MOTINOVA E-Bike 电气系统通信协议

(BMS-A)

[填入文件编号]

编　　制：　周雄

审　　核：　肖睿

批　　准：　王瑞瑾

武汉天腾动力科技有限公司

二〇二四年四月十五日

目录

1 总则

系统通信总线采用 CAN2.0A 协议标准帧格式进行数据传输，本文件制定了各设备 ID、报文传输格式、以及数据协议。

2 目的

提高 MOTINOVAE-Bike 电气系统各设备之间数据通信的信号稳定性、格式正确性及信息完整性。

3 适用范围

本文件适用于 MOTINOVA 小牙盘中置系统、MOTINOVA 大牙盘中置系统、MOTINOVA 轮毂电机控制系统的软件开发，以及智能电池保护板、各类仪表的功能开发。

4 修订记录

表1 修订记录

日期	修订人	修订内容	版本
20221109	周雄	增加 BMS 故障日志的定义和读取指令	V4.2
20230610	周雄	运行信息增加循环次数	V4.3
20230728	周雄	增加 OBC 和 HMI 读取 BMS 用户使用记录指令。	V4.4
20231013	周雄	1、BMS 运行信息增加剩余充电时间； 2、BMS 设计信息增加电池包电芯颗数。	V4.5
20240416	周雄	增加产品条码信息读写，满足长条码需求。	V4.6

MOTINOVA_E-Bike 电气系统通信协议(BMS)

