

美思固
MSSD-2EC 系列无刷电机驱动器
RS485 通信协议及寄存器

说明书

驱控一体，省钱！省心！



目录

1. 概述.....	1
2. 通讯协议.....	1
2.1 通讯参数.....	1
2.2 MODBUS-RTU 帧格式.....	1
2.2.1 0x03 读保持寄存器.....	1
2.2.2 0x06 写单个寄存器.....	2
2.2.3 0x10 写多个寄存器值.....	2
2.2.4 错误异常码.....	3
3. 标准寄存器定义.....	3
3.1 设备描述信息寄存器.....	3
3.2 实时状态寄存器.....	4
3.3 电机控制寄存器.....	6
3.4 电机参数配置寄存器.....	7
3.5 系统参数配置寄存器.....	9
3.6 应用参数 1 配置寄存器.....	10
3.7 应用参数 2 配置寄存器.....	10
3.8 通信参数寄存器.....	10
3.9 PID 参数寄存器.....	12
3.10 上报参数配置寄存器.....	14
4. 通讯示例指令.....	15
4.1 电机启动.....	15
4.2 电机停止.....	15
4.3 读取信息.....	16
4.4 参数设置.....	17
5. 上报功能说明.....	17
5.1 CAN 上报.....	17
5.2 TTL 上报.....	18
6. 附录.....	19
6.1 使用 WINDOWS 自带的计算器进行十进制与十六进制转换.....	19
6.2 CRC16 的计算.....	22
7. 免责声明.....	25



MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

修订历史:

更新日期	修订作者	版本	主要修订摘要数
20260709	Guo	V1.0	初版
20260709	Guo	V1.1	上报功能详细讲解

MSSD-2EC 无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

1. 概述

本说明适用于 **MSSD-2EC 系列** 无刷电机驱动器。

2. 通讯协议

本驱动器使用 MODBUS-RTU(国标 GB/T19582-2008)通讯协议进行通讯，支持一台主站控制多个从站（即驱动器），支持最多 255 个从站，支持通讯广播（从站地址为 0 视为广播帧），主站可以是单片机、PLC 或 PC 机等。

2.1 通讯参数

默认通讯参数：

- 波特率固定为 9600bps。
- 数据位为 8 位。
- 校验为“无+1 停止位”。
- 从站地址固定为 1。

用户自定义通讯参数：

- 波特率可配置范围 9600-115200bps。
- 数据位固定为 8 位。
- 校验方式可配置为：“偶+1 停止位”、“奇+1 停止位”和“无+2 停止位”。
- 从站地址可设置范围 1~255。

当数据帧通信中断超过 4 个字节的打包时间判定为帧接收超时。当发生帧超时的时候，之前接收到的数据将被视为无效。

2.2 MODBUS-RTU 帧格式

本驱动器支持 MODBUS 的 0x03(读保持寄存器)、0x06(写单个寄存器)和 0x10(写多个寄存器)功能码。

2.2.1 0x03 读保持寄存器

主站发送：

字节	1	2	3	4	5	6	7	8
内容	ADR	0x03	起始寄存器 高字节	起始寄存器 低字节	寄存器数 高字节	寄存器数 低字节	CRC 低字节	CRC 高字节

第 1 字节 ADR： 从站地址码（=001~254）。

第 2 字节 0x03： 读寄存器值功能码。

第 3、4 字节： 要读的寄存器开始地址。

第 5、6 字节： 要读的寄存器数量。

第 7、8 字节： 从字节 1 到 6 的 CRC16 校验码 。

从站回送：

字节	1	2	3	4、5	6、7	...	M-1、M	M+1	M+2
----	---	---	---	-----	-----	-----	-------	-----	-----

内容	ADR	0x03	字节总数	寄存器数据 1	寄存器数据 2	...	寄存器数据 M	CRC 低字节	CRC 高字节
----	-----	------	------	---------	---------	-----	---------	---------	---------

第 1 字节 ADR: 从站地址码 (=001~254)。

第 2 字节 0x03: 返回读功能码。

第 3 字节: 从 4 到 M (包括 4 及 M) 的字节总数。

第 4 到 M 字节: 寄存器数据。

第 M+1、M+2 字节: 从字节 1 到 M 的 CRC16 校验码。

当从站接收错误时, 从站回送:

字节	1	2	3	4	5
内容	ADR	0x83	异常码	CRC 低字节	CRC 高字节

第 1 字节 ADR: 从站地址码 (=001~254)。

第 2 字节 0x83: 读寄存器值出错。

第 3 字节 异常码: 见表 0.1。

第 4、5 字节: 从字节 1 到 3 的 CRC16 校验码。

2.2.2 0x06 写单个寄存器

主站发送:

字节	1	2	3	4	5	6	7	8
内容	ADR	0x06	寄存器高字节地址	寄存器低字节地址	数据高字节	数据低字节	CRC 码低字节	CRC 码高字节

当从站接收正确时, 从站回送:

字节	1	2	3	4	5	6	7	8
内容	ADR	0x06	寄存器高字节地址	寄存器低字节地址	数据高字节	数据低字节	CRC 码低字节	CRC 码高字节

当从站接收错误时, 从站回送:

