

伺服电机控制协议

适用驱动：V3

版本：V4.3

日期：2025.05

免责声明

感谢您购买脉塔智能电机驱动系统。在使用之前，请仔细阅读本声明，一旦使用，即被视为对本声明全部内容的认可和接受。请严格遵守手册、产品说明和相关的法律法规、政策、准则安装和使用该产品。在使用产品过程中，用户承诺对自己的行为及因此而产生的所有后果负责。因用户不当使用、安装、改装造成的任何损失，脉塔智能将不承担法律责任。

脉塔智能是苏州脉塔智能科技有限公司及其关联公司的商标。本文出现的产品名称、品牌等，均为其所属公司的商标或注册商标。

本产品及手册为脉塔智能版权所有。未经许可，不得以任何形式复制翻印。关于免责声明的最终解释权，归脉塔智能所有。



目录

伺服电机控制协议	1
免责声明	I
1. 通讯总线参数及报文格式	1
1.1. CAN 总线	1
1.1.1. 参数	1
1.1.2. 报文格式	1
1.2. RS485 总线	1
1.2.1. 参数	1
1.2.2. 报文格式	1
2. 单电机命令说明	2
2.1. 读取 PID 参数命令（0x30）	2
2.1.1. 指令说明	2
2.1.2. 发送数据域定义	2
2.1.3. 回复数据域定义	2
2.1.4. 功能索引说明	2
2.1.5. 通讯示例	3
2.2. 写入 PID 参数到 RAM 命令（0x31）	4
2.2.1. 指令说明	4
2.2.2. 发送数据域定义	4
2.2.3. 回复数据域定义	4
2.2.4. 功能索引说明	4
2.2.5. 通讯示例	5
2.3. 写入 PID 参数到 ROM 命令（0x32）	5
2.3.1. 指令说明	5
2.3.2. 发送数据域定义	5
2.3.3. 回复数据域定义	6
2.3.4. 功能索引说明	6
2.3.5. 通讯示例	6
2.4. 读取加速度命令（0x42）	7
2.4.1. 指令说明	7
2.4.2. 发送数据域定义	7
2.4.3. 回复数据域定义	7
2.4.4. 功能索引说明	8

2.4.5. 通讯示例	8
2.5. 写入加减速度到 RAM 和 ROM 命令 (0x43)	10
2.5.1. 指令说明	10
2.5.2. 发送数据域定义	10
2.5.3. 回复数据域定义	10
2.5.4. 功能索引说明	10
2.5.5. 通讯示例	10
2.6. 读取多圈编码器位置数据命令 (0x60)	13
2.6.1. 指令说明	13
2.6.2. 发送数据域定义	13
2.6.3. 回复数据域定义	14
2.6.4. 通讯示例	14
2.7. 读取多圈编码器原始位置数据命令 (0x61)	15
2.7.1. 指令说明	15
2.7.2. 发送数据域定义	15
2.7.3. 回复数据域定义	15
2.7.4. 通讯示例	16
2.8. 读取多圈编码器零偏数据命令 (0x62)	17
2.8.1. 指令说明	17
2.8.2. 发送数据域定义	17
2.8.3. 回复数据域定义	17
2.8.4. 通讯示例	18
2.9. 写入编码器多圈值到 ROM 作为电机零点命令 (0x63)	18
2.9.1. 指令说明	18
2.9.2. 发送数据域定义	19
2.9.3. 回复数据域定义	19
2.9.4. 通讯示例	19
2.10. 写入编码器当前多圈位置到 ROM 作为电机零点命令 (0x64)	20
2.10.1. 指令说明	20
2.10.2. 发送数据域定义	20
2.10.3. 回复数据域定义	20
2.10.4. 通讯示例	21
2.11. 读取单圈编码器命令 (0x90)	21
2.11.1. 指令说明	21

2.11.2. 发送数据域定义	22
2.11.3. 回复数据域定义	22
2.11.4. 通讯示例	22
2.12. 读取多圈角度命令 (0x92)	23
2.12.1. 指令说明	23
2.12.2. 发送数据域定义	23
2.12.3. 回复数据域定义	24
2.12.4. 通讯示例	24
2.13. 读取单圈角度命令 (0x94)	25
2.13.1. 指令说明	25
2.13.2. 发送数据域定义	25
2.13.3. 回复数据域定义	25
2.13.4. 通讯示例	26
2.14. 读取电机状态 1 和错误标志命令 (0x9A)	27
2.14.1. 指令说明	27
2.14.2. 发送数据域定义	27
2.14.3. 回复数据域定义	27
2.14.4. 通讯示例	28
2.15. 读取电机状态 2 命令 (0x9C)	29
2.15.1. 指令说明	29
2.15.2. 发送数据域定义	29
2.15.3. 回复数据域定义	30
2.15.4. 通讯示例	30
2.16. 读取电机状态 3 命令 (0x9D)	31
2.16.1. 指令说明	31
2.16.2. 发送数据域定义	31
2.16.3. 回复数据域定义	31
2.16.4. 通讯示例	32
2.17. 电机关闭命令 (0x80)	33
2.17.1. 指令说明	33
2.17.2. 发送数据域定义	33
2.17.3. 回复数据域定义	33
2.18. 电机停止命令 (0x81)	33
2.18.1. 指令说明	33

2.18.2. 发送数据域定义	34
2.18.3. 回复数据域定义	34
2.19. 转矩闭环控制命令（0xA1）	34
2.19.1. 指令说明	34
2.19.2. 发送数据域定义	34
2.19.3. 回复数据域定义	35
2.19.4. 通讯示例	35
2.20. 速度闭环控制命令（0xA2）	37
2.20.1. 指令说明	37
2.20.2. 发送数据域定义	37
2.20.3. 回复数据域定义	38
2.20.4. 通讯示例	38
2.21. 绝对位置闭环控制命令（0xA4）	40
2.21.1. 指令说明	40
2.21.2. 发送数据域定义	41
2.21.3. 回复数据域定义	41
2.21.4. 通讯示例	42
2.22. 单圈位置控制命令（0xA6）	44
2.22.1. 指令说明	44
2.22.2. 发送数据域定义	44
2.22.3. 回复数据域定义	45
2.22.4. 通讯示例	45
2.23. 增量位置闭环控制命令（0xA8）	47
2.23.1. 指令说明	47
2.23.2. 发送数据域定义	47
2.23.3. 回复数据域定义	48
2.23.4. 通讯示例	48
2.24. 力控位置闭环控制命令（0xA9）	50
2.24.1. 指令说明	50
2.24.2. 发送数据域定义	51
2.24.3. 回复数据域定义	51
2.24.4. 通讯示例	51
2.25. 系统运行模式获取（0x70）	54
2.25.1. 指令说明	54

2.25.2. 发送数据域定义	54
2.25.3. 回复数据域定义	54
2.25.4. 通讯示例	55
2.26. 系统复位指令（0x76）	55
2.26.1. 指令说明	55
2.26.2. 发送数据域定义	55
2.26.3. 回复数据域定义	56
2.26.4. 通讯示例	56
2.27. 系统抱闸释放指令（0x77）	56
2.27.1. 指令说明	56
2.27.2. 发送数据域定义	56
2.27.3. 回复数据域定义	57
2.28. 系统抱闸锁死指令（0x78）	57
2.28.1. 指令说明	57
2.28.2. 发送数据域定义	57
2.28.3. 回复数据域定义	57
2.29. 系统运行时间读取指令（0xB1）	57
2.29.1. 指令说明	57
2.29.2. 发送数据域定义	57
2.29.3. 回复数据域定义	58
2.29.4. 通讯示例	58
2.30. 系统软件版本日期读取指令（0xB2）	59
2.30.1. 指令说明	59
2.30.2. 发送数据域定义	59
2.30.3. 回复数据域定义	59
2.30.4. 通讯示例	60
2.31. 通讯中断保护时间设置指令（0xB3）	61
2.31.1. 指令说明	61
2.31.2. 发送数据域定义	61
2.31.3. 回复数据域定义	61
2.31.4. 通讯示例	61
2.32. 通讯波特率设置指令（0xB4）	63
2.32.1. 指令说明	63
2.32.2. 发送数据域定义	63

2.32.3. 回复数据域定义	63
2.32.4. 通讯示例	63
2.33. 电机型号读取指令 (0xB5)	64
2.33.1. 指令说明	64
2.33.2. 发送数据域定义	64
2.33.3. 回复数据域定义	65
2.33.4. 通讯示例	65
2.34. 主动回复功能指令 (0xB6)	66
2.34.1. 指令说明	66
2.34.2. 发送数据域定义	66
2.34.3. 回复数据域定义	66
2.34.4. 通讯示例	66
2.35. 功能控制指令 (0x20)	67
2.35.1. 指令说明	67
2.35.2. 发送数据域定义	67
2.35.3. 回复数据域定义	67
2.35.4. 功能索引说明	67
2.35.5. 通讯示例	68
3. CAN 多电机命令 (0x280+指令)	70
3.1. 指令说明	70
3.2. 通讯示例	70
4. CANID 设置指令 (0x79)	72
4.1. 指令说明	72
4.2. 发送数据域定义	72
4.3. 回复数据域定义	72
4.4. 通讯示例	73
5. 运动模式控制指令_CAN (0x400 + ID)	74
5.1. 指令说明	74
5.2. 发送数据域定义	75
5.3. 回复数据域定义	75
5.4. 通讯示例	76
6. RS485 多电机命令 (0xCD + 指令)	78
6.1. 指令说明	78
6.2. 通讯示例	78

7. RS485-ID 设置指令（0x79）	80
7.1. 指令说明	80
7.2. 发送数据域定义	80
7.3. 回复数据域定义	80
7.4. 通讯示例	81
8. 指示灯说明	82
8.1. 状态说明	82
8.2. 故障说明表格	82
9. 版本修订信息	83



1. 通讯总线参数及报文格式

1.1. CAN 总线

1.1.1. 参数

总线接口：CAN

波特率：1Mbps

1.1.2. 报文格式

标识符：单电机指令发送：0x140 + ID(1~32)

多电机指令发送：0x280

回复：0x240 + ID(1~32)

帧格式：数据帧

帧类型：标准帧

DLC：8 字节

1.2. RS485 总线

1.2.1. 参数

总线接口：RS485

波特率：115200bps, 500Kbps, 1Mbps, 1.5Mbps, 2.5Mbps

串口配置：8 位数据位、1 位停止位、无奇偶校验位

1.2.2. 报文格式

类型	数据定义	字节数	说明
帧头	0x3E	1	通讯帧头，用于识别。
ID 号	1~32	1	设备地址，对应每个电机的 ID 号。
数据长度	数据长度	1	数据域的长度，标准协议中长度固定 8 个字节。
数据域	数据内容	根据长度	标准协议中与 CAN 的数据域内容完全一致。
校验	CRC 校验	2	CRC16 校验，低位在前，高位在后。

2. 单电机命令说明

2.1. 读取 PID 参数命令 (0x30)

2.1.1. 指令说明

该指令可以读取电流环、速度环和位置环的 PID 参数，数据类型为 Float，通过索引值来确定，具体见 2.1.4 索引说明表。

2.1.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x30
DATA[1]	参数索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.1.3. 回复数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x30
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	参数低字节 1	DATA[4] = (uint8_t)(Value)
DATA[5]	参数字节 2	DATA[5] = (uint8_t)(Value>>8)
DATA[6]	参数字节 3	DATA[6] = (uint8_t)(Value>>16)
DATA[7]	参数字节 4	DATA[7] = (uint8_t)(Value>>24)

2.1.4. 功能索引说明

索引值	参数名称
0x01	电流环 KP

0x02	电流环 KI
0x04	速度环 KP
0x05	速度环 KI
0x07	位置环 KP
0x08	位置环 KI
0x09	位置环 KD

2.1.5. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x30	0x01	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x30	0x01	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: Data[1] = 0x01, 按照索引值表格, 代表电流环 KP。表示读取电流环 KP 参数。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x30	0x01	0x00	0x00	0x00	0x00	0x80	0x3F

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x30	0x01	0x00	0x00	0x00	0x00	0x80	0x3F	CRC16L	CRC16H

说明: 回复的帧数据中 Data[1] = 0x01, 按照索引值表格, 代表电流环 KP ,

Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x3F

80 00 00, 数据类型为 Float, 转换为十进制小数为 1.0 (可使用在线转换网站: 在线进制转换-IEE754 浮点数 16 进制转换 (speedfly.cn)), 代表电机当前电流环 KP

参数为 1.0。

2.2. 写入 PID 参数到 RAM 命令 (0x31)