1 系统组成

MC：电机控制器 Motor Controller

BMS：电池管理系统 Battery Management System

PBU：按键单元 Push Button Unit

HMI：显示单元 Human Machine Interface

OBC：车载计算机 OnBoard Computer

CDL：通讯适配器 CAN Dongle

APP：用户程序 Application

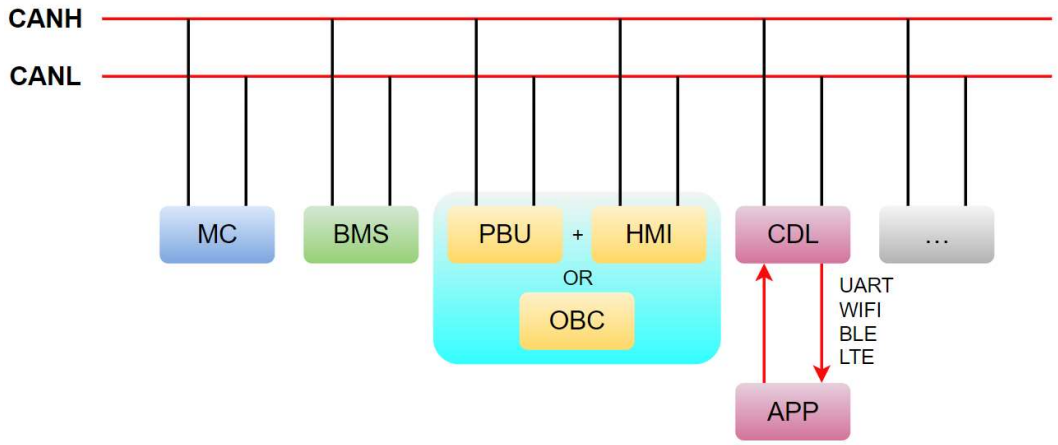

图1 系统通信接口示意图

2 硬件接口

2.1 通信接口参考电路

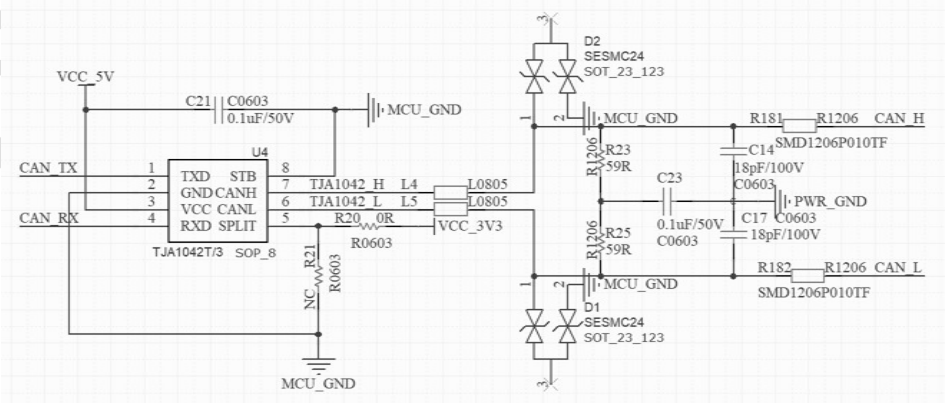

图2 通信接口参考电路

2.2 注意事项

图 2 展示的为各组件 CAN 通信接口参考电路，设计时注意以下几点：

1) ESD 保护电路设计需考虑电源线与通信线发生短路时，不会导致 ESD 保护器件被击穿，电源线最高电压按照 48V 系统最高工作电压 54.6V 测试；
2) 终端匹配电阻设计在 BMS 和 PBU/OBC 内部，其它部件预留；
3) BMS 需考虑通信接口电气隔离设计。

3 通信协议规则

3.1 软件配置

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%

3.2 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。

3.2.1 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。

3.2.2 ID 分配

表3 ID 分配

MC	Target	广播	MC	BMS	PBU/OBC/ECU	HMI	CDL
	CAN ID	0x710		0x712	0x713	0x714	0x715
BMS	Target	广播	MC	BMS	PBU	HMI	CDL
	CAN ID	0x720	0x721		0x723	0x724	0x725
PBU/OBC/ECU	Target	广播	MC	BMS	PBU	HMI	CDL
	CAN ID	0x730	0x731	0x732		0x734	0x735
HMI	Target	广播	MC	BMS	PBU	HMI	CDL
	CAN ID	0x740	0x741	0x742	0x743		0x745
CDL	Target	广播	MC	BMS	PBU	HMI	CDL
	CAN ID	0x750	0x751	0x752	0x753	0x754	

3.2.3 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

