



# 中置电机标定设备通信协议规范

[填入文件编号]

编 制： 周雄 \_\_\_\_\_  
审 核： \_\_\_\_\_  
批 准： \_\_\_\_\_

武汉天腾动力科技有限公司

二〇二三年七月三日



### 修改记录

修改日期	修改人	修改内容	版本
20230703	周雄	初版发布，通过标定设备实现对电机力矩传感器的标定和检验、电机编号信息的写入、订单参数的写入等。	V1.0

天腾动力



# 中置电机标定设备通信协议规范

## 1 适用范围

本规范仅适用于 MOTINOVA 大牙盘中置电机生产标定设备的开发。

## 2 系统组成

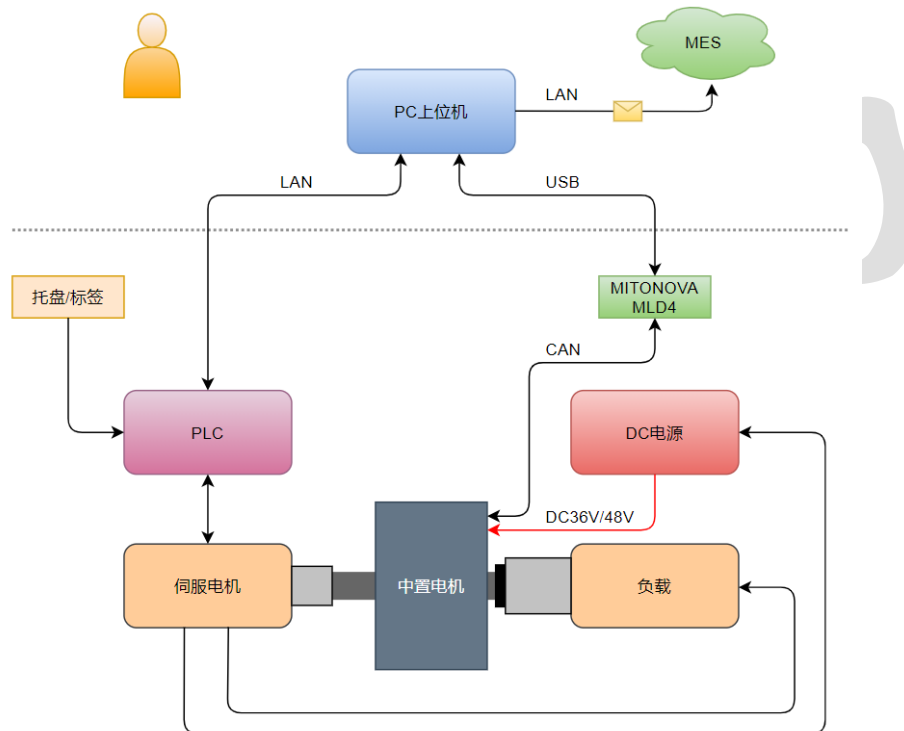


图1 标定设备组成示意图

※本规范仅对 PC 上位机与中置电机之间的通信内容进行定义。

## 3 系统接口

PC 上位机通过 MOTINOVA 联动装置盒 MLD4 转成 CAN 与连接电机，联动装置盒与 PC 通过 USB 连接，通信方式为 UART。

### 3.1 接口类型

- 1) UART，联动装置盒内置 USB 转 UART 模块
- 2) 波特率：115200bps
- 3) 数据位：8
- 4) 停止位：1
- 5) 校验：无



### 3.2 协议格式

表1 UART 协议格式

帧头	CAN ID	帧模式	命令段长度	命令字	数据段	校验位	帧尾
55 AA	ID	读/写/上报	LENGTH	COMMAND	DATA	CRC	F0

其中：

- 1) ID 为 UART 转 CAN 后的报文 ID，占用 2 个字节；
- 2) 帧模式包含读指令 0x11，写指令 0x16，和上报指令 0x0C；
- 3) LENGTH 命令段总长度，占用 1 字节，有效值为 0x02~0xFF；
- 4) COMMAND 为命令字，占用 2 个字节，第 1 字节为命令字序号，第 2 字节为数据段长度；
- 5) DATA 为数据段，长度为 LENGTH - 2；
- 6) CRC 为校验位，占用 4 字节，由帧头开始，CAN\_ID 插入到帧头和帧模式之间，计算到数据段最后一个字节，计算方法见附录 1，计算结果高字节在前，如：CAN\_ID 为 0x0715，数据帧为 55 AA 11 03 22 01 00 CRC1 CRC2 CRC3 CRC4 F0，CRC 计算函数输入数据为 55 AA 07 15 11 03 22 01 00，计算结果依次由高到低写入 CRC1、CRC2、CRC3、CRC4；
- 7) 数据段发送时，采用小端模式；

