

MOTINOVA_E-Bike 电机控制器 Bootloader 通信协议

[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

二〇二四年九月二十四日

目录

修订记录.....	3
1 适用范围.....	4
2 软件接口定义.....	4
2.1 接口配置.....	4
2.2 数据帧封装格式.....	4
2.2.1 数据帧格式.....	4
2.2.2 ID 分配.....	5
2.2.3 分包方式.....	5
3 升级协议及流程.....	6
3.1 升级流程图.....	6
3.2 升级协议.....	6
4 附录 1: CRC32 计算方法.....	9
4.1 CRC32 计算多项式表.....	9
4.2 CRC32 计算方法.....	10

修订记录

表1 修订记录

日期	修订人	修订内容	版本
20240924	周雄	基于标准协议整理, 用于描述 Boot 进行固件升级的流程和相关指令。	V1.0

1 适用范围

本文适用于 MOTINOVA E-Bike 电气系统各设备的固件升级, 包含流程图描述与相关的通信指令定义。

2 软件接口定义

2.1 接口配置

CAN 控制器推荐配置如下:

时钟频率: 1000KHz / 2000KHz

SJW: 1

BS1: 6

BS2: 1

波特率: 125Kbps/250Kbps

采样点: 87.5%

2.2 数据帧封装格式

所有传输的数据按照标准帧进行封装, 将传输数据按顺序填入数据帧中。以下对数据帧格式、ID 分配、分包方式进行介绍。

2.2.1 数据帧格式

协议描述了每帧数据内容, 包括帧头、帧模式、命令段长度、命令字、数据段、校验位、帧尾。每帧格式如下:

表2 数据帧格式

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) 帧头固定为 0x55 0xAA, 帧尾固定为 0xF0;
- 2) 帧模式包含读 0x11, 写 0x16, 和上报 0x0C, 任何设备收到写指令时, 需根据数据来源发送通用反馈指令;
- 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 0x0712, 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 12 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。

2.2.2 ID 分配

表3 ID 分配

MC	Target	广播	MC	BMS	OBC	CDL
	CAN ID	0x710		0x712	0x713	0x715
BMS	Target	广播	MC	BMS	OBC	CDL
	CAN ID	0x720	0x721		0x723	0x725
OBC	Target	广播	MC	BMS	OBC	CDL
	CAN ID	0x730	0x731	0x732		0x735
CDL	Target	广播	MC	BMS	OBC	CDL
	CAN ID	0x750	0x751	0x752	0x753	

2.2.3 分包方式

对于长度超过 8bytes 的数据帧, 按照 8+N 的方式分包, 每个数据包填入相同的 ID 号, 如下表所示:

表4 封装方式

包序号	1				N	
内容	ID	Byte1~Byte8	ID	Byte1~Byte8	ID	Byte1~ByteN

3 升级协议及流程

3.1 升级流程图

系统 CAN 总线上所有设备均可通过 CDL 完成升级, 升级流程图如下图所示:

