

# 伺服电机控制协议

## Servo Motor Control Protocol

适用驱动: V3

版本: V4.4

日期: 2025.09

**Applicable driver: V3**

**Version: V4.4**

**Date: 2025.09**

## 免责声明

### Disclaimer

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。

Thanks for choosing MYACTUATOR. Please read this statement carefully before using. Once used, this statement is deemed to be approved and accepted. Please install and use this product strictly in accordance with the manual, product description and relevant laws, regulations, policies and guidelines. In the process of using the product, the user undertakes to be responsible for his own behavior and all consequences arising therefrom. MYACTUATOR will not be liable for any loss caused by improper use, installation and modification of the user.

MYACTUATOR is the trademark of Suzhou Micro Actuator Technology Co., Ltd. and its affiliates. Product names, brands, etc. appearing in this document are trademarks or registered trademarks of their respective companies.

This product and manual are copyrighted by MYACTUATOR. Reproduction in any form is not permitted without permission. The final interpretation right of the disclaimer belongs to MYACTUATOR

## 目录 Catalog

|  |    |
|--|----|
| 免责声明 .....   | 1  |
| Disclaimer .....   | 1  |
| 1. 通讯总线参数及报文格式 .....   | 1  |
| Communication Bus Parameters and Message Format .....                    | 1  |
| 1.1. CAN 总线 .....  | 1  |
| CAN Bus .....  | 1  |
| 1.2. RS485 总线 .....  | 2  |
| RS485 Bus .....  | 2  |
| 2. 单电机指令说明 .....   | 3  |
| Single Motor Command Description .....                                   | 3  |
| 2.1. 读取 PID 参数指令 (0x30) .....  | 3  |
| Read PID Parameter Command (0x30) .....                                  | 3  |
| 2.2. 写入 PID 参数到 RAM 指令 (0x31) .....                                      | 6  |
| Write PID Parameters to RAM Command (0x31) .....                         | 6  |
| 2.3. 写入 PID 参数到 ROM 指令 (0x32) .....                                      | 9  |
| Write PID Parameters to ROM Command (0x32) .....                         | 9  |
| 2.4. 读取加速度指令 (0x42) .....  | 12 |
| Read Acceleration Command (0x42) .....                                   | 12 |
| 2.5. 写入加减速度到 RAM 和 ROM 指令 (0x43) .....                                   | 17 |
| Write Acceleration to RAM and ROM Command (0x43) .....                   | 17 |
| 2.6. 读取多圈编码器位置数据指令 (0x60) .....  | 24 |
| Read Multi-Turn Encoder Position Data Command (0x60) .....               | 24 |
| 2.7. 读取多圈编码器原始位置数据指令 (0x61) .....  | 27 |
| Read Multi-Turn Encoder Original Position Data Command (0x61) .....      | 27 |
| 2.8. 读取多圈编码器零偏数据指令 (0x62) .....  | 30 |
| Read Multi-Turn Encoder Zero Offset Data Command (0x62) .....            | 30 |
| 2.9. 写入编码器多圈值到 ROM 作为电机零点指令 (0x63) .....                                 | 33 |
| Write Encoder Multi-Turn Value to ROM as Motor Zero Command (0x63) ..... | 33 |

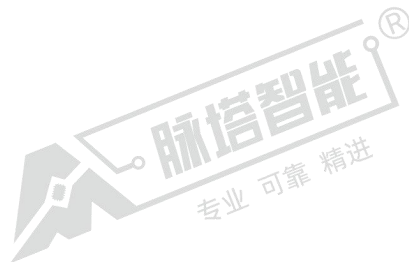
|  |    |
|--|----|
| 2.10. 写入编码器当前多圈位置到 ROM 作为电机零点指令 (0x64) .....   | 35 |
| Write the Current Multi-Turn Position of the Encoder to the ROM as the Motor Zero Command (0x64) ..... | 35 |
| 2.11. 读取多圈角度指令 (0x92) .....  | 38 |
| Read Multi-Turn Angle Command (0x92) .....   | 38 |
| 2.12. 读取单圈角度指令 (0x94) .....  | 41 |
| Read Single-Turn Angle Command (0x94) .....  | 41 |
| 2.13. 读取电机状态 1 和错误标志指令 (0x9A) .....  | 44 |
| Read Motor Status 1 and Error Flag Command (0x9A) .....  | 44 |
| 2.14. 读取电机状态 2 指令 (0x9C) .....   | 51 |
| Read Motor Status 2 Command (0x9C) .....   | 51 |
| 2.15. 读取电机状态 3 指令 (0x9D) .....   | 55 |
| Read Motor Status 3 Command (0x9D) .....   | 55 |
| 2.16. 电机关闭指令 (0x80) .....  | 58 |
| Motor Shutdown Command (0x80) .....  | 58 |
| 2.17. 电机停止指令 (0x81) .....  | 59 |
| Motor Stop Command (0x81) .....  | 59 |
| 2.18. 转矩闭环控制指令 (0xA1) .....  | 60 |
| Torque Closed-Loop Control Command (0xA1) .....  | 60 |
| 2.19. 速度闭环控制指令 (0xA2) .....  | 66 |
| Speed Closed-Loop Control Command (0xA2) .....   | 66 |
| 2.20. 绝对位置闭环控制指令 (0xA4) .....  | 73 |
| Absolute Position Closed-Loop Control Command (0xA4) .....   | 73 |
| 2.21. 单圈位置控制指令 (0xA6) .....  | 81 |
| Single-Turn Position Control Command (0xA6) .....  | 81 |
| 2.22. 增量位置闭环控制指令 (0xA8) .....  | 87 |
| Incremental Position Closed-Loop Control Command (0xA8) .....  | 87 |
| 2.23. 力控位置闭环控制指令 (0xA9) .....  | 94 |
| Force Control Position Closed-Loop Command (0xA9) .....  | 94 |

|   |     |
|---|-----|
| 2.24. SF 指令（0x72 带前馈速度的位置控制指令） .....                                    | 103 |
| SF Command (0x72 Position Control with Speed Feedforward) .....         | 103 |
| 2.25. TF 指令（0x73 带前馈扭矩的位置控制指令） .....                                    | 111 |
| TF Command (0x73 Position Control with Torque Feedforward) .....        | 111 |
| 2.26. 系统运行模式获取（0x70） .....  | 119 |
| System Operating Mode Acquisition (0x70) .....                          | 119 |
| 2.27. 系统复位指令（0x76） .....  | 122 |
| System Reset Command (0x76) .....                                       | 122 |
| 2.28. 系统抱闸释放指令（0x77） .....  | 123 |
| System Brake Release Command (0x77) .....                               | 123 |
| 2.29. 系统抱闸锁死指令（0x78） .....  | 124 |
| System Brake Lock Command (0x78) .....                                  | 124 |
| 2.30. 系统运行时间读取指令（0xB1） .....  | 125 |
| System Runtime Read Command (0xB1) .....                                | 125 |
| 2.31. 系统软件版本日期读取指令（0xB2） .....  | 128 |
| System Software Version Date Read Command (0xB2) .....                  | 128 |
| 2.32. 通讯中断保护时间设置指令（0xB3） .....  | 130 |
| Communication Interruption Protection Time Setting Command (0xB3) ..... | 130 |
| 2.33. 通讯波特率设置指令（0xB4） .....   | 134 |
| Communication Baud Rate Setting Command (0xB4) .....                    | 134 |
| 2.34. 电机型号读取指令（0xB5） .....  | 137 |
| Motor Model Reading Command (0xB5) .....                                | 137 |
| 2.35. 主动回复功能指令（0xB6） .....  | 140 |
| Active Reply Function Command (0xB6) .....                              | 140 |
| 2.36. 功能控制指令（0x20） .....  | 142 |
| Function Control Command (0x20) .....                                   | 142 |
| 3. CAN 多电机指令（0x280+指令） .....  | 148 |
| CAN Multi-Motor Command (0x280 + Command) .....                         | 148 |
| 3.1. 指令说明 .....   | 148 |

|   |     |
|---|-----|
| Instruction Description .....                                   | 148 |
| 3.2. 通讯示例 .....   | 149 |
| Communication Example .....                                     | 149 |
| 4. 运动模式控制指令_CAN (0x400 + ID) .....                              | 153 |
| Motion Mode Control Command_CAN (0x400 + ID) .....              | 153 |
| 4.1. 指令说明 .....   | 153 |
| Instruction Description .....                                   | 153 |
| 4.2. 发送数据域定义 (大端字节序) .....                                      | 154 |
| Send Data Field Definition (Big-endian byte order) .....        | 154 |
| 4.3. 回复数据域定义 (大端字节序) .....                                      | 155 |
| Reply Data Field Definition (Big-endian byte order) .....       | 155 |
| 4.4. 通讯示例 .....   | 156 |
| Communication Example .....                                     | 156 |
| 5. RS485 多电机指令 (0xCD + 指令) .....                                | 159 |
| RS485 Multi-Motor Command (0xCD + Command) .....                | 159 |
| 5.1. 指令说明 .....   | 159 |
| Instruction Description .....                                   | 159 |
| 5.2. 通讯示例 .....   | 159 |
| Communication Example .....                                     | 159 |
| 6. RS485 运动模式控制指令 .....   | 164 |
| RS485 Motion Mode Control Command .....                         | 164 |
| 6.1. 指令说明 .....   | 164 |
| Instruction Description .....                                   | 164 |
| 6.2. 发送数据域定义 (小端字节序) .....                                      | 164 |
| Transmit Data Field Definition (Little-Endian Byte Order) ..... | 164 |
| 6.3. 回复数据域定义 (小端字节序) .....                                      | 166 |
| Response Data Field Definition (Little-Endian Byte Order) ..... | 166 |
| 7. 指示灯说明 .....  | 168 |
| Indicator Light Description .....                               | 168 |

|                                    |     |
|------------------------------------|-----|
| 7.1. 状态说明 .....                    | 168 |
| Status Description .....           | 168 |
| 7.2. 故障说明表格 .....                  | 169 |
| Failure Description Form .....     | 169 |
| 8. 版本修订信息 .....                    | 171 |
| Version Revision Information ..... | 171 |







## 1. 通讯总线参数及报文格式

### Communication Bus Parameters and Message Format

#### 1.1. CAN 总线

##### CAN Bus

##### 1.1.1. 参数

###### Parameters

总线接口: CAN

波特率: 1Mbps

Bus interface: CAN

Baud rate: 1Mbps

##### 1.1.2. 报文格式

###### Message Format

标识符: 单电机指令发送:  $0x140 + ID(1\sim32)$

多电机指令发送:  $0x280$

回复:  $0x240 + ID(1\sim32)$

帧格式: 数据帧

帧类型: 标准帧

DLC: 8 字节

Identifier: Single motor command sending:  $0x140 + ID(1\sim32)$

Multi-motor command sending:  $0x280$

Reply:  $0x240 + ID(1\sim32)$

Frame format: Data frame

Frame Type: Standard Frame

DLC: 8 bytes

## 1.2. RS485 总线

### RS485 Bus

#### 1.2.1. 参数

##### Parameters

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

Bus interface: RS485

Baudrate: 115200bps, 500Kbps, 1Mbps, 1.5Mbps, 2.5Mbps

Serial port configuration: 8 data bits, 1 stop bit, no parity bit

#### 1.2.2. 报文格式

##### Message Format

| 类型<br>Type          | 数据定义<br>Data Definition | 字节数<br>Bytes                    | 说明<br>Description   |
|---------------------|-------------------------|---------------------------------|---|
| 帧头<br>Frame header  | 0x3E                    | 1                               | 通讯帧头, 用于识别。<br>Communication frame header, used for identification.   |
| ID                  | 1~32                    | 1                               | 设备地址, 对应每个电机的 ID 号。<br>Device address, corresponding to the ID number of each motor.                                |
| 数据长度<br>Data Length | 数据长度<br>Data Length     | 1                               | 数据域的长度, 标准协议中长度固定 8 个字节。<br>The length of the data field. In the standard protocol, the length is fixed to 8 bytes. |
| 数据域<br>Data field   | 数据内容<br>Data content    | 根据长度<br>According to the length | 标准协议中与 CAN 的数据域内容完全一致。<br>The content of the data field in the  |

|             |                     |   |   |
|-------------|---------------------|---|---|
|             |                     |   | standard protocol is exactly the same as that of the CAN.               |
| 校验<br>Check | CRC 校验<br>CRC Check | 2 | CRC16 校验, 低位在前, 高位在后。<br>CRC16 check, low order first, high order last. |

## 2. 单电机指令说明

### Single Motor Command Description

#### 2.1. 读取 PID 参数指令 (0x30)

##### Read PID Parameter Command (0x30)

##### 2.1.1. 指令说明

##### Instruction Description

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。

This command can read the PID parameters of the current, speed and position, the data type is Float, determined by the index value. For details, see 2.1.4 Index description table.

##### 2.1.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description             | Data                     |
|------------|-------------------------|--------------------------|
| DATA[0]    | 指令字节<br>Command byte    | 0x30                     |
| DATA[1]    | 参数索引<br>Parameter index | 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

#### Reply Data Field Definition

| Data field | Description                     | Data                           |
|------------|---------------------------------|--------------------------------|
| DATA[0]    | 指令字节<br>Command byte            | 0x30                           |
| DATA[1]    | 功能索引<br>Function index          | DATA[1] = (uint8_t)index       |
| DATA[2]    | NULL                            | 0x00                           |
| DATA[3]    | NULL                            | 0x00                           |
| DATA[4]    | 参数低字节 1<br>Parameter low byte 1 | DATA[4] = (uint8_t)(Value)     |
| DATA[5]    | 参数字节 2<br>Parameter byte 2      | DATA[5] = (uint8_t) (Value>>8) |
| DATA[6]    | 参数字节 3<br>Parameter byte 3      | DATA[6] = (uint8_t)(Value>>16) |
| DATA[7]    | 参数字节 4<br>Parameter byte 4      | DATA[7] = (uint8_t)(Value>>24) |

### 2.1.4. 功能索引说明

#### Function Index Description

| Index | Parameter                           |
|-------|-------------------------------------|
| 0x01  | 电流环 KP<br>Current loop KP parameter |
| 0x02  | 电流环 KI<br>Current loop KI parameter |
| 0x04  | 速度环 KP<br>Speed loop KP parameter   |

|      |                                      |
|------|--------------------------------------|
| 0x05 | 速度环 KI<br>Speed loop KI parameter    |
| 0x07 | 位置环 KP<br>Position loop KP parameter |
| 0x08 | 位置环 KI<br>Position loop KI parameter |
| 0x09 | 位置环 KD<br>Position loop KD parameter |

### 2.1.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

According to the index value table, Data[1] = 0x01, it means the current loop KP and indicates the read current loop KP parameter.

回复指令:

**Reply command:**

#### 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:

| Frame header | ID   | Length | 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 1.0, 代表电机当前电流环 KP 参数为 1.0。

#### Description:

In the frame data returned, Data[1]=0x01, it means the current loop KP parameter. From Data[4] to Data[7], these form a 32-bit data that is 0x3F800000, (Data[4] is the lowest bit, Data[7] is the highest bit), and the data type is Float. When convert it to decimal, the data is 1.0, this means that the current loop KP parameter at this time is 1.0.

The online conversion website can be used: <http://www.speedfly.cn/tools/hexconvert/>.

## 2.2. 写入 PID 参数到 RAM 指令 (0x31)

### Write PID Parameters to RAM Command (0x31)

#### 2.2.1. 指令说明

##### Instruction Description

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。

This command can write the parameters of current, speed, position loop KP and KI to RAM at one time, and it will not be saved after power off. The data type is Float, and it is determined by the index value. For details, see 2.2.4 Index Description Table. Be careful

to avoid writing parameters when the motor has just started and is in motion.

### 2.2.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description                     | Data                           |
|------------|---------------------------------|--------------------------------|
| DATA[0]    | 指令字节<br>Command byte            | 0x31                           |
| DATA[1]    | 功能索引<br>Function index          | DATA[1] = (uint8_t)index       |
| DATA[2]    | NULL                            | 0x00                           |
| DATA[3]    | NULL                            | 0x00                           |
| DATA[4]    | 参数低字节 1<br>Parameter low byte 1 | DATA[4] = (uint8_t)(Value)     |
| DATA[5]    | 参数字节 2<br>Parameter byte 2      | DATA[5] = (uint8_t) (Value>>8) |
| DATA[6]    | 参数字节 3<br>Parameter byte 3      | DATA[6] = (uint8_t)(Value>>16) |
| DATA[7]    | 参数字节 4<br>Parameter byte 4      | DATA[7] = (uint8_t)(Value>>24) |

### 2.2.3. 回复数据域定义

#### Reply Data Field Definition

回复数据内容和发送数据一致。

The content of the reply data is the same as the sent data.

### 2.2.4. 功能索引说明

#### Function Index Description

| Index | Parameter                           |
|-------|-------------------------------------|
| 0x01  | 电流环 KP<br>Current loop KP parameter |
| 0x02  | 电流环 KI                              |

|      |                                      |
|------|--------------------------------------|
|      | Current loop KI parameter            |
| 0x04 | 速度环 KP<br>Speed loop KP parameter    |
| 0x05 | 速度环 KI<br>Speed loop KI parameter    |
| 0x07 | 位置环 KP<br>Position loop KP parameter |
| 0x08 | 位置环 KI<br>Position loop KI parameter |
| 0x09 | 位置环 KD<br>Position loop KD parameter |

### 2.2.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] = 0x01, and it means the current loop KP parameter. From Data[4] to Data[7], these form a 32-bit data that is 0x3FC00000, (Data[4] is the lowest bit, Data[7] is the highest bit), and the data type is Float. When convert it to decimal, the data value is 1.5, this means that the current loop KP parameter is set to 1.5 and written to the RAM of the motor drive. In addition, the parameter is not saved after power off.

The online conversion website can be used: <http://www.speedfly.cn/tools/hexconvert/>.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | Data[0] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|
| 0x241 | 0x31    | 0x00    | 0x00    | 0x00    | 0x00    | 0xC0    | 0x3F    |

#### RS485:

| Frame header | ID   | Length | 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 指令 (0x32)

### Write PID Parameters to ROM Command (0x32)

#### 2.3.1. 指令说明

##### Instruction Description

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。只有在电机失能时，才能将参数成功保存到 ROM。

This command can write the parameters of current, speed, position loop KP and KI to ROM at one time, which can be saved after power off. The data type is Float and can be determined by the index value, as detailed in 2.2.4 Index Description. Be careful to avoid writing parameters when the motor has just started and is in motion. The parameters can be successfully saved to ROM only when the motor is disabled.

### 2.3.2. 发送数据域定义

#### Send Data Field Definition

| Data Field | Description                     | Data                             |
|------------|---------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte            | 0x32                             |
| DATA[1]    | 功能索引<br>Function index          | DATA[1] = (uint8_t)index         |
| DATA[2]    | NULL                            | 0x00                             |
| DATA[3]    | NULL                            | 0x00                             |
| DATA[4]    | 参数低字节 1<br>Parameter low byte 1 | DATA[4] = (uint8_t)(Value)       |
| DATA[5]    | 参数字节 2<br>Parameter byte 2      | DATA[5] = (uint8_t)(Value >> 8)  |
| DATA[6]    | 参数字节 3<br>Parameter byte 3      | DATA[6] = (uint8_t)(Value >> 16) |
| DATA[7]    | 参数字节 4<br>Parameter byte 4      | DATA[7] = (uint8_t)(Value >> 24) |

### 2.3.3. 回复数据域定义

#### Reply Data Field Definition

回复数据内容和发送数据一致。

The content of the reply data is the same as the sent data.

### 2.3.4. 功能索引说明

#### Function Index Description

| Index | Parameter                            |
|-------|--------------------------------------|
| 0x01  | 电流环 KP<br>Current loop KP parameter  |
| 0x02  | 电流环 KI<br>Current loop KI parameter  |
| 0x04  | 速度环 KP<br>Speed loop KP parameter    |
| 0x05  | 速度环 KI<br>Speed loop KI parameter    |
| 0x07  | 位置环 KP<br>Position loop KP parameter |
| 0x08  | 位置环 KI<br>Position loop KI parameter |
| 0x09  | 位置环 KD<br>Position loop KD parameter |

### 2.3.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

### Description:

Data[1] = 0x01, and it means the current loop KP parameter. From Data[4] to Data[7], these form a 32-bit data that is 0x3FC00000, (Data[4] is the lowest bit, Data[7] is the highest bit), and the data type is Float. When convert it to decimal, the data value is 1.5, this means that the current loop KP parameter is set to 1.5 and written to the ROM of the motor drive. In addition, the parameter is saved after the power is off.

The online conversion website can be used: <http://www.speedfly.cn/tools/hexconvert/>.

### 回复指令:

### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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. 读取加速度指令 (0x42)

### Read Acceleration Command (0x42)

#### 2.4.1. 指令说明

#### Instruction Description

主机发送该指令读取当前电机的加速度参数

The host sends this command to read the acceleration parameters of the current motor.

