

深圳市硕科数控科技有限公司

标准通讯协议手册

本协议主要用于M6505系列闭环运动步进驱动器和M4505系列开环运动步进驱动器

版权所有：深圳市硕科数控科技有限公司

版本：v4.x

最后一次修订日期：2013年5月20日

目 录

SHUOKE rRS232和RS485通讯协议V4.0 介绍	3
开环控制RTU讯格式通讯协议	4
闭环控制RTU讯格式通讯闭环控制通讯协议	22
ASC讯格式通讯协议	27
开环控制CAN通讯协议介绍	30
闭环控制CAN通讯协议介绍	35
开环控制与各种PLC进行通讯的协议介绍	37
闭环控制与各种PLC进行通讯的协议介绍	48

SHUOKE 高性能步进电机驱动器通讯协议V4.0 (FOR M6505&M4505系列, 增加闭环命令)

V1.5基本通讯协议如下:(2012年4月份)

- 一、 V1.5在V1.3的基础上,硬件的改动,根据型号不同,功能不同,但通讯协议基本相同(具体看稍后详细说明书)
 - A. 增加标准RS232和RS485的通讯接口
 - B. 增加运动状态输出功能,PLC或其他控制器,可以直接通过IO来判断是否正在运行中.
 - C. 增加IO调速功能,该功能可以通过IO进行8档调速,达到平滑启动和停止的效果.
 - D. 不同的模式通讯协议不同。
 - E. 每个命令之间间隔时间50MS以上。

二、 通讯协议的改动

针对部分客户提出的要求,v1.5通讯协议,将按照MODBUS的RTU通讯格式来进行通讯

通讯基本设置:9600, 8, 1, n, n.

在上电前2秒钟通讯波特率固定为9600,提示当前使用的波特率系统启动过后会自动且换到用户设置的波特率,同时提示系统启动成功。

三、 **ASC通讯格式在RTU后面做相关说明, CAN通讯协议在最后做相关说明。**

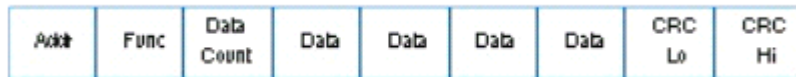
V3.0通讯协议大大增强了驱动器的性能,在高性能的硬件配合下,同时支持CAN, 485, 232的通讯,使M4505S和M8006系列驱动器成为国内和国际上**功能最强、性能最好**的一款步进驱动器,同时也是一款**独步于世界**的自主产品。

V4.0通讯协议集成了闭环控制的M6505系列运动控制器的闭环指令,其他命令与以前版本没有太大区别。不过可以到硕科数控网站下载各种运动模式的图形说明《M系列驱动器运动模式说明书.doc》。

有产品说明书、DEMO和参考原代码、详细使用说明书等,请直接到WWW.SHUOKE-TECH.COM网站下载。

讯格式:RTU (标准: 使用CRC校验)

(一)、RTU通讯的基本格式:



ADD:设备地址

FUNC: 功能代码 (命令码)

DATA COUNT: 数据长度, 从DATA的长度, 不包含CRC。

CRC校验: 标准CRC校验, 低位在前, 高位在后。

发送: 地址 + 命令 + 数据长度 + 32位参数 + CRC校验
数据长度

地址: 1-247, 广播地址为255

命令: 1-127为发送命令

数据长度: 从下一个字符开始统计, 到结束的字节数 (CRC: +2) .

参数: 4个字符 (32BIT) 的参数

校验: 前面所有的字符和取低8位.

正确返回: 地址 + 命令 + 数据长度 + 32位参数 + CRC校验
原来命令 数据长度

地址: 1-247

命令: 1-127为原发送命令, 原命令返回表示命令成功

数据长度: 从下一个字符开始统计, 到结束的字节数 (CRC: +2) .

参数: 最长4个字符 (32BIT) 的参数

校验: 前面所有的字符和取低8位.

失败返回: 地址 + 命令 + 数据长度 + 8位参数 + CRC校验
最高位置1 数据长度

地址: 1-247

命令: 1-127为发送命令, 最高位置1表示返回命令失败

数据长度: 从下一个字符开始统计, 到结束的字节数 (和校验: +1, CRC:

+2) .

参数: 返回失败的错误代码

0X01: 表示接收超时 (数据长度不对)

0x02: 表示校验错误

0x03: 该命令不存在

0X04: 表示参数超出范围

0X05: 表示该命令不支持广播

RTU模式专有握手命令，用于测试硬件通讯是否正常

(二)、基本通讯协议如下：

1、请求握手命令，不做CRC数据检查，可以参考返回的CRC校验

发送：

[设备地址] 0x01

[命令] 0x20

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

应答：

[设备地址] 0x01

[命令] 0x20

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

2、 查询设备地址命令为0x21，该命令支持广播

发送：

[设备地址] 0x00

[命令] 0x21

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

应答：

[设备地址] 0x01

[命令] 0x21

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

3、更改地址：地址数据1个字节，不支持广播，需重新启动

发送：

[设备地址] 0x01

[命令] 0x22

[数据长度] 0x01

[数据] 0Xxx (新地址)

[CRC校验1] 0xnn

[CRC校验2] 0xnn

应答：

[设备地址] 0x01

[命令] 0x22

[数据长度] 0x01

[数据] 0Xxx (新地址)

[CRC校验1] 0xnn

[CRC校验2] 0xnn

4、查询当前波特率

发送：

[设备地址] 0x01

[命令] 0x23

[数据长度] 0x00

应答：

[设备地址] 0x01

[命令] 0x23

[数据长度] 0x01

[数据]	无	[数据]	波特率代号
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

5、改变当前波特率，不支持广播，

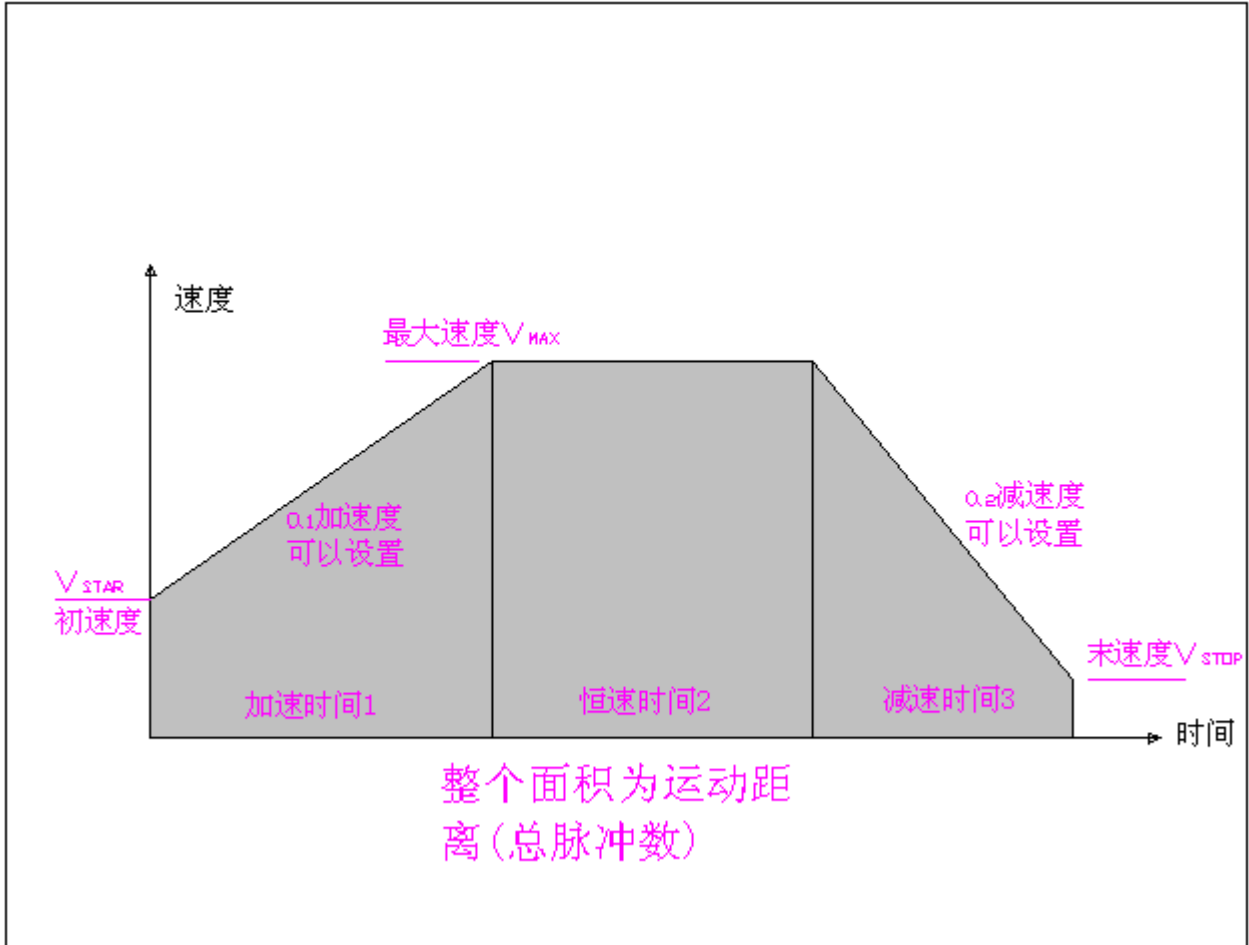
发送：		应答：	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x24	[命令]	0x24
[数据长度]	0x01	[数据长度]	0x01
[数据]	波特率代号	[数据]	波特率代号
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

