

MD403-x42 步进伺服专用驱动电调

用户手册

UM01010101 1.0.00 Date:2025/6/30

类别	内容
关键词	用户手册
摘要	

MD403-x42 步进伺服专用驱动电调

用户手册

User Manual

修订历史

版本	日期	原因
V1.0.00	2025/05/18	创建文档

目 录

1. 产品简介.....	1
1.1 产品概述.....	1
1.2 主要特点.....	1
1.3 产品参数.....	2
1.4 模式介绍.....	2
1.5 电机线序自识别说明.....	3
1.6 多机通信及同步控制.....	3
2. 模块接线说明.....	4
2.1 接口说明.....	4
2.2 电源输入要求.....	4
2.3 输入输出信号.....	5
2.4 报警信号输出.....	5
2.5 脉冲控制方式接线.....	5
2.6 串行通信接线方式.....	6
3. 安装说明.....	8
3.1 安装使用.....	8
3.2 磁编校准说明.....	10
4. 脉冲控制.....	11
4.1 脉冲信号要求.....	11
4.2 方向控制信号要求.....	11
5. 串行通信控制.....	13
5.1 通讯命令帧格式.....	13
5.1.1 子节点地址.....	13
5.1.2 功能代码.....	13
5.1.3 数据.....	13
5.1.4 CRC 校验.....	13
5.2 数据模型.....	14
5.3 功能代码定义.....	14
5.4 CRC 校验.....	15
5.5 功能码使用详解.....	15
5.5.1 04(0x04) 读输入寄存器.....	15
5.5.2 03(0x03) 读保持寄存器.....	16
5.5.3 06(0x06) 写单个寄存器.....	17
5.5.4 16(0x10) 写多个寄存器.....	17
5.6 电调模块 MODBUS 通讯定义.....	19
5.6.1 从机地址.....	19
5.6.2 寄存器定义.....	19
5.7 参数更新流程.....	20
5.7.1 参数读取示例.....	22
5.7.2 参数写入示例.....	22
5.8 电调常用通讯示例.....	22

- 5.8.1 速度控制.....22
 - 5.8.2 位置控制.....22
 - 5.8.3 控制模式选择.....22
 - 5.8.4 校准操作.....23
 - 5.8.5 多机同步控制.....23
- 6. 电机适配流程.....25
 - 6.1 修改电调参数.....25
 - 6.2 电机调试.....25
 - 6.3 固件升级.....26
- 7. 免责声明.....29

1. 产品简介

1.1 产品概述

MD403-x42 电调驱动模块是由立功科技.求远电子科技有限公司针对步进电机控制而设计的一款 FOC 步进 42 电机驱动器，驱动模块结构满足 42 电机安装结构要求，针对中低功率步进电机（支持开环、闭环、FOC 伺服）的高性能产品，适用于 15~30V 的步进电机，具有完善的保护机制和功能接口，满足各种应用场景。

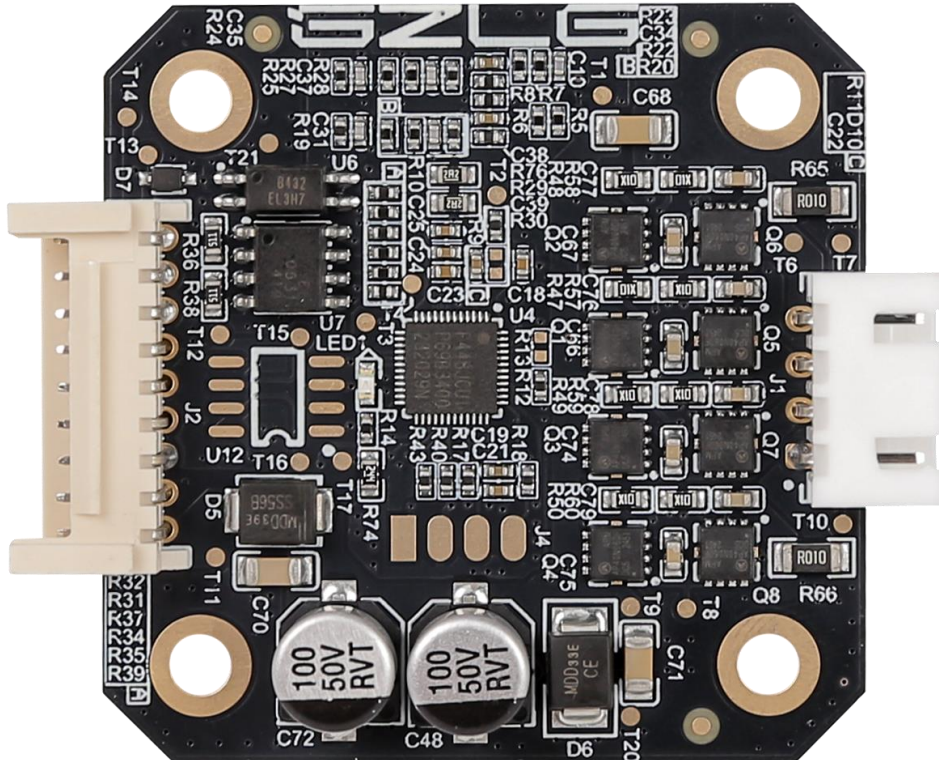


图 1.1 MD403-x42 模块正视图

1.2 主要特点

- 支持开环、闭环、FOC 伺服三种控制模式；
- 支持脉冲控制，串口速度，位置等多种指令控制；
- 速度环、位置环、电流环，三环闭环控制；
- 采用 16384 线磁编码器；
- 支持设置 1~256 任意细分参数；
- 支持编码器校准，支持多次校准；
- FOC 运行模式下，高转速范围、超低噪声、超低振动、超低发热；
- 最高转速达 2000RPM 以上，不同参数电机最高上限不同；
- 支持 T 型和 S 型加减速，启停更平稳；
- 全面的错误报警机制，支持过压、欠压、过流等异常报警；

- 尺寸：42mm * 42mm * 11mm
- 提供上位机，支持用户自适配电机参数，可适配各种 42 步进电机。

1.3 产品参数

驱动板型号	MD403-x42	MD403-S42
主控芯片	高性能 M4 处理器	
MOSFET	8 个高性能 MOS 双 H 桥驱动	
编码器	14bit 工业级高精度磁编码器	
控制方式	脉冲+TTL	脉冲+RS485
工作电压	15V~30V（推荐 24V）	
工作电流	0~3000mA	
信号输入 EN/DIR/STP	3.3V~24V 支持共阳/共阴输入	
脉冲频率范围	可达 200KHZ 以上	
细分支持	1~256 细分设置	
从机地址	1~247（用户可用）	
控制位置精度	小于 0.08°	
最高转速	可达 3000RPM+，不同电机上限不同	
控制方式	脉冲控制、串行指令控制	
闭环反馈频率	电流环 20KHZ、速度环 10KHZ、位置环 10KHZ	
参数可修改性	开放电机参数适配上位机，可自行适配电机参数	
工作模式	开环、闭环、FOC	
震动/噪声	FOC 模式下超低噪声、超低震动	
电机温度	FOC 模式下，电机基本无发热	
保护措施	过压、过流、欠压、堵转等保护措施	
其他功能	编码器自动校准、线序识别	