### 2.4.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description            | Data                     |
|------------|------------------------|--------------------------|
| DATA[0]    | 指令字节<br>Command byte   | 0x42                     |
| DATA[1]    | 功能索引<br>Function index | 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

#### Reply Data Field Definition

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。

The acceleration parameter is included in the drive response data. Acceleration data Accel is int32\_t type, the unit is 1dps/s, and the parameter range is 100-60000.

| Data field | Description            | Data                     |
|------------|------------------------|--------------------------|
| DATA[0]    | 指令字节<br>Command byte   | 0x42                     |
| DATA[1]    | 功能索引<br>Function index | DATA[1] = (uint8_t)index |
| DATA[2]    | NULL                   | 0x00                     |

|         |                                     |                                |
|---------|-------------------------------------|--------------------------------|
| DATA[3] | NULL                                | 0x00                           |
| DATA[4] | 加速度低字节 1<br>Acceleration low byte 1 | DATA[4] = (uint8_t)(Accel)     |
| DATA[5] | 加速度字节 2<br>Acceleration byte 2      | DATA[5] = (uint8_t)(Accel>>8)  |
| DATA[6] | 加速度字节 3<br>Acceleration byte 3      | DATA[6] = (uint8_t)(Accel>>16) |
| DATA[7] | 加速度字节 4<br>Acceleration byte 4      | DATA[7] = (uint8_t)(Accel>>24) |

#### 2.4.4. 功能索引说明

##### Function Index Description

| 索引值<br>Indexvalue | 指令名称<br>Commandname                       | 功能说明<br>Functiondescription  |
|-------------------|---|--|
| 0x00              | 位置规划加速度<br>Position planning acceleration | 位置规划中初始速度到最大速度的加速度值<br>Acceleration value from initial velocity to maximum velocity in position planning   |
| 0x01              | 位置规划减速度<br>Position planning deceleration | 位置规划中从最大速度到停止的减速度值<br>Deceleration value from maximum velocity to standstill in position planning  |
| 0x02              | 速度规划加速度<br>Speed planning acceleration    | 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<br>The acceleration value from the current speed to the target speed, including the acceleration in the positive and negative directions |
| 0x03              | 速度规划减速度<br>Speed planning deceleration    | 在相同方向上，从当前速度减速到目标速度的减速度值<br>The deceleration value to decelerate from the current velocity to the target velocity in the same  |

| 索引值<br>Indexvalue | 指令名称<br>Commandname | 功能说明<br>Functiondescription |
|-------------------|---------------------|-----------------------------|
|                   |                     | direction                   |

## 2.4.5. 通讯示例

### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | length | 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。

**Description:** Send a command to read the position planning acceleration.

回复指令:

Reply command:

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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] is 0x00, indicating the position planning acceleration value.

Data[4] to data[7] form one (Data[4] is the lowest bit, Data[7] is the highest bit) 32-bit data is 0x00002710, which means 10000 in decimal. It means that the acceleration of the motor position loop is 10000dps/s.

#### 示例 2:

#### 发送指令:

#### Example 2:

#### Send command:

#### 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:

| Frame header | ID   | Length | 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。

**Description:** Send a command to read the position planning deceleration.

#### 回复指令:

#### Reply command:

#### 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:

| Frame | ID | Length | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | CRC16L | CRC16H |
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|



|        |      |      |      |      |      |      |      |      |      |      |        |        |
|--------|------|------|------|------|------|------|------|------|------|------|--------|--------|
| header |      |      |      |      |      |      |      |      |      |      |        |        |
| 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。

#### Description:

Data[1] is 0x01, indicating the position planning deceleration value.

Data[4] to data[7] form a (Data[4] is the lowest bit, Data[7] is the highest bit) 32-bit data is 0x00002710, which means 10000 in decimal. It means that the deceleration of the motor position loop is 10000dps/s.

## 2.5. 写入加减速度到 RAM 和 ROM 指令 (0x43)

### Write Acceleration to RAM and ROM Command (0x43)

#### 2.5.1. 指令说明

##### Instruction Description

主机发送该指令写入加减速度到 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。指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。只有在电机失能时, 才能将参数成功保存到 ROM。

The host sends this command to write the acceleration into the RAM and ROM, which can be saved after power off. Acceleration data Accel is uint32\_t type, the unit is 1dps/s, and the parameter range is 100-60000. The command contains the acceleration and deceleration values in the position and velocity planning, which are determined by the index value. For details, see the index description table in 2.5.4. Be careful to avoid writing parameters when the motor has just started and is in motion. The parameters can be successfully saved to ROM only when the motor is disabled.

### 2.5.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description                         | Data                           |
|------------|-------------------------------------|--------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0x43                           |
| DATA[1]    | 功能索引<br>Function index              | DATA[1] = (uint8_t)index       |
| DATA[2]    | NULL                                | 0x00                           |
| DATA[3]    | NULL                                | 0x00                           |
| DATA[4]    | 加速度低字节 1<br>Acceleration low byte 1 | DATA[4] = (uint8_t)(Accel)     |
| DATA[5]    | 加速度字节 2<br>Acceleration byte 2      | DATA[5] = (uint8_t)(Accel>>8)  |
| DATA[6]    | 加速度字节 3<br>Acceleration byte 3      | DATA[6] = (uint8_t)(Accel>>16) |
| DATA[7]    | 加速度字节 4<br>Acceleration byte 4      | DATA[7] = (uint8_t)(Accel>>24) |

### 2.5.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，回复指令和接收指令一致。

The motor will reply to the host after receiving the command, and the reply command is the same as the received command.

### 2.5.4. 功能索引说明

#### Function Index Description

| 索引值<br>Index value | 指令名称、<br>Command name        | 功能说明<br>Function description                                       |
|--------------------|------------------------------|--|
| 0x00               | 位置规划加速度<br>position planning | 位置规划中初始速度到最大速度的加速度值<br>Acceleration value from initial velocity to |

|      |   |  |
|------|---|--|
|      | acceleration                              | maximum velocity in position planning  |
| 0x01 | 位置规划减速度<br>Position planning deceleration | 位置规划中从最大速度到停止的减速度值<br>Deceleration value from maximum speed to stop in position planning   |
| 0x02 | 速度规划加速度<br>Speed planning acceleration    | 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<br>The acceleration value from the current speed to the target speed, including the acceleration in the forward and reverse directions |
| 0x03 | 速度规划减速度<br>Speed planning deceleration    | 在相同方向上，从当前速度减速到目标速度的减速度值<br>In the same direction, the deceleration value from the current speed to the target speed   |

### 2.5.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] is 0x00, indicating the position planning acceleration value. Data[4] to data[7] form one 32-bit data is 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. It indicates that the position planning acceleration of 10000dps/s is written to the motor driver, and the value can be saved after the power is turned off.

回复指令:

Reply command:

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:

| Frame header | ID   | Length | 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：电机在收到指令后回复主机，回复指令和接收指令一致。

**Description:** The motor replies to the host after receiving the command, and the reply command is the same as the received command.

示例 2:

发送指令:

Example 2:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

**Description:**

Data[1] is 0x01, indicating the deceleration value of position planning.

Data[4] to data[7] form a 32-bit data, 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. Indicates that the position planning deceleration of 10000dps/s is written to the motor driver, and the value can be saved after the power is turned off.

**回复指令:**

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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:**

电机在收到指令后回复主机, 回复指令和接收指令一致。

**Description:**

The motor replies to the host computer after receiving the command, and the reply command is the same as the received command.

### 示例 3:

#### 发送指令:

#### Example 3:

#### Send command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] is 0x02, which indicates the acceleration value of speed planning.

Data[4] to data[7] form a 32-bit data, 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. Indicates that the speed planning acceleration of 10000dps/s is written to the motor driver, and the value can be saved after power off.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:

电机在收到指令后回复主机，回复指令和接收指令一致。

#### Description:

The motor replies to the host after receiving the command, and the reply command is the same as the received command.

示例 4:

发送指令:

Example 4:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] is 0x03, indicating the speed planning deceleration value.

Data[4] to data[7] form one 32-bit data, 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. Indicates that the speed



planning deceleration of 10000dps/s is written to the motor driver, and the value can be saved after the power is turned off.

回复指令:

Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:

电机在收到指令后回复主机，回复指令和接收指令一致。

Description:

The motor replies to the host after receiving the command, and the reply command is the same as the received command.

## 2.6. 读取多圈编码器位置数据指令 (0x60)

Read Multi-Turn Encoder Position Data Command (0x60)

### 2.6.1. 指令说明

Instruction Description

主机发送该指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。

The host sends this command to read the multi-turn position of the encoder, which represents the rotation angle of the motor output shaft.

### 2.6.2. 发送数据域定义

Send Data Field Definition

| Data field | Description | Data |
|------------|-------------|------|
| DATA[0]    | 指令字节        | 0x60 |



|         |              |      |
|---------|--------------|------|
|         | Command byte |      |
| 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

#### Reply Data Field Definition

电机在收到指令后回复主机，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。

The motor replies to the host after receiving the command, and the frame data contains the following parameters. Encoder multi-turn position encoder (int32\_t type, value range of multi-turn encoder, 4 bytes of valid data), which is the value after subtracting the encoder's multi-turn zero offset (initial position) from the original position of the encoder.

| Data field | Description                                  | Data                              |
|------------|--|-----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                         | 0x60                              |
| DATA[1]    | NULL   | 0x00                              |
| DATA[2]    | NULL   | 0x00                              |
| DATA[3]    | NULL   | 0x00                              |
| DATA[4]    | 编码器位置低字节 1<br>Encoder position low<br>byte 1 | DATA[4] = (uint8_t)(encoder)      |
| DATA[5]    | 编码器位置字节 2                                    | DATA[5] = (uint8_t)(encoder >> 8) |

|         |                                      |                                  |
|---------|--------------------------------------|----------------------------------|
|         | Encoder position byte 2              |                                  |
| DATA[6] | 编码器位置字节 3<br>Encoder position byte 3 | DATA[6] = (uint8_t)(encoder>>16) |
| DATA[7] | 编码器位置字节 4<br>Encoder position byte 4 | DATA[7] = (uint8_t)(encoder>>24) |

#### 2.6.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:

主机发送该指令以读取编码器多圈的位置。

Description:

The host sends this command to read the multi-turn position of the encoder.

回复指令:

Reply command:

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:

| Frame header | ID   | Length | 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。

#### Description:

Data[4] to data[7] form one 32-bit data, 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 10000 pulses.

## 2.7. 读取多圈编码器原始位置数据指令 (0x61)

### Read Multi-Turn Encoder Original Position Data Command (0x61)

#### 2.7.1. 指令说明

##### Instruction Description

主机发送该指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。

The host sends this command to read the multi-turn encoder home position, ie the multi-turn encoder value without the zero offset (home position).

#### 2.7.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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

#### Reply Data Field Definition

电机在收到指令后回复主机，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)。

The motor replies to the host after receiving the command, and the frame data contains the following parameters. Encoder multi-turn raw position encoderRaw (int32\_t type, value range, valid data 4 bytes).

| Data field | Description  | Data                                |
|------------|--|-------------------------------------|
| DATA[0]    | 指令字节<br>Command byte                               | 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<br>Encoder original position<br>byte 1 | DATA[4] = (uint8_t)(encoderRaw)     |
| DATA[5]    | 编码器原始位置字节 2<br>Encoder original position<br>byte 2 | DATA[5] = (uint8_t)(encoderRaw>>8)  |
| DATA[6]    | 编码器原始位置字节 3<br>Encoder original position<br>byte 3 | DATA[6] = (uint8_t)(encoderRaw>>16) |
| DATA[7]    | 编码器原始位置字节 4<br>Encoder original position           | DATA[7] = (uint8_t)(encoderRaw>>24) |

|  |        |  |
|--|--------|--|
|  | byte 4 |  |
|--|--------|--|

## 2.7.4. 通讯示例

### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:

主机发送该指令以读取编码器多圈的原始位置。

Description:

The host sends this command to read the original position of the encoder multi-turn.

回复指令:

Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[4] to data[7] form a 32-bit data, 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. Indicates that the current multi-turn encoder value of the motor is 10000 pulses, excluding the zero offset (initial position).

## 2.8. 读取多圈编码器零偏数据指令（0x62）

### Read Multi-Turn Encoder Zero Offset Data Command (0x62)

#### 2.8.1. 指令说明

##### Instruction Description

主机发送该指令以读取编码器多圈的零偏值（初始位置）。

The host sends this command to read the multi-turn zero offset value (initial position) of the encoder.

#### 2.8.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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

##### Reply Data Field Definition

电机在收到指令后回复主机，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）。

The motor replies to the host after receiving the command, and the frame data contains the following parameters. Encoder multi-turn zero offset encoderOffset (int32\_t type, value range, valid data 4 bytes).

| Data field | Description                        | Data                                   |
|------------|------------------------------------|--|
| DATA[0]    | 指令字节<br>Command byte               | 0x62                                   |
| DATA[1]    | NULL                               | 0x00                                   |
| DATA[2]    | NULL                               | 0x00                                   |
| DATA[3]    | NULL                               | 0x00                                   |
| DATA[4]    | 编码器零偏字节 1<br>Encoder offset byte 1 | DATA[4] = (uint8_t)(encoderOffset)     |
| DATA[5]    | 编码器零偏字节 2<br>Encoder offset byte 2 | DATA[5] = (uint8_t)(encoderOffset>>8)  |
| DATA[6]    | 编码器零偏字节 3<br>Encoder offset byte 3 | DATA[6] = (uint8_t)(encoderOffset>>16) |
| DATA[7]    | 编码器零偏字节 4<br>Encoder offset byte 4 | DATA[7] = (uint8_t)(encoderOffset>>24) |

#### 2.8.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:

主机发送该指令以读取编码器多圈的零偏值。

#### Description:

The host sends this command to read the multi-turn zero offset value of the encoder.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[4] to data[7] form one 32-bit data, 0x00002710, (Data[4] is the lowest bit, Data[7] is the highest bit), which means 10000 in decimal. It indicates that the current multi-turn encoder zero offset value of the motor is 10000 pulses.



## 2.9. 写入编码器多圈值到 ROM 作为电机零点指令 (0x63)

Write Encoder Multi-Turn Value to ROM as Motor Zero Command  
(0x63)

### 2.9.1. 指令说明

#### Instruction Description

主机发送该指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。只有在电机失能时，才能将参数成功保存到 ROM。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。

The host sends this command to set the zero offset (initial position) of the encoder, where the encoder multi-turn value to be written, encoderOffset, is of type int32\_t, (value range, 4 bytes of valid data). Be careful to avoid writing parameters when the motor has just started and is in motion. The parameters can be successfully saved to ROM only when the motor is disabled.

**Note:** After writing the position of the new zero point, the motor needs to be restarted to be effective. Because of the change of the zero offset, the new zero offset (initial position) should be used as a reference when setting the target position.

### 2.9.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description                         | Data                               |
|------------|-------------------------------------|------------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0x63                               |
| DATA[1]    | NULL                                | 0x00                               |
| DATA[2]    | NULL                                | 0x00                               |
| DATA[3]    | NULL                                | 0x00                               |
| DATA[4]    | 编码器零偏低字节 1<br>Encoder zero bias low | DATA[4] = (uint8_t)(encoderOffset) |

|         |                                       |                                       |
|---------|---------------------------------------|---------------------------------------|
|         | byte 1                                |                                       |
| DATA[5] | 编码器零偏字节 2<br>Encoder zero bias byte 2 | DATA[5] = (uint8_t)(encoderOffset>>8) |
| DATA[6] | 编码器零偏字节 3<br>Encoder zero bias byte 3 | DATA[6] = (uint8_t)(encoderOffset>>8) |
| DATA[7] | 编码器零偏字节 4<br>Encoder zero bias byte 4 | DATA[7] = (uint8_t)(encoderOffset>>8) |

### 2.9.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据和主机发送的指令相同。

The motor replies to the host after receiving the command, and the frame data is the same as the command sent by the host.

### 2.9.4. 通讯示例

#### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

### Description:

Data[4] to data[7] form one 32-bit data is 0x00002710,(Data[4] is the lowest bit,Data[7] is the highest bit),which means 10000 in decimal. It means to write 10000 pulses as multi-turn encoder zero offset.

回复指令:

### Reply command:

#### 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:

| Frame header | ID   | Length | 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:

电机在收到指令后回复主机，该帧数据和主机发送的指令相同。

### Description:

The motor replies to the host after receiving the command,and the frame data is the same as the command sent by the host.

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

Write the Current Multi-Turn Position of the Encoder to the ROM as the Motor Zero Command (0x64)

### 2.10.1. 指令说明

#### Instruction Description

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。只有在电机失能时，才能将参数成功保存到 ROM。

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。

Write the current encoder position of the motor as the multi-turn encoder zero offset (initial position) into the ROM. The parameters can be successfully saved to ROM only when the motor is disabled.

**Note:** After writing the new zero point position, you need to send 0x76 (system reset command) to restart the system to be effective. Because of the change of the zero offset, the new zero offset (initial position) should be used as a reference when setting the target position.

### 2.10.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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

#### Reply Data Field Definition

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

The motor replies to the host after receiving the command, and the encoderOffset in the data is the set zero offset value.

| Data field | Description | Data |
|------------|-------------|------|
| DATA[0]    | 指令字节        | 0x64 |

|         |  |  |
|---------|--|--|
|         | Command byte                               |  |
| DATA[1] | NULL                                       | 0x00                                   |
| DATA[2] | NULL                                       | 0x00                                   |
| DATA[3] | NULL                                       | 0x00                                   |
| DATA[4] | 编码器零偏低字节 1<br>Encoder zero bias low byte 1 | DATA[4] = (uint8_t)(encoderOffset)     |
| DATA[5] | 编码器零偏字节 2<br>Encoder zero bias byte 2      | DATA[5] = (uint8_t)(encoderOffset>>8)  |
| DATA[6] | 编码器零偏字节 3<br>Encoder zero bias byte 3      | DATA[6] = (uint8_t)(encoderOffset>>16) |
| DATA[7] | 编码器零偏字节 4<br>Encoder zero bias byte 4      | DATA[7] = (uint8_t)(encoderOffset>>24) |

#### 2.10.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | length | 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, 电机将当前多圈编码器值作为零偏(初始位置)写入 ROM

中。

#### Description:

After sending the 0x64 command, the motor will write the current multi-turn encoder value as the zero offset (initial position) into the ROM.

回复指令:

#### Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[4] to data[7] form a 32-bit data(Data[4] is the lowest bit,Data[7] is the highest bit),0x00002710,which means 10000 in decimal. Indicates that the multi-turn zero offset value (initial position) written to the motor is 10,000 pulses.

## 2.11. 读取多圈角度指令 (0x92)

### Read Multi-Turn Angle Command (0x92)

#### 2.11.1. 指令说明

##### Instruction Description

主机发送该指令以读取当前电机的多圈绝对角度值。

The host sends this command to read the current multi-turn absolute angle value of the motor.

### 2.11.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.11.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据中包含了以下参数。

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

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°/LSB。

Motor angle motorAngle, (int32\_t type, value range, valid data 4 bytes), unit 0.01°/LSB.

| Data field | Description          | Data                            |
|------------|----------------------|---------------------------------|
| DATA[0]    | 指令字节<br>Command byte | 0x92                            |
| DATA[1]    | NULL                 | 0x00                            |
| DATA[2]    | NULL                 | 0x00                            |
| DATA[3]    | NULL                 | 0x00                            |
| DATA[4]    | 角度低字节 1              | DATA[4] = (uint8_t)(motorAngle) |



|         |                        |                                     |
|---------|------------------------|-------------------------------------|
|         | Angle low byte 1       |                                     |
| DATA[5] | 角度字节 2<br>Angle byte 2 | DATA[5] = (uint8_t)(motorAngle>>8)  |
| DATA[6] | 角度字节 3<br>Angle byte 3 | DATA[6] = (uint8_t)(motorAngle>>16) |
| DATA[7] | 角度字节 4<br>Angle byte 4 | DATA[7] = (uint8_t)(motorAngle>>24) |

#### 2.11.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

Description:

After sending the 0x92 command, it will return the absolute angle of the motor output shaft.

回复指令:

Reply command:

CAN:



| ID    | 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:

| Frame header | ID   | Length | 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°。

#### Description:

Data[4] to data[7] form one 32-bit data,it is 0x00008CA0,(Data[4] is the lowest bit,Data[7] is the highest bit),which means the decimal is 36000,which is reduced by 100 times in units of 0.01°/LSB That is 36000\*0.01=360°. Indicates that the motor output shaft moves 360° in the positive direction relative to the zero position.

## 2.12. 读取单圈角度指令（0x94）

### Read Single-Turn Angle Command (0x94)

#### 2.12.1. 指令说明

##### Instruction Description

主机发送该指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，电机单圈角度取值范围为-180° ~180°。

The host sends this command to read the current single-turn angle of the motor. Note that this command is used as a single-turn data read command for direct-drive motors. The single-turn angle range of the motor is - 180° to +180° .

#### 2.12.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description | Data |
|------------|-------------|------|
| DATA[0]    | 指令字节        | 0x94 |

|         |              |      |
|---------|--------------|------|
|         | Command byte |      |
| 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

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据中包含了以下参数。

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

1. 电机输出轴单圈角度 Single circle angle, (int32\_t 类型, 数值范围, 有效数据 4 个字节), 单位 0.01°/LSB。

Motor Output Shaft Single-Turn Angle (int32\_t type, numerical range, 4 bytes of valid data), unit: 0.01° /LSB..

| Data field | Description                                  | Data  |
|------------|--|---|
| DATA[0]    | 指令字节<br>Command byte                         | 0x94  |
| DATA[1]    | NULL   | 0x00  |
| DATA[2]    | NULL   | 0x00  |
| DATA[3]    | NULL   | 0x00  |
| DATA[4]    | 单圈角度低字节 1<br>Single-turn angle low<br>byte 1 | DATA[4] = (uint8_t)(Single circle angle)    |
| DATA[5]    | 单圈角度字节 2<br>Single-turn angle byte           | DATA[5] = (uint8_t)(Single circle angle>>8) |

|         |   |  |
|---------|---|--|
|         | 2                                       |  |
| DATA[6] | 单圈角度字节 3<br>Single-turn angle byte<br>3 | DATA[6] = (uint8_t)(Single circle angle>>16) |
| DATA[7] | 单圈角度字节 4<br>Single-turn angle byte<br>4 | DATA[7] = (uint8_t)(Single circle angle>>24) |

#### 2.12.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

Description:

After sending the 0x94 command, it will return the motor single-turn angle.

