



测工标定一体机通信协议规范

[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

二〇二一年十一月十日



修改记录

修改日期	修改人	修改内容	版本
20190420	周雄	1、第一版	V1.0
20190422	周雄	1、修改通信接口及设置	V1.1
20200113	周雄	1、3.1.2 中，写入电机的参数预留字节增加部分内容； 2、3.1.4 中，查询力矩传感器零偏数值由读取一组数据更改为依次读取 3 组，且读取指令和电机返回数据指令作调整； 3、3.1.5 中，标定校正系数由计算 1 个标定系数更改为计算 3 个标定系数，并依次写入电机，且写入指令作调整； 4、3.2 测工流程之前读取力矩传感器零偏数值、力矩传感器标定系数更改为依次读取 3 个数值，且读取的指令有调整，读取启动值时发送的指令有作调整。	V2.0
20200601	周雄	1、修改 3.1.4 查询力矩传感器零点值的相关协议。	V2.1
20210707	周雄	1、删除标定前的系统清除、查询和写入参数等环节； 2、合并单传感器和三传感器的两点标定环节，实际使用时二选一； 3、新增传感器多点修正环节，实际使用时可选； 4、删除推行助力模式电机特性测试。	V3.0
20210909	周雄	1、读取电机型号、SN 号、配置参数环节作为独立环节列出，在装夹完成后必须执行； 2、单传感器/三传感器两点标定、传感器多点修正、功能测试三个环节都可在启动界面选择是否执行；	V3.1



武汉天腾动力科技有限公司

文件名称：测工标定一体机通信协议规范

文件编号：[填入文件编号]

版本：V3.2

共 18 页 第 3 页

		3、查询配置参数对应的控制参数 1 和控制参数 2 中增加了部分配置项目，见表 2。	
20211110	周雄	1、单传感器和三传感器的两点标定环节，在步骤二校正传感器系数中，增加关闭电机助力的操作； 2、单传感器和三传感器的两点标定环节，在步骤二校正传感器系数中，增加写入标定系数 1.00 的操作；	V3.2

保密



测工标定一体机通信协议规范

1 通信接口

- 1) CAN 总线
- 2) 波特率 125K、250K、500K、1M，要求提供设置界面可选，默认设置为 125K；
- 3) 帧类型：标准帧
- 4) 帧格式：数据帧

2 协议格式

表 1 CAN 协议格式

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

其中：

- 1) 帧模式包含读指令 0x11，写指令 0x16，和上报指令 0x0C；
- 2) LENGTH 命令段总长度，占用 1 字节，有效值为 0x02 ~ 0xFF；
- 3) 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；
- 4) DATA 为数据段，长度为 LENGTH - 2；
- 5) 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



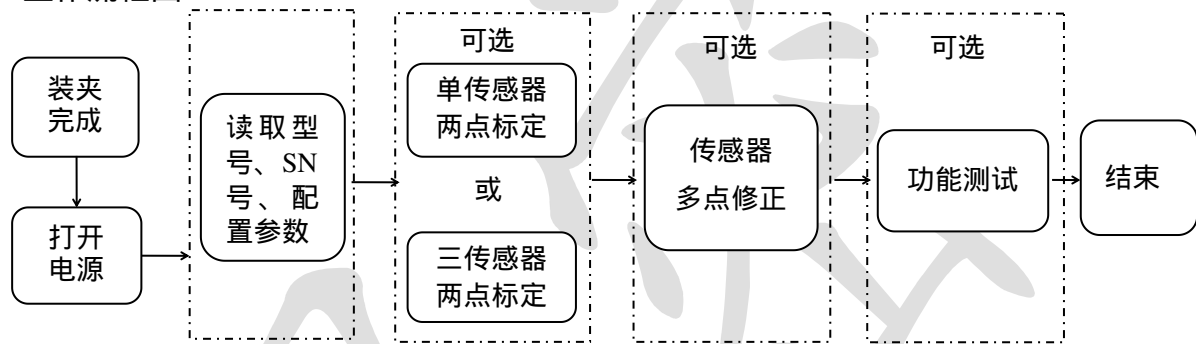
CRC1、CRC2、CRC3、CRC4；

6) 数据段发送时，采用小端模式；

7) 对于长度超过 8Bytes 的帧，采用 8+8+N 的方式，例如长度为 23Bytes 的帧，第一次发送前 8Bytes 数据，第二次发送中间 8Bytes 数据，第三次发送后 7Bytes 数据；