1.4 模式介绍

MD403-x42 控制板，可以通过上位机或串口指令来选择运行模式及控制模式，可以设置为速度模式、位置模式、力矩模式及开环模式如表 1.1。

1. 控制模式选择

控制模式	控制原理	特点	使用场景
开环	通过直接改变电角度控制电机转动	控制简单，不需要编码器就能运行； 电流固定，需调节工作电流，刚性强； 存在电机发热和振动情况；	转速 300RPM 以内
闭环	在开环控制的功能上加上丢步检测功能。	控制相对简单不丢步； 电流固定，需调节工作电流，刚性强； 存在电机发热和振动情况；	适合使用在位置要求高的场景

伺服	使用 FOC 电机控制技术，电流速度位置三环控制对电机进行控制	电流自动调节、噪声小、高转速、基本无发热和振动； 刚性较低；	高转速范围应用场景 低噪声应用场景 低抖动低发热场景
----	---------------------------------	-----------------------------------	----------------------------------

2. 运行模式选择

表 1.1 运行模式选择对照表

运行模式	控制模式	特点	使用场景
步进模式	伺服、闭环、开环	速度、位置由步进脉冲信号控制	3D 打印机、多轴机床等
位置模式	伺服、闭环、开环	位置、速度由串口控制	机器人、云台等
速度模式	伺服、闭环、开环	速度由串口控制	流量控制泵、挤出机等
力矩模式	伺服	力矩由串口控制	机械夹具等

1.5 电机线序自识别说明

MD403-x42 系列支持线序自识别，可避免因 A+/A-线序交叉，或 B+/B-线序交叉，以及 A 组和 B 组线序颠倒导致的方向不统一问题，上电后驱动器会再转子对齐瞬间进行位置校准，采用唯一方向运行，可以避免因为线序错乱导致的方向错误。

1.6 多机通信及同步控制

MD403-x42 支持多机通信和同步控制，每个设备可以独立分配从机地址，支持 1~247 地址分配。一条总线上支持多达 247 个 MD403-x42 作为从机设备。

- 多机控制：可以通过发送广播地址带控制功能码指令直接控制总线上的所有设备；
- 同步控制：可以独立向所需控制的设备发送控制指令，被控设备收到后不立即执行，直到收到同步指令再一起执行；

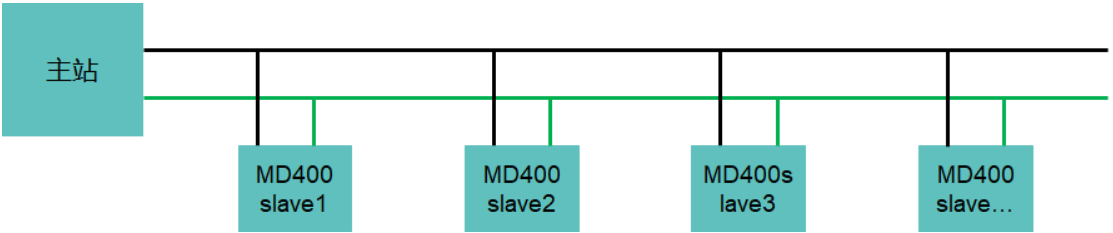


图 1.2 多机通信说明

2. 模块接线说明

MD403-x42 支持完善的输入输出控制信号，可以非常方便的接入所需系统。输入控制信号包含电调使能控制信号、刹车信号、模式选择信号以及调速信号，输出信号主要是异常状态的报警信号，支持上报异常状态。

2.1 接口说明

MD403-x42 步进驱动模块见下图。

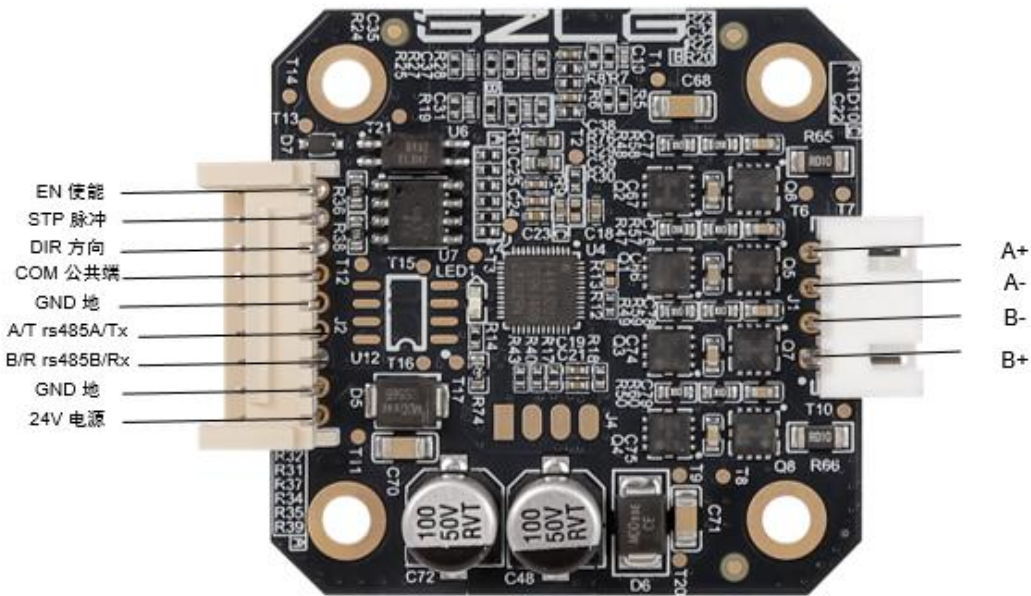


图 2.1 MD403-x42 模块接口说明

表 2.1 MD403-x42 模块接口功能描述

接口名称	功能描述	接口名称	功能描述
A+, A-	接电机一组相线	COM	光耦隔离公共端，输入 3.3V~24V 电平
B+, B-	接电机另一组相线	DIR	脉冲控制模式下的方向控制引脚
A/T, B/R	根据不同硬件版本，接 RS485/TTL	STP	脉冲控制模式下的脉冲输入引脚
24V, GND	电源输入	EN	模块使能端口，使能后控制电机运动信号有效；失能后，校准信号输入有效

2.2 电源输入要求

驱动器推荐采用 24V 电源供电，所使用的供电电源需要确保有一定的续流能力，避免因续流能力不足，导致电机运行时电流不够，拉低电源电压，导致电机运行不正常或报欠

压错误码。需要多大续流的电源，请自行根据负载和工况情况而定。

2.3 输入输出信号

MD403-x42 的控制信号使用差分输入，其他控制器可以通过这些控制引脚输入控制信号控制电机运行，支持的输入控制信号的说明，电调使能、限位、转向功能，具体控制逻辑详见表 2.2。

表 2.2 控制信号输入逻辑对照表

控制口	电平状态	控制功能	备注
使能	0（默认）	电调使能	电调能够驱动电机
	1	电调禁能	电调无法驱动电机
方向	0（默认）	电机正转	电机正转，仅脉冲步进模式下有效
	1	电机反转	电机反转，仅脉冲步进模式下有效

2.4 报警信号输出

MD403-x42 具有报警保护功能，在检测到异常情况时，将会关闭电机并输出报警信号，报警信号输出与板载 LED 相连，可通过 LED 闪烁频率确认，其中除了禁能控制之外，其余报警都需要排除报警原因后再启动。