波特率代号：

0x01: 300 **0x02:600** **0x03:1200** 0x04:2400 0x05:4800
0x06:9600 0x07:19200 0x08:38400 **0x09:57600(红色保留不用)**

////////////////////////////////////

运动参数（初速度，加速度，最大速度，减速度，停止速度）示意图：



以下用于设置位置运动参数

6、设置运动初速度：不支持广播（32位脉冲统计长度）

发送：

[设备地址] 0x01
 [命令] 0x25
 [数据长度] 4
 [数据] xx xx xx xx
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答：

[设备地址] 0x01
 [命令] 0x25
 [数据长度] 0
 [数据] 无
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据：4BYTE

运动初速度4BYTE，数据范围：0---100K，超出范围报错0X04

7、读取运动初速度：不支持广播（32位脉冲统计长度）

发送：

[设备地址] 0x01
 [命令] 0x26
 [数据长度] 0

应答：

[设备地址] 0x01
 [命令] 0x26
 [数据长度] 4

[数据]	无	[数据]	xx xx xx xx
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据: 4BYTE

运动初速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

8、设置运动最大速度: 不支持广播 (32位脉冲统计长度)

发送:		应答:	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x27	[命令]	0x27
[数据长度]	4	[数据长度]	0
[数据]	xx xx xx xx	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据: 4BYTE

运动最大速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

9、读取运动最大速度: 不支持广播 (32位脉冲统计长度)

发送:		应答:	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x28	[命令]	0x28
[数据长度]	0	[数据长度]	4
[数据]	无	[数据]	xx xx xx xx
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据: 4BYTE

运动最大速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

10、设置运动末速度: 不支持广播 (32位脉冲统计长度)

发送:		应答:	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x29	[命令]	0x29
[数据长度]	4	[数据长度]	0
[数据]	xx xx xx xx	[数据]	无

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE

运动末速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

11、读取设置运动末速度: 不支持广播 (32位脉冲统计长度)

发送:

应答:

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x2A

[命令] 0x2A

[数据长度] 0

[数据长度] 4

[数据] 无

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE

运动末速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

12、设置运动加速度: 不支持广播 (32位脉冲统计长度)

发送:

应答:

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x2B

[命令] 0x2B

[数据长度] 4

[数据长度] 0

[数据] xx xx xx xx

[数据] 无

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE (无负数, 用来做固定脉冲数运动的速度)

运动加速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

13、读取运动加速度: 不支持广播 (32位脉冲统计长度)

发送:

应答:

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x2C

[命令] 0x2C

[数据长度] 0

[数据长度] 4

[数据] 无

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE (无负数, 用来做固定脉冲数运动的速度)

运动加速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

14、设置运动减速度: 不运动, 不支持广播 (32位脉冲统计长度)

发送:

[设备地址] 0x01

[命令] 0x2D

[数据长度] 4

[数据] xxxxxxxx

[CRC校验1] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE

运动减速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

15、读取运动减速度: 不运动, 不支持广播 (32位脉冲统计长度)

发送:

[设备地址] 0x01

[命令] 0x2E

[数据长度] 0

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE

运动减速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

16、设置归零运动速度: 不运动, 不支持广播 (32位脉冲统计长度) (M4505)

不支持该命令)

发送:

[设备地址] 0x01

[命令] 0x2F

[数据长度] 4

[数据] xxxxxxxx

[CRC校验1] 0xnn

[CRC校验2] 0xnn

数据: 4BYTE

运动归零速度4BYTE, 数据范围: 0----100K, 超出范围报错0X04

应答:

[设备地址] 0x01

[命令] 0x2F

[数据长度] 0

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

17、读取归零运动速度：不运动，不支持广播（32位脉冲统计长度）**(M4505 不支持该命令)**

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x30	[命令] 0x30
[数据长度] 0	[数据长度] 4
[数据] 无	[数据] xxxxxxxx
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：4BYTE

运动归零速度4BYTE，数据范围：0----100K，超出范围报错0X04

////////////////////////////////////
 18、设置运动脉冲数并开始运动：不支持广播（32位脉冲统计长度）
 方向在脉冲正负数中，正数表示正方向，负数表示负方向。

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x31	[命令] 0x31
[数据长度] 4	[数据长度] 0
[数据] xxxxxxxx	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：4BYTE 运动脉冲数4BYTE，根据最高位是否为1来判断方向
 也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来判断脉冲数**（支持31位脉冲数）**。

19、设置运动脉冲数并开始运动：不支持广播（32位脉冲统计长度）

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x32	[命令] 0x32
[数据长度] 4	[数据长度] 0
[数据] xxxxxxxx	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：4BYTE，第一位为方向，1为正，0位负，运动脉冲数为后3位（支持24位无符号脉冲数）

////////////////////////////////////

以下用于恒速运动

20、设置运动速度，并恒速运动命令：不支持广播
方向在脉冲正负数中，正数表示正方向，负数表示负方向。

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x33	[命令] 0x33
[数据长度] 4	[数据长度] 0
[数据] xxxxxxxx	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：4BYTE 运动脉冲数4BYTE，根据最高位是否为1来判断方向
也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来判断脉冲数（支持31位脉冲数）。

21、设置运动速度，并恒速运动命令：不支持广播

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x34	[命令] 0x34
[数据长度] 4	[数据长度] 0
[数据] xx xx xx xx	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：4BYTE，第一位为方向，1为正，0位负，运动脉冲数为后3位（支持24位无符号脉冲数）

22、设置运动过程中，改变速度：不支持广播 (M4505不支持该命令)

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x35	[命令] 0x35
[数据长度] 4	[数据长度] 0
[数据] xxxxxxxx	[数据] 无

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据: 4BYTE只改速度, 不改方向

////////////////////////////////////

以下用于查询运动状态

23、查询运动速度: 不支持广播

发送:

应答:

[设备地址] 0x01 [设备地址] 0x01

[命令] 0x36 [命令] 0x36

[数据长度] 0 [数据长度] 0x4

[数据] 无 [数据] xxxxxxxx

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据: 4BYTE运动速度4BYTE, 最大不超过100K

