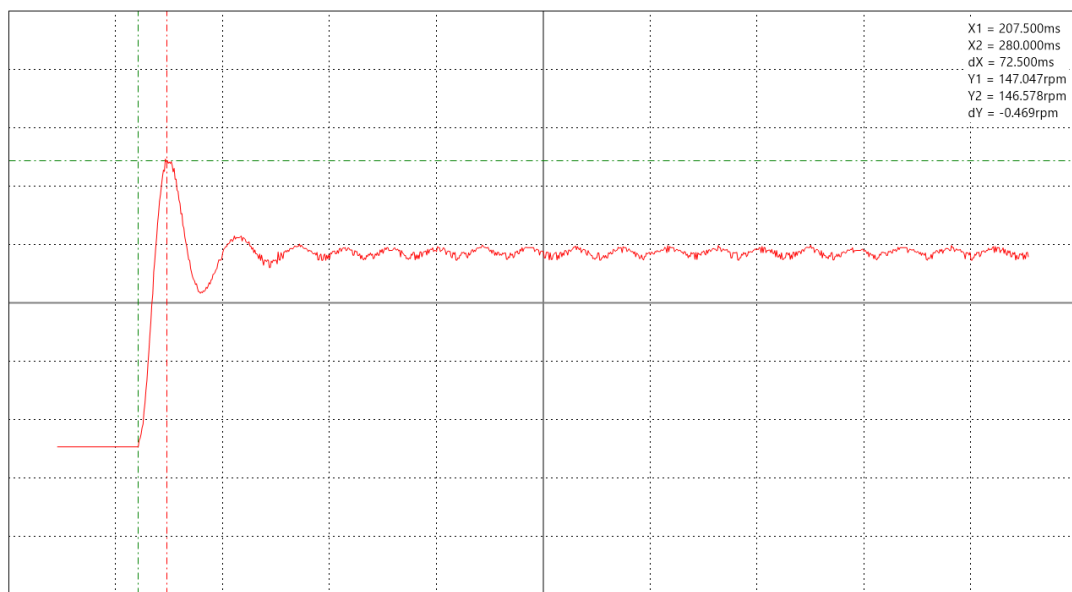


图 7-6 KP=500 KI=200

7.2 位置环整定

表 7-2 位置环整定相关参数

索引	参数名称	参数描述	取值范围	单位
60FB01	Integer16	位置环比例增益 0	位置环 kp	Hz
60FB02	Integer16	位置环速度前馈	位置环前馈	%
60FB03	Integer16	位置环加速度前馈	位置环的加速度前馈	DEC
60FB05	Unsigned8	平滑滤波	平滑滤波器参数调整	DEC
606500	Unsigned32	跟随误差窗口	跟随误差大于 606500h, 且时间超过 606600h, 报跟随误差错误	inc

606600	Unsigned16	跟随误差超时时间	跟随误差超时时间	ms
250809	Integer16	保留		DEC
606700	Unsigned32	位置到窗口	位置指令与实际位置的差值小于位置到窗口，且时间达到位置到窗口时间，认为位置到达	inc
60F400	Integer32	位置跟随误差	位置跟随误差	inc

7.2.1 位置环整定具体步骤如下

与速度环整定相似，我们常用位置环速度前馈和位置环比例增益两个参数进行整定，它们可以协同工作以提升系统的定位性能。大概步骤如下：

初始设置： 首先，将位置环比例增益和位置环速度前馈都设置为较小的初始值，以确保系统在调整过程中不会产生过大的超调或不稳定。

增大位置环比例增益： 逐步增大位置环比例增益，观察系统的定位性能是否有所改善。较大的比例增益可以使系统更快地响应位置误差，减小定位误差。

观察过冲和稳定性： 随着位置环比例增益的增大，观察系统的定位响应是否变得更快，同时注意是否出现了过冲现象以及稳定性是否受到影响。

增大位置环速度前馈： 在适当的比例增益下，逐步增大位置环速度前馈。速度前馈可以提前预测速度指令，进一步减小定位误差，特别是在变速或急停等情况下。

观察性能改善： 增大位置环速度前馈后，观察系统的定位性能是否有所改善。特别注意是否减小了位置误差，并且是否在快速变化的指令下保持了良好的稳定性。

参数协调与微调： 位置环速度前馈和位置环比例增益是相互关联的，它们的调节会相互影响。在微调过程中，根据实际情况，可能需要多次来回调节两个参数，以达到最佳的定位性能和稳定性。

7.3 综合调整

位置环整定和速度环整定是控制系统中两个重要的环节，它们相互影响，需要合理配合使用以实现优良的控制性能。以下是它们配合使用的一般步骤：

确定整定顺序： 通常情况下，应首先进行速度环整定，然后再进行位置环整定。这是因为位置环是在速度环的基础上进行控制，所以首先确保速度环的稳定性和性能是关键。

速度环整定： 调整速度环比例增益和积分增益，以达到快速的速度响应和稳定的速度跟踪。观察速度环的超调、稳定性和响应时间，并根据需求进行微调。

位置环整定： 在速度环稳定的基础上，调整位置环的比例增益。增大比例增益可以提高定位响应速度，但要注意超调现象。设置位置环的速度前馈参数，以减小位置误差，特别是在变速和急停等情况下。根据实际应用需求，调整位置环的积分增益，以消除静态误差。

整定过程可能需要多次迭代调整，以达到最佳的控制性能。根据实际测试结果，根据需要微调速度环和位置环的参数，直到达到满意的效果。

第八章 报警排除

8.1 驱动器 LED 灯报警排除

驱动器正常运行，LED 绿灯亮起；假若 LED 红灯亮起，则可以通过连接上位机进一步判断故障原因，可通过第五章 5.6 历史错误与报警 按步骤查看错误代码，再前往 8.2 代码报警排除 进一步解决问题。

8.2 代码报警排除（603F00）

错误代码	错误名称	错误原因	改正措施
FF40	编码器 ABZ 连接报警/通讯式编码器断线错误	<ol style="list-style-type: none"> 1. 编码器 ABZ 接线错误、编码器接头松动，ABZ 已损坏 2. 通讯式编码器接通松动、线序错误、编码器损坏，驱动器编码器 5V 输出损坏 	<ol style="list-style-type: none"> 1. 检查编码器线缆是否正确连接 2. 检查编码器接头是否松动 3. 检查编码器是否损坏 4. 更换电机或编码器 5. 检查编码器电源是否完好
FF41	编码器 UVW 连接报警/通讯式编码器多圈错误	<ol style="list-style-type: none"> 1. 霍尔 UVW 接线错误、霍尔接头松动，霍尔已损坏 2. 编码器曾经断开过所有电源(包含编码器电池) 	<ol style="list-style-type: none"> 1. 检查霍尔线缆是否正确连接 2. 检查霍尔结构是否松动 3. 检查霍尔是否损坏 4. 更换霍尔或编码器 5. 检查编码器电池电压，重新复位多圈错误
7305	编码器计数错误/通讯式编码器 CRC 错误	编码器受到干扰或反馈周期设置不正确	<ol style="list-style-type: none"> 1. 检查驱动器接地线是否连接良好 2. 检查设备的接地线是否良好 3. 使用独立电源给驱动器供电
4210	驱动器温度过高	驱动器功率模块温度达到报警值	<ol style="list-style-type: none"> 1. 增加风扇，改善电气柜散热环境 2. 适当增加驱动器安装距离 3. 检查电机与驱动器选型是否正确
3210	总线电压过高	<ol style="list-style-type: none"> 1. 电源电压超过允许的输入范围 2. 未接制动电阻 3. 制动电阻不匹配 	<ol style="list-style-type: none"> 1. 检查电源电压是否高于驱动器允许输入范围 2. 检查电源、电压是否稳定

			<ul style="list-style-type: none"> 3. 确认是否在减速时报错,若是,请考虑增加制动电阻 4. 确认负载惯量,重新评估制动电阻选型
3220	总线电压过低	<ul style="list-style-type: none"> 1. 电源电压低于允许的输入范围 2. 电源功率过低 	<ul style="list-style-type: none"> 1. 检查电源功率是否满足要求 2. 更换更大功率的电源
2320	驱动器输出短路	<ul style="list-style-type: none"> 1. 驱动器 UVW 和 PE 输出端 2. 存在短路问题 	<ul style="list-style-type: none"> 1. 检查电机动力线缆连接是否正确 2. 驱动器可能已损坏,请考虑更换驱动器
7110	制动电阻异常	未正确设置制动电阻参数	无
8611	跟随误差过大	<ul style="list-style-type: none"> 1. 控制环刚性太小 2. 电机相序不正确 3. 驱动器或电机功率太小 4. 最大跟随误差值太小 	<ul style="list-style-type: none"> 1. 适当增大“速度环比例增益”“位置环比例增益” 2. 更换电机 UV 相接线测试 3. 更改更大功率的电机和驱动器 4. 适当增大“最大跟随误差”
5112	逻辑低压	逻辑电压低于 18V, 电源电压被拉低	<ul style="list-style-type: none"> 1. 检查电源输出功率是否满足要求 2. 更换输出功率更大的电源
2350	电机或驱动器 iit 故障	<ul style="list-style-type: none"> 1. 机械装置被卡住或摩擦力过大 2. 电机相序不正确 3. 电机或驱动器功率太小 	<ul style="list-style-type: none"> 1. 检查电机是否带有抱闸,确认抱闸是否正常松开 2. 将驱动器断电,或将电机轴与负载脱开,检查电机和负载移动是否顺畅 3. 更换更大功率电机与驱动器
7310	速度跟随超差	<ul style="list-style-type: none"> 1. 控制环刚性太小 2. 速度跟随超差阈值过小 3. 电机线序不正确 4. 编码器信号出问题 	<ul style="list-style-type: none"> 1. 适当增大“速度环比例增益”“速度环积分增益” 2. 更换电机 UV 相接线测试 3. 适当增大“速度跟随超差阈值”

			4. 检查或更换编码器
4310	电机温度过高	电机温度超过报警值	1. 降低环境温度, 提高冷却条件 2. 降低电机加速度与减速度 3. 减小负载
7122	电机励磁错误/通讯式编码器其他错误(请参考30030208)	1. 电机 UVW 相序不正确 2. 编码器未连接	1. 交换U相和V相电机线 2. 检查编码器接否是否松动
5530	EEPROM 错误	驱动器上电初始化时, 读取 EEPROM 数据出错, 驱动器收到干扰, 或更新新的固件	1. 先初始化参数->存储控制环参数->存储电机参数->重启 2. 重新设置参数或导入参数文件 3. 以上两步不行请联系供应商
5210	电流传感器异常	电流传感器信号偏移或纹波太大	电流传感器损坏, 请联系供应商
2214	软件过流	电流 ADC 采样接近最大值	1. 适当减小“目标电流限制” 2. 或者联系供应商
8613	原点出错	无	无
3130	电机缺相	电机 UVW 线序松动或者漏接电机线	检查电机 UVW 相线缆是否松动或漏接
6320	电机数据出错	为设置电机型号, 或者电机型号不存在	正确设置电机型号, 保存电机参数并重启
ff10	用户 iit 故障	请参数电机或驱动器 iit 故障原因	请参数电机或驱动器 iit 故障
5443	预使能错误	I/O 输入口设置了“预使能”, 在给驱动器发使能命令之前, 相应输入口未收到信号	1. 不需要“预使能”功能时, 请将对应 I/O 动能取消 2. 需要“预使能”功能时, 请先在相应 I/O 口给硬件使能信号, 然后给驱动器发使能命令
5442	到达正限位	正限位信号被触发	检查排除触发限位的原因
5441	到达负限位	负限位信号被触发	检查排除触发限位的原因
FF30	脉冲频率过高	输入的脉冲频率超过允许值	检查输入脉冲频率

7500	通讯总线掉线	通讯总线松动, 或者控制器未按时给驱动器发送心跳信号	1. 根据控制器的心跳信号, 正确设置总线掉线时间 2. 检查控制器心跳信号是否按时发出
FF42	MCU 看门狗错误	无	请联系供应商

8.3 错误字报警排除

8.3.1 错误字 1 (260100)

错误位	错误名称	错误原因	改正措施
bit0	参考 26020010		
bit1	编码器 ABZ 连接报警/通讯式编码器断线错误	1. 编码器 ABZ 接线错误、编码器接头松动, ABZ 已损坏 2. 通讯式编码器接通松动、线序错误、编码器损坏, 驱动器编码器 5V 输出损坏	1. 检查编码器线缆是否正确连接 2. 检查编码器接头是否松动 3. 检查编码器是否损坏 4. 更换电机或编码器 5. 检查编码器电源是否完好
bit2	编码器 UVW 连接报警/通讯式编码器多圈错误	1. 霍尔 UVW 接线错误、霍尔接头松动, 霍尔已损坏 2. 编码器曾经断开过所有电源(包含编码器电池)	1. 检查霍尔线缆是否正确连接 2. 检查霍尔结构是否松动 3. 检查霍尔是否损坏 4. 更换霍尔或编码器 5. 检查编码器电池电压, 重新复位多圈错误
bit3	编码器计数错误/通讯式编码器 CRC 错误	编码器受到干扰或反馈周期设置不正确	1. 检查驱动器接地线是否连接良好 2. 检查设备的接地线是否良好 3. 使用独立电源给驱动器供电
bit4	驱动器温度过高	驱动器功率模块温度达到报警值	1. 增加风扇, 改善电气柜散热环境 2. 适当增加驱动器安装距离 3. 检查电机与驱动器选型是否正确
bit5	总线电压过高	1. 电源电压超过允许	1. 检查电源电压是否

		的输入范围 2. 未接制动电阻 3. 制动电阻不匹配	高于驱动器允许输入范围 2. 检查电源 电压是否稳定 3. 确认是否在减速时报错, 若是, 请考虑增加制动电阻 4. 确认负载惯量, 重新评估制动电阻选型
bit6	总线电压过低	1. 电源电压低于允许的输入范围 2. 电源功率过低	1. 检查电源功率是否满足要求 2. 更换更大功率的电源
bit7	驱动器输出短路	1. 驱动器 UVW 和 PE 输出端 2. 存在短路问题	1. 检查电机动力线缆连接是否正确 2. 驱动器可能已损坏, 请考虑更换驱动器
bit8	制动电阻异常	未正确设置制动电阻参数	无
bit9	跟随误差过大	控制环刚性太小 电机相序不正确 驱动器或电机功率太小 最大跟随误差值太小	1. 适当增大“速度环比例增益”“位置环比例增益” 2. 更换电机 UV 相接线测试 3. 更改更大功率的电机和驱动器 4. 适当增大“最大跟随误差”
bit10	逻辑低压	逻辑电压低于 18V, 电源电压被拉低	1. 检查电源输出功率是否满足要求 2. 更换输出功率更大的电源
bit11	电机或驱动器 iit 故障	1. 机械装置被卡住或摩擦力过大 2. 电机相序不正确 3. 电机或驱动器功率太小	1. 检查电机是否带有抱闸, 确认抱闸是否正常松开 2. 将驱动器断电, 或将电机轴与负载脱开, 检查电机和负载移动是否顺畅 3. 更换更大功率电机与驱动器
bit12	速度跟随超差	1. 控制环刚性太小 2. 速度跟随超差阈值	1. 适当增大“速度环比例增益”“速度环