回复指令:

Reply command:

CAN:

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

|       |      |      |      |      |      |      |      |      |
|-------|------|------|------|------|------|------|------|------|
| 0x241 | 0x94 | 0x00 | 0x00 | 0x00 | 0x4C | 0x1D | 0x00 | 0x00 |
|-------|------|------|------|------|------|------|------|------|

#### RS485:

| Frame header | ID   | Length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x94 | 0x00 | 0x00 | 0x00 | 0x4C | 0x1D | 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 0x00001D4C，表示十进制为 7500，按照  $0.01^{\circ}/\text{LSB}$  单位缩小 100 倍即  $7500 \times 0.01 = 75^{\circ}$ 。表示电机输出轴电机输出轴单圈角度为  $75^{\circ}$ 。

#### Description:

Data[4] through Data[7] form a 32-bit value (where Data[4] is the LSB and Data[7] is the MSB). For example, a value of 0x00001D4C corresponds to 7500 in decimal. Applying the scaling factor of  $0.01^{\circ}/\text{LSB}$  ( $7500 \times 0.01$ ), the result is  $75^{\circ}$ . This indicates that the motor output shaft's single-turn angle is  $75^{\circ}$ .

## 2.13. 读取电机状态 1 和错误标志指令 (0x9A)

### Read Motor Status 1 and Error Flag Command (0x9A)

#### 2.13.1. 指令说明

##### Instruction Description

该指令读取当前电机的温度、电压和错误状态标志

This command reads the current motor temperature, voltage and error status flags.

#### 2.13.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.13.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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)。

The motor replies to the host after receiving the command, and the frame data contains the following parameters:

1. Motor temperature temperature (int8\_t type, unit 1°C/LSB);
2. Brake control command: Indicates the state of the brake control command, 1 represents the brake release command, and 0 represents the brake lock command;
3. Voltage (uint16\_t type, unit 0.1V/LSB);
4. Error flag errorState (of type uint16\_t, each bit represents a different motor state).

| Data field | Description               | Data                                      |
|------------|---------------------------|---|
| DATA[0]    | 指令字节<br>Command byte      | 0x9A                                      |
| DATA[1]    | 电机温度<br>Motor temperature | DATA[1] = (uint8_t)(temperature)          |
| DATA[2]    | MOS 温度<br>MOS temperature | DATA[2] = (uint8_t)(motorMOS temperature) |
| DATA[3]    | 抱闸释放指令<br>Brake release   | DATA[3] = (uint8_t)(RlyCtrlRslt)          |

|         |                                      |                                    |
|---------|--------------------------------------|------------------------------------|
|         | command                              |                                    |
| DATA[4] | 电压低字节<br>Voltage low byte            | DATA[4] = (uint8_t)(voltage)       |
| DATA[5] | 电压高字节<br>Voltage high byte           | DATA[5] = (uint8_t)(voltage>>8)    |
| DATA[6] | 错误状态低字节 1<br>Error status low byte 1 | DATA[6] = (uint8_t)(errorState)    |
| DATA[7] | 错误状态字节 2<br>Error status byte 2      | DATA[7] = (uint8_t)(errorState>>8) |

备注:

Remark:

1. 系统异常状态值 System\_errorState 状态表 1 如下:

System abnormal state value System\_errorState state table 1 is as follows:

| System_errorState | 状态说明<br>Status Description  |
|-------------------|---|
| 0x0002            | 电机堵转<br>当电机电流大于“堵转电流”参数，持续时间超过“堵转时间”参数，将触发电机堵转错误。此错误无法自动恢复。<br>Motor stall<br>A Motor Stall Error will be triggered if the motor current exceeds the "Stall Current" parameter for a duration longer than the "Stall Time" parameter. This error is non-auto-recoverable. |
| 0x0004            | 低压<br>当母线电压小于“欠压保护电压”参数，将触发低压错误。当母线电压大于“欠压保护电压”参数，此错误将自动恢复。   |

|        |  |
|--------|--|
|        | <p>Low voltage</p> <p>A Low Voltage Error will be triggered when the bus voltage drops below the "Undervoltage Protection Voltage" parameter. This error will automatically clear once the bus voltage rises above the "Undervoltage Protection Voltage" threshold.</p>  |
| 0x0008 | <p>过压</p> <p>当母线电压大于“过压保护电压”参数，将触发过压错误。当母线电压小于“过压保护电压”参数，此错误将自动恢复。</p> <p>Over voltage</p> <p>An Overvoltage Error will be triggered when the bus voltage exceeds the "Overvoltage Protection Voltage" parameter. This error will automatically recover when the bus voltage drops below the "Overvoltage Protection Voltage" threshold.</p> |
| 0x0010 | <p>相电流过流</p> <p>当电机电流大于“最大电流”参数，将触发相电流过流错误。此错误无法自动恢复。</p> <p>Over current</p> <p>A Phase Overcurrent Error will be triggered when the motor current exceeds the "Maximum Current" parameter. This error is latched and cannot be automatically recovered.</p>  |
| 0x0040 | <p>功率超限</p> <p>Power overrun</p>   |
| 0x0080 | <p>标定参数写入错误</p> <p>Calibration parameter writing error</p>   |
| 0x0100 | <p>超速</p> <p>当电机输入端速度大于“最大转速”参数，将触发超速错误。</p> <p>当电机输入端速度小于“最大转速”参数，此错误将自动</p>  |



|        |  |
|--------|--|
|        | <p>恢复。</p> <p><b>Overspeed</b></p> <p>An Overspeed Error will be triggered when the motor input speed exceeds the "Maximum Speed" parameter. This error will automatically recover when the input speed drops below the "Maximum Speed" threshold.</p>   |
| 0x0800 | <p>元器件过温</p> <p>当 PCB 板温度大于 110 度，将触发元器件过温错误。</p> <p>当 PCB 板温度小于 90 度，此错误将自动恢复。</p> <p><b>Component Overtemperature</b></p> <p>A Component Overtemperature Error will be triggered when the PCB temperature exceeds 110 ° C. This error will automatically recover once the PCB temperature drops below 90° C.</p>   |
| 0x1000 | <p>电机温度过温</p> <p>当电机定子温度大于“过温保护温度”参数，将触发电机温度过温错误。</p> <p>当电机定子温度小于“过温恢复温度”参数，此错误将自动恢复。</p> <p><b>Motor temperature over temperature</b></p> <p>A Motor Overtemperature Error will be triggered when the motor stator temperature exceeds the "Overtemperature Protection Temperature" parameter. This error will automatically recover when the stator temperature drops below the "Overtemperature Recovery Temperature" parameter.</p> |
| 0x2000 | <p>编码器校准错误</p> <p>当电机校准失败时，将触发编码器校准错误。此错误无法自动恢复。</p> <p><b>Encoder calibration error</b></p>   |



|        |   |
|--------|---|
|        | An Encoder Calibration Error will be triggered if the motor calibration fails. This error is non-auto-recoverable.  |
| 0x4000 | <p>编码器数据错误</p> <p>当电机编码器数据出现异常时，将触发编码器数据错误。此错误无法自动恢复。</p> <p>Encoder Data Error</p> <p>An Encoder Data Error will be triggered if the motor encoder data is abnormal. This error is non-auto-recoverable.</p> |

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。

When multiple errors occur at the same time, the error status bits will be displayed superimposed. For example, if the number 0x0016 appears, it means the addition of 0x2+0x4+0x10, which means that there are three errors such as motor stall, low voltage, and phase current over current.

#### 2.13.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID | Length | 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。

#### Description:

After sending the 0x9A command, the temperature, voltage and error status flags of the motor will be returned.

回复指令:

#### Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

Data[3] indicates that the brake indicates the state of the brake control command, 1

represents the brake release command, and 0 represents the brake lock command. So 0x01 indicates that the current brake release command has been executed.

Data[4] and Data[5] (Data[4] is the low bit, Data[5] is the high bit) form 0x01E5, the decimal is 485, which is reduced by 10 times according to the unit of 0.1V/LSB,  $485 \times 0.1 = 48.5V$ , representing The current motor supply voltage is 48.5V.

Data[6] and Data[7] (Data[6] is low and Data[7] is high) form 0x0004, which indicates a low-voltage error according to the error description in the System\_errorState table.

## 2.14. 读取电机状态 2 指令 (0x9C)

### Read Motor Status 2 Command (0x9C)

#### 2.14.1. 指令说明

##### Instruction Description

该指令读取当前电机的温度、转速、编码器位置。

This command reads the temperature, speed and encoder position of the current motor.

#### 2.14.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.14.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

The motor replies to the host after receiving the command,and the frame data contains the following parameters.

- 1.Motor temperature temperature (int8\_t type,1°C/LSB);
- 2.The torque current value iq of the motor (int16\_t type,0.01A/LSB);
- 3.Motor output shaft speed (int16\_t type,1dps/LSB);
- 4.Motor output shaft angle (int16\_t type,1degree/LSB,maximum range ±32767degree).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0x9C                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5]    | 电机速度高字节<br>Motor speed high byte    | DATA[5] = (uint8_t)(speed>>8)    |
| DATA[6]    | 电机角度低字节                             | DATA[6] = (uint8_t)(degree)      |

|         |                                  |                                |
|---------|----------------------------------|--------------------------------|
|         | Motor angle low byte             |                                |
| DATA[7] | 电机角度高字节<br>Motor angle high byte | DATA[7] = (uint8_t)(degree>>8) |

#### 2.14.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:

该指令读取当前电机的温度、转速、编码器位置。

Description:

This command reads the current temperature, speed and encoder position of the motor.

回复指令:

Reply command:

CAN:

| ID    | 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:

| Frame | ID | Length | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | CRC16L | CRC16H |
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|

|        |      |      |      |      |      |      |      |      |      |      |        |        |
|--------|------|------|------|------|------|------|------|------|------|------|--------|--------|
| header |      |      |      |      |      |      |      |      |      |      |        |        |
| 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。

#### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3] 0x0064 is 100 in decimal, and it is  $100 \times 0.01 = 1A$  when scaled down by 100 times, which means that the actual current of the current motor is 1A.

The composite data 0x01F4 of Data[4] and Data[5] is 500 in decimal, which means the motor output shaft speed is 500dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7] 0x002D is 45 in decimal, which means that the motor output shaft moves 45 degrees in the positive direction relative to the zero position. The position of the motor output shaft is related to the number of lines of the motor encoder and the reduction ratio. For example, if the number of lines of the motor encoder is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.

## 2.15. 读取电机状态 3 指令 (0x9D)

Read Motor Status 3 Command (0x9D)

### 2.15.1. 指令说明

Instruction Description

该指令读取当前电机的温度和相电流数据。

This command reads the current motor temperature and phase current data.

### 2.15.2. 发送数据域定义

Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.15.3. 回复数据域定义

Reply Data Field Definition

电机在收到指令后回复主机，该帧数据包含了以下数据：

1. 电机温度 temperature (int8\_t 类型, 1°C/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。

The motor replies to the host after receiving the command, and the frame data contains the following data:

1. Motor temperature temperature (int8\_t type, 1°C/LSB);



2. Phase A current data,the data type is int16\_t,and the corresponding actual phase current is 0.01ALSB;

3.B-phase current data,the data type is int16\_t type,and the corresponding actual phase current is 0.01ALSB;

4. C-phase current data,the data type is int16\_t type,and the corresponding actual phase current is 0.01ALSB.

| Data field | Description                           | Data                             |
|------------|---------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                  | 0x9D                             |
| DATA[1]    | 电机温度<br>Motor temperature             | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | A 相电流低字节<br>Phase A current low byte  | DATA[2] = (uint8_t)(iA)          |
| DATA[3]    | A 相电流高字节<br>Phase A current high byte | DATA[3] = (uint8_t)(iA>>8)       |
| DATA[4]    | B 相电流低字节<br>Phase B current low byte  | DATA[4] = (uint8_t)(iB)          |
| DATA[5]    | B 相电流高字节<br>Phase B current high byte | DATA[5] = (uint8_t)(iB>>8)       |
| DATA[6]    | C 相电流低字节<br>Phase C current low byte  | DATA[6] = (uint8_t)(iC)          |
| DATA[7]    | C 相电流高字节<br>Phase C current high byte | DATA[7] = (uint8_t)(iC>>8)       |

#### 2.15.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**



**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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:**

该指令读取当前电机的温度和相电流数据。

**Description:**

This command reads the current motor temperature and phase current data.

**回复指令:**

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data 0x0BC2 of Data[2] and Data[3] is 3010 in decimal, and it is  $3010 \times 0.01 = 30.1\text{A}$  when scaled down by 100 times, which means that the actual current of the current phase A of the motor is 30.1A.

The composite data 0xFA10 of Data[4] and Data[5] is -1520 in decimal, and it is  $-1520 \times 0.01 = -15.2\text{A}$  when scaled down by 100 times, which means that the actual current of the current phase B of the motor is -15.2A.

The composite data 0xF9C0 of Data[6] and Data[7] is -1600 in decimal, and it is  $-1600 \times 0.01 = -16\text{A}$  when scaled down by 100 times, which means that the actual current of the current phase C of the motor is -16A.

## 2.16. 电机关闭指令 (0x80)

### Motor Shutdown Command (0x80)

#### 2.16.1. 指令说明

##### Instruction Description

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。

Turns off the motor output and also clears the motor running state, not in any closed loop mode.

#### 2.16.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.16.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，帧数据和主机发送相同。

The motor replies to the host after receiving the command, and the frame data is the same as that sent by the host.

## 2.17. 电机停止指令 (0x81)

### Motor Stop Command (0x81)

#### 2.17.1. 指令说明

##### Instruction Description

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。

Stop the motor, the closed-loop mode where the motor is still running, just stop the motor speed.

#### 2.17.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.17.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，帧数据和主机发送相同。

The motor replies to the host after receiving the command, and the frame data is the same as that sent by the host

### 2.18. 转矩闭环控制指令 (0xA1)

#### Torque Closed-Loop Control Command (0xA1)

##### 2.18.1. 指令说明

##### Instruction Description

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。主机发送该指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。

This command is a control command, which can be run when the motor is not faulty. The host sends this command to control the torque and current output of the motor. The control value iqControl is of type int16\_t and the unit is 0.01A/LSB.

For safety reasons, This command cannot open the brake directly. But, you can use the 0x77 command to open the brake first, then you can use A1 command.

##### 2.18.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description                          | Data                           |
|------------|--------------------------------------|--------------------------------|
| DATA[0]    | 指令字节<br>Command byte                 | 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<br>Torque current control | DATA[4] = (uint8_t)(iqControl) |

|         |   |                                   |
|---------|---|-----------------------------------|
|         | value low byte  |                                   |
| DATA[5] | 转矩电流控制值高字节<br>Torque current control<br>value high byte | DATA[5] = (uint8_t)(iqControl>>8) |
| DATA[6] | NULL  | 0x00                              |
| DATA[7] | NULL  | 0x00                              |

### 2.18.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

1. Motor temperature temperature (int8\_t type, 1°C/LSB).
2. The torque current value iq of the motor (int16\_t type, 0.01A/LSB).
3. Motor output shaft speed (int16\_t type, 1dps/LSB).
4. Motor output shaft angle (int16\_t type, 1degree/LSB, maximum range ±32767degree).

| Data field | Description                        | Data                             |
|------------|------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte               | 0xA1                             |
| DATA[1]    | 电机温度<br>Motor temperature          | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节                            | DATA[3] = (uint8_t)(iq>>8)       |

|         |                                  |                                |
|---------|----------------------------------|--------------------------------|
|         | Torque current high byte         |                                |
| DATA[4] | 电机速度低字节<br>Motor speed low byte  | DATA[4] = (uint8_t)(speed)     |
| DATA[5] | 电机速度高字节<br>Motor speed high byte | DATA[5] = (uint8_t)(speed>>8)  |
| DATA[6] | 电机角度低字节<br>Motor angle low byte  | DATA[6] = (uint8_t)(degree)    |
| DATA[7] | 电机角度高字节<br>Motor angle low byte  | DATA[7] = (uint8_t)(degree>>8) |

#### 2.18.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

Data[4] and data[5] represent the data size, Data[4] (0x64) is the low bit, and Data[5]

(0x00) is the high bit. So the actual data is 0x0064, which means decimal 100, which is  $100 \times 0.01 = 1A$  when reduced by 0.01A/LSB. Driving will be performed with 1A as the target current.

回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3], 0x0064, is 100 in decimal, and it is  $100 \times 0.01 = 1A$  when scaled down by 100 times, which means that the actual current of the



current motor is 1A.

The composite data 0x01F4 of Data[4] and Data[5] is 500 in decimal, which means the motor output shaft speed is 500dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7] 0x002D is 45 in decimal, which means that the motor output shaft moves 45 degrees in the positive direction relative to the zero position. The position of the motor output shaft is related to the number of lines of the motor encoder and the reduction ratio. For example, if the number of lines of the motor encoder is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 * 6 = 98304$  pulses.

**示例 2:**

**发送指令:**

**Example 2:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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  $-100 * 0.01 = -1A$ 。驱动会以 -1A 作为目标电流执行。

**Description:**

Data[4] and data[5] represent the data size, Data[4] (0x9C) is the low bit, Data[5]



(0xFF) is the high bit. So the actual data is 0xFF9C, which means decimal -100, which is  $-100 \times 0.01 = -1A$  when reduced by 0.01A/LSB. The drive will be performed with -1A as the target current.

回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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  $-100 \times 0.01 = -1A$ , 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。

**Description:**

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3] 0xFF9C is -100 in decimal, and it is

$-100 \times 0.01 = -1\text{A}$  when scaled down by 100 times, which means that the actual current of the current motor is  $-1\text{A}$ .

The composite data  $0\text{xFE0C}$  of Data[4] and Data[5] is  $-500$  in decimal, which means the motor output shaft speed is  $-500\text{dps}$ . There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7]  $0\text{xFFD3}$  is  $-45$  in decimal, which means that the motor output shaft moves in the opposite direction by  $-45$  degrees relative to the zero position. The position of the motor output shaft is related to the number of motor encoder lines and the reduction ratio. For example, if the number of motor encoder lines is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.

## 2.19. 速度闭环控制指令 (0xA2)

### Speed Closed-Loop Control Command (0xA2)

#### 2.19.1. 指令说明

##### Instruction Description

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。主机发送该指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  $0.01\text{dps/LSB}$ ，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  $0 \sim 255$ ，以额定电流的百分比为单位，即  $1\% \times \text{额定电流/LSB}$ 。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。

This command is a control command, which can be run when the motor is not faulty. The host sends this command to control the speed of the motor output shaft. The control value `speedControl` is `int32_t` type, and the corresponding actual speed is  $0.01\text{dps/LSB}$ . The control value `maxTorque` limits the maximum torque of the motor output shaft. It is `uint8_t` type, with a value range of 0 to 255. The unit is a percentage of the rated current, specifically 1% of the rated current per LSB (Least Significant Bit). If

the given current is 0 or greater than the stall current, the force control mode will not be activated. The maximum torque current of the motor will then be limited by the motor stall current value set in the setup software.

### 2.19.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description                        | Data                                    |
|------------|------------------------------------|---|
| DATA[0]    | 指令字节<br>Command byte               | 0xA2                                    |
| DATA[1]    | 最大扭矩<br>Max torque                 | DATA[2] = (uint8_t)(maxTorque)          |
| DATA[2]    | NULL                               | 0x00                                    |
| DATA[3]    | NULL                               | 0x00                                    |
| DATA[4]    | 速度控制低字节<br>Speed control low byte  | DATA[4] = (uint8_t)(speedControl)       |
| DATA[5]    | 速度控制<br>Speed control              | DATA[5] = (uint8_t)(speedControl >> 8)  |
| DATA[6]    | 速度控制<br>Speed control              | DATA[6] = (uint8_t)(speedControl >> 16) |
| DATA[7]    | 速度控制高字节<br>Speed control high byte | DATA[7] = (uint8_t)(speedControl >> 24) |

备注：

1. 该指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。

#### Remark:

1. The maximum torque current of the motor under this command is limited by the Max Torque Current value in the host computer;
2. In this control mode, the maximum acceleration of the motor is limited by the

Max Acceleration value in the host computer;

3. When the speed loop acceleration value is 0,the speed loop acceleration is limited by the maximum current output capability.

### 2.19.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

The motor replies to the host after receiving the command,and the frame data contains the following parameters.

1. Motor temperature temperature (int8\_t type,1°C/LSB);
2. The torque current value iq of the motor (int16\_t type,0.01A/LSB);
3. Motor output shaft speed (int16\_t type,1dps/LSB);
4. Motor output shaft angle (int16\_t type,1degree/LSB,maximum range±32767 degree).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0xA2                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5]    | 电机速度高字节                             | DATA[5] = (uint8_t)(speed>>8)    |

|         |                                  |                                |
|---------|----------------------------------|--------------------------------|
|         | Motor speed high byte            |                                |
| DATA[6] | 电机角度低字节<br>Motor angle low byte  | DATA[6] = (uint8_t)(degree)    |
| DATA[7] | 电机角度高字节<br>Motor angle high byte | DATA[7] = (uint8_t)(degree>>8) |

#### 2.19.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

Description:

Data[4] to data[7] form one (Data[4] is the lowest bit, Data[7] is the highest bit) 32-bit data is 0x00002710, which means 10000 in decimal. The sending command is reduced by 100 times according to 0.01dps/LSB, that is,  $10000 \times 0.01 = 100\text{dps}$ . The drive operates at the target speed of 100dps of the motor output shaft.

