

2017-A1_ICL_CAN协议

日期	版本	修改内容
2017.01.24	无	创建新的文档
2017.02.05	0.1	初版发布
2017.03.20	0.3	T0版本 1、增加属性值，约定大小端 2、结构体边界对齐，读写方法
2017.04.25	0.5x	T2工程版本 ver 0.55-2017.05.12 1、消除一些文本表达错误及排版错误； 2、对GPRS信号值作出了更详细的定义。
2017.05.24	0.6	T2工程版本 1、由于某些原因，改原P档灯为Ready灯 2、重定义结构体ver_info_t
2017.06.24	1.0	正式版本 1、增加了对低配仪表CAN特殊功能的说明。

1. mCAN 协议概述

mCAN协议是应用在逗哈智能机车产品的CAN总线上的整车CAN通信协议。

此《2017-A1 ICL_CAN协议》符合mCAN 协议。

协议的基本特征包括：

- 1、使用 CAN2.0B链路层协议。
- 2、使用的是29位ID长度的扩展CAN协议。
- 3、CAN波特率为500K。

2. mCAN的通讯规约

mCAN 通讯中，将CAN ID 进行分段分配，分配成为设备、通道、属性等。形成某设备的某通道的某属性值的CAN ID，成为一个唯一的CAN地址。每个经过分配的CAN ID形成一个属性值，每个属性值有只读、只写、可读可写三种类型。

每个遵守 mCAN 的设备，都需要提供一个属性分配表，即CAN ID分配表，用来说明每个属性所表示的意义以及其数据类型和长度。即每个属性值都已经过事先定义，详细的属性在此文档中定义。未经定义的属性值不在产品中使用。

3. 属性值的读写

mCAN中的属性值有两种访问方法：

读取：通过远程帧来进行读取；

写入：通过数据帧进行写入。

例如：用户想要读取某个电机控制器中当前电机的转速，这里假设电机控制器器件节点ID 为 0x31，电机通道 2 的通道值为0x01，转速属性的值为0x03，如下表所示：

节点ID	通道号	属性值	数据格式说明	数据类型
------	-----	-----	--------	------

0x31 : 电机控制器	0x02 : 电机状态	0x03 : 转速	当前电机转速 单位 (rpm)	S32
-----------------	----------------	--------------	----------------------	-----

则需要向CAN总线上发送ID为0x00310203的远程帧。电机控制器收到该远程帧的请求之后，将回复一个与发送ID相同的数据帧，该数据帧中包含了当前通道2上电机的转速。数据帧中电机的转速的数据长度、数据类型可以通过属性分配表来查得。本例中，回复的数据当作一个有符号整形解析，单位为rpm。

注意：0x31与31的区别，0x31为十六进制，31为十进制。

4. 超长帧的读写

在mCAN中，为访问方便，大多数属性值的数据都在8个字节内，只需要1个CAN帧进行收发。除此这外有部分的属性值的数据区会超过8个字节（CAN单帧的最大值），在mCAN中，称此帧为超长帧。在物理上，mCAN采用分成多帧的方式进行发送。

CAN 的 8 字节数据区	
数据序号	数据
1字节	7字节

读取：通过远程帧来进行读取，在约定的时间内设备返回所有的数据分帧，否则此数据不可用。

写入：通过数据帧来进行读取，需在约定的时间内写完所有的数据分帧，否则返回失败。

具体说明：

在mCAN中，超长数据帧数据实际上是分成多个CAN帧传输的。具体的数据采用的格式由协议自行约定，在本文档上，大多数数据采用结构体的方式进行约定。

- 1、mCAN数据通道，分为上下行，分别占用一个ID
- 2、超过8个字节，用CAN帧组成一个数据包，完成多字节发送。
- 3、一个数据包最多255个CAN帧组成，总数据最多 $255 \times 7 = 1785$ 字节，更多个数据包的传输需要应用层处理。