字节	1	2	3	4	5
内容	ADR	0x86	异常码	CRC 低字节	CRC 高字节

第 1 字节 ADR: 从站地址码 (=001~254)。

第 2 字节 0x86: 写寄存器值出错功能码。

第 3 字节 异常码: 见表 0.1。

第 4、5 字节: 从字节 1 到 3 的 CRC16 校验码。

2.2.3 0x10 写多个寄存器值

主站发送:

字节	1	2	3	4	5	6	7
内容	ADR	0x10	起始寄存器高字节地址	起始寄存器低字节地址	寄存器数量高字节	寄存器数量低字节	数据字节总数

字节	8,9	10,11	N,N+1	N+2	N+3
内容	寄存器数据 1	寄存器数据 2	寄存器数据 M	CRC 码低字节	CRC 码高字节



MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

当从站接收正确时，从站回送：

字节	1	2	3	4	5	6	7	8
内容	ADR	0x10	寄存器高字节地址	寄存器低字节地址	寄存器数量高字节	寄存器数量低字节	CRC 码低字节	CRC 码高字节

当从站接收错误时，从站回送：

字节	1	2	3	4	5
内容	ADR	0x90	异常码	CRC 低字节	CRC 高字节

第 1 字节 ADR：从站地址码（=001~254）。

第 2 字节 0x90：写寄存器值出错。

第 3 字节 异常码：见表 0.1。

第 4、5 字节：从字节 1 到 3 的 CRC16 校验码。

2.2.4 错误异常码

MODBUS 异常码：如表 0.1 所示

表 0.1 MODBUS 异常码表

异常码	含义
0x01	非法功能码
0x02	非法数据地址
0x03	非法数据值
0x04	从站设备故障
0x05	请求已被确认，但需要较长时间来处理请求
0x06	从设备忙
0x08	存储奇偶性差错
0x0A	不可用的网关
0x0B	网关目标设备响应失败

3. 标准寄存器定义

3.1 设备描述信息寄存器

设备描述信息寄存器如表 0.1 所示。

表 0.1 设备描述信息寄存器

寄存器地址	描述	取值范围	支持功能码	备注
-------	----	------	-------	----

0x0000 (0)	设备标识号		0x03	每个型号都有对应的设备标识号
0x0001 (1)	设备版本		0x03	高字节为主版本号，低字节为副版本号
0x0002 (2)	最大电流		0x03	数值乘以0.01为电流值，单位为A
0x0003 (3)	定制版本号		0x03	
0x0003~0x0007 (3)~(8)	保留		0x03	

3.2 实时状态寄存器

实时状态寄存器如表 0.2 所示。

表 0.2 实时状态寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x0008 (8)	右电机实时相电流	0~65535	0x03	数值乘以 0.01 为电流值，单位为 A
0x0009 (9)	右电机实时速度之高半字	-110000~110000 (1 极对)	0x03	实时转速,单位为 RPM
0x000A (10)	右电机实时速度之低半字			
0x000B (11)	右电机错误状态	0~18	0x03	0 无错误， 1: 堵转停机， 2: 相序未学习 3: 相位异常， 4: 相线过流 5: 母线过流， 6: 电压过高， 7: 电压过低， 8: 相位学习异常， 9: 过载 10: 485 通信中断， 11: CAN 通讯中断 12: 母线电压过高 13: 温度过高， 14: 无感启动失败 15: 相电流异常 16: TTL 通信异常 17: 编码器学习异
0x000C (12)	右电机实时 PWM	0~10000	0x03	数值乘以 0.1%为 PWM 百分比
0x000D (13)	右电路板温度	-200~200	0x03	单位为℃

0x000E (14)	右电机温度 ADC 值	0~4095	0x03	右电机温度 ADC 值（需驱动器有该功能）
0x000F (15)	右电机位移	-2147483648~ 2147483647	0x03	单位为°
0x0010 (16)				
0x0011(17)	右电机反向状态	0、1	0x03	0：无，1：反向
0x0012(18)	左电机实时相电流，	0~65535	0x03	数值乘以 0.01 为电流值，单位为 A
0x0013(19)	左电机实时速度之高半字	-110000~110000 (1 极对)	0x03	实时转速,单位为 RPM
0x0014(20)	左电机实时速度之低半字			
0x0015(21)	左电机错误状态	0~18	0x03	0 无错误， 1：堵转停机， 2：相序未学习 3：相位异常， 4：相线过流 5：母线过流， 6：电压过高， 7：电压过低， 8：相位学习异常， 9：过载 10：485 通信中断， 11：CAN 通讯中断 12：母线电压过高 13：温度过高， 14：无感启动失败 15：相电流异常 16：TTL 通信异常 17：编码器学习异
0x0016 (22)	左电机实时 PWM	0~10000	0x03	数值乘以 0.1%为 PWM 百分比
0x0017 (23)	左电路板温度	-200~200	0x03	单位为℃
0x0018 (24)	左电机温度 ADC 值	0~4095	0x03	左电机温度 ADC 值（需驱动器有该功能）
0x0019 (25)	左电机位移	-2147483648~ 2147483647	0x03	单位为°
0x001A (26)				
0x001B(27)	左电机反向状态	0、1	0x03	0：无，1：反向