回复指令:

## Reply command:

### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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  $100 \times 0.01 = 1A$ ，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。

### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3] 0x0064 is 100 in decimal, and it is  $100 \times 0.01 = 1A$  when scaled down by 100 times, which means that the actual current of the current motor is 1A.

The composite data 0x01F4 of Data[4] and Data[5] is 500 in decimal, which means the motor output shaft speed is 500dps. There is a reduction ratio relationship between

the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7] 0x002D is 45 in decimal, which means that the motor output shaft moves 45 degrees in the positive direction relative to the zero position. The position of the motor output shaft is related to the number of lines of the motor encoder and the reduction ratio. For example, if the number of lines of the motor encoder is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 * 6 = 98304$  pulses.

## 示例 2:

发送指令:

## Example 2:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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  $-10000 * 0.01 = -100\text{dps}$ 。驱动以电机输出轴-100dps 的速度为目标速度运行。

## Description:

Data[4] to data[7] form one (Data[4] is the lowest bit, Data[7] is the highest bit) 32-bit data is 0xFFFFD8F0, which means -10000 in decimal. The sending command is reduced by 100 times according to 0.01dps/LSB, that is  $-10000 * 0.01 = -100\text{dps}$ . The drive runs at the target speed of the motor output shaft -100dps.



回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

Data[1] = 0x32 is 50 in decimal,which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3] 0xFF9C is -100 in decimal,and it is -100\*0.01= -1A when scaled down by 100 times,which means that the actual current of the current motor is -1A.

The composite data 0xFE0C of Data[4] and Data[5] is -500 in decimal,which means the motor output shaft speed is -500dps. There is a reduction ratio relationship between



the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7], 0xFFD3, is -45 in decimal, which means that the motor output shaft moves in the opposite direction by -45 degrees relative to the zero position. The position of the motor output shaft is related to the number of motor encoder lines and the reduction ratio. For example, if the number of motor encoder lines is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 * 6 = 98304$  pulses.

## 2.20. 绝对位置闭环控制指令 (0xA4)

### Absolute Position Closed-Loop Control Command (0xA4)

#### 2.20.1. 指令说明

##### Instruction Description

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。主机发送该指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  $0.01\text{degree/LSB}$ ，即 36000 代表  $360^\circ$ ，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  $1\text{dps/LSB}$ 。

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。

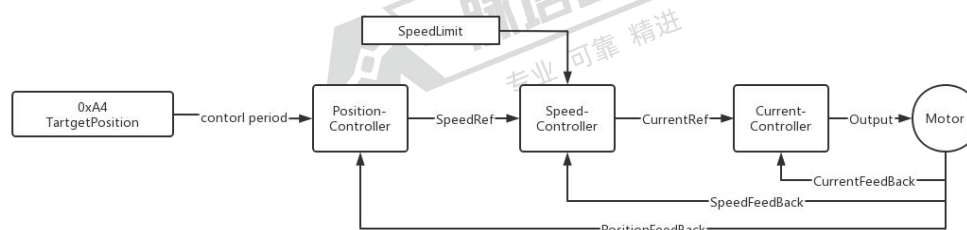


图 2-1 带速度限制的位置跟踪模式框图

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 maxSpeed 限制了位置运行过程中的最大速度，加速度由位置环设置的加速度决定。

This command is a control command, which can be run when the motor is not faulty. The host sends this command to control the position of the motor (multi-turn angle). The control value angleControl is int32\_t type, and the corresponding actual position is 0.01degree/LSB, that is, 36000 represents 360°, and the rotation direction of the motor is determined by the difference between the target position and the current position. The control value maxSpeed limits the maximum speed of the motor output shaft rotation, which is of type uint16\_t, corresponding to the actual speed of 1dps/LSB.

According to the position planning acceleration value set by the system, different operating modes will be different:

1. If the position loop acceleration is 0, then the position loop will enter the direct tracking mode, and directly track the target position through the PI controller. Among them, maxSpeed limits the maximum speed during the position operation process. If the maxSpeed value is 0, then it is completely output by the calculation result of the PI controller. As shown below.

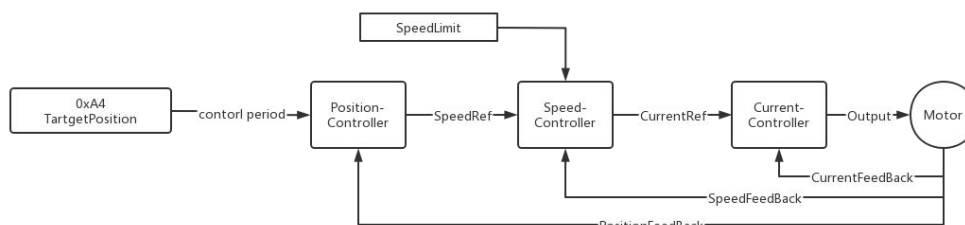


Figure 2-1 Block Diagram of Position Tracking Mode with Speed Limit

2. If the position loop acceleration is non-zero, the motor will operate in a velocity-profiled motion mode, where the motor handles the acceleration and deceleration phases. In this mode, "maxSpeed" limits the peak velocity during movement, while the actual acceleration is determined by the position loop's acceleration setting.

### 2.20.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description | Data |
|------------|-------------|------|
|------------|-------------|------|

|         |                                       |                                       |
|---------|---------------------------------------|---------------------------------------|
| DATA[0] | 指令字节<br>Command byte                  | 0xA4                                  |
| DATA[1] | NULL                                  | 0x00                                  |
| DATA[2] | 速度限制低字节<br>Speed limit low byte       | DATA[2] = (uint8_t)(maxSpeed)         |
| DATA[3] | 速度限制高字节<br>Speed limit high byte      | DATA[3] = (uint8_t)(maxSpeed>>8)      |
| DATA[4] | 位置控制低字节<br>osition control low byte   | DATA[4] = (uint8_t)(angleControl)     |
| DATA[5] | 位置控制<br>Position control              | DATA[5] = (uint8_t)(angleControl>>8)  |
| DATA[6] | 位置控制<br>Position control              | DATA[6] = (uint8_t)(angleControl>>16) |
| DATA[7] | 位置控制高字节<br>Position control high byte | DATA[7] = (uint8_t)(angleControl>>24) |

### 2.20.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

The motor replies to the host after receiving the command,and the frame data contains the following parameters.

1. Motor temperature temperature (int8\_t type,1°C/LSB).
2. The torque current value iq of the motor (int16\_t type,0.01A/LSB).
3. Motor output shaft speed (int16\_t type,1dps/LSB).
4. Motor output shaft angle (int16\_t type,1degree/LSB,maximum range ±32767 degree).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0xA4                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5]    | 电机速度高字节<br>Motor speed high byte    | DATA[5] = (uint8_t)(speed>>8)    |
| DATA[6]    | 电机角度低字节<br>Motor angle low byte     | DATA[6] = (uint8_t)(degree)      |
| DATA[7]    | 电机角度高字节<br>Motor angle high byte    | DATA[7] = (uint8_t)(degree>>8)   |

#### 2.20.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame | ID | Length | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | CRC16L | CRC16H |
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|

|        |      |      |      |      |      |      |      |      |      |      |        |        |
|--------|------|------|------|------|------|------|------|------|------|------|--------|--------|
| header |      |      |      |      |      |      |      |      |      |      |        |        |
| 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  $36000 \times 0.01 = 360^\circ$ 。电机会以输出轴相对零点位置正向移动  $360^\circ$ 。

### Description:

Data[2] and Data[3] form one (Data[2] is low,Data[3] is high) 16-bit data is 0x01F4,indicating the decimal 500dps motor output shaft speed. The drive will run the position loop at this speed as the maximum speed.

Data[4] to data[7] form a (Data[4] is the lowest bit,Data[7] is the highest bit) 32-bit data is 0x00008CA0,which means 36000 in decimal. The sending command is reduced by 100 times according to 0.01degree/LSB,that is, $36000 \times 0.01 = 360^\circ$ . The motor will move forward  $360^\circ$  with the output shaft relative to the zero position.

回复指令:

### Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data 0x0064 of Data[2] and Data[3] is 100 in decimal, which is  $100 \times 0.01 = 1A$  according to the 100-fold reduction, which means that the actual current of the motor is 1A.

The synthetic data 0x01F4 of Data[4] and Data[5] is 500 in decimal, which means the motor output shaft speed is 500dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, the motor speed is 6 times higher than the output shaft speed.

The composite data 0x002D of Data[6] and Data[7] is 45 in decimal, which means that the motor output shaft moves forward by 45 degrees relative to the zero position. The position of the motor output shaft is related to the number of lines of the motor encoder and the reduction ratio. For example, the number of lines of the motor encoder is 16384 and the reduction ratio is 6. Then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.

### 示例 2:

发送指令:

Example 2:

Send command:

CAN:



| ID    | 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:

| Frame header | ID   | Length | 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°。

#### Description:

Data[2] and Data[3] form one (Data[2] is low,Data[3] is high) 16-bit data is 0x01F4,indicating the decimal 500dps motor output shaft speed. The drive will run the position loop at this speed as the maximum speed.

Data[4] to data[7] form a (Data[4] is the lowest bit,Data[7] is the highest bit) 32-bit data is 0xFFFF7360,which means -36000 in decimal. The sending command is reduced by 100 times according to 0.01degree/LSB,that is,  $-36000 \times 0.01 = -360^\circ$ . The motor will move -360° in reverse with respect to the zero position of the output shaft.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID | Length | 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。

#### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

Data[2] and Data[3] synthesized data 0xFF9C is -100 in decimal, which is  $-100 \times 0.01 = -1A$  when scaled down by 100 times, which means the actual current of the motor is -1A.

The synthetic data 0xFE0C of Data[4] and Data[5] is -500 in decimal, which means that the motor output shaft speed is -500dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, the motor speed is 6 times higher than the output shaft speed.

The synthetic data 0xFFD3 of Data[6] and Data[7] is -45 in decimal, which means that the output shaft of the motor moves backward by -45 degrees relative to the zero position. The position of the motor output shaft is related to the number of lines of the motor encoder and the reduction ratio. For example, the number of lines of the motor encoder is 16384 and the reduction ratio is 6. Then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.



## 2.21. 单圈位置控制指令 (0xA6)

### Single-Turn Position Control Command (0xA6)

#### 2.21.1. 指令说明

##### Instruction Description

主机发送该指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°~359.99°；

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。

The host sends this command to control the position of the motor (single-turn angle). When the multi-lap save function is turned off,the default is single-lap mode. This instruction can be used in single-turn mode.

1. The angle control value angleControl is of uint16\_t type,the value range is 0~35999,and the corresponding actual position is 0.01degree/LSB,that is,the actual angle range is 0°~359.99°;

2. spinDirection sets the direction of motor rotation,which is uint8\_t type,0x00 means clockwise,and 0x01 means counterclockwise;

3. maxSpeed limits the maximum speed of motor rotation,which is of uint16\_t type,corresponding to the actual speed of 1dps/LSB.

#### 2.21.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data                    |
|------------|----------------------|-------------------------|
| DATA[0]    | 指令字节<br>Command byte | 0xA6                    |
| DATA[1]    | 转动方向字节               | DATA[1] = spinDirection |

|         |                                       |                                      |
|---------|---------------------------------------|--------------------------------------|
|         | Rotation direction byte               |                                      |
| DATA[2] | 速度限制低字节<br>Speed limit low byte       | DATA[2] = (uint8_t)(maxSpeed)        |
| DATA[3] | 速度限制高字节<br>Speed limit high byte      | DATA[3] = (uint8_t)(maxSpeed>>8)     |
| DATA[4] | 位置控制低字节<br>Position Control Low byte  | DATA[4] = (uint8_t)(angleControl)    |
| DATA[5] | 位置控制高字节<br>Position Control High byte | DATA[5] = (uint8_t)(angleControl>>8) |
| DATA[6] | NULL                                  | 0x00                                 |
| DATA[7] | NULL                                  | 0x00                                 |

### 2.21.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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)。

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

1. Motor temperature temperature (int8\_t type, 1°C/LSB);
2. The torque current value iq of the motor (int16\_t type, 0.01A/LSB);
3. Motor output shaft speed (int16\_t type, 1dps/LSB);
4. Encoder position value encoder (uint16\_t type, the value range of the encoder is determined by the number of bits of the encoder).

| Data Field | Description | Data |
|------------|-------------|------|
|------------|-------------|------|

|         |                                     |                                  |
|---------|-------------------------------------|----------------------------------|
| DATA[0] | 指令字节<br>Command byte                | 0xA6                             |
| DATA[1] | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2] | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3] | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4] | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5] | 电机速度高字节<br>Motor speed high byte    | DATA[5] = (uint8_t)(speed>>8)    |
| DATA[6] | 编码器值低字节<br>Encoder value low byte   | DATA[6] = (uint8_t)(encoder)     |
| DATA[7] | 编码器值高字节<br>Encoder value high byte  | DATA[7] = (uint8_t)(encoder>>8)  |

#### 2.21.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID | Length | 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。电机将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。

**Description:**

Data[1] is 0,which means the motor will rotate clockwise.

Data[2] and Data[3] form one (Data[2] is the low bit,Data[3] is the high bit) 16-bit data is 0x01F4,which means the decimal 500dps motor speed. The drive will run the position loop at this speed as the maximum speed.

Data[4] to data[7] form a (Data[4] is the lowest bit,Data[7] is the highest bit) 32-bit data is 0x8CA0,which means that the decimal system is 36000,and the unit is 0.01degree. The motor will move 360° clockwise. The 360-degree and 0-degree positions in the single-lap position coincide,so the position may also be 0 degrees at this time.

回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data 0x0064 of Data[2] and Data[3] is 100 in decimal, which is  $100 \times 0.01 = 1A$  according to the 100-fold reduction, which means that the actual current of the motor is 1A.

Data[4] and Data[5] synthesized data 0x01F4 is 500 in decimal, which means the motor speed is 500dps.

The synthetic data 0x03E8 of Data[6] and Data[7] is 1000 in decimal, which means that the value of the motor encoder relative to the zero position is 1000 pulses.

### 示例 2:

发送指令:

### Example 2:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

### Description:

Data[1] is 1,which means the motor will rotate counterclockwise.

Data[2] and Data[3] form one (Data[2] is the low bit,Data[3] is the high bit) 16-bit data is 0x01F4,which means the decimal 500dps motor speed. The drive will run the position loop at this speed as the maximum speed.

Data[4] to data[7] form a (Data[4] is the lowest bit,Data[7] is the highest bit) 32-bit data is 0x8CA0,which means that the decimal system is 36000,and the unit is 0.01degree. The motor will move 360° in a counterclockwise direction. The 360-degree and 0-degree positions in the single-lap position coincide,so the position may also be 0 degrees at this time.

回复指令:

### Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] = 0x32 is 50 in decimal,which means the motor temperature is 50 degrees at the moment.

The composite data 0x0064 of Data[2] and Data[3] is 100 in decimal,which is  $100 \times 0.01 = 1A$  according to the 100-fold reduction,which means that the actual current of the motor is 1A.

Data[4] and Data[5] synthesized data 0x01F4 is 500 in decimal,which means the motor speed is 500dps.

The synthetic data 0x03E8 of Data[6] and Data[7] is 1000 in decimal,which means that the value of the motor encoder relative to the zero position is 1000 pulses.

## 2.22. 增量位置闭环控制指令 (0xA8)

### Incremental Position Closed-Loop Control Command (0xA8)

#### 2.22.1. 指令说明

##### Instruction Description

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。主机发送该指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。

This command is a control command,which can be run when the motor is not faulty. The host sends this command to control the incremental position (multi-turn angle) of the motor,and run the input position increment with the current position as the starting point. The control value angleControl is of type int32\_t,and the corresponding actual position is



0.01degree/LSB,that is,36000 represents 360°,and the rotation direction of the motor is determined by the incremental position symbol.

The control value maxSpeed limits the maximum speed of the motor output shaft rotation,which is of type uint16\_t,corresponding to the actual speed of 1dps/LSB.

### 2.22.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description                           | Data                                  |
|------------|---------------------------------------|---------------------------------------|
| DATA[0]    | 指令字节<br>Command byte                  | 0xA8                                  |
| DATA[1]    | NULL                                  | 0x00                                  |
| DATA[2]    | 速度限制低字节<br>Speed limit low byte       | DATA[2] = (uint8_t)(maxSpeed)         |
| DATA[3]    | 速度限制高字节<br>Speed limit high byte      | DATA[3] = (uint8_t)(maxSpeed>>8)      |
| DATA[4]    | 位置控制低字节<br>Position control low byte  | DATA[4] = (uint8_t)(angleControl)     |
| DATA[5]    | 位置控制<br>Position control              | DATA[5] = (uint8_t)(angleControl>>8)  |
| DATA[6]    | 位置控制<br>Position control              | DATA[6] = (uint8_t)(angleControl>>16) |
| DATA[7]    | 位置控制高字节<br>Position control high byte | DATA[7] = (uint8_t)(angleControl>>24) |

### 2.22.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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）。

The motor replies to the host after receiving the command,and the frame data contains the following parameters.

1. Motor temperature temperature (int8\_t type,1°C/LSB).
2. The torque current value iq of the motor (int16\_t type,0.01A/LSB).
3. Motor output shaft speed (int16\_t type,1dps/LSB).
4. Motor output shaft angle (int16\_t type,1degree/LSB,maximum range ±32767degree).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0xA8                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5]    | 电机速度高字节<br>Motor speed high byte    | DATA[5] = (uint8_t)(speed>>8)    |
| DATA[6]    | 电机角度低字节<br>Motor angle low byte     | DATA[6] = (uint8_t)(degree)      |
| DATA[7]    | 电机角度高字节<br>Motor angle high byte    | DATA[7] = (uint8_t)(degree>>8)   |

#### 2.22.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

### Example 1:

#### Send command:

##### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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  $360^\circ$ 。

#### Description:

Data[2] and Data[3] form one (Data[2] is the low bit,Data[3] is the high bit) 16-bit data is 0x01F4,which means the decimal 500dps motor output shaft speed. The drive will run the position loop at this speed as the maximum speed.

Data[4] to data[7] form one (Data[4] is the lowest bit,Data[7] is the highest bit) 32-bit data is 0x00008CA0,which means 36000 in decimal. The sending command is reduced by 100 times according to 0.01degree/LSB,that is, $36000 \times 0.01 = 360^\circ$ . The motor will move  $360^\circ$  in the positive direction with the output shaft relative to the current position.

#### 回复指令:

#### Reply command:

##### CAN:

| ID | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[1] = 0x32 is 50 in decimal, which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3] 0x0064 is 100 in decimal, and it is  $100 \times 0.01 = 1A$  when scaled down by 100 times, which means that the actual current of the current motor is 1A.

The composite data 0x01F4 of Data[4] and Data[5] is 500 in decimal, which means the motor output shaft speed is 500dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7] 0x002D is 45 in decimal, which means that the motor output shaft moves 45 degrees in the positive direction relative to the zero position. The position of the motor output shaft is related to the number of lines of the

motor encoder and the reduction ratio. For example, if the number of lines of the motor encoder is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.

**示例 2:**

**发送指令:**

**Example 2:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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°。

**Description:**

Data[2] and Data[3] form one (Data[2] is the low bit, Data[3] is the high bit) 16-bit data is 0x01F4, which means the decimal 500dps motor output shaft speed. The drive will run the position loop at this speed as the maximum speed.

Data[4] to data[7] form one (Data[4] is the lowest bit, Data[7] is the highest bit) 32-bit data is 0xFFFF7360, which means -36000 in decimal. The sending command is reduced by 100 times according to 0.01degree/LSB, ie  $-36000 \times 0.01 = -360^\circ$ . The motor will move  $-360^\circ$  in the opposite direction relative to the current position with the output

shaft.

回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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\*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。

**Description:**

Data[1] = 0x32 is 50 in decimal,which means the motor temperature is 50 degrees at the moment.

The composite data of Data[2] and Data[3] 0xFF9C is -100 in decimal,and it is

$-100 \times 0.01 = -1\text{A}$  when scaled down by 100 times, which means that the actual current of the current motor is  $-1\text{A}$ .

The composite data  $0\text{xFE0C}$  of Data[4] and Data[5] is  $-500$  in decimal, which means the motor output shaft speed is  $-500\text{dps}$ . There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

The composite data of Data[6] and Data[7]  $0\text{xFFD3}$  is  $-45$  in decimal, which means that the motor output shaft moves in the opposite direction by  $-45$  degrees relative to the zero position. The position of the motor output shaft is related to the number of lines of the motor encoder and the reduction ratio. For example, if the number of lines of the motor encoder is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.

