Unit-RollerCAN

CAN 控 制 协 议

目录

1、通信协议结构

1.1 通信协议参数

采用 CAN 2.0 通讯接口，通信速率为 1Mbps，采用扩展帧格式。以下通信参数均使用十六进制进行表示。

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	指令代码	主机 ID+参数配置	目标 ID	数据区

2、配置寄存器

2.1 设备 ID 检测

- 功能说明： 请求指定 ID，可检查设备通信是否正常。
- 指令代码： 00H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	00H	bit8~15：主机 CAN ID	目标电机 CAN ID	00 00 00 00 00 00 00 00

- 发送案例：

功能	29bit 数据帧 ID	数据帧
获取设备 ID	00 00 00 A8	00 00 00 00 00 00 00 00

- 响应案例：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	00H	目标电机 CAN ID	FE	00

2.2 电机反馈数据

- 功能说明：用于获取电机的当前状态信息，包括速度、位置、电流和电压等。

- 指令代码： 02H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	02H	bit8~bit15:当前电机 CAN ID bit16~18:故障信息（0 无 1 有） bit18: 超范围 bit17: 堵转 bit16: 过压故障 bit19~21:模式 1: Speed Mode 2: Position Mode 3: Current Mode 4：Encoder Mode bit22~23:状态 0: Standby 1: Running 2: Error	主机 CAN ID	byte0~1: 当前速度 [-32768~32767]，单位（rpm） byte2~3: 当前位置 [-32768~32767]，单位（°） byte4~5:当前电流 [-32768~32767]，单位（mA） byte6~7:当前输入电压 [-32768~32767]，单位（V）

2.3 电机使能运行

- 功能说明：启用电机并进入运行状态。该指令用于启动电机，使其进入预设的运行模式。
- 指令代码： 03H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	03H	bit15~8：用来标识主机 CAN ID	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧
电机使能运行	030000A8	00 00 00 00 00 00 00 00

- 响应案例：

参考 2.2 电机反馈数据

2.4 电机停止运行

- 功能说明：停止电机的运行。发送此指令可立即停止电机的动作。
- 指令代码： 04H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	04H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧
电机使能运行	04 00 00 A8	00 00 00 00 00 00 00 00

- 响应案例：

参考 2.2 电机反馈数据

2.5 设置电机 CAN ID

- 功能说明：设置电机的 CAN 通信 ID。通过此指令可为电机分配唯一的通信标识符。
- 指令代码： 07H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	07H	bit8~15：用来标识主机 CAN ID bit16~23：预设置 CAN ID	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧	备注
设置电机 ID	07 01 00 A8	00 00 00 00 00 00 00 00	设置电机 ID 为 01
设置电机 ID	07 A8 00 01	00 00 00 00 00 00 00 00	设置电机 ID 为 A8

- 响应案例：

  参考 2.1 设备 ID 检测

2.6 解除电机堵转保护

- 功能说明：解除电机的堵转保护状态。用于清除电机堵转后的保护锁定，使电机恢复运行。
- 指令代码： 09H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	09H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧
解除电机堵转保护	09 00 00 A8	00 00 00 00 00 00 00 00

2.7 保存参数到 flash

- 功能说明：将当前参数保存到 Flash 中。该指令可将当前的电机设置永久保存在设备内部。
- 指令代码： 0AH
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	0AH	bit8~15：用来标识主机 CAN ID bit16~23：预设置 CAN ID	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧

保存参数到 flash	0A 00 00 A8	00 00 00 00 00 00 00 00

- 响应案例：

  参考 2.2 电机反馈数据

2.8 设置电机 CAN BPS

- 功能说明：设置电机的 CAN 通信速率。此功能可调节电机与主机之间的通信速度。
- 指令代码： 0BH
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	0BH	bit8~15：用来标识主机 CAN ID bit16~23：预设置 CAN BPS (0:1Mbps,1:500Kbps,2:125Kbps)	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧
设置电机通信为 1Mbps	0B 00 00 A8	00 00 00 00 00 00 00 00
设置电机通信为 500Kbps	0B 01 00 A8	00 00 00 00 00 00 00 00
设置电机通信为 125Kbps	0B 02 00 A8	00 00 00 00 00 00 00 00

