

CyberGear 微电机使用说明书

目录

目录
注意事项
法律声明
售后政策

1. 电机规格参数

- 1.1 外形及安装尺寸
- 1.2 标准使用状态
- 1.3 电气特性
- 1.4 机械特性

2. 驱动器产品信息

- 2.1 驱动器外观简介&产品规格
- 2.2 驱动器接口定义
 - 2.2.1 驱动器接口图
 - 2.2.2 驱动器接口推荐品牌及型号
 - 2.2.3 驱动器接口引脚定义
- 2.3 驱动器指示灯定义
- 2.4 主要器件及规格

3. 调试器使用说明

- 3.1 硬件配置
- 3.2 调试器界面及说明
- 3.3 电机设置
- 3.4 控制演示
- 3.5 固件更新

4. 驱动器通信协议及使用说明

- 4.1 通信协议类型说明
- 4.2 控制模式使用说明

注意事项

- 1、请按照本文规定的工作参数使用，否则会对本产品造成严重的损坏！
- 2、在关节运行时不可切换控制方式，如需切换需要发送停止运行命令后再做切换。
- 3、使用前请检查各部件是否完好，如发生部件缺失、损坏请及时联系技术支持。
- 4、请勿随意拆卸电机，以免出现无法恢复的故障。
- 5、确保电机连接时无短路，接口按要求正确连接。

法律声明

在使用本产品前，请用户务必仔细阅读本手册，按照本手册内容操作本产品。如用户违反本手册内容使用本产品，造成的任何财产损失、人身伤害事故，本公司不承担任何责任。因本产品由众多零部件构成，切勿让儿童接触本产品，以免发生意外事故。为延长产品使用寿命，请勿在高温、高压环境中使用本产品。本手册在印刷时已尽可能的包含各项功能介绍和使用说明。但由于产品功能不断完善、设计变更等，仍可能与用户购买的产品有不符之处。

本手册与实际产品在颜色、外观等方面可能有所偏差，请以实际产品为准。本手册由小米或其当地的子公司出版，小米随时可能对本手册中的印刷错误、不准确的最新信息进行必要的改进和更改，或对程序和/或设备进行改进，恕不另行通知。此类更改将上传到本手册的新版本中，请扫描本手册二维码进行获取。所有图片仅供功能说明参考，请以实物为准。

售后政策

本产品售后服务严格依据《中华人民共和国消费者权益保护法》、《中华人民共和国产品质量法》实行售后服务，服务内容如下：

1、保修期限及内容

(1) 凡在线上渠道下单购买本产品的用户，可在自签收次日起七日内享受无理由退货服务。退货时用户须出示有效购买凭证，并退回发票。用户须保证退货商品保持原有品质和功能、外观完好、商品本身及配件的商标和各种标识完整齐全，如有赠品需一并退回。如果商品出现人为损坏、人为拆机、包装箱缺失、零配件缺失的情况，不予办理退货。退货时产生的物流费用由用户承担（收费标准见“售后服务收费标准”）。如果用户未结清物流费用，将按实际发生额从退款金额中扣除。自收到退货商品之日起七日内向用户返还已支付的货款。退款方式与付款方式相同。具体到账日期可能会受银行、支付机构等因素影响。

(2) 自用户签收次日起7天内，发生非人为损坏性能故障，经由小米售后服务中心检测确认后，为用户办理退货业务，退货时用户须出示有效购买凭证，并退回发票。如有赠品需一并退回。

(3) 自用户签收次日起 7 天后至 15 天内，发生非人为损坏性能故障，经由小米售后服务中心检测确认后，为用户办理换货业务，更换整套商品。换货后，商品本身三包期重新计算。

(4) 自用户签收次日起 15 天后至 365 天内，经由小米售后服务中心检测确认后，属于产品本身质量故障，可免费提供维修服务。更换的故障产品归小米公司所有。无故障产品，将原样返回。本产品经过各项严格检测后出厂，如有非产品本身质量故障，我们将有权拒绝用户的退换货需求。

本手册售后政策若与店铺售后政策不一致的，以店铺的售后政策为准。

2、非保修条例以下情况不属于保修范围：

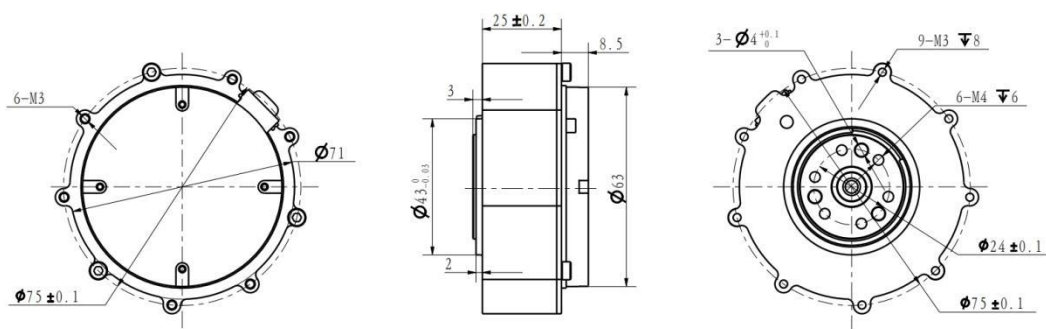
1. 超出保修条款所限定的保修期限。
2. 未按照说明书要求，错误使用造成的产品损坏损毁。
3. 不当的操作、维修、安装、改装、测试等不正当使用造成的损坏损毁。
4. 非质量故障引起的常规机械损耗、磨损。
5. 非正常工况下造成的损坏，包括但不限于跌落、撞击、液体浸入、剧烈撞击等。
6. 天灾（如水灾、火灾、雷击、地震等）或不可抗击力造成的损坏。
7. 超过峰值扭矩使用造成的损坏。
8. 非小米原装正品或无法提供合法购买凭证。
9. 其他非产品的设计、技术、制造、质量等问题导致的故障或损坏。
10. 将本产品应用于商业用途。