## 2.23. 力控位置闭环控制指令 (0xA9)

### Force Control Position Closed-Loop Command (0xA9)

#### 2.23.1. 指令说明

##### Instruction Description

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。主机发送该指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  $0.01\text{degree/LSB}$ ，即 36000 代表  $360^\circ$ ，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  $1\text{dps/LSB}$ 。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  $1\% \times \text{额定电流/LSB}$ 。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。

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。



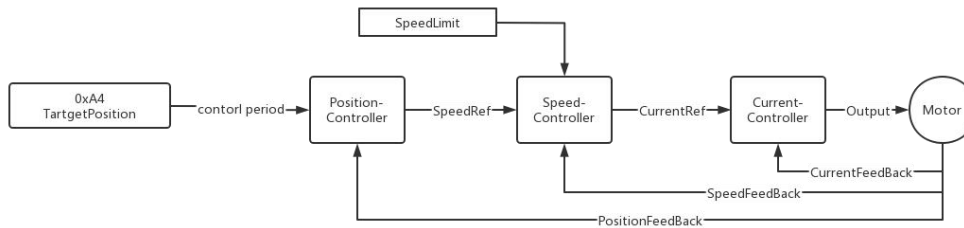


图 2-1 带速度限制的位置跟踪模式框图

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 **maxSpeed** 限制了位置运行过程中的最大速度，加速度由位置环设置的加速度决定。

This command is a control command that can be executed when there are no faults in the motor. The host computer sends this command to control the position (multi-turn angle) of the motor. The control value `angleControl` is of type `int32_t`, corresponding to an actual position of 0.01 degree/LSB. For example, 36000 represents 360°. The direction of motor rotation is determined by the difference between the target position and the current position. The control value `maxSpeed` limits the maximum rotational speed of the motor output shaft. It is of type `uint16_t`, corresponding to an actual speed of 1 dps/LSB (degrees per second). The control value `maxTorque` limits the maximum torque of the motor output shaft. It is of type `uint8_t`, with a value range of 0 to 255, representing the percentage of the rated current, specifically 1% of the rated current per LSB. If the given current exceeds the stall current, the force control mode will not be activated. The maximum torque current of the motor will then be limited by the motor stall current value set in the setup software.

1. If the position loop acceleration is set to 0, the position loop enters Direct Tracking Mode, where the target position is tracked directly via a PI controller. In this mode, `maxSpeed` defines the maximum velocity limit during operation. This is illustrated in the diagram below:

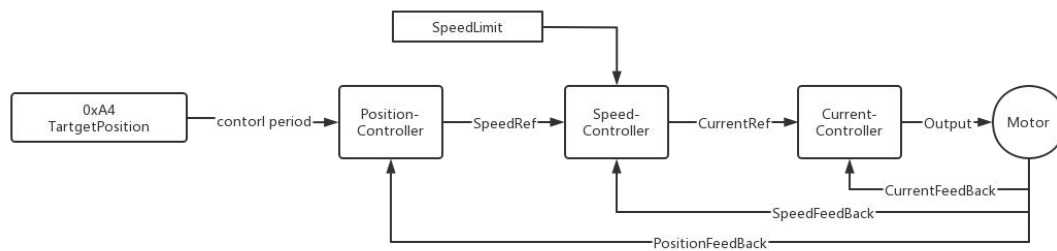


Figure 2-1: Block Diagram of Position Tracking Mode with Velocity Limiting

2. If the position loop acceleration is non-zero, the motor operates in a Profiled Motion Mode. In this mode, the motor autonomously manages the acceleration and deceleration phases. The maxSpeed parameter caps the peak velocity during movement, while the acceleration rate is determined by the position loop's acceleration setting.

### 2.23.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description                          | Data                                  |
|------------|--------------------------------------|---------------------------------------|
| DATA[0]    | 指令字节<br>Command byte                 | 0xA9                                  |
| DATA[1]    | 最大扭矩<br>Max Torque                   | DATA[2] = (uint8_t)(maxTorque)        |
| DATA[2]    | 速度限制低字节<br>Speed Limit Low Byte      | DATA[2] = (uint8_t)(maxSpeed)         |
| DATA[3]    | 速度限制高字节<br>Speed Limit High Byte     | DATA[3] = (uint8_t)(maxSpeed>>8)      |
| DATA[4]    | 位置控制低字节<br>Position Control Low Byte | DATA[4] = (uint8_t)(angleControl)     |
| DATA[5]    | 位置控制<br>Position Control             | DATA[5] = (uint8_t)(angleControl>>8)  |
| DATA[6]    | 位置控制<br>Position Control             | DATA[6] = (uint8_t)(angleControl>>16) |



|         |  |                                       |
|---------|--|---------------------------------------|
| DATA[7] | 位置控制高字节<br>Position Control High<br>Byte | DATA[7] = (uint8_t)(angleControl>>24) |
|---------|--|---------------------------------------|

### 2.23.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

After receiving the command, the motor replies to the host. The data frame includes the following parameters:

1. Motor temperature “temperature” (type int8\_t, 1°C/LSB);
2. Motor torque current value “iq” (type int16\_t, 0.01A/LSB);
3. Motor output shaft speed “speed” (type int16\_t, 1dps/LSB);
4. Motor output shaft angle (type int16\_t, 1degree/LSB, maximum range ±32767 degrees).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0xA9                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |

|         |                                  |                                |
|---------|----------------------------------|--------------------------------|
| DATA[5] | 电机速度高字节<br>Motor speed high byte | DATA[5] = (uint8_t)(speed>>8)  |
| DATA[6] | 电机角度低字节<br>Motor angle low byte  | DATA[6] = (uint8_t)(degree)    |
| DATA[7] | 电机角度高字节<br>Motor angle high byte | DATA[7] = (uint8_t)(degree>>8) |

#### 2.23.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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\*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\*0.01=360°。电机以输出轴相对零点位置正向移动 360°。

**Description:**

Data[1] is an 8-bit data value of 0x3C, which represents  $60 \times 1\% \times \text{rated current}$  in decimal. Data[2] and Data[3] form a 16-bit data value (Data[2] as the low byte and Data[3] as the high byte) of 0x01F4, which represents 500 dps (degrees per second) in decimal for the motor output shaft speed. The drive will operate with a maximum torque of 60% rated torque and a maximum speed of 500 dps in the position loop.

Data[4] to Data[7] form a 32-bit data value (Data[4] as the lowest byte and Data[7] as the highest byte) of 0x00008CA0, which represents 36000 in decimal. The command is scaled down by a factor of 100 according to 0.01 degree/LSB, i.e.,  $36000 \times 0.01 = 360^\circ$ . The motor will move the output shaft positively by  $360^\circ$  relative to the zero position.

回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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  $100 \times 0.01 = 1\text{A}$ , 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。

### Description:

Data[1] = 0x32 is 50 in decimal, indicating that the current motor temperature is 50°C.

Data[2] and Data[3] form the data 0x0064, which is 100 in decimal. According to the scaling factor of 100 times, this translates to  $100 \times 0.01 = 1\text{A}$ . Therefore, it represents that the actual current of the motor at this moment is 1A.

Data[4] and Data[5] form the data 0x01F4, which is 500 in decimal, representing the motor output shaft speed as 500 dp (degrees per second). There is a gear ratio relationship between the motor output shaft speed and the motor speed. If the gear ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

Data[6] and Data[7] form the data 0x002D, which is 45 in decimal, indicating that the motor output shaft has moved positively by 45 degrees relative to the zero position. The position of the motor output shaft is related to the motor encoder lines and the gear ratio. For example, if the motor encoder has 16384 lines and the gear ratio is 6, then the 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98304$  pulses.

### 示例 2:

#### 发送指令:

#### Example 2:

#### Send command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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  $60 \times 1\% \times$  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  $60\% \times$  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  $-36000 \times 0.01 = -360^\circ$ 。电机将以输出轴相对零点位置反向移动-360°。

#### Description:

Data[1] is an 8-bit data value of 0x3C, which represents  $60 \times 1\% \times$  rated current in decimal. Data[2] and Data[3] form a 16-bit data value (Data[2] as the low byte and Data[3] as the high byte) of 0x01F4, which represents 500 dps (degrees per second) in decimal for the motor output shaft speed. The drive will operate with a maximum torque of  $60\% \times$  rated torque and a maximum speed of 500 dps in the position loop.

Data[4] to Data[7] form a 32-bit data value (Data[4] as the lowest byte and Data[7] as the highest byte) of 0xFFFF7360, which represents -36000 in decimal. The command is scaled down by a factor of 100 according to 0.01 degree/LSB, i.e.,  $-36000 \times 0.01 = -360^\circ$ . The motor will move the output shaft negatively by  $-360^\circ$  relative to the zero position.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

### Description:

Data[1] = 0x32 is 50 in decimal, indicating that the motor temperature at this moment is 50 degrees.

Data[2] and Data[3] form the data 0xFF9C, which is -100 in decimal. After scaling down by a factor of 100, it becomes  $-100 \times 0.01 = -1A$ , representing that the actual current of the motor at present is -1A.

Data[4] and Data[5] form the data 0xFE0C, which is -500 in decimal, indicating that the motor output shaft speed is -500 dps (degrees per second). There is a gear ratio relationship between the motor output shaft speed and the motor speed. If the gear ratio is 6, then the motor speed is 6 times higher than the output shaft speed.

Data[6] and Data[7] form the data 0xFFD3, which is -45 in decimal, indicating that the motor output shaft has moved -45 degrees in the reverse direction relative to the zero position. The position of the motor output shaft is related to the motor encoder lines and the gear ratio. For example, if the motor encoder lines are 16384 and the gear ratio is 6, then 360 degrees of the motor output shaft correspond to  $16384 \times 6 = 98304$  pulses.



## 2.24. SF 指令（0x72 带前馈速度的位置控制指令）

### SF Command (0x72 Position Control with Speed Feedforward)

#### 2.24.1. 指令说明

##### Instruction Description

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。主机发送该指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  $0.01^{\circ}/\text{LSB}$ ，即 36000 代表  $360^{\circ}$ ，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  $1\text{dps}/\text{LSB}$ 。控制值 `Feedforward Speed` 设置了电机输出轴的前馈速度，为 `int8_t` 类型，取值范围为  $-128 \sim 127$ ，以额定速度的百分比为单位，即  $1\% \times \text{额定速度}/\text{LSB}$ 。

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。如下图所示。只有在跟踪模式，控制值 `Feedforward Speed` 才会发挥作用。

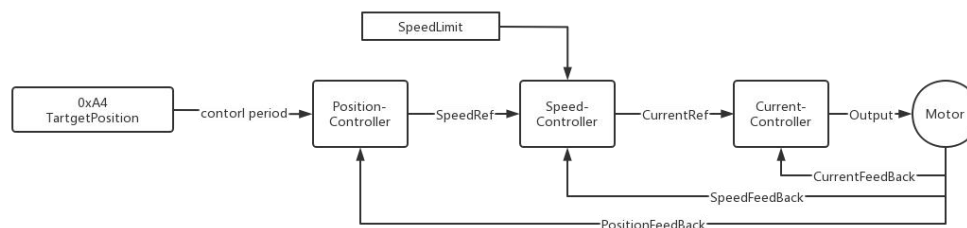


图 2-1 带速度限制的位置跟踪模式框图

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 `maxSpeed` 限制了位置运行过程中的最大速度，加速度由位置环设置的加速度决定。在“带速度规划的运动模式”下，控制值 `Feedforward Speed` 不发挥作用，指令的效果与“绝对位置闭环控制指令（0xA4）”完全一致。

This is a motion control command that can be executed as long as the motor is in a fault-free state. The host controller issues this command to control the motor's position (multi-turn angle). The control value `angleControl` (`int32_t`) specifies the target position with a resolution of  $0.01^{\circ}/\text{LSB}$  (e.g., a value of 36,000 represents  $360^{\circ}$ ), where the rotation direction is determined by the difference between the target and current positions.



The maxSpeed parameter (uint16\_t) limits the peak velocity of the motor output shaft with a resolution of 1 dps/LSB. Additionally, the Feedforward Speed value (int8\_t) sets the feedforward speed for the output shaft, ranging from -128 to 127, where each unit represents 1% of the rated speed per LSB.

1. When the position loop acceleration is set to 0, the system enters Direct Tracking Mode, where the target position is tracked directly via a PI controller. In this mode, the maxSpeed parameter defines the maximum velocity limit during operation, as illustrated in the block diagram below. The Feedforward Speed control value only takes effect in this tracking mode.

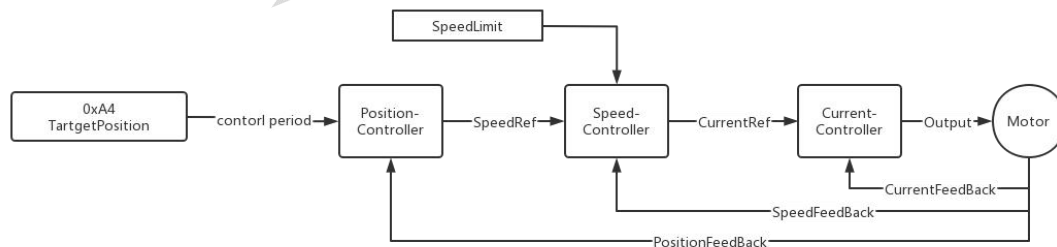


Figure 2-1: Block Diagram of Position Tracking Mode with Velocity Limiting

2. When the position loop acceleration is non-zero, the system operates in Profiled Motion Mode, where the motor autonomously manages the acceleration and deceleration ramps. In this mode, maxSpeed defines the peak velocity limit for the motion, while the acceleration rate is determined by the value configured in the position loop. In this Profiled Motion Mode, the Feedforward Speed control value does not take effect, and the command behaves exactly the same as the "Absolute Position Closed-Loop Control Command (0xA4)".

### 2.24.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description          | Data                                   |
|------------|----------------------|--|
| DATA[0]    | 指令字节<br>Command Byte | 0x72                                   |
| DATA[1]    | 前馈速度                 | DATA[2] = (uint8_t)(Feedforward Speed) |

|         |  |                                       |
|---------|--|---------------------------------------|
|         | Feedforward Speed                          |                                       |
| DATA[2] | 速度限制低字节<br>Max Speed (Low Byte)            | DATA[2] = (uint8_t)(maxSpeed)         |
| DATA[3] | 速度限制高字节<br>Max Speed (High Byte)           | DATA[3] = (uint8_t)(maxSpeed>>8)      |
| DATA[4] | 位置控制低字节<br>Position Control (Byte 1 / LSB) | DATA[4] = (uint8_t)(angleControl)     |
| DATA[5] | 位置控制<br>Position Control (Byte 2)          | DATA[5] = (uint8_t)(angleControl>>8)  |
| DATA[6] | 位置控制<br>Position Control (Byte 3)          | DATA[6] = (uint8_t)(angleControl>>16) |
| DATA[7] | 位置控制高字节<br>Position Control (Byte 4 / MSB) | DATA[7] = (uint8_t)(angleControl>>24) |

### 2.24.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

After receiving the command, the motor replies to the host. The data frame contains the following parameters:

1. Motor temperature (int8\_t, 1° C/LSB);
2. Motor torque current value iq (int16\_t, 0.01 A/LSB);
3. Motor output shaft speed (int16\_t, 1 dps/LSB);
4. Motor output shaft angle (int16\_t, 1 degree/LSB, maximum range ± 32767 degrees).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0x72                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5]    | 电机速度高字节<br>Motor speed high byte    | DATA[5] = (uint8_t)(speed>>8)    |
| DATA[6]    | 电机角度低字节<br>Motor angle low byte     | DATA[6] = (uint8_t)(degree)      |
| DATA[7]    | 电机角度高字节<br>Motor angle high byte    | DATA[7] = (uint8_t)(degree>>8)   |

#### 2.24.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | Data[0] | Data[1] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x141 | 0x73    | 0x3C    | 0xF4    | 0x01    | 0xA0    | 0x8C    | 0x00    | 0x00    |

RS485:

| Frame | ID | Length | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | CRC16L | CRC16H |
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|

|        |      |      |      |      |      |      |      |      |      |      |        |        |
|--------|------|------|------|------|------|------|------|------|------|------|--------|--------|
| header |      |      |      |      |      |      |      |      |      |      |        |        |
| 0x3E   | 0x01 | 0x08 | 0x73 | 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  $60 \times 1\%$  额定速度。

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  $60\%$  额定速度为前馈速度、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  $36000 \times 0.01 = 360^\circ$ 。电机会以输出轴相对零点位置正向移动  $360^\circ$ 。

#### Description:

Data[1] is an 8-bit data value 0x3C, representing decimal  $60 \times 1\%$  rated speed.

Data[2] and Data[3] form a 16-bit data value (Data[2] is low byte, Data[3] is high byte) of 0x01F4, representing decimal 500 dps motor output shaft speed. The drive will run the position loop with  $60\%$  rated speed as the feedforward speed and 500 dps as the maximum speed.

Data[4] through Data[7] form a 32-bit data value (Data[4] is the lowest byte, Data[7] is the highest byte) of 0x00008CA0, representing decimal 36000. The transmitted command is scaled down by a factor of 100 according to 0.01 degree/LSB, i.e.,  $36000 \times 0.01 = 360^\circ$ . The motor will move forward by  $360^\circ$  relative to the output shaft's zero position.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | Data[0] | Data[1] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x241 | 0x73    | 0x32    | 0x64    | 0x00    | 0xF4    | 0x01    | 0x2D    | 0x00    |

#### RS485:

| Frame | ID | Length | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | CRC16L | CRC16H |
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|

|        |      |      |      |      |      |      |      |      |      |      |        |        |
|--------|------|------|------|------|------|------|------|------|------|------|--------|--------|
| header |      |      |      |      |      |      |      |      |      |      |        |        |
| 0x3E   | 0x01 | 0x08 | 0x73 | 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。

#### Description:

Data[1] = 0x32, decimal 50, represents that the motor temperature is  $50^{\circ}\text{C}$  at this moment.

Data[2] and Data[3] synthesize the data 0x0064, decimal 100. Scaled down by a factor of 100, i.e.,  $100 \times 0.01 = 1\text{A}$ , this means that the current actual motor current is 1 A.

Data[4] and Data[5] synthesize the data 0x01F4, decimal 500, representing a motor output shaft speed of 500 dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, the motor speed is 6 times higher than the output shaft speed.

Data[6] and Data[7] synthesize the data 0x002D, decimal 45, indicating that the motor output shaft has moved forward by 45 degrees relative to the zero position. The motor output shaft position is related to the motor encoder line count and the reduction ratio. For example, if the motor encoder line count is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98,304$  pulses.

#### 示例 2:

发送指令:

**Example 2:**

**Send command:**

**CAN:**

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

**RS485:**

| Frame header | ID   | Length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x73 | 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  $60 \times 1\% \times$  额定速度。

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  $60\% \times$  额定扭矩为前馈扭矩、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  $-36000 \times 0.01 = -360^\circ$ 。电机将以输出轴相对零点位置反向移动  $-360^\circ$ 。

**Description:**

Data[1] is an 8-bit data value 0x3C, representing decimal  $60 \times 1\% \times$  rated speed.

Data[2] and Data[3] form a 16-bit data value (Data[2] is low byte, Data[3] is high byte) of 0x01F4, representing decimal 500 dps motor output shaft speed. The drive will run the position loop with  $60\% \times$  rated torque as the feedforward torque and 500 dps as the maximum speed.

Data[4] through Data[7] form a 32-bit data value (Data[4] is the lowest byte, Data[7] is the highest byte) of 0xFFFF7360, representing decimal -36000. The transmitted command is scaled down by a factor of 100 according to 0.01 degree/LSB, i.e.,  $-36000 \times 0.01 = -360^\circ$ . The motor will move backward by  $-360^\circ$  relative to the output



shaft's zero position.

回复指令:

**Reply command:**

**CAN:**

| ID    | Data[0] | Data[1] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x241 | 0x73    | 0x32    | 0x9C    | 0xFF    | 0x0C    | 0xFE    | 0xD3    | 0xFF    |

**RS485:**

| Frame header | ID   | Length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7 | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|----|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x73 | 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。

**Description:**

Data[1] = 0x32, decimal 50, represents that the motor temperature is 50° C at this moment.

Data[2] and Data[3] synthesize the data 0xFF9C, decimal - 100. Scaled down by a factor of 100, i.e., - 100\*0.01= - 1 A, this means that the current actual motor current is - 1 A.

Data[4] and Data[5] synthesize the data 0xFE0C, decimal - 500, representing a



motor output shaft speed of - 500 dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, the motor speed is 6 times higher than the output shaft speed.

Data[6] and Data[7] synthesize the data 0xFFD3, decimal - 45, indicating that the motor output shaft has moved backward by - 45 degrees relative to the zero position. The motor output shaft position is related to the motor encoder line count and the reduction ratio. For example, if the motor encoder line count is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98,304$  pulses.

## 2.25. TF 指令（0x73 带前馈扭矩的位置控制指令）

### TF Command (0x73 Position Control with Torque Feedforward)

#### 2.25.1. 指令说明

##### Instruction Description

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。主机发送该指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 `Feedforward Torque` 设置了电机输出轴的前馈扭矩，为 `int8_t` 类型，取值范围为 -128 ~ 127，以额定电流的百分比为单位，即 1%\*额定电流/LSB。

3. 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。如下图所示。

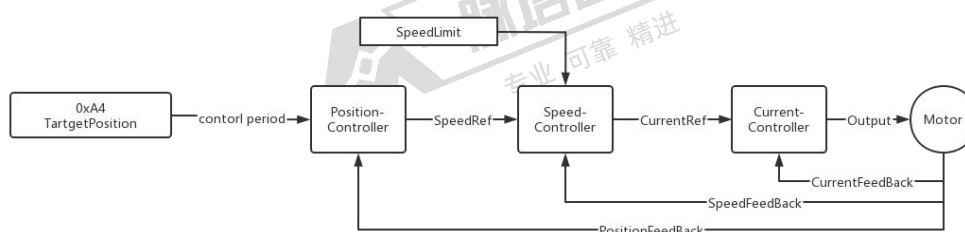


图 2-1 带速度限制的位置跟踪模式框图

4. 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 `maxSpeed` 限制了位置运行过程中的最大速度，加速度由位置环设置的加速度决定。

This is a motion control command that can be executed as long as the motor is in a fault-free state. The host controller issues this command to control the motor's position (multi-turn angle). The control value `angleControl` (`int32_t`) specifies the target position with a resolution of  $0.01^\circ / \text{LSB}$  (e.g., a value of 36,000 represents  $360^\circ$ ), where the rotation direction is determined by the difference between the target and current positions. The `maxSpeed` parameter (`uint16_t`) limits the peak velocity of the motor output shaft with a resolution of  $1 \text{ dps} / \text{LSB}$ . Additionally, the Feedforward Torque value (`int8_t`) sets the feedforward torque for the output shaft, ranging from -128 to 127, where each unit represents 1% of the rated current per LSB.