- 响应案例：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	0BH	bit8~15：用来标识主机 CAN ID bit16~23：预设置 CAN BPS (0:1Mbps,1:500Kbps,2:125Kbps)	电机 CAN ID	0

2.9 电机开启堵转保护

- 功能说明：开启电机的堵转保护功能。该功能可自动检测堵转并保护电机免受损坏。
- 指令代码： 0CH
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	0CH	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	0
响应案例：：应答电机反馈帧（见 2.2 电机反馈数据）				

- 发送案例：

功能	29bit 数据帧 ID	数据帧
电机开启堵转保护	0C0000A8	00 00 00 00 00 00 00 00

2.10 电机关闭堵转保护

- 功能说明：关闭电机的堵转保护功能。用于禁用电机的自动堵转保护机制。
- 指令代码： 0DH
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	0DH	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	0

- 发送案例：

功能	29bit 数据帧 ID	数据帧
电机关闭堵转保护	0D 00 00 A8	00 00 00 00 00 00 00 00

- 响应案例：

参考 2.2 电机反馈数据

2.11 电机参数读取

- 功能说明：读取电机的相关参数信息。该指令可返回电机的速度、位置等实时数据。
- 指令代码： 11H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	11H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	byte0~1: index byte2~3: 00 byte4~7:00

- 发送案例：

功能	29bit 数据帧 ID	数据帧
读取电机使能状态	11 00 00 A8	04 70 00 00 00 00 00 00
读取电机模式	11 00 00 A8	05 70 00 00 00 00 00 00
读取电机电流	11 00 00 A8	06 70 00 00 00 00 00 00
读取电机速度	11 00 00 A8	0A 70 00 00 00 00 00 00
读取电机位置	11 00 00 A8	16 70 00 00 00 00 00 00
读取电机位置模式最大电流	11 00 00 A8	17 70 00 00 00 00 00 00
读取电机速度模式最大电流	11 00 00 A8	18 70 00 00 00 00 00 00
读取电机速度模式的 Kp	11 00 00 A8	20 70 00 00 00 00 00 00
读取电机速度模式的 Ki	11 00 00 A8	21 70 00 00 00 00 00 00
读取电机速度模式的 Kd	11 00 00 A8	22 70 00 00 00 00 00 00
读取电机位置模式的 Kp	11 00 00 A8	23 70 00 00 00 00 00 00
读取电机位置模式的 Ki	11 00 00 A8	24 70 00 00 00 00 00 00
读取电机位置模式的 Kd	11 00 00 A8	25 70 00 00 00 00 00 00
读取电机实时 speed	11 00 00 A8	30 70 00 00 00 00 00 00
读取电机实时位置	11 00 00 A8	31 70 00 00 00 00 00 00
读取电机实时电流	11 00 00 A8	32 70 00 00 00 00 00 00
读取电机输入电压值	11 00 00 A8	34 70 00 00 00 00 00 00
读取电机温度	11 00 00 A8	35 70 00 00 00 00 00 00
读取 RGB 灯模式	11 00 00 A8	50 70 00 00 00 00 00 00
读取 RGB 灯颜色	11 00 00 A8	51 70 00 00 00 00 00 00
读取 RGB 灯亮度	11 00 00 A8	52 70 00 00 00 00 00 00

- 响应案例：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	11H	bit8~15：用来标识主机 CAN ID	电机 CAN ID	byte0~1: index，参数列表，详见表 1 byte2~3: 00 byte4~7:参数数据，1 字节，数据在 byte4

2.12 电机参数写入

- 功能说明：将设定的参数写入电机。通过此指令可以更改电机的工作参数。
- 指令代码： 12H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	12H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	byte0~1: index，详见表 1 byte2~3: 00 byte4~7:参数数据