- 4、数据包传输时，将分成n个CAN帧，形成1至n序列号。
- 5、总数据内容为：<1字节长度> + <4字节唯一ID号> + <2字节的校验> + <用户数据>
- 6、校验方式：目前采用和计算校验，为<1字节长度> + <4字节唯一ID号> + <用户数据>。
- 7、CAN帧的长度为每帧数据实际长度，每个帧的数据区的第一个字节为序列号，后7个字节为总数据内容。
- 8、接收时，必须接收完所有序号的CAN帧才算一个数据包接收完毕。
- 9、接收方完成接收，需回复相同ID。

内容为<1字节回复序号：0> + <4字节唯一ID号> + <1字节成功或失败>。

- 10、至最后一个序号的CAN帧传输完，2个CAN帧的发送间隔必须小于50ms，如果超时或接收方接受出现中途丢帧，此包均作废，回复失败，后面的接收，必须从序号0开始（新包传输）。

- 11、接收并校验成功，回复成功。
- 12、回复成功失败字节：0x00失败、0x01成功

第0帧

帧号 0	总帧数 $n(n \geq 1)$	1字节 消息类型	3字节 唯一ID	2字节 校验和
CAN 数据区（需要全8字节）				

第1-n帧

帧号 1-n	7字节用户数据
8字节CAN 数据区	

回复帧

帧号 0	回复标识 0	1字节 消息类型	3字节 唯一ID	1 字节成功失败 1成功 0 失败
CAN 数据区（只需前7字节）				

5. CAN帧体结构体：

mCAN使用CAN2.0B CAN链路层协议。ID 为29位ID，即扩展ID。

mCAN 将 CAN 的 29 位 ID 做如下分配：

CAN 29 位 ID				
R	Priority	Device ID	Channel	Property
1bit	4bits	8bits	8bits	8bits

R：标准定义。

长度：1 bits

R=0，表示该ID遵守mCAN协议。

R=1，表示该ID不遵守mCAN协议。

Priority：消息优先级。

长度：4 bits

该字段用于标明消息的优先级。字段值越小，优先级越大，如 Priority = 0 时优先级最高。

Device ID：设备 ID。

长度：8 bits

用于标识总线中的某个设备。

该字段一般由设备本身决定。如设备上的拨码开关，或者固化在设备内部。

Device ID 值为 0xFF 时，为广播 ID。

Channel：通道号。

长度：8 bits

用于表示器件内部各个功能模块的通道值。

Property：通道属性值。

长度：8 bits

用于标识某个功能通道的属性。

6. 大小端约定

对于多字节属性，mCAN协议采用小端格式，从CAN的第一个数据区填充。结构体边界对齐为1个字节。

如u32占用4个字节，u16占用2个字节。

7. CAN特殊功能约定

2017-A1车型存在低配和高配两种类型。

低配仪表有特殊需求：

1、需要在开机的时候由低配仪表(ICL)对电池(BMS)进行唤醒，唤醒方法是：在开机后，低配仪表(ICL)通过CAN向电池(BMS)发起CAN ID为0x000002F0的扩展数据帧，数据内容为8个0xFF，连续发送数量为10个。发送间隔为4ms。如果没有唤醒（没有收到电池(BMS)电量信息）则8秒后重复以上过程。

2、电池(BMS)应该在<休眠状态>时，由低配仪表(ICL)发送10个包之内或之后4秒内被唤醒，电池(BMS)立即进入<正常状态>，发送电池电量相关的信息，供仪表显示。

3、开机后，正常状态下，20秒后内仪表(ICL)没收到电池(BMS)电池电量的信息，则仪表显示电池电量为0。

8. 通用属性表

数据类型表：

类型名	说明	长度（字节）
u8	无符号单字节整型 0-255 或无符号字符型	1
u8[8]	无符号字符型数组	8
s8	有符号单字节整型 -127-128	1
u16	无符号短整型 0-65535	2
u32	无符号整型 0-4GB	4
{ u8 sn[14]; } SN_t;	编号，唯一ID	14
{ u8 shift; u8 err_code; } MCU_state2_t	shift：挡位 err_code：错误代码	2
{ u32 sw_ver; u8 hw_ver; } ver_info_t;	软硬件版本号	8