如果出现上述情况，用户需自行支付费用。

集团售后政策详见：<https://www.mi.com/service/serviceAgreement?id=17>

1 电机规格参数

1.1 外形及安装尺寸



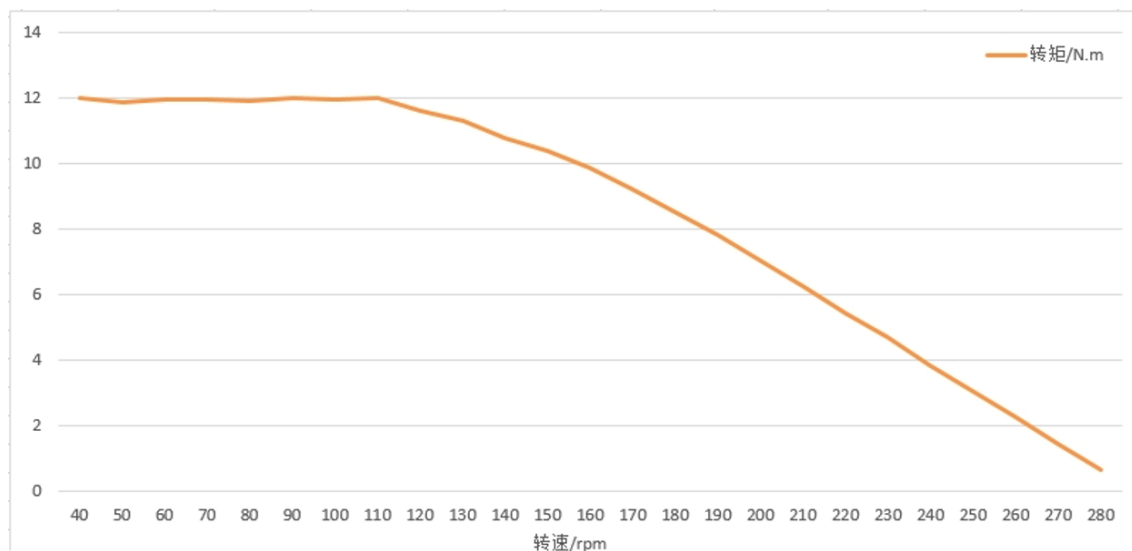
1.2 标准使用状态

- 1.2.1 额定电压： 24 VDC
- 1.2.2 使用电压范围： 16V—28 VDC
- 1.2.3 额定负载（CW）： 4 N.m

- 1.2.4 运转方向: CW/CCW 从出轴方向看
- 1.2.5 使用姿势: 出轴方向为水平或者垂直
- 1.2.6 标准使用温度: $25 \pm 5^{\circ}\text{C}$
- 1.2.7 使用温度范围: $-20 \sim 50^{\circ}\text{C}$
- 1.2.8 标准使用湿度: 65%
- 1.2.9 使用湿度范围: 5~85%, 无凝露
- 1.2.10 保存温度范围: $-30 \sim 70^{\circ}\text{C}$
- 1.2.11 绝缘等级: Class B

1.3 电气特性

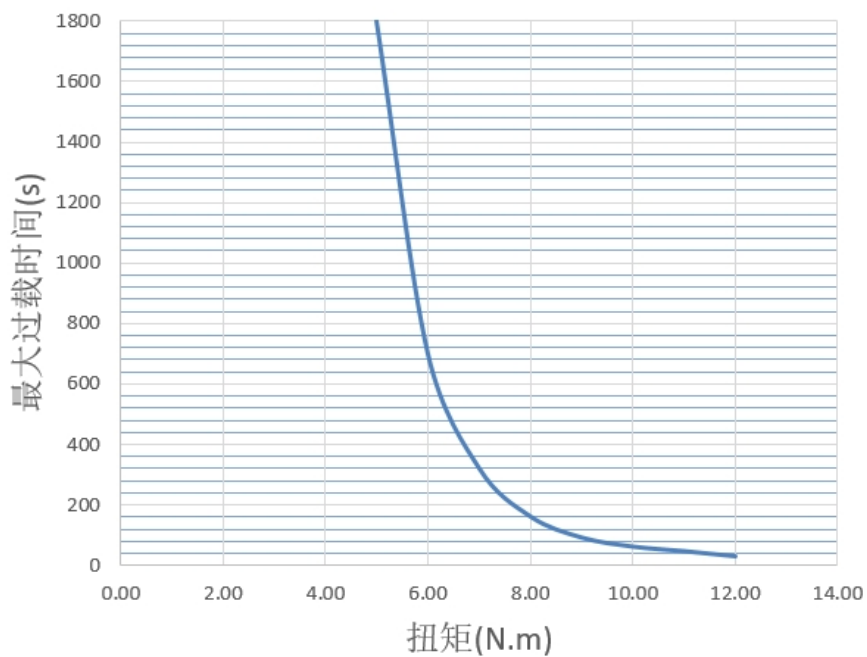
- 1.3.1 空载转速: $296 \text{ rpm} \pm 10\%$
- 1.3.2 空载电流: 0.5 Arms
- 1.3.3 额定负载: 4 N.m
- 1.3.4 额定负载转速: $240 \text{ rpm} \pm 10\%$
- 1.3.5 额定负载电流(峰值): $6.5 \text{ A} \pm 10\%$
- 1.3.6 峰值负载: 12 N.m
- 1.3.7 峰值电流(峰值): $23 \text{ A} \pm 10\%$
- 1.3.8 绝缘电阻/定子绕组: DC 500VAC, 100M Ohms
- 1.3.9 耐高压/定子与机壳: 600 VAC, 1s, 2mA
- 1.3.10 电机反电势: $0.054 \sim 0.057 \text{ V}_{\text{rms}}/\text{rpm}$
- 1.3.11 线电阻: $0.45 \Omega \pm 10\%$
- 1.3.12 转矩常数: $0.87 \text{ N.m}/\text{Arms}$
- 1.3.13 电机电感: $187 \sim 339 \mu\text{H}$
- 1.3.14 T-N 曲线



- 1.3.15 最大过载曲线
- 测试条件:
环境温度: 25°C
绕阻极限温度: 120°C

转速：24rpm

最大负载曲线



Load	Operating time (s)
12.00	28
11.00	45
10.00	60
9.00	90
8.00	160
7.00	320
6.00	700
5.00	1800
4.50	2500
4.00	rated

测试数据

1.4 机械特性

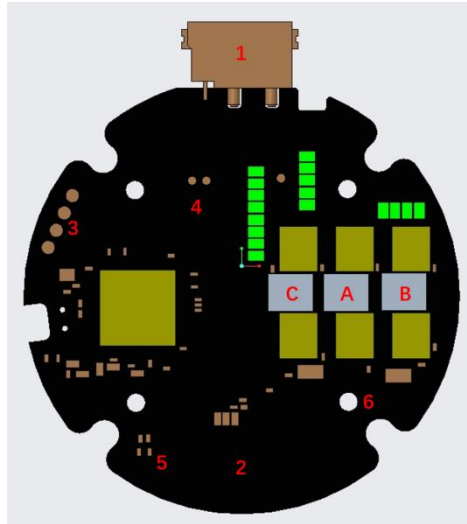
1.4.1 重量：317g±3g

1.4.2 极数：28 极

- 1.4.3 相数：3 相
- 1.4.4 驱动方式：FOC
- 1.4.5 减速比：7.75：1

2 驱动器产品信息

2.1 驱动器外观简介&产品规格

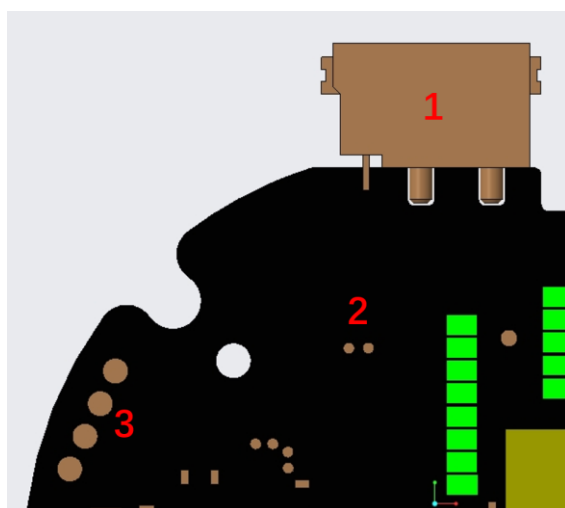


- 1. 24V 电源和 CAN 通信集成端子；
- 2. 硬件版本及镭雕二维码；
- 3. MCU 下载口；
- 4. CAN 通信测试点；
- 5. 指示灯；
- 6. 安装孔；
- 7. “C、A、B” 是三相绕组焊接点；