- 发送案例：

功能	29bit 数据帧 ID	数据帧	备注
设置电机电流	12 00 00 A8	06 70 00 00 40 0D 03 00	
速度模式			
设置电机模式为 speed mode	12 00 00 A8	05 70 00 00 01 00 00 00	
设置电机的 speed 速度 3000rpm	12 00 00 A8	0A 70 00 00 E0 93 04 00	16V 实际电机转速 2340rpm
设置电机的 speed 电流 2000mA	12 00 00 A8	18 70 00 00 40 0D 03 00	16V 实际电机工作电流 80-100mA
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
设置电机的 speed 速度 -3000rpm	12 00 00 A8	0A 70 00 00 20 6C FB FF	16v 实际电机转速-2340rpm
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
位置模式			

设置电机为 position	12 00 00 A8	05 70 00 00 02 00 00 00	
设置电机的位置 20000	12 00 00 A8	16 70 00 00 80 84 1E 00	
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
设置电机的位置-20000	12 00 00 A8	16 70 00 00 80 7B E1 FF	
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
电流模式			
设置电机为 current mode	12 00 00 A8	05 70 00 00 03 00 00 00	
设置电机的 current3000	12 00 00 A8	06 70 00 00 E0 93 04 00	上限显示 1200，16v 实际电机工作电流 80—90mA
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
设置电机的 current -3000	12 00 00 A8	06 70 00 00 20 6C FB FF	下限显示-1200，16v 实际电机工作电流 80—90mA
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
编码器模式			
设置电机为 encoder mode	12 00 00 A8	05 70 00 00 04 00 00 00	
设置电机的 encoder 20000	12 00 00 A8	33 70 00 00 80 84 1E 00	
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
设置电机的 encoder -20000	12 00 00 A8	33 70 00 00 80 7B E1 FF	
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
设置速度模式 PID			
设置电机为 speed mode	12 00 00 A8	05 70 00 00 01 00 00 00	
速度模式的 Kp X100000 Int 15	12 00 00 A8	20 70 00 00 00 6A 18 00	
速度模式的 Ki X10000000 Int 0.02	12 00 00 A8	21 70 00 00 40 0D 03 00	
速度模式的 Kd X100000 Int 500	12 00 00 A8	22 70 00 00 80 F0 FA 02	
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
设置位置模式 PID			
设置电机为 position mode	12 00 00 A8	05 70 00 00 02 00 00 00	
位置模式的的 Kp X100000 Int 15	12 00 00 A8	23 70 00 00 00 6A 18 00	
位置模式的的 Ki X10000000 Int 0.02	12 00 00 A8	24 70 00 00 40 0D 03 00	

位置模式的的 Kd X100000 Int 500	12 00 00 A8	25 70 00 00 80 F0 FA 02	
设置电机启动	12 00 00 A8	04 70 00 00 01 00 00 00	
设置电机关闭	12 00 00 A8	04 70 00 00 00 00 00 00	
其他设置			
写入大于 1 的数，保存参数到 flash	12 00 00 A8	02 70 00 00 02 00 00 00	
写入大于 1 的数，解除电机堵转保护	12 00 00 A8	03 70 00 00 02 00 00 00	
设置 RGB 自定义模式	12 00 00 A8	50 70 00 00 01 00 00 00	
设置 RGB 亮度	12 00 00 A8	52 70 00 00 32 00 00 00	
设置 RGB 颜色 白色	12 00 00 A8	51 70 00 00 FF FF FF 00	
设置 RGB 颜色 紫色	12 00 00 A8	51 70 00 00 80 00 80 00	
设置 RGB 颜色 灰色	12 00 00 A8	51 70 00 00 80 80 80 00	
设置 RGB 颜色 蓝色	12 00 00 A8	51 70 00 00 FF 00 00 00	