3. When the position loop acceleration is set to 0, the system enters Direct Tracking Mode, where the target position is tracked directly via a PI controller. In this mode, the `maxSpeed` parameter defines the maximum velocity limit during operation, as illustrated in the block diagram below.

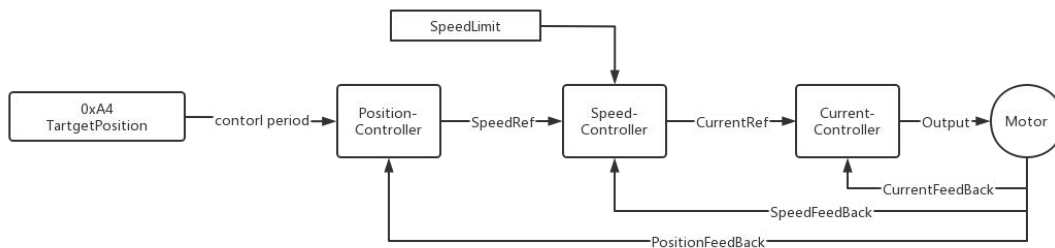


Figure 2-1: Block Diagram of Position Tracking Mode with Velocity Limiting

4. When the position loop acceleration is non-zero, the system operates in Profiled Motion Mode, where the motor autonomously manages the acceleration and deceleration ramps. In this mode, `maxSpeed` defines the peak velocity limit for the motion, while the acceleration rate is determined by the value configured in the position loop.

### 2.25.2. 发送数据域定义

#### Send Data Field Definition

| Data field | Description | Data |
|------------|-------------|------|
|------------|-------------|------|

|         |  |   |
|---------|--|---|
| DATA[0] | 指令字节<br>Command Byte                       | 0x73                                    |
| DATA[1] | 前馈扭矩<br>Feedforward Torque                 | DATA[2] = (uint8_t)(Feedforward Torque) |
| DATA[2] | 速度限制低字节<br>Max Speed (Low Byte)            | DATA[2] = (uint8_t)(maxSpeed)           |
| DATA[3] | 速度限制高字节<br>Max Speed (High Byte)           | DATA[3] = (uint8_t)(maxSpeed>>8)        |
| DATA[4] | 位置控制低字节<br>Position Control (Byte 1 / LSB) | DATA[4] = (uint8_t)(angleControl)       |
| DATA[5] | 位置控制<br>Position Control (Byte 2)          | DATA[5] = (uint8_t)(angleControl>>8)    |
| DATA[6] | 位置控制<br>Position Control (Byte 3)          | DATA[6] = (uint8_t)(angleControl>>16)   |
| DATA[7] | 位置控制高字节<br>Position Control (Byte 4 / MSB) | DATA[7] = (uint8_t)(angleControl>>24)   |

### 2.25.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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) 。

After receiving the command, the motor replies to the host. The data frame contains the following parameters:

1. Motor temperature (int8\_t, 1° C/LSB);
2. Motor torque current value iq (int16\_t, 0.01 A/LSB);

3. Motor output shaft speed (int16\_t, 1 dps/LSB);

4. Motor output shaft angle (int16\_t, 1 degree/LSB, maximum range  $\pm 32767$  degrees).

| Data field | Description                         | Data                             |
|------------|-------------------------------------|----------------------------------|
| DATA[0]    | 指令字节<br>Command byte                | 0x73                             |
| DATA[1]    | 电机温度<br>Motor temperature           | DATA[1] = (uint8_t)(temperature) |
| DATA[2]    | 转矩电流低字节<br>Torque current low byte  | DATA[2] = (uint8_t)(iq)          |
| DATA[3]    | 转矩电流高字节<br>Torque current high byte | DATA[3] = (uint8_t)(iq>>8)       |
| DATA[4]    | 电机速度低字节<br>Motor speed low byte     | DATA[4] = (uint8_t)(speed)       |
| DATA[5]    | 电机速度高字节<br>Motor speed high byte    | DATA[5] = (uint8_t)(speed>>8)    |
| DATA[6]    | 电机角度低字节<br>Motor angle low byte     | DATA[6] = (uint8_t)(degree)      |
| DATA[7]    | 电机角度高字节<br>Motor angle high byte    | DATA[7] = (uint8_t)(degree>>8)   |

#### 2.25.4. 通讯示例

Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

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

|       |      |      |      |      |      |      |      |      |
|-------|------|------|------|------|------|------|------|------|
| 0x141 | 0x73 | 0x3C | 0xF4 | 0x01 | 0xA0 | 0x8C | 0x00 | 0x00 |
|-------|------|------|------|------|------|------|------|------|

#### RS485:

| Frame header | ID   | Length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x73 | 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  $60\%$  额定扭矩为前馈扭矩、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  $36000 \times 0.01 = 360^\circ$ 。电机将以输出轴相对零点位置正向移动  $360^\circ$ 。

#### Description:

Data[1] is an 8-bit data value 0x3C, representing decimal  $60 \times 1\%$  rated current.

Data[2] and Data[3] form a 16-bit data value (Data[2] is low byte, Data[3] is high byte) of 0x01F4, representing decimal 500 dps motor output shaft speed. The drive will run the position loop with  $60\%$  rated torque as the feedforward torque and 500 dps as the maximum speed.

Data[4] through Data[7] form a 32-bit data value (Data[4] is the lowest byte, Data[7] is the highest byte) of 0x00008CA0, representing decimal 36000. The transmitted command is scaled down by a factor of 100 according to 0.01 degree/LSB, i.e.,  $36000 \times 0.01 = 360^\circ$ . The motor will move forward by  $360^\circ$  relative to the output shaft's zero position.

#### 回复指令:

#### Reply command:

#### CAN:

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

|       |      |      |      |      |      |      |      |      |
|-------|------|------|------|------|------|------|------|------|
| 0x241 | 0x73 | 0x32 | 0x64 | 0x00 | 0xF4 | 0x01 | 0x2D | 0x00 |
|-------|------|------|------|------|------|------|------|------|

#### RS485:

| Frame header | ID   | Length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x73 | 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。

#### Description:

Data[1] = 0x32, decimal 50, represents that the motor temperature is  $50^{\circ}\text{C}$  at this moment.

Data[2] and Data[3] synthesize the data 0x0064, decimal 100. Scaled down by a factor of 100, i.e.,  $100 \times 0.01 = 1\text{A}$ , this means that the current actual motor current is 1 A.

Data[4] and Data[5] synthesize the data 0x01F4, decimal 500, representing a motor output shaft speed of 500 dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, the motor speed is 6 times higher than the output shaft speed.

Data[6] and Data[7] synthesize the data 0x002D, decimal 45, indicating that the motor output shaft has moved forward by 45 degrees relative to the zero position. The motor output shaft position is related to the motor encoder line count and the reduction



ratio. For example, if the motor encoder line count is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98,304$  pulses.

## 示例 2:

发送指令:

### Example 2:

Send command:

CAN:

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

RS485:

| Frame header | ID   | Length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x73 | 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  $60 \times 1\% \times$  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  $60\% \times$  额定扭矩为前馈扭矩、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  $-36000 \times 0.01 = -360^\circ$ 。电机将以输出轴相对零点位置反向移动  $-360^\circ$ 。

### Description:

Data[1] is an 8-bit data value 0x3C, representing decimal  $60 \times 1\% \times$  rated current.

Data[2] and Data[3] form a 16-bit data value (Data[2] is low byte, Data[3] is high byte) of 0x01F4, representing decimal 500 dps motor output shaft speed. The drive will run the position loop with  $60\% \times$  rated torque as the feedforward torque and 500 dps as the maximum speed.

Data[4] through Data[7] form a 32-bit data value (Data[4] is the lowest byte, Data[7]



is the highest byte) of 0xFFFF7360, representing decimal -36000. The transmitted command is scaled down by a factor of 100 according to 0.01 degree/LSB, i.e.,  $-36000 \times 0.01 = -360^\circ$ . The motor will move backward by  $-360^\circ$  relative to the output shaft's zero position.

回复指令:

**Reply command:**

**CAN:**

| ID    | Data[0] | Data[1] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x241 | 0x73    | 0x32    | 0x9C    | 0xFF    | 0x0C    | 0xFE    | 0xD3    | 0xFF    |

**RS485:**

| Frame header | ID   | Length | D0   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0x73 | 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。

**Description:**

Data[1] = 0x32, decimal 50, represents that the motor temperature is  $50^\circ C$  at this moment.

Data[2] and Data[3] synthesize the data 0xFF9C, decimal - 100. Scaled down by a

factor of 100, i.e.,  $-100 \times 0.01 = -1$  A, this means that the current actual motor current is  $-1$  A.

Data[4] and Data[5] synthesize the data 0xFE0C, decimal  $-500$ , representing a motor output shaft speed of  $-500$  dps. There is a reduction ratio relationship between the motor output shaft speed and the motor speed. If the reduction ratio is 6, the motor speed is 6 times higher than the output shaft speed.

Data[6] and Data[7] synthesize the data 0xFFD3, decimal  $-45$ , indicating that the motor output shaft has moved backward by  $-45$  degrees relative to the zero position. The motor output shaft position is related to the motor encoder line count and the reduction ratio. For example, if the motor encoder line count is 16384 and the reduction ratio is 6, then 360 degrees of the motor output shaft corresponds to  $16384 \times 6 = 98,304$  pulses.

## 2.26. 系统运行模式获取 (0x70)

### System Operating Mode Acquisition (0x70)

#### 2.26.1. 指令说明

##### Instruction Description

该指令读取当前电机运行模式。

This command reads the current motor running mode.

#### 2.26.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |
|---------|------|------|

### 2.26.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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)。

The motor replies to the host after receiving the command, and the drive reply data contains the running state of the parameter `runmode`, which is of type `uint8_t`.

The motor operation mode has the following 3 states:

1. Current loop mode (0x01);
2. Speed loop mode (0x02);
3. Position loop mode (0x03).

| Data field | Description                    | Data                         |
|------------|--------------------------------|------------------------------|
| DATA[0]    | 指令字节<br>Command byte           | 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<br>Motor operating mode | DATA[7] = (uint8_t)(runmode) |

### 2.26.4. 通讯示例

#### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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:**

该指令读取当前电机运行模式。

**Description:**

This command reads the current motor running mode.

回复指令:

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

Data[7] = 0x03, according to the definition of the reply frame, it means that the current system is in the position loop mode.

## 2.27. 系统复位指令 (0x76)

### System Reset Command (0x76)

#### 2.27.1. 指令说明

##### Instruction Description

该指令用于复位系统程序。

This command is used to reset the system program.

#### 2.27.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

#### 2.27.3. 回复数据域定义

##### Reply Data Field Definition

电机收到指令后会复位，不再返回指令。

The motor will reset after receiving the command and will not return to the command.

#### 2.27.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

### Send command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

After sending the command,the system is reset and the program runs again.

## 2.28. 系统抱闸释放指令 (0x77)

### System Brake Release Command (0x77)

#### 2.28.1. 指令说明

##### Instruction Description

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

This command is used to open the system brake. The system will release the holding brake,and the motor will be in a movable state without being restricted by the holding brake.

#### 2.28.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.28.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据和主机发送的指令相同。

The motor replies to the host after receiving the command, and the frame data is the same as the command sent by the host.

## 2.29. 系统抱闸锁死指令 (0x78)

### System Brake Lock Command (0x78)

#### 2.29.1. 指令说明

##### Instruction Description

该指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。

This command is used to close the system holding brake. The holding brake locks the motor and the motor can no longer run. The holding brake is also in this state after the system is powered off.

#### 2.29.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.29.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据和主机发送的指令相同。

The motor replies to the host after receiving the command, and the frame data is the same as the command sent by the host.

## 2.30. 系统运行时间读取指令 (0xB1)

### System Runtime Read Command (0xB1)

#### 2.30.1. 指令说明

##### Instruction Description

该指令用于获取系统运行时间，单位 ms。

This command is used to obtain the system running time in ms.

#### 2.30.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.30.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，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。

The motor replies to the host after receiving the command,and the drive reply data contains the system running time SysRunTime,which is uint32\_t type,and the unit is ms.

| Data field | Description                               | Data                                |
|------------|---|-------------------------------------|
| DATA[0]    | 指令字节<br>Command byte                      | 0xB1                                |
| DATA[0]    | NULL                                      | 0x00                                |
| DATA[0]    | NULL                                      | 0x00                                |
| DATA[0]    | NULL                                      | 0x00                                |
| DATA[4]    | SysRunTime 低字节 1<br>SysRunTime low byte 1 | DATA[4] = (uint8_t)(SysRunTime)     |
| DATA[5]    | SysRunTime 字节 2<br>SysRunTime byte 2      | DATA[5] = (uint8_t)(SysRunTime>>8)  |
| DATA[6]    | SysRunTime 字节 3<br>SysRunTime byte 3      | DATA[6] = (uint8_t)(SysRunTime>>16) |
| DATA[7]    | SysRunTime 字节 4<br>SysRunTime byte 4      | DATA[7] = (uint8_t)(SysRunTime>>24) |

#### 2.30.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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:**

该指令读取当前系统已运行的时间。

**Description:**

This command reads the running time of the current system.

**回复指令:**

**Reply command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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 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。

**Description:**

Data[4] to Data[7] (Data[4] is low and Data[7] is high) = 0x10000000, decimal 268435456, indicating that the system has run for 268435456ms after restarting or resetting, about 74 hour.

## 2.31. 系统软件版本日期读取指令（0xB2）

### System Software Version Date Read Command (0xB2)

#### 2.31.1. 指令说明

##### Instruction Description

该指令用于获取系统软件版本更新日期。

This command is used to get the update date of the system software version.

#### 2.31.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

#### 2.31.3. 回复数据域定义

##### Reply Data Field Definition

电机在收到指令后回复主机，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。

The motor will reply to the host after receiving the command. The driver reply data contains the latest version date of the system software, VersionDate, which is of type uint32\_t. The date format is in the format of year, month, and day, such as 20211126.

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 0xB2 |

|         |   |                                      |
|---------|---|--------------------------------------|
| DATA[0] | NULL  | 0x00                                 |
| DATA[0] | NULL  | 0x00                                 |
| DATA[0] | NULL  | 0x00                                 |
| DATA[4] | VersionDate 低字节 1<br>VersionDate low byte 1 | DATA[4] = (uint8_t)(&VersionDate)    |
| DATA[5] | VersionDate 字节 2<br>VersionDate byte 2      | DATA[5] = (uint8_t)(VersionDate>>8)  |
| DATA[6] | VersionDate 字节 3<br>VersionDate byte 3      | DATA[6] = (uint8_t)(VersionDate>>16) |
| DATA[7] | VersionDate 字节 4<br>VersionDate byte 4      | DATA[7] = (uint8_t)(VersionDate>>24) |

#### 2.31.4. 通讯示例

##### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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:**

该指令读取当前软件版本日期。

**Description:**

This command reads the current software version date.

回复指令:

Reply command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

Description:

Data[4] to Data[7] (Data[4] is low and Data[7] is high) = 0x0134892E, decimal 20220206, indicating that the software version date is February 6, 2022.

## 2.32. 通讯中断保护时间设置指令 (0xB3)

### Communication Interruption Protection Time Setting Command (0xB3)

#### 2.32.1. 指令说明

##### Instruction Description

该指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。

This command is used to set the communication interruption protection time in ms. If the communication is interrupted for more than the set time, it will cut off the output brake lock. To run again, you need to establish stable and continuous communication first. Writing 0 means that the communication interruption protection function is not enabled.

#### 2.32.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description                                      | Data                                  |
|------------|--|---------------------------------------|
| DATA[0]    | 指令字节<br>Command byte                             | 0xB3                                  |
| DATA[1]    | NULL   | 0x00                                  |
| DATA[2]    | NULL   | 0x00                                  |
| DATA[3]    | NULL   | 0x00                                  |
| DATA[4]    | CanRecvTime_MS 低字节 1<br>CanRecvTime_MS low byte1 | DATA[4] = (uint8_t)(CanRecvTime_MS)   |
| DATA[5]    | CanRecvTime_MS 字节 2<br>CanRecvTime_MS byte2      | DATA[5]=(uint8_t)(CanRecvTime_MS>>8)  |
| DATA[6]    | CanRecvTime_MS 字节 3<br>CanRecvTime_MS byte3      | DATA[6]=(uint8_t)(CanRecvTime_MS>>16) |
| DATA[7]    | CanRecvTime_MS 字节 4<br>4CanRecvTime_MS byte4     | DATA[7]=(uint8_t)(CanRecvTime_MS>>24) |

### 2.32.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据和主机发送的指令相同。

The motor replies to the host after receiving the command, and the frame data is the same as the command sent by the host.

### 2.32.4. 通讯示例

#### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:



| ID    | 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:

| Frame header | ID   | Length | 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，电机将继续执行当前指令。

#### Description:

The data values are all 0, which means that the communication interruption protection function is not enabled. If the communication is interrupted, the motor will continue to execute the current command.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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指令相同。

#### Description:

The frame data is the same as the command sent by the host.

#### 示例 2:

#### 发送指令:

## Example 2:

### Send command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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。

#### Description:

Data[4] to Data[7] (Data[4] is low and Data[7] is high) constitute data 0x000003E8, decimal is 1000ms. It indicates that the communication interruption protection time is set to 1000ms, which is stored in the ROM and saved after power failure. Then, if the communication interval exceeds 1000ms, the communication interruption protection will be triggered, and the output lock brake will be cut off. When the communication interval is restored to within 1000ms, normal operation can be resumed.

#### 回复指令:

#### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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指令相同。

#### Description:

The frame data is the same as the command sent by the host.

### 2.33. 通讯波特率设置指令 (0xB4)

#### Communication Baud Rate Setting Command (0xB4)

##### 2.33.1. 指令说明

##### Instruction Description

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。只有在电机失能时，才能将参数成功保存到 ROM。

This instruction can set the communication baud rate of CAN and RS485 bus. The parameters will be saved in ROM after setting, and will be saved after power off, and will run at the modified baud rate when powered on again. The parameters can be successfully saved to ROM only when the motor is disabled.

##### 2.33.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data |
|------------|----------------------|------|
| DATA[0]    | 指令字节<br>Command byte | 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 |

### 2.33.3. 功能索引说明

#### Function Index Description

| Index | baudrate                         |
|-------|----------------------------------|
| 0x00  | CAN 波特率: 1M<br>RS485 波特率: 115200 |
| 0x01  | CAN 波特率: 500K<br>RS485 波特率: 5M   |
| 0x02  | CAN 波特率: 250K                    |

### 2.33.4. 回复数据域定义

#### Reply Data Field Definition

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。

Since the communication baud rate is modified, the reply command is random and need not be processed.

### 2.33.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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 1Mbps。

**Description:**

Data[7] = 0, which means the baud rate of RS485 is changed to 115200bps, and the baud rate of CAN is changed to 1Mbps.

示例 2:

发送指令:

**Example 2:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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 5Mbps, CAN 波特率改为 500Kbps。

**Description:**

Data[7] = 1, which means the RS485 baud rate is changed to 5Mbps, and the CAN baud rate is changed to 500Kbps.

示例 3:

发送指令:

**Example 3:**

**Send command:**

**CAN:**

| ID    | 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:

| Frame header | ID   | Length | 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。

## Description:

Data[7] = 2, which means the RS485 baud rate is changed to 1Mbps, and CAN is invalid.

## 2.34. 电机型号读取指令 (0xB5)

### Motor Model Reading Command (0xB5)

#### 2.34.1. 指令说明

##### Instruction Description

该指令用于读取电机型号, 电机型号最多包含 15 个字符, 通过索引, 每次可以读取其中的 5 个字符, 发送三条报文获取全部字符。

This command is used to read the motor model. The motor model contains up to 15 characters. By using an index, 5 characters can be read at a time, and three messages are sent to retrieve all characters.

