MOTINOVA 中置驱动系统通信协议(BMS)

[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

二〇一九年二月二十五日

修改记录

修改日期	修改人	修改内容	版本号
20190106	周雄	第一次发布	V1.0
20190117	周雄	大范围修改	V1.1
20190130	周雄	1、增加 CDL 和其它 UART、WIFI、BLE 等接口设备 APP 之间的检测协议 2、修改 HANDSHAKE 的命令字 3、CRC 校验更改为 CRC32 校验 4、增加 BMS 一级放电过流保护 5、增加 CDL 对 MC、BMS、PBU、HMI 写入生产信息指令	V1.2
20190201	周雄	1、增加 CDL 与 APP 专用指令 2、增加 PBU 写入 MC 用户配置参数	V1.3
20190214	周雄	1、CDL 写入 BMS 的校验码更正为 12bytes 2、版本信息格式调整 3、删除 PBU 广播的关机指令 4、增加 BMS 的 Mode 和 SN 写入指令	V1.4
20190225	周雄	1、数据发送改用小端模式； 2、增加 BMS 的 MCU 故障码； 3、增加 BMS 反馈给 CDL 的通用反馈指令； 4、调整 BMS 上报运行信息和历史信息顺序； 5、增加 CDL 发送给 BMS 的复位指令；	V1.5

MOTINOVA 中置驱动系统通信协议(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

CDL：通讯适配器 CAN Dongle

APP：用户程序 Application

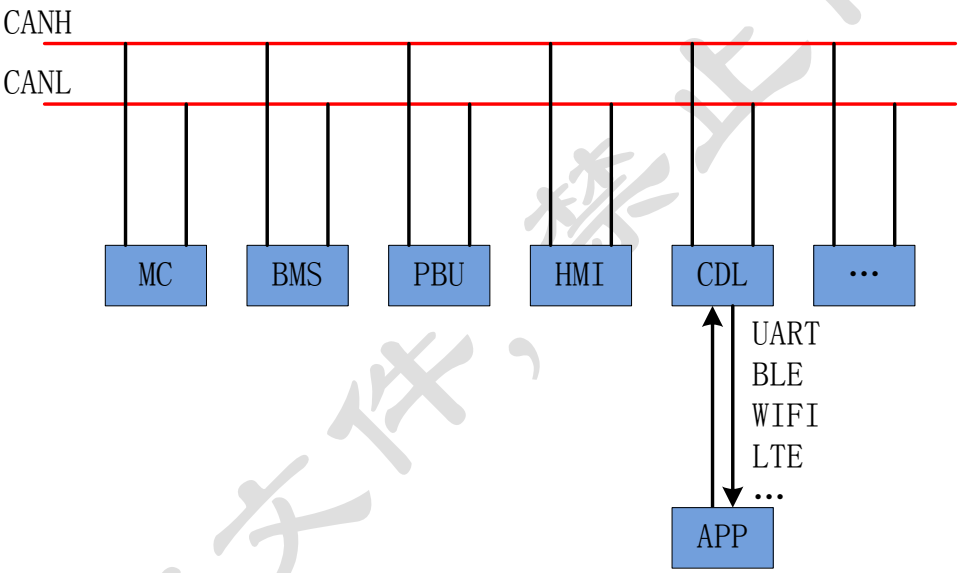

图1 系统通信接口示意图

2 通信协议规则

本协议主要描述 MOTINOVA 中置驱动系统各组件之间数据通信格式，只适用于 MOTINOVA 中置驱动系统内部组件之间通信。

2.1 硬件接口

接口类型：CAN2.0A

波特率：125kbps

2.2 数据帧封装格式

2.2.1 数据帧格式

协议描述了每帧数据内容，包括帧头、命令段长度、命令字、数据段、校验位、帧

尾。每帧格式如下：

表1 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；
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，计算结果依次写入 CRC1、CRC2、CRC3、CRC4，高位在前；
7) 数据发送时，采用小端模式。