0x001C(28)	电源电压	0~1000	0x03	单位为 V
0x001D (29)	实时系统模式	0~6	0x03	0: 独立速度闭环 1: 同源速度闭环 2: 差速闭环 3: 独立位置闭环 (暂无) 4: 同源位置闭环 (暂无) 5: 独立扭矩闭环 (暂无) 6: 同源扭矩闭环 (暂无)
0x001E (30)	控制模式	0x07、0x18、0x10 0x38、0x30	0x03	0-5 位中从低到高的 //第 5 位表示方波 0 或 FOC1, 第 4 位表示无感 0 或有感 1, //第 3 位表示开环 0 或闭环 1, //第 0-2 位表示,相序学习 111, 通讯控制 000, 航模 PWM001, 摇杆 010) (0bXXXXX111)为相序学习, (0b00011000)0x18 为方波有感/闭环/通讯控制, (0b00010000)0x10 为方波有感/开环/通讯控制, (0b00111000)0x38 为 FOC 有感/闭环/通讯控制, (0b00110000)0x30 为 FOC 有感/开环/通讯控制,
0x001F (31)	IN1 电平	0,1	0x03	0: 低电平 1: 高电平
0x0020(32)	IN2 电平	0,1	0x03	0: 低电平 1: 高电平
0x0021(33)	实时传感器	0,1	0x03	0: 霍尔 1: 编码器
0x0022~0x003F (34)~(63)	保留			

3.3 电机控制寄存器

电机控制寄存器如表 0.3 所示。

表 0.3 电机控制寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x0040 (64)	右电机停止	0, 1, 2	0x06	①当为独立速度开环/闭环控制时, 代表右电机停止; ②当为同源速度开环/闭环控制时, 代表右电机左电机停止; ③当为差速闭环控制时, 代表右电机左电机停止; (0: 正常停止 1: 紧急停止 2: 自由停止)



MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

0x0041 (65)	右电机速度的目标参数之高半字	闭环: -最高转速~最高转速	0x06, 0x10	①当为独立速度闭环控制时, 代表右电机速度目标, 则单位为 RPM; ②当为同源速度闭环控制时, 代表右电机和左电机的速度目标, 则单位为 RPM; (正数表示正转, 负数表示反转, 0 正常停止) ③当为差速闭环控制时, 代表直行速度; (正数表示前进, 负数表示后退)
0x0042 (66)	右电机速度的目标参数之低半字			
0x0043 (67)	右电机抱闸控制	0, 1	0x06	0: 断开抱闸 1: 闭合抱闸
0x0044 (68)	左电机停止	0, 1, 2	0x06	①当为独立速度开环/闭环控制时, 代表左电机停止; ②当为同源速度开环/闭环控制时, 无效; ③当为差速闭环控制时, 无效; (0: 正常停止 1: 紧急停止 2: 自由停止)
0x0045 (69)	左电机速度的目标参数之高半字	闭环: -最高转速~最高转速	0x06, 0x10	①当为独立速度闭环控制时, 代表左电机速度目标, 则单位为 RPM; (正数表示正转, 负数表示反转, 0 正常停止) ②当为同源速度闭环控制时, 无效; ③当为差速闭环控制时, 代表转弯速度; (正数表示右转, 负数表示左转)
0x0046 (70)	左电机速度的目标参数之低半字			
0x0047 (71)	左电机抱闸控制	0, 1	0x06	0: 打开抱闸 1: 关闭抱闸
0x0048 (72)	参数恢复出厂命	0, 1	0x06	1: 恢复出厂 其他: 无效
0x0049 (73)	测试控制	0, 1	0x06	0: 取消测试 1: 左电机相序学习 2: 右电机相序学习
0x004A~0x0057 (74)~ (87)	保留			

3.4 电机参数配置寄存器

电机参数配置寄存器如表 0.4 所示。

表 0.4 电机参数配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x0058 (88)	最大输出电流 (单路)	50~6000	0x03 0x06 0x10	数值乘以 0.01 为电流值, 单位为 A
0x0059 (89)	电机正转最大转速之高半字	0~110000 (1 极对)	0x03 0x06 0x10	单位为 RPM
0x005A (90)	电机正转最大转速之低半字			

0x005B (91)	电机反转最大转速之高半字	0~110000 (1 极对)	0x03 0x06 0x10	单位为 RPM
0x005C (92)	电机反转最大转速之低半字			
0x005D (93)	电机最小转速之高半字	0~110000 (1 极对)	0x03 0x06 0x10	单位为 RPM
0x005E (94)	电机最小转速之低半字			
0x005F (95)	闭环加速加速度	10~65535	0x03 0x06 0x10	单位为 RPM/S
0x0060 (96)	闭环减速加速度	10~65535	0x03 0x06 0x10	单位为 RPM/S
0x0061 (97)	保留			
0x0062 (98)	保留			
0x0063 (99)	保留			
0x0064 (100)	自动换向模式	0, 1	0x03 0x06 0x10	0: 正常换向 1: 急停换向
0x0065 (101)	电机霍尔电角度	0, 1	0x03 0x06 0x10	0: 60° 1: 120°
0x0066 (102)	传感器类型	0, 1	0x03 0x06 0x10	0: 霍尔 1: 霍尔加编码器 AB
0x0067 (103)	保留			
0x0068 (104)	右电机极对数	0~65535	0x03 0x06 0x10	定子南北磁极对数, 是磁极的一半
0x0069 (105)	右电机 1、2 相相序数据	高字节为 0~6 低字节为 0~6	0x03 0x06 0x10	低字节为 1 相 高字节为 2 相
0x006A (106)	右电机 3、4 相相序数据	高字节为 0~6 低字节为 0~6	0x03 0x06 0x10	低字节为 3 相 高字节为 4 相
0x006B (107)	右电机 5、6 相相序数据	高字节为 0~6 低字节为 0~6	0x03 0x06 0x10	低字节为 5 相 高字节为 6 相
0x006C (108)	右电机相序学习状态	0,1	0x03 0x06 0x10	0: 未学习 1: 已经学习
0x006D (109)	右电机电子驻车 (锁相) 最大电流	0~最大输出电流	0x03 0x06 0x10	0: 不停机锁相 其他数值: 数值乘以 0.01 为电流值, 单位为 A
0x006E (110)	右电机 FOC 相位补偿补偿角度	-18000~18000	0x03 0x06 0x10	数值为度数*100, 有方向
0x006F (111)	右电机 FOC 超前角度	-18000~18000	0x03 0x06 0x10	数值为度数*100, 有方向