### 4 工作流程图

测试环节包含准备、测试、结束三个阶段，其中准备阶段需完成电机的装夹、接线、读电机标签、启动按钮等；测试阶段包含力矩传感器 4 段标定、电机铭牌信息写入、订单参数写入等；结束阶段包含数据记录、结果判断、卸载等。

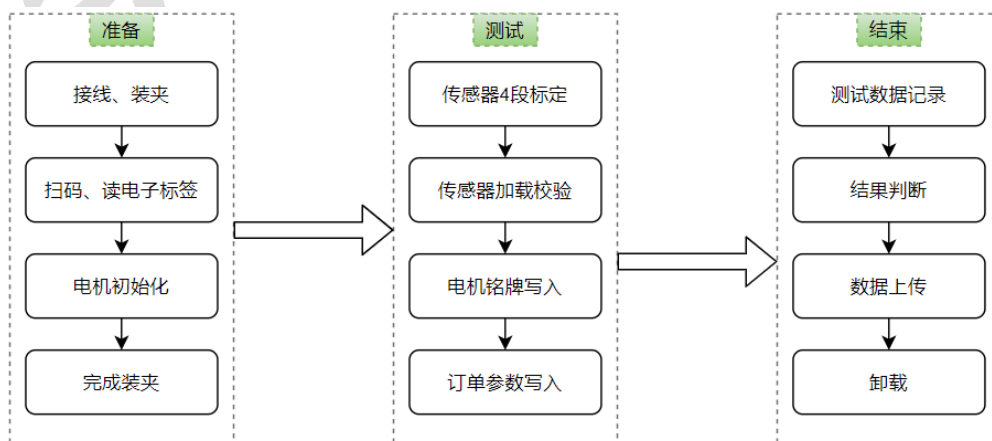


图2 测试简易流程图



上述测试流程中，中途如果存在异常，中断测试并显示警告信息，异常的测试数据也需要进行记录。

## 5 协议内容

### 5.1 测试准备

- 1) 电机安放到测试台后，连接相关线缆，操作启动按钮；
- 2) PLC 读取被测电机的电子标签信息，或用扫码枪获取被测电机的打码信息，通过以太网将电子标签信息发送给上位机；
- 3) 上位机接收电子标签信息后返回**应答指令**；
- 4) PLC 接收应答指令后打开直流电源开关，并发送**测试就绪指令**，等待开始测试。

### 5.2 电机初始化

- 1) 上位机接收到 PLC 的测试就绪指令后，发送指令 [ID:0x7FF, MODE:0x16, CMD:0x2201, DATA:0xF1] 使电机开机；
- 2) 等待 1s 后，上位机发送指令 [ID:0x751, MODE:0x16, CMD:0x2605, DATA: CLEAR] 使电机进入系统初始化，等待电机返回应答指令 [ID:0x715, MODE:0x0C, CMD:0xA903, DATA: ACK] 后提示电机已完成初始化；
- 3) 电机完成初始化后，再次发送指令 [ID:0x7FF, MODE:0x16, CMD:0x2201, DATA:0xF1] 使电机开机，等待 5s 电机完成初始化。

### 5.3 传感器 4 段标定

- 1) 电机完成初始化后，上位机发送**装夹指令**给 PLC，锁住电机的输出轴，装夹到位后 PLC 发送**装夹完成指令**；
- 2) 按照下列流程，依次完成传感器 4 段标定，4 个负载点的扭矩可通过操作界面进行修改：
  - a) 上位机发送**加载扭矩指令**给 PLC，PLC 按照设定的扭矩控制伺服电机，负载稳定后发送**加载完成指令**给上位机；
  - b) 上位机接收到加载完成指令后，发送**加载指令** [0x751, MODE:0x16, CMD:0x4104, DATA: Byte0 和 Byte1 表示加载点序号(1~4), Byte2 和 Byte3 表示加载扭矩(0.1Nm)] 给电机，电机完成标定后返回应答指令 [ID:0x715, MODE:0x0C, CMD:0xA903, DATA: ACK]；
  - c) 重复上面步骤完成 4 段标定。