属性表：

通道号	属性值	数据格式说明	读写	数据类型
1 升级 通道	1 创建文件	FILE_INFO 文件信息		u8
	2 写入文件	FILE_CON 文件内容		u8[8]
	3 传输结束	FILE_END 结束传输		
	4 重启	RESTART_SYS 远程帧，无需回复		
2 系统 通道	1 设备 SN1	XXX_SN_REG (XXX代表某设备) 表示设备SN，此SN代表设备唯一编号。 需按照逗号《设备SN管理规范》编制。 [必须实现]	R	SN_t 超长帧

通道号	属性值	数据格式说明	读写	数据类型
	2 设备 Boot load 版本	XXX_BOOT_VER_REG 读：读取设备 Bootloader的软硬件版本号。远程帧读取。设备收到该消息之后，返回软硬件版本号 sw_ver：软件版本 PH=INT hw_ver：硬件版本 PH=INT [若无Bootloader可以不实现]	R	ver_info_t
	3 设备 App 版本	XXX_APP_VER_REG 读：读取设备 Bootloader的软硬件版本号。远程帧读取。设备收到该消息之后，返回软硬件版本号 sw_ver：软件版本 PH=INT hw_ver：硬件版本 PH=INT [必须实现]	R	ver_info_t
	4 数据接 收	XXX_DATA_RX_REG 数据接收通道 用于设备间大量数据传输 [不必须]	R	超长帧 不定长
	5 数据发 送	XXX_DATA_TX_REG 数据发送通道 用于设备间大量数据传输 [不必须]		超长帧 不定长

9. VCU节点 属性表 (VCU ID 0x0001XXXX)

数据类型表：

类型名	说明	长度（字节）
u8	无符号单字节整型 0-255 或无符号字符型	1
u8[8]	无符号字符型数组	8
s8	有符号单字节整型 -127-128	1
u16	无符号短整型 0-65535	2
u32	无符号整型 0-4GB	4
{ u16 volt; u16 current; u16 speed; u8 temp_m; u8 temp_c; }MCU_state1_t	表示电机基本状态的结构体	8
{ u8 shift; u8 err_code; }MCU_state2_t	shift：挡位 err_code：错误代码	2
{ u32 sw_ver; u8 hw_ver; }ver_info_t;	软硬件版本号	8

类型名	说明	长度 (字节)
<pre>{ u16 year; u8 month; u8 data; u8 hour; u8 minter; u8 second; }time_t</pre>	表示时间值的结构体 年月日时分秒	7
<pre>{ u8 hi_lamp; u8 low_lamp; u8 L_lamp; u8 R_lamp; u8 atmosphere_lamp; u8 one_key_lamp; u8 break_lamp; u8 hi_key_lamp; }lamp_info_t;</pre>	//灯属性 <RW> : hi_lamp 远光灯 low_lamp 近光灯 L_lamp 左转灯 R_lamp 右转灯 atmosphere_lamp 氛围灯 one_key_lamp 一键启动灯 break_lamp 刹车灯 hi_key_lamp Hi键灯	8
<pre>{ u8 H_L_key; u8 L_R_key; u8 atmosphere_key; u8 one_key; u8 klaxon_key; u8 break_key; u8 hi_key; }key_state_t;</pre>	//按钮状 <R> : H_L_key 远、近光灯 L_R_key 左、右转向灯 atmosphere_key 氛围灯 one_key 一键启动 klaxon_key 电喇叭 break_key 刹车 hi_key hi键	7
<pre>{ u8 audioBt; u8 dataBt; }bt_link_t;</pre>	//蓝牙连接状态<R> : audioBt 音频蓝牙 dataBt 数据蓝牙	2
<pre>{ u8 audioBt; u8 music; u8 klaxon; }audio_info_t;</pre>	//音频状态 : audioBt 音频蓝牙<R> music 系统音乐<RW> klaxon 电喇叭<RW>	3
<pre>typedef struct { uint8_t cmd; uint8_t loop; uint8_t _NO; uint8_t volume; }music_play_info_t;</pre>	cmd : 0播放, 1停止 loop : 循环次数 0和1一样都为1次 music_NO : 音乐编号 volume : 播放时的音量	
<pre>{ u16 speed; u32 total_km; s8 vcu_temp; }vcu_state_t;</pre>	//VCU状态<R> : 车速 总公里数 VCU温度	7