2.2.1. 指令说明

该指令可以写入电流环、速度环和位置环的 PID 参数到 RAM 中，掉电后不保存，数据类型为 Float，通过索引值来确定，具体见 2.2.4 索引说明表。注意避免在电机刚启动以及运动时写入参数。

2.2.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x31
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	参数低字节 1	DATA[4] = (uint8_t)(Value)
DATA[5]	参数字节 2	DATA[5] = (uint8_t) (Value>>8)
DATA[6]	参数字节 3	DATA[6] = (uint8_t)(Value>>16)
DATA[7]	参数字节 4	DATA[7] = (uint8_t)(Value>>24)

2.2.3. 回复数据域定义

回复数据内容和发送数据一致。

2.2.4. 功能索引说明

索引值	参数名称
0x01	电流环 KP
0x02	电流环 KI
0x04	速度环 KP
0x05	速度环 KI
0x07	位置环 KP
0x08	位置环 KI
0x09	位置环 KD

2.2.5. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x31	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x31	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F	CRC16L	CRC16H

说明: Data[1] = 0x01, 按照索引值表格, 代表参数是电流环 KP; Data[4]到 data[7] 组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x 3F C0 00 00, 数据类型为 Float, 转换为十进制小数为 1.5 (可使用在线转换网站: 在线进制转换-IEE754 浮点数 16 进制转换 (speedfly.cn))。表示将电机的电流环 KP 参数设定为 1.5 写入到电机驱动的 RAM, 断电后参数不保存。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x31	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x31	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F	CRC16L	CRC16H

2.3. 写入 PID 参数到 ROM 命令 (0x32)

2.3.1. 指令说明

该指令可以写入电流环、速度环和位置环的 PID 参数到 ROM 中, 掉电后可以保存, 数据类型为 Float, 通过索引值来确定, 具体见 2.2.4 索引说明表。注意避免在电机刚启动以及运动时写入参数。

2.3.2. 发送数据域定义

数据域	说明	数据
-----	----	----

DATA[0]	命令字节	0x32
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	参数低字节 1	DATA[4] = (uint8_t)(Value)
DATA[5]	参数字节 2	DATA[5] = (uint8_t) (Value>>8)
DATA[6]	参数字节 3	DATA[6] = (uint8_t)(Value>>16)
DATA[7]	参数字节 4	DATA[7] = (uint8_t)(Value>>24)

2.3.3. 回复数据域定义

回复数据内容和发送数据一致。

2.3.4. 功能索引说明

索引值	参数名称
0x01	电流环 KP
0x02	电流环 KI
0x04	速度环 KP
0x05	速度环 KI
0x07	位置环 KP
0x08	位置环 KI
0x09	位置环 KD

2.3.5. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x32	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x32	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F	CRC16L	CRC16H

说明：Data[1] = 0x01，按照索引值表格，代表参数是电流环 KP；Data[4]到 data[7] 组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x 3F C0 00 00，数据类型为 Float，转换为十进制小数为 1.5（可使用在线转换网站：在线进制转换-IEE754 浮点数 16 进制转换（speedfly.cn））。表示将电机的电流环 KP 参数设定为 1.5 写入到电机驱动的 ROM，断电后参数可以保存。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x32	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x32	0x01	0x00	0x00	0x00	0x00	0xC0	0x3F	CRC16L	CRC16H

2.4. 读取加速度命令 (0x42)

2.4.1. 指令说明

主机发送该命令读取当前电机的的加速度参数

2.4.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x42
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.4.3. 回复数据域定义

驱动回复数据中包含了加速度参数。加速度数据 Accel 为 int32_t 类型，单位 1dps/s，参数范围 100-60000。

数据域	说明	数据
DATA[0]	命令字节	0x42
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	加速度低字节 1	DATA[4] = (uint8_t)(Accel)
DATA[5]	加速度字节 2	DATA[5] = (uint8_t)(Accel>>8)
DATA[6]	加速度字节 3	DATA[6] = (uint8_t)(Accel>>16)
DATA[7]	加速度字节 4	DATA[7] = (uint8_t)(Accel>>24)

2.4.4. 功能索引说明

索引值	指令名称	功能说明
0x00	位置规划加速度	位置规划中初始速度到最大速度的加速度值
0x01	位置规划减速度	位置规划中从最大速度到停止的减速度值
0x02	速度规划加速度	从当前速度加速到目标速度的加速度值，包括正反方向加速度
0x03	速度规划减速度	在相同方向上，从当前速度减速到目标速度的减速度值

2.4.5. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x42	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x42	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：发送指令读取位置规划加速度。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x42	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x42	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[1]为 0x00，表示位置规划加速度值。Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。代表电机位置环运行时加速度为 10000dps/s。

示例 2:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x42	0x01	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x42	0x01	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：发送指令读取位置规划减速度。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x42	0x01	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x42	0x01	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[1]为 0x01，表示位置规划减速度值。Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。代表电机位置环运行时减速度为 10000dps/s。

2.5. 写入加减速度到 RAM 和 ROM 命令 (0x43)

2.5.1. 指令说明

主机发送该命令写入加减速度到 RAM 和 ROM 中，掉电后可以保存。加速度数据 Accel 为 uint32_t 类型，单位 1dps/s，参数范围 100-60000。命令包含了位置和速度规划中的加速度和减速度值，通过索引值来确定，具体见 2.5.4 索引说明表。注意避免在电机刚启动以及运动时写入参数。

2.5.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x43
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	加速度低字节 1	DATA[4] = (uint8_t)(Accel)
DATA[5]	加速度字节 2	DATA[5] = (uint8_t)(Accel>>8)
DATA[6]	加速度字节 3	DATA[6] = (uint8_t)(Accel>>16)
DATA[7]	加速度字节 4	DATA[7] = (uint8_t)(Accel>>24)

2.5.3. 回复数据域定义

电机在收到命令后回复主机，回复命令和接收命令一致。

2.5.4. 功能索引说明

索引值	指令名称	功能说明
0x00	位置规划加速度	位置规划中初始速度到最大速度的加速度值
0x01	位置规划减速度	位置规划中从最大速度到停止的减速度值
0x02	速度规划加速度	从当前速度加速到目标速度的加速度值，包括正反方向加速度
0x03	速度规划减速度	在相同方向上，从当前速度减速到目标速度的减速度值

2.5.5. 通讯示例

示例 1:

发送指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x43	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[1]为 0x00，表示位置规划加速度值。Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。表示写入 10000dps/s 的位置规划加速度到电机驱动器，断电后值可以保存。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x43	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：电机在收到命令后回复主机，回复命令和接收命令一致。

示例 2:

发送指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x43	0x01	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x01	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[1]为 0x01，表示位置规划减速度值。Data[4]到 data[7]组成一个

(Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00002710, 表示十进制为 10000。表示写入 10000dps/s 的位置规划减速度到电机驱动器, 断电后值可以保存。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x43	0x01	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x01	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: 电机在收到命令后回复主机, 回复命令和接收命令一致。

示例 3:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x43	0x02	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x02	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: Data[1]为 0x02, 表示速度规划加速度值。Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00002710, 表示十进制为 10000。表示写入 10000dps/s 的速度规划加速度到电机驱动器, 断电后值可以保存。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x43	0x02	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
----	------	----	----	----	----	----	----	----	----	----	--------	--------

0x3E	0x01	0x08	0x43	0x02	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H
------	------	------	------	------	------	------	------	------	------	------	--------	--------

说明：电机在收到命令后回复主机，回复命令和接收命令一致。

示例 4：

发送指令：

CAN：

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x43	0x03	0x00	0x00	0x10	0x27	0x00	0x00

RS485：

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x03	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[1]为 0x03，表示速度规划减速度值。Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。表示写入 10000dps/s 的速度规划减速度到电机驱动器，断电后值可以保存。

回复指令：

CAN：

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x43	0x03	0x00	0x00	0x10	0x27	0x00	0x00

RS485：

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x43	0x03	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：电机在收到命令后回复主机，回复命令和接收命令一致。

2.6. 读取多圈编码器位置数据命令（0x60）

2.6.1. 指令说明

主机发送该命令以读取编码器多圈的位置，代表了电机转动过的编码器多圈值。

2.6.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x60

DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.6.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。编码器多圈位置 `encoder`（`int32_t` 类型，多圈编码器的数值范围，有效数据 4 个字节），为编码器原始位置减去编码器多圈零偏（初始位置）后的值。

数据域	说明	数据
DATA[0]	命令字节	0x60
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	编码器位置低字节 1	$\text{DATA}[4] = (\text{uint8_t})(\text{encoder})$
DATA[5]	编码器位置字节 2	$\text{DATA}[5] = (\text{uint8_t})(\text{encoder} \gg 8)$
DATA[6]	编码器位置字节 3	$\text{DATA}[6] = (\text{uint8_t})(\text{encoder} \gg 16)$
DATA[7]	编码器位置字节 4	$\text{DATA}[7] = (\text{uint8_t})(\text{encoder} \gg 24)$

2.6.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
----	------	----	----	----	----	----	----	----	----	----	--------	--------

0x3E	0x01	0x08	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H
------	------	------	------	------	------	------	------	------	------	------	--------	--------

说明：主机发送该命令以读取编码器多圈的位置。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x60	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x60	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 10000 个脉冲。

2.7. 读取多圈编码器原始位置数据命令（0x61）

2.7.1. 指令说明

主机发送该命令以读取多圈编码器原始位置，即没有包含零偏（初始位置）的多圈编码器值。

2.7.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x61
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.7.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。编码器多圈原始位置

encoderRaw (int32_t 类型, 数值范围, 有效数据 4 个字节)。

数据域	说明	数据
DATA[0]	命令字节	0x61
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	编码器原始位置字节 1	DATA[4] = (uint8_t)(encoderRaw)
DATA[5]	编码器原始位置字节 2	DATA[5] = (uint8_t)(encoderRaw>>8)
DATA[6]	编码器原始位置字节 3	DATA[6] = (uint8_t)(encoderRaw>>16)
DATA[7]	编码器原始位置字节 4	DATA[7] = (uint8_t)(encoderRaw>>24)

2.7.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x61	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x61	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 主机发送该命令以读取编码器多圈的原始位置。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x61	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x61	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为

0x00002710，表示十进制为 10000，代表电机当前多圈编码器值为 10000 个脉冲，不包括零偏（初始位置）的。

2.8. 读取多圈编码器零偏数据命令（0x62）

2.8.1. 指令说明

主机发送该命令以读取编码器多圈的零偏值（初始位置）。

2.8.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x62
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.8.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。编码器多圈零偏 encoderOffset（int32_t 类型，数值范围，有效数据 4 个字节）。

数据域	说明	数据
DATA[0]	命令字节	0x62
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	编码器零偏字节 1	DATA[4] = (uint8_t)(encoderOffset)
DATA[5]	编码器零偏字节 2	DATA[5] = (uint8_t)(encoderOffset>>8)
DATA[6]	编码器零偏字节 3	DATA[6] = (uint8_t)(encoderOffset>>16)
DATA[7]	编码器零偏字节 4	DATA[7] = (uint8_t)(encoderOffset>>24)

2.8.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x62	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x62	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 主机发送该命令以读取编码器多圈的零偏值。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x62	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x62	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00002710, 表示十进制为 10000。代表电机当前多圈编码器零偏值为 10000 个脉冲。

2.9. 写入编码器多圈值到 ROM 作为电机零点命令 (0x63)

2.9.1. 指令说明

主机发送该命令以设置编码器的零偏 (初始位置), 其中, 需要写入的编码器多圈值 encoderOffset 为 int32_t 类型, (数值范围, 有效数据 4 个字节)。注意避免在电机刚启动以及运动时写入参数。

注意: 写入新零点的位置后需要重新启动电机才会有效。因为零偏的改变, 设置目标位置时应以新的零偏 (初始位置) 为参考。

2.9.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x63
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	编码器零偏低字节 1	DATA[4] = (uint8_t)(encoderOffset)
DATA[5]	编码器零偏字节 2	DATA[5] = (uint8_t)(encoderOffset>>8)
DATA[6]	编码器零偏字节 3	DATA[6] = (uint8_t)(encoderOffset>>8)
DATA[7]	编码器零偏字节 4	DATA[7] = (uint8_t)(encoderOffset>>8)

2.9.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据和主机发送的命令相同。

2.9.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x63	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x63	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00002710, 表示十进制为 10000。表示写入 10000 个脉冲作为多圈编码器零偏。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x63	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x63	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：电机在收到命令后回复主机，该帧数据和主机发送的命令相同。