- 响应案例：

参考 2.2 电机反馈数据

3、 CAN to I2C 转发控制指令集

该指令集用于发送控制指令，通过 CAN 接口指令的方式，实现 Roller485 电机 I2C 端口的数据读写。

3.1 I2C 从机参数读取

通过 CAN 转 I2C 接口控制更多的 Roller 系列电机

- 功能说明：通过 CAN 指令读取 I2C 从机的参数信息。用于从连接的 I2C 设备获取数据。
- 指令代码： 13H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	13H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	byte0~1: index 详见表 1 byte2~3: I2C Address

				byte4~7: 00

- 发送案例：

注：A8 是主电机的 id，64 是从电机的 id

功能	29bit 数据帧 ID	数据帧
电机状态与模式读取		
读取电机使能状态	13 00 00 A8	04 70 64 00 00 00 00 00
读取电机模式	13 00 00 A8	05 70 64 00 00 00 00 00
读取电机电流	13 00 00 A8	06 70 64 00 00 00 00 00
读取电机速度	13 00 00 A8	0A 70 64 00 00 00 00 00
读取电机位置	13 00 00 A8	16 70 64 00 00 00 00 00
控制参数读取		
读取电机位置模式最大电流	13 00 00 A8	17 70 64 00 00 00 00 00
读取电机速度模式最大电流	13 00 00 A8	18 70 64 00 00 00 00 00
读取电机速度模式的 Kp	13 00 00 A8	20 70 64 00 00 00 00 00
读取电机速度模式的 Ki	13 00 00 A8	21 70 64 00 00 00 00 00
读取电机速度模式的 Kd	13 00 00 A8	22 70 64 00 00 00 00 00
读取电机位置模式的 Kp	13 00 00 A8	23 70 64 00 00 00 00 00
读取电机位置模式的 Ki	13 00 00 A8	24 70 64 00 00 00 00 00
读取电机位置模式的 Kd	13 00 00 A8	25 70 64 00 00 00 00 00
电机实时数据读取		
读取电机实时 speed	13 00 00 A8	30 70 64 00 00 00 00 00
读取电机实时位置	13 00 00 A8	31 70 64 00 00 00 00 00
读取电机实时电流	13 00 00 A8	32 70 64 00 00 00 00 00
读取电机输入电压值	13 00 00 A8	34 70 64 00 00 00 00 00
读取电机温度	13 00 00 A8	35 70 64 00 00 00 00 00
电机 RGB 模式与显示		
读取电机 RGB 模式	13 00 00 A8	50 70 64 00 00 00 00 00
读取电机 RGB 颜色	13 00 00 A8	51 70 64 00 00 00 00 00
读取电机 RGB 亮度	13 00 00 A8	52 70 64 00 00 00 00 00

- 响应案例：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7

描述	13H	bit8~15：用来标识主机 CAN ID	电机 CAN ID	byte0~1: index，参数列表，详见表 1 byte2~3: 00 byte4~7:参数数据，1 字节，数据在 byte4

3.2 I2C 从机参数写入

- 功能说明：通过 CAN 指令写入 I2C 从机的参数信息。可通过该指令向 I2C 设备发送控制或设定数据。
- 指令代码： 14H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	14H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	byte0~1: index，详见表 1 byte2~3: I2C Address byte4~7:参数数据

- 发送案例：

功能	29bit 数据帧 ID	数据帧
设置电机电流	14 00 00 A8	06 70 64 00 40 0D 03 00
速度模式		
设置电机模式为 speed mode	14 00 00 A8	05 70 64 00 01 00 00 00
设置电机的 speed 速度 3000rpm	14 00 00 A8	0A 70 64 00 E0 93 04 00
设置电机的 speed 电流 2000mA	14 00 00 A8	18 70 64 00 40 0D 03 00
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
设置电机的 speed 速度 -3000rpm	14 00 00 A8	0A 70 64 00 20 6C FB FF
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
位置模式		
设置电机为 position	14 00 00 A8	05 70 64 00 02 00 00 00