24、查询运动绝对位置: 不支持广播

发送:

应答:

[设备地址] 0x01 [设备地址] 0x01

[命令] 0x37 [命令] 0x37

[数据长度] 0 [数据长度] 0x4

[数据] 无 [数据] xxxxxxxx

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据: 4BYTE 运动绝对位置带符号

25、绝对位置清零: 不支持广播

发送:

应答:

[设备地址] 0x01 [设备地址] 0x01

[命令] 0x38 [命令] 0x38

[数据长度] 0 [数据长度] 0x0

[数据] 无 [数据] 无

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

26、查询运动相对位置: 不支持广播

发送:

应答:

[设备地址] 0x01 [设备地址] 0x01

[命令] 0x39 [命令] 0x39

[数据长度] 0	[数据长度] 0x4
[数据] 无	[数据] xxxxxxxx
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据: 4BYTE 运动相对位置带符号

27、相对位置清零: 不支持广播

发送:	应答:
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x3A	[命令] 0x3A
[数据长度] 0	[数据长度] 0x0
[数据] 无	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

28、查询运动状态命令: 不支持广播

发送:	应答:
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x3B	[命令] 0x3B
[数据长度] 0x00	[数据长度] 0x05
[数据] 无	[数据] xx xx xx xx xx
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据:

[数据1]: 1表示运动中, 0表示停止
 [数据2]: LL-左限位IO状态 1表示有效 0表示无效
 [数据3]: 0-原点IO状态 1表示有效 0表示无效
 [数据4]: RL-右限位IO状态 1表示有效 0表示无效
 [数据5]: EN-使能IO状态 1表示有效 0表示无效

以下命令用于停止操作

29、急停命令: 支持广播

发送:	应答:
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x3C	[命令] 0x3C
[数据长度] 0x00	[数据长度] 0x00

[数据]	无	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

30、减速停止命令：支持广播

发送：		应答：	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x3D	[命令]	0x3D
[数据长度]	0x00	[数据长度]	0x00
[数据]	无	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

31、自动查找原点命令1：(M4505不支持该命令)

发送：		应答：	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x3e	[命令]	0x3e
[数据长度]	0x01	[数据长度]	0x00
[数据]	0/1	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据：0表示从反方向开始查找，1表示从正方向开始查找，遇上限位自动反方向查找，查找到原点以后，自动将相对位置和绝对位置清零。

自动查找原点命令2：(M4505不支持该命令)

发送：		应答：	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x4e	[命令]	0x3e
[数据长度]	0x01	[数据长度]	0x00
[数据]	0/1	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据：0表示从反方向开始查找，1表示从正方向开始查找，先到限位，然后反方向查找，查找到原点以后，自动将相对位置和绝对位置清零。

自动查找原点命令3：(M4505不支持该命令)

发送：		应答：	
-----	--	-----	--

[设备地址]	0x01	[设备地址]	0x01
[命令]	0x4f	[命令]	0x3e
[数据长度]	0x01	[数据长度]	0x00
[数据]	0/1	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据：0表示从反方向开始查找，1表示从正方向开始查找，遇上限位自动反方向查找，查找到原点以后(该命令用来做扩展命令使用)。

32、设置与西门子PLC通讯内存映射地址

通讯地址 = 映射地址 + 命令代码

发送：		应答：	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x3f	[命令]	0x3f
[数据长度]	0x02	[数据长度]	0x00
[数据]	0xnn, 0xnn	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

说明：具体看最后 驱动器与西门子PLC通讯部分。

以下高级命令说明：

33、设置IO取代命令的命令：(M4505不支持该命令)

发送：		应答：	
[设备地址]	0x01	[设备地址]	0x01
[命令]	0x40	[命令]	0x40
[数据长度]	0x07	[数据长度]	0x00
[数据]	0-7BYTE	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据第1位为0x01-0x0f表示IO命令的编号，后面是命令的相关数据。

34、读取IO取代命令的命令：(M4505不支持该命令)

发送:

[设备地址] 0x01
 [命令] 0x41
 [数据长度] 0x01
 [数据] 0x01-0x0f
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答:

[设备地址] 0x01
 [命令] 0x41
 [数据长度] 0xnn
 [数据] 0-7BYTE
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据: 0x01-0x0f表示IO命令的编号, 返回数据该IO命令的相关数据。

35、查询IO状态的命令: (M4505不支持该命令)

发送:

[设备地址] 0x01
 [命令] 0x42
 [数据长度] 0x00
 [数据] 无
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答:

[设备地址] 0x01
 [命令] 0x41
 [数据长度] 0x05
 [数据] xx xx xx xx xx
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据: 返回数据依次为: I01/I02/I03/I04/I05 相关0/1数据。

36、其他输入状态的命令: (M4505不支持该命令)

发送:

[设备地址] 0x01
 [命令] 0x43
 [数据长度] 0x00
 [数据] 无
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答:

[设备地址] 0x01
 [命令] 0x41
 [数据长度] 0x05
 [数据] xx xx xx xx xx
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据: 返回数据依次为: 脉冲pu/方向dr/使能en/f1/f2 相关0/1数据。

37、设置运动中改变运动速度因子命令: (M4505S特有命令)

该命令专为0X45, 0X46命令使用, 无其他作用

发送:

[设备地址] 0x01
 [命令] 0x44

应答:

[设备地址] 0x01
 [命令] 0x41

[数据长度] 0x04	[数据长度] 0x00
[数据] 32位数据	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：设置每次0X45，0X46命令所改变的速度数量，该数据称为F。

38、运动中定量改变速度（加）命令：(M4505S特有命令)

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x45	[命令] 0x41
[数据长度] 0x00	[数据长度] 0x00
[数据] 无	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：运动速度 = 当前运动速度 + 改变运动速度因子F。

39、运动中定量改变速度（减）命令：(M4505S特有命令)

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x46	[命令] 0x41
[数据长度] 0x00	[数据长度] 0x00
[数据] 无	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：运动速度 = 当前运动速度 - 改变运动速度因子F。

40、恢复出厂设置命令：(M4505S特有命令)

发送：	应答：
[设备地址] 0x01	[设备地址] 0x01
[命令] 0x47	[命令] 0x47
[数据长度] 0x00	[数据长度] 0x00
[数据] 无	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：所有变量全部重新初始化，重新启动后生效。

41、接收命令是否取消CRC校验检查命令：(返回不取消) (M4505S特有命令)

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x48	[命令] 0x48
[数据长度] 0x01	[数据长度] 0x01
[数据] 0x00/0x01	[数据] 0x00/0x01
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：0x00:要检查CRC校验, 0x01:不检查CRC校验。

说明：取消校验会有20%的出错概率，建议不取消。

42、是否固定返回8个长度命令(包含CRC10个)：(M4505S特有命令)

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x49	[命令] 0x49
[数据长度] 0x01	[数据长度] 0x01
[数据] 0x00/0x01	[数据] 0x00/0x01
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：0x00:不是固定8个字符长度, 0x01:固定8个字符长度。

说明：驱动器接受命令同时支持这两种格式，建议用不固定长度格式。

43、CAN总线发送数据使用标准帧还是扩展帧：(M4505S特有命令)

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x4a	[命令] 0x4a
[数据长度] 0x01	[数据长度] 0x01
[数据] 0x00/0x01	[数据] 0x00/0x01
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：0x00:标准帧, 0x01:扩展帧。(扩展帧扩展ID固定等于0)

44、CAN总线发送数据通讯波特率设置：(M4505S特有命令)

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x4b	[命令] 0x4b
[数据长度] 0x01	[数据长度] 0x01

[数据]	0xnn	[数据]	0xnn
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

数据：数据范围是1-18, 默认设置是12(100K通讯波特率)