2.10. 写入编码器当前多圈位置到 ROM 作为电机零点命令 (0x64)

2.10.1. 指令说明

将电机当前编码器位置作为多圈编码器零偏（初始位置）写入到 ROM，注意避免在电机刚启动以及运动时写入参数。

注意：写入后新的零点位置后需要发送 0x76（系统复位指令）重启系统后才会有效。因为零偏的改变，设置目标位置时应以新的零偏（初始位置）为参考。

2.10.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x64
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.10.3. 回复数据域定义

电机在收到命令后回复主机，数据中 encoderOffset 为设置的零偏值。

数据域	说明	数据
DATA[0]	命令字节	0x64
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	编码器零偏低字节 1	DATA[4] = (uint8_t)(encoderOffset)

DATA[5]	编码器零偏字节 2	DATA[5] = (uint8_t)(encoderOffset>>8)
DATA[6]	编码器零偏字节 3	DATA[6] = (uint8_t)(encoderOffset>>16)
DATA[7]	编码器零偏字节 4	DATA[7] = (uint8_t)(encoderOffset>>24)

2.10.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x64	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x64	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 发送 0x64 指令后, 电机会将当前多圈编码器值作为零偏 (初始位置) 写入 ROM 中。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x64	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x64	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00002710, 表示十进制为 10000。表示写入电机的多圈零偏值 (初始位置) 为 10000 个脉冲。

2.11. 读取单圈编码器命令 (0x90)

2.11.1. 指令说明

主机发送该命令以读取编码器的当前位置, 注意当前指令是作为单圈数据读取指令,

针对直驱电机使用。

2.11.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x90
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.11.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 编码器位置 `encoder`，为编码器原始位置减去编码器零偏后的值。
2. 编码器原始位置 `encoderRaw`。
3. 编码器零偏 `encoderOffset`，该点作为电机角度的零点。

数据域	说明	数据
DATA[0]	命令字节	0x90
DATA[1]	NULL	0x00
DATA[2]	编码器位置低字节	$DATA[1] = (uint8_t)(encoder)$
DATA[3]	编码器位置高字节	$DATA[2] = (uint8_t)(encoder >> 8)$
DATA[4]	编码器原始位置低字节	$DATA[3] = (uint8_t)(encoderRaw)$
DATA[5]	编码器原始位置高字节	$DATA[4] = (uint8_t)(encoderRaw >> 8)$
DATA[6]	编码器零偏低字节	$DATA[5] = (uint8_t)(encoderOffset)$
DATA[7]	编码器零偏高字节	$DATA[6] = (uint8_t)(encoderOffset >> 8)$

2.11.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x90	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x90	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 发送 0x90 指令后, 将返回电机单圈编码器值。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x90	0x00	0x33	0x08	0xBE	0x2C	0x8B	0x24

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x90	0x00	0x33	0x08	0xBE	0x2C	0x8B	0x24	CRC16L	CRC16H

说明: Data[2]到 data[3]组成一个 (Data[2]为最低位, Data[3]为最高位) 16 位数据为 0x0833, 表示十进制为 2099, 表示电机当前相对零偏的编码器位置是 2099 个脉冲。

Data[4]到 data[5]组成一个 (Data[4]为最低位, Data[5]为最高位) 16 位数据为 0x2CBE, 表示十进制为 11454, 表示电机当前的编码器原始位置是 11454 个脉冲。Data[6]到 data[7]组成一个 (Data[6]为最低位, Data[7]为最高位) 16 位数据为 0x248B, 表示十进制为 9355, 表示电机的零偏位置是 9355 个脉冲。

2.12. 读取多圈角度命令 (0x92)

2.12.1. 指令说明

主机发送该命令以读取当前电机的多圈绝对角度值。

2.12.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x92

DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.12.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机输出轴角度 `motorAngle`，(int32_t 类型，数值范围，有效数据 4 个字节)，单位 $0.01^{\circ}/\text{LSB}$ 。

数据域	说明	数据
DATA[0]	命令字节	0x92
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	角度低字节 1	$\text{DATA}[4] = (\text{uint8_t})(\text{motorAngle})$
DATA[5]	角度字节 2	$\text{DATA}[5] = (\text{uint8_t})(\text{motorAngle} \gg 8)$
DATA[6]	角度字节 3	$\text{DATA}[6] = (\text{uint8_t})(\text{motorAngle} \gg 16)$
DATA[7]	角度字节 4	$\text{DATA}[7] = (\text{uint8_t})(\text{motorAngle} \gg 24)$

2.12.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x92	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
----	------	----	----	----	----	----	----	----	----	----	--------	--------

0x3E	0x01	0x08	0x92	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H
------	------	------	------	------	------	------	------	------	------	------	--------	--------

说明：发送 0x92 指令后，将返回电机输出轴绝对角度。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x92	0x00	0x00	0x00	0xA0	0x8C	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x92	0x00	0x00	0x00	0xA0	0x8C	0x00	0x00	CRC16L	CRC16H

说明：Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00008CA0，表示十进制为 36000，按照 0.01° /LSB 单位缩小 100 倍即 36000*0.01=360°。表示电机输出轴相对零点位置正向移动 360°。

2.13. 读取单圈角度命令（0x94）

2.13.1. 指令说明

主机发送该命令以读取当前电机的单圈角度。注意当前指令是作为单圈数据读取指令，针对直驱电机使用。

2.13.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x94
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.13.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1.电机单圈角度 circleAngle, 为 uint16_t 类型数据, 以编码器零点为起始点, 顺时针增加, 再次到达零点时数值回 0, 单位 0.01°/LSB, 数值范围 0~35999。

数据域	说明	数据
DATA[0]	命令字节	0x94
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	单圈角度低字节	DATA[5] = (uint8_t)(circleAngle)
DATA[7]	单圈角度高字节	DATA[6] = (uint8_t)(circleAngle>>8)

2.13.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x94	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x94	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 发送 0x94 指令后, 将返回电机单圈角度。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x94	0x00	0x00	0x00	0x00	0x00	0x10	0x27

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x94	0x00	0x00	0x00	0x00	0x00	0x10	0x27	CRC16L	CRC16H

说明：Data[6]到 data[7]组成一个（Data[6]为最低位，Data[7]为最高位）16 位数据为 0x2710，表示十进制为 10000，单位是 0.01°，表示电机当前在相对零点位置 100°位置。

2.14. 读取电机状态 1 和错误标志命令（0x9A）

2.14.1. 指令说明

该命令读取当前电机的温度、电压和错误状态标志

2.14.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x9A
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.14.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据包含了以下参数：

- 1.电机温度 temperature（int8_t 类型，单位 1°C/LSB）；
- 2.抱闸控制指令：表示抱闸控制指令状态，1 代表抱闸释放指令，0 代表抱闸锁死指令；
- 3.电压 voltage（uint16_t 类型，单位 0.1V/LSB）；
- 4.错误标志 errorState（为 uint16_t 类型，各个位代表不同的电机状态）。

数据域	说明	数据
DATA[0]	命令字节	0x9A
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	MOS 温度	DATA[2] = (uint8_t)(motorMOStemperature)
DATA[3]	抱闸释放指令	DATA[3] = (uint8_t)(RlyCtrlRslt)

DATA[4]	电压低字节	DATA[4] = (uint8_t)(voltage)
DATA[5]	电压高字节	DATA[5] = (uint8_t)(voltage>>8)
DATA[6]	错误状态低字节 1	DATA[6] = (uint8_t)(errorState)
DATA[7]	错误状态字节 2	DATA[7] = (uint8_t)(errorState>>8)

备注:

1.系统异常状态值 System_errorState 状态表 1 如下:

System_errorState 值	状态说明
0x0002	电机堵转
0x0004	低压
0x0008	过压
0x0010	相电流过流
0x0040	功率超限
0x0080	标定参数写入错误
0x0100	超速
0x0800	元器件过温
0x1000	电机温度过温
0x2000	编码器校准错误
0x4000	编码器数据错误

2.多个错误同时出现时,错误状态位会叠加显示。例如出现 0x0016 的数字,则表示 0x2+0x4+0x10 相加,也就是此时有电机堵转、低压、相电流过流三个错误出现。

2.14.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x9A	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
----	------	----	----	----	----	----	----	----	----	----	--------	--------

0x3E	0x01	0x08	0x9A	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H
------	------	------	------	------	------	------	------	------	------	------	--------	--------

说明：发送 0x9A 令后，将返回电机的温度、电压和错误状态标志。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x9A	0x32	0x00	0x01	0xE5	0x01	0x04	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x9A	0x32	0x00	0x01	0xE5	0x01	0x04	0x00	CRC16L	CRC16H

说明：Data[1]=0x32 十进制为 50，代表此刻电机温度为 50 度。Data[3]表示抱闸表示抱闸控制指令状态，1 代表抱闸释放指令，0 代表抱闸锁死指令，所以 0x01 表示当前抱闸释放指令已经执行。Data[4]和 Data[5]（Data[4]为低位，Data[5]为高位）组成 0x01E5，十进制为 485，按照 0.1V/LSB 的单位缩小 10 倍， $485 \times 0.1 = 48.5V$ ，代表当前电机供电电压为 48.5V。Data[6]和 Data[7]（Data[6]为低位，Data[7]为高位）组成 0x0004，对照 System_errorState 表中的错误说明，表示低压错误。

2.15. 读取电机状态 2 命令 (0x9C)

2.15.1. 指令说明

该命令读取当前电机的温度、转速、编码器位置。

2.15.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x9C
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00

DATA[7]	NULL	0x00
---------	------	------

2.15.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

- 1.电机温度 temperature (int8_t 类型, 1°C/LSB);
- 2.电机的转矩电流值 iq (int16_t 类型, 0.01A/LSB);
- 3.电机输出轴转速 speed (int16_t 类型, 1dps/LSB);
- 4.电机输出轴角度 (int16_t 类型, 1degree/LSB,最大范围±32767degree)。

数据域	说明	数据
DATA[0]	命令字节	0x9C
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	电机角度低字节	DATA[6] = (uint8_t)(degree)
DATA[7]	电机角度高字节	DATA[7] = (uint8_t)(degree>>8)

2.15.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x9C	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x9C	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 该命令读取当前电机的温度、转速、编码器位置。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x9C	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x9C	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100, 按照 100 倍比例缩小即为 $100 \times 0.01 = 1A$, 那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500, 代表电机输出轴转速为 500dps。电机输出轴转速和电机转速之间存在减速比的关系, 如果减速比为 6, 那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0x002D 十进制为 45, 代表电机输出轴相对零点位置正向移动 45 度。电机输出轴位置和电机编码器线数和减速比有关, 例如电机编码器线数为 16384, 减速比为 6, 那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

2.16. 读取电机状态 3 命令 (0x9D)

2.16.1. 指令说明

该命令读取当前电机的温度和相电流数据。

2.16.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x9D
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.16.3. 回复数据域定义

电机在收到命令后回复主机, 该帧数据包含了以下数据:

- 1.电机温度 temperature (int8_t 类型, 1℃/LSB)
- 2.A 相电流数据, 数据类型为 int16_t 类型, 对应实际相电流为 0.01ALSB。
- 3.B 相电流数据, 数据类型为 int16_t 类型, 对应实际相电流为 0.01ALSB。
- 4.C 相电流数据, 数据类型为 int16_t 类型, 对应实际相电流为 0.01ALSB。

数据域	说明	数据
DATA[0]	命令字节	0x9D
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	A 相电流低字节	DATA[2] = (uint8_t)(iA)
DATA[3]	A 相电流高字节	DATA[3] = (uint8_t)(iA>>8)
DATA[4]	B 相电流低字节	DATA[4] = (uint8_t)(iB)
DATA[5]	B 相电流高字节	DATA[5] = (uint8_t)(iB>>8)
DATA[6]	C 相电流低字节	DATA[6] = (uint8_t)(iC)
DATA[7]	C 相电流高字节	DATA[7] = (uint8_t)(iC>>8)

2.16.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x9D	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x9D	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 该命令读取当前电机的温度和相电流数据。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x9D	0x32	0xC2	0x0B	0x10	0xFA	0xC0	0xF9

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x9D	0x32	0xC2	0x0B	0x10	0xFA	0xC0	0xF9	CRC16L	CRC16H

说明：Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0BC2 十进制为 3010，按照 100 倍比例缩小即为 $3010 \times 0.01 = 30.1A$ ，那么代表当前电机 A 相实际电流为 30.1A。Data[4]和 Data[5]合成数据 0xFA10 十进制为 -1520，按照 100 倍比例缩小即为 $-1520 \times 0.01 = -15.2A$ ，那么代表当前电机 B 相实际电流为 -15.2A。Data[6]和 Data[7]合成数据 0xF9C0 十进制为 -1600，按照 100 倍比例缩小即为 $-1600 \times 0.01 = -16A$ ，那么代表当前电机 C 相实际电流为 -16A。