4 通信内容

4.1 MC 命令字定义

表5 MC 命令字定义

ID	模式	命令字	功能	数据段	备注
广播指令					
0x710	0x0C	0x1305	关机就绪（返回指令）	ASCII 字符	READY
发送给 BMS					
0x712	0x11	0x3009	BMS 在线检测	ASCII 字符	HANDSHAKE

			（主动发送，收到返回或超时停止）		
0x712	0x11	0x3100	查询 BMS 物理 ID（主动发送，收到返回或超时停止）		
0x712	0x11	0x3200	查询BMS校验码（主动发送，收到返回或超时停止）		
0x712	0x11	0x3300	查询BMS设计信息（主动发送，收到返回或超时停止）		

4.2 BMS 命令字定义

表6 BMS 命令字定义

ID	模式	命令字	功能	数据段	备注
广播指令					
0x720	0x0C	0x1010	电池运行信息（返回指令）	电压:2bytes 平均电流:2bytes 剩余容量:2bytes 满充容量:2bytes 电芯温度:1byte 剩余电量:1byte 运行状态:1byte (按位或输出) SOH:1byte 循环次数:2bytes 剩余充电时间:2bytes	1mV 1mA，有符型，放电为负，充电为正 1mAh 1mAh +40℃ 0～100% 0x00:休眠 0x01:充电器接入 0x02:预留 0x04:预留 0x08:预留 0x10:预留 0x20:预留 0x40:预留 0x80:预留 0～100% 次 1min
0x720	0x0C	0x1120	电芯电压（返回指令）	Cell_1:2bytes ……	1mV ……

				Cell_16:2bytes 不足部分填充 0x00	1mV
0x720	0x0C	0x1204	BMS 故障码（存在故障时 200ms 自动发送，故障消失后停止发送）	高 16 位: 0x0001:充电过压警告 0x0002:放电低压警告 0x0004:充电过流警告 0x0008:放电过流警告 0x0010:充电高温警告 0x0020:充电低温警告 0x0040:放电高温警告 0x0080:放电低温警告 0x0100:MOS 高温警告 低 16 位: 0x0001:二级放电过流保护 0x0002:充电过流保护 0x0004:短路保护 0x0008:过放保护 0x0010:过充保护 0x0020:放电低温保护 0x0040:放电高温保护 0x0080:充电低温保护 0x0100:充电高温保护 0x0200:放电 MOS 故障 0x0400:充电 MOS 故障 0x0800:温度传感器故障 0x1000:预留	按位或输出，0-正常，1-故障