表 2.3 报警信号对照表

报警代码	报警原因	报警脉冲频率	备注
0	正常工作	1Hz	正常状态
4	软件过流	5 Hz	电流超过用户配置软件过流值
8	过压	6 Hz	电压超过用户配置最大电压
16	欠压	7 Hz	电压小于用户配置最小电压
256	堵转	10 Hz	由于负载造成电机转速过低或电流过大
1024	紧急停机	12 Hz	电机由异常情况而处于紧急停止状态
2048	电流偏置校准错误	13 Hz	电流偏置校准错误
4096	编码器校准错误	14 Hz	编码器校准错误
32768	未使能	2 Hz	外部使能控制处于未使能状态

当出现各种异常报警时，电机将会进入报警保护状态，测试需要根据电机异常情况进行排查，之后再重新使能，控制器将会清除故障码，重新正常工作；如果异常状态未处理，系统重新运行依旧会进入保护状态。

2.5 脉冲控制方式接线

本驱动器支持两种控制方式，一种方式采用脉冲控制，一种方式采用串行端口控制，本节介绍采用脉冲方式控制的实现。可以通过串口指令，将控制方式设置为脉冲控制方式或串行指令方式，如设置使用脉冲方式控制，脉冲输入的频率表征了控制电机运动的速度，脉冲

是数量表征了电机运行位置（配合方向控制引脚），接线如下。

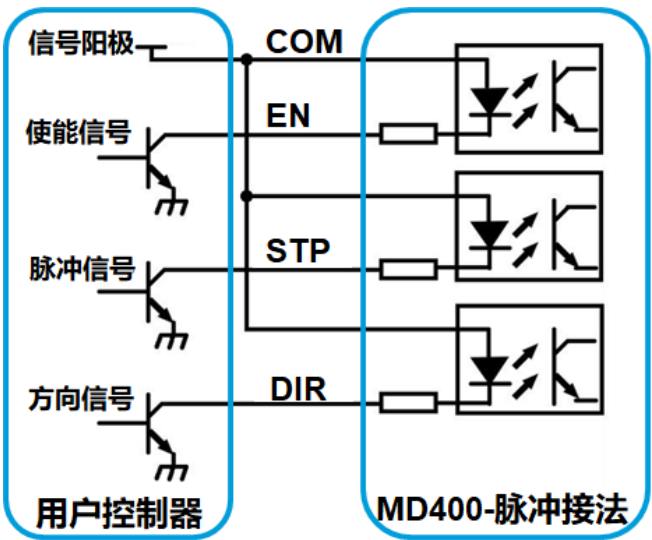


图 2.2 MD403-x42 脉冲控制接法要求

2.6 串行通信接线方式

TTL 硬件版本，串口接线采用如下接线方式。

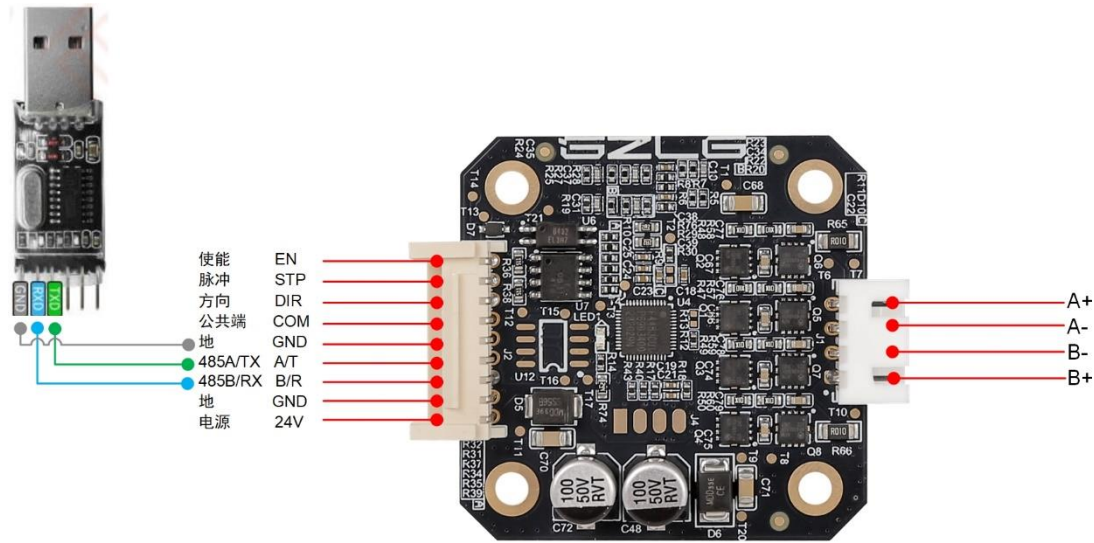


图 2.3 TTL 硬件版本接线

RS485 硬件版本，接线方式如下所示。

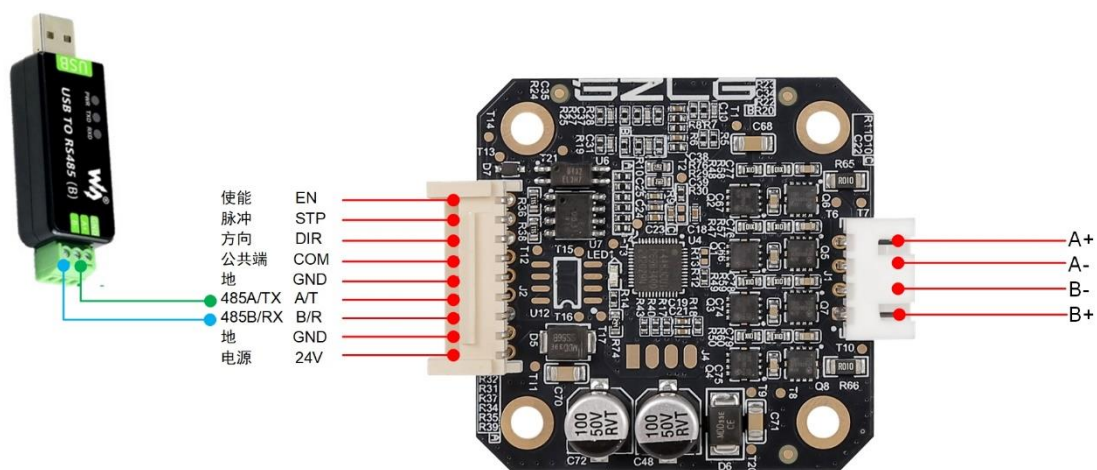


图 2.4 RS485 硬件版本接线

3. 安装说明

MD403-x42 板子中心带有磁编码器，适配 42 步进电机，需要在 42 步进电机尾部轴上粘贴径向磁铁，径向磁铁的磁通量 20mT~100mT，安装好磁铁后，再将板子安装在电机尾部，具体要求如下。

3.1 安装使用

该产品适用 42 步进电机进行使用：

1. 在电机轴后端加装径向磁铁，如图，尽量安装同轴，避免偏心磁编码器获取的数据误差加大：



图 3.1 编码器对磁铁的安装要求

2. 将电机驱动板安装到电机背后，将板子的螺丝孔对齐电机的螺丝孔位，并加装适当高度的垫片，确保磁铁正对着编码器，以及**确保间距在 1.5~3mm 之间**，安装得越好，控制精度就越高，如图 3.2。DISP 和 AG 尽量小，确保磁编码器芯片可以正常捕获转子上磁铁的磁场。