2.17. 电机关闭命令 (0x80)

2.17.1. 指令说明

关闭电机输出，同时清除电机运行状态，不在任何闭环模式下。

2.17.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x80
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.17.3. 回复数据域定义

电机在收到命令后回复主机，帧数据和主机发送相同。

2.18. 电机停止命令 (0x81)

2.18.1. 指令说明

停止电机，将电机速度停下来，并使电机保持不动，不会因为外力移动。

2.18.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x81
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.18.3. 回复数据域定义

电机在收到命令后回复主机，帧数据和主机发送相同。

2.19. 转矩闭环控制命令 (0xA1)

2.19.1. 指令说明

该指令为控制指令，在电机没有故障的情况下可以运行该指令。主机发送该命令以控制电机的转矩电流输出，控制值 iqControl 为 int16_t 类型，单位为 0.01A/LSB。

出于安全考虑，抱闸款电机无法直接使用此指令，若使用 0xA1 指令，首先需要通过 0x77 指令打开抱闸。

2.19.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xA1
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	转矩电流控制值低字节	DATA[4] = (uint8_t)(iqControl)
DATA[5]	转矩电流控制值高字节	DATA[5] = (uint8_t)(iqControl>>8)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.19.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

- 1.电机温度 temperature (int8_t 类型, 1°C/LSB);
- 2.电机的转矩电流值 iq (int16_t 类型, 0.01A/LSB);
- 3.电机输出轴转速 speed (int16_t 类型, 1dps/LSB);
- 4.电机输出轴角度 (int16_t 类型, 1degree/LSB,最大范围±32767degree)。

数据域	说明	数据
DATA[0]	命令字节	0xA1
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	电机角度低字节	DATA[6] = (uint8_t)(degree)
DATA[7]	电机角度高字节	DATA[7] = (uint8_t)(degree>>8)

2.19.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA1	0x00	0x00	0x00	0x64	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA1	0x00	0x00	0x00	0x64	0x00	0x00	0x00	CRC16L	CRC16H

说明: Data[4]和 data[5]代表数据大小, Data[4] (0x64) 为低位, Data[5] (0x00) 为高位。所以实际数据为 0x0064 表示十进制 100, 按照 0.01A/LSB 缩小即为 100*0.01=1A。驱动会以 1A 作为目标电流执行。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA1	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA1	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100, 按照 100 倍比例缩小即为 $100 \times 0.01 = 1A$, 那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500, 代表电机输出轴转速为 500dps。电机输出轴转速和电机转速之间存在减速比的关系, 如果减速比为 6, 那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0x002D 十进制为 45, 代表电机输出轴相对零点位置正向移动 45 度。电机输出轴位置和电机编码器线数和减速比有关, 例如电机编码器线数为 16384, 减速比为 6, 那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

示例 2:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA1	0x00	0x00	0x00	0x9C	0xFF	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA1	0x00	0x00	0x00	0x9C	0xFF	0x00	0x00	CRC16L	CRC16H

说明: Data[4]和 data[5]代表数据大小, Data[4] (0x9C) 为低位, Data[5] (0xFF) 为高位。所以实际数据为 0xFF9C 表示十进制 -100, 按照 0.01A/LSB 缩小即为 $-100 \times 0.01 = -1A$ 。驱动会以 -1A 作为目标电流执行。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
------	---------	---------	---------	---------	---------	---------	---------	---------

0x241	0xA1	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF
-------	------	------	------	------	------	------	------	------

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA1	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF	CRC16L	CRC16H

说明：Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0xFF9C 十进制为-100，按照 100 倍比例缩小即为-100*0.01=-1A，那么代表当前电机实际电流为-1A。Data[4]和 Data[5]合成数据 0xFE0C 十进制为-500,代表电机输出轴转速为-500dps。电机输出轴转速和电机转速之间存在减速比的关系，如果减速比为 6，那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0xFFD3 十进制为-45，代表电机输出轴相对零点位置反向移动-45 度。电机输出轴位置和电机编码器线数和减速比有关，例如电机编码器线数为 16384，减速比为 6，那么电机输出轴的 360 度对应 $16384*6 = 98304$ 个脉冲。

2.20. 速度闭环控制命令 (0xA2)

2.20.1. 指令说明

该指令为控制指令，在电机没有故障的情况下可以运行该指令。主机发送该命令以控制电机输出轴的速度，控制值 speedControl 为 int32_t 类型，对应实际转速为 0.01dps/LSB，控制值 maxTorque 限制了电机输出轴的最大扭矩，为 uint8_t 类型，取值范围为 0~255，以额定电流的百分比为单位，即 $1\% * \text{额定电流} / \text{LSB}$ 。若给定的电流为 0 或者大于堵转电流，则不开启力控模式，电机的最大转矩电流由上位机中的电机堵转电流值限制。

2.20.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xA2
DATA[1]	最大扭矩	DATA[2] = (uint8_t)(maxTorque)
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	速度控制低字节	DATA[4] = (uint8_t)(speedControl)
DATA[5]	速度控制	DATA[5] = (uint8_t)(speedControl>>8)

DATA[6]	速度控制	DATA[6] = (uint8_t)(speedControl>>16)
DATA[7]	速度控制高字节	DATA[7] = (uint8_t)(speedControl>>24)

备注:

- 1.该命令下电机的最大转矩电流由上位机中的电机堵转电流值限制。
- 2.该控制模式下，电机的最大加速度由上位机中的速度环加速度值限制。
- 3.当速度环加速度值为 0 时，速度环加速度由电流最大输出能力限制。

2.20.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 temperature (int8_t 类型，1°C/LSB)；
- 2.电机的转矩电流值 iq (int16_t 类型，0.01A/LSB)；
- 3.电机输出轴转速 speed (int16_t 类型，1dps/LSB)；
- 4.电机输出轴角度 (int16_t 类型，1degree/LSB,最大范围±32767degree)。

数据域	说明	数据
DATA[0]	命令字节	0xA2
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	电机角度低字节	DATA[6] = (uint8_t)(degree)
DATA[7]	电机角度高字节	DATA[7] = (uint8_t)(degree>>8)

2.20.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA2	0x00	0x00	0x00	0x10	0x27	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA2	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明：Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。发送指令按照 0.01dps/LSB 缩小 100 倍，即 $10000 \times 0.01 = 100\text{dps}$ 。驱动以电机输出轴 100dps 的速度为目标速度运行。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA2	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA2	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00	CRC16L	CRC16H

说明：Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100，按照 100 倍比例缩小即为 $100 \times 0.01 = 1\text{A}$ ，那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500，代表电机输出轴转速为 500dps。电机输出轴转速和电机转速之间存在减速比的关系，如果减速比为 6，那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0x002D 十进制为 45，代表电机输出轴相对零点位置正向移动 45 度。电机输出轴位置和电机编码器线数和减速比有关，例如电机编码器线数为 16384，减速比为 6，那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

示例 2:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA2	0x00	0x00	0x00	0xF0	0xD8	0xFF	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA2	0x00	0x00	0x00	0xF0	0xD8	0xFF	0xFF	CRC16L	CRC16H

说明：Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0xFFFFD8F0，表示十进制为-10000。发送指令按照 0.01dps/LSB 缩小 100 倍，即-10000*0.01=-100dps。驱动以电机输出轴-100dps 的速度为目标速度运行。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA2	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA2	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF	CRC16L	CRC16H

说明：Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0xFF9C 十进制为-100，按照 100 倍比例缩小即为-100*0.01=-1A，那么代表当前电机实际电流为-1A。Data[4]和 Data[5]合成数据 0xFE0C 十进制为-500,代表电机输出轴转速为-500dps。电机输出轴转速和电机转速之间存在减速比的关系，如果减速比为 6，那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0xFFD3 十进制为-45，代表电机输出轴相对零点位置反向移动-45 度。电机输出轴位置和电机编码器线数和减速比有关，例如电机编码器线数为 16384，减速比为 6，那么电机输出轴的 360 度对应 $16384*6 = 98304$ 个脉冲。

2.21. 绝对位置闭环控制命令 (0xA4)

2.21.1. 指令说明

该指令为控制指令，在电机没有故障的情况下可以运行该指令。主机发送该命令以控制电机的位置（多圈角度），控制值 angleControl 为 int32_t 类型，对应实际位置为 0.01degree/LSB，即 36000 代表 360°，电机转动方向由目标位置和当前位置的差值决定。控制值 maxSpeed 限制了电机输出轴转动的最大速度，为 uint16_t 类型，对应实际转速 1dps/LSB。

根据系统设置的位置规划加速度值的不同运行模式会有区别：

1. 如果位置环加速度为 0，那么位置环将进入直接跟踪模式，通过 PI 控制器直接跟踪目标位置。其中 maxSpeed 限制了位置运行过程中的最大速度，如果 maxSpeed

值为 0，那么就完全由 PI 控制器计算结果输出。如下图所示。

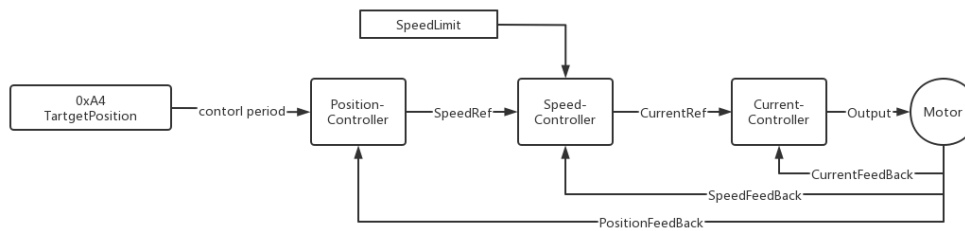


图 2-1 带速度限制的位置跟踪模式框图

2. 如果位置环加速度不为 0，那么将运行带速度规划的运动模式，由电机来完成加速减速过程。其中的最大运行速度由 `maxSpeed` 决定，加速度由位置环设置的加速度决定。

2.21.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xA4
DATA[1]	NULL	0x00
DATA[2]	速度限制低字节	DATA[2] = (uint8_t)(maxSpeed)
DATA[3]	速度限制高字节	DATA[3] = (uint8_t)(maxSpeed>>8)
DATA[4]	位置控制低字节	DATA[4] = (uint8_t)(angleControl)
DATA[5]	位置控制	DATA[5] = (uint8_t)(angleControl>>8)
DATA[6]	位置控制	DATA[6] = (uint8_t)(angleControl>>16)
DATA[7]	位置控制高字节	DATA[7] = (uint8_t)(angleControl>>24)

2.21.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

- 1.电机温度 `temperature` (`int8_t` 类型, 1°C/LSB);
- 2.电机的转矩电流值 `iq` (`int16_t` 类型, 0.01A/LSB);
- 3.电机输出轴转速 `speed` (`int16_t` 类型, 1dps/LSB);
- 4.电机输出轴角度 (`int16_t` 类型, 1degree/LSB,最大范围±32767degree)。

数据域	说明	数据
DATA[0]	命令字节	0xA4
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)

DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	电机角度低字节	DATA[6] = (uint8_t)(degree)
DATA[7]	电机角度高字节	DATA[7] = (uint8_t)(degree>>8)

2.21.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA4	0x00	0xF4	0x01	0xA0	0x8C	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA4	0x00	0xF4	0x01	0xA0	0x8C	0x00	0x00	CRC16L	CRC16H

说明: Data[2]和 Data[3]组成一个(Data[2]为低位, Data[3]为高位)16 位数据为 0x01F4, 表示十进制 500dps 电机输出轴速度。驱动会以这个速度为最大速度运行位置环。

Data[4]到 data[7]组成一个(Data[4]为最低位, Data[7]为最高位)32 位数据为 0x00008CA0, 表示十进制为 36000。发送指令按照 0.01degree/LSB 缩小 100 倍, 即 $36000 \times 0.01 = 360^\circ$ 。电机将以输出轴相对零点位置正向移动 360° 。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA4	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA4	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]

合成数据 0x0064 十进制为 100，按照 100 倍比例缩小即为 $100 \times 0.01 = 1A$ ，那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500，代表电机输出轴转速为 500dps。电机输出轴转速和电机转速之间存在减速比的关系，如果减速比为 6，那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0x002D 十进制为 45，代表电机输出轴相对零点位置正向移动 45 度。电机输出轴位置和电机编码器线数和减速比有关，例如电机编码器线数为 16384，减速比为 6，那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

