

# 绿鲸智能换电电池系统通讯协议

## 小鲸换电



公司名称：杭州绿鲸科技有限公司

文件版本：Rev01.07

提交日期：2020-04-24

# 版本更新记录

时间	更新摘要	作者/更新者	版本
2020-03-04	撰写初稿	王友峰	Rev01
2020-03-05	增加 30122 寄存器（DTU 外接电源电压、DTU 内置电池电压）	王友峰	Rev01.01
2020-03-17	1、增加 modbus-tcp 协议事件出理箱说明 2、增加说明关于存储字符串数据寄存器的寄存器说明 3、更正 30622 30624 寄存器过流延时改为过流释放 4、增加 30649、30650 语音相关寄存器	王友峰	Rev01.02
2020-03-24	1、删除原第 5 章 crc 校验表 2、增加第五章系统间通讯 3、增加第六章 ota	王友峰	Rev01.03
2020-04-02	1、修改寄存器 30027-30036，将 DTU 编号备注给车架号使用 2、修改寄存器 30041-30050，将 IMEI 改为 DTU 设备号，用于阿里云注册 3、新增寄存器 30061-30070，存储设备的 IMEI 信息，新增寄存器后的原寄存器后移 4、修改寄存器 30601-30641，改为只读寄存器，将 bms 参数配置转为由 bmsota 实现，防恶意攻击修改电池配置 5、修改寄存器 30630，将均衡精度改为均衡压差	王友峰	Rev01.04
2020-04-10	1、修改第六章 O T A，增加 producttype	王友峰	Rev01.05

	2、修改 ota 文件命名方式		
2020-04-17	1、修改 ota 文件名命名方式	王友峰	Rev01.06
2020-04-24	1、修改 30112 寄存器将高字节用为 ota 文件缓存状态使用	王友峰	Rev01.07



# 目录

## 目录

绿鲸智能换电电池系统通讯协议.....	- 1 -
目录.....	- 2 -
版本更新记录.....	- 2 -
1. 协议概要.....	- 5 -
2. Modbus-RTU 协议介绍.....	- 6 -
2.1. 硬件通讯要求.....	- 6 -
2.2. 数据传输报文格式.....	- 6 -
2.2.1. 读寄存器命令 03.....	- 6 -
2.2.2. 写寄存器命令 16.....	- 8 -
2.2.3. 写文件寄存器命令 21.....	- 9 -
3. Modbus-TCP 协议介绍.....	- 1 -
3.1. 协议要求.....	- 1 -
3.2. 数据传输报文格式.....	- 2 -
3.2.1. 读寄存器命令 03.....	- 2 -
3.2.2. 写寄存器命令 16.....	- 4 -
4. 系统地址及寄存器.....	- 6 -
4.1. 系统寄存器表.....	- 6 -
4.2. 表 8 故障码列表.....	- 17 -
5. CRC 校验码表.....	错误！未定义书签。
5.1. CRC 低位字节值表 .....	错误！未定义书签。
5.2. CRC 高位字节值表 .....	错误！未定义书签。

## 1. 协议概要

Modbus 通讯协议主要有三种: Modbus RTU, Modbus ASCII 和 Modbus TCP 三种。

其中 Modbus RTU 与 Modbus ASCII 均为支持 RS-485 总线的通信协议。Modbus TCP 协议则是在 RTU 协议上加一个 MBAP 报文头, 由于 TCP 是基于可靠连接的服务, RTU 协议中的 CRC 校验码就不再需要, 所以在 Modbus TCP 协议中是没有 CRC 校验码, 用一句比较通俗的话说就是: Modbus TCP 协议就是 Modbus RTU 协议在前面加上五个 0 以及一个 6, 去掉 CRC 校验码。

为适应换电柜项目需要, 锂电池与换电柜门锁使用 RS-485 通讯, 采用标准的 Modbus-RTU 协议。而与服务器进行的通讯采用 MQTT 的方式, MQTT 建立在 TCP 基础上, 所以采用 Modbus-TCP 协议, 然而考虑到服务器的并发情况并非 TCP 单个设备连接, 则特意对 Modbus-TCP 协议进行适当的变种, 以适应服务器的需求。