0x0070 (112)	右电机是否反向	0,1	0x03 0x06 0x10	0: 无 1: 反向
0x0071 (113)	左电机极对数	0~65535	0x03 0x06 0x10	定子南北磁极对数，是磁极的一半
0x0072 (114)	左电机 1、2 相相序数据	高字节为 0~6 低字节为 0~6	0x03 0x06 0x10	低字节为 1 相 高字节为 2 相
0x0073 (115)	左电机 3、4 相相序数据	高字节为 0~6 低字节为 0~6	0x03 0x06 0x10	低字节为 3 相 高字节为 4 相
0x0074 (116)	左电机 5、6 相相序数据	高字节为 0~6 低字节为 0~6	0x03 0x06 0x10	低字节为 5 相 高字节为 6 相
0x0075 (117)	左电机相序学习状态	0,1	0x03 0x06 0x10	0: 未学习 1: 已经学习
0x0076 (118)	左电机电子驻车（锁相）最大电流	0~最大输出电流	0x03 0x06 0x10	0: 不停机锁相 其他数值：数值乘以 0.01 为电流值，单位为 A
0x0077 (119)	左电机 FOC 相位补偿补偿角度	-18000~18000	0x03 0x06 0x10	数值为度数*100，有方向
0x0078 (120)	左电机 FOC 超前角度	-18000~18000	0x03 0x06 0x10	数值为度数*100，有方向
0x0079 (121)	左电机是否反向	0,1	0x03 0x06 0x10	0: 无 1: 反向
0x007A (122)	编码器线数设置		0x03 0x06 0x10	
0x007B~0x0083 (123)~ (131)	保留			

3.5 系统参数配置寄存器

系统参数配置寄存器如表 0.5 所示。

表 0.5 系统参数配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x0084 (132)	过压保护电压	8~60	0x03 0x06 0x10	单位为 V
0x0085 (133)	欠压保护电压	8~60	0x03 0x06 0x10	单位为 V
0x0086 (134)	是否进行电压保护	0,1	0x03 0x06 0x10	0: 不保护 1: 保护
0x0087 (135)	系统模式	0~4	0x03 0x06 0x10	0: 保留 1: 独立速度闭环 2: 保留 3: 同源速度闭环

				4: 差速闭环
0x0088 (136)	保留			
0x0089 (137)	FOC 闭环模式	0, 1	0x03 0x06 0x10	0: 单速度闭环控制 1: 速度电流双闭环控制
0x008A (138)	电机差速转弯最大速度	0~110000 (1 极对)	0x03 0x06 0x10	单位为 RPM
0x008B (139)	堵转停机时间	0~66635	0x03 0x06 0x10	单位为 mS
0x008C (140)	相过流关闭电流	100~8000	0x03 0x06 0x10	数值乘以 0.01 为电流值, 单位为 A
0x008D~0x9A (141)~ (155)	保留			

3.6 应用参数 1 配置寄存器

应用参数 1 配置寄存器如表 0.6 所示。

表 0.6 应用参数 1 配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x009B~0xAF (156)~ (175)	保留			

3.7 应用参数 2 配置寄存器

应用参数 2 配置寄存器如表 0.7 应用参数 2 配置寄存器 所示。

表 0.7 应用参数 2 配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x00B0 (176)	右电机抱闸输出类型	0,1,2	0x03 0x06 0x10	0: 不输出, 1: 停机自动 刹车, 2: 通讯控制
0x00B1 (177)	右电机停车刹车设置	0,1	0x03 0x06 0x10	0: 不生效, 1: 停止刹车 生效
0x00B2(178)	左电机抱闸输出类型	0,1,2	0x03 0x06 0x10	0: 不输出, 1: 停机自动 刹车, 2: 通讯控制
0x00B3 (179)	左电机停车刹车设置	0,1	0x03 0x06 0x10	0: 不生效, 1: 停止刹车 生效
0x00B4~0xC7 (180)~ (199)	保留			