8) 每一包数据发送时需填充对应的 ID。

3 工作流程图



上述流程中，中途如果存在异常，中断测试并显示警告信息。当单传感器和三传感器两点标定都未选择时，无法选择进入多点修正。

4 协议内容

4.1 查询和存储型号、SN、配置参数等

1) 完成电机装夹后，打开电源；

2) 发送指令[ID：0x751，帧模式：0x11，命令字：0x1200，数据：空]查询电机的版本信息，从接收到的电机版本信息[ID：0x710，帧模式：0x0C，命令字：0x1240，数据：依次包含电机型号、电机序列号、硬件版本、软件版本共四条信息，每条信息长度为 16 字符，按照 ASCII 编码，结束符为'.'，无效填充 0x20]中提取电机型号、电机序列号、硬件版本、软件版本等信息，存储为日志文件，文件名称需要包含电机型号、电机序列号以及时间信息，例如 VS7500_M210701A0001A_202107011008.txt。



- 3) 发送指令[ID : 0x751 , 帧模式 : 0x11 , 命令字 : 0x1A00 , 数据 : 空]查询控制参数 1 ,
- 发送指令[ID : 0x751 , 帧模式 : 0x11 , 命令字 : 0x3000 , 数据 : 空]查询控制参数 2 ,
- 发送指令[ID : 0x751 , 帧模式 : 0x11 , 命令字 : 0x1C00 , 数据 : 空]查询马达参数 ,
- 依次将接收到的控制参数 1[ID : 0x715 , 帧模式 : 0x0C , 命令字 : 0xA020 , 数据 : 参考表 2 有关控制参数 1 的定义] , 和控制参数 2[ID : 0x715 , 帧模式 : 0x0C , 命令字 : 0xAD20 , 数据 : 参考表 2 有关控制参数 2 的定义] , 及马达参数[ID : 0x715 , 帧模式 : 0x0C , 命令字 : 0xA110 , 数据 : 参考表 2 有关马达参数的定义] , 将接收到的控制参数 1、控制参数 2、马达参数加入到日志文件。

表 2 电机返回参数指令

ID	帧模式	命令字	功能	数据段	数据含义
0x715	0x0C	0xA020	控制参数 1	指拨模式:1byte	0x55-不支持 0xAA-支持
				启动模式:1byte	1-柔和, 2-正常, 3-强劲
				停机时间:2bytes	1ms
				限速值:1byte	1km/h
				下降速度:1byte	1 ~ 5 个等级
				前飞:1byte	T
				后飞:1byte	T
				限流:1byte	1A
				温度预警:1byte	+40°C
				温度保护:1byte	+40°C
				无码表支持:1byte	0x55-不支持 0xAA-支持
				轮胎周长:1byte	1cm



				电机系列号:1byte	1 ~ 15
				ECO 助力比增益:1byte	0.01 倍 , 50 ~ 150
				ECO 加速度增益:1byte	0.01 倍 , 50 ~ 150
				NOMA 助力比增益:1byte	0.01 倍 , 50 ~ 150
				NOMA 加速度增益:1byte	0.01 倍 , 50 ~ 150
				SPORT 助力比增益:1byte	0.01 倍 , 50 ~ 150
				SPORT 加速度增益:1byte	0.01 倍 , 50 ~ 150
				TURBO 助力比增益:1byte	0.01 倍 , 50 ~ 150
				TURBO 加速度增益:1byte	0.01 倍 , 50 ~ 150
				SMART 助力比增益:1byte	0.01 倍 , 50 ~ 150
				SMART 加速度增益:1byte	0.01 倍 , 50 ~ 150
				速度检测信号个数:1byte	1 ~ 100
				踏频启动信号个数:1byte	1 ~ 100
				速度传感器信号来源:1byte	0x55- 车轮触发 0xAA- 踏频估算 0xEE- 通讯获取