属性表 :

通道号	属性值	数据格式说明	读写	数据类型
1 升级 通道		见“通用属性”章节		

通道号	属性值	数据格式说明	读写	数据类型
2 系统通道		见“通用属性”章节		
3 功能通道	3 系统模式	VCU_MODE_STATE_REG 系统模式。 系统只能处于1-5模式之一。 1：防盗模式 2：解锁模式 3：驾驶模式 4：娱乐模式 5：升级模式 读： 当前模式状态，可能是1,2,3,4,5状态之一。 [ICL]认为： 1、4为锁定状态 2、3为为解锁状态 5为升级状态。 --->[ICL]	R	u8
	4 当前单位	VCU_UNIT 当前单位： 指的是显示单位，而不是速度或总里程的单位。 0：公制 1：英制 读： 1、改变后发出 2、远程帧 3、1Hz --->[ICL]	R	u8
3 功能通道	5 车速	VCU_SPEED_REG 车速 PH=INT/1000 Km/h 读： 1、远程帧 2、1Hz --->[ICL]	R	u32
	6 总里程	VCU_TOTAL_REG 总里程 PH=INT/1000 Km 读： 1、远程帧 2、1Hz --->[ICL]	R	u32
	9 绝对时间	VCU_DISPLAY_TIME 仪表显示时间 参见义<time_t>定义，年-月-日-时-分-秒。 读： 1、远程帧获取 2、1Hz --->[ICL]	R	time_t

通道号	属性值	数据格式说明	读写	数据类型
3 功能 通道	12 巡航	VCU_CSS 定速巡航 0：未定速巡航 1：定速巡航 读： 1、远程帧获取 2、变改时发出 3、1Hz --->[ICL]	R	u8
	13 温度	VCU_TEMP 温度 -128度至128度 读： 1、远程帧获取 2、1Hz --->[ICL]	R	s8
	14 故障码	VCU_ERR_CODE 故障码 2字节，除“00”外，按实际内容显示。 如“01”“0C”等，按段位显示 可能的字符有：“0-9”，“ACEFHJPLU” 读： 1、发生故障后1Hz发送，直至故障消除 --->[ICL]	R	u8[2]
	15 充电	VCU_CHARGE_STATE 充电状态 0：未充电 1：充电中 读： 1、远程帧获取 2、1Hz --->[ICL]	R	u8
	16 限速	VCU_SPEED_CTL 限制车的速度 0：不限速度 其它：限速度值（公里/小时） 写： 写入值进行限速值设定，成功返回限速值，失败返回-1。 --->[ICL]	RW	s16