3.8 通信参数寄存器



MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

通讯参数配置寄存器如表 0.8 通讯参数配置寄存器 所示。

表 0.8 通讯参数配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x00C8 (200)	485通讯设备地址	0~255	0x03 0x06 0x10	Modbus 从设备地址
0x00C9 (201)	485通讯串口波特率	0~4	0x03 0x06 0x10	0: 9600 1: 19200 2: 38400 3: 57600 4: 115200 单位为 bps
0x00CA (202)	485 通讯串口检验方式	0~3	0x03 0x06 0x10	0: 无+停止位 1: 偶+1 停止位 2: 奇+1 停止位 3: 无+2 停止位
0x00CB (203)	485通讯中断制动时间	0~66635	0x03 0x06 0x10	单位为 S
0x00CC (204)	保留			
0x00CD (205)	CAN通讯节点ID	0~127	0x03 0x06 0x10	CAN 通讯节点 ID
0x00CE (206)	CAN通讯串口波特率	0~13	0x03 0x06 0x10	0: 10kbps 1: 20kbps 2: 40kbps 3: 50kbps 4: 80kbps 5: 100kbps 6: 125kbps 7: 200kbps 8: 250kbps 9: 400kbps 10: 500kbps 11: 600kbps 12: 800kbps 13: 1Mbps 单位为 bps
0x00CF (207)	CAN通讯中断制动时间	0~66635	0x03 0x06 0x10	单位为 S
0x00D0 (208)	保留			

0x00D1 (209)	保留			
0x00D2 (210)	TTL 设备地址	0~255	0x03 0x06 0x10	Modbus 从设备地址
0x00D3 (211)	TTL 通讯串口波特率	0~4	0x03 0x06 0x10	0: 9600 1: 19200 2: 38400 3: 57600 4: 115200 单位为 bps
0x00D4 (212)	TTL 通讯串口检验方式	0~3	0x03 0x06 0x10	0: 无+停止位 1: 偶+1 停止位 2: 奇+1 停止位 3: 无+2 停止位
0x00D5 (213)	TTL 通讯中断制动时间	0~66635	0x03 0x06 0x10	单位为 S
0x00D6~0xDF (214)~ (223)	保留			

3.9 PID 参数寄存器

PID 参数配置寄存器如表 0.9 PID 参数配置寄存器所示。

表 0.9 PID 参数配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x00E0 (224)	速度闭环 P 系数高半字	建议 1~327680	0x03 0x10	数值除以 32768 为实际值
0x00E1 (225)	速度闭环 P 系数低半字			
0x00E2 (226)	速度闭环 I 系数高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00E3 (227)	速度闭环 I 系数低半字			
0x00E4 (228)	速度闭环 D 系数高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00E5 (229)	速度闭环 D 系数低半字			
0x00E6 (230)	D 轴 P 系数高半字	建议 1~327680	0x03 0x10	数值除以 32768 为实际值
0x00E7 (231)	D 轴 P 系数低半字			
0x00E8 (232)	D 轴 I 系数高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00E9 (233)	D 轴 I 系数低半字			
0x00EA (234)	D 轴 D 系数高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值



MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

0x00EB (235)	D 轴 D 系数低半字			
0x00EC (236)	Q 轴 P 系数高半字	建议 1~327680	0x03 0x10	数值除以 32768 为实际值
0x00ED (237)	Q 轴 P 系数低半字			
0x00EE (238)	Q 轴 I 系数高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00EF (239)	Q 轴 I 系数低半字			
0x00F0 (240)	Q 轴 D 系数高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00F1 (241)	Q 轴 D 系数低半字			
0x00F2 (242)	编码器驻车速度闭环 PI 控制之 P 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00F3 (243)	编码器驻车速度闭环 PI 控制之 P 参数之低半字			
0x00F4 (244)	编码器驻车速度闭环 PI 控制之 I 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00F5 (245)	编码器驻车速度闭环 PI 控制之 I 参数之低半字			
0x00F6 (246)	编码器驻车速度闭环 PI 控制之 D 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00F7 (247)	编码器驻车速度闭环 PI 控制之 D 参数之低半字			
0x00F8 (248)	编码器驻车 D 轴 PI 控制之 P 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00F9 (249)	编码器驻车 D 轴 PI 控制之 P 参数之低半字			
0x00FA (250)	编码器驻车 D 轴 I 控制之 I 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00FB (251)	编码器驻车 D 轴 PI 控制之 I 参数之低半字			
0x00FC (252)	编码器驻车 D 轴 PI 控制之 D 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00FD(253)	编码器驻车 D 轴 PI 控制之 D 参数之低半字			
0x00FE (254)	编码器驻车 Q 轴 PI 控制之 P 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x00FF (255)	编码器驻车 Q 轴 PI 控制之 P 参数之低半字			

0x0100 (256)	编码器驻车 Q 轴 I 控制之 I 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x0101 (257)	编码器驻车 Q 轴 PI 控制之 I 参数之低半字			
0x0102 (258)	编码器驻车 Q 轴 PI 控制之 D 参数之高半字	建议 1~326780	0x03 0x10	数值除以 32768 为实际值
0x0103 (259)	编码器驻车 Q 轴 PI 控制之 D 参数之低半字			
0x0104 (260)	霍尔驻车速度闭环 PI 控制之 P 参数之高半字	建议 1~326780	0x03 0x10	数值为实际值
0x0105 (261)	霍尔驻车速度闭环 PI 控制之 P 参数之低半字			
0x0106 (262)	保留			
0x0107 (263)	保留			
0x0108 (264)	保留			
0x0109 (265)	保留			
0x00~0xE7 (226)~ (231)	保留			

3.10 上报参数配置寄存器

上报参数配置寄存器如表 0.7 应用参数 2 配置寄存器 所示。(具体功能详细见 [5. 上报功能说明](#))