产品规格	
额定工作电压	24VDC
允许最大电压	28VDC
额定工作电流	6.5A
最大允许电流	23A
待机功耗	≤18mA
CAN 总线比特率	1Mbps
尺寸	Φ58mm
工作环境温度	-20℃至 50℃
控制板允许最大温度	80℃
编码器分辨率	14bit（单圈绝对值）

2.2 驱动器接口定义

2.2.1 驱动器接口图

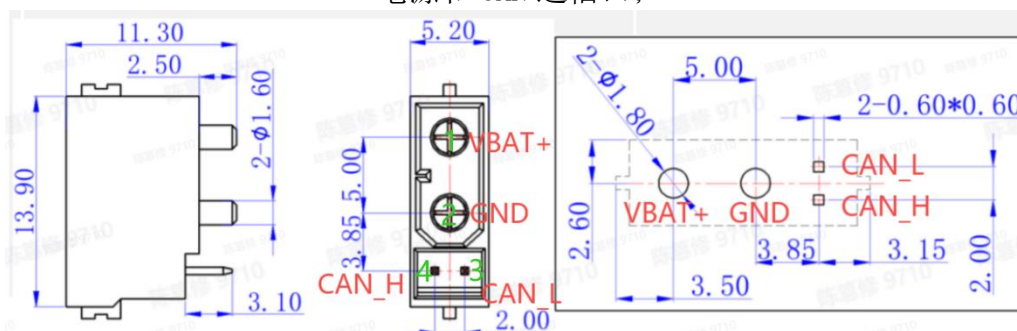


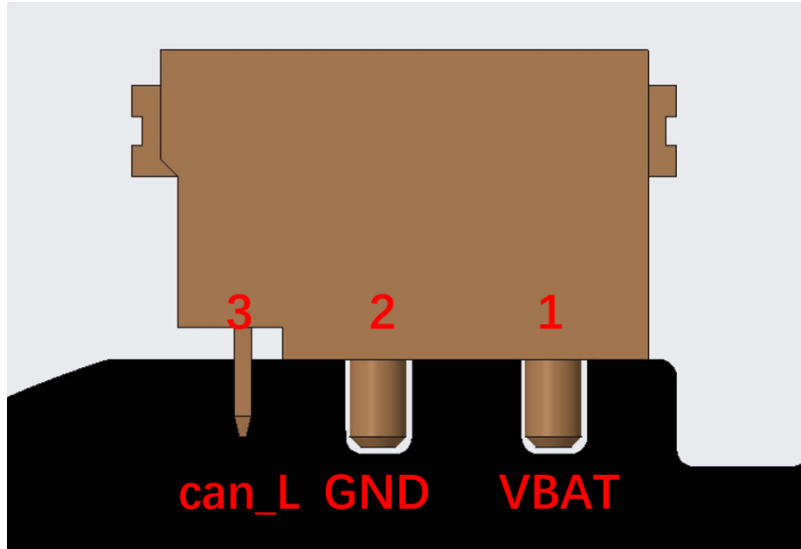
2.2.2 驱动器接口推荐品牌及型号

序号	板端型号	品牌厂家	线端型号	品牌厂家
1	XT30PB(2+2)- M.G.B	AMASS (艾迈斯)	XT30(2+2)- F.G.B	AMASS (艾迈斯)
2	2.0mm-2P 焊盘	/	2.0mm-2P 探针	/
3	2.54mm-4P 焊盘	/	2.54mm-4P 探针	/

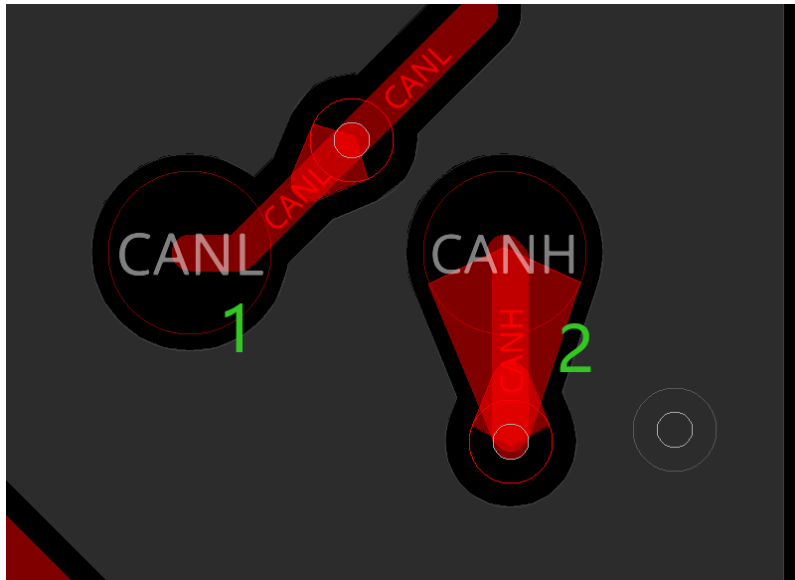
2.2.3 驱动器接口引脚定义

电源和 CAN 通信口；

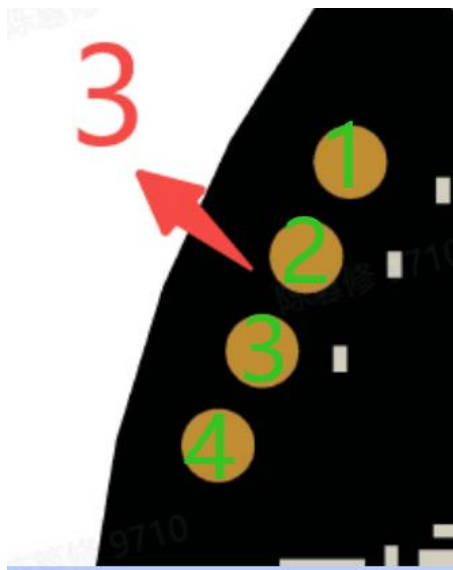




CAN 通信测试焊盘

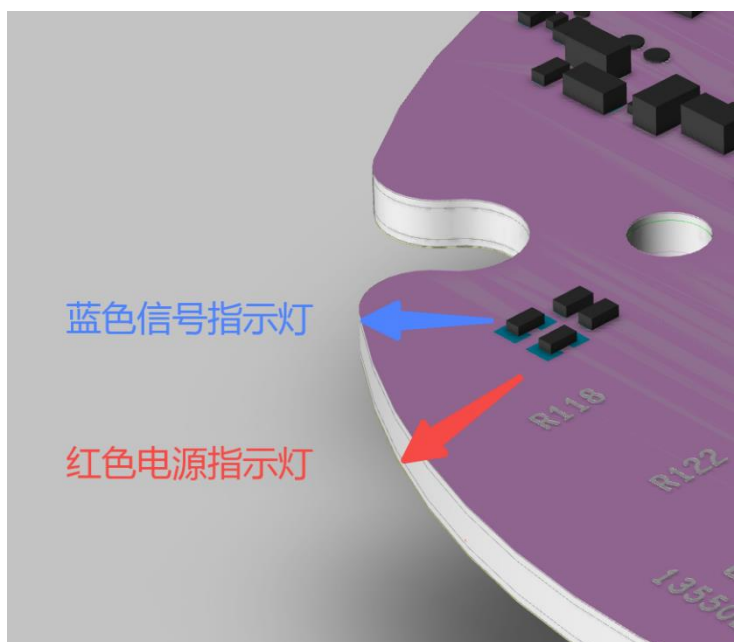


下载口



序号	接口功能	引脚	说明
1	电源及 CAN 通信	1	电源正极 (+)
		2	电源负极 (-)
		3	CAN 通信低侧 CAN_L
		4	CAN 通信高侧 CAN_H
2	CAN 通信测试点	1	CAN 通信低侧 CAN_L
		2	CAN 通信高侧 CAN_H
3	下载口	1	SWDIO (数据)
		2	SWCLK (时钟)
		3	3V3 (正极 3.3V)
		4	GND (负极地)