## 2. Modbus-RTU 协议介绍

### 2.1. 硬件通讯要求

物理通讯：RS485。

波特率：9600bps。

数据格式：(8N1)、8 位数据位、无奇偶校验位、1 位停止位。

链路协议：Modbus-RTU

响应延时：200ms

由于 RS485 无总线仲裁机制，无法实现多主节点组网，只能实现一主多从的主从网络。通讯机制为请求/应答机制，主设备逐个轮询各个从设备。各个从设备只有在主设备发出请求的时候通过响应与主设备交互数据，主设备通过读操作获取从设备数据；主设备通过写操作使从设备获得数据，从设备之间不能直接交互数据。

整个系统中，锂电池做为从机，换电柜作为主机。

### 2.2. 数据传输报文格式

#### 2.2.1. 读寄存器命令 03

表 1:Modbus 主机发起读报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	从设备地址	1 – 253	设备ID地址
2	功能码	03	读取保持寄存器数据
3	寄存器起始地址(高位)	0x00–0xFF	保持寄存器地址高位

4	寄存器起始地址(低位)	0x00–0xFF	保持寄存器地址地位
5	读取的寄存器个数高位	0	应答报文帧长限制不超过125个字节
6	读取的寄存器个数低位	1-125	应答报文帧长限制不超过125个字节
7	低位循环冗余码校验	0x00–0xFF	循环冗余码校验字低位
8	高位循环冗余码校验	0x00–0xFF	循环冗余码校验字高位

表 2:Modbus 从机读应答报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	从设备地址	1 – 253	设备ID地址
2	功能码	03	读取保存寄存器数据
3	字节计数N	0-255	返回数据字节数
4	第1个寄存器数据的高位字节	0x00–0xFF	第1个数据的高位字节-Hi
5	第1个寄存器数据的低位字节	0x00–0xFF	第1个数据的低位字节-Lo
6	第2个寄存器数据的高位字节	0x00–0xFF	第2个数据的高位字节-Hi
7	第2个寄存器数据的低位字节	0x00–0xFF	第2个数据的低位字节-Lo
...	...		
2N+2	第N个寄存器数据的高位字节	0x00 – 0xFF	第N个数据的高位字节-Hi
2N+3	第N个寄存器数据的低位字节	0x00 – 0xFF	第N个数据的低位字节-Lo
2N+4	循环冗余校验字低位	0x00–0xFF	循环冗余校验字低位
2N+5	循环冗余校验字高位	0x00–0xFF	循环冗余校验字高位
N是基于请求寄存器数量的返回数据字节数。如果有N个请求寄存器，则有返回2N数据，返回整帧长度2N+5。			

### 2.2.2. 写寄存器命令 16

表 3:Modbus 主机写请求命令报文数据包

字节	Modbus报文描述	数据范围	数据含义
1	从设备地址	1 – 253	设备ID地址
2	功能码	16	16写多个寄存器功能码
3	寄存器起始地址高位	0x00–0xFF	写入数据的寄存器起始地址高位
4	寄存器起始地址低位	0x00–0xFF	写入数据的寄存器起始地址低位
5	写入的寄存器个数高位	0	写入数据的寄存器个数高位
6	写入的寄存器个数低位	0x01–0x78	写入数据的寄存器个数低位
7	字节数	0x00–0xF0	2*N(N为寄存器个数)
8	第1个寄存器数据的高位字节	0x00–0xFF	第1个数据的高位字节-Hi
9	第1个寄存器数据的低位字节	0x00–0xFF	第1个数据的低位字节-Lo
10	第2个寄存器数据的高位字节	0x00–0xFF	第2个数据的高位字节-Hi
11	第2个寄存器数据的低位字节	0x00–0xFF	第2个数据的低位字节-Lo
...	...		
2N+6	第N个寄存器数据的高位字节	0x00 – 0xFF	第N个数据的高位字节-Hi
2N+7	第N个寄存器数据的低位字节	0x00 – 0xFF	第N个数据的低位字节-Lo
2N+8	循环冗余校验字低位	0x00–0xFF	循环冗余校验字低位
2N+9	循环冗余校验字高位	0x00–0xFF	循环冗余校验字高位