- 3) 完成标定后需发送指令[0x751, MODE:0x11, CMD:0x4000, DATA:NULL]读取传感器参数，传感器返回参数如下表：

表2 传感器返回参数

ID	帧模式 (MODE)	命令字 (CMD)	功能	数据段 (DATA)	数据大小 (BYTE)	单位
0x715	0x0C	0xB528	传感器 参数	出厂零点	2	
				历史零点 1	2	
				历史零点 2	2	
				历史零点 3	2	
				最新零点	2	
				最大转矩	2	0.1Nm
				第一阶段加 载值	2	0.1Nm
				第一阶段标 定值	2	
				第二阶段加 载值	2	0.1Nm
				第二阶段标 定值	2	
				第三阶段加 载值	2	0.1Nm
				第三阶段标 定值	2	
				第四阶段加 载值	2	0.1Nm
				第四阶段标 定值	2	
踏频传感器 脉冲数	2					
车速传感器 脉冲数	2					
预留	8					

#### 5.4 传感器加载校验

- 1) 电机完成传感器 4 段标定后，上位机根据上述设置的 4 个负载点依次发送**加载扭矩指令**给 PLC，PLC 按照设定的扭矩控制伺服电机，负载稳定后发送**加载完成指令**给上位机；
- 2) 上位机读取电机发送的运行信息中踩踏扭矩值，与加载扭矩是否一致，测量误



差为±2Nm；

电机发送的运行信息内容见下表：

表3 电机运行信息

ID	帧模式 (MODE)	命令字 (CMD)	功能	数据段 (DATA)	数据大小 (BYTE)	单位
0x710	0x0C	0x1020	电机运行信息	车速	2	km/h
				输出转速	2	rpm
				电功率	2	W, 需放大 2 倍显示
				母线电压	2	mV
				母线电流	2	mA
				踏频	1	rpm
				踩踏力矩	1	N.m
				踩踏方向	1	0: 正向 1: 反向 2: 停止
				助力档位	1	0x00: OFF 0x01: ECO 0x02: NORM 0x03: SPORT 0x04: TURBO 0x22: WALK 0x33: SMART
				大灯状态	1	0xF0: 关 0xF1: 开
				剩余电量	1	%
				续航里程	2	km
				力矩 AD	2	
				平均功耗	1	0.01Ah/km
				PCB 温度	1	-40, 1℃
绕组温度	1	-40, 1℃				
MCU 温度	1	-40, 1℃				
预留	1					

### 5.5 电机铭牌写入

在测试准备阶段上位机已获取到 PLC 发送的铭牌信息, 包含电机型号和电机 SN 号, 上位机分别发送写入电机型号指令 [ID: 0x751, MODE: 0x16, CMD: 0x2210, DATA: 铭牌中电机型号, 长度不足 16Bytes 时以 0x2E 填充] 和写入电机 SN 号指令 [ID: 0x751, MODE: 0x16, CMD: 0x2310, DATA: 铭牌中电机 SN 号, 长度不足 16Bytes 时以 0x2E 填



充]完成铭牌写入。

## 5.6 订单参数写入

上位机导入参数设置表，根据协议依次写入电机校验密钥、自定义字符串 1、自定义字符串 2、自定义字符串 3、生产信息、马达信息、整车信息、控制器参数、其它传感器参数、助力参数等，通信协议参考下表：

表4 写入订单参数通信协议

功能	ID	帧模式 (MODE)	命令字 (CMD)	数据段 (DATA)	数据大小 (BYTE)	内容/单位
电机校验密钥	0x751	0x16	0x1108	ASCII, 结束符为 0x2E	8	
自定义字符串 1	0x751	0x16	0x1410	ASCII, 结束符为 0x2E	16	
自定义字符串 2	0x751	0x16	0x1610	ASCII, 结束符为 0x2E	16	
自定义字符串 3	0x751	0x16	0x1810	ASCII, 结束符为 0x2E	16	
生产信息	0x751	0x16	0x2420	生产商	8	ASCII, 0x2E 结束
				生产地	8	ASCII, 0x2E 结束
				生产日期	8	ASCII, 按照 YYYYMMDD
				产品标识 <sup>[1]</sup>	8	ASCII, 0x2E 结束
马达信息	0x751	0x16	0x3B28	存储标志	2	固定为 1
				极对数	2	1-65535
				电阻	2	0.1mOhm
				D 轴电感	2	0.01uH
				Q 轴电感	2	0.01uH
				磁链	2	0.001mWb
				Id 最大值	2	0.01A
				Id 最小值	2	0.01A
				额定转速	2	1rpm
				额定功率	2	1W
				额定电流	2	0.01A
				额定电压	2	0.1V
				惯量	2	$10^{-7} \text{kgm}^2$
				最大转矩	2	0.1Nm
预留	12	0x00				
整车信息	0x751	0x16	0x3D1C	存储标志	2	固定为 1