CAN波特率 1:1M 2:900K 3:800K 4:750K 5:600K 6:500K 7:400K 8:250K 9:200K
10:150K 11:125K 12:100K 13:80K 14:50K 15:25K 16:20K 17:10K 18:5K

45、由232或485发送的数据直接到CAN发送出来：（用于CAN通讯调试，仅RTU）
232或485： CAN发送： (M4505S特有命令)

[设备地址]	0x01	[设备地址]	0x01
[命令]	0x4c	[命令]	0x4c
[数据长度]	0x05	[数据长度]	0x05
[数据]	5个任意数据	[数据]	5个任意数据
[CRC校验1]	0xnn		
[CRC校验2]	0xnn		

说明：1. CRC校验在转给CAN的时候，直接去掉。用于CAN调试

46、由CAN发送的数据直接到232或485发送出来：（用于CAN通讯调试，仅RTU）
CAN发送： 232或485： (M4505S特有命令)

[设备地址]	0x01	[设备地址]	0x01
[命令]	0x4d	[命令]	0x4d
[数据长度]	0x05	[数据长度]	0x05
[数据]	5个任意数据	[数据]	5个任意数据
		[CRC校验1]	0xnn
		[CRC校验2]	0xnn

说明：CRC校验在转给串口的时候，自动添加上去。用于CAN调试

47、设置运动脉冲数并开始S曲线运动：不支持广播（32位脉冲统计长度）
方向在脉冲正负数中，正数表示正方向，负数表示负方向。(M4505S特有命令)
发送： 应答：

[设备地址]	0x01	[设备地址]	0x01
[命令]	0x4e	[命令]	0x4e
[数据长度]	4	[数据长度]	0
[数据]	xxxxxxxx	[数据]	无

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据：4BYTE 运动脉冲数4BYTE，根据最高位是否为1来判断方向
也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来判断脉冲数（支持31位脉冲数）。

48、设置运动脉冲数并开始S曲运动：不支持广播（32位脉冲统计长度）
方向在脉冲正负数中，正数表示正方向，负数表示负方向。（M4505S特有命令）

发送：

应答：

[设备地址] 0x01 [设备地址] 0x01

[命令] 0x4f [命令] 0x4f

[数据长度] 4 [数据长度] 0

[数据] xxxxxxxx [数据] 无

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据：4BYTE，第一位为方向，运动脉冲数为后3位（支持24位无符号脉冲数）

以下为闭环运动控制所专有命令：

1、设置运动反馈脉冲数并开始运动：不支持广播（32位反馈脉冲统计长度）方向在脉冲正负数中，正数表示正方向，负数表示负方向。

发送：

应答：

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x55

[命令] 0x55

[数据长度] 4

[数据长度] 0

[数据] xxxxxxxx

[数据] 无

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据：4BYTE 运动反馈脉冲数4BYTE，根据最高位是否为1来判断方向也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来反馈判断脉冲数（支持31位脉冲数）。

2、使用电子齿轮运算，设置运动反馈脉冲数并开始运动：不支持广播（32位反馈脉冲统计长度）方向在脉冲正负数中，正数表示正方向，负数表示负方向。

实际反馈脉冲数 = 设定反馈脉冲数 * 齿轮分子 / 齿轮分母

最高支持数据为运算后的实际反馈脉冲数为31位反馈脉冲数

发送：

应答：

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x56

[命令] 0x56

[数据长度] 4

[数据长度] 0

[数据] xxxxxxxx

[数据] 无

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据：4BYTE 运动反馈脉冲数4BYTE，根据最高位是否为1来判断方向也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来反馈判断脉冲数（支持31位设定脉冲数）。

3、设置运动反馈脉冲绝对位置并运动：不支持广播（32位反馈脉冲统计长度）方向在脉冲正负数中，正数负数表示编码器反馈的绝对位置。

发送：

应答：

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x57

[命令] 0x57

[数据长度] 4

[数据长度] 0

[数据] xxxxxxxx

[数据] 无

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据：4BYTE 运动反馈绝对位置数4BYTE，根据数的绝对位置值来判断是否运动以及运动距离（支持31位编码器绝对位置数，计算后的实际运动距离最大为31位数据）。

4、读取闭环绝对位置

发送：

应答：

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x58

[命令] 0x58

[数据长度] 0

[数据长度] 0x4

[数据] 无

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据：4BYTE运动反馈位置，编码器的反馈位置。

5、闭环绝对位置设置一个值

发送：

应答：

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x59

[命令] 0x59

[数据长度] 0（或4）

[数据长度] 0x00

[数据] xx xx xx xx

[数据]无

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据：设置4BYTE运动反馈绝对位置，无数值就将该变量清零。

6、读取闭环相对位置

发送：

应答：

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x5a

[命令] 0x5a

[数据长度] 0

[数据长度] 0x4

[数据] 无

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

数据：4BYTE运动反馈相对位置，编码器的反馈相对位置。

7、闭环相对位置设置一个值

发送:

[设备地址] 0x01
 [命令] 0x5b
 [数据长度] 0 (或4)
 [数据] xx xx xx xx
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答:

[设备地址] 0x01
 [命令] 0x5b
 [数据长度] 0x00
 [数据] 无
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据: 设置4BYTE运动反馈相对位置, 无数值就将该变量清零。

8、设置编码器方向

发送:

[设备地址] 0x01
 [命令] 0x5c
 [数据长度] 0x01
 [数据] xx
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答:

[设备地址] 0x01
 [命令] 0x5c
 [数据长度] 0x00
 [数据] 无
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据: 0编码器正方向, 1编码器反方向。

9、设置运动方向

发送:

[设备地址] 0x01
 [命令] 0x5d
 [数据长度] 0x01
 [数据] xx
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

应答:

[设备地址] 0x01
 [命令] 0x5d
 [数据长度] 0x00
 [数据] 无
 [CRC校验1] 0xnn
 [CRC校验2] 0xnn

数据: 0正常运动方向, 1反方向运动。

10、设置自动调整积累误差开关

发送:

[设备地址] 0x01
 [命令] 0x5e
 [数据长度] 0x01
 [数据] xx

应答:

[设备地址] 0x01
 [命令] 0x5e
 [数据长度] 0x00
 [数据] 无

[CRC校验1] 0xnn [CRC校验1] 0xnn

[CRC校验2] 0xnn [CRC校验2] 0xnn

数据：0无积累误差，1有积累误差。

11、设置电子齿轮分子

发送：

[设备地址] 0x01

[命令] 0x5f

[数据长度] 0x04

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验2] 0xnn

应答：

[设备地址] 0x01

[命令] 0x5f

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

12、读取电子齿轮分子

发送：

[设备地址] 0x01

[命令] 0x60

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

应答：

[设备地址] 0x01

[命令] 0x60

[数据长度] 0x04

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验2] 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

13、设置电子齿轮分母

发送：

[设备地址] 0x01

[命令] 0x61

[数据长度] 0x04

[数据] xx xx xx xx

[CRC校验1] 0xnn

[CRC校验2] 0xnn

应答：

[设备地址] 0x01

[命令] 0x61

[数据长度] 0x00

[数据] 无

[CRC校验1] 0xnn

[CRC校验2] 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

14、读取电子齿轮分母

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x62	[命令] 0x62
[数据长度] 0x00	[数据长度] 0x04
[数据] 无	[数据] xx xx xx xx
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

15、设置编码器AB相信号允许误差（默认200，建议不改动）

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x65	[命令] 0x65
[数据长度] 0x04	[数据长度] 0x00
[数据] xx xx xx xx	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：默认200，用来检查AB相信号线是否连接好，如果运动异常，可以先检查反馈有信号没有接好，没有接好会自动运动中停止。