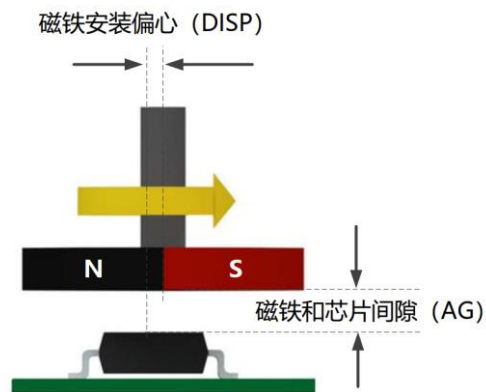


图 3.2 编码器对磁铁的安装要求

3. 将电机驱动板安装到电机背后，注意磁铁与磁编码器位置，尽量安装对正，



图 3.3 编码器对磁铁的安装要求

安装偏心 and 间隙过大带来的精度影响，可参见下图：

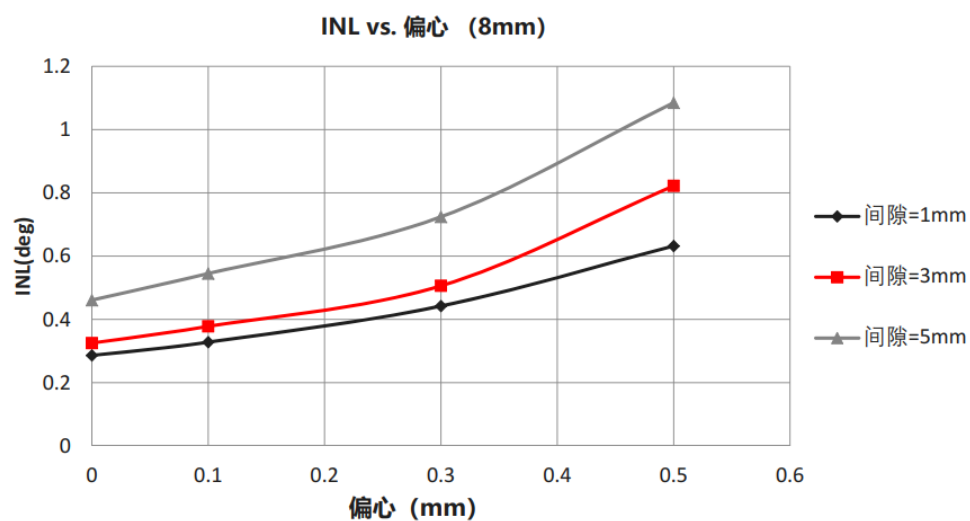


图 3.4 编码器对磁铁的安装要求

3.2 磁编校准说明

由于安装板子和磁铁时，不可避免有安装误差，可以使用软件校准磁编码器，可以大大地减小由安装误差带来的周期性角度误差，校准过程电机需开环正反转运动，在运动结束后完成编码器数据的校准。

关于校准，有如下使用须知和注意事项：

- 模块安装后，必须进行一次校准，校准完成后就可以正常使用，未校准前，给脉冲或给运动指令，电机不运行，**每个模块安装后必须校准且仅需校准一次，校准数据将会一直保存，后续上电无需校准可直接使用；**
- 校准过程中，不得通过外力强行拖动或阻碍电机校准运动，否则校准数据将会有问题；
- 校准前，建议将控制电机运动中心位置，给电机校准预留正反 2 圈左右的运动空间再进行校准，不建议电机处于机械边沿触发校准；
- 如果因为堵转、机械卡顿、机械抖动等情况导致校准数据不可靠，校准后运行效果不佳，可在解决机械问题后，重新进行校准；
- 本驱动器支持多次校准，支持重发校准指令；

4. 脉冲控制

4.1 脉冲信号要求

本驱动器软件支持 1~256 细分输入控制电机运行,最高转速可大于 2000 转,保险起见,建议按 2000RPM 使用。本节以 1.8° 步距角的步进电机为例,在不同细分下,不同转速输入的脉冲频率要求如下:

转速	2000RPM	1000RPM	500RPM	输入频率范围
1 细分	6.666KHZ	3.333KHZ	1.667KHZ	0HZ~6.666KHZ
2 细分	13.332KHZ	6.666KHZ	3.333KHZ	0HZ~13.332KHZ
4 细分	26.664KHZ	13.332KHZ	6.666KHZ	0HZ~26.664KHZ
8 细分	53.328KHZ	26.664KHZ	13.332KHZ	0HZ~53.328KHZ
16 细分	106.656KHZ	53.328KHZ	26.664KHZ	0HZ~106.656KHZ
32 细分	213.312KHZ	106.656KHZ	53.328KHZ	0HZ~213.312KHZ
64 细分	426.624KHZ	213.312KHZ	106.656KHZ	0HZ~426.624KHZ
128 细分	853.248KHZ	426.624KHZ	213.312KHZ	0HZ~853.248KHZ
256 细分	1706.496KHZ	853.248KHZ	426.624KHZ	0HZ~1706.496KHZ

关于脉冲频率,有如下注意事项:

- 输入脉冲频率需小于等于最高转速对应的频率,如此可以实现输入脉冲停止则电机停止的效果。
- 反之,会出现输入控制信号停止,但电机跑够输入脉冲的步数才停止。例如:1 细分下,按 100KHZ 频率提供脉冲 10S 后停止脉冲输入,共计 1000K 步的行程,由于电机最高转速限制,电机以最高转速跑完这 1000K 步的行程才会停止。
- 输入脉冲频率越高、走线越长越容易受干扰,建议选择合适的细分和控制频率范围和脉冲信号走线。

4.2 方向控制信号要求

方向控制信号需要与输入的控制脉冲信号形成配合,如果输入的脉冲频率对应的电机转速大于 1000RPM,建议脉冲信号保持时间 T1 大于 1ms 后再切换方向,如图 4.1 所示。避免高转速下直接切换电机运行方向,电机相电流过大导致电机冲击、发热和损坏。电机运动过程中急切电机运行方向,在受到惯性作用不可避免会带来一点机械抖动。

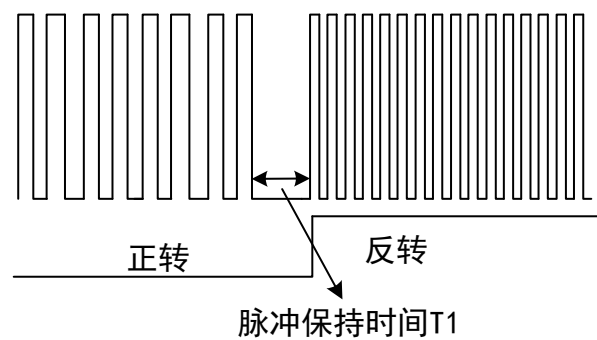


图 4.1 方向控制信号切换要求

5. 串行通信控制

MD403-x42 电调模块的通信协议使用 MODBUS RTU 协议，电调模块作为 MODBUS 从机设备。不同硬件版本支持方式不同，主要支持 TLL 串口和 RS485 两种通信方式。用做串口及 RS485 通信默认参数为：波特率 115200，数据位 8bits，停止位 1bits，校验方式是无校验。