表 0.10 上报参数配置寄存器

寄存器地址	描述	取值范围	支持功能码	备注
0x0118 (280)	右电机电流速度状态 CAN 自动上报 ID	1~127	0x03 0x06 0x10	
0x0119 (281)	右电机位置 CAN 自动上报 ID	1~127	0x03 0x06 0x10	
0x011A (282)	右电机自动上报时间	0,10 或以上	0x03 0x06 0x10	单位为 mS, 最小 10ms,0 则不上报。
0x011B (283)	左电机电流速度状态 CAN 自动上报 ID	1~127	0x03 0x06 0x10	
0x011C (284)	左电机位置 CAN 自动上报 ID	1~127	0x03 0x06 0x10	
0x011D (285)	左电机自动上报时间	0,10 或以上	0x03 0x06 0x10	单位为 mS, 最小 10ms,0 则不上报。
0x011E (286)	右电机 TTL 自动上报 ID	1~255	0x03 0x06 0x10	
0x011F (287)	右电机 TTL 自动上报时间	0,10 或以上	0x03 0x06 0x10	单位为 mS, 最小 10ms,0 则不上报
0x0120 (288)	左电机 TTL 自动上报 ID	1~255	0x03 0x06 0x10	



MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

0x0121 (289)	左电机 TTL 自动上报时间	0,10 或以上	0x03 0x06 0x10	单位为 mS, 最小 10ms,0 则不上报
0x0122~0x0128 (289)~ (296)	保留			

4. 通讯示例指令

4.1 电机启动

以独立速度闭环控制模式下，右电机以 1000RPM 转速正转为例：01 10 00 41 00 02 04 00 00 03 E8 D7 67

01: 从机地址，如果一条总线有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

10: 写多寄存器的功能码，目标启动寄存器由高、低半字寄存器合成为 1 个目标值，所以需要用写多寄存器指令。

00 41: 寄存器写值的起始寄存器地址，即设定右电机速度的目标寄存器之高半字地址，详见表 0.3。

00 02: 写入寄存器数量，高、低半字寄存器共 2 个寄存器。

04: 数据字节总数，一个寄存器为 2 个字节，写 2 个寄存器就为 4 个字节。

00 00 03 E8: 十进制的 1000 的 32 位十六进制为 00 00 03 E8，所以分别给第一个寄存器写入 00 00，给第二个寄存器写入 03 E8，即表示正转 1000。

D7 67: CRC 校验码。

以独立速度闭环控制模式下，左电机以 500RPM 转速反转为例：01 10 00 45 00 02 04 FF FF FE 0C 77 E1

01: 从机地址，如果一条总线有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

10: 写多寄存器的功能码，目标启动寄存器由高、低半字寄存器合成为 1 个目标值，所以需要用写多寄存器指令。

00 45: 寄存器写值的起始寄存器地址，即设定左电机速度的目标寄存器之高半字地址，详见表 0.3。

00 02: 写入寄存器数量，高、低半字寄存器共 2 个寄存器。

04: 数据字节总数，一个寄存器为 2 个字节，写 2 个寄存器就为 4 个字节。

FF FF FE 0C: 十进制的-500 的 32 位十六进制为 FF FF FE 0C，所以分别给第一个寄存器写入 FF FF，给第二个寄存器写入 FE 0C，即表示反转 500。

77 E1: CRC 校验码。

4.2 电机停止

以独立速度闭环控制模式下，右电机正常停止为例：01 06 00 40 00 00 88 1E

01: 从机地址，如果一条总线有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

06: 写单寄存器的功能码，控制电机停止仅需给对应停止寄存器写值。

00 40: 寄存器写值的寄存器地址，即右电机停止寄存器地址。

00 00: 给寄存器写入 00 00，即正常停止。

88 1E: CRC 校验码。

4.3 读取信息

以读取右电机实时相电流和转速为例，主站发送：01 03 00 08 00 03 84 09

01: 从机地址，如果一条总线上有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

03: 读多寄存器的功能码。

00 08: 寄存器读取的起始寄存器地址，即右电机实时电流寄存器地址，详见表 0.2。

00 03: 读取寄存器数量，右电机实时电流寄存器和实时速度高、低半字寄存器共 3 个寄存器。

84 09: CRC 校验码。

假设从机回复：01 03 06 01 44 00 00 01 37 90 ED

01: 从机地址。

03: 读多寄存器的功能码。

06: 字节总数。

01 44: 读取到的第一个寄存器的数据，即右电机实时电流寄存器的数据，转换为十进制为 324，再乘单位 0.01A，即右电机的实时相电流为 3.24A。

00 00 01 37: 读取到的第二、三个寄存器的数据，即右电机实时速度高半字和低半字寄存器的数据，两个寄存器合成为一个值后再转换为十进制为 311，即右电机的实时转速为正转 311RPM。