表 4:Modbus 从机写应答报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	从设备地址	1 – 253	设备ID地址
2	功能码	16	16写寄存器功能码
3	寄存器起始地址高位	0x00–0xFF	写入数据的寄存器起始地址高位
4	寄存器起始地址低位	0x00–0xFF	写入数据的寄存器起始地址低位
5	写入的寄存器个数高位	0	写入数据的寄存器个数高位



6	写入的寄存器个数低位	0x01–0x78	写入数据的寄存器个数低位
7	循环冗余校验字低位	0x00–0xFF	循环冗余校验字低位
8	循环冗余校验字高位	0x00–0xFF	循环冗余校验字高位

### 2.2.3. 写文件寄存器命令 21

注：标准 modbus 写文件命令中数据为大端模式，stm32 单片机程序文件为小端模式，为了提高效率，双方都不做大小端转换进行透明传输。对于需要 ota 升级功能的 BMS，需要支持该命令。

表 5:Modbus 主机发起写文件报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	从设备地址	1 – 253	设备ID地址
2	功能码	21	21为写文件功能码
3	请求数据长度	0x07-0xFB	7个字节固定字段加上写文件数据长度,每包最大传输字节为:0xFB-0x07=0xF4
4	子请求x,引用类型	0x06	引用类型固定为0x06
5	子请求x,文件号	0x0001–0xFFFF	文件号，表示当前传输为第几个文件 虽然文件号允许范围为1到0xffff,但是当文件数量大于10，与从机设备的互操作性可能会受到影响
6	子请求x,记录号	0x0000-0x270F	每个文件中最大包含10000条记录
7	子请求x,记录长度	N	N为字节，记录该子请求中记录的数长度
8	子请求x,记录数据1高地址	0x00-0xFF	第1个数据的高位字节-Hi
9	子请求x,记录数据1低地址	0x00-0xFF	第1个数据的低位字节-Lo

...	...	...	
2N+6	子请求x,记录数据N高地址	0x00-0xFF	第N个数据的高位字节-Hi
2N+7	子请求x,记录数据N低地址	0x00-0xFF	第N个数据的低位字节-Lo
2N+8	循环冗余校验字低位	0x00-0xFF	循环冗余校验字低位
2N+9	循环冗余校验字高位	0x00-0xFF	循环冗余校验字高位
...	....		
...	子请求x+1...	....	下一组数据...

表 6:Modbus 从机写文件应答报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	从设备地址	1 – 253	设备ID地址
2	功能码	21	21为写文件功能码
3	请求数据长度	0x07-0xFB	7个字节固定字段加上写文件数据长度
4	子请求x,引用类型	0x06	引用类型固定为0x06
5	子请求x,文件号	0x0001-0xFFFF	文件号，表示当前传输为第几个文件 虽然文件号允许范围为1到0xffff,但是当文件数量大于10，与从机设备的互操作性可能会受到影响
6	子请求x,记录号	0x0000-0x270F	每个文件中最大包含10000条记录
7	子请求x,记录长度	N	N为字节，记录该子请求中记录的数长度
8	子请求x,记录数据1高地址	0x00-0xFF	第1个数据的高位字节-Hi
9	子请求x,记录数据1低地址	0x00-0xFF	第1个数据的低位字节-Lo

...	...	...	
2N+6	子请求x,记录数据N高地址	0x00-0xFF	第N个数据的高位字节-Hi
2N+7	子请求x,记录数据N低地址	0x00-0xFF	第N个数据的低位字节-Lo
2N+8	循环冗余校验字低位	0x00-0xFF	循环冗余校验字低位
2N+9	循环冗余校验字高位	0x00-0xFF	循环冗余校验字高位
...	....		
...	子请求x+1...	....	下一组数据...