5.1 通讯命令帧格式

通信模式是主从方式，也即主机请求，从机应答的方式。电调模块作为从机，控制电调模块的设备为主机。主请求与从应答都使用表 5.1 所示的帧格式。

表 5.1 帧格式

子节点 地址	功能代 码	数据	CRC	
1 字节	1 字节	0 到 252 字节	2 字节	
			CRC 低	CRC 高

5.1.1 子节点地址

合法的子节点地址为十进制 0 - 247。每个子设备被赋予 1 - 247 范围中的地址。主节点通过将子节点的地址放到报文的地址域对子节点寻址。当子节点返回应答时，它将自己的地址放到应答报文的地址域以让主节点知道哪个子节点在回答。

5.1.2 功能代码

指明主节点要执行的动作。读数据或者写数据。

5.1.3 数据

与功能码对应的数据。如：写数据时的写入内容，写入数量；读数据时的读取内容，读入数量。

5.1.4 CRC 校验

16 位循环冗余校验，对整个报文的内容进行 CRC 校验，最终生成一个 16 位的校验码。

5.2 数据模型

MODBUS 提供四种数据类型，分别是 1bit 的离散量输入与线圈，还有 16bit 的输入寄存器与保持寄存器。如表 5.2 所示。

表 5.2 数据模型

基本表格	对象类型	访问类型	内容
离散量输入	单个比特	只读	I/O 系统提供这种类型数据
线圈	单个比特	读写	通过应用程序改变这种类型数据
输入寄存器	16 比特字	只读	I/O 系统提供这种类型数据
保持寄存器	16 比特字	读写	通过应用程序改变这种类型数据

MD403-x42 电调模块中只使用 16bit 位的数据类型，也就是输入保持寄存器与保持寄存器。

输入寄存器是只读的，输入寄存器保存电调的实时信息，如实际转速，实际位置，当前电压、电流等。这些参数都是只读的，用户通过读取输入寄存器得到电调信息。

保持寄存器是可读可写的，保持寄存器中保存电调的目标速度，目标位置，目标力矩等参数。用户通过写入保持寄存器来设置电调速度，位置等信息。

5.3 功能代码定义

协议中的功能代码表明主节点要执行的动作。MODBUS 中公共功能码如图 5.1 所示。电调模块只使用 16 比特访问这一类的功能。常用的有 03, 04, 06, 16 这几个功能码。也就是读取输入寄存器，读取保持寄存器，写入保持寄存器。使用这几个功能码就能实现读取电调信息，以及设置电调信息。

				功能码			
				码	子码	(十六进制)	页
数据访问		物理离散量输入	读输入离散量	02		02	11
	比特访问	内部比特	读线圈	01		01	10
		或	写单个线圈	05		05	16
			物理线圈	写多个线圈	15		0F
	16 比特访问	输入存储器	读输入寄存器	04		04	14
		内部存储器 或 物理输出存储器	读多个寄存器	03		03	13
			写单个寄存器	06		06	17
			写多个寄存器	16		10	39
			读/写多个寄存器	23		17	47
			屏蔽写寄存器	22		16	46
	文件记录访问		读文件记录	20	6	14	42
			写文件记录	21	6	15	44
封装接口		读设备识别码	43	14	2B		

图 5.1 MODBUS 公用功能代码

5.4 CRC 校验

CRC 校验是基于循环冗余校验 (CRC - Cyclical Redundancy Checking) 算法的错误检验域。CRC 域检验整个报文的内容，校验码包含由两个 8 位字节组成的一个 16 位值。

CRC 的计算，开始对一个 16 位寄存器预装全 1。然后将报文中的连续的 8 位字节对其进行后续的计算。只有字符中的 8 个数据位参与生成 CRC 的运算，起始位，停止位和校验位不参与 CRC 计算。

CRC 的生成过程中，每个 8 位字符与寄存器中的值异或。然后结果向最低有效位(LSB)方向移动(Shift)1 位，而最高有效位(MSB)位置充零。然后提取并检查 LSB：如果 LSB 为 1，则寄存器中的值与一个固定的预置值异或；如果 LSB 为 0，则不进行异或操作。这个过程将重复直到执行完 8 次移位。完成最后一次（第 8 次）移位及相关操作后，下一个 8 位字节与寄存器的当前值异或，然后又同上面描述过的一样重复 8 次。当所有报文中子节都运算之后得到的寄存器中的最终值，就是 CRC。当 CRC 附加在报文之后时，首先附加低字节，然后是高字节。

具体实现可参考附录 A。

5.5 功能码使用详解

5.5.1 04(0x04) 读输入寄存器

功能码 0x04 用于读取输入寄存器的值。通过主机发送要读取的输入寄存器起始地址、读取数量，从机就会返回对应寄存器的值。通信的具体数据格式如表 5.3、表 5.4 所示。

表 5.3 主机请求格式

功能码	1 个字节	0x04
输入寄存器起始地址	2 个字节	0x0000 至 0xFFFF
输入寄存器数量	2 个字节	0x0001 至 0x007D

表 5.4 从机响应格式

功能码	1 个字节	0x04
字节数	1 个字节	2×N
输入寄存器值	2×N 个字节 (N = 请求帧中的寄存器数量)	0x0000 至 0xFFFF

表 5.5 中提供了一个具体的示例。示例中请求读取节点地址 0x0A 的输入寄存器 0x09，结果为输入寄存器 0x09 对应数据为 0x000A。

表 5.5 输入寄存器读取示例

请求		响应	
节点地址	0x0A	节点地址	0x0A
功能码	0x04	功能码	0x04
起始地址 Hi	0x00	字节数	0x02

起始地址 Lo	0x09	输入寄存器 Hi	0x00
输入寄存器数量 Hi	0x00	输入寄存器 Lo	0x0A
输入寄存器数量 Lo	0x01	CRC 校验 Lo	0x9C
CRC 校验 Lo	0xE0	CRC 校验 Hi	0xF6
CRC 校验 Hi	0xB3		

5.5.2 03(0x03) 读保持寄存器

功能码 0x03 用于读取保持寄存器的值。通过主机发送要读取的保持寄存器起始地址、读取的数量，从机就会返回对应寄存器的值。通信的具体数据格式如表 5.6、表 5.7 所示

表 5.6 主机请求格式

功能码	1 个字节	0x03
保持寄存器起始地址	2 个字节	0x0000 至 0xFFFF
保持寄存器数量	2 个字节	0x0001 至 0x007D

表 5.7 从机响应格式

功能码	1 个字节	0x03
字节数	1 个字节	2×N
输入寄存器值	2×N 个字节 (N = 请求帧中的寄存器数量)	0x0000 至 0xFFFF

表 5.8 提供了具体读取的示例, 示例中请求读取节点地址 0x0A 的保持寄存器 0x03-0x05, 读取出 0x03-0x05 保持寄存器的值为: 0x0004, 0x0005, 0x0006。