				0x2000:一级过流保护 0x4000:AFE 故障 0x8000:MCU 故障	
0x720	0x0C	0x1308	关机指令（主动发送，收到返回或超时停止）	ASCII 字符	常按开关键 2s,或监测到母线电流小于 30mA 且 CAN 总线空闲持续 10s 后，执行发送 SHUTDOWN，延时 1s 后，关闭放电开关
0x720	0x0C	0x1410	电池设计信息（返回指令）	设计容量:2bytes 设计电压:1byte 电芯型号:8bytes 电芯数量:1byte 预留:4bytes	1mAh 1V ASCII，0x2E 结束，无效填充 0x20 个 填充 0x00
0x720	0x0C	0x1540	电池版本信息（返回指令）	ASCII 字符	排列顺序为: MODEL、SN、HW、FW; 每条信息长度为 16 bytes，结束符为'.',无效填充 0x20; FW 命名格式为 Vxrxrx_YYYYMMDD.
0x720	0x0C	0x160C	电池物理 ID（返回指令）	ID:12bytes	不足位填充 1
0x720	0x0C	0x170C	电池校验码（返回指令）	校验码:12bytes	
0x720	0x0C	0x1810	用户使用记录（返回指令）	电芯最高温:1byte 电芯最低温:1byte 最近充电间隔时间:2bytes 最大充电间隔时间:2bytes 预留:10bytes	+40℃ +40℃ 小时 小时 填充 0x00
发送给 MC					
0x721	0x0C	0x3005	在线检测反馈（返回指令）	ASCII 字符	READY
发送给 CDL					
0x725	0x0C	0x5028	电池 BMS 历史信息（返回指令）	电芯最高温:1byte 电芯最低温:1byte 最大放电电流:2bytes 最大充电电	+40℃ +40℃ 无符型，单位 1mA 无符型，单位 1mA

				流:2bytes 循环次数:2bytes 最近充电间隔时间:2bytes 最大充电间隔时间:2bytes 充电过流保护次数:2bytes 放电过流保护次数:2bytes 过充保护次数:2bytes 过放保护次数:2bytes 短路保护次数:2bytes 充电低温保护次数:2bytes 充电高温保护次数:2bytes 放电低温保护次数:2bytes 放电高温保护次数:2bytes 运行时间:4bytes SOH:1byte 预留:5bytes	 次 小时 小时 次 次 次 次 次 次 次 次 1min 0～100% 填充 0x00
0x725	0x0C	0x5120	生产信息（返回指令）	生产商:8bytes 生产地:8bytes 生产日期:8bytes 预留:8bytes	ASCII，0x2E 结束，无效填充 0x20 ASCII，0x2E 结束，无效填充 0x20 ASCII,YYYYMMDD 填充 0x00
0x725	0x0C	0x5210	自定义可存储字符串 1（返回指令）	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x725	0x0C	0x5310	自定义可存储字符串 2（返回指令）	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x725	0x0C	0x5410	自定义可存储字符串 3（返回指令）	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x725	0x0C	0x5503	通用反馈指令（返回指令）	ASCII 字符	ACK

0x725	0x0C	0x5688	存储器指定起始和结束地址的数据	起始地址:4bytes 结束地址:4bytes 数据:128bytes	结束地址-起始地址＜128 时，无效部分填充 0xFF
0x725	0x0C	0x5720	产品条码	ASCII 字符	无效填充 0x00

4.3 PBU/OBC/ECU 命令字定义

表7 PBU/OBC/ECU 命令字定义

ID	模式	命令字	功能	数据段	备注
广播指令（未注明为 PBU/OBC/ECU 通用）					
0x730	0x0C	0x1405	关机就绪（返回指令）	ASCII 字符	READY
发送给 BMS（未注明为 PBU/OBC/ECU 通用）					
0x732	0x11	0x5000	查询BMS运行信息（主动发送，收到返回或超时停止）		
0x732	0x11	0x5100	OBC/ECU 查询 BMS 版本信息（主动发送，收到返回或超时停止）		
0x732	0x11	0x5200	OBC/ECU 查询 BMS 设计信息（主动发送，收到返回或超时停止）		
0x732	0x11	0x5300	OBC/ECU 查询 BMS 电芯电压（主动发送，收到返回或超时停止）		
0x732	0x11	0x5400	OBC/ECU 查询 BMS 用户使用记录（主动发送，收到返回或超时停止）		