设置电机的位置 20000	14 00 00 A8	16 70 64 00 80 84 1E 00
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
设置电机的位置-20000	14 00 00 A8	16 70 64 00 80 7B E1 FF
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
电流模式		
设置电机为 current mode	14 00 00 A8	05 70 64 00 03 00 00 00
设置电机的 current3000	14 00 00 A8	06 70 64 00 E0 93 04 00
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
设置电机的 current -3000	14 00 00 A8	06 70 64 00 20 6C FB FF
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
编码器模式		
设置电机为 encoder mode	14 00 00 A8	05 70 64 00 04 00 00 00
设置电机的 encoder 20000	14 00 00 A8	33 70 64 00 80 84 1E 00
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
设置电机的 encoder -20000	14 00 00 A8	33 70 64 00 80 7B E1 FF
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
速度模式 PID 设置		
设置电机为 speed mode	14 00 00 A8	05 70 64 00 01 00 00 00
速度模式的 Kp X100000 Int 15	14 00 00 A8	20 70 64 00 00 6A 18 00
速度模式的 Ki X10000000 Int 0.02	14 00 00 A8	21 70 64 00 40 0D 03 00
速度模式的 Kd X100000 Int 500	14 00 00 A8	22 70 64 00 80 F0 FA 02
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00
设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
位置模式 PID 设置		
设置电机为 position mode	14 00 00 A8	05 70 64 00 02 00 00 00
位置模式的的 Kp X100000 Int 15	14 00 00 A8	23 70 64 00 00 6A 18 00
位置模式的的 Ki X10000000 Int 0.02	14 00 00 A8	24 70 64 00 40 0D 03 00
位置模式的的 Kd X100000 Int 500	14 00 00 A8	25 70 64 00 80 F0 FA 02
设置电机启动	14 00 00 A8	04 70 64 00 01 00 00 00

设置电机关闭	14 00 00 A8	04 70 64 00 00 00 00 00
其他设置		
写入大于 1 的数，保存参数到 flash	14 00 00 A8	02 70 64 00 02 00 00 00
写入大于 1 的数，解除电机堵转保护	14 00 00 A8	03 70 64 00 02 00 00 00
设置 RGB 自定义模式	14 00 00 A8	50 70 64 00 01 00 00 00
设置 RGB 亮度	14 00 00 A8	52 70 64 00 32 00 00 00
设置 RGB 颜色为白色	14 00 00 A8	51 70 64 00 FF FF FF 00
设置 RGB 颜色为紫色	14 00 00 A8	51 70 64 00 80 00 80 00
设置 RGB 颜色为灰色	14 00 00 A8	51 70 64 00 80 80 80 00
设置 RGB 颜色为蓝色	14 00 00 A8	51 70 64 00 FF 00 00 00

- 响应案例：

参考 2.2 电机反馈数据

3.3 I2C Read Raw

- 功能说明：通过 CAN 接口读取 I2C 设备的原始数据。此指令用于从 I2C 设备读取未经处理的原始数据。
- 指令代码： 15H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	15H	bit8~15：用来标识主机 CAN ID	目标电机 CAN ID	byte0: I2C Address byte1：读取的字节数，最大 8

- 发送案例：

注: 主电机 CAN ID 为 A8H,从电机 I2C ADDR 为 64H，因 bit23: is stop bit 需设置为 1，故此为 E4 。

发送写入数据长度标识为 01+寄存器地址的 I2C Write Raw 起始帧用于标识这是一次读取操作。

功能	29bit 数据帧 ID	数据帧
开始通信，并指定 I2C 从机地址 64H 和寄存器 00H	16 E4 00 A8	01 00 00 00 00 00 00 00
从 64H 地址读取 6 个字节	15 00 00 A8	64 06 00 00 00 00 00 00

- 响应案例：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	15H	bit8~15：用来标识主机 CAN ID bit16~23：is read success（0：failed， 1：success）	电机 CAN ID	读取的数据