通道号	属性值	数据格式说明	读写	数据类型																																													
4 灯	1 车灯控制	<p>VCU_LAMP_STATE_REG 车灯状态值。</p> <table><tr><th>项目</th><th>状态1</th><th>状态2</th><th>状态3</th></tr><tr><td>hi_lamp 远光灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>low_lamp 近光灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>L_lamp 左转灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>R_lamp 右转灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>atmosphere_lamp 氛围灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>one_key_lamp 一键启动灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>break_lamp 刹车灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr><tr><td>hi_key_lamp Hi键灯</td><td>1：打开</td><td>0：关闭</td><td>2：保持</td></tr></table> <p>写： 写入上值进行控制。（仅限在娱乐模式下）</p> <p>读： 1、远程帧读取仅有状态1，状态2两种值，得到车灯状态； 2、改变后发出； 3、1HZ。</p> <p>--->[ICL]</p>	项目	状态1	状态2	状态3	hi_lamp 远光灯	1：打开	0：关闭	2：保持	low_lamp 近光灯	1：打开	0：关闭	2：保持	L_lamp 左转灯	1：打开	0：关闭	2：保持	R_lamp 右转灯	1：打开	0：关闭	2：保持	atmosphere_lamp 氛围灯	1：打开	0：关闭	2：保持	one_key_lamp 一键启动灯	1：打开	0：关闭	2：保持	break_lamp 刹车灯	1：打开	0：关闭	2：保持	hi_key_lamp Hi键灯	1：打开	0：关闭	2：保持	RW	lamp_info_t									
	项目	状态1	状态2	状态3																																													
hi_lamp 远光灯	1：打开	0：关闭	2：保持																																														
low_lamp 近光灯	1：打开	0：关闭	2：保持																																														
L_lamp 左转灯	1：打开	0：关闭	2：保持																																														
R_lamp 右转灯	1：打开	0：关闭	2：保持																																														
atmosphere_lamp 氛围灯	1：打开	0：关闭	2：保持																																														
one_key_lamp 一键启动灯	1：打开	0：关闭	2：保持																																														
break_lamp 刹车灯	1：打开	0：关闭	2：保持																																														
hi_key_lamp Hi键灯	1：打开	0：关闭	2：保持																																														
	3 Ready灯	<p>VCU_READY_LIGHT Ready灯（由原P档灯改来），仅供仪表显示使用。</p> <p>1：打开 0：关闭</p> <p>读： 1、远程帧读取，得到打开，关闭2两种车灯状态； 2、改变后发出。</p> <p>--->[ICL]</p>	R	uint8_t																																													
5 按钮	1 把手状态	<p>VCU_GET_HANDLE_STATE 获取把手当前状态。</p> <table><tr><th>项目</th><th>状态1</th><th>状态2</th><th>状态3</th><th>状态4</th></tr><tr><td>H_L_key</td><td>0：远光</td><td>1：近光</td><td>-</td><td>-</td></tr><tr><td>L_R_key</td><td>0：空</td><td>1：左转</td><td>2：右转</td><td></td></tr><tr><td>atmosphere key</td><td>0：未按</td><td>1：按下</td><td>-</td><td>-</td></tr><tr><td>R_lamp 右转灯</td><td>0：未按</td><td>1：按下</td><td>-</td><td>-</td></tr><tr><td>one_key</td><td>0：未按</td><td>1：单击</td><td>2：双击</td><td>3：长按</td></tr><tr><td>hi_key</td><td>0：未按</td><td>1：单击</td><td>2：双击</td><td>3：长按</td></tr><tr><td>klaxon_key</td><td>0：未按</td><td>1：按下</td><td>-</td><td>-</td></tr><tr><td>break_key</td><td>0：未按</td><td>1：按下</td><td>-</td><td>-</td></tr></table> <p>读： 1、远程帧把手状态状态，含义如上； 2、改变后发出； 3、1Hz。</p> <p>--->[ICL]</p>	项目	状态1	状态2	状态3	状态4	H_L_key	0：远光	1：近光	-	-	L_R_key	0：空	1：左转	2：右转		atmosphere key	0：未按	1：按下	-	-	R_lamp 右转灯	0：未按	1：按下	-	-	one_key	0：未按	1：单击	2：双击	3：长按	hi_key	0：未按	1：单击	2：双击	3：长按	klaxon_key	0：未按	1：按下	-	-	break_key	0：未按	1：按下	-	-	R	key_state_t
项目	状态1	状态2	状态3	状态4																																													
H_L_key	0：远光	1：近光	-	-																																													
L_R_key	0：空	1：左转	2：右转																																														
atmosphere key	0：未按	1：按下	-	-																																													
R_lamp 右转灯	0：未按	1：按下	-	-																																													
one_key	0：未按	1：单击	2：双击	3：长按																																													
hi_key	0：未按	1：单击	2：双击	3：长按																																													
klaxon_key	0：未按	1：按下	-	-																																													
break_key	0：未按	1：按下	-	-																																													
7 蓝牙	3 数据 蓝牙连接状态	<p>VCU_DATA_BT_LINK 0：未连接 1：已连接</p> <p>读： 1、改变后发出； 2、1Hz。</p> <p>--->[ICL]</p>	R	bt_link_t																																													