		过小 3. 电机线序不正确 4. 编码器信号出问题	积分增益” 2. 更换电机 UV 相接 测试 3. 适当增大“速度跟 随超差阈值” 4. 检查或更换编码器
bit13	电机温度过高	电机温度超过报警值	1. 降低环境温度, 提高 冷却条件 2. 降低电机加速度与 减速度 3. 减小负载
bit14	电机励磁错误/通讯式编 码器其他错误(请参考 30030208)	1. 电机 UVW 相序不正 确 2. 编码器未连接	1. 交换U相和V相电机 线 2. 检查编码器接是否 松动
bit15	Eeprom 报警	驱动器上电初始化时, 读取 eeprom 数据出 错, 驱动器收到干扰, 或更新新的固件	1. 先初始化参数->存 储控制环参数->存储 电机参数->重启 2. 重新设置参数或导 入参数文件 3. 以上两步不行请联 系供应商

8.3.2 错误字 2 (260200)

错误位	错误名称	错误原因	改正措施
bit0	电流传感器异常	电流传感器信号偏移 或纹波太大	电流传感器损坏, 请联 系供应商
bit1	软件过流	电流 ADC 采样接近最 大值	1. 适当减小“目标电 流限制” 2. 或者联系供应商
bit2	找原点错误	无	无
bit3	电机缺相	电机 UVW 线序松动或 者漏接电机线	检查电机 UVW 相线缆 是否松动或漏接
bit4	电机配置	为设置电机型号, 或者 电机型号不存在	正确设置电机型号, 保 存电机参数并重启
bit5	用户 iit	请参照电机或驱动器 iit 故障原因	请参照电机或驱动器 iit 故障
bit6	保留		
bit7	保留		
bit8	预使能错误	I0 输入口设置了“预 使能”, 在给驱动器发 使能命令之前, 相应输 入口未收到信号	1. 不需要“预使能” 功能时, 请将对应 I0 动能取消 2. 需要“预使能”功

			能时，请先在相应 I/O 口给硬件使能信号，然后给驱动器发使能命令
bit9	正限位错误	正限位信号被触发	检查排除触发限位的原因
bit10	负限位错误	负限位信号被触发	检查排除触发限位的原因
bit11	脉冲频率过高	输入的脉冲频率超过允许值	检查输入脉冲频率
bit12	总线掉线错误	通讯总线松动，或者控制器未按时给驱动器发送心跳信号	1. 根据控制器的心跳信号，正确设置总线掉线时间 2. 检查控制器心跳信号是否按时发出
bit13	全闭环下编码器计数方向错误	1. A 相和 B 相信号接错 2. 错误编码器类型的设置 3. 编码器的 A 相和 B 相信号之间存在相位差	1. 检查编码器接线是否正确 2. 检查编码器类型设置是否正确 3. 检查编码器分辨率是否设置错误 4. 检查反馈信号是否存在相位差
bit14	主编码器连接错误	主编码器连接错误	检查编码器接口是否正常连接
bit15	主编码器计数错误	编码器受到干扰或反馈周期设置不正确	1. 检查驱动器接地线是否连接良好 2. 检查设备的接地线是否良好 3. 使用独立电源给驱动器供电

第九章 常用对象列表

对象列表说明

对象属性中，包含了数据类型、操作权限以及参数单位

其中操作权限有：R——可读

W——可写

若对象属性一栏中没有参数单位，则表示参数单位默认为内部单位（DEC）

注意：

对参数进行写操作，应当注意单位换算，即将需要的数据换算成内部单位（DEC）。具体参数相关换算可以参考本章最后一节 单位换算

9.1 常用对象列表

9.1.1 控制参数

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
控制字	604000	60400010	0x7400	Unsigned16 RW	bit0: Switch on bit1: Enable voltage bit2: Quick stop bit3: Enable operation bit4: Set-point(1 模式), Homing operation start(6 模式), Enable ip mode(7 模式) bit5: Change set immediately(1 模式) bit6: 0:related 1:abs (1 模式) bit7: Fault reset bit8: Halt bit9/bit10: Reserved bit11~bit15:Manufacturer specific
状态字	604100	60410010	0x7410	Unsigned16 R	bit0: Ready to switch on bit1: Switch on bit2: Operation enabled bit3: Fault bit4: Voltage enabled bit5: Quick stop bit6: Switch on disabled bit7: Warning bit8: Manufacturer specific bit9: Remote

					bit10: Target reached bit11: Internal limit active 限位中 bit12: Set-point ack(1 模式), speed=0(3 模式), Homing attained(6 模式), Ip-Mode active(7 模式) bit13: Following error(1 模 式), Homing error(6 模式) bit14: speed=0 bit15: Manufacturer specific
工作 模式	606000	60600008	0x7600	Integer8 R	工作模式 -4: 脉冲模式 -3: 立即速度模式 1: 轮廓位置模式 3: 轮廓速度模式 4: 转矩模式 6: 原点模式 7: 插补模式

9.1.2 DIN 模式

参数 名称	UART 通讯 地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
多段位置 控制 0	202001	20200120	0x4601	Integer32 RW	通过 I0 口的输入选择电机的 绝对位置 如 I0 口的多段位置控制 2-多段 位置控制 1-多段位置控制 0 分别 为 010 时, 表示电机将会运行到 多段位置控制 2 的位置。
多段位置 控制 1	202002	20200220	0x4602		
多段位置 控制 2	202003	20200320	0x4603		
多段位置 控制 3	202004	20200420	0x4604		
多段位置 控制 4	202005	20200520	0x4605		
多段位置 控制 5	202006	20200620	0x4606		
多段位置 控制 6	202007	20200720	0x4607		
多段位置 控制 7	202008	20200820	0x4608		

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
多段速控制 0	202009	20200920	0x4609	Integer32 RW rpm	通过 IO 口的输入选择电机轮廓运行速度，如 IO 口的多段速控制 2-多段速控制 1-多段速控制 0 分别为 010 时，表示电机将会以多段速控制 2 的速度运行。
多段速控制 1	20200A	20200A20	0x460A		
多段速控制 2	20200B	20200B20	0x460B		
多段速控制 3	20200C	20200C20	0x460C		
多段速控制 4	20200D	20200D20	0x460D		
多段速控制 5	20200E	20200E20	0x460E		
多段速控制 6	20200F	20200F20	0x460F		
多段速控制 7	202010	20201020	0x4610		

9.1.3 输入输出参数

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
输入口极性	201001	20100110	0x4201	Unsigned16 RW	输入信号极性定义 bit0: Din1 bit1: Din2 bit2: Din3 ...
输入模拟	201002	20100210	0x4202	Unsigned16 RW	输入信号模拟 bit0: Din1 bit1: Din2 bit2: Din3 ...
输入 1 功能	201003	20100310	0x4203	Unsigned16 RW	数字输入 1 功能定义 (16 进制) 0001: 驱动器使能 0002: 驱动器错误复位 0004: 预使能 0008: Kvi 清零 0010: 正限位 0020: 负限位 0040: 原点信号

					0080: 速度指令反向 0100: 多段速度控制 0 0200: 多段速度控制 1 0400: 多段速度控制 2 0800: 外部输入故障 1000: 紧急停止 2000: 找原点 4000: 保留 8001: 多段位置控制 0 8002: 多段位置控制 1 8004: 多段位置控制 2 8008: 电子齿轮选择 0 8010: 电子齿轮选择 1 8020: 电子齿轮选择 2
输入 2 功能	201004	20100410	0x4204	Unsigned16 RW	数字输入 2 功能定义 (详情请参考 201003 帮助)
输入 3 功能	201005	20100510	0x4205	Unsigned16 RW	数字输入 3 功能定义 (详情请参考 201003 帮助)
输入 4 功能	201006	20100610	0x4206	Unsigned16 RW	数字输入 4 功能定义 (详情请参考 201003 帮助)
输入 5 功能	201007	20100710	0x4207	Unsigned16 RW	数字输入 5 功能定义 (详情请参考 201003 帮助)
输入 6 功能	201008	20100810	0x4208	Unsigned16 RW	数字输入 6 功能定义 (详情请参考 201003 帮助)
输入 7 功能	201009	20100910	0x4209	Unsigned16 RW	数字输入 7 功能定义 (详情请参考 201003 帮助)
输入 8 功能	20100A	20100A10	0x420A	Unsigned16 RW	数字输入 8 功能定义 (详情请参考 201003 帮助)
输入状态	20100B	20100B10	0x420B	Unsigned16 R	bit0: Din1 bit1: Din2 bit2: Din3 ...
输出极性	20100E	20100E10	0x420E	Unsigned16 RW	输出口极性定义
输出模拟	20100F	20100F10	0x420F	Unsigned16 RW	输出口模拟
输出 1 功能	201010	20101010	0x4210	Unsigned16	输出口 1 定义

				RW	0001: 驱动器就绪 0002: 驱动器错误 0004: 电机位置到 0008: 电机零速 0010: 电机抱闸刹车 0020: 电机速度到 0040: 索引信号出现 0080: 力矩模式下达到最大限制速度 0100: 电机锁轴 0200: 出现限位 0400: 已找到原点 0800: 目标扭矩到达
输出 2 功能	201011	20101110	0x4211	Unsigned16 RW	输出口 2 定义(详情请参考 20100F 帮助)
输出 3 功能	201012	20101210	0x4212	Unsigned16 RW	输出口 3 定义(详情请参考 20100F 帮助)
输出 4 功能	201013	20101310	0x4213	Unsigned16 RW	输出口 4 定义(详情请参考 20100F 帮助)
输出 5 功能	201014	20101410	0x4214	Unsigned16 RW	输出口 5 定义(详情请参考 20100F 帮助)
输出 6 功能	201015	20101510	0x4215	Unsigned16 RW	输出口 6 定义(详情请参考 20100F 帮助)
输出 7 功能	201016	20101610	0x4216	Unsigned16 RW	输出口 7 定义(详情请参考 20100F 帮助)
输出 8 功能	201017	20101710	0x4217	Unsigned16 RW	输出口 8 定义(详情请参考 20100F 帮助)
输出口状态	201018	20101810	0x4218	Unsigned16 RW	输出口状态 bit0: Dout1 bit1: Dout2 bit2: Dout3 ...

9.1.4 模拟输入模式

以下参数没有分配 Modbus 地址，常用上位机软件修改参数

参数名称	UART 通讯地址	CANopen 通讯地址	对象属性	参数详情
模拟量原始值	250201	25020110	Unsigned 16R	模拟量原始值
模拟输入校正增益	250202	25020210	Integer16 RW	外部模拟信号校正增益(与硬件参数相关)

模拟输入校正偏移	250203	25020308	Integer16 RW	外部模拟信号校正偏移(与硬件参数相关)
模拟输入滤波系数	250204	25020410	Unsigned8 RW	外部模拟信号滤波系数
模拟输入偏移	250205	25020510	Integer16 RW	外部模拟信号偏移
模拟输入死区	250206	25020610	Integer16 RW	外部模拟信号死区数据设定
模拟输入有效数据	250207	25020708	Integer16 RW	外部模拟信号的滤波输出
模拟 - 速度控制	250208	25020810	Unsigned8 RW	模拟信号控制速度, 3 模式, -3 模式有效 0: 无控制 1: Ain 控制速度
模拟 - 高端死区	250209	25020910	Integer16 RW	模拟量控制时, 实际输入绝对值高于此数据时, 输出将变为 0 默认值 0, 代表无效
模拟 - 低端死区	25020A	25020A20	Integer16 RW	模拟量控制时, 实际输入绝对值低于此数据时, 输出将变为 0 默认值 0, 代表无效
模拟 - 速度因数	25020B	25020B08	Integer32 RW	模拟量到速度系数
模拟 - 力矩控制	25020C	25020C10	Unsigned8 RW	模拟信号控制力矩, 4 模式有效 0: 无控制 1: Ain 作为力矩给定 2: Ain 作为最大力矩给定
模拟 - 力矩因数	25020D	25020D10	Integer16 RW	模拟量到力矩系数
模拟 - 最大力矩因数	250201	25020110	Integer16 RW	模拟量到最大力矩系数

9.1.5 脉冲模式

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
电子齿轮分子 0	250801	25080110	0x5601	Integer16 RW	电子齿轮分子

电子齿轮分母 0	250802	25080210	0x5602	Unsigned16 RW	电子齿轮分母
脉冲模式	250803	25080308	0x5603	Unsigned8 RW	脉冲模式控制 0:脉冲方向模式 1:双脉冲模式 2:增量式编码器模式
原始脉冲数据	250804	25080420	0x5604	Integer32 RW	从 I/O 口接入的脉冲计数值
电子齿轮换算脉冲数据	250805	25080520	0x5605	Integer32 RW	经过电子齿轮换算后的脉冲数据
原始脉冲频率	250806	25080610	0x5606	Integer16 R kHz	从 I/O 口接入的脉冲频率
电子齿轮换算脉冲频率	250807	25080710	0x5607	Integer16 R kHz	经过电子齿轮换算的计数值 电子齿轮换算脉冲数据=原始脉冲数据*电子齿轮比
脉冲滤波系数	250808	25080810	0x5608	Unsigned16 RW	脉冲滤波参数
电子齿轮分子 0	250801	25080110	0x5601	Integer16 RW	电子齿轮比 = 电子齿轮分子 / 电子齿轮分母 与 DIN 模式相似, 通过 I/O 口选择电子齿轮, 如 I/O 口的电子齿轮选择 2-电子齿轮选择制 1-电子齿轮选择 0 分别为 010 时, 表示电机将会使用电子齿轮 2
电子齿轮分母 0	250802	25080210	0x5602	Integer16 RW	
电子齿轮分子 1	250901	25090110	0x5701	Integer16 RW	
电子齿轮分母 1	250902	25090210	0x5702	Unsigned16 RW	
电子齿轮分子 2	250903	25090310	0x5703	Integer16 RW	
电子齿轮分母 2	250904	25090410	0x5704	Unsigned16 RW	
电子齿轮分子 3	250905	25090510	0x5705	Integer16 RW	
电子齿轮分母 3	250906	25090610	0x5706	Unsigned16 RW	