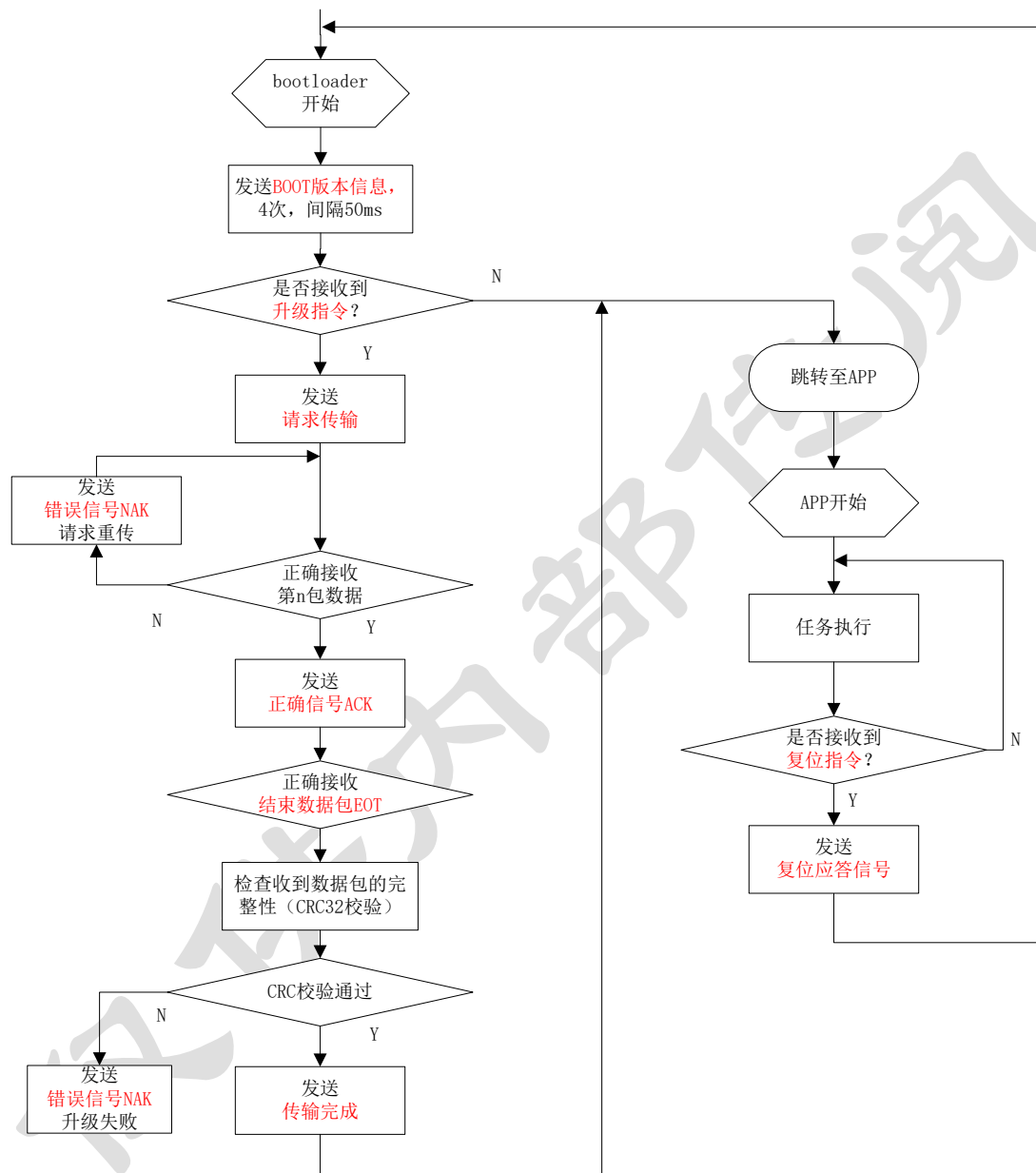


图1 升级流程图

3.2 升级协议

APP 程序中复位指令和复位应答指令:

表5 复位指令

ID	模式	命令字	功能	数据段	备注
发送给 MC					
0x751	0x16	0x2505	复位指令	ASCII 字符串	RESET
MC 应答					
0x715	0x0C	0xA903	通用反馈指令 (返回指令)	ASCII 字符	ACK
发送给 BMS					
0x752	0x16	0x4205	复位指令	ASCII 字符串	RESET
BMS 应答					
0x725	0x0C	0x5503	通用反馈指令 (返回指令)	ASCII 字符	ACK
发送给 OBC					
0x753	0x16	0x6205	复位指令	ASCII 字符串	RESET
OBC 应答					
0x735	0x0C	0x9003	通用反馈指令 (返回指令)	ASCII 字符	ACK

Bootloader 中的协议描述如下:

表6 CDL 发出指令

指令功能	设备名称	ID	模式	命令段长度	命令字	数据段
升级指令	MC	0x751	0x16	0x0D	0xA10B	ASCII 字符: "MC_UPD" + 数据包大小 4 字节 (如 78351B, 内容为: 0F 32 01 00)
	BMS	0x752				ASCII 字符: "BMS_UPD" + 数据包大小 4 字节
	OBC	0x753				ASCII 字符: "OBC_UPD" + 数据包大小 4 字节
数据包格式	MC	0x751	0x16	0x87	0xA385	SOH(01)+序号(1~65535)+总包数(1~65535)+数据(长度 128B, 无效填充 0xFF)
	BMS	0x752				
	OBC	0x753				