### 3. Modbus-TCP 协议介绍

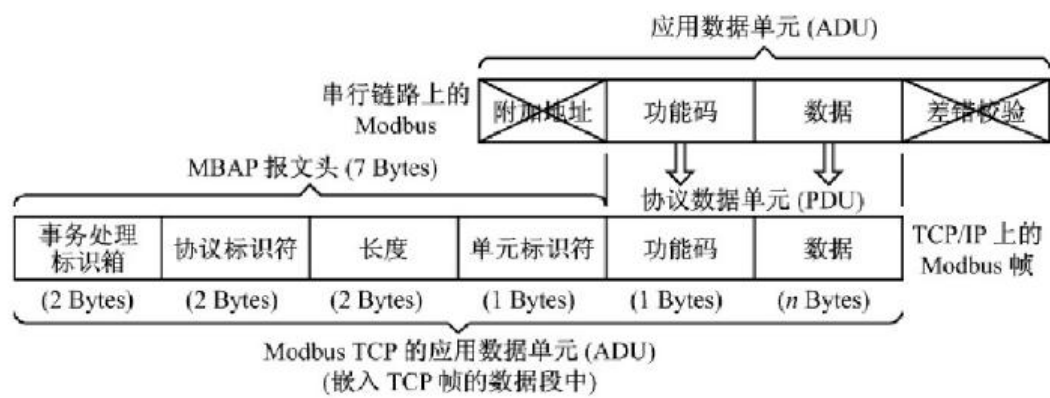
#### 3.1. 协议要求

物理通讯：MQTT 基于 TCP。

链路协议：Modbus-TCP

响应超时：10s

Modbus-TCP 与 Modbus-RTU 数据帧的区别：



MBAP 报文头

数据域	长度	描述
事务处理标识符	2Bytes	请求/响应事务处理标识符
协议标识符	2Bytes	0=Modbus协议
长度	2Bytes	随后字节的数量
单元标识符	1Bytes	远程从站的识别

Modbus-RTU 与 Modbus-TCP 读指令对比：

	MBAP报文头	地址码	功能码	寄存器地址	寄存器数量	CRC校验
Modbus RTU	无	01	03	01 8E	00 04	25 DE
Modbus TCP	00 00 00 00 00 06 00	无	03	01 8E	00 04	无

Modbus-RTU 与 Modbus-TCP 写指令对比：

	MBAP报文头	地址码	功能码	寄存器地址	寄存器数量	数据长度	正文	CRC校验
RTU	无	01	10	01 8E	00 01	02	00 00	A8 7E
TCP	00 00 00 00 00 09 00	无	10	01 8E	00 01	02	00 00	无

整个系统中，在形式上锂电池做为从机，服务器作为主机。然而根据需求锂电池可以使用寄存器读命令应答的主动上报寄存器信息。

### 3.2. 数据传输报文格式

为方便服务器处理设备应答，Modbus-TCP 协议中将事务处理标识箱填充为寄存器起始地址。

#### 3.2.1. 读寄存器命令 03

表 1:Modbus 主机发起读报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	事务处理标识箱(高位)	0x00-0xFF	事务处理标识箱高位 说明：填充寄存器起始地址(高位)
2	事务处理标识箱(低位)	0x00-0xFF	事务处理标识箱低位 说明：填充寄存器起始地址(低位)
3	协议标识符(高位)	00	00=Modbus

4	协议标识符(低位)	00	00=Modbus
5	长度(高位)	0x00–0xFF	随后的字节数量高位
6	长度(低位)	0x00–0xFF	随后的字节数量低位
7	单元标识符	01	单元标识符
8	功能码	03	读取保持寄存器数据
9	寄存器起始地址(高位)	0x00–0xFF	保持寄存器地址高位
10	寄存器起始地址(低位)	0x00–0xFF	保持寄存器地址地位
11	读取的寄存器个数高位	0	应答报文帧长限制不超过125个字节
12	读取的寄存器个数低位	1-125	应答报文帧长限制不超过125个字节

表 2:Modbus 从机读应答报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	事务处理标识箱(高位)	0x00–0xFF	事务处理标识箱高位 说明: 填充寄存器起始地址(高位)
2	事务处理标识箱(低位)	0x00–0xFF	事务处理标识箱低位 说明: 填充寄存器起始地址(低位)
3	协议标识符(高位)	00	00=Modbus
4	协议标识符(低位)	00	00=Modbus
5	长度(高位)	0x00–0xFF	随后的字节数量高位
6	长度(低位)	0x00–0xFF	随后的字节数量低位
7	单元标识符	01	单元标识符
8	功能码	03	读取保存寄存器数据
9	字节计数N	0-255	返回数据字节数
10	第1个寄存器数据的高位字节	0x00–0xFF	第1个数据的高位字节-Hi
11	第1个寄存器数据的低位字节	0x00–0xFF	第1个数据的低位字节-Lo
12	第2个寄存器数据的高位字节	0x00–0xFF	第2个数据的高位字节-Hi
13	第2个寄存器数据的低位字节	0x00–0xFF	第2个数据的低位字节-Lo
...	...		