9.1.6 设备站号和波特率

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
设备	2F8001	2F800108	0x6001	Unsigned8	驱动器站号

站号				RW	
CAN 波特率	2F8002	2F800208	0x6002	Unsigned8 RW bps	CAN 波特率设置 100: 1M 50: 500k 25: 250k 12: 125k 10: 100k 5: 50k 2: 20k 需要重启动
UART 波特率	2F8003	2F800308	0x6003	Unsigned8 RW bps	用于设置串口的波特率 设定值 波特率 0: 4800 1: 9600 2: 14400 3: 19200 4: 38400 5: 56000 6: 57600 7: 115200 需要重启动
RS485 波特率	2F8004	2F800410	0x6004	Unsigned16 RW bps	用于设置 RS485 的波特率 48: 4800 96: 9600 192: 19200 384: 38400 576: 57600 1152: 115200 需要重启动

9.1.7 电机状态

参数名称	UAR 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
实际位置	606300	60630020	0x7630	Integer32R	电机当前位置值
实际速度	606C00	606C0020	0x76C0	Integer32R	滤波器后的实际速度
实际电流 i_q	607800	60780010	0x7780	Integer16R	实际电流

9.1.8 速度模式

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
目标速度	60FF00	60FF0020	0x8C00	Integer32 RW rpm	用户预期的速度值
轮廓加速度	608300	60830020	0x7930	Unsigned32 RW rps/s	从当前速度加速到指定速度的加速度
轮廓减速度	608400	60840020	0x7940	Unsigned32 RW rps/s	从当前速度减速到指定速度的减速度

9.1.9 位置模式

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
目标位置	607A00	607A0020	0x77A0	Integer32 RW	位置模式下的定位目标位置
轮廓运行速度	608100	60810020	0x7910	Unsigned32 RW rpm	到达指定位置的速度

9.1.10 力矩模式

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
目标扭矩%	607100	60710010	0x7710	Integer16 RW %	目标扭矩/额定扭矩*100%
最大速度限制 rpm	608000	60800010	0x7900	Unsigned16 RW rpm	力矩模式下，电机能与运行的最大速度

9.1.11 速度环参数

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
速度环比例增益 0	60F901	60F90110	0x8601	Unsigned16 RW	速度环的 kp
速度环	60F902	60F90210	0x8602	Unsigned16	速度环的 ki

积分增益 0				RW	
陷波滤波器	60F903	60F90308	0x8603	Unsigned8 RW	速度环的陷波滤波频率设定 BW=Notch_N*10+100 [Hz]
陷波滤波器控制	60F904	60F90408	0x8604	Unsigned8 RW	速度环的陷波滤波器控制
速度反馈滤波	60F905	60F90508	0x8605	Unsigned8 RW	速度环的速度反馈滤波 BW=Speed_Fb_N*20+100 [Hz]
速度反馈模式	60F906	60F90608	0x8606	Unsigned8 RW	速度反馈模式选择 位 0: 直接反馈 位 1: 观测器反馈 位 2: 低通反馈阶数, 0 = 2 阶, 1 = 1 阶 位 4: 速度指令滤波器, 0 = 不开启, 1 = 开启 默认值: 0, 即 2 阶低通滤波
速度环 Kvb 阈值	60F907	60F90720	0x8607	Integer32 RW	速度环退积分电流阈值
速度环积分限制	60F908	60F90820	0x8608	Integer32 RW	速度环积分限制
速度环 Kvb	60F909	60F90910	0x8609	Unsigned16 RW	速度环退积分增益
速度到窗口	60F90A	60F90A20	0x860A	Integer32 RW	速度误差窗口
失速判断时间	60F91C	60F91C10	0x861C	Unsigned16 RW ms	失速判断时间
陷波滤波器 - 输入	60F914	60F91410	0x8614	Integer16 R	陷波滤波器的输入

9.1.12 位置环参数

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
位置环比例增益 0	60FB01	60FB0110	0x8801	Integer16 RW	位置环 kp
位置	60FB02	60FB0210	0x8802	Integer16	位置环前馈

环速度前馈				RW	
位置环加速度前馈	60FB03	60FB0310	0x8803	Integer16 RW	位置环的加速度前馈
平滑滤波	60FB05	60FB0508	0x8805	Unsigned8 RW	平滑滤波器参数调整
跟随误差窗口	606500	60650020	0x7650	Unsigned32 RW	跟随误差大于 6065h, 且时间超过 6066h, 报跟随误差错误
位置到窗口	606700	60670020	0x7670	Unsigned32 RW	位置指令与实际位置的差值小于位置到窗口, 且时间达到位置到窗口时间, 认为位置到达
位置跟随误差	60F400	60F40020	0x8100	Integer32 R	位置跟随误差

9.1.13 电流环参数

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
目标电流限制	607300	60730010	0x7730	Unsigned16 RW Ap	电流指令最大值
电流环比例增益	60F601	60F60110	0x8301	Unsigned16 RW	电流环 kp
电流环积分增益	60F602	60F60210	0x8302	Unsigned16 RW	电流环 ki
电流补偿因数	60F603	60F60310	0x8303	Unsigned16 RW	电流补偿因数
电压反馈因数	60F604	60F60410	0x8304	Integer16 RW	电压反馈因数
内部最大力矩限制	60F60C	60F60C10	0x830C	Integer16R	驱动器内部有效最大力矩限制
设备实际 Iit	60F60D	60F60D10	0x830D	Unsigned16R %	驱动器过温保护的 实际数据
设备最大 Iit	60F60E	60F60E10	0x830E	Unsigned16R Ap	驱动器过温保护的 最大值
电机实际 Iit	60F60F	60F60F10	0x830F	Unsigned16R %	电机过温保护的 实际数据

电机最大 Iit	60F610	60F61010	0x8310	Unsigned16 R Ap	电机过温保护的 最大值
-------------	--------	----------	--------	-----------------------	----------------

9.1.14 原点模式

参数名称	UART 通讯 地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
原点 模式	609800	60980008	0x7C00	Integer8 RW	寻找原点的模式，详情请看第六章
原 点 转 折 信 号 速 度	609901	60990120	0x7D01	Unsigned32 RW rpm	寻找转折信号时的速度
原 点 信 号 速 度	609902	60990220	0x7D02	Unsigned32 RW rpm	寻找原点时的速度
上 电 找 原 点	609903	60990308	0x7D03	Unsigned8 RW	第一次上电使能的时候自动开始找原点
寻 找 原 点 最 大 电 流	609904	60990410	0x7D04	Integer16 RW Ap	寻找原点时的最大电流设定
原 点 偏 移 模 式	609905	60990508	0x7D05	Unsigned8 RW	原点偏移模式控制 0: 运行到原点偏移 1: 运行到事件触发点，结束后实际位置将变为-原点偏移
原 点 索 引 信 号 盲 区	609906	60990608	0x7D06	Unsigned8 RW	当原点模式同时使用限位/原点开关与索引信号时，找原点过程中遇到开关信号后的盲区内忽略所检测到的索引信号。 0 : 0rev; 1:0.25rev; 2: 0.5rev; 默认值为 0; 当该值设为 1 时找原点过程中将根据索引信号与开关信号的位置偏移将该值设为 0 或者 2. 机械固定后需将该参数保存, 机械更改后重设为 1 即可。
原 点 加 速 度	609A00	609A0020	0x7E00	Unsigned32 RW rps/s	寻找原点时的加速度

9.1.15 错误字

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
错误字	260100	26010010	0x5A00	Unsigned16 R	错误状态 bit0: 请参考 26020010 bit1: 编码器 ABZ 连接报警/通讯式编码器断线错误 bit2: 编码器 UVW 连接报警/通讯式编码器多圈错误 bit3: 编码器计数错误/通讯式编码器 CRC 错误 bit4: 驱动器温度过高 bit5: 总线电压过高 bit6: 总线电压过低 bit7: 驱动器输出短路 bit8: 制动电阻异常 bit9: 跟随误差过大 bit10: 逻辑低压报警 bit11: 电机或驱动器 iit 报警 bit12: 速度超差 bit13: 电机温度过高 bit14: 电机励磁错误/通讯式编码器其他错误(请参考 30030208) bit15: Eeprom 报警
错误状态字 2	260200	26020010	0x5B00	Unsigned16 R	错误状态 bit0: 电流传感器 bit1: 驱动器输出短路 bit2: 找原点错误 bit3: 电机缺相 bit4: 电机数据出错 bit5: 用户 iit bit6: 保留 bit7: 保留 bit8: 预使能错误 bit9: 正限位错误 bit10: 负限位错误 bit11: 脉冲频率过高 bit12: 总线掉线错误 bit13: 全闭环下编码器计数方向错误 bit14: 主编码器连接错误

					bit15: 主编码器计数错误
--	--	--	--	--	-----------------

9.1.16 停止模式

参数名称	UART 通讯地址	CANopen 通讯地址	Modbus 通讯地址	对象属性	参数详情
快速停止模式	605A00	605A0010	0x7510	Integer16 RW	快速停止模式 0: 不受控停止 1: (0x6084) 斜坡减速停止 2: (0x6085) 快速停止减速度停止 5: (0x6084) 斜坡减速停止, 最后停在快速停止状态 6: (0x6085) 快速停止减速度停止, 最后停在快速停止状态
关机停止模式	605B00	605B0010	0x7520	Integer16 RW	关机停止模式 (控制字由 0x0F→0x06) 0: 不受控停止 1: (0x6084) 斜坡减速停止 2: (0x6085) 快速停止减速度停止
禁止停止模式	605C00	605C0010	0x7530	Integer16 RW	禁止停止模式 (控制字由 0x0F→0x07) 0: 不受控停止 1: (0x6084) 斜坡减速停止 2: (0x6085) 快速停止减速度停止
暂定模式	605D00	605D0010	0x7540	Integer16 RW	暂停模式 1: (0x6084) 斜坡减速停止 2: (0x6085) 快速停止减速度停止
报错停止模式	605E00	605E0010	0x7550	Integer16 RW	错误急停模式 0: 立即停止 1: (0x6084) 斜坡减速停止 2: 使用 (0x6085) 急停减速度停止

9.2 单位转换

注意:

- 单位转换公式里面的 N 表示的是转换前的值
- 在写入参数时, 需要先将参数值转换内部单位 (DEC), 再转为十六进制后进行写入
- 位置单位无需转换

例:

假设目标速度为 500rpm, 反馈精度为 131072, 那么 500rpm 转为 DEC 有

$\text{rpm} \rightarrow \text{DEC} = N / 1875 * 512 * \text{反馈精度}$

$\text{DEC} = 500 / 1875 * 512 * 131072 * 1000 / 1000 = 17895697$

转为十六进制即 0111 1111, 再将十六进制数写进目标速度的对应地址就可以了

9.2.1 电流转换公式

SVD4812RC-AA 驱动器最大电流 (DEC) = 450

SVD4822RC-AA 驱动器最大电流 (DEC) = 800

$\text{DEC} \rightarrow \text{Ap} = N / 2048 * [\text{最大电流 (DEC)}] / 10$

$\text{Ap} \rightarrow \text{DEC} = N * 2048 / [\text{最大电流 (DEC)}] * 10$

$\text{DEC} \rightarrow \text{Arms} = N / 2048 * [\text{最大电流 (DEC)}] / 10 / 1.414$

$\text{Arms} \rightarrow \text{DEC} = N * 2048 / [\text{最大电流 (DEC)}] * 10 * 1.414$

9.2.2 速度转换公式

$\text{DEC} \rightarrow \text{rpm} = N * 1875 / 512 / \text{反馈精度}$

$\text{rpm} \rightarrow \text{DEC} = N / 1875 * 512 * \text{反馈精度}$

9.2.3 加速度 (减速度) 转换公式

$\text{DEC} \rightarrow \text{rps/s} = N * 1000 * 4000 / 65536 / \text{反馈精度}$

$\text{rps/s} \rightarrow \text{DEC} = N / 1000 / 4000 * 65536 * \text{反馈精度}$

9.2.4 力矩转换公式

$\text{DEC} \rightarrow \% = N / 10$

$\% \rightarrow \text{DEC} = N * 10$

第十章 UART 通讯

本型号驱动器仅支持一对一模式,通过 CN2 口进行 UART 通讯,详细请看第四章系统接口及配线。

10.1 UART 通讯格式

默认波特率为 115200bps, 可以通过 2F8003 (索引号) 进行修改。

本机 UART 通讯格式为: 115200, 8, N, 1

也就是波特率固定为 115200bps, 数据位数为 8 位, 无奇偶校验, 1 个停止位。

UART 通讯协议

本机 UART 通讯遵循严格的主从站协议。上位机/主控器发一帧或者连续多帧数据给本机, 本机接收到正确数据将回应一帧或者连续多帧相应的数据。

UART 通讯协议采用固定的十个字节格式:

设备地址	命令字	对象地址			数据区				校验
Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10
设备 ID	命令字	索引		子索引	低位——>高位 表示 0x01 则为 01 00 00 00				Check

设备 ID: 由拨码开关进行设置, 详情见第四章 (万能站号为 127)

命令字: 根据用户的目的取不同的值, 可参考下面的列表。

对象地址: 通过查询对象列表得知, 其中索引 (Byte3-Byte4) 也从低位——>高位

数据区: 用于存贮发送的数据或者接受到的数据

若数据区的四个字节为 0F 65 77 83, 则正确的 HEX 格式为 83 77 65 0F

Check: 校验位, 取值为前 9 个字节的和取反后的后面 1 个字节

命令字具体定义如下:

(1) 读命令, 主机读取本机相关对象地址的数据

主机发送命令字	本机回复命令字	对应意义
0x40	0x4f	对象有效数据是 Byte6
0x40	0x4b	对象有效数据是 Byte6-Byte7
0x40	0x43	对象有效数据是 Byte6-Byte9
0x40	0x80	校验出错, 错误字包含在 Byte6-Byte9

(2) 写命令, 主机往本机相关对象地址写数据

主机发送命令字	本机回复命令字	对应意义
0x23	0x60	对象有效数据是 Byte6-Byte9
	0x80	错误, Byte6-Byte9 中包含错误代码
0x2b	0x60	对象有效数据是 Byte6-Byte7
	0x80	错误, Byte6-Byte9 中包含错误代码
0x2f	0x60	对象有效数据是 Byte6
	0x80	错误, Byte6-Byte9 中包含错误代码