2.3 驱动器指示灯定义



指示灯定义	
电源指示灯 (亮时红灯)	电源指示灯，用于指示 MCU 3.3V 电源情况，总电源输入 24V 时，该灯亮红色，则证明整个网络供电正常；如果 24V 输入电源时，该指示灯不亮需要立刻断掉电源；
信号指示灯 (亮时蓝灯)	当信号灯闪烁时，证明 MCU 运行正常；并且驱动芯片运行正常；

2.4 主要器件及规格

序号	项目	规格	数量
1	MCU 芯片	GD32F303RET6	1 PCS
2	驱动芯片	6EDL7141	1 PCS
3	磁编码器芯片	AS5047P	1 PCS
4	热敏电阻	NXFT15XH103FEAB021/NCP18XH103F03RB	2 PCS
5	功率 MOS	JMGG031V06A	6 PCS

3 调试器使用说明(扫描纸质说明书末尾二维码获取调试器)

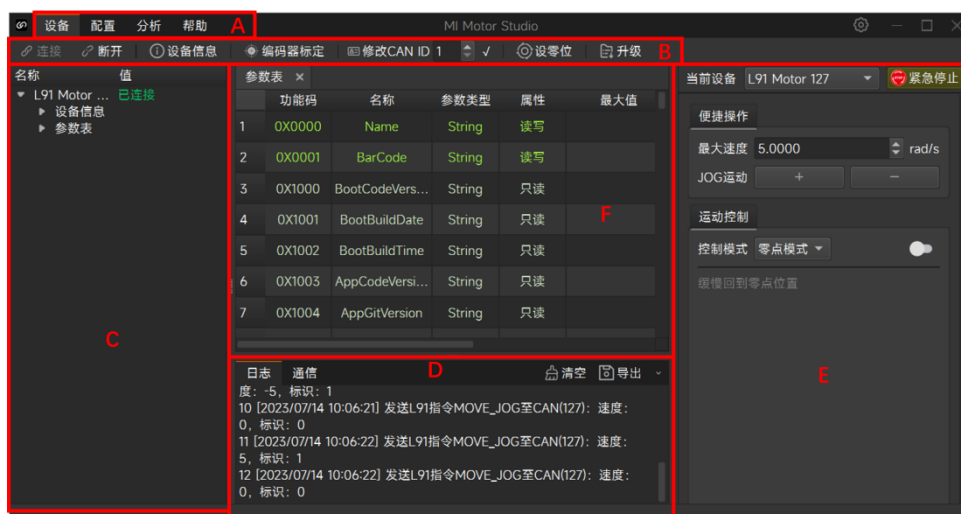
3.1 硬件配置

关节电机采用 CAN 通信方式，通信线有两根，通过 can 转 USB 工具与调试器相连，调试器需要提前安装 ch340 驱动，默认工作在 AT 模式。

需要注意的是，我们是根据特定的 can 转 USB 工具开发的调试器，因此需要用我们推荐的串口工具来进行调试器调试，如果想要移植到其他调试器平台可以参照说明书的第三章进行开发。

can 转 USB 工具推荐使用 YourCee 的 USB-CAN 模块，对应串口协议的帧头为 41 54，帧尾为 0D 0A。

3.2 调试器界面及说明



主要包括:

A. 模块选择

- 设备模块
- 配置模块
- 分析模块
- 帮助模块

B. 子模块选择

设备模块包括

- 连接或断开电机设备
- 电机设备信息
- 电机编码器标定
- 修改电机 CAN ID
- 设置电机的机械零位
- 电机程序升级

配置模块包括：

- 参数表，可以查看并修改电机参数
- 上传参数，可以将电机中参数上传到参数表中
- 下载参数，可以将参数表中数据下载到电机中
- 导出参数，可以将参数表中数据下载到本地
- 恢复出厂，可以将参数表中数据恢复出厂设置
- 清除警告，可以清除电机报错，如温度过高等

分析模块包括：

- 示波器，可以查看参数随时间变化曲线
- 频率，可以调整查看数据的频率
- 信道，可以配置查看的数据
- 开始、停止绘图
- 输出波形数据到本地

帮助模块包括：

- 使用说明，可以打开使用说明书
- 关于，可以查看软件信息

C. 电机信息查询

- 设备信息
- 参数表信息

D. 数据栏

- 日志信息
- 通信信息

E. 运行调试区

- 选择设备
- 便捷操作区，可以快速控制电机正反转
- 运动控制区，可以控制电机按各模式运行

F. 子模块显示区

3.3 电机设置

3.3.1 电机连接设置



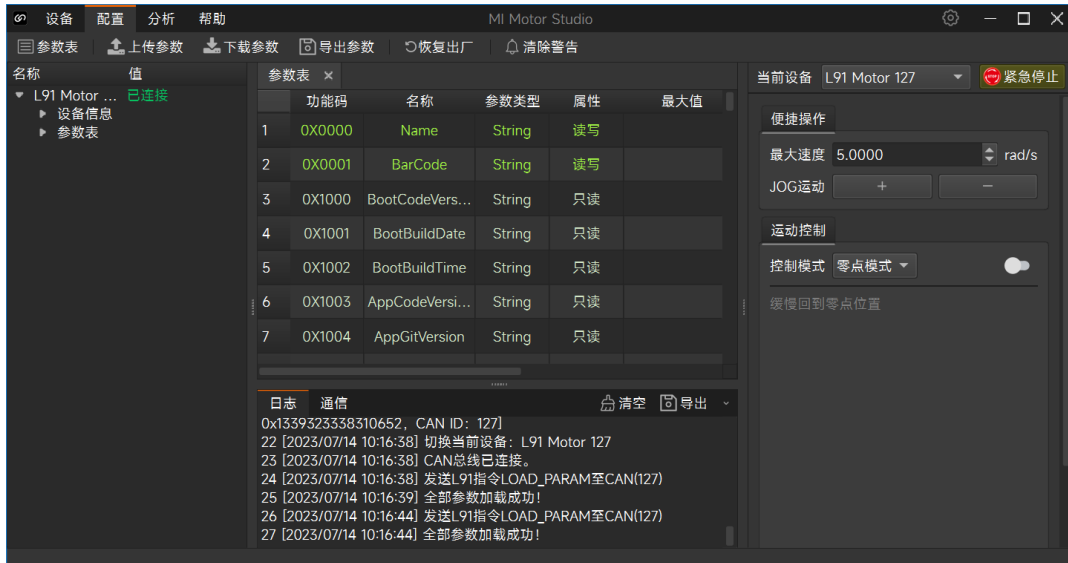
连接 can 转 USB 工具（安装 ch340 驱动，默认工作在 AT 模式），选择设备模块，单击连接子模块，选择对应串口连接。