					轮胎周长	2	1cm
					电控传动比	2	0.1
					助力最大限速	2	1km/h
					推行模式限速	2	1km/h
					前牙盘齿数	2	1T
					后牙盘齿数	2	1T
					助力方案 1 编号	2	每 2 bit 代表 1 个档位的助力曲线，由低到高依次表示 1-5 档
					助力方案 2 编号	2	每 2 bit 代表 1 个档位的助力曲线，由低到高依次表示 1-5 档
					前后灯电压	2	高 8 位： Bit7-Bit4： 尾灯模式 Bit3-Bit0： 尾灯电压： 6:6V, 12:12V 低 8 位： 前灯电压： 6:6V, 12:12V
					周长微调值	2	±10cm, 有符
					启动模式	2	1-柔和, 2-正常, 3-强劲
					开关机控制时间	2	高 8 位： Bit7-Bit4: 开机延时，单位 0.2s，设置 0 采用默认值 Bit3-Bit0: 关机延时，单位 0.2s，设置 0 采用默认值 低 8 位： 自动关机时间，单位 1min，设置 0 为不自动关机



控制器参数	0x751	0x16	0x3F22	预留	2	0
				存储标志	2	固定为 1
				位置传感器 零点 <sup>[2]</sup>	2	1-65535
				位置传感器 当前零点 <sup>[3]</sup>	2	1-65535
				峰值电流	2	0.01A
				过流保护阈 值	2	0.01A
				过压保护阈 值	2	0.1V
				欠压保护阈 值	2	0.1V
				超速保护阈 值	2	1rpm
				过热保护阈 值	2	1℃
				过热保护恢 复阈值	2	1℃
				过热限功率 阈值	2	1℃
其它传感器参 数	0x751	0x16	0x420E	预留	12	0
				存储标志	2	固定为 1
				踏频传感器 脉冲数	2	1-100
				车速传感器 脉冲数	2	1-100
助力参数	0x751	0x16	0x4450	预留	8	0
				存储标志	2	固定为 1
				预留	2	0
				零速启动增 益	2	
				巡航启动增 益	2	
				助力转矩曲 线编号 <sup>[4]</sup>	2	1-15
				助力踏频曲 线编号 <sup>[5]</sup>	2	1-15
				转矩曲线 a	4	有符
				转矩曲线 b	4	有符
				转矩曲线 c	4	有符
转矩曲线 d	4	有符				
踏频曲线 a	4	有符				



				踏频曲线 b	4	有符
				踏频曲线 c	4	有符
				踏频曲线 d	4	有符
				助力启动阈值	2	0.1Nm
				助力停机阈值	2	0.1Nm
				启动电流增长阶梯	2	
				启动踏频脉冲	2	1-100
				转矩滤波踏频脉冲	2	1-100
				待速转速	2	
				待速最大电流	2	
				限速起始值	2	1km/h
				限速停机值	2	1km/h
				踏频占比	2	
				预留	16	0

备注：

[1] 产品标识根据电机产品系列选择, CITY 电机选择 MM\_MC1. , MTB 电机选择 MM\_MT1. , CARGO 电机选择 M\_MG1. ;

[2] 每个位置传感器零点不同, 在初始化时电机检测后自动保存, 写入时需先从电机读出, 修改其它参数后再写入;

[3] 每个位置传感器当前零点不同, 在初始化时电机检测后自动保存, 写入时需先从电机读出, 修改其它参数后再写入;

[4] 每次只能写入一条助力转矩曲线, 该指令需要重复写入 15 次, 每次写入的助力转矩曲线编号不同;

[5] 每次只能写入一条助力踏频曲线, 该指令需要重复写入 5 次, 每次写入的助力踏频曲线编号不同。

## 5.7 测试数据记录

上述每一步测试过程操作的日志需要进行记录保存, 保存信息至少需要包含以下信息:

- 1) 被测试电机的铭牌信息;