10.2 UART 通讯实例

以下实例默认设备 ID 为 01，波特率为 115200

10.2.1 速度模式应用

速度转换公式

速度填入数的值= $N/1875*512*$ 电机编码器分辨率（反馈精度），其中 N 为转速（rpm）

这里以 17bit 电机为例，则电机编码器分辨率=131072

目标速度 10rpm 转换十进制为 357913，十六进制为 57619

目标速度-10rpm 转换十进制为 -357913，十六进制为 FFFA 89E7

轮廓加速度转换公式

加速度填入数的值= $N/1000/4000*65536*$ 电机编码器分辨率（反馈精度），其中 N 为转速（rps/s）
电机编码器分辨率参数同上。

轮廓加速度 50rps/s 转换十进制为 107374，十六进制为 1 A36E

索引号	变量名称	数据类型	设置值	报文	备注
606000	工作模式	Integer8	3	01 2F 60 60 00 03 00 00 00 0D	工作模式 设置为 3
60FF00	目标速度	Integer32	10rpm	01 23 FF 60 00 19 76 05 00 E9	目标速度 设置为 10 rpm
			-10rpm	01 23 FF 60 00 E7 89 FA FF 14	目标速度 设置 为 -10 rpm
608300	轮廓 加速度	Unsigned32	50rps/s	01 23 83 60 00 6E A3 01 00 E7	轮廓加速 度设置为 50rps/s
608400	轮廓 减速度	Unsigned32	50rps/s	01 23 84 60 00 6E A3 01 00 E6	轮廓减速 度设置为 50rps/s
604000	控制字	Unsigned16	F	012 B40 60 00 0F 00 00 00 25	将控制字 设为 F, 对 电机锁轴

按照上述操作执行，电机将以轮廓加速度为 50rps/s 加速至 10rpm 或-10rpm，若再对目标速度写 0，电机将以轮廓减速度为 50rps/s 减至 0

10.2.2 绝对位置模式应用

轮廓运行速度与上述速度模式转换方式一致

通过转换得 50rpm 可转换为十进制 1789569，十六进制为 1B 4E81

目标位置 10000 inc 转换为十六进制为 2710

若为 -10000 inc 转换为十六进制为 FFFF D8F0

索引号	变量名称	数据类型	设置值	报文	备注
606000	工作模式	Integer8	1	01 2F 60 60 00 01 00 00 00 0F	工作模式 设置为 1
608100	轮廓运行速度	Unsigned32	50rpm	01 23 81 60 00 81 4E 1B 00 11	将轮廓运行速度设置为 50 rpm
607A00	目标位置	Integer32	10000 inc	01 23 7A 60 00 10 27 00 00 CB	将目标设为 10000 inc
			-10000 inc	01 23 7A 60 00 F0 D8 FF FF 3C	将目标设为 -10000 inc
604000	控制字	Unsigned16	0x2F-> 0x3F	01 2B 40 60 00 2F 00 00 00 05 01 2B 40 60 00 3F 00 00 00 F5	控制字从 2F 设为 3F

按照上述操作执行，电机将以 50rpm 的速度运行到 10000 inc 或 -10000 inc 位置的地方。

10.2.3 相对位置模式应用

轮廓运行速度与上述速度模式转换方式一致

通过转换得 50rpm 可转换为十进制 1789569，十六进制为 1B 4E81

目标位置 10000 inc 转换为十六进制为 2710

若为 -10000 inc 转换为十六进制为 FFFF D8F0

索引号	变量名称	数据类型	设置值	报文	备注
606000	工作模式	Integer8	1	01 2F 60 60 00 01 00 00 00 0F	工作模式 设置为 1
608100	轮廓运行速度	Unsigned32	50rpm	01 23 81 60 00 81 4E 1B 00 11	将轮廓运行速度设置为 50 rpm
607A00	目标位置	Integer32	10000 inc	01 23 7A 60 00 10 27 00 00 CB	将目标设为 10000 inc
			-10000 inc	01 23 7A 60 00 F0 D8 FF FF 3C	将目标设为 -10000 inc
604000	控制字	Unsigned16	0x4F->	01 2B 40 60 00 4F	控制字从

			0x5F	00 00 00 E5 01 2B 40 60 00 5F 00 00 00 D5	4F 设为 5F
--	--	--	------	---	----------

按照上述操作执行，电机在原有的位置上，再以 50rpm 的速度正方向运行或反方向 10000inc 的距离。

相对位置与绝对位置操作方法差不多，相对位置是在原有的位置上移动 10000inc，绝对位置是将电机移动到 10000inc 位置上。

10.2.4 力矩模式的应用

扭矩转换公式

填入数的值（十进制）=N*10

若设 10%，经过转换则为 100，十六进制数为 64

最大限制速度与上述速度模式转换方式一致

通过转换得 5000rpm 可转换为十进制 178 956 970，十六进制为 AAA AAAA

索引号	变量名称	数据类型	设置值	报文	备注
606000	工作模式	Integer8	4	01 2F 60 60 00 04 00 00 00 0C	工作模式 设置为 4
607100	目标 扭矩%	Unsigned32	10	01 2B 71 60 00 64 00 00 00 9F	目标扭矩设 为 10%（占额 定扭矩的 10%）
607F00	最大限制 速度	Unsigned32	5000 rpm	01 23 7F 60 00 AA AA AA 0A 58	将最大限制 速度设为 5000 rpm
604000	控制字	Unsigned16	F	01 2B 40 60 00 0F 00 00 00 25	将控制字设 为 F

按照上述操作，电机将以额定扭矩的 10%进行运转。

10.2.5 原点模式的应用

原点转折信号速度、原点信号速度与上述速度模式转换方式一致

通过转换得 50rpm 可转换为十进制 1789569，十六进制为 1B 4E81

原点模式 17 转换十六进制为 11

索引号	变量名称	数据类型	设置值	报文	备注
606000	工作模式	Integer8	6	01 2F 60 60 00 06 00 00 00 0A	工作模式 设置为 6
609901	原点转折	Unsigned32	50rpm	01 23 99 60 01 81	原点转折信

	信号速度			4E 1B 00 F8	号速度设为 50 rpm
609902	原点信号 速度	Unsigned32	50rpm	01 23 99 60 02 81 4E 1B 00 F7	原点信号速 度设为 50rpm
609800	原点模式	Integer8	17	01 2F 98 60 00 11 00 00 00 C7	原点模式设 为 17 模式
604000	控制字	Unsigned16	F→ 1F	01 2B 40 60 00 0F 00 00 00 25 01 2B 40 60 00 1F 00 00 00 15	控制字从F设 为 1F

按照上述操作执行，电机将按照原点模式 17，以原点转折信号速度 50rpm、原点信号速度 50rpm 的速度寻找原点。

注意：

应当注意数值单位换算，详细可以见第六章 常用对象列表 后面的单位换算。

通讯错误代码表

序号	错误代码	解释
1	0x05040001	无效命令
2	0x06010001	只写参数
3	0x06010002	只读参数
4	0x06020000	无效索引
5	0x06040041	无法映射
6	0x06060000	设备硬件故障
7	0x06070010	数据长度错误
8	0x06070013	对象数据过长
9	0x06070013	对象数据过短
10	0x06090011	无效子索引
11	0x06090030	无效值
12	0x06090031	Value 太高
13	0x06090032	Value 太低
14	0x08000000	一般错误
15	0x08000021	错误的控制字
16	0x08000022	错误的状态字
17	0x08000023	无对象字典

第十一章 RS485 通讯

11.1 RS485 通讯硬件介绍

本型号驱动器通过 CNA1 (IN) 口和 CNB1 (OUT) 进行 RS485 进行通讯。具体详情请看第四章 系统接口与配线

11.2 RS485 通讯格式

本机 RS485 波特率可以通过上位机进行设置，或者通过命令进行修改。

RS485				搜索
索引	子索引	名称	数据类型	属性
2F80	04	RS485波特率	Unsigned16	RWSLB

用于设置RS485的波特率

48: 4800

96: 9600

192: 19200

384: 38400

576: 57600

1152: 115200

需要重新启动

图 11-1 RS485 波特率相关设置

11.2.1 通讯协议

本型号支持 Modbus RTU 通讯协议，其内部对象为不连续的 16 位数据寄存器。报文格式如下：

站号	功能码	数据区	校验位	
Byte1	Byte2	ByteN	Byte (N+1)	Byte (N+2)

站号：本机 RS485 地址, 由驱动器拨码开关 S1 的 SW1-SW4 决定, 详情见第四章(万能站号为 127)

功能码：根据用户的目的取不同的值，可参考下面的列表。

数据区：存储了索引号以及操作相关的数据

校验位：用于通讯的时候进行校验，本型号驱动器默认使用 ModbusCRC16 校验

11.2.2 常用功能介绍

读数据寄存器 (0x03)

发送报文:

设备 ID	功能码	对象 Modbus 地址		读取寄存器数		CRC
		高字节	低字节	高字节	低字节	
1 字节	03	1 字节	1 字节	1 字节	1 字节	2 字节

回复报文:

设备 ID	命令字	返回字节数 高字节	寄存器数据		CRC
			高字节	低字节	
1 字节	03	1 字节	1 字节	1 字节	2 字节

注意:

若出现地址不存在等响应错误, 则返回的功能码为 0x81

写单数据寄存器 (0x06)

设备 ID	功能码	Modbus 地址		修改内容		CRC
		高字节	低字节	高字节	低字节	
1 字节	06	1 字节	1 字节	1 字节	1 字节	2 字节

注意:

若对象写入成功, 则返回原文

写多保持寄存器 (0x10)

设备 ID	功能码	Modbus 地址	数据长度 (word)		写入数据 字节数	低位数据		高位数据		CRC
			高字节	低字节		高字节	低字节	高字节	低字节	
1 字节	10	2 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	2 字节

回复报文:

设备 ID	功能码	Modbus 地址	数据长度 (word)		CRC
			高字节	低字节	
1 字节	10	2 字节	1 字节	1 字节	2 字节

注意:

若写出现地址不存在、对只读数据进行写入等非法操作, 则返回的功能码位 0x90。

11.3 RS485 通讯实例

下面以 1 站号为例

关于目标速度等参数等转换详情, 可以参考 UART 通讯实例, 也可以参考第六章对象列表相关参数的转换公式, 这里不做过多解释。

11.3.1 读实际位置及回应报文

实际位置 Modbus 地址为 0x7630

则读实际位置发送报文:

01 03 76 30 00 02 DE 4C

回应报文:

01 03 04 2B 60 71 26 56 43

01 03 76 30 00 02 DE 4C

01: 站号

03: 功能码

76 30: 实际位置 Modbus 地址

00 02: 读取寄存器数, 2word=32bit

DE 4C: Modbus CRC16 校验位

01 03 04 2B 60 71 26 56 43

01: 站号

03: 功能码

04: 返回字节数

2B 60 71 26: 返回的数据, 转换为十六进制数为 7126 2B60, 转为十进制数字位

1 898 326 880, 即实际位置为 1 898 326 880 inc

56 43: Modbus CRC16 校验位

11.3.2 速度模式

速度填入数的值(十进制数) $=N/1875*512*$ 电机编码器分辨率(反馈精度), 其中N为转速(rpm)

这里举例的电机编码分辨率为 131072

目标速度为 100rpm, 进行转换则十进制为 3 579 139, 十六进制为 36 9D03

Modbus 地址	变量名称	数据类型	设置值	报文	备注
0x7600	工作模式	Integer8	3	01 06 76 00 00 03 D3 83	工作模式 设置为 3
0x8C00	目标速度	Integer32	100rpm	01 10 8C 00 00 02 04 9D 03 00 36 98 D3	目标速度 设置为 100 rpm
0x7930	轮廓 加速度	Unsigned32	50rps/s	01 10 79 30 00 02 04 A3 6E 00 01 7D 70	轮廓加速 度设置 为 50rps/s
0x7940	轮廓 减速度	Unsigned32	50rps/s	01 10 79 40 00 02 04 A3 6E 00 01 7A 54	轮廓减 速度 设置 为 50rps/s
0x7400	控制字	Unsigned16	F	01 06 74 00 00 0F D2 3E	将控制字 设为 F, 对 电机锁轴

01 06 76 00 00 03 D3 83

01: 站号

06: 写单数据寄存器
 76 00: 工作模式对应的 RS485 地址
 00 03: 将工作模式设置为 3
 D3 83: Modbus CRC16 校验位

01 10 8C 00 00 02 04 9D 03 00 36 98 D3
 01: 站号
 10: 写多保持寄存器
 8C 00: 目标速度对应的 RS485 地址
 02: 写入的数据为 2 个 WORD(4 个 Byte)
 04: 写入的数据为 4 个 Byte
 9D 03 00 36: 对应的十六进制数字为 369D03, 转换为十进制则为 3579139, 对应 100rpm 的速度, 具体转换公式可以参考 UART 通讯实例或者对象列表。

轮廓加速度、轮廓减速度和控制字与上述例子类似, 这里不做过多解释。

11.3.3 绝对位置模式

50rpm 转换为十进制则为 1789569, 转换为十六进制则为 1B 4FAD

Modbus 地址	变量名称	数据类型	设置值	报文	备注
0x7600	工作模式	Integer8	1	01 06 76 00 00 01 52 42	工作模式设置为 1
0x7910	轮廓运行速度	Unsigned32	50rpm	01 10 79 10 00 02 04 4F AD 00 1B 3A 0F	将轮廓运行速度设置为 50 rpm
0x77A0	目标位置	Integer32	10000 inc	01 10 77 A0 00 02 04 27 10 00 00 B1 54	将目标位置设置为 10000 inc
0x7400	控制字	Unsigned16	0x2F→ 0x3F	01 06 74 00 00 2F D3 E6 01 06 74 00 00 3F D2 2A	控制字从 2F 设为 3F

绝对位置模式与相对为位置模式操作方法类似, 这里不做介绍
 轮廓运行速度与上述速度模式计算一样, 这里我们介绍一下目标位置的计算:
 若目标位置为 10000inc, 则可直接换算成十六进制即 2710, 无需其他换算。

11.3.4 力矩模式

扭矩转换公式
 填入数的值 (十进制) = N*10
 若设 10%, 经过转换则为 100, 十六进制数为 64

最大限制速度与上述速度模式转换方式一致

通过转换得 5000rpm 可转换为十进制 178 956 970，十六进制为 AAA AAAA

Modbus 地址	变量名称	数据类型	设置值	报文	备注
0x7600	工作模式	Integer8	4	01 06 76 00 00 04 92 41	工作模式设置为 4
0x7710	目标扭矩%	Unsigned32	10	01 06 77 10 00 64 92 50	目标扭矩设为 10% (占额定扭矩的 10%)
0x77F0	最大限制速度	Unsigned32	5000 rpm	01 10 77 F0 00 02 04 AA AA 0A AA 38 3E	将最大限制速度设为 5000 rpm
0x7400	控制字	Unsigned16	F	01 06 74 00 00 0F D2 3E	将控制字设为 F