				周长微调值:1byte 1cm , 有符型 低压保护阈 mV 值:2bytes 推行限速:1byte 0.1km/h 推行马达转 1rpm 速:1byte	
0x715	0x0C	0xAD20	控制参数 2	俯仰角零度偏 移:2bytes 0.1°, ±900 横滚角零度偏 移:2bytes 0.1°, ±900 支持角度传感器标 志:1byte 0x55-不支持 0xAA-支持 尾灯工作模 式:1byte 1-模式 1 , 2-模式 2 , 3-模式 3 , 4-模式 4 前灯电压:1byte 0x06:6V 0x0C:12V 0xFF:随电池电压 尾灯电压:1byte 0x06:6V 0x0C:12V 0xFF:随电池电压 预留:24bytes 填充 0x00	
0x715	0x0C	0xA110	马达参数	额定功率:2bytes 1W 额定转速:2bytes 1rpm 定子电阻:2bytes 1mOhm 定子 Lq:2bytes 1mH 定子 Ld:2bytes 1mH 反电动势:2bytes 1mV 额定电压:1bytes 1V	



预留:3bytes

填充 0x00

4.2 单传感器两点标定

4.2.1 步骤一(查询传感器零点值)

- 1) 发送指令[ID : 0x751, 帧模式 : 0x11, 命令字 : 0x2000, 数据 : 空]查询传感器零点值, 电机返回数据[ID : 0x715, 帧模式 : 0x0C, 命令字 : 0xA318, 数据 : (Data24 << 8) + Data23 的计算结果作为当前的力矩传感器零点值];
- 2) 将获得的力矩传感器零点值加入到日志文件, 同时判断该数值是否在允许的范围之内(判据条件可进行设定)。

4.2.2 步骤二(校正传感器系数)

- 1) 发送指令[ID : 0x751, 帧模式 : 0x16, 命令字 : 0x2A01, 数据 : 100]将标定系数 1.00 写入电机, 电机写入成功将返回成功指令[ID : 0x710, 帧模式 : 0x0C, 命令字 : 0xA903, 数据 : ACK], 设备接收到该指令作为判断正常的标志;
- 2) 发送指令[ID : 0x751, 帧模式 : 0x16, 命令字 : 0x1901, 数据 : 0x01]使电机进入配置模式, 此时电机将以 200ms 周期上传电机运行信息, 信息内容见表 3:
- 3) 发送指令[ID : 0x751, 帧模式 : 0x16, 命令字 : 0x2802, 数据 : 0x00 0x00]使电机进入关闭助力, 表 3 的运行信息中助力档位数值为 0x00;

表 3 电机运行信息

ID	帧模式	命令字	功能	数据段	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 0x05:预留 0x22:WALK 0x33:SMART
				大灯开关状态	1	0xF0:关 0xF1:开
				剩余电量	1	%
				续航里程	2	km
				ODO 里程	2	km
				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	-

- 4) 伺服电机输入转速设定为 60rpm，输入力矩设定为 75N.m，等待输入力矩稳定后开始读取电机运行信息中的踩踏力矩值，持续读取 25 组数据，取其平均值作为电机测量值，根据测量值÷75 得到标定系数（精度 0.01）；



- 5) 判断计算的标定系数是否在设定的范围内（判据条件可进行设定）；
- 6) 计算上一步获取到的 25 组数据的最大值、最小值、标准差，分别判断是否在设定的范围内（判断条件可进行设定）；
- 7) 发送指令[ID：0x751，帧模式：0x16，命令字：0x2A01，数据：标定系数 * 100]将标定系数写入电机，同时将标定系数加入到日志文件，电机写入成功将返回成功指令[ID：0x710，帧模式：0x0C，命令字：0xA903，数据：“ACK”]，设备接收到该指令作为判断正常的标志。