- 2) 日志执行的时间，包含年、月、日、时、分、秒；
- 3) 读取的传感器参数，参考表 2；
- 4) 传感器加载校验的加载值和电机测量值。

测试异常时也需要进行数据记录，可在记录文件名后增加 NG 的标识，以便于区分。

## 5.8 结果判断

测试结束前需要对传感器的标定结果进行判断，主要从下面几个方面进行判断：

表5 传感器标定合格判断依据

检验项目	判断方法	判断依据
零点值	读取传感器参数中出厂零点	可设置范围区间
灵敏度 1	(标定 1 - 零点) / 加载 1	可设置范围区间
灵敏度 2	(标定 2 - 标定 1) / (加载 2-加载 1)	可设置范围区间
灵敏度 3	(标定 3 - 标定 2) / (加载 3-加载 2)	可设置范围区间
灵敏度 4	(标定 4 - 标定 3) / (加载 4-加载 3)	可设置范围区间
加载 1 校验	加载 1 测量 - 加载 1	$\pm 2\text{Nm}$
加载 2 校验	加载 2 测量 - 加载 2	$\pm 2\text{Nm}$
加载 3 校验	加载 3 测量 - 加载 3	$\pm 2\text{Nm}$
加载 4 校验	加载 4 测量 - 加载 4	$\pm 2\text{Nm}$
量程	读取标定 4 数据	$\leq 3800$