示例 2:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA4	0x00	0xF4	0x01	0x60	0x73	0xFF	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA4	0x00	0xF4	0x01	0x60	0x73	0xFF	0xFF	CRC16L	CRC16H

说明: Data[2]和 Data[3]组成一个(Data[2]为低位, Data[3]为高位)16 位数据为 0x01F4, 表示十进制 500dps 电机输出轴速度。驱动会以这个速度为最大速度运行位置环。Data[4]到 data[7]组成一个(Data[4]为最低位, Data[7]为最高位)32 位数据为 0xFFFF7360, 表示十进制为-36000。发送指令按照 0.01degree/LSB 缩小 100 倍, 即 $-36000 \times 0.01 = -360^\circ$ 。电机将以输出轴相对零点位置反向移动-360°。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA4	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA4	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]

合成数据 0xFF9C 十进制为-100，按照 100 倍比例缩小即为 $-100 \times 0.01 = -1A$ ，那么代表当前电机实际电流为-1A。Data[4]和 Data[5]合成数据 0xFE0C 十进制为-500,代表电机输出轴转速为-500dps。电机输出轴转速和电机转速之间存在减速比的关系，如果减速比为 6，那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0xFFD3 十进制为-45，代表电机输出轴相对零点位置反向移动-45 度。电机输出轴位置和电机编码器线数和减速比有关，例如电机编码器线数为 16384，减速比为 6，那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

2.22. 单圈位置控制命令 (0xA6)

2.22.1. 指令说明

主机发送该命令以控制电机的位置（单圈角度）。在多圈保存功能关闭时，默认为单圈模式。该指令可在单圈模式下使用，主要应用在直驱电机上。

1.角度控制值 angleControl 为 uint16_t 类型，数值范围 0~35999，对应实际位置为 0.01degree/LSB，即实际角度范围 $0^\circ \sim 359.99^\circ$ ；

2.spinDirection 设置电机转动的方向，为 uint8_t 类型，0x00 代表顺时针，0x01 代表逆时针；

3.maxSpeed 限制了电机转动的最大速度，为 uint16_t 类型，对应实际转速 1dps/LSB。

2.22.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xA6
DATA[1]	转动方向字节	DATA[1] = spinDirection
DATA[2]	速度限制低字节	DATA[2] = (uint8_t)(maxSpeed)
DATA[3]	速度限制高字节	DATA[3] = (uint8_t)(maxSpeed>>8)
DATA[4]	位置控制低字节	DATA[4] = (uint8_t)(angleControl)
DATA[5]	位置控制高字节	DATA[5] = (uint8_t)(angleControl>>8)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.22.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

- 1.电机温度 temperature (int8_t 类型, 1°C/LSB);
- 2.电机的转矩电流值 iq (int16_t 类型, 0.01A/LSB);
- 3.电机输出轴转速 speed (int16_t 类型, 1dps/LSB);
- 4.编码器位置值 encoder (uint16_t 类型, 编码器的数值范围由编码器位数决定)。

数据域	说明	数据
DATA[0]	命令字节	0xA6
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	编码器值低字节	DATA[6] = (uint8_t)(encoder)
DATA[7]	编码器值高字节	DATA[7] = (uint8_t)(encoder>>8)

2.22.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA6	0x00	0xF4	0x01	0xA0	0x8C	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA6	0x00	0xF4	0x01	0xA0	0x8C	0x00	0x00	CRC16L	CRC16H

说明: Data[1]为 0, 表示电机将按照顺时针方向旋转。Data[2]和 Data[3]组成一个 (Data[2]为低位, Data[3]为高位) 16 位数据为 0x01F4, 表示十进制 500dps 电机速度。驱动会以这个速度为最大速度运行位置环。Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x8CA0, 表示十进制为 36000, 单位是

0.01degree。电机以顺时针方向移动到 360°。单圈位置中 360 度和 0 度位置重合，所以此时位置也可能为 0 度。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA6	0x32	0x64	0x00	0xF4	0x01	0xE8	0x03

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA6	0x32	0x64	0x00	0xF4	0x01	0xE8	0x03	CRC16L	CRC16H

说明：Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100，按照 100 倍比例缩小即为 100*0.01=1A，那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500，代表电机转速为 500dps。Data[6]和 Data[7]合成数据 0x03E8 十进制为 1000，代表电机编码器相对于零点位置的值是 1000 个脉冲。

示例 2:

发送指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA6	0x01	0xF4	0x01	0xA0	0x8C	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA6	0x01	0xF4	0x01	0xA0	0x8C	0x00	0x00	CRC16L	CRC16H

说明：Data[1]为 1，表示电机将按照逆时针方向旋转。Data[2]和 Data[3]组成一个（Data[2]为低位，Data[3]为高位）16 位数据为 0x01F4，表示十进制 500dps 电机速度。驱动会以这个速度为最大速度运行位置环。Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x8CA0，表示十进制为 36000，单位是 0.01degree。电机以逆时针方向移动到 360°。单圈位置中 360 度和 0 度位置重合，所以此时位置也可能为 0 度。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA6	0x32	0x64	0x00	0xF4	0x01	0xE8	0x03

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA6	0x32	0x64	0x00	0xF4	0x01	0xE8	0x03	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100, 按照 100 倍比例缩小即为 $100 \times 0.01 = 1A$, 那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500, 代表电机转速为 500dps。Data[6]和 Data[7]合成数据 0x03E8 十进制为 1000, 代表电机编码器相对于零点位置的值是 1000 个脉冲。

2.23. 增量位置闭环控制命令 (0xA8)

2.23.1. 指令说明

该指令为控制指令, 在电机没有故障的情况下可以运行该指令。主机发送该命令以控制电机的增量位置 (多圈角度), 以当前位置为起点运行输入的位置增量。控制值 angleControl 为 int32_t 类型, 对应实际位置为 0.01degree/LSB, 即 36000 代表 360°, 电机转动方向由增量位置符号决定。

控制值 maxSpeed 限制了电机输出轴转动的最大速度, 为 uint16_t 类型, 对应实际转速 1dps/LSB。

2.23.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xA8
DATA[1]	NULL	0x00
DATA[2]	速度限制低字节	DATA[2] = (uint8_t)(maxSpeed)
DATA[3]	速度限制高字节	DATA[3] = (uint8_t)(maxSpeed >> 8)
DATA[4]	位置控制低字节	DATA[4] = (uint8_t)(angleControl)
DATA[5]	位置控制	DATA[5] = (uint8_t)(angleControl >> 8)

DATA[6]	位置控制	DATA[6] = (uint8_t)(angleControl>>16)
DATA[7]	位置控制高字节	DATA[7] = (uint8_t)(angleControl>>24)

2.23.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

- 1.电机温度 temperature (int8_t 类型, 1°C/LSB);
- 2.电机的转矩电流值 iq (int16_t 类型, 0.01A/LSB);
- 3.电机输出轴转速 speed (int16_t 类型, 1dps/LSB);
- 4.电机输出轴角度 (int16_t 类型, 1degree/LSB,最大范围±32767degree)。

数据域	说明	数据
DATA[0]	命令字节	0xA8
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	电机角度低字节	DATA[6] = (uint8_t)(degree)
DATA[7]	电机角度高字节	DATA[7] = (uint8_t)(degree>>8)

2.23.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA8	0x00	0xF4	0x01	0xA0	0x8C	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA8	0x00	0xF4	0x01	0xA0	0x8C	0x00	0x00	CRC16L	CRC16H

说明: Data[2]和 Data[3]组成一个(Data[2]为低位, Data[3]为高位)16 位数据为 0x01F4, 表示十进制 500dps 电机输出轴速度。驱动会以这个速度为最大速度运行位置环。

Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00008CA0, 表示十进制为 36000。发送指令按照 0.01degree/LSB 缩小 100 倍, 即 $36000 \times 0.01 = 360^\circ$ 。电机将以输出轴相对当前位置正向移动 360° 。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA8	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA8	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100, 按照 100 倍比例缩小即为 $100 \times 0.01 = 1A$, 那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500, 代表电机输出轴转速为 500dps。电机输出轴转速和电机转速之间存在减速比的关系, 如果减速比为 6, 那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0x002D 十进制为 45, 代表电机输出轴相对零点位置正向移动 45 度。电机输出轴位置和电机编码器线数和减速比有关, 例如电机编码器线数为 16384, 减速比为 6, 那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

示例 2:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA8	0x00	0xF4	0x01	0x60	0x73	0xFF	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA8	0x00	0xF4	0x01	0x60	0x73	0xFF	0xFF	CRC16L	CRC16H

说明: Data[2]和 Data[3]组成一个 (Data[2]为低位, Data[3]为高位) 16 位数据为 0x01F4, 表示十进制 500dps 电机输出轴速度。驱动会以这个速度为最大速度运行位置环。

Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0xFFFF7360，表示十进制为-36000。发送指令按照 0.01degree/LSB 缩小 100 倍，即 $-36000 \times 0.01 = -360^\circ$ 。电机将以输出轴相对当前位置反向移动-360°。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA8	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA8	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF	CRC16L	CRC16H

说明：Data[1] = 0x32 十进制为 50，代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0xFF9C 十进制为-100，按照 100 倍比例缩小即为 $-100 \times 0.01 = -1A$ ，那么代表当前电机实际电流为-1A。Data[4]和 Data[5]合成数据 0xFE0C 十进制为-500，代表电机输出轴转速为-500dps。电机输出轴转速和电机转速之间存在减速比的关系，如果减速比为 6，那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0xFFD3 十进制为-45，代表电机输出轴相对零点位置反向移动-45 度。电机输出轴位置和电机编码器线数和减速比有关，例如电机编码器线数为 16384，减速比为 6，那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

2.24. 力控位置闭环控制命令 (0xA9)

2.24.1. 指令说明

该指令为控制指令，在电机没有故障的情况下可以运行该指令。主机发送该命令以控制电机的位置（多圈角度），控制值 angleControl 为 int32_t 类型，对应实际位置为 0.01degree/LSB，即 36000 代表 360° ，电机转动方向由目标位置和当前位置的差值决定。控制值 maxSpeed 限制了电机输出轴转动的最大速度，为 uint16_t 类型，对应实际转速 1dps/LSB。控制值 maxTorque 限制了电机输出轴的最大扭矩，为 uint8_t 类型，取值范围为 0~255，以额定电流的百分比为单位，即 $1\% \times \text{额定电流} / \text{LSB}$ 。若给定的电流大于堵转电流，则不开启力控模式，电机的最大转矩电流由上位机中的电机堵转电流值限制。

2.24.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xA9
DATA[1]	最大扭矩	DATA[2] = (uint8_t)(maxTorque)
DATA[2]	速度限制低字节	DATA[2] = (uint8_t)(maxSpeed)
DATA[3]	速度限制高字节	DATA[3] = (uint8_t)(maxSpeed>>8)
DATA[4]	位置控制低字节	DATA[4] = (uint8_t)(angleControl)
DATA[5]	位置控制	DATA[5] = (uint8_t)(angleControl>>8)
DATA[6]	位置控制	DATA[6] = (uint8_t)(angleControl>>16)
DATA[7]	位置控制高字节	DATA[7] = (uint8_t)(angleControl>>24)

2.24.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据中包含了以下参数。

- 1.电机温度 temperature (int8_t 类型, 1°C/LSB);
- 2.电机的转矩电流值 iq (int16_t 类型, 0.01A/LSB);
- 3.电机输出轴转速 speed (int16_t 类型, 1dps/LSB);
- 4.电机输出轴角度 (int16_t 类型, 1degree/LSB,最大范围±32767degree)。

数据域	说明	数据
DATA[0]	命令字节	0xA9
DATA[1]	电机温度	DATA[1] = (uint8_t)(temperature)
DATA[2]	转矩电流低字节	DATA[2] = (uint8_t)(iq)
DATA[3]	转矩电流高字节	DATA[3] = (uint8_t)(iq>>8)
DATA[4]	电机速度低字节	DATA[4] = (uint8_t)(speed)
DATA[5]	电机速度高字节	DATA[5] = (uint8_t)(speed>>8)
DATA[6]	电机角度低字节	DATA[6] = (uint8_t)(degree)
DATA[7]	电机角度高字节	DATA[7] = (uint8_t)(degree>>8)

2.24.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA9	0x3C	0xF4	0x01	0xA0	0x8C	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA9	0x3C	0xF4	0x01	0xA0	0x8C	0x00	0x00	CRC16L	CRC16H