90 ED: CRC 校验码。

以读取左电机实时相电流和转速为例，主站发送：01 03 00 12 00 03 A5 CE

01: 从机地址，如果一条总线上有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

03: 读多寄存器的功能码。

00 12: 寄存器读取的起始寄存器地址，即左电机实时电流寄存器地址，详见表 0.2。

00 03: 读取寄存器数量，左电机实时电流寄存器和实时速度高、低半字寄存器共 3 个寄存器。

A5 CE: CRC 校验码。

假设从机回复：01 03 06 01 25 FF FF FE 9E AC 8F

01: 从机地址。

03: 读多寄存器的功能码。

06: 字节总数。

01 25: 读取到的第一个寄存器的数据，即左电机实时电流寄存器的数据，转换为十进制为 293，再乘单位 0.01A，即左电机的实时相电流为 2.93A。

FF FF FE 9E: 读取到的第二、三个寄存器的数据，即左电机实时速度高半字和低半字寄存器的数据，两个寄存器合成为一个值后再转换为十进制为-354，即左电机的实时转速为反转 354RPM。

AC 8F: CRC 校验码。



4.4 参数设置

以将最大输出电流设置为 25A 为例（可断电保存）：01 10 00 58 00 01 02 09 C4 AC 8B

01: 从机地址，如果一条总线上有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

10: 写多寄存器的功能码。用写多寄存器功能码设置参数断电才能保存，若用写单寄存器功能码设置参数，断电重启后会恢复到设置前的参数。

00 58: 寄存器写值的起始寄存器地址，即最大输出电流寄存器的地址。

00 01: 写入寄存器数量，最大输出电流寄存器仅有一个寄存器。

02: 数据字节总数，一个寄存器为 2 个字节，写 1 个寄存器就为 2 个字节。

09 C4: 给寄存器写入 09 C4，转化为十进制为 2500，再乘上单位 0.01A，即为 25A。

AC 8B: CRC 校验码。

以将最大输出电流设置为 25A 为例（不可断电保存）：01 06 00 58 09 C4 0F DA

01: 从机地址，如果一条总线上有多台从机设备，则需要把每台设备设置为不同地址，驱动器出厂默认地址为 1。

06: 写单寄存器的功能码。用写多寄存器功能码设置参数断电才能保存，若用写单寄存器功能码设置参数，断电重启后会恢复到设置前的参数。

00 58: 寄存器写值的寄存器地址，即最大输出电流寄存器的地址。

09 C4: 给寄存器写入 09 C4，转化为十进制为 2500，再乘上单位 0.01A，即为 25A。

0F DA: CRC 校验码。

5. 上报功能说明

5.1 CAN 上报

① 电流速度状态自动上报

驱动器根据设置的自动上报 ID 和时间间隔，按照以下上报数据格式，自动上报电流速度状态数据。

上报数据格式如下：

CAN 标识	数据字节 0-1	数据字节 2-5	数据字节 6	数据字节 7
速度电流状态上报 ID	实时电流	实时速度	实时状态	无效补 0

② 位置自动上报

驱动器根据设置的自动上报 ID 和时间间隔，按照以下上报数据格式，自动上报电流速度状态数据。

上报数据格式如下：

CAN 标识	数据字节 0-4	数据字节 5-7
位置上报 ID	实时位置	无效补 0

数据采用大端方式：高字节在前，低字节在后。

以右电机电流速度状态自动上报为例， 00 C8 00 00 03 E8 00 00

00 C8: 右电机实时电流，转换为十进制为 200，再乘单位 0.01A，即右电机的实时相电流为 2.00A。表示当前右电机电流为 2.00A

00 00 03 E8: 右电机实时速度，转换为十进制为 1000，表示当前右电机转速为 1000RPM。

00: 右电机实时状态，由表 0.2 中右电机错误状态寄存器可知，0 表示当前右电机无错误。

00: 无效补 0。

以右电机位置上报为例 FF FF D8 F0 00 00 00 00

FF FF D8 F0: 右电机实时位置，转换为十进制为-10000，即右电机当前位置为-10000°。

00 00 00 00: 无效补 0

5.2 TTL 上报

当设置了电机 TTL 自动上报 ID 寄存器，以及电机 TTL 自动上报时间寄存器后，TTL 通讯会按设置的上报时间和上报的 ID 地址周期性的发送电机的电流速度状态以及位置数据。

发送数据格式如下：

第 1 字节：TTL 上报 ID。

第 2 字节 0x03： 返回读功能码。

第 3 字节：字节总数 11 个。

第 4 到 5 字节： 电机实时电流大小。

第 6 到 9 字节： 电机实时转速。

第 10 字节： 电机实时状态。

第 11 到 14 字节：电机实时位置。

第 15 到 16 字节：CRC16 校验码。

以左电机电流速度状态位置 TTL 上报为例 02 03 0B 01 5E 00 00 0B B8 00 00 00 15 B3 C0 0E

02: 左电机 TTL 自动上报地址。

03: 返回读功能码。

0B: 字节总数 11 个。

01 5E: 左电机电流，转换为十进制为 350，再乘单位 0.01A，即左电机的实时相电流为 3.50A。表示当前左电机电流为 3.50A

00 00 0B B8: 左电机速度，转换为十进制为 3000，表示当前左电机转速为 3000RPM。

00: 左电机状态，由表 0.2 中左电机错误状态寄存器可知，0 表示当前左电机无错误。

00 00 15 B3: 左电机位置，转换为十进制为 5555，即右电机当前位置为 5555°。

C0 0E: CRC16 校验码。

6. 附录

6.1 使用 Windows 自带的计算器进行十进制与十六进制转换

➤ 使用 Windows XP 自带计算器进行十进制到十六进制的转换步骤如下：

1) 打开系统自带的计算器，如图 0.1 所示。



图 0.1 Windows XP 自带的计算器

2) 选择“查看”-“科学型”菜单项，计算器界面将切换为如图 0.2 所示。



图 0.2 计算器界面切换科学型后的界面

3) 点击“十进制”按钮，输入要转换为十六进制的十进制数，我们以-100 为例进行说明，通过右边数字按钮首先输入 100，然后按“+/-”按钮输入负号，如图 0.3 所示。