2N+8	第N个寄存器数据的高位字节	0x00 – 0xFF	第N个数据的高位字节-Hi
2N+9	第N个寄存器数据的低位字节	0x00 – 0xFF	第N个数据的低位字节-Lo
N是基于请求寄存器数量的返回数据字节数。如果有N个请求寄存器，则有返回2N数据，返回整帧长度2N+9。			

### 3.2.2. 写寄存器命令 16

表 3:Modbus 主机写请求命令报文数据包

字节	Modbus报文描述	数据范围	数据含义
1	事务处理标识箱(高位)	0x00–0xFF	事务处理标识箱高位 说明：填充寄存器起始地址(高位)
2	事务处理标识箱(低位)	0x00–0xFF	事务处理标识箱低位 说明：填充寄存器起始地址(低位)
3	协议标识符(高位)	00	00=Modbus
4	协议标识符(低位)	00	00=Modbus
5	长度(高位)	0x00–0xFF	随后的字节数量高位
6	长度(低位)	0x00–0xFF	随后的字节数量低位
7	单元标识符	01	单元标识符
8	功能码	16	16写多个寄存器功能码
9	寄存器起始地址高位	0x00–0xFF	写入数据的寄存器起始地址高位
10	寄存器起始地址低位	0x00–0xFF	写入数据的寄存器起始地址低位
11	写入的寄存器个数高位	0	写入数据的寄存器个数高位
12	写入的寄存器个数低位	0x01–0x78	写入数据的寄存器个数低位
13	字节数	0x00–0xF0	2*N(N为寄存器个数)
14	第1个寄存器数据的高位字节	0x00–0xFF	第1个数据的高位字节-Hi
15	第1个寄存器数据的低位字节	0x00–0xFF	第1个数据的低位字节-Lo
16	第2个寄存器数据的高位字节	0x00–0xFF	第2个数据的高位字节-Hi
17	第2个寄存器数据的低位字节	0x00–0xFF	第2个数据的低位字节-Lo

...	...		
2N+12	第N个寄存器数据的高位字节	0x00 – 0xFF	第N个数据的高位字节-Hi
2N+13	第N个寄存器数据的低位字节	0x00 – 0xFF	第N个数据的低位字节-Lo

表 4:Modbus 从机写应答报文格式数据包

字节	Modbus报文描述	数据范围	数据含义
1	事务处理标识箱(高位)	0x00–0xFF	事务处理标识箱高位 说明：填充寄存器起始地址(高位)
2	事务处理标识箱(低位)	0x00–0xFF	事务处理标识箱低位 说明：填充寄存器起始地址(低位)
3	协议标识符(高位)	00	00=Modbus
4	协议标识符(低位)	00	00=Modbus
5	长度(高位)	0x00–0xFF	随后的字节数量高位
6	长度(低位)	0x00–0xFF	随后的字节数量低位
7	单元标识符	01	单元标识符
8	功能码	16	16写寄存器功能码
9	寄存器起始地址高位	0x00–0xFF	写入数据的寄存器起始地址高位
10	寄存器起始地址低位	0x00–0xFF	写入数据的寄存器起始地址低位
11	写入的寄存器个数高位	0	写入数据的寄存器个数高位
12	写入的寄存器个数低位	0x01–0x78	写入数据的寄存器个数低位



## 4. 系统地址及寄存器

注意：寄存器中存储字符串时，如果字符串长度小于寄存器长度，则低位寄存器填充0。

### 4.1. 系统寄存器表

**注意：**

所有寄存器皆可以读，所有寄存器在开机时上报一次，与BMS连接或者断开（清除BMS相关数据）时上报一次。（开机时若与BMS已经连接则按照BMS连接上报，开机时若与BMS已经断开则按照BMS断开上报。）