结束 数据 包	MC	0x751	0x16	0x03	0xA401	EOT (04)
	BMS	0x752				
	OBC	0x753				

表7 控制器发出指令

指令功能	设备名称	ID	模式	命令段长度	命令字	数据段
BOOT 版本 信息	MC	0x715	0x0C	0x0B	0xC109	ASCII 字符: "MC"+"Vx. x. x"
	BMS	0x725				ASCII 字符: "BMS"+"Vx. x. x"
	OBC	0x735				ASCII 字符: "OBC"+"Vx. x. x"
正确 信号	MC	0x715	0x0C	0x04	0xC202	当前包号, 2byte
	BMS	0x725				
	OBC	0x735				
错误 信号	MC	0x715	0x0C	0x04	0xC302	当前包号, 2byte
	BMS	0x725				
	OBC	0x735				
请求 传输	MC	0x715	0x0C	0x04	0xC402	0x00 0x00
	BMS	0x725				
	OBC	0x735				
传输 完成	MC	0x715	0x0C	0x04	0xC502	0x00 0x00
	BMS	0x725				
	OBC	0x735				

4 附录 1: CRC32 计算方法

4.1 CRC32 计算多项式表

```
1. uint32_t Crc32Table[ 256 ] =
2. {
3.     0x00000000, 0x04C11DB7, 0x09823B6E, 0x0D4326D9, 0x130476DC, 0x17C56B6B,
4.     0x1A864DB2, 0x1E475005, 0x2608EDB8, 0x22C9F00F, 0x2F8AD6D6, 0x2B4BCB61,
5.     0x350C9B64, 0x31CD86D3, 0x3C8EA00A, 0x384FBDBD, 0x4C11DB70, 0x48D0C6C7,
6.     0x4593E01E, 0x4152FDA9, 0x5F15ADAC, 0x5BD4B01B, 0x569796C2, 0x52568B75,
7.     0x6A1936C8, 0x6ED82B7F, 0x639B0DA6, 0x675A1011, 0x791D4014, 0x7DDC5DA3,
8.     0x709F7B7A, 0x745E66CD, 0x9823B6E0, 0x9CE2AB57, 0x91A18D8E, 0x95609039,
9.     0x8B27C03C, 0x8FE6DD8B, 0x82A5FB52, 0x8664E6E5, 0xBE2B5B58, 0xBAEA46EF,
10.    0xB7A96036, 0xB3687D81, 0xAD2F2D84, 0xA9EE3033, 0xA4AD16EA, 0xA06C0B5D,
11.    0xD4326D90, 0xD0F37027, 0xDB056FE, 0xD9714B49, 0xC7361B4C, 0xC3F706FB,
12.    0xCEB42022, 0xCA753D95, 0xF23A8028, 0xF6FB9D9F, 0xFBB8BB46, 0xFF79A6F1,
13.    0xE13EF6F4, 0xE5FFEB43, 0xE8BCCD9A, 0xEC7DD02D, 0x34867077, 0x30476DC0,
14.    0x3D044B19, 0x39C556AE, 0x278206AB, 0x23431B1C, 0x2E003DC5, 0x2AC12072,
15.    0x128E9DCF, 0x164F8078, 0x1B0CA6A1, 0x1FCDDB16, 0x018AEB13, 0x054BF6A4,
16.    0x0808D07D, 0x0CC9CDCA, 0x7897AB07, 0x7C56B6B0, 0x71159069, 0x75D48DDE,
17.    0x6B93DDDB, 0x6F52C06C, 0x6211E6B5, 0x66D0FB02, 0x5E9F46BF, 0x5A5E5B08,
18.    0x571D7DD1, 0x53DC6066, 0x4D9B3063, 0x495A2DD4, 0x44190B0D, 0x40D816BA,
19.    0xACA5C697, 0xA864DB20, 0xA527FDF9, 0xA1E6E04E, 0xBFA1B04B, 0xBB60ADFC,
20.    0xB6238B25, 0xB2E29692, 0x8AAD2B2F, 0x8E6C3698, 0x832F1041, 0x87EE0DF6,
21.    0x99A95DF3, 0x9D684044, 0x902B669D, 0x94EA7B2A, 0xE0B41DE7, 0xE4750050,
22.    0xE9362689, 0xEDF73B3E, 0xF3B06B3B, 0xF771768C, 0xFA325055, 0xFE34DE2,
23.    0xC6BCF05F, 0xC27DEDE8, 0xCF3ECB31, 0xCBFFD686, 0xD5B88683, 0xD1799B34,