表 5.8 保持寄存器读取示例

请求		响应	
节点地址	0x0A	节点地址	0x0A
功能码	0x03	功能码	0x03
起始地址 Hi	0x00	字节数	0x06
起始地址 Lo	0x03	寄存器值(0x03) Hi	0x00
保持寄存器数量 Hi	0x00	寄存器值(0x03) Lo	0x04
保持寄存器数量 Lo	0x03	寄存器值(0x04) Hi	0x00
CRC 校验 Lo	0xF4	寄存器值(0x04) Lo	0x05
CRC 校验 Hi	0xB0	寄存器值(0x05) Hi	0x00
		寄存器值(0x05) Lo	0x06

		CRC 校验 Lo	0x33
		CRC 校验 Hi	0x86

5.5.3 06(0x06) 写单个寄存器

功能码 0x06 用于写入单个保持寄存器的值。通过主机发送要写入的保持寄存器地址、写入的数值，从机就能完成写入，写入完成后发出应答。通信的具体数据格式如表 5.9、表 5.10 所示。

表 5.9 主机请求格式

功能码	1 个字节	0x06
保持寄存器地址	2 个字节	0x0000 至 0xFFFF
写入的寄存器值	2 个字节	0x0000 至 0xFFFF

表 5.10 从机响应格式

功能码	1 个字节	0x06
保持寄存器地址	2 个字节	0x0000 至 0xFFFF
写入的寄存器值	2 个字节	0x0000 至 0xFFFF

表 5.11 提供了具体的写入示例。示例中请求把节点地址 0x0A 的保持寄存器 0x06 写入为 0x1122。

表 5.11 写单个保持寄存器示例

请求		响应	
节点地址	0x0A	节点地址	0x0A
功能码	0x06	功能码	0x06
保持寄存器地址 Hi	0x00	保持寄存器地址 Hi	0x00
保持寄存器地址 Lo	0x06	保持寄存器地址 Lo	0x06
写入的寄存器值 Hi	0x11	写入的寄存器值 Hi	0x11
写入的寄存器值 Lo	0x22	写入的寄存器值 Lo	0x22
CRC 校验 Lo	0xF4	CRC 校验 Lo	0xF4
CRC 校验 Hi	0xF8	CRC 校验 Hi	0xF8

5.5.4 16(0x10) 写多个寄存器

功能码 0x0A 用于写入多个保持寄存器的值。通过主机发送要写入的保持寄存器起始地址、写入的寄存器数量、写入的数值，从机就能完成写入动作，写入完成后发出应答。具体

通信数据格式如表 5.12、表 5.13 所示。

表 5.12 主机请求示例

功能码	1 个字节	0x10
保持寄存器起始地址	2 个字节	0x0000 至 0xFFFF
写入的寄存器数量	2 个字节	0x0001 至 0x0078
写入的字节数	1 个字节	2×N
写入的寄存器值	2×N 个字节 (N=写入的寄存器数量)	0x0000 至 0xFFFF

表 5.13 从机响应格式

功能码	1 个字节	0x10
保持寄存器地址	2 个字节	0x0000 至 0xFFFF
写入的寄存器数量	2 个字节	0x0001 至 0x0078

表 5.14 提供了写入的具体示例。示例中请求把节点地址 0x0A 的保持寄存器 0x06 0x07 写入为 0x1122 0x3344。

表 5.14 写入多个保持寄存器示例

请求		响应	
节点地址	0x0A	节点地址	0x0A
功能码	0x10	功能码	0x10
寄存器起始地址 Hi	0x00	寄存器起始地址 Hi	0x00
寄存器起始地址 Lo	0x06	寄存器起始地址 Lo	0x06
写入寄存器数量 Hi	0x00	写入寄存器数量 Hi	0x00
写入寄存器数量 Lo	0x02	写入寄存器数量 Lo	0x02
写入的字节数	0x04	CRC 校验 Lo	0xA0
写入寄存器值 1Hi	0x11	CRC 校验 Hi	0xB2
写入寄存器值 1Lo	0x22		
写入寄存器值 2Hi	0x33		
写入寄存器值 2Lo	0x44		
CRC 校验 Lo	0xF4		
CRC 校验 Hi	0xF8		

5.6 电调模块 MODBUS 通讯定义

5.6.1 从机地址

默认为 0x0A, 可通过修改通信指令修改从机地址。

5.6.2 寄存器定义

电调模块的输入寄存器(只读)定义如表 5.15 所示。

表 5.15 输入寄存器定义

输入寄存器地址	电调参数	单位
0x1000	当前转速(高 16 位)	RPM
0x1001	当前转速(低 16 位)	
0x1002	当前位置(高 16 位)	编码器线数
0x1003	当前位置(低 16 位)	
0x1004	当前力矩电流(高 16 位)	mA
0x1005	当前力矩电流(低 16 位)	
0x1006	运行错误码(高 16 位)	电机状态码
0x1007	运行错误码(低 16 位)	
0x1008	母线电压(高 16 位)	mV
0x1009	母线电压(低 16 位)	
0x100A	软件版本号	0x4xxx
0x100B	硬件版本号	0xAABB(AA:硬件版本 BB:硬件子版本)

电调模块的保持寄存器(可读可写)定义如表 5.16 所示。

表 5.16 保持寄存器定义

保持寄存器地址	电调参数	单位
0x0000	目标转速(高 16 位)	RPM
0x0001	目标转速(低 16 位)	
0x0002	目标位置(高 16 位)	编码器线数
0x0003	目标位置(低 16 位)	
0x0004	目标力矩电流(高 16 位)	mA
0x0005	目标力矩电流(低 16 位)	
0x0006	运行模式	1-力矩模式,2-速度模式 3-位置模式,4-脉冲模式

0x0007	从机地址	1 - 247
0x0008	校准	1-启动编码器校准
0x000e	同步功能码	0x0A 设定同步参数 0x0F 为启动同步运行
0x000f	参数保存标志	写入 0x55AA 后参数将更新到 FLASH 中，写入成功后电调将此值 清零

5.7 参数更新流程

MD 电调的参数更新协议使用 MODBUS RTU 协议。电调的参数表在 MODBUS 保持寄存器中，每个参数占用 2 个保持寄存器，即每个参数大小为 4 个字节。参数使用浮点数进行保存，读取参数后，需要将数据转化为浮点数。

MD 系列电调伺服参数分布如表 5.17 所示。

表 5.17 伺服参数

参数名称	寄存器起始地址	参数名称	寄存器起始地址
步距角	0x10	电流带通滤波使能	0x6C
ADC 校准（步进电机此参数留空）	0x12	电流带通滤波低频点 Hz	0x6E
编码器校准模式	0x14	电流带通滤波高频点 Hz	0x70
预充电时间 ms	0x16	速度前馈放大	0x72
转子对齐时间 ms	0x18	速度前馈滤波 Hz	0x74
调用频率 Hz	0x1A	速度滤波使能	0x76
额定转速 RPM	0x1C	速度滤波频率 Hz	0x78
速度环频率 Hz	0x1E	位置 PID 比例系数 KP	0x7A
位置环频率 Hz	0x20	位置 PID 积分系数 TI	0x7C
软件过流值 A	0x22	位置 PID 饱和系数 KC	0x7E
最大相电流 A	0x24	位置 PID 微分系数 TD	0x80
额定相电压 V	0x26	位置 PID 最大输出	0x82
额定电压 V	0x28	位置 PID 定点格式	0x84
过压电压 V	0x2A	位置前馈放大	0x86
欠压电压 V	0x2C	位置前馈滤波 Hz	0x88
流电流 A	0x2E	PWM 载波频率 Hz	0x8A