16、设置编码器没有信号输入最大动作脉冲数（默认5000，建议不改动）

发送：

应答：

[设备地址] 0x01	[设备地址] 0x01
[命令] 0x66	[命令] 0x66
[数据长度] 0x04	[数据长度] 0x00
[数据] xx xx xx xx	[数据] 无
[CRC校验1] 0xnn	[CRC校验1] 0xnn
[CRC校验2] 0xnn	[CRC校验2] 0xnn

数据：默认5000，用来检查编码器信号线是否连接好，如果运动异常，可以先检查编码器反馈有信号没有接好，没有接好会自动运动中停止。

通讯格式:ASC (标准:使用和校验)

对于一条 RTU 协议的命令可以简单的通过以下的步骤转化为 ASCII 协议的命令:

- 1、把命令的 CRC 校验去掉,并且计算出 LRC 校验取代。
- 2、把生成的命令串的每一个字节转化成对应的两个字节的 ASCII 码,比如 0x03 转化成 0x30,0x33 (0 的 ASCII 码和 3 的 ASCII 码)。
- 3、在命令的开头加上起始标记“:”,它的 ASCII 码为 0x3A。
- 4、在命令的尾部加上结束标记 CR,LF(0xD,0xA),此处的 CR,LF 表示回车和换行的 ASCII 码。

1、ASC与RTU的基本区别

- A. 将RTU通讯格式的CRC校验改成和校验,计算出校验结果
- B. 将命令使用BCD码的格式(1个字符变为2个字符)
- C. 前面加上“:”,后面加上0x0d, 0x0a

2、ASC通讯的基本格式:

发送:地址 + 命令 + 数据长度 + 32位参数 + 和校验

地址:1-247

命令:1-127为发送命令

数据长度:从下一个字符开始统计,到结束的字节数.

参数:4个字符(32BIT)的参数

和校验:前面所有的字符和取低8位.

正确返回:地址 + **命令** + 数据长度 + 32位参数 + 和校验

地址:1-247

命令:1-127为发送命令,原命令返回表示命令成功

数据长度:从下一个字符开始统计,到结束的字节数(和校验:+1, CRC:

+2).

参数:最长4个字符(32BIT)的参数

和校验:前面所有的字符和取低8位.

失败返回:地址 + 命令 + 数据长度 + 8位参数 + 和校验

地址:1-247

命令:1-127为发送命令,最高位置1表示返回命令失败

数据长度:从下一个字符开始统计,到结束的字节数(和校验: +1, CRC: +2) .

参数:返回失败的错误代码

0X01: 表示接收超时(数据长度不对)

0x02: 表示校验错误

0x03: 该命令不存在

0X04: 表示参数超出范围

和校验:前面所有的字符和取低8位.

3、基本通讯协议如下:

(具体协议内容参见RTU通讯协议,以下例子举例协议转换过程)

例如:更改地址

更改地址:地址数据1个字节,不支持广播

发送:(RTU)

应答:(RTU)

[设备地址] 0x01

[设备地址] 0x02

[命令] 0x22

[命令] 0x22

[数据长度] 0x01

[数据长度] 0x01

[数据] 0X02 (新地址)

[数据] 0X02 (新地址)

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

////////////////////////////////////

更改地址:地址数据1个字节,不支持广播

发送:(ASC)

应答:(ASC)

[设备地址] 0x01

[设备地址] 0x01

[命令] 0x22

[命令] 0x22

[数据长度] 0x01

[数据长度] 0x01

[数据] 0X02 (新地址)

[数据] 0X02 (新地址)

[LRC校验] 0xnn

[LRC校验] 0xnn

ASC通讯过程

ASC格式发送为:

‘:’ + ‘0’ +‘1’ +‘2’ +‘2’ +‘0’ +‘1’ +‘0’ +‘2’ +‘n’ +‘n’ +0x0d+0x0a

0x01 0x22 0x01 0x02 (LRC) 0xnn

ASC格式应答为:

‘:’ + ‘0’ +‘2’ +‘2’ +‘2’ +‘0’ +‘1’ +‘0’ +‘2’ +‘n’ +‘n’ +0x0d +0x0a

0x02 0x22 0x01 0x02 (LRC) 0xnn

注意: 1、[设备地址][命令][数据]ASC和RTU都一样, [数据长度]ASC比RTU模式少1。

2、ASC使用LRC校验, RTU使用CRC校验。

3、‘:’ 0x0d 0x0a 这3个不加入LRC校验

CAN 通讯说明

1、同时支持CAN的标准通讯模式和扩展通讯模式，扩展ID固定为0；标准ID是设备设定的8位地址即1-250之间的地址：地址0仍然作为广播使用：251-255仍然保留。

2、CAN波特率支持： 1:1M 2:900K 3:800K 4:750K 5:600K
6:500K 7:400K 8:250K 9:200K 10:150K 11:125K 12:100K 13:80K
14:50K 15:25K 16:20K 17:10K 18:5K 默认为100k

3、CAN通讯的通讯协议内容基本和MODBUS的RTU模式通讯协议内容一致（但不带CRC校验码），但特别说明三点：

A. CAN通讯协议的数据固定长度为8个数据，不足8位的数据，可以直接在数据后面补0，直到补足8位数据为止；

B. CAN通讯的过滤条件是：

a. (标准帧)CAN标准ID = 数据第一个字符 = 设备地址

b. (扩展帧)CAN扩展ID = 数据第一个字符 = 设备地址

c. CAN的数据长度 = 8

C. 在2毫秒时间内如果驱动器发出的数据没有被接收，自动取消发送。

4、CAN通讯调试过程可以参考通讯协议0X4C和0X4D命令，这两条命令是专门用来测试CAN通讯相关数据收发测试（IO取代命令的设置，只能在232或485中完成）。

5、实际使用举例如下：

通讯协议

更改地址：地址数据1个字节，不支持广播

发送：（RTU）

应答：（RTU）

[设备地址] 0x01

[设备地址] 0x02

[命令] 0x22

[命令] 0x22

[数据长度] 0x01

[数据长度] 0x01

[数据] 0X02（新地址）

[数据] 0X02（新地址）

[CRC校验1] 0xnn

[CRC校验1] 0xnn

[CRC校验2] 0xnn

[CRC校验2] 0xnn

在CAN通讯的情况下：

CAN标准ID=[设备地址] 0x01

CAN扩展ID = 0

CAN的数据长度 = 8

CAN的数据 = 0x01 0x22 0x01 0X02 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

CAN通讯常用指令列表

1、通讯测试命令(按RTU格式去掉CRC校验，补足8位，就是CAN协议)

注：以RTU格式为标准，转换关系如下：

(如1：查当前速度命令)

	地址	命令	长度	数据	CRC校验放弃
RTU的数据 =	0x00	0x36	0x00		CRC1 CRC2
CAN的数据 =	0x00	0x21	0x00	0X00 0X00 0X00 0X00 0X00	
	地址	命令	长度	数据 (补5个0x00, 总长8BYTE)	

(如2：设定初速度命令)

	地址	命令	长度	数据	CRC校验放弃
CAN的数据 =	0x01	0x25	0x04	0Xnn 0Xnn 0Xnn 0Xnn	CRC1 CRC2
CAN的数据 =	0x01	0x25	0x04	0Xnn 0Xnn 0Xnn 0Xnn 0X00	
说明：	地址	命令	数据长度	数据 (速度)	(补1个0x00, 总长8BYTE)

A. 从CAN发送数据，串口检测是否接收到数据的命令：

CAN标准ID=[设备地址] 0x01

CAN扩展ID = 0

CAN的数据长度 = 8

CAN的数据 = 0x01 0x4d 0x05 0Xnn 0Xnn 0Xnn 0Xnn 0Xnn

说明： 地址 命令 数据长度 数据

串口将接受到 0x01 0x4d 0x05 0Xnn 0Xnn 0Xnn 0Xnn 0Xnn +CRC

B. 从串口发送数据，CAN检测是否接收到数据的命令：

串口发数据 = 0x01 0x4c 0x05 0Xnn 0Xnn 0Xnn 0Xnn 0Xnn +CRC

说明： 地址 命令 数据长度 数据

CAN将接收： 0x01 0x4c 0x05 0Xnn 0Xnn 0Xnn 0Xnn 0Xnn

2、查询地址 (广播命令)