#### 2.34.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description          | Data                      |
|------------|----------------------|---------------------------|
| DATA[0]    | 指令字节<br>Command byte | 0xB5                      |
| DATA[1]    | 标记位<br>Flag bit      | 固定为 0x01<br>Fixed to 0x01 |

|         |             |                          |
|---------|-------------|--------------------------|
| DATA[2] | 索引<br>Index | DATA[1] = (uint8_t)index |
| DATA[3] | NULL        | 0x00                     |
| DATA[4] | NULL        | 0x00                     |
| DATA[5] | NULL        | 0x00                     |
| DATA[6] | NULL        | 0x00                     |
| DATA[7] | NULL        | 0x00                     |

### 2.34.3. 回复数据域定义

#### Reply Data Field Definition

| Data field | Description          | Data                      |
|------------|----------------------|---------------------------|
| DATA[0]    | 指令字节<br>Command byte | 0xB5                      |
| DATA[1]    | 标记位<br>Flag bit      | 固定为 0x01<br>Fixed to 0x01 |
| DATA[2]    | 索引<br>Index          | DATA[1] = (uint8_t)index  |
| DATA[3]    | Character 1          | Char1                     |
| DATA[4]    | Character 2          | Char2                     |
| DATA[5]    | Character 3          | Char3                     |
| DATA[6]    | Character 4          | Char4                     |
| DATA[7]    | Character 5          | Char5                     |

### 2.34.4. 索引说明

#### Index Description

|                    |  |
|--------------------|--|
| 索引值<br>Index value | 功能说明<br>Function Description   |
| 0x01               | 读取电机型号的第一到第五个字符。<br>Read the first to fifth characters of the motor model. |
| 0x02               | 读取电机型号的第六到第十个字符。   |



|      |   |
|------|---|
|      | Read the sixth to tenth characters of the motor model.                              |
| 0x03 | 读取电机型号的第十一到第十五个字符。<br>Read the eleventh to fifteenth characters of the motor model. |

### 2.34.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | Data[0] | Data[1] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x141 | 0xB5    | 0x01    | 0x01    | 0x00    | 0x00    | 0x00    | 0x00    | 0x00    |

RS485:

| Frame header | ID   | length | D0   | D1   | D2   | D3   | D4   | D5   | D6   | D7   | CRC16L | CRC16H |
|--------------|------|--------|------|------|------|------|------|------|------|------|--------|--------|
| 0x3E         | 0x01 | 0x08   | 0xB5 | 0x01 | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | CRC16L | CRC16H |

说明:

发送指令读取电机型号的第一到第五个字符。

Description:

Send a command to read the first to fifth characters of the motor model.

回复指令:

Reply command:

CAN:

| ID    | Data[0] | Data[1] | Data[2] | Data[3] | Data[4] | Data[5] | Data[6] | Data[7] |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x241 | 0xB5    | 0x01    | 0x01    | 0x58    | 0x38    | 0x53    | 0x32    | 0x56    |

RS485:

| Frame | ID | length | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | CRC16L | CRC16H |
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|
|-------|----|--------|----|----|----|----|----|----|----|----|--------|--------|

|        |      |      |      |      |      |      |      |      |      |      |        |        |
|--------|------|------|------|------|------|------|------|------|------|------|--------|--------|
| header |      |      |      |      |      |      |      |      |      |      |        |        |
| 0x3E   | 0x01 | 0x08 | 0xB5 | 0x01 | 0x01 | 0x58 | 0x38 | 0x53 | 0x32 | 0x30 | CRC16L | CRC16H |

说明:

该指令回复 5 个 ACSII 码, 通过查表得到电机型号的第一到第五个字符为: X8S2V。以上过程重复三次即可获取完整的电机型号。

#### Description:

This command returns 5 ASCII codes. By looking up the table, the first to fifth characters of the motor model are: X8S2V. Repeating the above process three times yields the complete motor model.

## 2.35. 主动回复功能指令 (0xB6)

### Active Reply Function Command (0xB6)

#### 2.35.1. 指令说明

##### Instruction Description

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。

This command is used to select the specified command to actively reply at a fixed time, and more than 1 command can be specified, and different commands will be cyclically and alternately replied according to the set time. If an active reply command is set, the motor will not reply after receiving the command. Only valid for CAN version, 485 version does not support this function.

#### 2.35.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description                          | Data   |
|------------|--------------------------------------|--|
| DATA[0]    | 指令字节<br>Command byte                 | 0xB6   |
| DATA[1]    | 指定主动回复的指令<br>Specify the command for | 回复指令包括: 0x60、0x61、0x62、0x92、0x9A、0x9C、0x9D、0x9E; |

|         |   |  |
|---------|---|--|
|         | proactive response  | Reply commands include: 0x60、0x61、0x62、0x92、0x9A、0x9C、0x9D、0x9E;   |
| DATA[2] | 主动回复使能位<br>Unsolicited reply enable bit                         | 0: 关闭该指令的主动回复功能;<br>1: 使能该指令的主动回复功能;<br>0: Turn off the active reply function of this command;<br>1: Enable the active reply function of this command; |
| DATA[3] | 回复间隔参数低 8 位<br>The lower 8 bits of the reply interval parameter | 回复间隔时间, 单位 10ms。多条指令回复时交替循环回复。<br>Reply interval time,unit 10ms. Alternate loop reply when replying multiple commands.                                 |
| DATA[4] | 回复间隔参数高 8 位<br>The high 8 bits of the reply interval parameter  |  |
| DATA[5] | NULL  | NULL   |
| DATA[6] | NULL  | NULL   |
| DATA[7] | NULL  | NULL   |

### 2.35.3. 回复数据域定义

#### Reply Data Field Definition

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。

After enabling it,the data will not be returned,and the motor will actively reply to the selected command content according to the set time interval.

### 2.35.4. 通讯示例

#### Example of Communication

示例 1:

发送指令:

Example 1:

Send command:

CAN:

| ID    | 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。

#### Description:

Enable 0x60 active reply command,the time interval is 20ms. After sending this command,the motor will not reply when receiving the command,but will reply 0x60 command at intervals of 10ms.

## 2.36. 功能控制指令 (0x20)

### Function Control Command (0x20)

#### 2.36.1. 指令说明

##### Instruction Description

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。只有在电机失能时, 才能将参数成功保存到 ROM。

This instruction is used to use some specific functions. It is a compound function instruction,which can contain multiple function control instructions.Be careful to avoid writing parameters when the motor has just started and is in motion.The parameters can be successfully saved to ROM only when the motor is disabled.

#### 2.36.2. 发送数据域定义

##### Send Data Field Definition

| Data field | Description            | Data                     |
|------------|------------------------|--------------------------|
| DATA[0]    | 指令字节<br>Command byte   | 0x20                     |
| DATA[1]    | 功能索引<br>Function index | DATA[1] = (uint8_t)index |
| DATA[2]    | NULL                   | 0x00                     |

|         |   |                                |
|---------|---|--------------------------------|
| DATA[3] | NULL                                    | 0x00                           |
| DATA[4] | 输入参数低字节 1<br>Input parameter low byte 1 | DATA[4] = (uint8_t)(Value)     |
| DATA[5] | 输入参数字节 2<br>Input parameter byte 2      | DATA[5] = (uint8_t)(Value>>8)  |
| DATA[6] | 输入参数字节 3<br>Input parameter byte 3      | DATA[6] = (uint8_t)(Value>>16) |
| DATA[7] | 输入参数字节 4<br>Input parameter byte 4      | DATA[7] = (uint8_t)(Value>>24) |

### 2.36.3. 回复数据域定义

#### Reply Data Field Definition

电机在收到指令后回复主机，该帧数据和主机发送的指令相同。

The motor replies to the host computer after receiving the command, and the frame data is the same as the command sent by the host computer.

### 2.36.4. 功能索引说明

#### Function Index Description

| 索引值<br>Index value | 指令名称<br>Command name               | 功能说明<br>Function description   |
|--------------------|------------------------------------|--|
| 0x01               | 清除多圈值<br>Clear multi-turn value    | 清除电机多圈值、更新零点并保存。重启后生效。<br>Clear motor multi-turn value, update zero point and save. It will take effect after restarting.        |
| 0x02               | CANID 滤波器使能<br>CANID filter enable | 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；<br>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；<br>此值会保存在 FLASH，掉电后会记录写入的数 |

|      |   |  |
|------|---|--|
|      |   | <p>值。</p> <p>The value “1” means that the CANID filter is enabled,which can improve the efficiency of motor sending and receiving in CAN communication;</p> <p>The value “0” means the disabled CANID filter,which needs to be disabled when the multi-motor control command 0x280,0x300 is required;</p> <p>This value will be saved in FLASH,and the written value will be recorded after power off.</p>   |
| 0x03 | <p>错误状态发送使能</p> <p>Error status transmission enable</p>                       | <p>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；</p> <p>Value 值为 0 代表失能该功能；</p> <p>The value “1” means that this function is enabled.</p> <p>After the motor appears in an error state,it actively sends the status command 0x9A to the bus with a sending cycle of 100ms. Stop sending after the error status disappears;</p> <p>The value “0” means the function is disabled.</p> |
| 0x04 | <p>多圈值掉电时保存使能</p> <p>The multi-turn value is saved when the power is off.</p> | <p>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；</p> <p>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。</p> <p>The value “1” means that this function is enabled,and the motor will save the current multi-turn value before powering off;</p>  |

|      |   |  |
|------|---|--|
|      |   | The value “0” means that this function is disabled; at this time,the system defaults to single lap mode; it will take effect after restarting.   |
| 0x05 | 设置 CANID<br>Set CANID   | Value 值代表将要修改的 CANID 号，保存到 ROM，重启后生效。<br>The value means the CANID number that is going to be modified, which will be saved to ROM and take effect after a reboot.                       |
| 0x06 | 设置位置运行最大正角度<br>Set the maximum positive angle for the position operation mode | 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。<br>The value represents the maximum positive angle value for the position operation mode, which is set and saved to ROM to take effect immediately. |
| 0x07 | 设置位置运行最大负角度<br>Set the maximum negative angle for the position operation mode | 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。<br>The value represents the maximum negative angle value for the position operation mode, which is set and saved to ROM to take effect immediately. |
| 0x08 | 错误状态自动恢复使能<br>Error State Auto-Recovery Enable                                | Value 值为 1 代表使能该功能，过压、低压、超速，三种错误状态将会在达到恢复条件后自动恢复；<br>Value 值为 0 代表失能该功能，过压、低压、超速，三种错误状态将不会自动恢复。<br>其余错误状态不受此功能控制。  |



|  |  |  |
|--|--|--|
|  |  | <p>When the Value is 1, the function is enabled. The three error states—overvoltage, undervoltage, and overspeed—will automatically recover once the recovery conditions are met.</p> <p>When the Value is 0, the function is disabled. The three error states—overvoltage, undervoltage, and overspeed—will not automatically recover.</p> <p>Other error states are not controlled by this function.</p> |
|--|--|--|

### 2.36.5. 通讯示例

#### Communication Example

示例 1:

发送指令:

**Example 1:**

**Send command:**

**CAN:**

| ID    | 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:**

| Frame header | ID   | Length | 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。

**Description:**

Data[1] = 0x01, according to the index value table, the representative function is to clear the multi-turn value.

回复指令:

### Reply command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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指令相同。

#### Description:

The frame data is the same as the command sent by the host.

#### 示例 2:

#### 发送指令:

#### Example 2:

#### Send command:

#### CAN:

| ID    | 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:

| Frame header | ID   | Length | 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, 注意使能后无法使用 0x280 多电机指令, 再次使用 0x280 指令前需要失能 CANID 滤波器。

#### Description:

Data[1] = 0x01, according to the index value table, the representative function is to

enable the CANID filter. Note that the 0x280 multi-motor command cannot be used after enabling, and the CANID filter needs to be disabled before using the 0x280 command again.

回复指令:

Reply command :

CAN:

| ID    | 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:

| Frame header | ID   | Length | 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指令相同。

Description:

The frame data is the same as the command sent by the host computer.

### 3. CAN 多电机指令 (0x280+指令)

CAN Multi-Motor Command (0x280 + Command)

#### 3.1. 指令说明

Instruction Description

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。

The ID number is 280, which means that multiple motors correspond to the same command at the same time. The content and function of the instruction are the same as those of the single-motor instruction.

## 3.2. 通讯示例

### Communication Example

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

Suppose there are 4 motors on the CAN bus, and the ID numbers are 141, 142, 143, and 144 respectively.

示例 1:

发送指令:

Example 1:

Send command:

| ID    | 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。

Description:

4 motors receive the 0x80 motor shutdown command at the same time (see 2.30 for details), and then all 4 motors immediately execute the motor shutdown command.

回复指令:

Reply command:

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。

4 motors reply at the same time, and the reply ID is their own ID number respectively. The reply sequence depends on the respective delays on the bus.

| 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。

Description:

The motor whose ID number is 0x241 returns the corresponding command.

| ID    | 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。

#### Description:

The motor whose ID number is 0x242 returns the corresponding command.

| ID    | 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。

#### Description:

The motor whose ID number is 0x243 returns the corresponding command.

| ID    | 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。

#### Description:

The motor whose ID number is 0x244 returns the corresponding command.

#### 示例 2:

发送指令:

#### Example 2:

#### Send command:

| ID    | 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。

### Description:

4 motors receive the 0x60 read multi-turn encoder position data command at the same time (see 2.21 for details), and then the 4 motors reply to their respective multi-turn encoder position data.

### 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。

### Reply command:

4 motors reply at the same time, and the reply ID is their own ID number respectively. The reply sequence depends on the respective delays on the bus.

| ID    | 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。

### Description:

The motor reply data with ID number 0x241 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00002710, which means the decimal is 10000. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 10000 pulses.

| ID    | 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。

### Description:

The motor reply data with ID number 0x242 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00004E20, which means 20000 in decimal. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 20000 pulses.

| 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。

#### Description:

The motor reply data with ID number 0x243 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00007530, which means 30000 in decimal. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 30000 pulses.

| 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。

#### Description:

In the response data of the motor with ID 0x244, Data[4] through Data[7] form a 32-bit data value (Data[4] is the lowest byte, Data[7] is the highest byte) of 0x00009C40, which represents decimal 40000. This indicates that the current multi- turn encoder value of the motor relative to the multi- turn zero offset (initial position) is 40,000 pulses.



## 4. 运动模式控制指令\_CAN (0x400 + ID)

### Motion Mode Control Command\_CAN (0x400 + ID)

#### 4.1. 指令说明

##### Instruction Description

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: -12.566 到 12.566, 单位 rad; 数据类型为 uint16\_t, 取值范围为 0~65535, 其中 0 代表-12.566, 65535 代表 12.566, 0~65535 中间的所有数值, 按比例映射至 -12.566~12.566。

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: 负电机最大扭矩到正电机最大扭矩, 单位 Nm (最大扭矩参数可在上位机中查看); 数据类型为 12 位无符号整数, 取值范围为 0~4095, 其中 0 代表-电机最大扭矩, 4095 代表电机最大扭矩, 0~4095 中间的所有数值, 按比例映射至-电机最大扭矩~电机最大扭矩。

功能表达式:

$$I_{qRef} = [kp * (p_{des} - p_{fd\_实际位置}) + kd * (v_{des} - v_{fb\_实际速度}) + t_{ff}] / KT\_OUT \text{ 扭矩系数};$$

I<sub>qRef</sub> 为最后给定电机的输出电流大小。

The command consists of 5 input parameters: p\_des (desired position), v\_des(desired velocity), t\_ff (feedforward torque), kp (position deviation coefficient), kd (speed deviation coefficient).

Each parameter has a preset range size:

p\_des: -12.566 to 12.566 in rad,the data type is uint16\_t, with a value range of 0 to 65535. Here, 0 represents -12.566, and 65535 represents 12.566. All values between 0 and 65535 are mapped proportionally to the range of -12.566 to 12.566.

v\_des: -45 to 45,in rad/s,the data type is a 12-bit unsigned integer, with a value range of 0 to 4095. Here, 0 represents -45, and 4095 represents 45. All values between 0 and 4095 are mapped proportionally to the range of -45 to 45.

kp: 0 to 1000,the data type is a 12-bit unsigned integer, with a value range of 0 to 4095. Here, 0 represents 0, and 4095 represents 1000. All values between 0 and 4095 are mapped proportionally to the range of 0 to 1000.

kd: 0 to 5,the data type is a 12-bit unsigned integer, with a value range of 0 to 4095. Here, 0 represents 0, and 4095 represents 5. All values between 0 and 4095 are mapped proportionally to the range of 0 to 5.

t\_ff: From -Motor Max Torque to +Motor Max Torque, in Nm (the Max Torque parameter can be viewed in the host software).The data type is a 12-bit unsigned integer, with a value range of 0 to 4095. Among these, 0 represents-Motor Max Torque, and 4095 represents+Motor Max Torque. All values between 0 and 4095 are proportionally mapped to the range from-Motor Max Torque to+Motor Max Torque.

Function expression:

$$I_{qRef} = [kp*(p\_des - p\_fd\_actual \text{ position}) + kd*(v\_des - v\_fb\_actual \text{ speed}) + t\_ff]/KT\_OUT\text{torque coefficient};$$

I<sub>qRef</sub> is the output current of the last given motor.

## 4.2. 发送数据域定义（大端字节序）

Send Data Field Definition (Big-endian byte order)

| 数据域<br>Data field | 数据划分<br>Data partition | 数据组合<br>Data combination | 数据定义<br>Data definition | 数据范围<br>Data range |
|-------------------|------------------------|--------------------------|-------------------------|--------------------|
| DATA[0]           | 4-7bit                 | p_des[8-15]              | p_des 的高 8 位数据          | 16 位范围             |
|                   | 0-3bit                 |                          | p_des Upper 8-bit data  | 16-bit range       |

|         |        |             |                        |              |
|---------|--------|-------------|------------------------|--------------|
| DATA[1] | 4-7bit | p_des[0-7]  | p_des 的低 8 位数据         |              |
|         | 0-3bit |             | p_des Lower 8-bit data |              |
| DATA[2] | 4-7bit | v_des[4-11] | v_des 的高 8 位数据         | 12 位范围       |
|         | 0-3bit |             | v_des Upper 8-bit data |              |
| DATA[3] | 4-7bit | v_des[0-3]  | v_des 的低 4 位数据         | 12-bit range |
|         | 0-3bit | kp[8-11]    | kp 的高 4 位数据            | 12 位范围       |
| DATA[4] | 4-7bit | kp[0-7]     | kp 的低 8 位数据            |              |
|         | 0-3bit |             | kp Lower 8-bit data    | 12-bit range |
| DATA[5] | 4-7bit | kd[4-11]    | kd 的高 8 位数据            | 12 位范围       |
|         | 0-3bit |             | kd Upper 8-bit data    |              |
| DATA[6] | 4-7bit | kd[0-3]     | kd 的低 4 位数据            | 12-bit range |
|         | 0-3bit | t_ff[8-11]  | t_ff 的高 4 位数据          | 12 位范围       |
| DATA[7] | 4-7bit | t_ff[0-7]   | t_ff 的低 8 位数据          |              |
|         | 0-3bit |             | t_ff Lower 8-bit data  | 12-bit range |

### 4.3. 回复数据域定义（大端字节序）

#### Reply Data Field Definition (Big-endian byte order)

| 数据域<br>Data field | 数据划分<br>Data partition | 数据组合<br>Data combination | 数据定义<br>Data definition                 | 数据范围<br>Data range     |
|-------------------|------------------------|--------------------------|---|------------------------|
| DATA[0]           | 7-0bit                 | CANID[0-7]               | 设备 CAN 地址号<br>Device CAN address number | 8 位范围<br>8-bit range   |
| DATA[1]           | 4-7bit                 | p[8-15]                  | 当前位置 p 的高 8 位数据                         | 16 位范围<br>16-bit range |
|                   | 0-3bit                 |                          | p_des Upper 8-bit data                  |                        |
| DATA[2]           | 4-7bit                 | p[0-7]                   | 当前位置 p 的低 8 位数据                         |                        |

|         |        |         |                        |              |
|---------|--------|---------|------------------------|--------------|
|         | 0-3bit |         | p_des Lower 8-bit data |              |
| DATA[3] | 4-7bit | v[4-11] | 当前速度 v 的高 8 位数据        | 12 位范围       |
|         | 0-3bit |         | v_des Upper 8-bit data |              |
| DATA[4] | 4-7bit | v[0-3]  | 当前速度 v 的低 4 位数据        | 12-bit range |
|         |        |         | v_des Lower 4-bit data |              |
|         | 0-3bit | t[8-11] | 当前力矩 t 的高 4 位数据        | 12 位范围       |
|         |        |         | t_ff Upper 4-bit data  |              |
| DATA[5] | 4-7bit | t[0-7]  | 当前力矩 t 的低 8 位数据        | 12-bit range |
|         | 0-3bit |         | t_ff Lower 8-bit data  |              |
| DATA[6] | 4-7bit | NULL    | NULL                   | NULL         |
|         | 0-3bit | NULL    | NULL                   | NULL         |
| DATA[7] | 4-7bit | NULL    | NULL                   | NULL         |
|         | 0-3bit | NULL    | NULL                   | NULL         |

#### 4.4. 通讯示例

##### Communication Example

示例 1:

发送指令: ID 号 0x401

Example 1:

Send command: ID number 0x401



| 数据域<br>Data field | 数据<br>Data | 数据划分<br>Data partition |     | 数据定义<br>Data definition               | 数据范围<br>Data range   | 数据计算说明<br>Data calculation instructions   |
|-------------------|------------|------------------------|-----|---------------------------------------|--|---|
| DATA[0]           | 0xE6       | 4-7bit                 | 0xE | p_des 值为 0xE666;                      | -12.5rad~12.5rad   | p_des=(58982/65535)*25+<br>(-12.5)=9.99rad  |
|                   |            | 0-3bit                 | 0x6 | 十进制为 (58982)                          | 共 25rad  |   |
| DATA[1]           | 0x66       | 4-7bit                 | 0x6 | p_des value is 0xE                    | (-)12.5rad~12.5rad   |   |
|                   |            | 0-3bit                 | 0x6 | 666 decimal is (58982)                | d total 25rad  |   |
| DATA[2]           | 0x82       | 4-7bit                 | 0x8 | v_des 值为 0x82E;                       | -45rad/s~45rad/s   | v_des=(2094/4095)*90+(-45)=1.021 rad/s  |
|                   |            | 0-3bit                 | 0x2 | 十进制为 (2094)                           | 共 90rad/s  |   |
| DATA[3]           | 0xE0       | 4-7bit                 | 0xE | v_des value is 0x8                    | (-)45rad/s~45rad/s   |   |
|                   |            | 0-3bit                 | 0x0 | 2E decimal is (2094)                  | total 90rad/s  |   |
| DATA[4]           | 0x52       | 4-7bit                 | 0x5 | kp 值为 0x52; 十进制为 (82)                 | 0~500 共 500  | kp=(82/4095)*500+0=10.012   |
|                   |            | 0-3bit                 | 0x2 | kp value is 0x52 decimal is (82)      | 0~500 total 500  |   |
| DATA[5]           | 0x33       | 4-7bit                 | 0x3 | kd 值为 0x333; 十进制为 (819)               | 0~50 共 50  | kd=(819/4095)*50=10   |
|                   |            | 0-3bit                 | 0x3 | kd value is 0x333 decimal is (819)    | 0~50 total 50  |   |
| DATA[6]           | 0x3B       | 4-7bit                 | 0x3 |                                       |  | t_ff=(2901/4095)*2*电机最大扭矩+(-电机最大扭矩)=0.416*电机最大扭矩<br>t_ff=(2901/4095)*2Motor Max Torque+(-Motor Max Torque)=0.416*Motor Max Torque |
|                   |            | 0-3bit                 | 0xB |                                       |  |   |
| DATA[7]           | 0x55       | 4-7bit                 | 0x5 | t_ff 值为 0xB55; 十进制为 (2901)            | -电机最大扭矩~电机最大扭矩 共 2*电机最大扭矩  |   |
|                   |            | 0-3bit                 | 0x5 | t_ff value is 0xB55 decimal is (2901) | -Motor Max Torque to+Motor Max Torque, total range of 2*Motor Max Torque |   |