3.3.2 基本设置



- (1) 修改电机 id 号。
- (2) 电机磁编标定，电机板与电机重新安装，或电机线重新换顺序连接等，需要重新进行磁编标定。
- (3) 设置零位（掉电丢失），设置当前位置为 0。
- (4) 电机程序升级，当电机程序有更新时，点击升级按钮选中升级文件即可进行升级。

3.3.3 参数表



成功连接电机后，点击配置模块中的参数表模块，日志中会显示全部参数加载成功，说明成功读取到电机相关参数（注：参数表需要在电机处于待机状态下进行配置，如果电机处于运行状态则无法进行参数表刷新）界面会显示电机的相关参数，蓝色的参数为电机内部的存储参数，可以在相应参数后面的当前值栏进行修改，点击下载参数可以将调试器中参数下载到电机中，点击上传参数可以将电机中的参数上传到调试器中，电机恢复绿色参数为观测参数，为采集得到的参数，可进行实时观测。

注：电机的转矩限制、保护温度、过温时间请勿随意更改。因违规操作本产品导致对人体造成伤害，或对关节造成不可逆的损伤，我司将不承担任何法律责任。

参数表							
功能码	名称	参数类型	属性	最大值	最小值	当前值（供参考）	备注
0X0000	Name	String	读/写			yyyyyyyyyyyyyyyyyy	
0X0001	BarCode	String	读/写			yyyyyyyyyyyyyyyyyy	
0X1000	BootCodeVersion	String	只读			0.1.5	
0X1001	BootBuildDate	String	只读			Mar 16 2022	
0X1002	BootBuildTime	String	只读			20:22:09	
0X1003	AppCodeVersion	String	只读			0.1.5	电机程序版本号
0X1004	AppGitVersion	String	只读			7b844b0fM	
0X1005	AppBuildDate	String	只读			Apr 14 2022	
0X1006	AppBuildTime	String	只读			20:30:22	
0X1007	AppCodeName	String	只读			dog_motor	
0X2000	echoPara1	uint16	配置	74	5	5	
0X2001	echoPara2	uint16	配置	74	5	5	

0X2002	echoPara3	uint16	配置	74	5	5	
0X2003	echoPara4	uint16	配置	74	5	5	
0X2004	echoFreHz	uint32	读/写	10000	1	500	
0X2005	MechOffset	float	设定	7	-7	4.619583	电机磁编码器角度偏置
0X2006	MechPos_init	float	读/写	50	-50	4.52	初始多圈时的参考角度
0X2007	limit_torque	float	读/写	12	0	12	转矩限制
0X2008	I_FW_MAX	float	读/写	33	0	0	弱磁电流值, 默认 0
0X2009	motor_index	uint8	设定	20	0	1	电机 index, 标记电机关节位置
0X200a	CAN_ID	uint8	设定	127	0	1	本节点 id
0X200b	CAN_MASTER	uint8	设定	127	0	0	can 主机 id
0X200c	CAN_TIMEOUT	uint32	读/写	100000	0	0	can 超时阈值, 默认 0
0X200d	motorOverTemp	int16	读/写	1500	0	800	电机保护温度值, temp (度) *10
0X200e	overTempTime	uint32	读/写	1000000	1000	20000	过温时间
0X200f	GearRatio	float	读/写	64	1	7.75	传动比
0X2010	Tq_caliType	uint8	读/写	1	0	1	转矩标定方法设定
0X2011	cur_filt_gain	float	读/写	1	0	0.9	电流滤波参数
0X2012	cur_kp	float	读/写	200	0	0.025	电流 kp
0X2013	cur_ki	float	读/写	200	0	0.0258	电流 ki
0X2014	spd_kp	float	读/写	200	0	2	速度 kp
0X2015	spd_ki	float	读/写	200	0	0.021	速度 ki
0X2016	loc_kp	float	读/写	200	0	30	位置 kp
0X2017	spd_filt_gain	float	读/写	1	0	0.1	速度滤波参数
0X2018	limit_spd	float	读/写	200	0	2	位置模式速度限制
0X2019	limit_cur	float	读/写	23	0	23	位置、速度模式电流限制

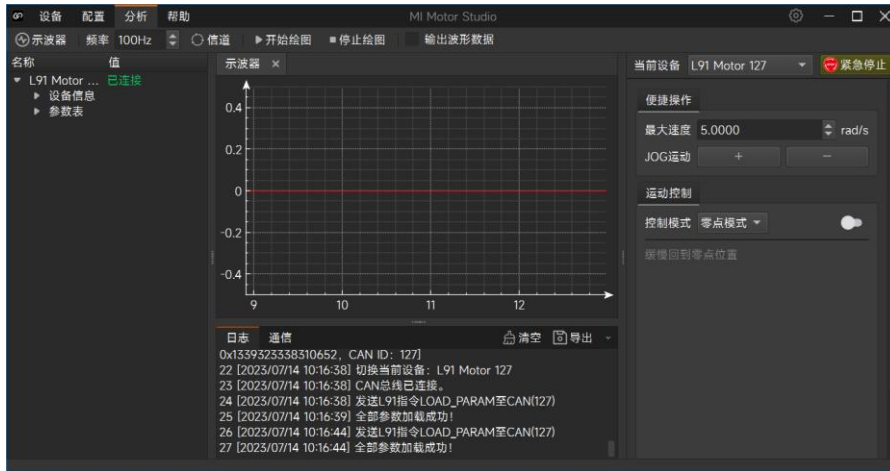
0X3000	timeUse0	uint16	只读			5	
0X3001	timeUse1	uint16	只读			0	
0X3002	timeUse2	uint16	只读			10	
0X3003	timeUse3	uint16	只读			0	
0X3004	encoderRaw	int16	只读			11396	磁编码器采样值
0X3005	mcuTemp	int16	只读			337	mcu 内部温度, *10
0X3006	motorTemp	int16	只读			333	电机 ntc 温度, *10
0X3007	vBus (mv)	uint16	只读			24195	母线电压
0X3008	adc1Offset	int32	只读			2084	adc 采样通道 1 零电流偏置
0X3009	adc2Offset	int32	只读			2084	adc 采样通道 2 零电流偏置
0X300a	adc1Raw	uint16	只读			1232	adc 采样值 1
0X300b	adc2Raw	uint16	只读			1212	adc 采样值 2
0X300c	VBUS	float	只读			24.195	母线电压 V
0X300d	cmdId	float	只读			0	id 环指令, A
0X300e	cmdIq	float	只读			0	iq 环指令, A
0X300f	cmdIcref	float	只读			0	位置环指令, rad
0X3010	cmdspdref	float	只读			0	速度环指令, rad/s
0X3011	cmdTorque	float	只读			0	转矩指令, nm
0X3012	cmdPos	float	只读			0	mit 协议角度指 令
0X3013	cmdVel	float	只读			0	mit 协议速度指 令
0X3014	rotation	int16	只读			1	圈数
0X3015	modPos	float	只读			4.363409	电机未计圈机械 角度, rad
0X3016	mechPos	float	只读			0.777679	负载端计圈机械 角度, rad
0X3017	mechVel	float	只读			0.036618	负载端转 速, rad/s
0X3018	elecPos	float	只读			4.714761	电气角度
0X3019	ia	float	只读			0	U 线电流, A