CAN标准ID=[设备地址] 0x00 (地址为0)

CAN扩展ID = 0

CAN的数据长度 = 8

CAN的数据 = 0x00 0x21 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

3、其他非广播命令/返回部分看上面RTU通讯协议（RTU通讯协议为标准）

固定的部分

CAN标准ID=[设备地址] 0x01（地址非0）

CAN扩展ID = 0

CAN的数据长度 = 8

发送内容部分

握手命令

CAN的数据 = 0x01 0x20 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

查地址命令

CAN的数据 = 0x01 0x22 0x01 0X02 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据（新地址） 填0

查循波特率命令

CAN的数据 = 0x01 0x23 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

改波特率命令

CAN的数据 = 0x01 0x24 0x01 0X06 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据（6=9600） 填0

////////////////////////////////////

设定初速度命令

CAN的数据 = 0x01 0x25 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明： 地址 命令 数据长度 数据（速度） 填0

查初速度命令

CAN的数据 = 0x01 0x26 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

设定最大速度命令

CAN的数据 = 0x01 0x27 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明： 地址 命令 数据长度 数据（速度） 填0

查最大速度命令

CAN的数据 = 0x01 0x28 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

设定末速度命令

CAN的数据 = 0x01 0x29 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明： 地址 命令 数据长度 数据（速度） 填0

查末速度命令

CAN的数据 = 0x01 0x2a 0x00 0X00 0X00 0X00 0X00 0X00

说明: 地址 命令 数据长度 数据 填0

设定加速度命令

CAN的数据 = 0x01 0x2b 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (速度) 填0

查加速度命令

CAN的数据 = 0x01 0x2c 0x00 0X00 0X00 0X00 0X00 0X00

说明: 地址 命令 数据长度 数据 填0

设定减速度命令

CAN的数据 = 0x01 0x2d 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (速度) 填0

查减速度命令

CAN的数据 = 0x01 0x2e 0x00 0X00 0X00 0X00 0X00 0X00

说明: 地址 命令 数据长度 数据 填0

设定归零速度命令

CAN的数据 = 0x01 0x2f 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (速度) 填0

查归零速度命令

CAN的数据 = 0x01 0x30 0x00 0X00 0X00 0X00 0X00 0X00

说明: 地址 命令 数据长度 数据 填0

////////////////////////////////////

固定脉冲运动命令1

CAN的数据 = 0x01 0x31 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (脉冲) 填0

固定脉冲运动命令2

CAN的数据 = 0x01 0x32 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (方向+脉冲) 填0

恒速运动命令1

CAN的数据 = 0x01 0x33 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (速度) 填0

恒速运动命令2

CAN的数据 = 0x01 0x34 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (方向+速度) 填0

运动中改变速度命令

CAN的数据 = 0x01 0x35 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明: 地址 命令 数据长度 数据 (速度) 填0

自动查找原点命令

CAN的数据 = 0x01 0x3E 0x01 0Xnn 0Xnn 0Xnn 0Xnn 0X00
 说明: 地址 命令 数据长度 数据 (速度) 填0

急停命令

CAN的数据 = 0x01 0x3C 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

减速停止命令

CAN的数据 = 0x01 0x3D 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

////////////////////////////////////

查当前速度命令

CAN的数据 = 0x01 0x36 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

查绝对位置命令

CAN的数据 = 0x01 0x37 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

绝对位置清零命令

CAN的数据 = 0x01 0x38 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

查相对位置命令

CAN的数据 = 0x01 0x39 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

相对位置清零命令

CAN的数据 = 0x01 0x3A 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

查当前运动状态命令

CAN的数据 = 0x01 0x3B 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

查询IN1-5状态命令

CAN的数据 = 0x01 0x42 0x00 0X00 0X00 0X00 0X00 0X00
 说明: 地址 命令 数据长度 数据 (新地址) 填0

以下闭环相关命令

使用前进/后退（反馈脉冲数）运动命令

CAN的数据 = 0x01 0x55 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明：地址 命令 数据长度 数据（闭环脉冲） 填0

使用电子齿轮进行运动命令

CAN的数据 = 0x01 0x56 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明：地址 命令 数据长度 数据（闭环脉冲） 填0

使用绝对位置进行运动命令

CAN的数据 = 0x01 0x57 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明：地址 命令 数据长度 数据（闭环脉冲） 填0

查闭环绝对位置命令

CAN的数据 = 0x01 0x58 0x00 0X00 0X00 0X00 0X00 0X00

说明：地址 命令 数据长度 数据 填0

设置（或清0）闭环绝对位置命令

CAN的数据 = 0x01 0x59 0/4 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明：地址 命令 数据长度 数据（位置） 填0

查闭环相对位置命令

CAN的数据 = 0x01 0x5a 0x00 0X00 0X00 0X00 0X00 0X00

说明：地址 命令 数据长度 数据 填0

设置（或清0）相环绝对位置命令

CAN的数据 = 0x01 0x5b 0/4 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明：地址 命令 数据长度 数据（位置） 填0

设置编码器反馈方向命令

CAN的数据 = 0x01 0x5c 0x01 0/1 0X00 0X00 0X00 0X00

说明：地址 命令 数据长度 数据 填0

设置运动方向命令（是否反方向运动）

CAN的数据 = 0x01 0x5d 0x01 0/1 0X00 0X00 0X00 0X00

说明：地址 命令 数据长度 数据 填0

设置是否消除累计误差命令（0消除/1不消除）

CAN的数据 = 0x01 0x5e 0x01 0/1 0X00 0X00 0X00 0X00

说明：地址 命令 数据长度 数据 填0

设置电子齿轮分子（最大16位数）

CAN的数据 = 0x01 0x5f 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明：地址 命令 数据长度 数据 填0

读取电子齿轮分子（最大16位数）

CAN的数据 = 0x01 0x60 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

设置电子齿轮分母（最大16位数）

CAN的数据 = 0x01 0x61 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明： 地址 命令 数据长度 数据 填0

读取电子齿轮分母（最大16位数）

CAN的数据 = 0x01 0x62 0x00 0X00 0X00 0X00 0X00 0X00

说明： 地址 命令 数据长度 数据 填0

设置编码器AB信号允许误差（默认200，建议不变）

CAN的数据 = 0x01 0x65 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明： 地址 命令 数据长度 数据 填0

设置编码器无信号允许运动脉冲数（默认5000，建议不变）

CAN的数据 = 0x01 0x66 0x04 0Xnn 0Xnn 0Xnn 0Xnn 0X00

说明： 地址 命令 数据长度 数据 填0

各种品牌PLC标准通讯的标准格式

模式说明 PLC为主模式，驱动器为从模式，仅支持485通讯

一、写命令格式（功能码固定为0x10）

通讯中寄存区始地址 = 驱动器映射地址 + 协议命令码.