说明: Data[1]为一个 8 位数据 0x3C,表示十进制 $60 \times 1\%$ 额定电流, Data[2]和 Data[3]组成一个 (Data[2]为低位, Data[3]为高位) 16 位数据为 0x01F4, 表示十进制 500dps 电机输出轴速度, 驱动会以 60% 额定扭矩为最大扭矩运、以 500dps 为最大速度运行位置环。Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0x00008CA0, 表示十进制为 36000。发送指令按照 0.01degree/LSB 缩小 100 倍, 即 $36000 \times 0.01 = 360^\circ$ 。电机以输出轴相对零点位置正向移动 360° 。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA9	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA9	0x32	0x64	0x00	0xF4	0x01	0x2D	0x00	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0x0064 十进制为 100, 按照 100 倍比例缩小即为 $100 \times 0.01 = 1A$, 那么代表当前电机实际电流为 1A。Data[4]和 Data[5]合成数据 0x01F4 十进制为 500, 代表电机输出轴转速为 500dps。电机输出轴转速和电机转速之间存在减速比的关系, 如果减速比为 6, 那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0x002D 十进制为 45, 代表电机输出轴相对零点位置正向移动 45 度。电机输出轴位置和电机编码器线数和减速比有关, 例如电机编码器线数为 16384, 减速比为 6, 那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

示例 2:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xA9	0x3C	0xF4	0x01	0x60	0x73	0xFF	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA9	0x3C	0xF4	0x01	0x60	0x73	0xFF	0xFF	CRC16L	CRC16H

说明: Data[1]为一个 8 位数据 0x3C,表示十进制 $60 \times 1\%$ 额定电流, Data[2]和 Data[3]组成一个 (Data[2]为低位, Data[3]为高位) 16 位数据为 0x01F4, 表示十进制 500dps 电机输出轴速度, 驱动会以 60% 额定扭矩为最大扭矩运、以 500dps 为最大速度运行位置环。Data[4]到 data[7]组成一个 (Data[4]为最低位, Data[7]为最高位) 32 位数据为 0xFFFF7360, 表示十进制为-36000。发送指令按照 0.01degree/LSB 缩小 100 倍, 即 $-36000 \times 0.01 = -360^\circ$ 。电机以输出轴相对零点位置反向移动 -360° 。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xA9	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xA9	0x32	0x9C	0xFF	0x0C	0xFE	0xD3	0xFF	CRC16L	CRC16H

说明: Data[1] = 0x32 十进制为 50, 代表此刻电机温度为 50 度。Data[2]和 Data[3]合成数据 0xFF9C 十进制为-100, 按照 100 倍比例缩小即为 $-100 \times 0.01 = -1A$, 那么代表当前电机实际电流为-1A。Data[4]和 Data[5]合成数据 0xFE0C 十进制为-500, 代表电机输出轴转速为-500dps。电机输出轴转速和电机转速之间存在减速比的关系, 如果减速比为 6, 那么电机转速比输出轴转速高 6 倍。Data[6]和 Data[7]合成数据 0xFFD3 十进制为-45, 代表电机输出轴相对零点位置反向移动-45 度。电机输出轴位置和电机编码器线数和减速比有关, 例如电机编码器线数为 16384, 减速比为 6, 那么电机输出轴的 360 度对应 $16384 \times 6 = 98304$ 个脉冲。

2.25. 系统运行模式获取 (0x70)

2.25.1. 指令说明

该命令读取当前电机运行模式。

2.25.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x70
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.25.3. 回复数据域定义

电机在收到命令后回复主机，驱动回复数据中包涵了参数 runmode 运行状态,为 uint8_t 类型。

电机运行模式有以下 3 种状态：

- 1.电流环模式(0x01);
- 2.速度环模式(0x02);
- 3.位置环模式(0x03)。

数据域	说明	数据
DATA[0]	命令字节	0x70
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00

DATA[7]	电机运行模式	DATA[7] = (uint8_t)(runmode)
---------	--------	------------------------------

2.25.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x70	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x70	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 该命令读取当前电机运行模式。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x70	0x00	0x00	0x00	0x00	0x00	0x00	0x03

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x70	0x00	0x00	0x00	0x00	0x00	0x00	0x03	CRC16L	CRC16H

说明: Data[7] = 0x03, 按照回复帧定义, 表示当前系统处于位置环模式。

2.26. 系统复位指令 (0x76)

2.26.1. 指令说明

该命令用于复位系统程序。

2.26.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x76
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00

DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.26.3. 回复数据域定义

电机收到指令后会复位，不再返回指令。

2.26.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x76	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x76	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：发送指令后，系统复位，程序重新运行。

2.27. 系统抱闸释放指令（0x77）

2.27.1. 指令说明

该命令用于开启系统抱闸。系统会松开抱闸，电机处于可运动状态不受抱闸制动器限制。

2.27.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x77
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00

DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.27.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据和主机发送的命令相同。

2.28. 系统抱闸锁死指令 (0x78)

2.28.1. 指令说明

该命令用于关闭系统抱闸。抱闸会锁住电机，此时电机无法再运行。系统断电后抱闸制动器也是处于这个状态。

2.28.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x78
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.28.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据和主机发送的命令相同。

2.29. 系统运行时间读取指令 (0xB1)

2.29.1. 指令说明

该命令用于获取系统运行时间，单位 ms。

2.29.2. 发送数据域定义

数据域	说明	数据
-----	----	----

DATA[0]	命令字节	0xB1
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.29.3. 回复数据域定义

电机在收到命令后回复主机，驱动回复数据中包涵了系统已运行时间 SysRunTime，为 uint32_t 类型，单位为单位 ms。

数据域	说明	数据
DATA[0]	命令字节	0xB1
DATA[0]	NULL	0x00
DATA[0]	NULL	0x00
DATA[0]	NULL	0x00
DATA[4]	SysRunTime 低字节 1	DATA[4] = (uint8_t)(SysRunTime)
DATA[5]	SysRunTime 字节 2	DATA[5] = (uint8_t)(SysRunTime>>8)
DATA[6]	SysRunTime 字节 3	DATA[6] = (uint8_t)(SysRunTime>>16)
DATA[7]	SysRunTime 字节 4	DATA[7] = (uint8_t)(SysRunTime>>24)

2.29.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
----	------	----	----	----	----	----	----	----	----	----	--------	--------

0x3E	0x01	0x08	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H
------	------	------	------	------	------	------	------	------	------	------	--------	--------

说明：该命令读取当前系统已运行的时间。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x10

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x10	CRC16L	CRC16H

说明：Data[4]到 Data[7]（Data[4]为低位，Data[7]为高位）组成 = 0x10000000，十进制位 268435456，表示系统再重启或者复位后已经运行了 268435456ms，大概是 74 个小时。

2.30. 系统软件版本日期读取指令（0xB2）

2.30.1. 指令说明

该命令用于获取系统软件版本更新日期。

2.30.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xB2
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.30.3. 回复数据域定义

电机在收到命令后回复主机，驱动回复数据中包涵了系统软件最新版本日期

VersionDate,为 uint32_t 类型,日期格式按照年月日格式, 如 20211126。

数据域	说明	数据
DATA[0]	命令字节	0xB2
DATA[0]	NULL	0x00
DATA[0]	NULL	0x00
DATA[0]	NULL	0x00
DATA[4]	VersionDate 低字节 1	DATA[4] = (uint8_t)(VersionDate)
DATA[5]	VersionDate 字节 2	DATA[5] = (uint8_t)(VersionDate>>8)
DATA[6]	VersionDate 字节 3	DATA[6] = (uint8_t)(VersionDate>>16)
DATA[7]	VersionDate 字节 4	DATA[7] = (uint8_t)(VersionDate>>24)

2.30.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB2	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB2	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 该命令读取当前软件版本日期。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xB2	0x00	0x00	0x00	0x2E	0x89	0x34	0x01

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB2	0x00	0x00	0x00	0x2E	0x89	0x34	0x01	CRC16L	CRC16H

说明: Data[4]到 Data[7] (Data[4]为低位, Data[7]为高位) 组成 = 0x0134892E, 十

进制位 20220206，表示软件版本日期为 2022 年 2 月 6 日。

2.31. 通讯中断保护时间设置指令 (0xB3)

2.31.1. 指令说明

该命令用于设置通讯中断保护时间，单位 ms。如果通讯中断超过设置时间会切断输出抱闸锁死。再次运行需要先建立稳定连续的通讯。如果写 0 表示不使能通讯中断保护功能。注意避免在电机刚启动以及运动时写入参数。

2.31.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xB3
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	CanRecvTime_MS 低字节 1	DATA[4] = (uint8_t)(CanRecvTime_MS)
DATA[5]	CanRecvTime_MS 字节 2	DATA[5] = (uint8_t)(CanRecvTime_MS >> 8)
DATA[6]	CanRecvTime_MS 字节 3	DATA[6] = (uint8_t)(CanRecvTime_MS >> 16)
DATA[7]	CanRecvTime_MS 字节 4	DATA[7] = (uint8_t)(CanRecvTime_MS >> 24)

2.31.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据和主机发送的命令相同。

2.31.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB3	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB3	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：数据值都为 0，表示不使能通讯中断保护功能，如果通讯中断，电机会计续执行当前指令。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xB3	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB3	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：该帧数据和主机发送的命令相同。

示例 2:

发送指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB3	0x00	0x00	0x00	0xE8	0x03	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB3	0x00	0x00	0x00	0xE8	0x03	0x00	0x00	CRC16L	CRC16H

说明：Data[4]到 Data[7]（Data[4]为低位，Data[7]为高位）组成数据 0x000003E8，十进制为 1000ms。表示设置通讯中断保护时间为 1000ms，存入 ROM 掉电后保存。那么通讯间隔如果超过 1000ms 就会触发通讯中断保护，切断输出锁死抱闸等。通讯间隔恢复到 1000ms 之内可以重新正常运行。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xB3	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
----	------	----	----	----	----	----	----	----	----	----	--------	--------

0x3E	0x01	0x08	0xB3	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H
------	------	------	------	------	------	------	------	------	------	------	--------	--------

说明：该帧数据和主机发送的命令相同。

2.32. 通讯波特率设置指令（0xB4）

2.32.1. 指令说明

该指令可以设置 CAN 和 RS485 总线的通信波特率。参数设置后会保存在 ROM 中，断电后会保存，再次上电时会以修改后的波特率运行。

2.32.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xB4
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	baudrate	DATA[7] = (uint8_t)baudrate

2.32.3. 回复数据域定义

由于修改了通讯波特率，所以回复指令是随机内容无需处理。

2.32.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB4	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB4	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：Data[7] = 0，代表 RS485 波特率改为 115200bps，CAN 波特率改为 500Kbps。

示例 2：

发送指令：

CAN：

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB4	0x00	0x00	0x00	0x00	0x00	0x00	0x01

RS485：

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB4	0x00	0x00	0x00	0x00	0x00	0x00	0x01	CRC16L	CRC16H

说明：Data[7] = 1，代表 RS485 波特率改为 500Kbps，CAN 波特率改为 1Mbps。

示例 3：

发送指令：

CAN：

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB4	0x00	0x00	0x00	0x00	0x00	0x00	0x02

RS485：

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB4	0x00	0x00	0x00	0x00	0x00	0x00	0x02	CRC16L	CRC16H

说明：Data[7] = 2，代表 RS485 波特率改为 1Mbps，CAN 无效。

2.33. 电机型号读取指令 (0xB5)

2.33.1. 指令说明

该指令用于读取电机型号，读取出的数据为 ACSII 码，可以通过查 ACSII 码表来转换成对应的实际符号。

2.33.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xB5

DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

2.33.3. 回复数据域定义

数据域	说明	数据
DATA[0]	命令字节	0xB5
DATA[1]	电机型号 1	Type1(ASCII)
DATA[2]	电机型号 2	Type2(ASCII)
DATA[3]	电机型号 3	Type3(ASCII)
DATA[4]	电机型号 4	Type4(ASCII)
DATA[5]	电机型号 5	Type5(ASCII)
DATA[6]	电机型号 6	Type6(ASCII)
DATA[7]	电机型号 7	Type7(ASCII)

2.33.4. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB5	0x00	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB5	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 发送指令读取电机型号。

回复指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0xB5	0x58	0x38	0x53	0x32	0x56	0x31	0x30

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0xB5	0x58	0x38	0x53	0x32	0x56	0x31	0x30	CRC16L	CRC16H

说明: 该指令回复 7 个 ACSII 码, 通过查表得到电机型号对应 7 个字符为: X8S2V10。

2.34. 主动回复功能指令 (0xB6)