回复指令：ID 号 0x501

Reply command: ID No. 0x501

| 数据域<br>Data field | 数据<br>Data | 数据划分<br>Data partition |      | 数据定义<br>Data definition  | 数据范围<br>Data range   | 数据计算说明<br>Data calculation instructions  |
|-------------------|------------|------------------------|------|--|--|--|
| DATA[0]           | 0x01       | 7-0bit                 | 0x1  | CANID  | 0-32   | 设备地址 ID 号<br>Device address ID number  |
| DATA[1]           | 0xE6       | 4-7bit                 | 0xE  | p 值为 0xE666;<br>十进制为 (58982)<br>p_des value is 0xE666 decimal is (58982) | -12.5rad~12.5rad<br>共 25rad<br>(-)12.5rad~12.5rad total 25rad  | $p\_des = (58982/65535) * 25 + (-12.5) = 9.99 \text{ rad}$   |
| DATA[2]           | 0x65       | 0-3bit                 | 0x6  |  |  |  |
| DATA[3]           | 0x82       | 4-7bit                 | 0x8  | v 值为 0x82E;<br>十进制为 (2094)<br>v_des value is 0x82E decimal is (2094)     | -45rad/s~45rad/s<br>共 90rad/s<br>(-)45rad/s~45rad/s total 90rad/s  | $v\_des = (2094/4095) * 90 + (-45) = 1.021 \text{ rad/s}$  |
| DATA[4]           | 0xEB       | 0-3bit                 | 0x2  |  |  |  |
| DATA[5]           | 0x55       | 4-7bit                 | 0xE  | t 值为 0xB55;<br>十进制为 (2901)<br>t_ff value is 0xB55 decimal is (2901)      | -电机最大扭矩~<br>电机最大扭矩<br>共 2 * 电机最大扭矩<br>(-)Motor Max Torque to (+)Motor Max Torque,<br>total range of 2*Motor Max Torque | $t\_ff = (2901/4095) * 2 * \text{电机最大扭矩} + (-\text{电机最大扭矩}) = 0.416 * \text{电机最大扭矩}$<br>$t\_ff = (2901/4095) * 2 * \text{Motor Max Torque} + (-\text{Motor Max Torque}) = 0.416 * \text{Motor Max Torque}$ |
|                   |            | 0-3bit                 | 0x5  |  |  |  |
| DATA[6]           | 0x0        | 4-7bit                 | NULL | NULL   | NULL   | NULL   |



|         |     |        |      |      |      |      |
|---------|-----|--------|------|------|------|------|
|         |     | 0-3bit | NULL | NULL | NULL | NULL |
| DATA[7] | 0x0 | 4-7bit | NULL | NULL | NULL | NULL |
|         |     | 0-3bit | NULL | NULL | NULL | NULL |

## 5. RS485 多电机指令 (0xCD + 指令)

### RS485 Multi-Motor Command (0xCD + Command)

#### 5.1. 指令说明

##### Instruction Description

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。

The ID number is 0xCD, which means that multiple motors correspond to the same command at the same time. The content and function of the instruction are the same as those of the single-motor instruction. For details, please refer to the single-motor instruction.

#### 5.2. 通讯示例

##### Communication Example

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

Suppose there are 4 motors on the RS485 bus, and the ID numbers are 01, 02, 03, 04 respectively.

示例 1:

发送指令:

Example 1:

Send command:

| Frame header | ID   | Length | 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。

#### Description:

4 motors receive the 0x80 motor shutdown command at the same time (see 2.30 for details), and then all 4 motors immediately execute the motor shutdown command.

#### 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。

#### Reply command:

4 motors reply at the same time, and the reply ID is their own ID number respectively. The reply sequence depends on the respective delays on the bus.

| Frame header | ID   | Length | 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。

#### Description:

The motor whose ID number is 0x01 returns the corresponding command.

| Frame header | ID   | Length | 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。

#### Description:

The motor whose ID number is 0x02 returns the corresponding command.

| Frame header | ID   | Length | 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。

#### Description:

The motor whose ID number is 0x03 returns the corresponding command.

| Frame header | ID   | Length | 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。

#### Description:

The motor whose ID number is 0x04 returns the corresponding command.

示例 2:

发送指令:

Example 2:

Send command:

| Frame header | ID   | Length | 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。

#### Description:

4 motors receive the 0x60 read multi-turn encoder position data command at the same time (see 2.21 for details), and then the 4 motors reply to their respective multi-turn encoder position data.

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。

### Reply command:

4 motors reply at the same time, and the reply ID is their own ID number respectively. The reply sequence depends on the respective delays on the bus.

| Frame header | ID   | Length | 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。

### Description:

The motor reply data with ID number 0x01 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00002710, which means the decimal is 10000. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 10000 pulses.

| Frame header | ID   | Length | 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。

### Description:

The motor reply data with ID number 0x02 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00004E20, which means 20000 in decimal. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 20000 pulses.

| Frame header | ID   | Length | 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。

**Description:**

The motor reply data with ID number 0x03 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00007530, which means 30000 in decimal. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 30000 pulses.

| Frame header | ID   | Length | 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。

**Description:**

The motor reply data with ID number 0x04 consists of Data[4] to data[7] (Data[4] is the lowest bit, Data[7] is the highest bit). The 32-bit data is 0x00009C40, which means 40000 in decimal. The multi-turn encoder value representing the current relative multi-turn zero offset (initial position) of the motor is 40000 pulses.

## 6. RS485 运动模式控制指令

### RS485 Motion Mode Control Command

#### 6.1. 指令说明

##### Instruction Description

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)。上述参数均以带减速器的关节输出端参数进行描述。

##### 功能表达式：

$$IqRef = [kp * (p\_des - p\_fb\_实际位置) + kd * (v\_des - v\_fb\_实际速度) + t\_ff] / KT\_OUT$$

扭矩系数;IqRef 为最后给定电机的输出电流大小。

This command consists of five input parameters: p\_des (desired position), v\_des (desired velocity), t\_ff (feedforward torque), kp (position error coefficient), and kd (velocity error coefficient). All of the above parameters are described in terms of the joint output side with a reducer.

##### Functional Expression:

$$IqRef = [kp * (p\_des - p\_fb\_actual\ position) + kd * (v\_des - v\_fb\_actual\ velocity) + t\_ff] / KT\_OUT$$

(torque coefficient); IqRef is the final output current magnitude supplied to the motor.

#### 6.2. 发送数据域定义（小端字节序）

##### Transmit Data Field Definition (Little-Endian Byte Order)

| 字节<br>Byte | 变量名<br>Variable Name | 说明<br>Description                          |
|------------|----------------------|--|
| 1          | Start0               | 帧头，固定为 0xFE<br>Frame header, fixed to 0xFE |
| 2          | Start1               | 帧头，固定为 0xEE<br>Frame header, fixed to 0xEE |
| 3          | Motor_ID             | 电机编号，可以为 0x01、0x02、0x03、0xBB。(0xBB         |

|    |       |  |
|----|-------|--|
|    |       | 代表向所有电机广播)<br>Motor ID, can be 0x01, 0x02, 0x03, or 0xBB (0xBB indicates broadcast to all motors)                |
| 4  | Mode  | 电机运行模式, 可为 0x00(停转)、0x0A(闭环伺服控制)<br>Motor operation mode, can be 0x00 (stop) or 0x0A (closed-loop servo control) |
| 5  | t_ff  | 电机前馈力矩, float 型数据, 单位为 Nm<br>Motor feedforward torque, float data type, unit: Nm                                 |
| 6  |       |  |
| 7  |       |  |
| 8  |       |  |
| 9  | v_des | 电机目标转速, float 型数据, 单位为 rad/s<br>Motor target speed, float data type, unit: rad/s                                 |
| 10 |       |  |
| 11 |       |  |
| 12 |       |  |
| 13 | p_des | 电机目标位置, float 型数据, 单位为 rad<br>Motor target position, float data type, unit: rad                                  |
| 14 |       |  |
| 15 |       |  |
| 16 |       |  |
| 17 | kp    | 电机位置刚度 kp, float 型数据<br>Motor position stiffness kp, float data type   |
| 18 |       |  |
| 19 |       |  |
| 20 |       |  |
| 21 | kd    | 电机位置刚度 kd, float 型数据<br>Motor position stiffness kd, float data type   |
| 22 |       |  |
| 23 |       |  |
| 24 |       |  |

|    |          |                             |
|----|----------|-----------------------------|
| 25 | CRC_Data | CRC32 校验码<br>CRC32 checksum |
| 26 |          |                             |
| 27 |          |                             |
| 28 |          |                             |

### 6.3. 回复数据域定义（小端字节序）

#### Response Data Field Definition (Little-Endian Byte Order)

| 字节<br>Byte | 变量名<br>Variable Name | 说明<br>Description  |
|------------|----------------------|--|
| 1          | Start0               | 帧头，固定为 0xFE<br>Frame header, fixed to 0xFE   |
| 2          | Start1               | 帧头，固定为 0xEE<br>Frame header, fixed to 0xEE   |
| 3          | Motor_ID             | 电机编号，可以为 0x01、0x02、0x03。（广播模式下电机不返回状态）<br>Motor ID, can be 0x01, 0x02, or 0x03 (motors do not return status in broadcast mode) |
| 4          | Mode                 | 电机当前运行模式<br>Motor current operation mode   |
| 5          | Temp                 | 电机当前温度，数据类型为 int8_t，单位为摄氏度<br>Motor current temperature, data type int8_t, unit: °C  |
| 6          | Motor_Err            | 错误状态（报文中为小端字节序，但以大端字节序进行说明）：<br>0x0002：电机堵转<br>0x0004：低压<br>0x0008：过压<br>0x0010：相电流过流<br>0x0080：元器件过温<br>0x1000：电机温度过温         |
| 7          |                      |  |



|    |          |  |
|----|----------|--|
|    |          | <p>0x2000: 编码器校准错误</p> <p>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。</p> <p>Error status (little- endian in the message, but described in big- endian):</p> <p>0x0002: Motor stall</p> <p>0x0004: Undervoltage</p> <p>0x0008: Overvoltage</p> <p>0x0010: Phase current overcurrent</p> <p>0x0080: Component overtemperature</p> <p>0x1000: Motor overtemperature</p> <p>0x2000: Encoder calibration error</p> <p>When multiple errors occur simultaneously, the error status bits are added together. For example, a value of 0x0016 indicates 0x2 + 0x4 + 0x10, meaning that the motor has three errors at that time: stall, undervoltage, and phase current overcurrent.</p> |
| 8  | Reserved | 保留<br>Reserved   |
| 9  | T        | 电机当前输出力矩, float 型数据, 单位为 Nm<br>Motor current output torque, float data type, unit: Nm  |
| 10 |          |  |
| 11 |          |  |
| 12 |          |  |
| 13 | W        | 电机当前实际转速, float 型数据, 单位为 rad/s<br>Motor current actual speed, float data type, unit: rad/s   |
| 14 |          |  |
| 15 |          |  |

|    |          |   |
|----|----------|---|
| 16 |          |   |
| 17 | Pos      | 电机当前实际位置，float 型数据，单位为 rad<br>Motor current actual position, float data type, unit: rad |
| 18 |          |   |
| 19 |          |   |
| 20 |          |   |
| 21 | CRC_Data | CRC32 校验码<br>CRC32 checksum   |
| 22 |          |   |
| 23 |          |   |

## 7. 指示灯说明

### Indicator Light Description

#### 7.1. 状态说明

##### Status Description

- 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。
- When the indicator light is solid on,it means the motor is running normally;
- Slow flashing indicates that the motor has a secondary error. If the recovery condition is reached,it will automatically return to normal operation,and the indicator light will be solid on for a long time;
- Flickering quickly indicates that the motor has a first-level error,and the motor cannot recover from the error. It is necessary to check the motor fault and restart before it can continue to run.

## 7.2. 故障说明表格

Failure Description Form

| 故障名称<br>Fault Name                  | 说明<br>Description  | 错误等级<br>Error Level |
|-------------------------------------|--|---------------------|
| 硬件过流<br>Hardware over-current       | 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。<br>If the motor current exceeds the limit value, there may be short circuit, phase loss, loss of control, motor damage, etc.                                | 一级<br>Level 1       |
| 堵转错误<br>Stall error                 | 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。<br>After the current reaches the stall current, the speed is very low and continues for a period of time. It indicates that the motor load is too large.   | 一级<br>Level 1       |
| 欠压错误<br>Under-voltage error         | 电源输入低于设定欠压值<br>The power input is lower than the set undervoltage value  | 二级<br>Level 2       |
| 过压错误<br>Over-voltage error          | 电源输入高于设定过压值<br>The power input is higher than the set overvoltage value  | 二级<br>Level 2       |
| 相电流过流<br>Phase-current over-current | 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。<br>The software detects that the motor current exceeds the limit value, and there may be short circuit, phase loss, loss of control, motor damage, etc. | 一级<br>Level 1       |
| 功率超限错误                              | 电源输入电流超过限制值, 可能存在负   | 二级                  |

|   |   |                          |
|---|---|--------------------------|
| Power overrun error                                     | <p>载过大或者转速过高的情况。</p> <p>If the input current of the power supply exceeds the limit value,there may be a situation where the load is too large or the speed is too high.</p> | Level 2                  |
| <p>标定参数读取错误</p> <p>Calibration parameter read error</p> | <p>写入参数失败引起参数丢失</p> <p>Failed to write parameters causing parameter loss.</p>   | <p>一级</p> <p>Level 1</p> |
| <p>超速错误</p> <p>Over-speed error</p>                     | <p>电机运行转速超过极限值，可能存在超压、拖拽使用。</p> <p>The motor running speed exceeds the limit value,there may be over pressure and drag use.</p>   | <p>二级</p> <p>Level 2</p> |
| <p>电机过温错误</p> <p>Motor over-temperature error</p>       | <p>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。</p> <p>If the motor temperature exceeds the set value,there may be short circuit,parameter error,and long-term overload use.</p>       | <p>二级</p> <p>Level 2</p> |
| <p>编码器校准错误</p> <p>Encoder calibration error</p>         | <p>编码器校准结果与标准值偏差过大。</p> <p>The encoder calibration result deviates too much from the standard value.</p>  | <p>二级</p> <p>Level 2</p> |

## 8. 版本修订信息

### Version Revision Information

#### 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 指令 (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 指令 (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指令 (0xA9)
- b. 在速度闭环控制指令 (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

#### V4.4 版本:

1) 版本修订内容:

- a. 加入 TF 指令 (0x73 带前馈扭矩的位置控制指令)
- b. 完善力控位置闭环控制指令 (0xA9)
- c. MIT 指令中的 kd 参数取值范围由 0~5 拓展为 0~50
- d. MIT 指令中的 t<sub>ff</sub> 参数取值范围由 -24NM~24NM 改为 -电机最大扭矩~电机最大扭矩
- e. 读取单圈角度指令 (0x94) 返回的单圈值范围更改为 ±180°, 数据类型为 int32\_t 类型, 有效数据 4 个字节, 单位 0.01°/LSB
- f. 为电机型号读取指令 (0xB5) 添加更详细的使用说明
- g. 功能控制指令 (0x20) 中加入错误状态自动恢复使能
- h. 删除读取单圈编码器指令 (0x90) 和 CANID 设置指令 (0x79)
- i. 在“读取电机状态 1 和错误标志指令 (0x9A)”中补充电机错误的触发和恢复条



件。

j. 修改 TF 指令中的控制值 Feedforward Torque 数据类型为 int8\_t 类型, 取值范围为 -128 ~ 127,

k. CAN 通讯的 MIT 指令中的 p\_des 取值范围由 -12.5 到 12.5 改为 -12.566 到 12.566, 单位 rad;

1. 加入 SF 指令 (0x72 带前馈扭矩的位置控制指令)。

2) 版本修改日期: 2026.03.25

#### Version V3.1:

1) Version revision content:

a. Revise the definition of reply data in 5.0 operation control command.

2) Version revision date: 2022.6.23

#### Version V3.2:

1) Version revision content:

a. Add the description of indicator lights.

2) Version revision date: 2022.7.27

#### Version V3.3:

1) Version revision content:

a. Add function control command 0x20 a function: add CAN filter disable control function.

2) Version revision date: 2022.7.31

#### Version V3.4:

1) Version revision content:

a. Add position tracking instruction 0xA3;

b. In the 0x43 command, add the settings of 4 values of acceleration and deceleration for position planning and speed planning.

2) Version revision date: 2022.8.17

#### Version V3.5:

1) Version revision content:

a. Increase the position tracking command 0xA5 with speed limit;

b.Add function control command 0x20: error status sending and multi-turn value power-down save selection function;

c.Add the 0xB5 command to read the motor model.

2) Version revision date: 2022.9.05

#### **Version V3.6:**

1) Version revision content:

a.Increase RS485 broadcast instruction description 0xCD.

2) Version revision date:2022.10.13

#### **Version V3.7:**

1) Version revision content:

a.Remove the A3 instruction;

b.Merge A5 into A4;

c.Add A6 single-turn position command;

d.Add the command 0x90 to read the single-turn encoder;

e.Add the command 0x94 to read the single-turn angle of the motor.

2)Version revision date:2022.11.26

#### **Version V3.8:**

1)Version revision content:

a.The RS485 protocol baud rate of 2Mbps is modified to 2.5Mbps.

2)Version revision date:2022.11.26

#### **Version V3.9:**

1)Version revision content:

a.Add 485 serial port configuration instructions;

b.Add the function index in the 0x42 command,and you can read the acceleration and deceleration values of position and speed through the index;

c.Add 0xB6 active reply function.

2)Version revision date:2023.3.11

#### **Version V4.0:**

1)Version revision content:

- a.Add the function of setting CANID in the 0x20 command;
- b.Add the maximum positive angle limit value in the 0x20 command;
- c.Add the maximum negative angle limit value in the 0x20 command.

2)Version revision date: 2023.10.16

#### **Version V4.1:**

1) Version revision content:

- a.Modify the circleAngle of the motor to be the uint16\_t type data.

2)Version revision date: 2024.2.13

#### **Version V4.2:**

1) Version revision content:

- a.Modify the Read PID Parameters command (0x30) to read the PID parameters of the current,speed,and position loops using the index;
- b.Modify the Write PID Parameters to RAM command (0x31),and use the index to write the PID parameters of the current loop,velocity loop,and position loop to RAM;
- c.Modify the Write PID Parameters to ROM command (0x32),use the index to write the PID parameters of the current loop,speed loop and position loop to ROM.

2)Version revision date: 2024.5.28

#### **Version V4.3:**

1) Version revision content:

- a. Added force control position closed-loop control command (0xA9);
- b. Included force control mode in the speed closed-loop control command (0xA2);
- c. Improved MIT command;
- d. Added encoder data abnormal error.

2) Version revision date: 2024.5.12

#### **V4.4 Version:**

1) Revision Content:

- a. Added the TF command (0x73, position control command with feedforward torque).
- b. Improved the force-controlled position closed-loop control command (0xA9).
- c. Expanded the value range of the kd parameter in the MIT command from 0–5 to 0–50.

- d. Changed the value range of the  $t_{ff}$  parameter in the MIT command from  $-24\text{ Nm}$  to  $+24\text{ Nm}$  to  $-\text{Motor Max Torque}$  to  $+\text{Motor Max Torque}$ .
- e. Changed the single-turn value returned by the read single-turn angle command (0x94) to a range of  $\pm 180^\circ$ , data type `int32_t`, 4 valid bytes, unit  $0.01^\circ/\text{LSB}$ .
- f. Added more detailed usage instructions for the motor model read command (0xB5).
- g. Added error state auto-recovery enable to the function control command (0x20).
- h. Removed the read single-turn encoder command (0x90) and the CAN ID setting command (0x79).
- i. Added motor error trigger and recovery conditions to the "Read Motor Status 1 and Error Flags" command (0x9A).
- j. Modified the data type of the Feedforward Torque control value in the TF command to `int8_t` with a range of  $-128$  to  $127$ .
- k. Changed the  $p_{des}$  value range in the CAN communication MIT command from  $-12.5$  to  $12.5\text{ rad}$  to  $-12.566$  to  $12.566\text{ rad}$ .

2) Revision Date: March 25, 2026