0X301a	ib	float	只读			0	V 线电流, A
0X301b	ic	float	只读			0	W 线电流, A
0X301c	tick	uint32	只读			31600	
0X301d	phaseOrder	uint8	只读			0	标定方向标记
0X301e	iqf	float	只读			0	iq 滤波值, A
0X301f	boardTemp	int16	只读			359	板上温度, *10
0X3020	iq	float	只读			0	iq 原值, A
0X3021	id	float	只读			0	id 原值, A
0X3022	faultSta	uint32	只读			0	故障状态值
0X3023	warnSta	uint32	只读			0	警告状态值
0X3024	drv_fault	uint16	只读			0	驱动芯片故障值
0X3025	drv_temp	int16	只读			48	驱动芯片温度值, 度
0X3026	Uq	float	只读			0	q 轴电压
0X3027	Ud	float	只读			0	d 轴电压
0X3028	dtc_u	float	只读			0	U 相输出占空比
0X3029	dtc_v	float	只读			0	V 相输出占空比
0X302a	dtc_w	float	只读			0	W 相输出占空比
0X302b	v_bus	float	只读			24.195	闭环中 vbus
0X302c	v_ref	float	只读			0	闭环 vq, vd 合成电压
0X302d	torque_fdb	float	只读			0	转矩反馈值, nm
0X302e	rated_i	float	只读			8	电机额定电流
0X302f	limit_i	float	只读			27	电机限制最大电流

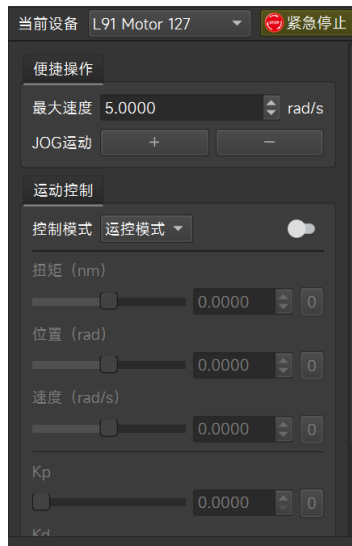
3.3.4 示波器

该界面支持观看观察实时数据所生成的图谱, 可观测的数据包括电机 Id/Iq 电流、温度、输出端实时转速、转子 (编码器) 位置、输出端位置等。

点击分析模块中的示波器模块, 信道内选定合适的参数 (参数含义可参考 3.3.3), 设置输出频率后点击开始绘图即可观测数据图谱, 停止绘图即可停止观测图谱。

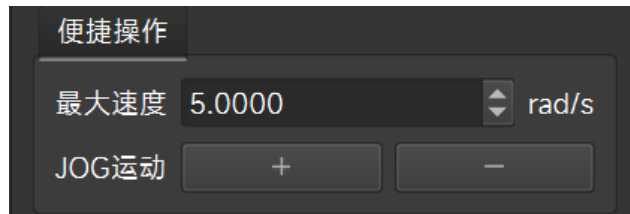


3.4 控制演示



jog 运行:

设置最大速度，点击运行后，点击 JOG 运行即可让电机正反运行



控制模式切换:

在运动模式界面可以进行电机控制模式的转换



3.4.1 零点模式



点击右侧开关按钮，电机会缓慢回到机械零位位置

3.4.2 运控模式



点击右侧开关按钮，然后设置五个参数值，点击开始或连续发送，电机将返回反馈帧并按目标指令运行；再次点击右侧开关按钮，电机将停机。

3.4.2 电流模式



手动切换电流模式，点击右侧开关按钮，然后设置 I_q 电流指令值，开始或连续发送，电机将跟随电流指令运行，再次点击右侧开关按钮，电机将停机。

点击控制模式右侧开关按钮，输入正弦化自动测试的幅值和频率，然后点击正弦化自动测试右侧开关按钮，电机的 i_q (A) 会按设定的幅值和频率来运行。

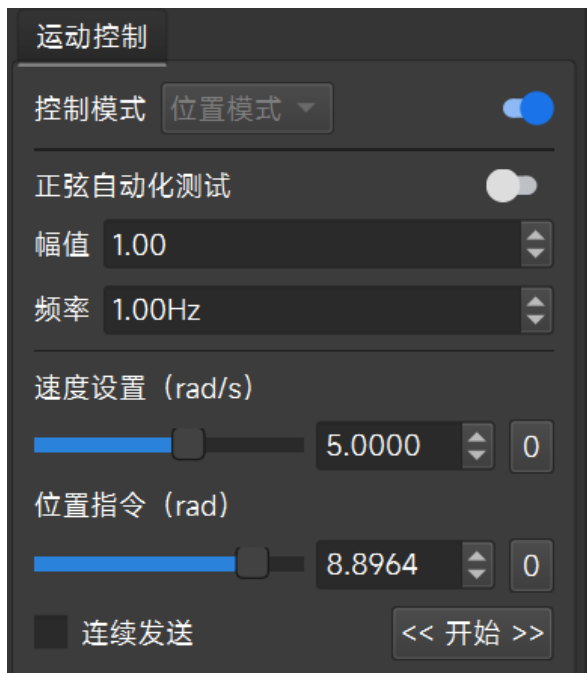
3.4.3 速度模式



手动切换速度模式，点击右侧开关按钮，然后设置速度指令值（ $-30 \sim 30 \text{rad/s}$ ），开始或连续发送，电机将跟随速度指令运行，再次点击右侧开关按钮，电机将停机。

点击控制模式右侧开关按钮，输入正弦化自动测试的幅值和频率，然后点击正弦化自动测试右侧开关按钮，电机的速度（rad/s）会按设定的幅值和频率来运行。

3.4.4 位置模式



手动切换位置模式，点击右侧开关按钮，然后设置位置指令值（rad），开始或连续发送，电机将跟随目标位置指令运行，再次点击右侧开关按钮，电机将停机。可通过设置速度，修改位置跟随的最大速度。

点击控制模式右侧开关按钮，输入正弦化自动测试的幅值和频率，然后点击正弦化自动测试右侧开关按钮，电机的位置（rad）会按设定的幅值和频率来运行。

3.5 固件更新



第一步，点击设备模块的升级，选择待烧录 bin 文件；第二步，确认升级，电机开始更新固件，进度完成后，电机更新完成，自动重启。

4 驱动器通信协议及使用说明

电机通信为 CAN 2.0 通信接口，波特率 1Mbps，采用扩展帧格式，如下所示：

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	通信类型	数据区 2	目标地址	数据区 1

电机支持的控制模式包括：

运控模式：给定电机运控 5 个参数；

电流模式：给定电机指定的 Iq 电流；

速度模式：给定电机指定的运行速度；

位置模式：给定电机指定的位置，电机将运行到该指定的位置；

4.1 通信协议类型说明

4.1.1 获取设备 ID（通信类型 0）；获取设备的 ID 和 64 位 MCU 唯一标识符

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	0	bit15~8: 用来标识主机 CAN_ID	目标电机 CAN_ID	0

应答帧:

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	0	目标电机 CAN_ID	OXFE	64 位 MCU 唯一标识符

4.1.2 运控模式电机控制指令（通信类型 1）用来向电机发送控制指令

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	1	Byte2: 力矩 (0~65535) 对应 (-12Nm~12Nm)	目标电机 CAN_ID	Byte0~1: 目标角度 [0~65535] 对应 (-4π~4π) Byte2~3: 目标角速度 [0~65535] 对应 (-30rad/s~30rad/s) Byte4~5: Kp [0~65535] 对应 (0.0~500.0) Byte6~7: Kd [0~65535] 对应 (0.0~5.0)

应答帧: 应答电机反馈帧(见通信类型 2)

4.1.3 电机反馈数据（通信类型 2）用来向主机反馈电机运行状态

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	2	Bit8~Bit15: 当前电机 CAN ID bit21~16: 故障信息 (0 无 1 有) bit21: 未标定 bit20: HALL 编码故障 bit19: 磁编码故障 bit18: 过温 bit17: 过流 bit16: 欠压故障 bit22~23: 模式状态 0 : Reset 模式 [复位] 1 : Cali 模式 [标定] 2 : Motor 模式 [运行]	主机 CAN_ID	Byte0~1: 当前角度 [0~65535] 对应 $(-4\pi \sim 4\pi)$ Byte2~3: 当前角速度 [0~65535] 对应 $(-30\text{rad/s} \sim 30\text{rad/s})$ Byte4~5: 当前力矩 [0~65535] 对应 $(-12\text{Nm} \sim 12\text{Nm})$ Byte6~7: 当前温度: Temp(摄氏度) *10

4.1.4 电机使能运行 (通信类型 3)

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	3	bit15~8: 用来标识主 CAN_ID	目标电机 CAN_ID	

应答帧: 应答电机反馈帧(见通信类型 2)

4.1.5 电机停止运行 (通信类型 4)

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7

描述	4	bit15~8:用来标识主 CAN_ID	目标电机 CAN_ID	正常运行时，data 区需清 0; Byte[0]=1 时：清故障；
----	---	----------------------	-------------	---------------------------------------

应答帧：应答电机反馈帧(见通信类型 2)

4.1.6 设置电机机械零位（通信类型 6）会把当前电机位置设为机械零位（掉电丢失）

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	6	bit15~8:用来标识主 CAN_ID	目标电机 CAN_ID	Byte[0]=1

应答帧：应答电机反馈帧(见通信类型 2)

4.1.7 设置电机 CAN_ID（通信类型 7）更改当前电机 CAN_ID ， 立即生效。

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	7	bit15~8:用来标识主 CAN_ID Bit16~23: 预设置 CAN_ID	目标电机 CAN_ID	

应答帧：应答电机广播帧(见通信类型 0)

4.1.8 单个参数读取（通信类型 17）

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	17	bit15~8:用来标识主 CAN_ID	目标电机 CAN_ID	Byte0~1: index, 参数列表详见 4.1.11 Byte2~3: 00 Byte4~7: 00

应答帧:

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	17	bit15~8:目标电机 CAN_ID	主机 CAN_ID	Byte0~1: index, 参数列表详见 4.1.11 Byte2~3: 00 Byte4~7: 参数数据, 1 字节数据在 Byte4

4.1.9 单个参数写入（通信类型 18）（掉电丢失）

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	18	bit15~8:用来标识主 CAN_ID	目标电机 CAN_ID	Byte0~1: index, 参数列表详见 4.1.11 Byte2~3: 00 Byte4~7: 参数数据

应答帧: 应答电机反馈帧(见通信类型 2)

4.1.10 故障反馈帧（通信类型 21）

数据域	29 位 ID			8Byte 数据区
大小	Bit28~bit24	bit23~8	bit7~0	Byte0~Byte7
描述	21	bit15~8:用来标识 主 CAN_ID	电机 CAN_ID	Byte0~3: fault 值(非 0:有故障, 0: 正常) bit16:A 相电流采样过流 bit15~bit8:过载故障 bit7:编码器未标定 bit5:C 相电流采样过流 bit4:B 相电流采样过流 bit3:过压故障 bit2:欠压故障 bit1:驱动芯片故障 bit0:电机过温故障, 默认 80 度 Byte4~7: warning 值 bit0: 电机过温预警, 默认 75 度

4.1.11 可读写单个参数列表（7019-701D 为最新版本固件可读）

参数 index	参数名称	描述	类型	字节数	单位/说明	R/W 读写权限
0X7005	run_mode	0: 运控模式 1: 位置模式 2: 速度模式 3: 电流模式	uint8	1		W/R
0X7006	iq_ref	电流模式 Iq 指令	float	4	-23~23A	W/R
0X700A	spd_ref	转速模式转速指令	float	4	-30~30rad/s	W/R
0X700B	imit_torque	转矩限制	float	4	0~12Nm	W/R
0X7010	cur_kp	电流的 Kp	float	4	默认值 0.125	W/R
0X7011	cur_ki	电流的 Ki	float	4	默认值 0.0158	W/R
0X7014	cur_filt_gain	电流滤波系数 filt_gain	float	4	0~1.0, 默认值 0.1	W/R
0X7016	loc_ref	位置模式角度指令	float	4	rad	W/R
0X7017	limit_spd	位置模式速度限制	float	4	0~30rad/s	W/R
0X7018	limit_cur	速度位置模式 电流限制	float	4	0~23A	W/R
0x7019	mechPos	负载端计圈机械角度	float	4	rad	R
0x701A	iqf	iq 滤波值	float	4	-23~23A	R
0x701B	mechVel	负载端转速	float	4	-30~30rad/s	R
0x701C	VBUS	母线电压	float	4	V	R
0x701D	rotation	圈数	int16	2	圈数	W/R

4.2 控制模式使用说明

4.2.1 程序样例

以下提供各种模式控制电机实例（以 gd32f303 为例）

下面为各种实例调用库，函数与宏定义

```
#define P_MIN -12.5f
#define P_MAX 12.5f
#define V_MIN -30.0f
#define V_MAX 30.0f
#define KP_MIN 0.0f
#define KP_MAX 500.0f
#define KD_MIN 0.0f
#define KD_MAX 5.0f
#define T_MIN -12.0f
#define T_MAX 12.0f
struct exCanIdInfo{
uint32_t id:8;
uint32_t data:16;
uint32_t mode:5;
uint32_t res:3;
};
can_receive_message_struct rxMsg;
can_transmit_message_struct txMsg={
    .tx_sfid = 0,
    .tx_efid = 0xff,
    .tx_ft = CAN_FT_DATA,
    .tx_ff = CAN_FF_EXTENDED,
    .tx_dlen = 8,
};
#define txCanIdEx (((struct exCanIdInfo)&(txMsg.tx_efid)))
#define rxCanIdEx (((struct exCanIdInfo)&(rxMsg.rx_efid))) //将扩展帧
id 解析为自定义数据结构
int float_to_uint(float x, float x_min, float x_max, int bits){
    float span = x_max - x_min;
    float offset = x_min;
    if(x > x_max) x=x_max;
    else if(x < x_min) x= x_min;
    return (int) ((x-offset)*((float)((1<<bits)-1))/span);
}
#define can_txd() can_message_transmit(CAN0, &txMsg)
#define can_rxd() can_message_receive(CAN0, CAN_FIFO1, &rxMsg)
```