举例说明

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x28	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x28	0x00 0x02	0xnn 0xnn

二、读命令格式（功能码固定为0x03）

举例说明

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x28	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

三、西门子寄存区起始地址映射需要注意的地方

Modbus RTU 从站地址与 S7-200的地址对应

Modbus 地址总是以00001、30004之类的形式出现。S7-200内部的数据存储区与 Modbus 的0、1、3、4共4类地址的对应关系如下:

表 1. Modbus 地址对应表

Modbus 地址	S7-200数据区
00001 ~ 00128	Q0.0 ~ Q15.7
10001 ~ 10128	I0.0 ~ I15.7
30001 ~ 30032	AIW0 ~ AIW62
40001 ~ 4xxxx	T ~ T + 2 * (xxxx - 1)

其中 T 为 S7-200 中的缓冲区起始地址，即 HoldStart。

如果已知 S7-200 中的 V 存储区地址，推算 Modbus 地址的公式如下：

Modbus 地址 = 40000 + (T/2+1) ; T 为偶数

西门子寄存器地址注意: (可能)通讯地址 = (VB地址+1)/2, 因为vb是字节,而西门子用的是字(2字节)

驱动器接收命令的寄存器地址 = (驱动器寄存器映射地址 + 0x20)

原因: 0x20是驱动器命令列表的第一命令码

通讯中地址 = 驱动器映射地址 + 协议命令码.(每一个命令对应一个地址)

四、写映射寄存器数据命令(可通过工具写入)

30、设置与西门子PLC通讯内存映射地址

通讯地址 = 映射地址 + 命令代码

发送:

应答:

[设备地址]	0x01	[设备地址]	0x01
[命令]	0x3f	[命令]	0x3f
[数据长度]	0x02	[数据长度]	0x00
[数据]	0xnn, 0xnn	[数据]	无
[CRC校验1]	0xnn	[CRC校验1]	0xnn
[CRC校验2]	0xnn	[CRC校验2]	0xnn

五、所有写命令

写地址的操作地址为的起始地址为0x20:

1、更改地址: 地址数据1个字节, 不支持广播, 需重新启动

寄存区始地址 = 映射地址+0x22

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x22	0x00 0x02	0x04	0x02 0x00 0x00 0x00 新地址 补3个0x00数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x22	0x00 0x02	0xnn 0xnn

2、改变当前波特率, 不支持广播,

寄存区始地址 = 映射地址+0x24

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x24	0x00 0x02	0x04	0x06 0x00 0x00 0x00 波特率代号 补3个0x00数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x24	0x00 0x02	0xnn 0xnn

波特率代号:

0x01: 300 0x02: 600 0x03: 1200 0x04: 2400 0x05: 4800
 0x06: 9600 0x07: 19200 0x08: 38400 0x09: 57600 (红色保留不用)

////////////////////////////////////

运动参数（初速度，加速度，最大速度，减速度，停止速度）示意图：

以下用于设置位置运动参数

3、设置运动初速度：不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x25

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x25	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 初速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x25	0x00 0x02	0xnn 0xnn

运动初速度4BYTE，数据范围：0----100K，超出范围报错0X04

4、设置运动最大速度：不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x27

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x27	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 最大速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x27	0x00 0x02	0xnn 0xnn

运动最大速度4BYTE，数据范围：0----100K，超出范围报错0X04

5、设置运动末速度：不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x29

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x29	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 末速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x29	0x00 0x02	0xnn 0xnn

运动末速度4BYTE，数据范围：0----100K，超出范围报错0X04

6、设置运动加速度：不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x2b

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x2b	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 加速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x2b	0x00 0x02	0xnn 0xnn

运动加速度4BYTE，数据范围：0----100K，超出范围报错0X04

7、设置运动减速度：不运动，不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x2d

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x2d	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 减速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x2d	0x00 0x02	0xnn 0xnn

运动减速度4BYTE，数据范围：0----100K，超出范围报错0X04

8、设置归零查找原点运动速度：不运动，不支持广播（32位脉冲统计长度）

(M4505不支持该命令)

寄存器始地址 = 映射地址+0x2f

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x2f	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 归零速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x2f	0x00 0x02	0xnn 0xnn

运动归零速度4BYTE，数据范围：0----100K，超出范围报错0X04

////////////////////////////////////

9、设置运动脉冲数并开始运动：不支持广播（32位脉冲统计长度）

方向在脉冲正负数中，正数表示正方向，负数表示负方向。

寄存区始地址 = 映射地址+0x31

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x31	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x31	0x00 0x02	0xnn 0xnn

**数据: 4BYTE 运动脉冲数4BYTE, 根据最高位是否为1来判断方向
也就是根据32位的有符号数来根据正负号来判断方向, 根据数的绝对值来判断脉冲数 (支持31位脉冲数)。**

10、设置运动脉冲数并开始运动: 不支持广播 (32位脉冲统计长度)

第一位为方向, 1正方向, 0负方向, 运动脉冲数为后3位

寄存区始地址 = 映射地址+0x32

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x32	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 方向 正脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x32	0x00 0x02	0xnn 0xnn

数据: 4BYTE, 第一位为方向, 1正方向, 0负方向, 运动脉冲数为后3位 (支持24位无符号脉冲数)

////////////////////////////////////

以下用于恒速运动

11、设置运动速度, 并恒速运动命令: 不支持广播

方向在脉冲正负数中, 正数表示正方向, 负数表示负方向。

寄存区始地址 = 映射地址+0x33

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x33	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x33	0x00 0x02	0xnn 0xnn

**数据: 4BYTE 运动脉冲数4BYTE, 根据最高位是否为1来判断方向
也就是根据32位的有符号数来根据正负号来判断方向, 根据数的绝对值来判断脉冲数 (支持31位脉冲数)。**

12、设置运动速度，并恒速运动命令：不支持广播

第一位为方向，1正方向，0负方向，运动脉冲数为后3位

寄存器始地址 = 映射地址+0x34

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x34	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 方向 正脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x34	0x00 0x02	0xnn 0xnn

数据：4BYTE，第一位为方向，1正方向，0负方向，运动脉冲数为后3位（支持24位无符号脉冲数）

13、设置运动过程中，改变速度：不支持广播 (M4505不支持该命令)

改变速度, 不改变方向。

寄存器始地址 = 映射地址+0x35

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x35	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 新速度数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x35	0x00 0x02	0xnn 0xnn

数据：4BYTE，0-100K速度数据

14、绝对位置清零：不支持广播

寄存器始地址 = 映射地址+0x38

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x38	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 无效任意数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x38	0x00 0x02	0xnn 0xnn

15、相对位置清零：不支持广播

寄存器始地址 = 映射地址+0x3A

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x3A	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x3A	0x00 0x02	0xnn 0xnn

以下命令用于停止操作

16、急停命令：支持广播

寄存器始地址 = 映射地址+0x3C

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x3C	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn
无效任意数据						

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x3C	0x00 0x02	0xnn 0xnn

17、减速停止命令：支持广播

寄存器始地址 = 映射地址+0x3D

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x3D	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn
无效任意数据						

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x3D	0x00 0x02	0xnn 0xnn

18、自动查找原点命令：(M4505不支持该命令)

寄存器始地址 = 映射地址+0x3E

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x3E	0x00 0x02	0x04	0xnn 0x00 0x00 0x00	0xnn 0xnn
方向 无效任意数据						

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x3E	0x00 0x02	0xnn 0xnn

数据：0表示从反方向开始查找，1表示从正方向开始查找，遇上限位自动反方向查找，查找到原点以后，自动将相对位置和绝对位置清零。

六、所有读命令

1、请求握手命令，不做CRC数据检查，可以参考返回的CRC校验

寄存器始地址 = 映射地址+0x20

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x20	0x00 0x00	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	CRC校验
0x01	0x03	0x00	0xnn 0xnn

2、 查询设备地址命令为0x21，该命令支持广播

寄存器始地址 = 映射地址+0x21

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x00	0x03	0x00 0x21	0x00 0x00	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	CRC校验
0x01	0x03	0x00	0xnn 0xnn

3、查询当前波特率

寄存器始地址 = 映射地址+0x23

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x23	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0x01 0xnn 0xnn 0xnn	0xnn 0xnn

波特率代号 补零

4、读取运动初速度：不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x26

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x26	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

初速度数据

运动初速度4BYTE，数据范围：0----100K，超出范围报错0X04

5、读取运动最大速度：不支持广播（32位脉冲统计长度）

寄存器始地址 = 映射地址+0x28

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x28	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验