4.4 HMI 命令字定义

表8 HMI 命令字定义

ID	模式	命令字	功能	数据段	备注
广播指令					
0x740	0x0C	0x1305	关机就绪	ASCII 字符	READY

			（返回指令）		
发送给 BMS					
0x742	0x11	0x5000	查询 BMS 版本信息（主动发送，收到返回或超时停止）		
0x742	0x11	0x5100	查询 BMS 设计信息（主动发送，收到返回或超时停止）		
0x742	0x11	0x5200	查询 BMS 电芯电压（主动发送，收到返回或超时停止）		
0x742	0x11	0x5300	HMI 查询 BMS 用户使用记录（主动发送，收到返回或超时停止）		

4.5 CDL 命令字定义

表9 CDL 命令字定义

ID	模式	命令字	功能	数据段	备注
发送给 BMS					
0x752	0x11	0x3000	查询电池物理 ID		
0x752	0x11	0x3100	查询电池校验码		
0x752	0x16	0x320C	写入电池校验码	校验码:12bytes	
0x752	0x11	0x3300	查询电池版本信息		
0x752	0x11	0x3400	查询电池运行信息		
0x752	0x11	0x3500	查询电芯电压		
0x752	0x11	0x3600	查询电池设计信息		
0x752	0x11	0x3700	查询电池生产信息		
0x752	0x11	0x3800	查询电池历史信息		
0x752	0x11	0x3900	查询自定义可存储字符串 1		

0x752	0x16	0x3A10	写入自定义可存储字符串 1	ASCII 字符串	以 0x2E 结束，无效填充 0x20
0x752	0x11	0x3B00	查询自定义可存储字符串 2		
0x752	0x16	0x3C10	写入自定义可存储字符串 2	ASCII 字符串	以 0x2E 结束，无效填充 0x20
0x752	0x11	0x3D00	查询自定义可存储字符串 3		
0x752	0x16	0x3E10	写入自定义可存储字符串 3	ASCII 字符串	以 0x2E 结束，无效填充 0x20
0x752	0x16	0x3F20	写入生产信息（可选，仅供生产商写入）	生产商:8bytes 生产地:8bytes 生产日期:8bytes 预留:8bytes	ASCII，0x2E 结束，无效填充 0x20 ASCII，0x2E 结束，无效填充 0x20 ASCII,YYYYMMDD 填充 0x00
0x752	0x16	0x4010	写入 BMS Mode（可选，仅供生产商写入）	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x752	0x16	0x4110	写入 BMS SN（可选，仅供生产商写入）	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x752	0x16	0x4205	复位指令	ASCII 字符串	RESET
0x752	0x11	0x4308	读取存储器指定地址数据	起始地址:4bytes 结束地址:4bytes	读取数据大小≤128Bytes
0x752	0x11	0x4500	读取产品条码		
0x752	0x16	0x4620	写入产品条码	ASCII 字符	无效填充 0x00