颜色意义：

状态信息，设备可以主动整包上报。

其余颜色，服务器在需要时来读取或设置。

寄存器不存在时填FF。

作用域	Reg#	Register Name	Length (bytes)	Unit	Offset	Access (R/W)	备注
规格信息寄存器							
BMS	30000~30009	电池包编码	20			R	电池包编码规则如下： (ASCII 码表示) <b>产品代号: 4bytes</b> 由厂商生产时定义 例如: NDFE <b>电池规格: 4bytes</b> 例如: 6020, 4820 <b>电池生产日期: 6bytes</b> 年月日各 2bytes 例如: 200106 <b>电池迭代型号: 1bytes</b> 从 A~Z, 剔除 'I', 'O'

						<p>避免与阿拉伯数字‘1’，‘0’混淆</p> <p><b>电池接口数：1bytes</b></p> <p>A:上部单口充放电，</p> <p>B:上部充放电口+底部充放电口</p> <p><b>序列号：4bytes</b></p> <p>例如：</p> <p>‘NDFE6020191110AB0001’，</p> <p>上报数据为：(16进制)</p> <p>4e44464536303230313931313130414230303031</p> <p><b>注意：寄存器中存储字符串时，如果字符串长度小于寄存器长度，则低位寄存器填充0，下同。</b></p>
30010~30019	BMS 板编号	20		0	R	<p>BMS 板编码规则如下： (ASCII 码表示)</p> <p>产品代号：8bytes (内涵厂商名+型号)</p> <p>例如：LSDBMS01</p> <p>硬件版本：2bytes 例如：01</p> <p>生产日期：6bytes 例如：200303</p> <p>序列号：4bytes 例如：0001</p> <p>“LSDBMS01012003030001”</p>
30020	单体电池串数	1 (高字节)	1 节	0	R	如 17:17 串，20: 20 串
	电池类型	1 (低字节)			R	0x01: 三元；0x02: 磷酸铁锂

	30021	电池包额定容量	2	10mAh	0	R	2000*10mAh=20Ah
	30022	电池包标称电压	2	0.1V	0	R	480*0.1V=48V
	30023	温度传感器个数	1（高字节）	1个	0	R	例如：3个
		电池包生产-年	1（低字节）			R	0-99（2000-2099）
	30024	电池包生产-月	1（高字节）			R	1~12
		电池包生产-日	1（低字节）			R	1~31
	30025	BMS 软硬件版本	2			R	硬件（高字节）、软件（低字节）
	30026	BMS 设备通讯协议版本	2			R	例如：101=1.01 版
DTU	30027-30036	DTU 编号 (作为车架号使用)	20			R	DTU 板编码规则如下： (ASCII 码表示) 产品代号：8bytes (内涵厂商名+型号) 例如：LSDDTU01 硬件版本：2bytes 例如：01 生产日期：6bytes 例如：200303 序列号：4bytes 例如：0001 “LSDDTU01012003030001”
	30037	DTU 模块类型	1			R	2:2G, 3:3G, 4:4G, , 5:5G, 6:Cat1
		DTU 安装类型	1			R	1: 分体 2: 合体
	30038	DTU 模块内核	2			R	例如：36

		版本					
	30039	DTU 硬软件版本	2			R	硬件(高字节)、软件(低字节)
	30040	DTU 设备通讯协议版本	2			R	例如: 101=1.01 版
	30041-30050	DTU 设备号 (原 IMEI, 作为阿里云注册使用)	20			R	DTU 板编码规则如下: (ASCII 码表示) 产品代号: 8bytes (内涵厂商名+型号) 例如: LSDDTU01 硬件版本: 2bytes 例如: 01 生产日期: 6bytes 例如: 200303 序列号: 4bytes 例如: 0001 “LSDDTU01012003030001”
	30051~30060	SIM 卡 ICCID	20			R	共 20 位: 例: “898602b11119c0072913”
	30061-30070	IMEI	20			R	一般 15-17 位 例: “866262045705166”
	30071	DTU 与 BMS 连接状态	1			R	0: 未连接 1: 连接
变化状态寄存器							
BMS	30100	电池包状态	1 (高字节)			R	0x00: 搁置; 0x01: 放电; 0x02: 充电
		SOC	1 (低字节)	1%	0	R	0~100