## 5.9 数据上传

每台电机的测试数据都需要上传服务器，文件名需包含电机铭牌信息和测试日期及时间等，对于异常的电机测试数据在文件名末尾添加 NG 字符。。

## 5.10 卸载

完成数据上传后，上位机发送**测试结束指令**给 PLC，PLC 关闭电源开关，松开夹具，结束测试，等待下一台电机测试开始。



## 6 附录 1：CRC32 计算方法

### 6.1 CRC32 计算多项式表

```
uint32_t Crc32Table[ 256 ] =
```

```
{
```

```
    0x00000000, 0x04C11DB7, 0x09823B6E, 0x0D4326D9, 0x130476DC, 0x17C56B6B,  
    0x1A864DB2, 0x1E475005, 0x2608EDB8, 0x22C9F00F, 0x2F8AD6D6, 0x2B4BCB61,  
    0x350C9B64, 0x31CD86D3, 0x3C8EA00A, 0x384FBDBD, 0x4C11DB70, 0x48D0C6C7,  
    0x4593E01E, 0x4152FDA9, 0x5F15ADAC, 0x5BD4B01B, 0x569796C2, 0x52568B75,  
    0x6A1936C8, 0x6ED82B7F, 0x639B0DA6, 0x675A1011, 0x791D4014, 0x7DDC5DA3,  
    0x709F7B7A, 0x745E66CD, 0x9823B6E0, 0x9CE2AB57, 0x91A18D8E, 0x95609039,  
    0x8B27C03C, 0x8FE6DD8B, 0x82A5FB52, 0x8664E6E5, 0xBE2B5B58, 0xBAEA46EF,  
    0xB7A96036, 0xB3687D81, 0xAD2F2D84, 0xA9EE3033, 0xA4AD16EA, 0xA06C0B5D,  
    0xD4326D90, 0xD0F37027, 0xDDB056FE, 0xD9714B49, 0xC7361B4C, 0xC3F706FB,  
    0xCEB42022, 0xCA753D95, 0xF23A8028, 0xF6FB9D9F, 0xFBB8BB46, 0xFF79A6F1,  
    0xE13EF6F4, 0xE5FFEB43, 0xE8BCCD9A, 0xEC7DD02D, 0x34867077, 0x30476DC0,  
    0x3D044B19, 0x39C556AE, 0x278206AB, 0x23431B1C, 0x2E003DC5, 0x2AC12072,  
    0x128E9DCF, 0x164F8078, 0x1B0CA6A1, 0x1FCDBB16, 0x018AEB13, 0x054BF6A4,  
    0x0808D07D, 0x0CC9CDCA, 0x7897AB07, 0x7C56B6B0, 0x71159069, 0x75D48DDE,  
    0x6B93DDDB, 0x6F52C06C, 0x6211E6B5, 0x66D0FB02, 0x5E9F46BF, 0x5A5E5B08,  
    0x571D7DD1, 0x53DC6066, 0x4D9B3063, 0x495A2DD4, 0x44190B0D, 0x40D816BA,  
    0xACA5C697, 0xA864DB20, 0xA527FDF9, 0xA1E6E04E, 0xBFA1B04B, 0xBB60ADFC,  
    0xB6238B25, 0xB2E29692, 0x8AAD2B2F, 0x8E6C3698, 0x832F1041, 0x87EE0DF6,  
    0x99A95DF3, 0x9D684044, 0x902B669D, 0x94EA7B2A, 0xE0B41DE7, 0xE4750050,  
    0xE9362689, 0xEDF73B3E, 0xF3B06B3B, 0xF771768C, 0xFA325055, 0xFEF34DE2,  
    0xC6BCF05F, 0xC27DEDE8, 0xCF3ECB31, 0xCBFFD686, 0xD5B88683, 0xD1799B34,  
    0xDC3ABDED, 0xD8FBA05A, 0x690CE0EE, 0x6DCDFD59, 0x608EDB80, 0x644FC637,  
    0x7A089632, 0x7EC98B85, 0x738AAD5C, 0x774BB0EB, 0x4F040D56, 0x4BC510E1,  
    0x46863638, 0x42472B8F, 0x5C007B8A, 0x58C1663D, 0x558240E4, 0x51435D53,  
    0x251D3B9E, 0x21DC2629, 0x2C9F00F0, 0x285E1D47, 0x36194D42, 0x32D850F5,  
    0x3F9B762C, 0x3B5A6B9B, 0x0315D626, 0x07D4CB91, 0x0A97ED48, 0x0E56F0FF,  
    0x1011A0FA, 0x14D0BD4D, 0x19939B94, 0x1D528623, 0xF12F560E, 0xF5EE4BB9,  
    0xF8AD6D60, 0xFC6C70D7, 0xE22B20D2, 0xE6EA3D65, 0xEBA91BBC, 0xEF68060B,  
    0xD727BBB6, 0xD3E6A601, 0xDEA580D8, 0xDA649D6F, 0xC423CD6A, 0xC0E2D0DD,  
    0xCDA1F604, 0xC960EBB3, 0xBD3E8D7E, 0xB9FF90C9, 0xB4BCB610, 0xB07DABA7,  
    0xAE3AFBA2, 0xAABFE615, 0xA7B8COCC, 0xA379DD7B, 0x9B3660C6, 0x9FF77D71,  
    0x92B45BA8, 0x9675461F, 0x8832161A, 0x8CF30BAD, 0x81B02D74, 0x857130C3,  
    0x5D8A9099, 0x594B8D2E, 0x5408ABF7, 0x50C9B640, 0x4E8EE645, 0x4A4FFBF2,
```



```
0x470CDD2B, 0x43CDC09C, 0x7B827D21, 0x7F436096, 0x7200464F, 0x76C15BF8,  
0x68860BFD, 0x6C47164A, 0x61043093, 0x65C52D24, 0x119B4BE9, 0x155A565E,  
0x18197087, 0x1CD86D30, 0x029F3D35, 0x065E2082, 0x0B1D065B, 0x0FDC1BEC,  
0x3793A651, 0x3352BBE6, 0x3E119D3F, 0x3AD08088, 0x2497D08D, 0x2056CD3A,  
0x2D15EBE3, 0x29D4F654, 0xC5A92679, 0xC1683BCE, 0xCC2B1D17, 0xC8EA00A0,  
0xD6AD50A5, 0xD26C4D12, 0xDF2F6BCB, 0xDBEE767C, 0xE3A1CBC1, 0xE760D676,  
0xEA23F0AF, 0xEEE2ED18, 0xFOA5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4,  
0x89B8FD09, 0x8D79E0BE, 0x803AC667, 0x84FBDBD0, 0x9ABC8BD5, 0x9E7D9662,  
0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,  
0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4
```

```
};
```

## 6.2 CRC32 计算方法

```
uint32_t CRC32_Calculate( uint8_t *pData, uint16_t Length )  
{  
    uint32_t nReg;  
    uint32_t nTemp = 0;  
    uint16_t i, n;  
  
    nReg = 0xFFFFFFFF;  
    for ( n = 0; n < Length; n++ )  
    {  
        nReg ^= (uint32_t) pData[ n ];  
        for ( i = 0; i < 4; i++ )  
        {  
            nTemp = Crc32Table[ ( uint8_t )( ( nReg >> 24 ) & 0xFF ) ];  
            nReg <<= 8;  
            nReg ^= nTemp;  
        }  
    }  
    return nReg;  
}
```