按照上述操作，电机将以额定扭矩的 10%进行运转。

11.3.5 原点模式

50rpm 转换为十进制则为 1789569，转换为十六进制则为 1B 4FAD

Modbus 地址	变量名称	数据类型	设置值	报文	备注
0x7600	工作模式	Integer8	6	01 06 76 00 00 06 13 80	工作模式设置为 6
0x7D01	原点转折信号速度	Unsigned32	50rpm	01 10 7D 01 00 02 04 4F AD 00 1B C8 0F	原点转折信号速度设为 50 rpm
0x7D02	原点信号速度	Unsigned32	50rpm	01 10 7D 02 00 02 04 4F AD 00 1B 88 1A	原点信号速度设为 50rpm
0x7C00	原点模式	Integer8	17	01 06 7C 00 00 11 50 56	原点模式设为 17 模式
0x7400	控制字	Unsigned16	F->1F	01 06 74 00 00 1F D3 F2	控制字从 F 设为 1F

按照上述操作执行，电机将按照原点模式 17，以原点转折信号速度 50rpm、原点信号速度 50rpm 的速度寻找原点。

11.3.6 RS485-PDO 模式（读取多个参数）

该模式适用与本型号，其余型号见参考说明书详情，使用该模式具体步骤如下

第一步：将 TPD01 站号设定为一个指定地址（不能与已使用的 Modbus 地址重复），同时将 TPD01

映射设为所需要返回的索引号(跟 CAN 通讯相同的地址)。下面例图映射的是实际位置(60630020)和实际速度(606C0020)。



N	索引	数据类型	名称	设定值	当前值	单位
1	180001	Unsigned32	TPDO1站号	9020	181	HEX
2	180002	Unsigned8	TPDO1传输类型		254	DEC
3	180003	Unsigned16	TPDO1禁止时间		10	DEC
4	180005	Unsigned16	TPDO1事件时间		0	DEC
5	1A0000	Unsigned8	TPDO1有效映射对象个数	0	0	DEC
6	1A0001	Unsigned32	TPDO1映射1	60630020	60630020	HEX
7	1A0002	Unsigned32	TPDO1映射2	606C0020	606c0020	HEX
8	1A0003	Unsigned32	TPDO1映射3		0	HEX
9	1A0004	Unsigned32	TPDO1映射4		0	HEX

图 11-2 读多个参数

第二步：发送报文，格式如下

设备 ID	命令字	对象 Modbus 地址		读取字节数		CRC
		高字节	低字节	高字节	低字节	
1 字节	03	1 字节	1 字节	1 字节	1 字节	2 字节

这里我们发送报文：01 03 90 20 00 04 68 C3

回应报文：01 03 08 29 DC 04 12 78 80 00 1D 2A AE

01 03 90 20 00 04 68 C3

01：站号

03：读功能码

90 20：RS485-PDO 模式下的地址

00 04：读取总寄存器数

68 C3：校验位

01 03 08 29 DC 04 12 78 80 00 1D 2A AE

01：站号

03：读功能码

08：读取的数据总字节数

29 DC 04 12：表示的是实际位置 (TPDO 映射 1)，十六进制数为 412 29DC 十进制数为 68 299 228，即当前位置为 68 299 228

78 80 00 1D：表示的是实际速度 (TPDO 映射 2)，十六进制数为 1D 7880 十进制数为 1 931 392

11.3.7 RS485-PDO 模式 (写多个参数)

具体操作步骤同上，我们这里对驱动器进行写速度模式为例。

如下图，我们映射了目标速度 (60FF0020)，工作模式 (60600008)，控制字 (60400010)

N	索引	数据类型	名称	设定值	当前值	单位
1	140001	Unsigned32	RPDO1站号	9020	9020	HEX
2	140002	Unsigned8	RPDO1传输类型		254	DEC
3	140003	Unsigned16	RPDO1禁止时间		10	DEC
4	160000	Unsigned8	RPDO1有效映射对象个数		0	DEC
5	160001	Unsigned32	RPDO1映射1	60FF0020	60ff0020	HEX
6	160002	Unsigned32	RPDO1映射2	60600008	60600008	HEX
7	160003	Unsigned32	RPDO1映射3	60400010	60400010	HEX
8	160004	Unsigned32	RPDO1映射4		0	HEX

图 11-3 写多个参数

设置完成后，

发送报文：01 10 90 20 00 04 08 9D 03 00 36 00 03 00 0F 75 61

01：站号

10：写多保持寄存器

90 20：RS485-PDO 模式下的地址

00 04：修改内容为 4 个 WORD 大小，目标速度四个字节，工作模式两个字节（原本只有一个字节，这里我们对高位进行补零），控制字两个字节

08：修改内容为 8 个 Byte 大小（为什么不是 7 个 Byte 的原因上面有描述）

9D 03 00 36：目标速度修改内容，十六进制为 0036 9D03，经过换算为 100rpm。

00 03：工作模式修改内容，将工作模式设置为 3

00 0F：控制字修改内容，对控制字写 F

75 61：校验位

注意：

若想读取多个参数，应切换到 TPDO (Transmit Process Data Object)；

若想写多个参数，应切换到 RPDO (Receive Process Data Object)

第十二章 CANopen 通讯

CANopen (Controller Area Network Open) 是一种基于 CAN 总线的通信协议，用于实现工业自动化和控制系统中不同设备之间的数据交换和通信。它提供了一套标准化的通信和设备描述方式，使得各种设备能够在同一个网络上协同工作。

以下是 CANopen 通讯的简单介绍：

通信介质： CANopen 使用 CAN 总线作为通信介质。CAN 总线是一种高度可靠的串行通信协议，常用于工业环境中，具有抗干扰能力强、实时性好的特点。

设备类型： CANopen 通信可以用于连接各种类型的设备，如电机驱动器、传感器、控制器、HMI（人机界面）等。这些设备可以是不同厂家生产的，只要它们符合 CANopen 通信协议规定的通信方式和数据结构。

通信协议： CANopen 定义了一套通信协议，包括数据帧格式、通信对象、对象字典、PDO (Process Data Object)、SDO (Service Data Object) 等。这些协议规定了设备之间的数据交换方式和通信流程。

对象字典： CANopen 使用对象字典来描述设备的参数、状态和功能。对象字典中包含了一系列索引和子索引，每个索引对应一个特定的参数或功能。设备可以通过读写对象字典来进行数据交换和配置。

PDO 和 SDO： PDO 是用于实时数据传输的对象，SDO 用于非实时数据传输和配置。PDO 可以在设备之间高效地传输实时数据，而 SDO 用于配置和管理设备的参数。

节点： CANopen 网络中的每个设备被称为一个节点。每个节点都有一个唯一的节点 ID，用于在网络中识别和寻址。

下表是 CANopen 对象字典的列举与说明

参数名称	CANopen 地址			属性含义
	索引	子索引	数据长度	
工作模式	6060	00	08	RW
控制字	6040	00	10	RW
目标速度	60FF	00	20	RW

这里对仅数据长度以及属性含义进行解释

数据长度： 08——数据长度为 1Byte

10——数据长度为 2Byte

20——数据长度为 4Byte

属性含义： R——可读；

W——可写；

M——映射

S——可存贮（掉电不丢失）

12.1 硬件说明

本型号驱动器支持一对多模式，若使用 CAN 通讯方式进行通讯，则默认 CAN1A(IN) 进行输入，CAN1B(OUT) 进行输出，详细请看第四章接口及配线。

12.2 CAN 通讯

12.2.1 EDS

EDS 是“Electronic Data Sheet”的缩写，它是用于描述 CANopen 设备的通信参数、对象字典、映射信息和功能等详细信息的电子数据表格。EDS 文件通常以 XML 格式编写，用于配置和识别 CANopen 设备，并确保设备之间的正确通信。

在 CANopen 网络中，每个设备都有一个对象字典，用于存储设备的参数、状态和功能信息。EDS 文件描述了这个对象字典的结构和内容，以及设备的通信参数、PDO（Process Data Object）映射等信息。通过读取设备的 EDS 文件，用户可以了解设备的功能和通信参数，从而能够正确地配置和集成设备到 CANopen 网络中。

EDS 文件的内容通常包括：

- 设备描述信息，如设备名称、制造商信息等。
- 通信参数，如节点 ID、波特率等。
- 对象字典的结构和内容，包括对象索引、子索引、数据类型和访问权限等。
- PDO 映射信息，描述 PDO 通信中数据的映射和传输方式。

通过使用 EDS 文件，用户可以方便地配置和集成各种不同的 CANopen 设备，确保设备能够正确地进行通信和数据交换。

12.2.2 SDO

(1) SDO 简介

在 CANopen 通信协议中，SDO 代表“Service Data Object”，它是一种用于在 CANopen 网络中进行参数配置、状态查询和数据交换的通信机制。SDO 机制允许主站 (Master) 设备与从站 (Slave) 设备之间进行点对点的通信，以读取或写入从站设备的对象字典中的数据。对象字典是 CANopen 协议中的一个重要概念，它是一种数据结构，用于存储设备的参数、状态和功能信息。通过 SDO 通信，主站可以向从站发送请求，读取或写入特定对象字典中的数据。

SDO 通信的基本流程如下：

- 主站发送一个 SDO 请求帧，其中包含了要访问的从站设备的节点 ID、要读取或写入的对象字典索引和子索引等信息。
- 从站收到 SDO 请求后，执行相应的操作，比如读取或写入对象字典中的数据。
- 从站将操作结果封装在 SDO 响应帧中，发送回主站。
- 主站收到 SDO 响应后，解析其中的数据，完成读取或写入操作。

本机 CAN 接口支持 Canopen SDO 数据传输协议，SDO 主要用来在设备之间传输低优先级的对象，这种数据传输跟 MODBUS 的方式类似，即主站发出后，需要从站返回数据响应：

Client→Server/Server→Client. SDO 的基本结构如下表所示：

SDO specifier	Command	对象索引	对象子索引	最大 4 字节数据
Byte0		Byte1-Byte2	Byte3	Byte4-Byte7

对象索引的 2 个字节和数据对象的 4 个字节都是低位在前，高位在后排列的，比如说对象索引为 0x606C，那么 Byte1=6C，Byte2=60。

(2) SDO 读取参数

读取参数时, 发送 SDO 报文格式如下表

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	发送命令字	对象索引		对象子索引		00		

读取参数时, 接收 SDO 报文格式如下表

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	接收命令字	对象索引		对象子索引		最大 4 字节数据		

下面是读取参数时相关的发送命令字以及接受命令字

发送命令字	接受命令字	说明
0x40	0x43	接收数据为 4 个字节
	0x4B	接收数据为 2 个字节
	0x4F	接收数据为 1 个字节
	0x80	接收数据存在错误

(3) SDO 修改参数

修改参数时, 发送 SDO 报文格式如下表

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	发送命令字	对象索引		对象子索引		最大 4 字节数据		

修改参数时, 接收 SDO 报文格式如下表

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	接收命令字	对象索引		对象子索引		最大 4 字节数据		

下面是修改参数时相关的发送命令字以及接受命令字

发送命令字	接受命令字	说明
0x23	0x60	成功发送数据为 4 个字节的报文
0x23	0x80	SDO 报文发送失败
0x2B	0x60	成功发送数据为 2 个字节的报文
0x2B	0x80	SDO 报文发送失败
0x2F	0x60	成功发送数据为 1 个字节的报文
0x2F	0x80	SDO 报文发送失败

(4) SDO 通讯错误代码表

序号	错误代码	解释
1	0x05040001	无效命令
2	0x06010001	只写参数
3	0x06010002	只读参数
4	0x06020000	无效索引
5	0x06040041	无法映射
6	0x06060000	设备硬件故障
7	0x06070010	数据长度错误
8	0x06070013	对象数据过长
9	0x06070013	对象数据过短
10	0x06090011	无效子索引
11	0x06090030	无效值
12	0x06090031	Value 太高
13	0x06090032	Value 太低
14	0x08000000	一般错误
15	0x08000021	错误的控制字
16	0x08000022	错误的状态字
17	0x08000023	无对象字典

12.2.3 PDO

(1) PDO 简介

在 CANopen 通信协议中，PDO 代表“Process Data Object”，它是一种用于实时数据交换的通信机制，用于在 CANopen 网络中快速传输设备之间的实时数据。PDO 机制允许 CANopen 网络中的设备之间周期性地交换数据，通常用于传递实时控制、监测和反馈数据。与 SDO (Service Data Object) 不同，PDO 通信是点对多点的，一个主站可以将数据广播给多个从站。

PDO 通信的基本流程如下：

- 主站配置 PDO 通信参数，包括周期、映射对象字典等。
- 从站按照主站配置的参数，周期性地向网络中发送 PDO 数据帧，其中包含实时数据。
- 其他设备收到 PDO 数据帧后，可以解析其中的数据并做出相应的响应。

PDO 通信的特点包括：

实时性： PDO 通信是基于 CAN 总线的实时通信机制，适用于需要快速传输实时数据的应用场景。

高效性： 由于 PDO 通信是周期性的广播方式，可以在网络中快速传递数据，适用于实时控制和监测。

预定义映射： PDO 通信需要预先配置映射对象字典，从而明确了数据的传输格式和周期。

(2) PDO COB-ID

在 CANopen 通信协议中，PDO COB-ID 代表“Process Data Object Communication Object Identifier”，它是用于标识和区分不同 PDO 通信对象的唯一标识符。PDO 通信是用于实时数据

交换的通信机制，允许 CANopen 设备之间周期性地传输数据。每个 PDO 通信对象都有一个独特的 PDO COB-ID，用于在 CAN 总线上标识该对象的数据帧。

PDO COB-ID 的组成包括以下部分：

11 位或 29 位标识符： 标识 CAN 总线上的通信对象，11 位标识符适用于标准 CANopen 网络，而 29 位标识符适用于扩展 CANopen 网络。（本型号里面只使用 11 位标识符，即只使用了标准帧）

节点 ID (Node ID)： 指定 PDO 的发送或接收节点。这是一个在 CANopen 网络中唯一的标识，用于确定数据流的源和目的地。

COB 类型： 指示数据帧的类型，如 TPDO（传输）或 RPDO（接收），详细内容下面有介绍。

10	9	8	7	6	5	4	3	2	1	0
Function Code				Node-ID						

Function Code: 数据传输的功能码，定义各种报文的传输级别，功能码越小，优先级越高。

Node-ID: 设备站号，取值范围为 1~127

预定义主/从连接集

对象	COB-ID
NMT Module Control	000H
SYNC 080H	080H
TIME SSTAMP	100H
对象	COB-ID
紧急	081H-0FFH
TPDO1 (发送)	181H-1FFH
RPDO1 (接收)	201H-27FH
TPDO2 (发送)	281H-2FFH
RPDO2 (接收)	301H-37FH
TPDO3 (发送)	381H-3FFH
RPDO3 (接收)	401H-47FH
TPDO4 (发送)	481H-4FFH
RPDO4 (接收)	501H-57FH
SDO (发送/服务器)	581H-5FFH
SDO (接收/客户)	601H-67FH
NMT Error Control	701H-77FH