	30101	BMS 故障变化	1 (高字节)			R	0: 未变化 1: 发生变化, 服务器记录
		BMS 故障数	1 (低字节)			R	当前故障数
	30102~ 30103	故障码	4 (字节)			R	1 位代表一个故障, 一共 32 位, 见表 8 BMS 部分
	30104	电池包总电压	2	0.1V	0	R	
	30105	电池包实时充 放电电流	2	0.1A	-3200	R	
	30106	最高单体电压	2	1mV	0	R	
	30107	最低单体电压	2	1mV	0	R	
	30108	平均单体电压	2	1mV	0	R	
	30109	电池包最高温 度	1 (高字节)	1°C	-40	R	
		电池包最低温 度	1 (低字节)	1°C	-40	R	
	30110	充放电 MOS 温 度	1 (高字节)	1°C	-40	R	
		均衡电阻温度	1 (低字节)	1°C	-40	R	预留
	30111	充电 mos 状态	1 (高字节)		0	R	0x01: 断开; 0x02: 吸合
		放电 mos 状态	1 (低字节)		0	R	0x01: 断开; 0x02: 吸合
	30112	ota 固件缓存 状态	1 (高字节)		0	R	0x00: 无缓存  0x01: 有缓存  (BMS 只在充电或搁置时 进行自升级, 放电使用过 程中禁止自升级, 自升级 结束清除该状态)
		磁场状态检测	1 (低字节)		0	R	预留  0x01: 未检测到;  0x02: 检测到
DTU	30113	纬度标识	1 (高字节)			R	0x4e: 标识 N 北纬; 0x53: 标识 S 南纬

		经度标识	1（低字节）			R	0x57: 标识 W 西经; 0x45: 标识 E 东经
	30114~ 30115	纬度	4	0.000001 。	0	R	
	30116~ 30117	经度	4	0.000001 。	0	R	
	30118	网络信号强度	1（高字节）		0	R	0~31
		预留	1（低字节）		0	R	
	30119	定位方式	1（高字节）		0	R	0x01: GPS; 0x02: 基站定位
		GPS 搜星数	1（低字节）		0	R	0~100
	30120	速度	2	0.01km/h	0	R	速度只有在定位方式为 GPS 定位时有效, 其他状态默认为 0.
	30121	海拔	2	m	0	R	速度只有在定位方式为 GPS 定位时有效, 其他状态默认为 0.
	30122	DTU 外接电源电压	1（高字节）	1V	0	R	如 60:60V
		DTU 内置电池电压	1（低字节）	0.1V	0	R	如 37: 3.7V
	30123	DTU 故障变化	1（高字节）			R	0: 未变化 1: 发生变化, 服务器记录
		DTU 故障数	1（低字节）			R	当前故障数
	30124	故障码	2（字节）			R	1 位代表一个故障, 一共 16 位, 见表 8 DTU 部分
单体电压寄存器							
BMS	30200	单体电池 1 电压	2	1mV	0	R	
	30201	单体电池 2 电压	2	1mV	0	R	

	...	...	...				
	302xx(x x 为 N-1)	单体电池 N 电压	2	1mV	0	R	其中 N 为寄存器 30002 高字节定义, N 默认小于 100。当前取二十个寄存器
温度寄存器							
BMS	30300	温度传感器 1	1 (高字节)	1℃	-40	R	
		温度传感器 2	1 (低字节)	1℃	-40	R	
	30301	温度传感器 3	1 (高字节)	1℃	-40	R	
		温度传感器 4	1 (低字节)	1℃	-40	R	
	...	...	...				
	303xx(x x 为 N/2-1 如 有小数 进 1)	温度传感器 N-1	1 (高字节)	1℃	-40	R	
		温度传感器 N	1 (低字节)	1℃	-40	R	其中 N 为寄存器 30006 高字节定义, N 默认小于 10。当前默认取三个寄存器
历史数据寄存器							
BMS	30500	历史最高单体电压	2	1mV	0	R	
	30501	历史最低单体电压	2	1mV	0	R	
	30502	历史最大单体压差	2	1mV	0	R	
	30503	历史最高温度	1 (高字节)	1℃	-40	R	
		历史最低温度	1 (低字节)	1℃	-40	R	
	30504	历史最大放电电流	2	0.1A	-3200	R	
	30505	历史最大充电电流	2	0.1A	-3200	R	
	30506	BMS 充放电次数	2		0	R	连续充电 30 分钟算一次, 表示该电池被充过多少次