注：CDL 发送的所有指令均按照定时 200ms 发送，其中查询指令收到返回的信息或超时 1s 停止发送，写入指令收到通用反馈指令或超时 1s 停止发送。

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, 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, 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;
}
```

6 附录 2：升级协议及流程

6.1 升级流程图

系统CAN总线上除CDL外所有设备均需通过CDL完成升级，升级流程图如下图所示：

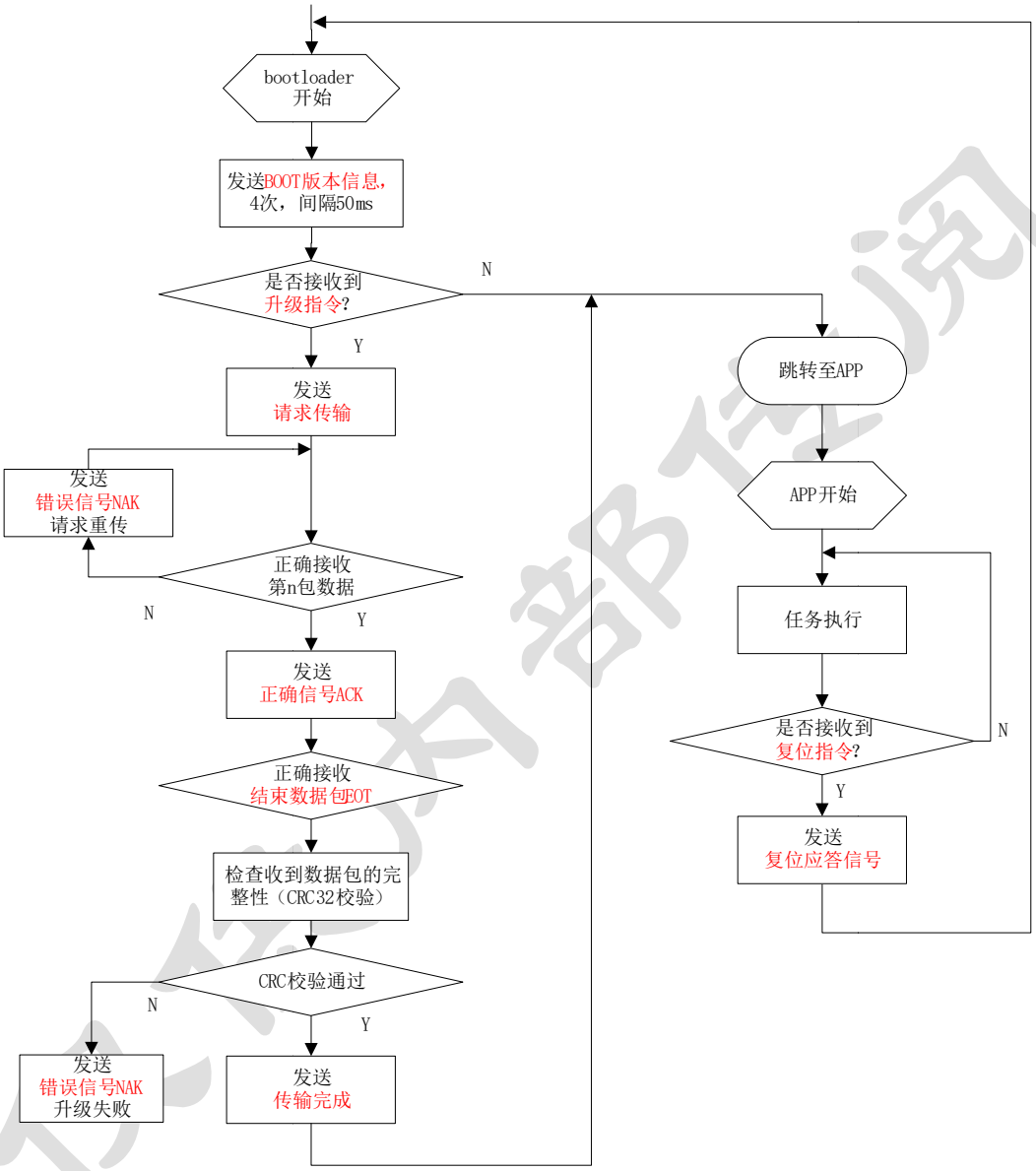

图3 升级流程图

6.2 升级协议

APP程序中复位指令和复位应答指令参考第3章有关CDL发送给各设备的复位指令，以及各设备发送给CDL的通用应答指令。

Bootloader 中的协议描述如下：

表10 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	BMS	0x752	0x16	0x0D	0xA10B	ASCII 字符："BMS_UPD" + 数据包大小 4 字节
数据包格式	BMS	0x752	0x16	0x87	0xA385	SOH(01)+序号(1～65535)+总包数(1～65535)+数据(长度 128B，无效填充 0xFF)
结束数据包	BMS	0x752	0x16	0x03	0xA401	EOT(04)

表11 设备发出指令

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	BMS	0x725	0x0C	0x0B	0xC109	ASCII 字符："BMS"+"Vx.x.x"
正确信号	BMS	0x725	0x0C	0x04	0xC202	当前包号，2byte
错误信号	BMS	0x725	0x0C	0x04	0xC302	当前包号，2byte
请求传输	BMS	0x725	0x0C	0x04	0xC402	0x00 0x00
传输完成	BMS	0x725	0x0C	0x04	0xC502	0x00 0x00

7 附录 3：BMS 故障日志数据结构定义

7.1 目的

1）电池运行过程中出现故障时，BMS 及时记录故障发生时刻的故障码、运行信息等数据；

2）通过上位机导出故障日志，便于对故障进行分析。

7.2 方法

1）BMS 在单片机 Flash 中设计不少于 1024Bytes 用于存储故障日志；

2）每条故障日志大小为 128bytes；

3）所有日志循环存储，即始终存储最近的 n 条故障日志；

4）上位机发送指令给 BMS，BMS 及时返回 Flash 中保存的故障日志。

7.3 协议

上位机采用下列指令读取故障日志：

表12 上位机读取故障日志协议

CAN ID	帧模式	命令字	数据段
0x752	0x11	0x4308	读取存储器指定地址数据：起始地址：4bytes，结束地址:4bytes，读取数据大小≤128Bytes

BMS 接收到上位机读取指令时，按照下列协议返回 Flash 中保存的数据：

表13 BMS 返回故障日志协议

CAN ID	帧模式	命令字	数据段
0x725	0x0C	0x5688	返回存储器指定起始和结束地址的数据：起始地址 4bytes，结束地址:4bytes，数据:128bytes 结束地址-起始地址＜128 时，填充 0xFF

BMS 处理源码示例如下：