通道号	属性值	数据格式说明	读写	数据类型
	4 音频蓝牙连接状态	VCU_AUDIO_BT_LINK 0: 未连接 1: 已连接 读: 1、改变后发出; 2、1Hz。 --->[ICL]	R	bt_link_t
以下VCU代为转发的模块数据，由于他们目前无法响应mCAN协议				
21 MCU 通道	1 设备SN1	MCU_SN_REG (XXX代表某设备) 表示设备SN，此SN代表设备唯一编号。 需按照逗哈《设备SN管理规范》编制。	RW	C14_t 超长帧
	2 MCU 版本	sw_ver: 软件版本 PH=INT hw_ver: 硬件版本 PH=INT		ver_info_t
	4 MCU 状态1	volt 电压: PH=INT*0.125 V current 电流: PH=INT*0.125 - 1000 mA speed 转速(有正反转): PH=INT*0.25 - 4000 rpm temp_m 电机温度: PH=INT - 40 °C temp_c 控制器温度: PH=INT - 40 °C		MCU_state1_t
	5 MCU 状态2	shift 档位: 0: P档 1: 1档 2: 2档 3: R档 err_code 电机故障码: 非0: 故障码 0: 正常 --->[ICL]		MCU_state2_t

10.TBOX节点功能属性表：(TBOX ID 0x0020XXXX)

数据类型表：

类型名	说明	长度（字节）
u8	无符号单字节整型 0-255 或无符号字符型	1
{ u8 csq; u8 gps_effective; u8 number; } GPS_satellite_t;	表示GPS卫星状态的结构体 csq: GPS信号强弱 gps_effective: 定位是否有效 number: GPS卫星数量	3

<pre>{ uint8_t csq; uint8_t is_sim_card_in; uint8_t reg_state; uint8_t link_state; } GPRS_csq_t;</pre>	表示GPRS网络状态的结构体 csq : 信号质量, 0-100, 分为5级 is_sim_card_in : 是否插入SIM卡 reg_state : 卡注册状态 link_state : 连网状态	4
--	--	---

数据类型表 :

通道号	属性值	数据格式说明	读写	数据类型
4 GPS	1 GPS卫星 参数	TBOX_GET_GPS_CSQ 卫星参数 csq : GPS信号强弱 0-255 gps_effective : 定位是否有效 0 : 无效 1 : 有效 number : GPS卫星数量 0-255 读 : 1、远程帧 2、1Hz	R	GPS_satellite_t
5 GPRS	12	TBOX_GPRS_CSQ GPRS信号质量, 范围从0至100 0 : 0格 1-20 : 1格 21-40 : 2格 41-60 : 3格 61-80 : 4格 81-100 : 5格 读 : 1、远程帧 2、1Hz	R	GPRS_csq_t

11. 仪表 : (ICL ID 0x0022XXXX)

数据类型表 :

类型名	说明	长度(byte)
u8	无符号单字节整型 0-255 或无符号字符型	1
<pre>{ u16 pitch; u16 roll; u16 yaw; u8 is_eff } imu_info_t;</pre>	表示姿态数据的结构体	6

属性表 :