2.2.2 ID分配

表2 ID 分配

MC	Target	广播	MC	BMS	PBU	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	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	

2.2.3 封装方式

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3 封装方式

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

3 通信内容

3.1 MC命令字定义

表4 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	ASCII 字符	READY
发送给 BMS					
0x712	0x11	0x3009	BMS 在线检测	ASCII 字符	HANDSHAKE
0x712	0x11	0x3100	查询 BMS 物理 ID		
0x712	0x11	0x3200	查询BMS校验码		
0x712	0x11	0x3300	查询BMS设计信息		

3.2 BMS命令字定义

表5 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	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) 预留:5bytes	1mV 1mA，有符型，放电为负，充电为正 1mAh 1mAh +40℃ 0～100% 0x00:休眠 0x01:充电 0x02:放电 0x04:预留 0x08:预留 0x10:预留 0x20:预留 0x40:预留 0x80:预留
0x720	0x0C	0x1120	电芯电压	Cell_1:2bytes …… Cell_16:2bytes 不足部分填充 0x00	1mV …… 1mV

	0x720	0x0C	0x1204	上报 BMS 故障码	高 16 位:0x0000 低 16 位 0x0000:无故障 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 0x1000:一级过流警告 0x2000:一级过流保护 0x4000:AFE 故障 0x8000:MCU 故障	按位或输出，0-正常，1-故障，存在故障时 200ms 自动发送，故障消失后停止发送
	0x720	0x0C	0x1308	关机指令	ASCII 字符	SHUTDOWN，收到 MC、PBU、HMI 的 READY 或超时 2s 后，关闭放电开关
	0x720	0x0C	0x1410	电池设计信息	设计容量:2bytes 设计电压:1byte 电芯型号:8bytes 预留:5bytes	1mAh 1V ASCII，0x2E 结束，无效填充 0x20
	0x720	0x0C	0x1540	上报电池版本信息	ASCII 字符	排列顺序为：MODE、SN、HW、FW； 每条信息长度为 16 bytes，结束符为

					'.',无效填充 0x20
0x720	0x0C	0x160C	返回电池物理 ID	ID:12bytes	不足位填充 1
0x720	0x0C	0x170C	返回电池校验码	校验码:12bytes	
发送给 MC					
0x721	0x0C	0x3005	在线检测反馈	ASCII 字符	READY
发送给 CDL					
0x725	0x0C	0x5028	电池 BMS 历史信息	电芯最高温:1byte 电芯最低温:1byte 最大放电电流:2bytes 最大充电电流: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	+40℃ +40℃ 1mA 1mA 次 小时 小时 次 次 次 次 次 次 次 次 1min 0～100%
0x725	0x0C	0x5120	生产信息	生产商:8bytes 生产地:8bytes	ASCII，0x2E 结束，无效填充 0x20 ASCII，0x2E 结束，

				生产日期:8bytes 预留:8bytes	无效填充 0x20 ASCII,YYYYMMDD
0x725	0x0C	0x5210	自定义可存储字符串 1	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	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	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x725	0x0C	0x5503	通用反馈指令	ASCII 字符	ACK

3.3 PBU命令字定义

表6 PBU 命令字定义

ID	模式	命令字	功能	数据段	备注
广播指令					
0x730	0x0C	0x1008	关机指令	ASCII 字符	SHUTDOWN，收到 MC、HMI 的 READY 或超时 2s 后，关闭放电开关
0x730	0x0C	0x1405	关机就绪	ASCII 字符	READY
发送给 BMS					
0x732	0x0C	0x5000	查询BMS运行信息		

3.4 HMI命令字定义

表7 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
发送给 BMS					
0x742	0x11	0x5000	查询BMS版本信息		

3.5 CDL命令字定义

表8 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	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
0x752	0x16	0x4010	写入 BMS Mode	ASCII 字符	结束符为 0x2E，无效填充 0x20
0x752	0x16	0x4110	写入 BMS SN	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

4 附录 1

4.1 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 };
```

4.2 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;
}
```