```
1.  case 0x4308: //读取存储器制定地址数据
2.  {
3.      do
4.      {
5.          uint32_t DataLength, AddressBegin, AddressEnd;
6.          AddressBegin = (uint32_t)((Data[0] << 24) + (Data[1] << 16) + (Data[2] << 8)
    + (Data[3]));
7.          AddressEnd = (uint32_t)((Data[4] << 24) + (Data[5] << 16) + (Data[6] << 8) +
    (Data[7]));
8.          if(AddressBegin <= AddressEnd)
9.          {
```

```
            DataLength = AddressEnd - AddressBegin+ 1;
            memcpy((uint8_t*)(Data + 8), (uint8_t*)(AddressBegin), DataLength);
            SendData(ID_MC_TO_CDL, MODE_REPORT, (0x5608 + DataLength), (uint8_t*)Dat
    a);
        }
    }while(0);
    break;
}
```

7.4 数据结构定义

```
//故障日志记录信息 128 bytes，1K 空间可以保存最近 8 条故障记录
typedef struct
{
    uint16_t Error_Index;          //故障列表索引，2 bytes，地址偏移 0
    uint16_t NotesInfo1;           //备注信息 1，2 Bytes，地址偏移 2
    uint16_t NotesInfo2;           //备注信息 2，2 Bytes，地址偏移 4
    uint16_t NotesInfo3;           //备注信息 3，2Bytes，地址偏移 6
    uint32_t ErrorCode;            //故障码，4 bytes，地址偏移 8
    uint16_t CellVoltage_1;        //电芯 1 电压，单位 mV，2 Bytes，地址偏移 12
    uint16_t CellVoltage_2;        //电芯 2 电压，单位 mV，2 Bytes，地址偏移 14
    uint16_t CellVoltage_3;        //电芯 3 电压，单位 mV，2 Bytes，地址偏移 16
    uint16_t CellVoltage_4;        //电芯 4 电压，单位 mV，2 Bytes，地址偏移 18
    uint16_t CellVoltage_5;        //电芯 5 电压，单位 mV，2 Bytes，地址偏移 20
    uint16_t CellVoltage_6;        //电芯 6 电压，单位 mV，2 Bytes，地址偏移 22
    uint16_t CellVoltage_7;        //电芯 7 电压，单位 mV，2 Bytes，地址偏移 24
    uint16_t CellVoltage_8;        //电芯 8 电压，单位 mV，2 Bytes，地址偏移 26
    uint16_t CellVoltage_9;        //电芯 9 电压，单位 mV，2 Bytes，地址偏移 28
    uint16_t CellVoltage_10;       //电芯 10 电压，单位 mV，2 Bytes，地址偏移 30
    uint16_t CellVoltage_11;       //电芯 11 电压，单位 mV，2 Bytes，地址偏移 32
    uint16_t CellVoltage_12;       //电芯 12 电压，单位 mV，2 Bytes，地址偏移 34
    uint16_t CellVoltage_13;       //电芯 13 电压，单位 mV，2 Bytes，地址偏移 36
    uint16_t CellVoltage_14;       //电芯 14 电压，单位 mV，2 Bytes，地址偏移 38
    uint16_t CellVoltage_15;       //电芯 15 电压，单位 mV，2 Bytes，地址偏移 40
    uint16_t CellVoltage_16;       //电芯 16 电压，单位 mV，2 Bytes，地址偏移 42
    int32_t ChargeCurrent;         //充电电流，单位 mA，放电为负值，4Bytes，地址偏移 44
    int32_t DisChargeCurrent;      //放电电流，单位 mA，放电为负值，4Bytes，地址偏移 48
    uint32_t FullChargeCapacity;   //满充容量，单位 mAh，4 Bytes，地址偏移 52
    uint32_t RemainCapacity;       //剩余容量，单位 mAh，4 Bytes，地址偏移 56
    uint16_t CycleCount;           //循环次数，2 Bytes，地址偏移 60
    uint8_t Temperature_1;         //温度 1，单位℃，偏移 40，1 Byte，地址偏移 62
    uint8_t Temperature_2;         //温度 2，单位℃，偏移 40，1 Byte，地址偏移 63
    uint8_t Temperature_3;         //温度 3，单位℃，偏移 40，1 Byte，地址偏移 64
```

```
33.     uint8_t Temperature_4;          //温度 4，单位℃，偏移 40，1 Byte，地址偏移 65
34.     uint8_t SOC;                    //剩余电量，1 Byte，地址偏移 66
35.     uint8_t SOH;                    //电荷寿命，1 Byte，地址偏移 67
36.     uint8_t AFE_Status;             //AFE 状态，1 Byte，地址偏移 68
37.     uint8_t Working_State;            //工作状态，1 Byte，地址偏移 69
38.     uint16_t MaxChargeTime;          //最大充电持续时间，单位 min，地址偏移 70
39.     uint16_t MaxBetweenChargeTime;    //最大充电间隔时间，单位 min，地址偏移 72
40.     uint16_t MaxBetweenDisChargeTime; //最大放电持续时间，单位 min ，地址偏移 74
41.     uint16_t LastBetweenDisChargeTime;//最大放电间隔时间，单位 min，地址偏移 76
42.     uint16_t MaxUVPTime;               //最大欠压保护间隔时间，单位 min,地址偏移 78
43.     RTC_Struct_t RTC;                 //实时时钟，8Bytes，地址偏移 80
44.     RTC_Struct_t LastChargeTime;      //最近充电时钟，8Bytes,地址偏移 88
45.     RTC_Struct_t LastDisChargeTime;   //最近放电时钟，8Bytes,地址偏移 96
46.     RTC_Struct_t LastUVP_RTC;          //最近欠压保护时钟，8Bytes,地址偏移 104
47.     RTC_Struct_t LastUVP_Active_RTC;  //最近欠压保护激活时钟，8Bytes,地址偏移 112
48.     RTC_Struct_t LastFCC_Update_RTC   //最近满充容量更新时钟，8Bytes,地址偏移 120
49. }BMS_ErrorLogSaveInfo_Struct_t;
50.
51. //RTC 数据结构
52. typedef struct
53. {
54.     uint8_t RS1;                  //预留，0x00
55.     uint8_t Year;                 //年
56.     uint8_t Mouth;                //月
57.     uint8_t Date;                 //日
58.     uint8_t RS2;                  //预留，0x00
59.     uint8_t Hour;                 //时
60.     uint8_t Minute;               //分
61.     uint8_t Second;               //秒
62. }
```