注意：

- 1、COB-ID 越小，优先级越高；
- 2、每一个级别的 COB-ID 前面的功能码是固定格式；
- 3、COB-ID 为 00H、80H、100H、701H-77FH、081H-0FFH 均为系统管理格式。

通过 PDO COB-ID，CANopen 设备可以识别和传输实时数据。主站可以配置各个 PDO 通信对象的 COB-ID，从而实现不同设备之间的实时数据交换。

(3) 发送 PDO(TPDO)

通过 PDO，主站（控制器）可以向从站（设备）发送实时数据，发送 PDO 的功能码（COB-ID）

为:

- 0x180+伺服站号
- 0x280+伺服站号
- 0x380+伺服站号
- 0x480+伺服站号

(4) 接收 PDO(RPDO)

通过 PDO, 从站(设备)可以向主站(控制器)发送实时数据, 发送 PDO 的功能码(COB-ID)

为:

- 0x200+伺服站号
- 0x300+伺服站号
- 0x400+伺服站号
- 0x500+伺服站号

(5) PDO 传输方式

在 CANopen 通信协议中, PDO(Process Data Object)有两种主要的传输类型, 分别是同步传输和异步传输:

同步传输: 同步传输是由同步报文触发的传输方式, 其传输类型范围为 0 到 240。在这种传输模式下, 主站必须具备发送同步报文的能力, 通常以固定频率(最高可达 1KHZ)发送周期性的同步报文。一旦驱动器接收到同步报文, 它会在接收后立即发送 PDO 数据。同步传输可以分为两种方式:

- 非周期同步传输: 在这种方式下, PDO 数据可以由远程帧预触发传输, 或者根据设备子协议中定义的特定事件预触发传输。驱动器在接收到同步报文后, 将 PDO 数据立即发送一次。
- 周期同步传输: 在这种方式下, PDO 数据的传输会在每接收到 1 到 240 个 SYNC 消息后触发。驱动器将在接收到一定数量的同步报文后, 才发送一次 PDO 数据。

同步报文格式

COB-ID	DLC
0x80	0

异步传输: 异步传输的传输类型为 254 或 255。在异步传输中, 当从站报文数据发生改变时, 无论主站是否请求, 从站都会立即发送 PDO 数据。此外, 异步传输还可以定义相同报文两次发送之间的时间间隔, 以避免某些高优先级报文一直占用总线的情况。在异步传输中, PDO 的数值越低, 优先级越高。可以将异步传输总结为下面两种功能:

- 逢变即发功能: 从站报文数据改变后即发送, 不管主站是否询问, 而且可以定义同一个报文两次发送之间的时间间隔, 避免高优先级报文一直占据总线(PDO 的数值越低优先级越高)。
- 事件时间定时上报功能: 设置事件时间, 驱动器将向控制器周期性上传数据。

(6) PDO 禁止时间

在 CANopen 通信协议中, PDO 禁止时间(PDO Inhibit Time)是一个重要的参数, 用于控制 PDO 数据帧的传输间隔。禁止时间是指在 PDO 数据帧发送之后, 禁止再次发送 PDO 的一段时间, 以防止过于频繁的数据传输。

具体来说，当一个 PDO 数据帧被发送后，PDO 禁止时间会启动，从而阻止在一定时间内再次发送相同的 PDO 数据帧。这个时间间隔可以设置为一个固定的值，以确保数据传输不会过于密集。禁止时间的设置可以避免网络拥塞和数据冲突，从而提高通信的可靠性和效率。

在 CANopen 网络中，每个 PDO 通信对象都可以配置自己的禁止时间，以适应不同的应用场景和需求。通过合理设置禁止时间，可以确保 PDO 数据的稳定传输，避免数据丢失和冲突，从而实现可靠的实时数据交换和通信。

（7）心跳报文与节点保护

心跳报文和节点保护都是为了提高 CANopen 网络的可靠性和稳定性而设计的机制。心跳报文用于实时监测节点状态，节点保护则是在发生异常情况时采取预防措施，以保护节点和整个通信网络的安全和正常运行。

心跳报文：从站定期发送心跳报文到主站。如果主站在设定的时间内没有接收到下一个心跳报文，就会认为从站可能出现故障。

心跳报文格式——（0x700+节点号）+状态

状态——0：启动，4：停止，5：运行，127：预操作

节点保护：主站定期发送报文给从站，如果从站在设定的时间内没有收到主站的报文，就会触发报警。报警时间为“监督时间 × 寿命因子”。

主站请求报文格式——（0x700+节点号）（远程帧）

从站响应报文格式——（0x700+节点号）+状态：

状态——数据部分包括一个触发位（bit7），触发位必须在每次节点保护应答中交替置“0”或者“1”。

触发位在第一次节点保护请求时置为“0”。位 0 到位 6（bit0~6）表示节点状态：

0：初始化，1：未连接，2：连接，3：操作，4：停止，5：运行，127：预操作。

消费者心跳时间：消费者心跳时间包含了了节点保护时间、寿命因子，节点保护 ID 三个部分。其中 bit0~bit15 为心跳保护时间，bit16~bit23 为心跳保护 ID。这是为了方便用户使用衍生而来的。

SVD4812RC-AA 和 SVD4822RC-AA 两款驱动器支持心跳报文和节点保护两种保护模式。

（8）NMT 管理

在网络初始化过程中，CANopen 支持扩展的 boot-up，也支持最小化 boot-up 过程。可以用节点状态转换图表示这种初始化过程。

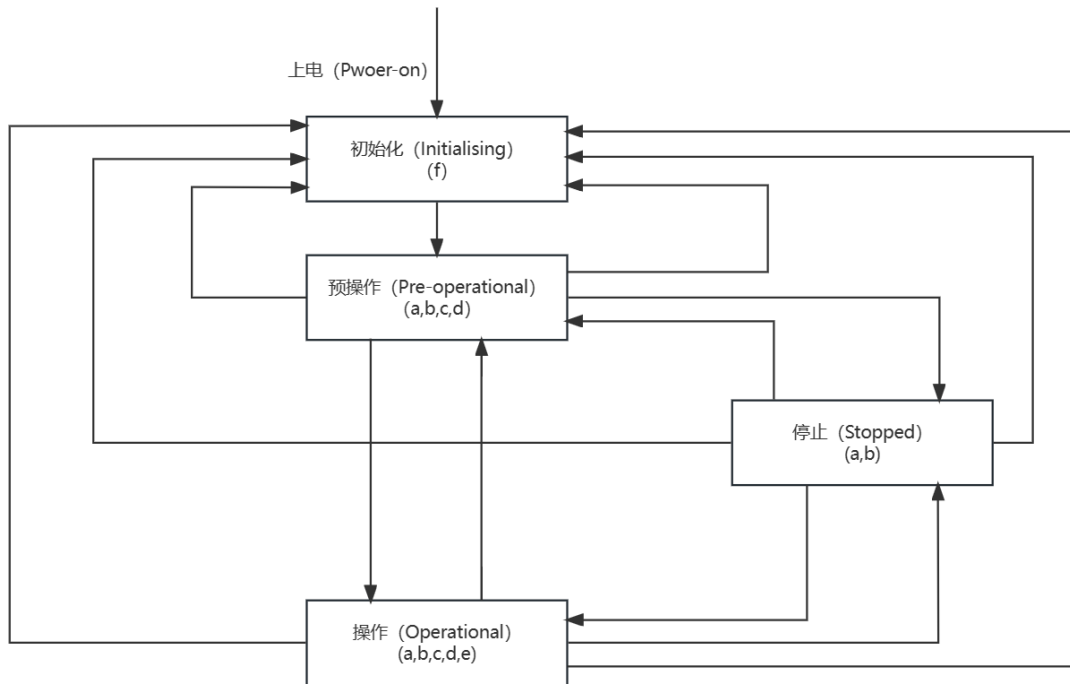


图 12-1 状态转换图

- a: MNT
- b: Node Guard
- c: SDO
- d: Emergency
- e: PDO
- f: Boot-up

初始化 (Initialising)：节点上电或复位后的第一个状态，节点执行基本的硬件和软件初始化，不能进行任何通信。

预操作 (Pre-operational)：节点初始化完成后自动进入的状态，节点可以进行 SDO 通信，但不能进行 PDO 通信。这个状态通常用于配置节点参数和 PDO 映射。

操作 (Operational)：节点可以进行所有类型的通信，包括 SDO 和 PDO。这个状态是节点正常工作的状态。（一般通过 NMT 管理进入操作状态：如 000 01 01，下文有详细介绍。）

停止 (Stopped)：节点停止所有通信，除了 NMT 和心跳（如果启用）。这个状态可以用于实现特定的应用行为。

NMT 管理报文格式如下

COB-ID	DLC	Byte0	Byte1
0x000	02	命令字	站号

命令字相关内容如下

命令字	NMT 服务
0x01	开启节点，开始 PDO 传输
0x02	关闭节点，关闭 PDO 传输
0x80	进入预操作状态
0x81	复位节点
0x82	复位通信

当 Node-ID=0 时，所有的 NMT 从设备被寻址。

注意:

可以通过 NMT 管理报文来实现在各种模式之间切换, 只有 NMT-Master 节点能够传送 NMT Module Control 报文, 所有从设备都必须支持 NMT 模块控制服务, 同时 NMT Module Control 消息不需要应答。只有处于操作状态下, PDO 才能传输 (一般通过 PDO)。

12.3 CAN 通讯实例

12.3.1 SDO 通讯实例

下面默认站号为 01

(1) 写速度模式

关于目标速度等参数等转换详情, 可以参考 UART 通讯实例, 也可以参考第六章对象列表相关参数的转换公式, 这里不做过多解释。

10rpm 转换为十六进制为 5 7619

50rps/s 转为十六进制为 1 A36E

CAN 地址	变量名称	设定值	报文	备注
60600008	工作模式	3	601 2F 60 60 00 03 00 00 00	工作模式设置为 3
60FF0020	目标速度	10rpm	601 23 FF 60 00 19 76 05 00	目标速度设置为 10rpm
60830020	轮廓加速度	50rps/s	601 23 83 60 00 6E A3 01 00	轮廓加速度设置为 50rps/s
60840020	轮廓减速度	50rps/s	601 23 84 60 00 6E A3 01 00	轮廓减速度设置为 50rps/s
60400010	控制字	F	601 2B 40 60 00 0F 00 00 00	将控制字设为 F, 对电机锁轴

按照上述操作执行, 电机将以轮廓加速度为 50rps/s 加速至 10rpm, 若再对目标速度写 0, 电机将以轮廓减速度为 100rps/s 减至 0。

(2) 写相对位置模式

关于目标速度等参数等转换详情, 可以参考 UART 通讯实例, 也可以参考第六章对象列表相关参数的转换公式, 这里不做过多解释。

50rpm 可转换为十进制 1789569, 十六进制为 1B 4E81

10000 转为十六进制为 2710

CAN 地址	变量名称	设定值	报文	备注
60600008	工作模式	1	601 2F 60 60 00 01 00 00 00	工作模式设置为 1
60810020	轮廓运行速度	50rpm	601 23 81 60 00 81 4E 1B 00	将轮廓运行速度设置为 50 rpm
607A0020	目标位置	10000 inc	601 23 7A 60 00 10 27 00 00	将目标设为 10000 inc
60400010	控制字	0x4F->0x5F	601 2B 40 60 00 4F 00 00 00	控制字从 4F

			601 2B 40 60 00 5F 00 00 00	设为 5F
--	--	--	-----------------------------	-------

按照上述操作执行，电机在原有的位置上，再以 50rpm 的速度运行 10000inc 距离。

(3) SDO 读取状态字、实际速度

通过查询对象列表可知

状态字 CAN 地址：6041002B

实际位置 CAN 地址：60630023

在上面 SDO 部分已经介绍了，执行读取指令时命令字都使用 0x40

发送报文目的	发送报文	接受报文
①获取状态字	601 40 41 60 00 00 00 00 00	581 4B 41 60 00 31 02 00 00
②获取实际位置	601 40 63 60 00 00 00 00 00	581 43 63 60 00 0D 0D 00 00
③获取实际速度	601 40 6C 60 00 00 00 00 00	581 43 6C 60 00 00 DB FF FF

对上述回应报文解析：

①获取状态字：581 4B 41 60 00 31 02 00 00

4B 表示返回的数据是一个 16 位的数据，将返回的数据（3102）转换成十六进制则为 0231，表示状态字为 231。

②获取实际位置：581 43 63 60 00 0D 0D 00 00

43 表示返回的数据是一个 32 位的数据，将返回的数据（0D0D 0000）转换为十六进制则为 0D0D，再次转换为十进制为 3341

③获取实际速度：581 43 6C 60 00 00 DB FF FF

43 表示返回的数据是一个 32 位的数据，将返回的数据（00DB FFFF）转换为十六进制转为 FFFF 00DB，由于其最高位为 F，所以实际值为负数-65317，转换位 rpm 即-2。

注意：

负的速度及位置最高位为 1，且以补码形式进行传输。若实际位置为-5000，若在计算机里面表示则为 FFFF EC78。

12.3.2 PDO 配置

若通过上位机进行配置则可以按照一下步骤：

在工作区点击“参数列表”->点击“TPDO 配置”