4.2.3 步骤三(设定启动值)

- 1) 保持电机在配置模式下，伺服电机输入转速设定为 60rpm，输入力矩设定为 13N.m，等待输入力矩稳定后开始读取电机运行信息中的踩踏力矩值，持续读取 25 组数据，取其平均值作为电机测量值；
- 2) 判断启动值是否在允许的范围之内（判据条件可进行设定）；
- 3) 计算上一步获取到的 25 组数据的最大值、最小值、标准差，分别判断是否在设定的范围内（判断条件可进行设定）；
- 4) 发送指令[ID：0x751，帧模式：0x16，命令字：0x2B02，数据：电机测量值 * 28]将启动值写入电机，同时将该数值加入到日志文件，电机写入成功将返回成功指令[ID：0x715，帧模式：0x0C，命令字：0xA903，数据：“ACK”]，设备接收到该指令作为判断正常的标志。

4.3 三传感器两点标定

4.3.1 步骤一(查询零点值)

- 1) 发送指令[ID：0x751，帧模式：0x11，命令字：0x3201，数据：传感器序号 1、2、3]依次查询 3 个力矩传感器零点偏移数值，电机返回数据[ID：0x715，帧模式：0x0C，



命令字：0xAE1A，数据：传感器序号 1、2、3+零点值，其中第 1 字节 Data1 表示传感器序号， $(Data25 \ll 8) + Data24$ 的计算结果作为当前的力矩传感器零点值]，3 个传感器的零点值需要分步依次读取，如果有读取失败，可提示警告并结束测试；

2) 将获得的 3 个传感器零点值加入到日志文件，并分别判断 3 个数值是否在允许的范围之内（判据条件可进行设定）。

4.3.2 步骤二（校正传感器系数）

- 1) 发送指令[ID：0x751，帧模式：0x16，命令字：0x3402，数据：传感器序号 1、2、3 + 标定系数 100]分别将 3 个标定系数 1.00 写入电机，3 个标定系数需要分步写入，收到电机返回的应答时方可写入下一个标定系数，如果有未收到电机的应答时，可提示警告并结束测试，电机的应答指令为[ID：0x715，帧模式：0x0C，命令字：0xA903，数据："ACK"]；
- 2) 发送指令[ID：0x751，帧模式：0x16，命令字：0x1901，数据：0x01]使电机进入配置模式，此时电机将以 200ms 周期上传电机运行信息，信息内容见表 4：
- 3) 发送指令[ID：0x751，帧模式：0x16，命令字：0x2802，数据：0x00 0x00]使电机进入关闭助力，表 4 的运行信息中助力档位数值为 0x00；

表 4 电机运行信息

ID	帧模式	命令字	功能	数据段	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 0x05:预留 0x22:WALK 0x33:SMART
				大灯开关状态	1	0xF0:关 0xF1:开
				剩余电量	1	%
				续航里程	2	km
				ODO 里程	2	km
				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℃
				单次里程	2	0.1km
				单次时间	2	s
				传感器 1	1	N,m
				传感器 2	1	N,m
				传感器 3	1	N,m



				预留	1	-
--	--	--	--	----	---	---

- 4) 伺服电机输入转速设定为 60rpm ,输入力矩设定为 75N.m 的力 ,等待输入力矩稳定后开始分别读取电机运行信息中传感器 1 (Data29) 、传感器 2 (Data30) 传感器 3 (Data31)的力矩值 ,持续读取 25 组数据 ,分别取其平均值作为电机的 3 个测量值 ,根据测量值÷75 得到 3 个标定系数 (精度 0.01);
- 5) 分别判断 3 个计算的标定系数是否在设定的范围内 (判据条件可进行设定);
- 6) 分别计算上一步获取到的 3 个 25 组数据的最大值、最小值、标准差 ,并分别判断是否在设定的范围内 (判断条件可进行设定);
- 7) 发送指令[ID : 0x751 , 帧模式 : 0x16 , 命令字 : 0x3402 , 数据 : 传感器序号 1、2、3+标定系数 * 100]分别将 3 个标定系数写入电机 ,同时将 3 个标定系数加入到日志文件进行保存 , 3 个标定系数需要分步写入 , 收到电机返回的应答时方可写入下一个标定系数 , 如果有未收到电机的应答时 , 可提示警告并结束测试 , 电机的应答指令为[ID : 0x715 , 帧模式 : 0x0C , 命令字 : 0xA903 , 数据 : "ACK"]。