下面列举常见的通信类型发送:

1、电机使能运行帧（通信类型 3）

```
void motor_enable(uint8_t id, uint16_t master_id)
{
    txCanIdEx.mode = 3;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    txCanIdEx.data = 0;
    can_txd();
}
```

2、运控模式电机控制指令（通信类型 1）

```
void motor_controlmode(uint8_t id, float torque, float
MechPosition, float speed, float kp, float kd)
{
    txCanIdEx.mode = 1;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = float_to_uint(torque, T_MIN, T_MAX, 16);
    txMsg.tx_dlen = 8;
    txMsg.tx_data[0]=float_to_uint(MechPosition, P_MIN, P_MAX, 16)>>8;
    txMsg.tx_data[1]=float_to_uint(MechPosition, P_MIN, P_MAX, 16);
    txMsg.tx_data[2]=float_to_uint(speed, V_MIN, V_MAX, 16)>>8;
    txMsg.tx_data[3]=float_to_uint(speed, V_MIN, V_MAX, 16);
    txMsg.tx_data[4]=float_to_uint(kp, KP_MIN, KP_MAX, 16)>>8;
    txMsg.tx_data[5]=float_to_uint(kp, KP_MIN, KP_MAX, 16);
    txMsg.tx_data[6]=float_to_uint(kd, KD_MIN, KD_MAX, 16)>>8;
    txMsg.tx_data[7]=float_to_uint(kd, KD_MIN, KD_MAX, 16);
    can_txd();
}
```

3、电机停止运行帧（通信类型 4）

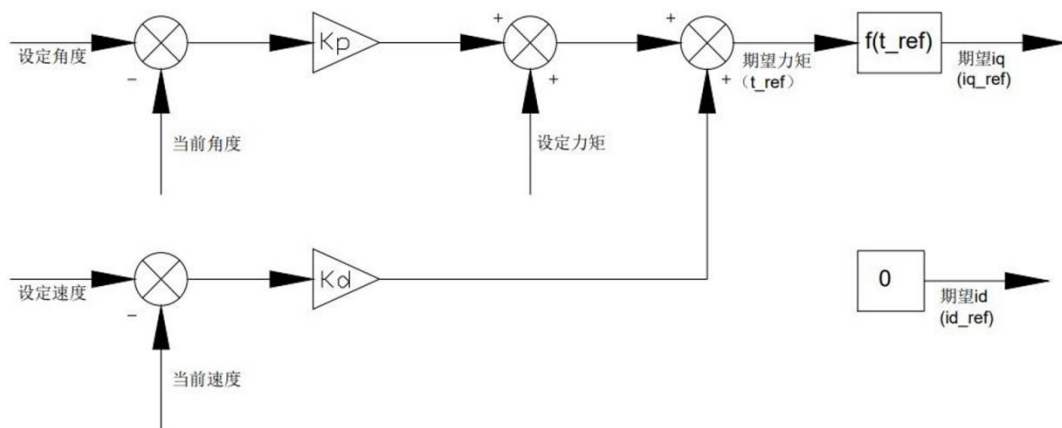
```
void motor_reset(uint8_t id, uint16_t master_id)
{
    txCanIdEx.mode = 4;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    for(uint8_t i=0;i<8;i++)
    {
        txMsg.tx_data[i]=0;
    }
    can_txd();
}
```

```

}
4、电机模式参数写入命令（通信类型 18，运行模式切换）
uint8_t runmode;
uint16_t index;
void motor_modechange(uint8_t id, uint16_t master_id)
{
    txCanIdEx.mode = 0x12;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    for(uint8_t i=0;i<8;i++)
    {
        txMsg.tx_data[i]=0;
    }
    memcpy(&txMsg.tx_data[0],&index,2);
    memcpy(&txMsg.tx_data[4],&runmode,1);
    can_txd();
}
5、电机模式参数写入命令（通信类型 18，控制参数写入）
uint16_t index;
float ref;
void motor_write(uint8_t id, uint16_t master_id)
{
    txCanIdEx.mode = 0x12;
    txCanIdEx.id = id;
    txCanIdEx.res = 0;
    txCanIdEx.data = master_id;
    txMsg.tx_dlen = 8;
    for(uint8_t i=0;i<8;i++)
    {
        txMsg.tx_data[i]=0;
    }
    memcpy(&txMsg.tx_data[0],&index,2);
    memcpy(&txMsg.tx_data[4],&ref,4);
    can_txd();
}

```

4.2.2 运控模式



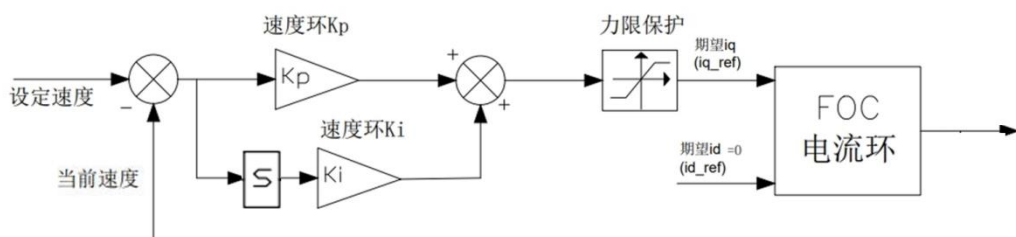
电机上电后默认处于运控模式；

发送电机使能运行帧（通信类型 3）-->发送运控模式电机控制指令（通信类型 1）-->收到电机反馈帧（通信类型 2）

4.2.3 电流模式

发送电机模式参数写入命令（通信类型 18）设置 runmode 参数为 3 ---> 发送电机使能运行帧（通信类型 3）--> 发送电机模式参数写入命令（通信类型 18）设置 iq_ref 参数为预设电流指令

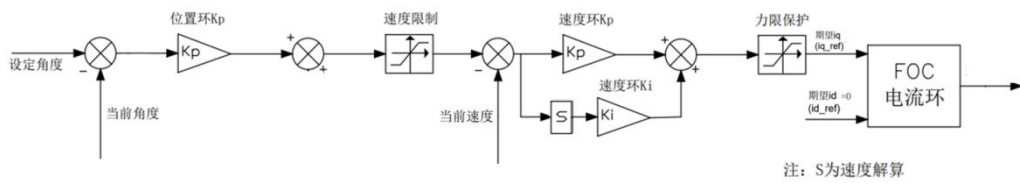
4.2.4 速度模式



注：S为速度解算

发送电机模式参数写入命令（通信类型 18）设置 runmode 参数为 2 ---> 发送电机使能运行帧（通信类型 3）--> 发送电机模式参数写入命令（通信类型 18）设置 limit_cur 参数为预设最大电流指令-->发送电机模式参数写入命令（通信类型 18）设置 spd_ref 参数为预设速度指令

4.2.5 位置模式



发送电机模式参数写入命令（通信类型 18）设置 runmode 参数为 1 --> 发送电机使能运行帧（通信类型 3）--> 发送电机模式参数写入命令（通信类型 18）设置 limit_spd 参数为预设最大速度指令-->发送电机模式参数写入命令（通信类型 18）设置 loc_ref 参数为预设位置指令

4.2.6 停止运行

发送电机停止运行帧（通信类型 4）