图 0.3 在计算器里输入“-100”

4) 然后点击左边“十六进制”按钮，此时，我们先前输入的十进制数-100 被转换为__int64 类型的整数以十六进制进行显示。如果我们需以 long 型、short 型或 char 型整数的十六进制进行显示，可分别按右边的“双字”、“单字”和“字节”按钮进行显示，显示结果如图 0.4 所示。



图 0.4 “-100”转换为 short 型并以十六进制显示

➤ 使用 Windows 7 自带计算器进行十进制到十六进制的转换步骤如下：

1) 打开系统自带的计算器，如图 0.5 所示。

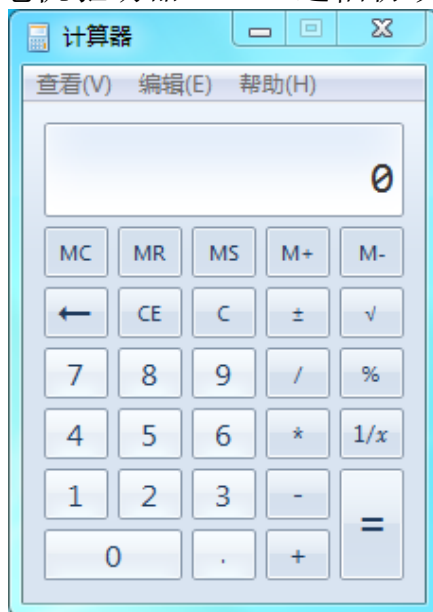


图 0.5 Windows 7 自带的计算器

2) 选择“查看”-“程序员”菜单项，计算器界面将切换如图 0.6 所示。



图 0.6 Windows 7 计算器程序员界面

3) 点击“十进制”按钮，输入要转换为十六进制的十进制数，我们以-100 为例进行说明，通过数字按钮首先输入 100，然后按“±”按钮输入负号，如图 0.7 所示。



图 0.7 Windows 7 计算器程序员界面输入“-100”

4) 然后点击“十六进制”按钮，此时，我们先前输入的十进制数-100 被转换为__int64 类型的整数以十六进制进行显示。如果我们需以 long 型、short 型或 char 型整数的十六进制进行显示，可分别按左下方的“双字”、“字”和“字节”单选按钮进行显示，如图 0.8 所示。



图 0.8 “-100”转换为 short 型并以十六进制显示

6.2 CRC16 的计算

C 语言 CRC 生成函数如[程序 3.1](#)所示。所有的可能的 CRC 值都被预装在两个数组中，当计算报文内容时可以进行简单的索引即可。一个数组含有 16 位 CRC 域的所有 256 个可能的高位字节，另一个数组含有低位字节的值。这种索引访问 CRC 的方式提供了比对报文缓冲区的每个新字符都计算新的 CRC 更快的方法。

注意：此函数内部执行高/低 CRC 字节的交换。此函数返回的是已经经过交换的 CRC 值。也就是说，从该函数返回的 CRC 值可以直接放置于报文用于发送。

函数使用两个参数：

unsigned char *puchMsg; 指向含有用于生成 CRC 的二进制数据报文缓冲区的指针。

unsigned short usDataLen; 报文缓冲区的字节数。

注：以下 CRC16 生成函数程序摘自 <MODBUS over Serial Line Specification and Implementation Guide V1.02> 。

程序 3.1 CRC16 生成函数程序清单

/* 高位字节的 CRC 值*/

```
static unsigned char auchCRCHi[] = {  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,  
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,  
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,  
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,  
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,  
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
    0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,  
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,  
    0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,  
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,  
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
    0x40  
};
```

/* 低位字节的 CRC 值*/

```
static char auchCRCLo[] = {
```

MSSD-2EC 系列无刷电机驱动器 RS485 通信协议及寄存器说明 V1.1

0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40

};

unsigned short CRC16(puchmsg, usDataLen) /* 函数以 unsigned short 类型返回 CRC */

unsigned char *puchMsg, /* 用于计算 CRC 的报文*/

unsigned short usDataLen /* 报文中的字节数*/

{

unsigned char uchCRCHi = 0xFF; /* CRC 的高字节初始化*/

unsigned char uchCRCLo = 0xFF; /* CRC 的低字节初始化*/

unsigned ulIndex ; /* CRC 查询表索引*/

while (usDataLen--) /* 完成整个报文缓冲区*/

{

```
ulIndex = uchCRCLo ^ *puchMsg++; /* 计算 CRC */
```

```
uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex];
```

```
uchCRCHi = auchCRCLo[uIndex];
```

```
}
```

```
return (uchCRCHi << 8 | uchCRCLo);
```

```
}
```

7. 免责声明

本文档只是提供本产品的使用说明。本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其他方式授予任何知识产权许可。且本产品的销售和/或使用我们不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。本产品为商业级产品，本产品并非设计用于医疗、救生或维生等用途。我们可能随时对产品规格及产品描述做出修改，恕不另行通知。