4.3.3 步骤三(设定启动值)

参考 4.2.3 描述的步骤。

4.4 传感器多点修正

当需要进行传感器多点修正时 , 完成启动值设定后将进行该修正环节 , 否则直接跳转到功能测试环节。修正点的个数通过配置文件进行修改 , 一般提供的范围不超过 15 个点 , 按照配置文件中设置的负载值 , 从小到大的顺序依次完成修正。配置文件参考格式如下 :

表 5 多点修正配置文件参考格式

修正点序号	负载大小
-------	------



1	10Nm
2	20Nm
...	...

- 1) 保持电机在配置模式下，伺服电机输入转速设定为 60rpm，输入力矩设定为当前修正点，等待输入力矩稳定后，发送指令[ID : 0x751 , 帧模式 : 0x16 , 命令字 : 0x3702 , 数据 : Data1 为修正点序号 , Data2 为负载大小]将当前修正点参数写入电机；
- 2) 电机完成修正后，返回当前的修正系数[ID : 0x715 , 帧模式 : 0x0C , 命令字 : 0xB105 , 数据 : Data1 为修正点序号 , Data2-Data3 为 AD 采集值 , Data4 为负载大小 , Data5 为修正系数]，设备接收到该修正点的修正系数后，将接收到的数据加入到日志，并开始修正下一个点，设备可以电机返回的修正系数作为依据，判断修正是否成功；
- 3) 完成配置文件中设定的所有修正点后，结束。

4.5 助力功能测试

助力功能测试采用定点的模式进行测试，测试点的个数通过配置文件进行设置，每个测试点需要判断测量或计算值是否在设定的范围之内。测试配置文件参考格式如下：

表 6 功能测试配置文件参考格式

负载大小	保持时间	采集时间	输入电压	输入电流	电功率	输入转速	输入转矩	输入功率	输出转速	输出转矩	输出功率	效率	助力比
...	...	...	...	...	...	...	...	...	...	...	...	...	...
伺服输入转速			伺服输入最大力矩						/	/	/	/	

上述表中测量值或计算值需要设定最小值和最大值，用于判断当前测试点的功能是否合格。

- 8) 保持电机处于配置模式下，发送指令[ID : 0x751 , 帧模式 : 0x16 , 命令字 : 0x2802 ,



数据：0x04 0x00]使电机进入 TURBO 档位助力模式，电机进入 TURBO 档后，表 3 的

运行信息中助力档位数值为 0x04；

9) 调整负载到当前测试点；

10) 启动伺服电机，输入转速和最大输出力矩按照测试配置文件中的相关设定；

11) 达到设定的采集时间后，分别采集输入电压、输入电流、电功率、输入转速、输入转矩、输入功率、输出转速、输出转矩、输出功率，按照 $(\text{输出功率} - \text{输入功率}) / \text{电功率}$ 计算得到效率，按照 $(\text{输出功率} - \text{输入功率}) / \text{输入功率}$ 计算得到助力比；

12) 判断上述的测量值和计算值是否在测试配置文件的范围之内；

13) 完成测试配置文件中所有点的测试后，结束。



5 附录 1：CRC32 计算方法

5.1 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, 0xE5FFE6B3, 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, 0xAAFBE615, 0xA7B8C0CC, 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, 0xF0A5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4,
0x89B8FD09, 0x8D79E0BE, 0x803AC667, 0x84FBDBD0, 0x9ABC8BD5, 0x9E7D9662,
0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,
0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4

};

5.2 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;
}
```