3.4 I2C Write Raw

- 功能说明：通过 CAN 接口向 I2C 设备写入原始数据。用于将原始控制数据发送到 I2C 设备。
- 指令代码： 16H
- 指令包格式：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	16H	bit8~15：用来标识主机 CAN ID bit16~22：I2C Address bit23：is stop bit	目标电机 CAN ID	byte0: 写入的字节数，最大 7 byte1-byte7：写入的数据

- 发送案例：

注: 主电机 CAN ID 为 A8H,从电机 I2C ADDR 为 64H，因 bit23: is stop bit 需设置为 1，故此为 E4 。

写入寄存器情况的数据帧格式：写入长度 寄存器地址 数据 1-6

功能	29bit 数据帧 ID	数据帧
向 I2C 地址 64H 写入 3 个 byte	16 E4 00 A8	03 00 00 01 00 00 00 00

- 响应案例：

数据域	29bit 数据帧 ID			8byte 数据区
长度	bit24~28	bit8~23	bit0~7	byte0~byte7
描述	16H	bit8~15：用来标识主机 CAN ID bit16~23：is write success（0：failed， 1：success）	电机 CAN ID	0

4、 附录 1

4.1 可读写单个参数列表

表 1 可读写单个参数列表

参数 index(HEX)	参数名称	描述	类型	字节数	单位/说明	R/W
7002	save flash	保存参数到 flash	uint8	1	写入大于 1 的数，保存参数到 flash	W
7003	release protection	解除电机堵转保护	uint8	1	写入大于 1 的数，解除电机堵转保护	W
7004	on/off	电机使能/关闭	uint8	1	1: 电机使能, 0: 电机关闭	W/R
7005	run_mode	电机模式	uint8	1	1: Speed Mode, 2: Position Mode, 3: Current Mode, 4: Encoder Mode	W/R
7006	current (mA)	电流 (X100)	int32_t	4	-120000~120000	W/R
700A	speed (rpm)	速度 (X100)	int32_t	4	-2100000000~2100000000	W/R
7016	position (°)	位置 (X100)	int32_t	4	-2100000000~2100000000	W/R
7017	position_max_current	位置模式最大电流 (X100)	int32_t	4	-120000~120000	W/R
7018	speed_max_current	速度模式最大电流 (X100)	int32_t	4	-120000~120000	W/R
7020	speed_kp	速度模式的 Kp (X100000)	uint32_t	4	P 设置值=0.00001*100000	W/R
7021	speed_ki	速度模式的 Ki (X10000000)	uint32_t	4	I 设置值=0.00001*10000000	W/R
7022	speed_kd	速度模式的 Kd (X100000)	uint32_t	4	D 设置值=0.00001*100000	W/R
7023	position_kp	位置模式的 Kp (X100000)	uint32_t	4	P 设置值=0.00001*100000	W/R
7024	position_ki	位置模式的 Ki	uint32_t	4	I 设置值=0.00001*10000000	W/R

		(X10000000)				
7025	position_kd	位置模式的 Kd (X100000)	uint32_t	4	D 设置值=0.00001*100000	W/R
7030	Speed Readback	实时速度 (X100)	int32_t	4	实际速度=Speed Readback/100	R
7031	Position Readback	实时位置 (X100)	int32_t	4	实际位置=Position Readback/100	R
7032	Current Readback	实时电流 (X100)	int32_t	4	实际电流=Current Readback/100	R
7033	Encoder Counter	编码器值	int32_t	4	仅在编码器模式下可用	W/R
7034	VIN	输入电压 (X100)	int32_t	4	实际电压=VIN X100/100	R
7035	Temp	SOC 温度	int32_t	4		R
7050	rgb mode	RGB 模式	uint8	1	0: 系统默认, 1: 用户自定义	W/R
7051	rgb color	RGB 颜色	uint32_t	4	rgb color = RGB-B + RGB-G * 256 + RGB-R * 65536	W/R
7052	rgb brightness	RGB 亮度	uint8_t	1	0-100	W/R