2.34.1. 指令说明

该指令用于选择指定的指令主动定时回复, 可以指定 1 条以上的指令, 不同指令会循环交替按照设定时间主动回复。如果设定了主动回复的指令, 那么电机在接收指令后就不再回复了。仅对 CAN 版本有效, 485 版本不支持此功能。

2.34.2. 发送数据域定义

数据域	定义	说明
DATA[0]	命令字节	0xB6
DATA[1]	指定主动回复的指令	回复指令包括: 0x60、0x61、0x62、0x92、0x9A、0x9C、0x9D、0x9E;
DATA[2]	主动回复使能位	0: 关闭该指令的主动回复功能; 1: 使能该指令的主动回复功能;
DATA[3]	回复间隔参数低 8 位	回复间隔时间, 单位 10ms。多条指令回复时交替循环回复。
DATA[4]	回复间隔参数高 8 位	
DATA[5]	NULL	NULL
DATA[6]	NULL	NULL
DATA[7]	NULL	NULL

2.34.3. 回复数据域定义

使能后不在回复数据, 电机将按照设定的时间间隔主动回复选择的指令内容。

2.34.4. 通讯示例

示例 1:

发送指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0xB6	0x60	0x01	0x01	0x00	0x00	0x00	0x00

说明：使能 0x60 主动回复指令，时间间隔为 10ms。发送该指令后，电机在接收到指令时不再回复，而是间隔 10ms 时间循环回复 0x60 指令。

2.35. 功能控制指令 (0x20)

2.35.1. 指令说明

该指令用于一些特定功能的使用，是一条复合功能指令，可以包含多条功能控制指令。注意避免在电机刚启动以及运动时写入参数。

2.35.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x20
DATA[1]	功能索引	DATA[1] = (uint8_t)index
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	输入参数低字节 1	DATA[4] = (uint8_t)(Value)
DATA[5]	输入参数字节 2	DATA[5] = (uint8_t)(Value>>8)
DATA[6]	输入参数字节 3	DATA[6] = (uint8_t)(Value>>16)
DATA[7]	输入参数字节 4	DATA[7] = (uint8_t)(Value>>24)

2.35.3. 回复数据域定义

电机在收到命令后回复主机，该帧数据和主机发送的命令相同。

2.35.4. 功能索引说明

索引值	指令名称	功能说明
0x01	清除多圈值	清除电机多圈值、更新零点并保存。重启后生效。
0x02	CANID 滤波器使能	Value 值为 1 代表使能 CANID 滤波器，可以提高 CAN 通讯中电机收发效率；

		Value 值为 0 代表失能 CANID 滤波器，在需要多电机控制指令 0x280 时需要设置为失能；此值会保存在 FLASH，掉电后会记录写入的数值。
0x03	错误状态发送使能	Value 值为 1 代表使能该功能，在电机出现错误状态后主动向总线发送状态指令 0x9A，发送周期为 100ms。错误状态消失后停止发送； Value 值为 0 代表失能该功能；
0x04	多圈值掉电时保存使能	Value 值为 1 代表使能该功能，电机在掉电前会保存当前多圈值； Value 值为 0 代表失能该功能；此时系统默认为单圈模式；重启后生效。
0x05	设置 CANID	Value 值代表将要修改的 CANID 号，保存到 ROM，重启后生效。
0x06	设置位置运行最大正角度	Value 值代表设置的位置运行最大正角度值，保存到 ROM，立即生效。
0x07	设置位置运行最大负角度	Value 值代表设置的位置运行最大负角度值，保存到 ROM，立即生效。

2.35.5. 通讯示例

示例 1:

发送指令:

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x20	0x01	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x20	0x01	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：Data[1] = 0x01，按照索引值表格，代表功能是清除多圈值。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x20	0x01	0x00	0x00	0x00	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x20	0x01	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：该帧数据和主机发送的命令相同。

示例 2:

发送指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x141	0x20	0x02	0x00	0x00	0x01	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x20	0x02	0x00	0x00	0x01	0x00	0x00	0x00	CRC16L	CRC16H

说明：Data[1] = 0x01，按照索引值表格，代表功能是使能 CANID 滤波器，注意使用后无法使用 0x280 多电机指令，再次使用 0x280 指令前需要失能 CANID 滤波器。

回复指令：

CAN:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x20	0x02	0x00	0x00	0x01	0x00	0x00	0x00

RS485:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x20	0x02	0x00	0x00	0x01	0x00	0x00	0x00	CRC16L	CRC16H

说明：该帧数据和主机发送的命令相同。

3. CAN 多电机命令 (0x280+指令)

3.1. 指令说明

ID 号为 280，表示多个电机同时响应同一条指令。指令内容和功能与单电机指令相同，具体可参见单电机指令。

3.2. 通讯示例

假设 CAN 总线上有 4 个电机，ID 号分别是 141，142，143，144。

示例 1:

发送指令:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x280	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00

说明：4 个电机同时收到 0x80 电机关闭指令（具体参见 2.30），然后 4 个电机立即全部执行电机关闭的指令。

回复指令:

4 个电机同时回复，回复 ID 分别为自己的 ID 号。回复顺序取决于各自在总线上的延时。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00

说明：ID 号为 0x241 的电机返回对应指令。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x242	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00

说明：ID 号为 0x242 的电机返回对应指令。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x243	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00

说明：ID 号为 0x243 的电机返回对应指令。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x244	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00

说明：ID 号为 0x244 的电机返回对应指令。

示例 2:

发送指令:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x280	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00

说明：4 个电机同时收到 0x60 读多圈编码器位置数据指令（具体参见 2.21），然后 4 个电机分别回复各自的多圈编码器位置数据。

回复指令:

4 个电机同时回复，回复 ID 分别为自己的 ID 号。回复顺序取决于各自在总线上的延时。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x241	0x60	0x00	0x00	0x00	0x10	0x27	0x00	0x00

说明：ID 号为 0x241 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 10000 个脉冲。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x242	0x60	0x00	0x00	0x00	0x20	0x4E	0x00	0x00

说明：ID 号为 0x242 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00004E20，表示十进制为 20000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 20000 个脉冲。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x243	0x60	0x00	0x00	0x00	0x30	0x75	0x00	0x00

说明：ID 号为 0x243 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00007530，表示十进制为 30000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 30000 个脉冲。

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x244	0x60	0x00	0x00	0x00	0x40	0x9C	0x00	0x00

说明：ID 号为 0x244 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00009C40，表示十进制为 40000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 40000 个脉冲。

4. CANID 设置指令（0x79）

4.1. 指令说明

该命令用于设置和读取 CANID。通讯 ID 使用 0x300，所有总线上的设备都会接收和处理这条指令，修改时需要注意是否连接了多个设备，那样可能会同时将多个设备的 ID 修改为相同的。

主机发送该命令设置和读取 CAN ID，参数如下。

- 1.读写标志位 wReadWriteFlag 为 bool 类型，1 读 0 写；
- 2.CANID, 大小范围(#1~#32), uint16_t 类型（和上位机功能同步），设备标识符 0x140 + ID(1~32)。

4.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x79
DATA[1]	NULL	0x00
DATA[2]	读写标志位	DATA[2] = wReadWriteFlag
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	CANID	DATA[7] = CANID(1~32)

4.3. 回复数据域定义

- 1.电机在收到命令后回复主机，分为如下两种情况，
- 2.设置 CANID,范围 1-32，返回原指令；
- 3.读取 CANID,返回参数如下。

数据域	说明	数据
DATA[0]	命令字节	0x79
DATA[0]	NULL	0x00
DATA[0]	读写标志位	DATA[2] = wReadWriteFlag
DATA[0]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	CANID 低字节 1	DATA[6] = (uint8_t*)(CANID)
DATA[7]	CANID 字节 2	DATA[7] = (uint8_t)(CANID>>8)

4.4. 通讯示例

示例 1:

发送指令:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x300	0x79	0x00	0x00	0x00	0x00	0x00	0x00	0x02

说明: Data[2] = 0 表示写 CANID。Data[7] = 1 表示将电机 CANID 设置为 2, 即发送 ID 为 0x142, 回复 ID 为 0x242。

回复指令:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x300	0x79	0x00	0x00	0x00	0x00	0x00	0x00	0x02

说明: 与发送指令相同。

示例 2:

发送指令:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x300	0x79	0x00	0x01	0x00	0x00	0x00	0x00	0x00

说明: Data[2] = 1 表示读 CANID。

回复指令:

ID 号	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
------	---------	---------	---------	---------	---------	---------	---------	---------

0x300	0x79	0x00	0x01	0x00	0x00	0x00	0x42	0x02
-------	------	------	------	------	------	------	------	------

说明：Data[6]和 Data[7] 组成 0x242 表示电机发送 ID 为 0x142，回复 ID 为 0x242。

5. 运动模式控制指令_CAN (0x400 + ID)

5.1. 指令说明

该指令由 5 个输入参数组成：p_des(期望位置)，v_des(期望速度)，t_ff(前馈力矩)，kp(位置偏差系数)，kd(速度偏差系数)。

每个参数都预设了范围大小：

p_des: -12.5 到 12.5，单位 rad；

数据类型为 uint16_t，取值范围为 0~65535，其中 0 代表-12.5，65535 代表 12.5，0~65535 中间的所有数值，按比例映射 至-12.5~12.5。

v_des: -45 到 45，单位 rad/s；

数据类型为 12 位无符号整数，取值范围为 0~4095，其中 0 代表-45，4095 代表 45，0~4095 中间的所有数值，按比例映射至-45~45。

kp: 0 到 500；

数据类型为 12 位无符号整数，取值范围为 0~4095，其中 0 代表 0，4095 代表 500，0~4095 中间的所有数值，按比例映射至 0~500。

kd: 0 到 5；

数据类型为 12 位无符号整数，取值范围为 0~4095，其中 0 代表 0，4095 代表 5，0~4095 中间的所有数值，按比例映射至 0~5。

t_ff: -24 到 24，单位 N·m；

数据类型为 12 位无符号整数，取值范围为 0~4095，其中 0 代表-24，4095 代表 24，0~4095 中间的所有数值，按比例映射至-24~24。

功能表达式：

$$I_{qRef} = [kp \cdot (p_{des} - p_{fd_实际位置}) + kd \cdot (v_{des} - v_{fb_实际速度}) + t_{ff}] \cdot KT_扭矩系数;$$

I_{qRef} 为最后给定电机的输出电流大小。

5.2. 发送数据域定义（大端字节序）

数据域	数据划分	数据组合	数据定义	数据范围
DATA[0]	4-7bit	p_des[8-15]	p_des 的高 8 位数据	16 位范围
	0-3bit			
DATA[1]	4-7bit	p_des[0-7]	p_des 的低 8 位数据	
	0-3bit			
DATA[2]	4-7bit	v_des[4-11]	v_des 的高 8 位数据	12 位范围
	0-3bit			
DATA[3]	4-7bit	v_des[0-3]	v_des 的低 4 位数据	
	0-3bit	kp[8-11]	kp 的高 4 位数据	
DATA[4]	4-7bit	kp[0-7]	kp 的低 8 位数据	
	0-3bit			
DATA[5]	4-7bit	kd[4-11]	kd 的高 8 位数据	12 位范围
	0-3bit			
DATA[6]	4-7bit	kd[0-3]	kd 的低 4 位数据	
	0-3bit	t_ff[8-11]	t_ff 的高 4 位数据	
DATA[7]	4-7bit	t_ff[0-7]	t_ff 的低 8 位数据	
	0-3bit			

5.3. 回复数据域定义（大端字节序）

数据域	数据划分	数据组合	数据定义	数据范围
DATA[0]	7-0bit	CANID[0-7]	设备 CAN 地址号	8 位范围
DATA[1]	4-7bit	p[8-15]	当前位置 p 的高 8 位数据	16 位范围
	0-3bit			
DATA[2]	4-7bit	p[0-7]	当前位置 p 的低 8 位数据	
	0-3bit			
DATA[3]	4-7bit	v[4-11]	当前速度 v 的高 8 位数据	12 位范围
	0-3bit			

DATA[4]	4-7bit	v[0-3]	当前速度 v 的低 4 位数据	
	0-3bit	t[8-11]	当前力矩 t 的高 4 位数据	
DATA[5]	4-7bit	t[0-7]	当前力矩 t 的低 8 位数据	12 位范围
	0-3bit			
DATA[6]	4-7bit	NULL	NULL	NULL
	0-3bit			
DATA[7]	4-7bit	NULL	NULL	NULL
	0-3bit	NULL	NULL	

5.4. 通讯示例

示例 1:

发送指令: ID 号 0x401

数据域	数据	数据划分		数据定义	数据范围	数据计算说明
DATA[0]	0xE	4-7bit	0xE	p_des 值为	-12.5rad~12.5rad 共 25rad	$p_des = (58982/65535) * 25 + (-12.5) = 9.99rad$
	6	0-3bit	0x6	0xE666; 十		
DATA[1]	0x66	4-7bit	0x6	进制为		
		0-3bit	0x6	(58982)		
DATA[2]	0x82	4-7bit	0x8	v_des 值为	-45rad/s~45rad/s 共 90rad/s	$v_des = (2094/4095) * 90 + (-45) = 1.021 rad/s$
		0-3bit	0x2	0x82E; 十		
DATA[3]	0xE0	4-7bit	0xE	进制为		
		0-3bit	0x0	(2094)		
DATA[4]	0x52	4-7bit	0x5	kp 值为	0~500 共 500	$kp = (82/4095) * 500 + 0 = 10.012$
		0-3bit	0x2	0x52; 十进		
DATA[5]	0x33	4-7bit	0x3	制为 (82)	0~5 共 5	$kd = (819/4095) * 5 + 0 = 1$
		0-3bit	0x3	kd 值为		
DATA[6]	0x3B	4-7bit	0x3	0x333; 十		
		0-3bit	0xB	进制为		
				(819)		
				t_ff 值为	-24N-m~24N-m	$t_ff = (2901/4095) * 48$

DATA[7]	0x55	4-7bit	0x5	0xB55; 十进制为 (2901)	共 48N-m	+(-24)=10.004 N-m
		0-3bit	0x5			

回复指令：ID 号 0x501

数据域	数据	数据划分		数据定义	数据范围	数据计算说明
DATA[0]	0x01	7-0bit	0x1	CANID 号	0-32	设备地址 ID 号
DATA[1]	0xE6	4-7bit	0xE	p 值为 0xE666; 十进制为 (58982)	-12.5rad~12.5rad 共 25rad	$p_des = (58982/65535) * 25 + (-12.5) = 9.99 \text{ rad}$
		0-3bit	0x6			
DATA[2]	0x65	4-7bit	0x6	v 值为 0x82E; 十进制为 (2094)	-45rad/s~45rad/s 共 90rad/s	$v_des = (2094/4095) * 90 + (-45) = 1.021 \text{ rad/s}$
		0-3bit	0x6			
DATA[3]	0x82	4-7bit	0x8	t 值为 0xB55; 十进制为 (2901)	-24N-m~24N-m 共 48N-m	$t_ff = (2901/4095) * 48 + (-24) = 10.004 \text{ N-m}$
		0-3bit	0x2			
DATA[4]	0xEB	4-7bit	0xE	NULL	NULL	NULL
		0-3bit	0xB			
DATA[5]	0x55	4-7bit	0x5	NULL	NULL	NULL
		0-3bit	0x5			
DATA[6]		4-7bit	NULL	NULL	NULL	NULL
		0-3bit	NULL			
DATA[7]		4-7bit	NULL	NULL	NULL	NULL
		0-3bit	NULL			

6. RS485 多电机命令 (0xCD + 指令)

6.1. 指令说明

ID 号为 0xCD，表示多个电机同时相应同一条指令。指令内容和功能与单电机指令相同，具体可参见单电机指令。

6.2. 通讯示例

假设 RS485 总线上有 4 个电机，ID 号分别是 01，02，03，04。

示例 1:

发送指令:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0xCD	0x08	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：4 个电机同时收到 0x80 电机关闭指令（具体参见 2.30），然后 4 个电机立即全部执行电机关闭的指令。

回复指令:

4 个电机同时回复，回复 ID 分别为自己的 ID 号。回复顺序取决于各自在总线上的延时。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：ID 号为 0x01 的电机返回对应指令。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x02	0x08	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：ID 号为 0x02 的电机返回对应指令。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x03	0x08	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：ID 号为 0x03 的电机返回对应指令。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x04	0x08	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明：ID 号为 0x04 的电机返回对应指令。

示例 2:

发送指令:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0xCD	0x08	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: 4 个电机同时收到 0x60 读多圈编码器位置数据指令（具体参见 2.21），然后 4 个电机分别回复各自的多圈编码器位置数据。

回复指令:

4 个电机同时回复，回复 ID 分别为自己的 ID 号。回复顺序取决于各自在总线上的延时。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x01	0x08	0x60	0x00	0x00	0x00	0x10	0x27	0x00	0x00	CRC16L	CRC16H

说明: ID 号为 0x01 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00002710，表示十进制为 10000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 10000 个脉冲。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x02	0x08	0x60	0x00	0x00	0x00	0x20	0x4E	0x00	0x00	CRC16L	CRC16H

说明: ID 号为 0x02 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00004E20，表示十进制为 20000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 20000 个脉冲。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x03	0x08	0x60	0x00	0x00	0x00	0x30	0x75	0x00	0x00	CRC16L	CRC16H

说明: ID 号为 0x03 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00007530，表示十进制为 30000。代表电机当前相对多圈零偏（初始位置）的多圈编码器值为 30000 个脉冲。

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0x04	0x08	0x60	0x00	0x00	0x00	0x40	0x9C	0x00	0x00	CRC16L	CRC16H

说明: ID 号为 0x244 的电机回复数据中 Data[4]到 data[7]组成一个（Data[4]为最低位，Data[7]为最高位）32 位数据为 0x00009C40，表示十进制为 40000。代表电机当

前相对多圈零偏（初始位置）的多圈编码器值为 40000 个脉冲。

7. RS485-ID 设置指令（0x79）

7.1. 指令说明

该命令用于设置和读取 RS485 ID。通讯 ID 使用 0xCD，所有总线上的设备都会接收和处理这条指令，修改时需要注意是否连接了多个设备，那样可能会同时将多个设备的 ID 修改为相同的。

主机发送该命令设置和读取 RS485 ID，参数如下。

- 1.读写标志位 wReadWriteFlag 为 bool 类型，1 读 0 写；
- 2.RS485-ID,大小范围(#1~#32),uint16_t 类型（和上位机功能同步）,设备标识符 ID(1~32)。

7.2. 发送数据域定义

数据域	说明	数据
DATA[0]	命令字节	0x79
DATA[1]	NULL	0x00
DATA[2]	读写标志位	DATA[2] = wReadWriteFlag
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	RS485ID	DATA[7] = RS485ID(1~32)

7.3. 回复数据域定义

- 1.电机在收到命令后回复主机，分为如下两种情况，
- 2.设置 RS485ID,范围 1-32，返回原指令；
- 3.读取 RS485ID,返回参数如下。

数据域	说明	数据
DATA[0]	命令字节	0x79

DATA[0]	NULL	0x00
DATA[0]	读写标志位	DATA[2] = wReadWriteFlag
DATA[0]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	RS485ID 低字节 1	DATA[6] = (uint8_t*)(RS485ID)
DATA[7]	RS485ID 字节 2	DATA[7] = (uint8_t)(RS485ID>>8)

7.4. 通讯示例

示例 1:

发送指令:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0xCD	0x08	0x79	0x00	0x00	0x00	0x00	0x00	0x00	0x02	CRC16L	CRC16H

说明: Data[2] = 0 表示写 RS485ID。Data[7] = 1 表示将电机 RS485ID 设置为 2。

回复指令:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0xCD	0x08	0x79	0x00	0x00	0x00	0x00	0x00	0x00	0x02	CRC16L	CRC16H

说明: 与发送指令相同。

示例 2:

发送指令:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0xCD	0x08	0x79	0x00	0x01	0x00	0x00	0x00	0x00	0x00	CRC16L	CRC16H

说明: Data[2] = 1 表示读 RS485ID。

回复指令:

帧头	ID 号	长度	D0	D1	D2	D3	D4	D5	D6	D7	CRC16L	CRC16H
0x3E	0xCD	0x08	0x79	0x00	0x01	0x00	0x00	0x00	0x00	0x02	CRC16L	CRC16H

说明: Data[7] = 0x2 表示电机发送 ID 为 0x2, 回复 ID 为 0x2。

8. 指示灯说明

8.1. 状态说明

- 指示灯长亮表示电机正常运行；
- 慢闪表示电机出现二级错误，如果达到恢复条件后自动恢复正常运行，指示灯长亮；
- 快闪表示电机出现一级错误，电机无法恢复错误，需要检查电机故障并重启后才可能继续运行。

8.2. 故障说明表格

故障名称	说明	错误等级
硬件过流	电机电流超过极限值，可能存在短路、缺相、失控、电机损坏等情况。	一级
堵转错误	电流在达到堵转电流后转速很低，并持续一段时间。说明电机负载过大。	一级
欠压错误	电源输入低于设定欠压值	二级
过压错误	电源输入高于设定过压值	二级
相电流过流	软件检测电机电流超过极限值，可能存在短路、缺相、失控、电机损坏等情况。	一级
功率超限错误	电源输入电流超过限制值，可能存在负载过大或者转速过高的情况。	二级
标定参数读取错误	写入参数失败引起参数丢失	一级
超速错误	电机运行转速超过极限值，可能存在超压、拖拽使用。	二级
电机过温错误	电机温度超过设定值，可能存在短路、参数错误、长时间过载使用的情况。	二级
编码器校准错误	编码器校准结果与标准值偏差过大。	二级

9. 版本修订信息

V3.1 版本:

- 1) 版本修订内容:
 - a. 修正 5.0 运控指令中回复数据定义。
- 2) 版本修订日期: 2022.6.23

V3.2 版本:

- 1) 版本修订内容:
 - a. 增加指示灯说明。
- 2) 版本修订日期: 2022.7.27

V3.3 版本:

- 1) 版本修订内容:
 - a. 增加功能控制指令 0x20: 清除多圈值功能和 CAN 滤波器使能控制功能。
- 2) 版本修订日期: 2022.7.31

V3.4 版本:

- 1) 版本修订内容:
 - a. 增加位置跟踪指令 0xA3;
 - b. 在 0x43 指令中增加位置规划和速度规划加速度和减速度 4 个值的设置。
- 2) 版本修订日期: 2022.8.17

V3.5 版本:

- 1) 版本修订内容:
 - a. 增加带速度限制的位置跟踪指令 0xA5;
 - b. 增加功能控制指令 0x20: 错误状态发送和多圈值掉电保存选择功能;
 - c. 增加 0xB5 指令读取电机型号。
- 2) 版本修订日期: 2022.9.05

V3.6 版本:

- 1) 版本修订内容:
 - a. 增加 RS485 广播指令说明 0xCD。
- 2) 版本修订日期: 2022.10.13

V3.7 版本:

- 1) 版本修订内容:
 - a. 去掉 A3 指令;
 - b. 将 A5 合并到 A4 中;
 - c. 增加 A6 单圈位置指令;
 - d. 增加读取单圈编码器指令 0x90;
 - e. 增加读取电机单圈角度指令 0x94。
- 2) 版本修订日期: 2022.11.26

V3.8 版本:

- 1) 版本修订内容:
 - a. RS485 协议 2Mbps 波特率修改为 2.5Mbps。
- 2) 版本修订日期: 2022.11.26

V3.9 版本:

- 1) 版本修订内容:
 - a. 增加 485 串口配置说明;
 - b. 增加 0x42 指令中功能索引, 可以通过索引读取位置和速度的加速度和减速度值;
 - c. 增加 0xB6 主动回复功能。
- 2) 版本修改日期: 2023.3.11

V4.0 版本:

- 1) 版本修订内容:
 - a. 在 0x20 指令中增加设置 CANID 功能;
 - b. 在 0x20 指令中增加设置最大正角度限制值;
 - c. 在 0x20 指令中增加设置最大负角度限制值。
- 2) 版本修改日期: 2023.10.16

V4.1 版本:

- 1) 版本修订内容:
 - a. 修改电机单圈角度 circleAngle, 为 uint16_t 类型数据。

2) 版本修改日期: 2024.2.13

V4.2 版本:

1) 版本修订内容:

- a. 修改写入 PID 参数到 RAM 命令 (0x31), 使用索引写入电流环、速度环和位置环的 PID 参数到 RAM;
- b. 修改写入 PID 参数到 ROM 命令 (0x32), 使用索引写入电流环、速度环和位置环的 PID 参数到 ROM。

2) 版本修改日期: 2024.5.28

V4.3 版本:

1) 版本修订内容:

- a. 加入力控位置闭环控制命令 (0xA9)
- b. 在速度闭环控制命令 (0xA2) 中加入力控模式
- c. 完善 MIT 指令
- d. 加入编码器数据异常错误

2) 版本修改日期: 2025.5.12