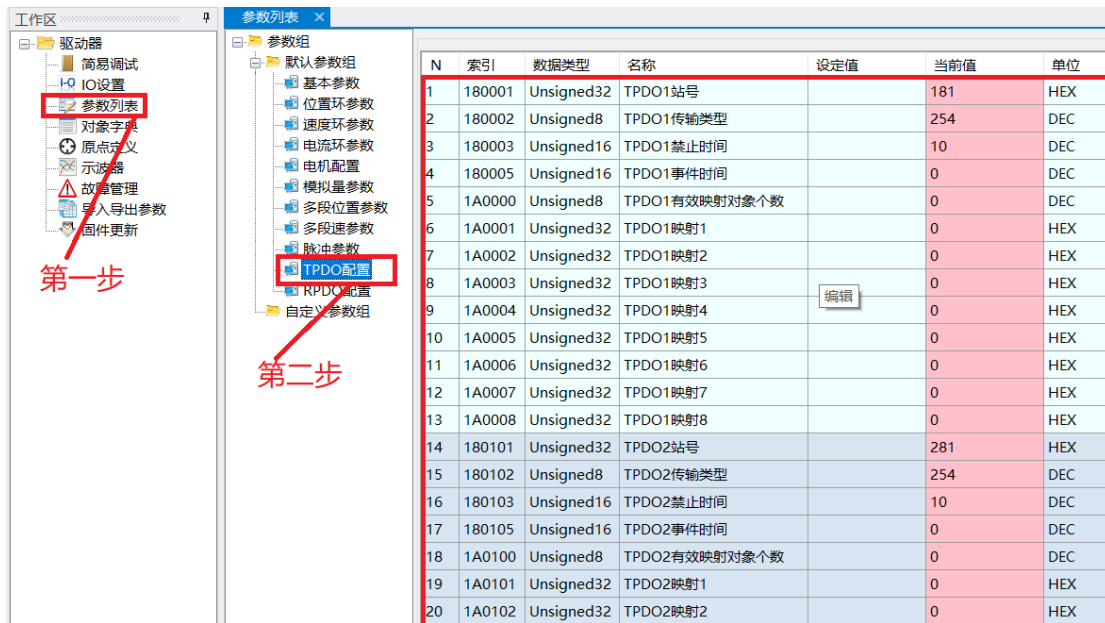


图 12-2 TPDO 配置

TPDO 站号：由驱动器拨码开关决定，详情请看第四章 系统接口及配线

TPDO 传输类型：

- 0:非周期同步模式，即收到同步报文且数据发生改变就发送
- 1~240:周期同步，即收到 x 个同步报文就发送数据，x 为设置值
- 241~253:保留
- 254/255:非周期非同步，此时“事件时间”有效。事件时间非 0，则经过事件时间之后立即发送；为 0，则数据发送改变且距离上次发送时间已大于禁止时间，即发送

TPDO 禁止时间：间隔一段时间后发送 PDO 数据帧，避免网络拥塞和数据冲突，单位为 ms

TPDO 事件时间：驱动器向控制器发送 PDO 的周期时间，单位 ms，

TPDO 有效映射对象个数：所设置的映射个数。

TPDO 映射 1-8：配置 CANopen 控制对象

注意：

使用异步传输逢变即发功能，应设置相应的禁止时间，将事件时间设置为 0；

使用时间定时上报功能：应设置相应的事件时间，将禁止时间设置为 0

每个 PDO 里面所映射对象长度之和不能超过 8 个字节

SDO 通讯常用对象列表

CANopen 地址	参数名称	含义	默认值
10050020	同步 ID	传输类型为 1-240 同步模式时有用，异步模式时不需要设置。	80
100C0010	节点保护时间	主站以节点保护时间周期性地发送远程帧，询问从节点状态。从节点需在一定时间内回应，否则主站视为从节点掉线，驱动器报警。 保护时间*寿命因子=节点保护的寿命时间	1000
100D0008	寿命因子		3
100E0020	节点保护 ID	700+驱动器 ID	
10140020	紧急报文站号	80+驱动器站号	
10160120	消费者	"bit0~bit15:心跳保护时间	

	心跳时间	bit16~bit23:心跳保护 ID"	
2F800208	CAN 波特率	CAN 波特率设置 100: 1M 50: 500k 25: 250k 12: 125k 10: 100k 5: 50k 2: 20k	50
30110410	ECAN 同步丢失计数	同步模式下监控通讯状态, 数值不发送变化说明通讯状况良好, 若数值不断变化说明有干扰或同步周期设置不正确。	
60070010	通讯中断模式	CAN 通讯中断模式, 超过设定时间后仍没有收到节点保护报文的动作逻辑。 0: 不处理 1/2: 报错, 并松轴 3: 警告, 并急停"	0

12.3.3 NMT 管理实例

由 12.2.3 小节可知, NMT 管理报文格式为

COB-ID	DLC	Byte0	Byte1
0x000	02	命令字	站号

下面以 1 站号为例:

开启节点的命令字为 0x01, 则开启 PDO 节点报文为: 000 01 01;

关闭节点的命令字为 0x02, 则关闭 PDO 节点报文为: 000 02 01;

复位节点的命令字为 0x81, 则关闭 PDO 节点报文为: 000 81 01;

2.3.4 PDO 通讯实例

(1) TPDO 同步通讯实例

1	180001	Unsigned32	TPDO1站号		181	HEX
2	180002	Unsigned8	TPDO1传输类型	3	3	DEC
3	180003	Unsigned16	TPDO1禁止时间	20	20	DEC
4	180005	Unsigned16	TPDO1事件时间	0	0	DEC
5	1A0000	Unsigned8	TPDO1有效映射对象个数	2	2	DEC
6	1A0001	Unsigned32	TPDO1映射1	60410010	60410010	HEX
7	1A0002	Unsigned32	TPDO1映射2	60630020	60630020	HEX
8	1A0003	Unsigned32	TPDO1映射3		0	HEX
9	1A0004	Unsigned32	TPDO1映射4		0	HEX
10	1A0005	Unsigned32	TPDO1映射5		0	HEX

图 12-3 参数映射

我们将 TPD01 站号设置为 181. 传输类型设为 3 (接受到 3 个同步报文, 发送一次数据), 禁止时间设为 20, TPD01 映射 1 映射了 60410010, TPD01 映射 2 映射了 60630020。

发送报文: 000 01 01 以此开启报文 (没有回应报文)

000: COB-ID 01: 开启, 若输入 02 则为关闭 01: 站号

下面通过“创芯科技”的 CAN 接口软件进行演示



发送三次 0x80 (0x80 指的是 COB-ID) 后, 即可接收到驱动器回复的报文



收到的报文如下: B1 02 17 F0 06 00

表示状态字 (60410010) 里面的数据为 02B1

实际位置 (60630020) 里面的数据为 0006F017 (454679)

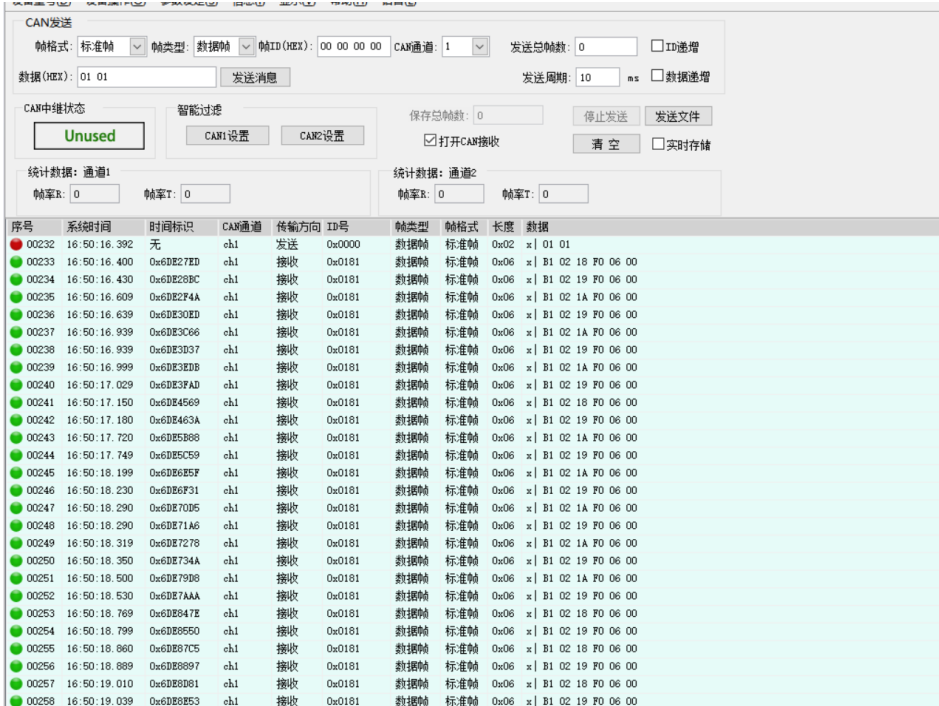
(2) TPDO 异步通讯实例

N	索引	数据类型	名称	设定值	当前值	单位
1	180001	Unsigned32	TPDO1站号		181	HEX
2	180002	Unsigned8	TPDO1传输类型	254	254	DEC
3	180003	Unsigned16	TPDO1禁止时间	20	20	DEC
4	180005	Unsigned16	TPDO1事件时间		0	DEC
5	1A0000	Unsigned8	TPDO1有效映射对象个数	2	2	DEC
6	1A0001	Unsigned32	TPDO1映射1	60410010	60410010	HEX
7	1A0002	Unsigned32	TPDO1映射2	60630020	60630020	HEX
8	1A0003	Unsigned32	TPDO1映射3		0	HEX
9	1A0004	Unsigned32	TPDO1映射4		0	HEX
10	1A0005	Unsigned32	TPDO1映射5		0	HEX
11	1A0006	Unsigned32	TPDO1映射6		0	HEX
12	1A0007	Unsigned32	TPDO1映射7		0	HEX
13	1A0008	Unsigned32	TPDO1映射8		0	HEX

图 12-4 参数映射

我们将 TPDO1 站号设置为 181。传输类型设为 254，禁止时间设为 20，TPDO1 映射 1 映射了 60410010，TPDO1 映射 2 映射了 60630020。

则发送报文：000 01 01 以此开启报文即可接收到驱动器回复的报文。当数据有变化时每隔 20ms（TPDO 禁止时间）返回一次数据。



该截图展示了 CAN 发送配置界面。在“CAN发送”区域，格式选择为“标准帧”，数据类型为“数据帧”，帧ID (HEX) 为“00 00 00 00”，CAN通道为“1”，发送总帧数为“0”，并勾选了“ID递增”。数据 (HEX) 输入为“01 01”，并勾选了“发送消息”。在“CAN中继状态”区域，显示为“Unused”，并勾选了“打开CAN接收”。下方有“统计数据：通道1”和“统计数据：通道2”的显示框，均显示帧率为0。

序号	系统时间	时间标识	CAN通道	传输方向	ID号	帧类型	帧格式	长度	数据
00232	16:50:16.392	无	ch1	发送	0x0000	数据帧	标准帧	0x02	x 01 01
00233	16:50:16.400	0x6DE27ED	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 18 F0 06 00
00234	16:50:16.430	0x6DE28DC	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00235	16:50:16.609	0x6DE2F4A	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00236	16:50:16.639	0x6DE30ED	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00237	16:50:16.939	0x6DE3C66	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00238	16:50:16.939	0x6DE3D37	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00239	16:50:16.999	0x6DE3EDB	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00240	16:50:17.029	0x6DE3FAD	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00241	16:50:17.150	0x6DE4669	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 18 F0 06 00
00242	16:50:17.180	0x6DE463A	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00243	16:50:17.720	0x6DE5B98	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00244	16:50:17.749	0x6DE5C59	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00245	16:50:18.199	0x6DE6E5F	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00246	16:50:18.230	0x6DE6F31	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00247	16:50:18.290	0x6DE70D5	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00248	16:50:18.290	0x6DE71A6	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00249	16:50:18.319	0x6DE7278	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00250	16:50:18.350	0x6DE734A	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00251	16:50:18.500	0x6DE79D8	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 1A F0 06 00
00252	16:50:18.530	0x6DE7AAA	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00253	16:50:18.769	0x6DE847E	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 18 F0 06 00
00254	16:50:18.799	0x6DE8E50	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00255	16:50:18.860	0x6DE87C5	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 18 F0 06 00
00256	16:50:18.889	0x6DE8987	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00
00257	16:50:19.010	0x6DE8D81	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 18 F0 06 00
00258	16:50:19.039	0x6DE8E53	ch1	接收	0x0181	数据帧	标准帧	0x06	x B1 02 19 F0 06 00

接收到的报文如下：B1 02 17 F0 06 00

表示状态字（60410010）里面的数据为 02B1

实际位置（60630020）里面的数据为 0006F017（454679）

若需要定时上报，只需要把 TPDO 禁止时间改为 0，TPDO 事件时间设置为相应时间（比如 50），驱动器将定时发送 TPDO 数据（不管数据是否有变化）。

（3）RPDO 通讯实例

N	索引	数据类型	名称	设定值	当前值	单位
1	140001	Unsigned32	RPDO1站号		201	HEX
2	140002	Unsigned8	RPDO1传输类型	254	254	DEC
3	140003	Unsigned16	RPDO1禁止时间	10	10	DEC
4	160000	Unsigned8	RPDO1有效映射对象个数	2	2	DEC
5	160001	Unsigned32	RPDO1映射1	60FF0020	60ff0020	HEX
6	160002	Unsigned32	RPDO1映射2	60400010	60400010	HEX
7	160003	Unsigned32	RPDO1映射3		0	HEX
8	160004	Unsigned32	RPDO1映射4		0	HEX
9	160005	Unsigned32	RPDO1映射5		0	HEX

我们将 RPDO1 站号设置为 201，传输类型设为 254，禁止时间设为 10，RPDO1 映射 1 映射了 60FF0020（目标速度），TPDO1 映射 2 映射了 60400010（控制字）。

发送报文：000 01 01 以此开启节点（无返回报文）

再发送 201 03 9D 36 00 0F 00 即可将多个参数同上写入驱动器（无返回报文）

N	索引	数据类型	名称	设定值	当前值	单位
7	604000	Unsigned16	控制字		f	HEX
8	607A00	Integer32	目标位置		0	inc
9	608100	Unsigned32	轮廓运行速度		99.999	rpm
10	608300	Unsigned32	轮廓加速度		49.997	rps/s
11	608400	Unsigned32	轮廓减速度		49.997	rps/s
12	60FF00	Integer32	目标速度		99.999	rpm