MD403-x42 步进伺服专用驱动电调

用户手册

User Manual

编码器校准速度 V	0x30	PWM 死区时间 ns	0x8C
预充电占空比	0x32	编码器分辨率	0x8E
开环拖动加速度 RPM/s	0x34	编码器转速滤波频率	0x90
电流滤波频率 HZ	0x36	编码器调用频率	0x92
运行模式	0x38	编码器比例	0x94
电流加减速模式	0x3A	编码器积分	0x96
电流加速度 mA/s	0x3C	步进细分	0x98
电流减速度 mA/s	0x3E	恒流参数 1	0x9A
速度加减速模式	0x40	恒流参数 2	0x9C
速度加速度 RPM/s	0x42	恒流限制电压	0x9E
速度减速度 RPM/s	0x44	校准对齐电压	0xA0
位置加减速模式	0x46	校准开环电压	0xA2
位置加速度 P/s	0x48	控制模式	0xA4
位置减速度 P/s	0x4A	电流环使能	0xA6
位置 S 形 J 参数 ms	0x4C	D 轴控制最大电压 V	0xA8
D 轴 PID 比例系数 KP	0x4E	速度堵转检测使能	0xAA
D 轴 PID 积分系数 TI	0x50	速度堵转检测速度 rpm	0xAC
D 轴 PID 饱和系数 KC	0x52	速度堵转检测时间 ms	0xAE
D 轴 PID 饱和系数 Td	0x54	位置堵转检测使能	0xB0
Q 轴 PID 比例系数 KP	0x56	位置堵转检测位置线	0xB2
Q 轴 PID 积分系数 TI	0x58	位置堵转检测时间 ms	0xB4
Q 轴 PID 饱和系数 KC	0x5A	电流堵转检测使能	0xB6
Q 轴 PID 饱和系数 Td	0x5C	电流堵转检测阈值 mA	0xB8
速度 PID 比例系数 KP	0x5E	电流堵转检测时间 ms	0xBA
速度 PID 积分系数 TI	0x60		
速度 PID 饱和系数 KC	0x62		
速度 PID 微分系数 TD	0x64		
电流最大限定 A	0x66		
电流低通滤波使能	0x68		
电流低通滤波频率 Hz	0x6A		

5.7.1 参数读取示例

以伺服电机的“速度 PID 积分系数 TI”参数为例，由表 5.17 可知该参数寄存器起始地址为 0x60。

则读取该参数的指令如下：

主控→电调：0A 03 00 60 00 02 C5 6E

电调→主控：0A 03 04 3E 2B 02 0C 3C 76

由电调回复指令可知，该参数大小为 0x3e2b020c,将其转化为浮点数，为 0.167000。说明此时电调中的“速度 PID 积分系数 TI”参数为 0.167。

5.7.2 参数写入示例

如果想修改参数，如将“速度 PID 积分系数 TI”修改为 0.213，将浮点数 0.213 转化为十六进制为 0x3e5a1cac,可得写入指令如下：

主控→电调：0A 10 00 60 00 02 04 3E 5A 1C AC F5 E5

电调→主控：0A 10 00 60 00 02 40 AD

参数修改后，还需要发送 flash 写入指令，将参数保存到 flash。参数保存指令如下：

主控→电调：0A 06 00 0F 55 AA 07 9D

电调→主控：0A 06 00 0F 55 AA 07 9D

经上述指令后，参数就已经保存到电调中，需注意重启电调后参数才开始生效。

5.8 电调常用通讯示例

5.8.1 速度控制

1. 获取当前电机转速

请求帧：0A 04 10 00 00 02 74 70

应答帧：0A 04 04 00 00 03 E8 41 FA

获取到的电机转速为 0x000003E8, 即为 1000RPM。

2. 设置电机转速

请求帧：0A 10 00 00 00 02 04 00 00 03 E8 D6 35

应答帧：0A 10 00 00 00 02 40 B3

设置电机的转速为 1000RPM。

5.8.2 位置控制

1. 获取当前电机位置

请求帧：0A 04 10 02 00 02 D5 B0

应答帧：0A 04 04 00 00 0B B8 46 06

获取到的电机位置为 0x00000BB8, 即为 3000 线。

2. 设置电机位置

请求帧：0A 10 00 02 00 02 04 00 00 13 88 5A 04

应答帧：0A 10 00 02 00 02 E1 73

设置电机的位置为 5000 线。

5.8.3 控制模式选择

力矩控制模式的切换仅支持伺服运行模式，切换模式后，发送复位指令或重启后生效。

1. 设置电调为力矩模式

请求帧：0A 10 00 06 00 01 02 00 01 14 C6

应答帧：0A 10 00 06 00 01 E0 B3

2. 设置电调为速度模式

请求帧：0A 10 00 06 00 01 02 00 02 54 C7

应答帧：0A 10 00 06 00 01 E0 B3

3. 设置电调为位置模式

请求帧：0A 10 00 06 00 01 02 00 03 95 07

应答帧：0A 10 00 06 00 01 E0 B3

4. 重启电调

请求帧：0A 06 00 0A 12 21 64 0B

5.8.4 校准操作

使用串口控制校准仅支持在电机静止条件下，电机校准现象为正转两圈，反转两圈，校准期间请勿干扰，否则可能导致校准失败，校准完成后，即可正常使用，正常只需要校准一次。

1. 请求校准

请求帧：0A 10 00 08 00 01 02 00 01 15 E8

应答帧：0A 10 00 08 00 01 81 70

5.8.5 多机同步控制

- 1) **多机控制**：可以通过发送广播地址带控制功能码指令直接控制总线上的所有设备，从机无需回复，广播地址为 0；

广播帧设定转速示例：

1. 广播转速 300rpm

请求帧：00 10 00 00 00 02 04 00 00 0B B8 D6 35 1A 7B

- 2) **同步控制**：可以独立向所需控制的设备发送控制指令，被控设备收到后不立即执行，直到收到同步指令再一起执行；

广播帧进入同步运动：

00 10 00 0E 00 01 02 00 0A 2A E9

从机 1 设定目标位置：

01 10 00 02 00 02 04 00 00 FF FF 73 C6

从机 2 设定目标位置：

02 10 00 02 00 02 04 00 00 FF FF 7C 82

从机 3 设定目标位置：

03 10 00 02 00 02 04 00 00 FF FF 78 7E

....