0x01 0x03 0x04 0xnn 0xnn 0xnn 0xnn 0xnn 0xnn
最大速度数据

运动最大速度4BYTE，数据范围：0----100K，超出范围报错0X04

6、读取设置运动末速度：不支持广播（32位脉冲统计长度）

寄存区始地址 = 映射地址+0x2a

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x2a	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

末速度数据

运动末速度4BYTE，数据范围：0----100K，超出范围报错0X04

7、读取运动加速度：不支持广播（32位脉冲统计长度）

寄存区始地址 = 映射地址+0x2c

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x2c	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

加速度数据

运动加速度4BYTE，数据范围：0----100K，超出范围报错0X04

8、读取运动减速度：不运动，不支持广播（32位脉冲统计长度）

寄存区始地址 = 映射地址+0x2e

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x2e	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

减速度数据

运动减速度4BYTE，数据范围：0----100K，超出范围报错0X04

9、读取归零运动速度：不运动，不支持广播（32位脉冲统计长度）(M4505不支持该命令)

寄存区始地址 = 映射地址+0x30

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x30	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

归零速度数据

运动减速度4BYTE，数据范围：0----100K，超出范围报错0X04

////////////////////////////////////

以下用于查询运动状态

10、查询运动速度：不支持广播

寄存器始地址 = 映射地址+0x36

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x36	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

当前速度数据

数据：4BYTE运动速度4BYTE，最大不超过100K

11、查询运动绝对位置：不支持广播

寄存器始地址 = 映射地址+0x37

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x37	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

当前绝对位置数据

数据：4BYTE 绝对位置带符号的32位数

12、查询运动相对位置：不支持广播

寄存器始地址 = 映射地址+0x39

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x39	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

当前相对位置数据

数据：4BYTE 绝对位置带符号的32位数

13、查询运动状态命令：不支持广播

寄存器始地址 = 映射地址+0x3b

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x3b	0x00 0x04	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x08	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn 0xnn 0xnn

前5位有效,后面补0

数据:

[数据1]: 1表示运动中, 0表示停止

[数据2]: LL-左限位I0状态 1表示有效 0表示无效

[数据3]: 0-原点I0状态 1表示有效 0表示无效

[数据4]: RL-右限位I0状态 1表示有效 0表示无效

[数据5]: EN-使能I0状态 1表示有效 0表示无效

以下用于闭环运动控制：用于M6505闭环运动控制步进驱动器

1、设置运动反馈脉冲数并开始运动：不支持广播（32位反馈脉冲统计长度）方向在脉冲正负数中，正数表示正方向，负数表示负方向。

寄存区始地址 = 映射地址+0x55

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x55	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x55	0x00 0x02	0xnn 0xnn

数据：4BYTE 运动反馈脉冲数4BYTE，根据最高位是否为1来判断方向也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来反馈判断脉冲数（支持31位脉冲数）。

2、使用电子齿轮运算，设置运动反馈脉冲数并开始运动：不支持广播（32位反馈脉冲统计长度）方向在脉冲正负数中，正数表示正方向，负数表示负方向。

实际反馈脉冲数 = 设定反馈脉冲数 * 齿轮分子 / 齿轮分母

最高支持数据为运算后的实际反馈脉冲数为31位反馈脉冲数

寄存区始地址 = 映射地址+0x56

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x56	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x56	0x00 0x02	0xnn 0xnn

数据：4BYTE 运动反馈脉冲数4BYTE，根据最高位是否为1来判断方向也就是根据32位的有符号数来根据正负号来判断方向，根据数的绝对值来反馈判断脉冲数（支持31位设定脉冲数）。

3、设置运动反馈脉冲绝对位置并运动：不支持广播（32位反馈脉冲统计长度）方向在脉冲正负数中，正数负数表示编码器反馈的绝对位置。

寄存区始地址 = 映射地址+0x57

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x57	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x57	0x00 0x02	0xnn 0xnn

数据：4BYTE 运动反馈绝对位置数4BYTE，根据数的绝对位置值来判断是否

运动以及运动距离（支持31位编码器绝对位置数，计算后的实际运动距离最大为31位数据）。

4、读取闭环绝对位置

寄存器始地址 = 映射地址+0x58

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x58	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

数据：4BYTE运动反馈位置，编码器的反馈位置。

5、闭环绝对位置设置一个值(用于清0)

寄存器始地址 = 映射地址+0x59

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x59	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x59	0x00 0x02	0xnn 0xnn

数据：设置4BYTE运动反馈绝对位置，无数值就将该变量清零。

6、读取闭环相对位置

寄存器始地址 = 映射地址+0x5a

PLC发送:

地址	功能读	寄存器始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x5a	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

7、闭环相对位置设置一个值(用于清0)

发送：

应答：

寄存器始地址 = 映射地址+0x5b

PLC发送:

地址	功能写	寄存器始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x5b	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存器始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x5b	0x00 0x02	0xnn 0xnn

数据：设置4BYTE运动反馈相对位置，无数值就将该变量清零。

8、设置编码器方向

发送： **应答：**

寄存区始地址 = 映射地址+0x5c

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x5c	0x00 0x01	0x02	0x01/0x00 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x5C	0x00 0x01	0xnn 0xnn

数据：0编码器正方向，1编码器反方向。

9、设置运动方向

发送： **应答：**

寄存区始地址 = 映射地址+0x5d

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x5d	0x00 0x01	0x02	0x01/0x00 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x5d	0x00 0x01	0xnn 0xnn

数据：0正常运动方向，1反方向运动。

10、设置自动调整积累误差开关

发送： **应答：**

寄存区始地址 = 映射地址+0x5e

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x5e	0x00 0x01	0x02	0x01/0x00 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x5e	0x00 0x01	0xnn 0xnn

数据：0无积累误差，1有积累误差。

11、设置电子齿轮分子

发送： **应答：**

寄存区始地址 = 映射地址+0x5f

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x5f	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x5f	0x00 0x02	0xnn 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

12、读取电子齿轮分子

寄存区始地址 = 映射地址+0x60

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x60	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

13、设置电子齿轮分母

发送:

应答:

寄存区始地址 = 映射地址+0x61

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x61	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn 带正负脉冲数据	0xnn 0xnn

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x61	0x00 0x02	0xnn 0xnn

数据：16位电子齿轮数，大于16位，只取16无符号位数。

14、读取电子齿轮分母

寄存区始地址 = 映射地址+0x62

PLC发送:

地址	功能读	寄存区始地址	寄存器数量	CRC校验
0x01	0x03	0x00 0x62	0x00 0x02	0xnn 0xnn

驱动器返回

地址	功能读	数据字节数	传递数据	CRC校验
0x01	0x03	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

15、设置编码器AB相信号允许误差（默认200，建议不改动）

寄存区始地址 = 映射地址+0x65

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x65	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x65	0x00 0x02	0xnn 0xnn

数据：默认200，用来检查AB相信号线是否连接好，如果运动异常，可以先检查反馈有信号没有接好，没有接好会自动运动中停止。

16、设置编码器没有信号输入最大动作脉冲数（默认5000，建议不改动）

寄存区始地址 = 映射地址+0x66

PLC发送:

地址	功能写	寄存区始地址	寄存器数量	数据字节数	传递数据	CRC校验
0x01	0x10	0x00 0x66	0x00 0x02	0x04	0xnn 0xnn 0xnn 0xnn	0xnn 0xnn

带正负脉冲数据

驱动器返回

地址	功能写	寄存区始地址	寄存器数量	CRC校验
0x01	0x10	0x00 0x66	0x00 0x02	0xnn 0xnn

数据：默认5000，用来检查编码器信号线是否连接好，如果运动异常，可以先检查编码器反馈有信号没有接好，没有接好会自动运动中停止。