24.    0xDC3ABDED, 0xD8FBA05A, 0x690CE0EE, 0x6DCDFD59, 0x608EDB80, 0x644FC637,
25.    0x7A089632, 0x7EC98B85, 0x738AAD5C, 0x774BB0EB, 0x4F040D56, 0x4BC510E1,
26.    0x46863638, 0x42472B8F, 0x5C007B8A, 0x58C1663D, 0x558240E4, 0x51435D53,
27.    0x251D3B9E, 0x21DC2629, 0x2C9F00F0, 0x285E1D47, 0x36194D42, 0x32D850F5,
28.    0x3F9B762C, 0x3B5A6B9B, 0x0315D626, 0x07D4CB91, 0x0A97ED48, 0x0E56F0FF,
29.    0x1011A0FA, 0x14D0BD4D, 0x19939B94, 0x1D528623, 0xF12F560E, 0xF5EE4BB9,
30.    0xF8AD6D60, 0xFC6C70D7, 0xE22B20D2, 0xE6EA3D65, 0xEBA91BBC, 0xEF68060B,
31.    0xD727BBB6, 0xD3E6A601, 0xDEA580D8, 0xDA649D6F, 0xC423CD6A, 0xC0E2D0DD,
32.    0xCDA1F604, 0xC960EBB3, 0xBD3E8D7E, 0xB9FF90C9, 0xB4BCB610, 0xB07DABA7,
33.    0xAE3AFBA2, 0xAAFBE615, 0xA7B8C0CC, 0xA379DD7B, 0x9B3660C6, 0x9FF77D71,
34.    0x92B45BA8, 0x9675461F, 0x8832161A, 0x8CF30BAD, 0x81B02D74, 0x857130C3,
35.    0x5D8A9099, 0x594B8D2E, 0x5408ABF7, 0x50C9B640, 0x4E8EE645, 0x4A4FFBF2,
36.    0x470CDD2B, 0x43CDC09C, 0x7B827D21, 0x7F436096, 0x7200464F, 0x76C15BF8,
37.    0x68860BFD, 0x6C47164A, 0x61043093, 0x65C52D24, 0x119B4BE9, 0x155A565E,
38.    0x18197087, 0x1CD86D30, 0x029F3D35, 0x065E2082, 0x0B1D065B, 0x0FDC1BEC,
39.    0x3793A651, 0x3352BBE6, 0x3E119D3F, 0x3AD08088, 0x2A97D08D, 0x2056CD3A,
40.    0x2D15EBE3, 0x29D4F654, 0xC5A92679, 0xC1683BCE, 0xCC2B1D17, 0xC8EA00A0,
```

```
41. 0xD6AD50A5, 0xD26C4D12, 0xDF2F6BCB, 0xDBEE767C, 0xE3A1CBC1, 0xE760D676,  
42. 0xEA23F0AF, 0xEEE2ED18, 0xF0A5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4,  
43. 0x89B8FD09, 0x8D79E0BE, 0x803AC667, 0x84FBDDBD, 0x9ABC8BD5, 0x9E7D9662,  
44. 0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,  
45. 0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4  
46. };
```

4.2 CRC32 计算方法

```
1. uint32_t CRC32_Calculate( uint8_t *pData, uint16_t Length )  
2. {  
3.     uint32_t nReg;  
4.     uint32_t nTemp = 0;  
5.     uint16_t i, n;  
6.  
7.     nReg = 0xFFFFFFFF;  
8.     for ( n = 0; n < Length; n++ )  
9.     {  
10.        nReg ^= (uint32_t) pData[ n ];  
11.        for ( i = 0; i < 4; i++ )  
12.        {  
13.            nTemp = Crc32Table[ ( uint8_t )( ( nReg >> 24 ) & 0xFF ) ];  
14.            nReg <<= 8;  
15.            nReg ^= nTemp;  
16.        }  
17.    }  
18.    return nReg;  
19. }
```