广播帧同步运动：

00 10 00 0E 00 01 02 00 0F EA EA

6. 电机适配流程

MD403-x42 出厂前默认支持的电机为 BL42 步进电机，用户所用电机不尽相同，此时用户需要根据自己的电机参数更新到电调，用户可使用 MD Tool 软件可实现参数更新。

6.1 修改电调参数

参数更新功能可以查看当前电调设置的电机参数，并在软件上修改参数。

在参数更新界面，点击“电调参数获取”按钮，可获取当前连接电调的参数。可直接在输入框中修改对应的参数，修改完后，点击“电调参数写入”按钮，即可完成写入。如图 6.1 所示。



图 6.1 参数修改

6.2 电机调试

用户可以在上位机设定工作模式、设定参数，也可以通过上位机设定转速、力矩、和位置等运行参数，点击上位机的读取电调参数，会实时通过串口读取电调参数，并在上位机进行显示和波形刷新，可以初步通过波形验证参数调试。

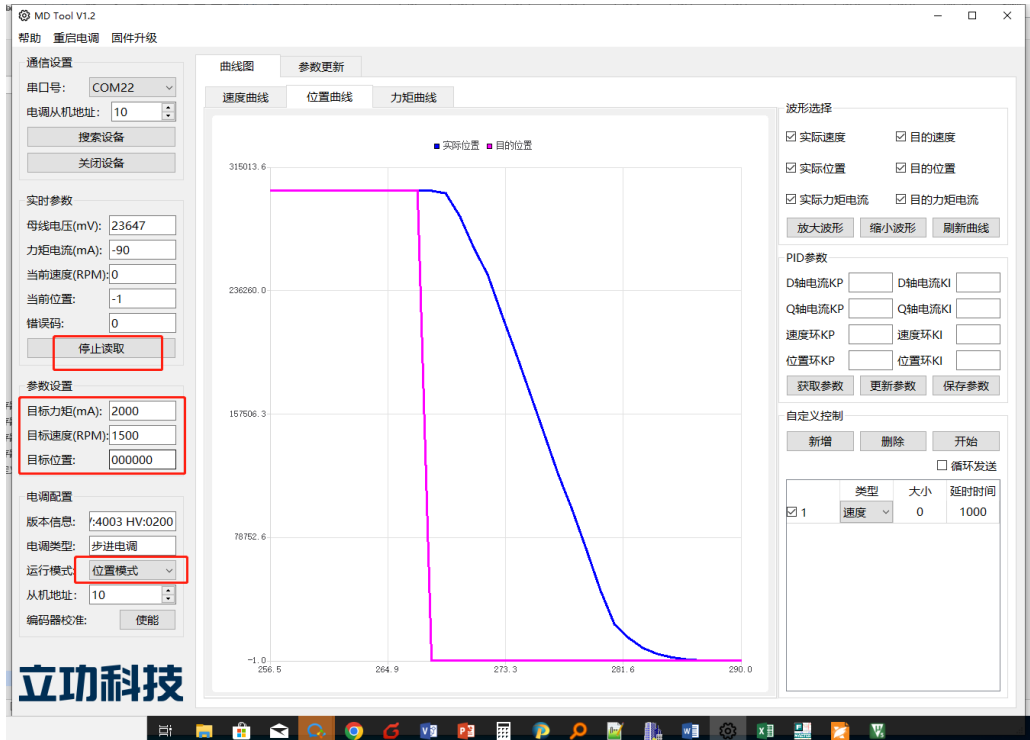


图 6.2 电机调试

6.3 固件升级

当需要进行固件升级时，可以通过上位机选择所需升级的固件进行升级。

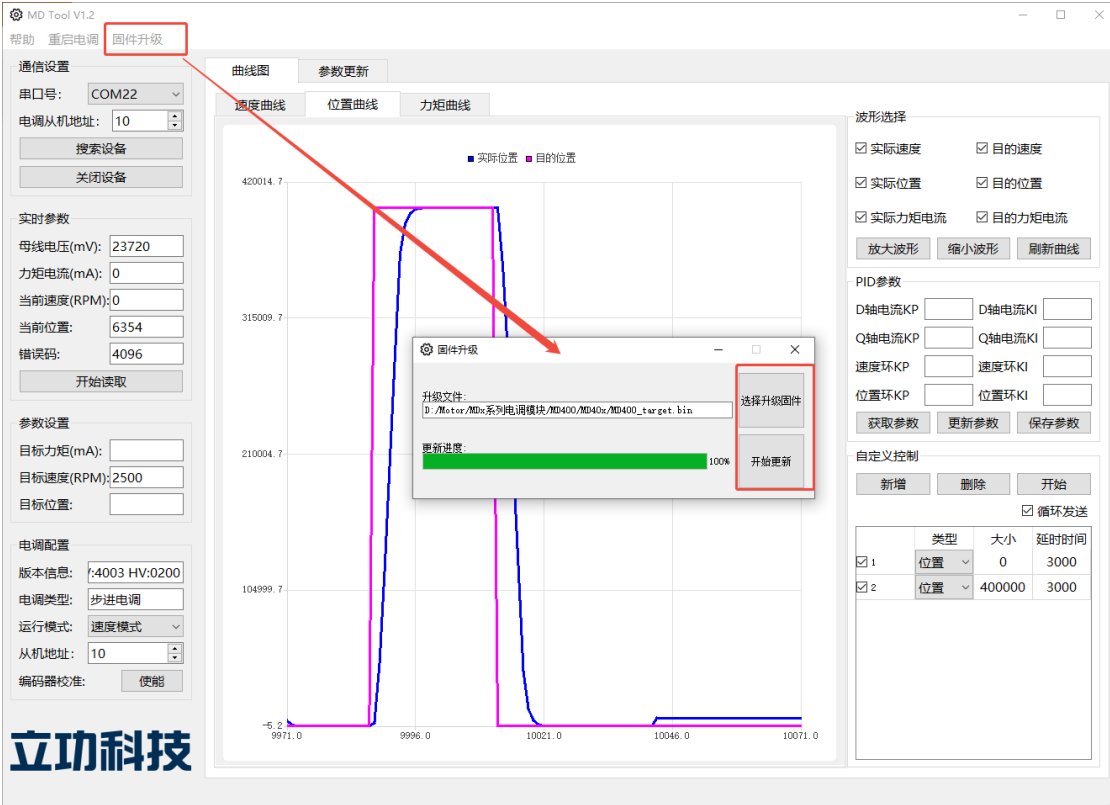


图 6.3 固件升级

```
static const UCHAR aucCRCHi[] = {  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40  
};  
  
static const UCHAR aucCRCLo[] = {  
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,  
    0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0xF0, 0xCF, 0xCE, 0x0E,  
    0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9,  
    0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,  
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,  
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,  
    0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,  
    0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,  
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF,
```

```

0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB,
0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x42, 0x97,
0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89,
0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
0x41, 0x81, 0x80, 0x40
};

```

USHORT

usMBCRC16(UCHAR * pucFrame, USHORT usLen)

```

{
    UCHAR        ucCRCHi = 0xFF;
    UCHAR        ucCRCLo = 0xFF;
    int          iIndex;

    while( usLen-- )
    {
        iIndex = ucCRCLo ^ *( pucFrame++ );
        ucCRCLo = ( UCHAR )( ucCRCHi ^ ucCRCHi[iIndex] );
        ucCRCHi = ucCRCLo[iIndex];
    }
    return ( USHORT )( ucCRCHi << 8 | ucCRCLo );
}

```

7. 免责声明

本着为用户提供更好服务的原则，广州立功科技股份有限公司（下称“立功科技”）在本手册中将尽可能地为用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，立功科技不能完全保证该文档在任何时段的时效性与适用性。立功科技有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问立功科技官方网站或者与立功科技工作人员联系。感谢您的包容与支持！

专业 · 专注成就梦想

Dreams come true with professionalism and dedication.

广州立功科技股份有限公司

更多详情请访问

www.zlgmcu.com

欢迎拨打全国服务热线

400-888-2705