通道号	属性值	数据格式说明	数据类型
1 升级通道		见“通用属性”章节	

2 系统通道		见“通用属性”章节		
3 功能通道	1 传感器 感光值	ICL_LIGHT_SENSE 感光值的百分比 PH=INT% 读：设备返回仪表上灯光传感器的感光值，如果感光值PH为255，此值无效。 1、远程帧 2、改变后立即发出 改变值暂定在2秒内超过20%则立即发出。 3、0.5HZ	R	u8
	2 姿态参数	ICL_GET_ATTITUDE pitch 俯仰 roll 横滚 yaw 航向 is_eff 是否有效 读：设备返回仪表的姿态，如果is_eff为0，此值无效。 1、远程帧 2、改变后立即发出 3、0.5HZ	R	imu_info_t
以下是仪表需要的CAN信号				
VCU				
		获取数据蓝牙连接状态，用于显示	[VCU]<---	
		获取音频蓝牙连接状态，用于显示	[VCU]<---	
		获取是否锁定状态，用于显示	[VCU]<---	
		获取VCU是否升级模式状态，用于显示	[VCU]<---	
		获取当前时间，用于显示	[VCU]<---	
		获取当前档位，用于显示	[VCU]<---	
		获取当前左右转状态，用于显示	[VCU]<---	
		获取当前大灯状态，用于显示	[VCU]<---	
		获取当前车速状态，用于显示	[VCU]<---	
		获取当前累计里程状态，用于显示	[VCU]<---	
Tbox				
		获取是否定位状态，用于显示	[Tbox]<---	
		获取GPS信号值，用于显示	[Tbox]<---	
		获取GPRS信号值，用于显示	[Tbox]<---	
BMS				
		获取当前电量，用于显示	[BMS]<---	

12.BMS节点功能属性表：(MCU ID 0x0085XXXX)

参数说明：

PH为实际值，INT为总线传输数据。

数据类型表：

类型名	说明	长度(byte)
u8	无符号单字节整型 0-255 或无符号字符型	1
u8[8]	无符号字符型数组	8
u16	无符号短整型 0-65535	2
{ u32 sw_ver; u8 hw_ver; }ver_info_t;	软硬件版本号	8
{ u16 Parm1; u16 Parm2; u16 Parm3; u16 Parm4; }bms_u16_t;	4个u16的参数	8
{ u8 Parm1; u8 Parm2; u8 Parm3; u8 Parm4; u8 Parm5; u8 Parm6; u8 Parm7; u8 Parm8; }bms_u8_t;	8个u8的参数	8

属性表：

通道号	属性值	数据格式说明	读写	数据类型
1 升级 通道	见“通用属性”章节			
2 系统 通道	见“通用属性”章节			

通道号	属性 值	数据格式说明	读写	数据类型
3 功能	3	<p>BMS_GET_GROUP1_STATE2 第1组电池状态2<用户状态> Parm1：总循环次数 PH=INT Parm2：相对电量百分比（仪表ICL显示） PH=INT% Parm3：设计容量 PH=INT 当型号为4816时：INT=0x10 当型号为4820时：INT=0x14 当型号为4826时：INT=0x1A 当型号为4832时：INT=0x20 Parm4：保护状态 c0x0001：充电温度过高保护 0x0002：充电温度过低保护 0x0004：放电温度过高保护 0x0008：放电温度过低保护 0x0010：掉线保护 0x0020：保留 0x0040：电池高压保护 0x0080：MOSFET温度过高 0x0100：电芯过压保护 0x0200：电池封装低压保护 0x0400：电芯低压保护 0x0800：充电过流保护 0x1000：放电过流保护 0x2000：短路保护 0x4000：保留 0x8000：保留 频率：1Hz 按《2017-A1 BMS样品规格书V0.1》 注意： a) 当控制器收到电池为0x10或0x14，不能应用在高配置的车上。 b) 当控制器收到电池为0x1A或0x20，可以应用在低配置的车上。 c) 电池出现高温保护时，整车控制系统应立即控制电机控制器或其它大功率用电设备马上切断电源，不能再给电机或其他大功率用电设备</p> <p style="text-align: right;">--->[ICL]</p>	R	u16[4]