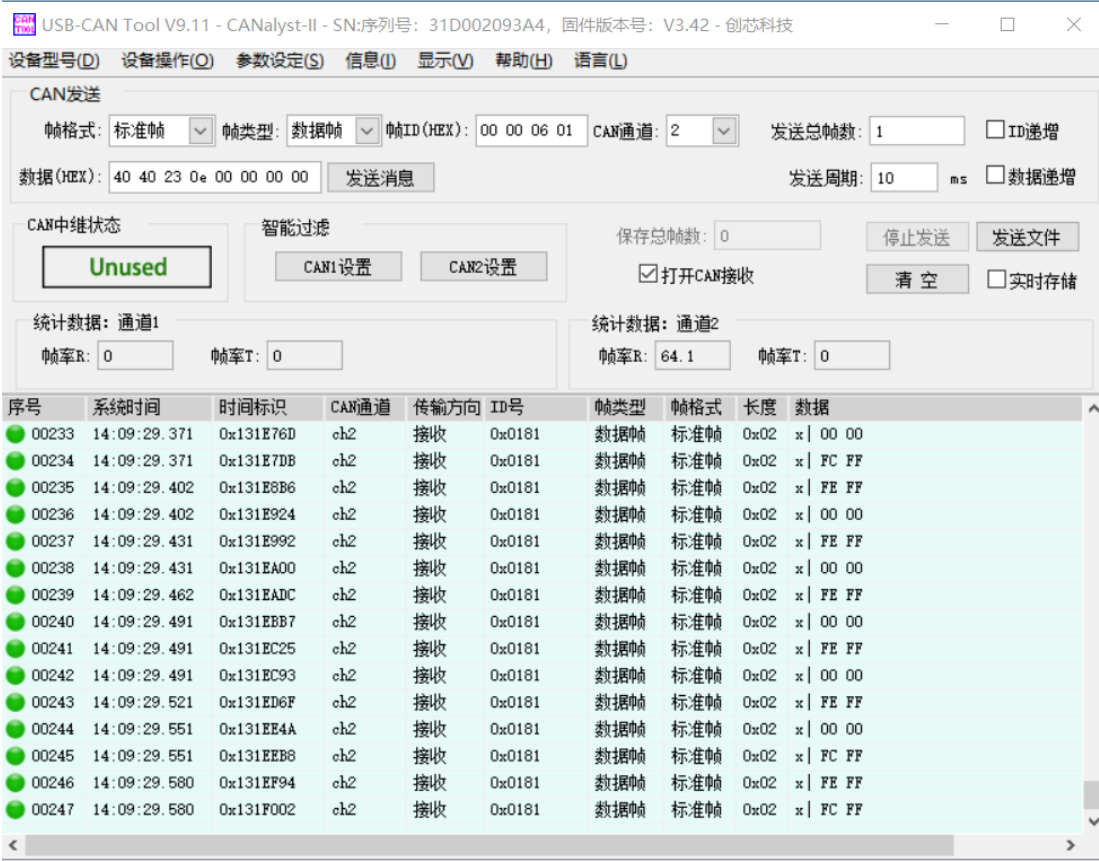
（4）keba

本驱动器除了通过 NMT 管理开启 pdo，还可以通过修改 0x23400E——keba 的值来开启 pdo，实例如下：

- 1、先设置 PDO 相关参数，这里我们设置通过 TPDO 读取实际电流的值

N	索引	数据类型	名称	设定值	当前值	单位
1	180001	Unsigned32	TPDO1站号		181	HEX
2	180002	Unsigned8	TPDO1传输类型		254	DEC
3	180003	Unsigned16	TPDO1禁止时间		10	DEC
4	180005	Unsigned16	TPDO1事件时间		0	DEC
5	1A0000	Unsigned8	TPDO1有效映射对象个数	1	1	DEC
6	1A0001	Unsigned32	TPDO1映射1	60780010	60780010	HEX
7	1A0002	Unsigned32	TPDO1映射2		0	HEX
8	1A0003	Unsigned32	TPDO1映射3		0	HEX
9	1A0004	Unsigned32	TPDO1映射4		0	HEX
10	1A0005	Unsigned32	TPDO1映射5		0	HEX
11	1A0006	Unsigned32	TPDO1映射6		0	HEX
12	1A0007	Unsigned32	TPDO1映射7		0	HEX
13	1A0008	Unsigned32	TPDO1映射8		0	HEX

2、接着修改 keba 为 1， pdo 开启自动上传实际电流数值



The screenshot shows the USB-CAN Tool V9.11 interface. The top menu bar includes options like '设备型号(D)', '设备操作(O)', '参数设定(S)', '信息(I)', '显示(V)', '帮助(H)', and '语言(L)'. The main area is divided into several sections:

- CAN发送 (CAN Send):** Includes fields for '帧格式' (Standard Frame), '帧类型' (Data Frame), '帧ID (HEX)' (00 00 06 01), 'CAN通道' (2), '发送总帧数' (1), and '发送周期' (10 ms).
- CAN中继状态 (CAN Relay Status):** Shows 'Unused' and buttons for 'CAN1设置' and 'CAN2设置'.
- 智能过滤 (Smart Filter):** Includes '保存总帧数' (0) and '打开CAN接收' (checked).
- 统计数据 (Statistics):** Shows frame rates for '通道1' and '通道2'.
- 数据列表 (Data List):** A table with columns: 序号, 系统时间, 时间标识, CAN通道, 传输方向, ID号, 帧类型, 帧格式, 长度, 数据. It shows a series of received frames on channel 2 with ID 0x0181.

3、若要关闭 PDO，需重新修改为 0，然后存储控制参数并重启

12.3.5 节点保护实例

报文无数据，注意要设置为远程帧

主机发送报文：0x700+站号

从站（驱动器）响应报文：0x700+节点号+状态

下面以站号 1 为例

则主机发送报文：0x701（连续发送状态下）

从机回应报文：0x701 7F

0x701 FF

在连续发送状态下，从机轮流回应两条报文，即触发位在每次节点保护应答中交替置“0”或者“1”（上面节点保护中有提到）

12.3.6 心跳报文实例

CAN 地址	变量名称	设置值	发送及回复报文
10170010	生产者心跳时间	1000	601 2B 17 10 00 E8 03 00 00
			581 60 17 10 00 E8 03 00 00

按照上述进行设置以及发送报文，每隔 1s 从站（驱动器）将上报报文

我们再次将生产者心跳时间清零，即可关断回复报文。

00125	15:15:39.011	0xB2B6163	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00126	15:15:40.032	0xB2B8870	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00127	15:15:41.023	0xB2BAF7E	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00128	15:15:42.012	0xB2BD68B	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00129	15:16:03.043	0xB2F0B13	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00130	15:16:04.061	0xB2F321E	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00131	15:16:05.052	0xB2F592C	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00132	15:16:06.042	0xB2F8039	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00133	15:16:07.061	0xB2FA747	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F
00134	15:16:08.052	0xB2FCE54	ch2	接收	0x0701	数据帧	标准帧	0x01	x 7F

图 12-5 心跳报文返回

附录一 配置第三方电机

如果您想要配置第三方电机，您需要注意以下几点：

- 您需要选择与富兴机电的伺服驱动器兼容的第三方电机，例如具有相同或相近的额定电压、额定电流、额定扭矩等参数的电机。
- 您需要根据您的应用场景和需求，选择合适的电机型号和规格，电机最高转速、额定扭矩，额定电流等。
- 您需要按照富兴机电提供的说明书，正确地连接电机和驱动器的通讯线和电源线，并设置好相关的参数，如通讯协议、控制模式等。
- 您需要测试电机的运行效果，如转速、扭矩、温度等，并根据实际情况进行调整或优化。

进行第三方电机配置，根据根据第三方电机图纸，确认编码器类型为增量式还是通讯式，之后找到下表相关参数。

增量式：

参数名称	参数设定值
电机型号	XDDA
反馈类型	Bit0:ABZ 接线检查 bit1:UVW 接线检查 bit2:UVW 为 OC 输出 例如：编码器 ABZ、UVW 全部为差分输入，输入 3 编码器 ABZ 为差分，UVW 为 OC 输入，则输入 5
反馈精度	将单位改为 DEC 编码器线数×4 例如：编码器线数为 2500 线，则输入 10000
反馈周期	同反馈精度
电机极对数	参考电机图纸
励磁模式	20
励磁电流	默认为 3.295 Ap
励磁时间	默认设为 1000ms
电机 iit 电流	将单位改为 Arms, 然后输入电机额定电流
电机 iit 时间	默认设为 60 秒
电机最大电流	将单位改为 Arms, 然后输入电机最大电流
相电感	参考电机图纸
相电阻	参考电机图纸
反向电动势	参考电机图纸
扭矩系数	参考电机图纸
转子惯量	参考电机图纸
抱闸占空比	默认为 70%
抱闸延时	默认为 150ms
电机电流环带宽	默认设为 2000Hz

通讯式：

参数名称	参数设定值
电机型号	单圈绝对值：XTDA 多圈绝对值：XMDA
反馈类型	8
反馈精度	编码器单圈位数为：x 则输入的数值= 2^x 例如：单圈位数为 17 位 则输入 131072
反馈周期	将单位改为 HEX 高 8 位为多圈位数，低 8 位为单圈位数。 例如：单圈绝对值 17 位编码器， 则输入 17 多圈绝对值 17 位编码器，则输入 1617
电机极对数	参考电机图纸
励磁模式	20
励磁电流	默认为 3.295 Ap
励磁时间	默认设为 1000ms
电机 iit 电流	将单位改为 Arms 然后输入电机额定电流
电机 iit 时间	默认设为 60 秒
电机最大电流	将单位改为 Arms 然后输入电机最大电流
相电感	参考电机图纸
相电阻	参考电机图纸
反向电动势	参考电机图纸
扭矩系数	参考电机图纸
转子惯量	参考电机图纸
抱闸占空比	默认为 70%
抱闸延时	默认为 150ms
电机电流环带宽	默认设为 2000Hz

1. 在上位软件的电机配置界面，按照上表正确输入电机参数。
2. 存储电机参数后重启驱动器。
3. 电机初始化。
4. 存储控制参数。
5. 在电机配置界面中的“相位校正控制”栏输入“1”。

11	608400	Unsigned32	轮廓减速度		7.629	rps/s
12	60FF00	Integer32	目标速度	500	499.877	rpm
13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		32.980	Ap
15	232001	Unsigned16	相位数据1		282	DEC
16	232002	Unsigned16	相位数据2		111	DEC
17	232003	Unsigned16	相位数据3		198	DEC
18	232004	Unsigned16	相位数据4		457	DEC
19	232005	Unsigned16	相位数据5		367	DEC
20	232006	Unsigned16	相位数据6		29	DEC
21	232007	Unsigned16	索引相位数据		1000	DEC
22	232008	Unsigned8	相位校正控制	1	0	DEC
23	2F8004	Unsigned16	RS485波特率		19200	Bandr...

6. “相位校正控制”写入 1 开始自整定霍尔角度。

7. 等待“相位校正控制”变为 0 时，即自整定完成。

12	60FF00	Integer32	目标速度	500	499.877	rpm
13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		32.980	Ap
15	232001	Unsigned16	相位数据1		0	DEC
16	232002	Unsigned16	相位数据2		0	DEC
17	232003	Unsigned16	相位数据3		0	DEC
18	232004	Unsigned16	相位数据4		0	DEC
19	232005	Unsigned16	相位数据5		0	DEC
20	232006	Unsigned16	相位数据6		0	DEC
21	232007	Unsigned16	索引相位数据		1000	DEC
22	232008	Unsigned8	相位校正控制	1	3	DEC
23	2F8004	Unsigned16	RS485波特率		19200	Bandr...

13	607100	Integer16	目标扭矩%		0	%
14	607300	Unsigned16	目标电流限制		32.980	Ap
15	232001	Unsigned16	相位数据1		70	DEC
16	232002	Unsigned16	相位数据2		111	DEC
17	232003	Unsigned16	相位数据3		198	DEC
18	232004	Unsigned16	相位数据4		457	DEC
19	232005	Unsigned16	相位数据5		411	DEC
20	232006	Unsigned16	相位数据6		28	DEC
21	232007	Unsigned16	索引相位数据		1000	DEC
22	232008	Unsigned8	相位校正控制	1	0	DEC
23	2F8004	Unsigned16	RS485波特率		19200	Bandr...

8. 存储电机参数后重启驱动器。

9. 将目标电流限设为 $5A_{peak}$ ，然后使能电机，给速度看电机能否运行。

10. 若电机无法运行，可重复第 6 步。

11. 若多次尝试仍无法运行，请尝试将电机参数界面的“电机选旋转方向”修改为“1”。

参数列表		简易调试	读写参数配置	故障管理		
N	索引	数据类型	名称	设定值	当前值	单位
3	641001	Unsigned32	电机型号		XDDA	ASCII
4	641002	Unsigned8	反馈类型		3	HEX
5	641003	Unsigned32	反馈精度		4096	inc/r
6	641004	Unsigned32	反馈周期		0	HEX
7	641005	Unsigned8	电机极对数		5	2p/r
8	641006	Unsigned8	励磁模式		20	DEC
9	641007	Integer16	励磁电流		3.295	Ap
10	641008	Unsigned16	励磁时间		1000	ms
11	641009	Unsigned16	电机iit电流		14.1	Ap
12	64100A	Unsigned16	电机iit时间		59.904	S
13	64100B	Unsigned16	电机最大电流		43.8	Ap
14	64100C	Unsigned16	相电感		0.52	mH
15	64100D	Unsigned16	相电阻		0.28	Ohm
16	64100E	Unsigned16	反向电动势		8.3	Vrms/...
17	64100F	Unsigned16	扭矩系数		0.13	Nm/A...
18	641010	Unsigned16	转子惯量		0.58	kg*c...
19	641013	Unsigned8	电机旋转方向	1	1	DEC
20	300303	Unsigned8	绝对值编码器复位命令		0	HEX
21	232008	Unsigned8	相位校正控制		0	DEC
22	232007	Unsigned16	索引相位数据		1000	DEC
23	641017	Unsigned8	电机抱闸		0	DEC
24	64101E	Unsigned8	抱闸电压		24	V

12. 存储电机参数后重启，之后重复第 6 步。

附录二 制动电阻使用

驱动器里面的制动电阻是一种用于消耗电机制动时产生的再生能量的元件,它可以保护驱动器不被过压损坏,同时实现快速停机或减速的效果。制动电阻的原理是将电机的动能和磁能转化为热能,通过电阻发热的方式散发出去,从而降低直流母线上的电压。

本型号驱动器通过 CN9 的 RB+、RB-进行制动电阻的连接,同时通过软件设置正确的制动电阻阻值、制动电阻功率等。制动电阻的选择与电机型号相关,具体选型可以参照下表

电机功率	制动电阻阻值[Ω]	制动电阻功率[W]	制动电阻耐压[VDC]
50W	27	100	500
100W	10	100	500
200W	5	100	500
400W	3.5	200	500
750W	0.8	700	500

参数名称	UART 通讯地址	对象属性	单位	参数描述
制动电阻阻值	60F701	Unsigned16 RW	Ω	制动电阻阻值
制动电阻功率	60F702	Unsigned16 RW	W	制动电阻标称功率
制动电阻 时间常数	60F703	Unsigned16 RW	S	制动电阻时间常数 时间为 N*256/1000
斩波电压点	651006	Unsigned16 RW	V	驱动器斩波电压点

附录三 保险丝规格选择

电机功率 (W)	保险丝参考规格
50	3A/58VDC
100	5A/58VDC
200	10A/58VDC
400	20A/58VDC
750	40A/58VDC