							电
	30507	BMS 循环次数	2		0	R	表示该电池经历过多少次充放电循环
参数设置寄存器							
BMS	30600	充电 MOS 外控使能	1（高字节）		0	R/W	0x01：断开使能；0x02：吸合使能, 合体版使用 默认：02
		放电 MOS 外控使能	1（低字节）		0	R/W	0x01：断开使能；0x02：吸合使能, 合体版使用 默认：02
	30601	充电高温保护	1（高字节）	1℃	-40	R	
		充电高温释放	1（低字节）	1℃	-40	R	
	30602	充电低温保护	1（高字节）	1℃	-40	R	
		充电低温释放	1（低字节）	1℃	-40	R	
	30603	充电高温延时	1（高字节）	1s	0	R	
		充电低温延时	1（低字节）	1s	0	R	
	30604	放电高温保护	1（高字节）	1℃	-40	R	
		放电高温释放	1（低字节）	1℃	-40	R	
	30605	放电低温保护	1（高字节）	1℃	-40	R	
		放电低温释放	1（低字节）	1℃	-40	R	
	30606	放电高温延时	1（高字节）	1s	0	R	
		放电低温延时	1（低字节）	1s	0	R	
	30607	MOS 高温保护	1（高字节）	1℃	-40	R	
		MOS 高温释放	1（低字节）	1℃	-40	R	
	30608~30610	预留	6		0	R	
	30611	整组过压保护	2	10mV	0	R	
	30612	整组过压释放	2	10mV	0	R	
	30613	整组欠压保护	2	10mV	0	R	



30614	整组欠压释放	2	10mV	0	R	
30615	整组欠压延时	1（高字节）	1s	0	R	
	整组过压延时	1（低字节）	1s	0	R	
30616	单体过压保护	2	1mV	0	R	
30617	单体过压释放	2	1mV	0	R	
30618	单体欠压保护	2	1mV	0	R	
30619	单体欠压释放	2	1mV	0	R	
30620	单体欠压延时	1（高字节）	1s	0	R	
	单体过压延时	1（低字节）	1s	0	R	
30621	充电过流保护	2	1A	0	R	
30622	充电过流延时	1（高字节）	1s	0	R	
	充电过流释放	1（低字节）	1s	0	R	
30623	放电过流保护	2	1A	0	R	
30624	放电过流延时	1（高字节）	1s	0	R	
	放电过流释放	1（低字节）	1s	0	R	
30625~ 30628	预留	8		0	R	
30629	均衡开启电压	2	1mV	0	R	单体高于这个值开启均衡功能
30630	均衡压差	2	1mV	0	R	
30631	均衡时间	2	1s	0	R	均衡持续的时间
30632	预留	2		0	R	
30633	电池功能配置	2		0	R	
30634~ 30637	预留	8		0	R	
30638	硬件单体过压	2	1mV	0	R	
30639	硬件单体欠压	2	1mV	0	R	
30640	硬件过流和延时时间	2	1s	0	R	

30641	硬件欠压和过流延时时间	2	1s	0	R	
30642	磁场检测使能	2		0	R/W	(*预留) 0x00: 不控制, 归 BMS 自动控制 0x01: 磁场检查打开 0x02: 磁场检查关闭
30643	强制进入仓储模式	2		0	R/W	0x00: 正常工作模式 0x01: 强制进入仓储模式, BMS 进入低功耗模式, 充电恢复正常工作昨天
30644	使能充电状态	2		0	R/W	0x00: 不能充电 0x01: 使能充电 (1、充电完成后, 30 分钟后自动进入不能充电 2、馈电后, 允许充电充电到 20%恢复到正常模式)
30645	bmsota 升级口令	2		0	R/W	0x0001: 开始准备升级(柜体->BMS) 0x0002: 升级文件传输结束(柜体->BMS) 0x0003: 升级过程中发生错误, 强制退出升级(柜体->BMS) 0x0004: 正常应答, 准备接收升级数据(BMS->柜体) 0x0005: 错误应答, ota 正在给通讯板升级(BMS->柜体) 0x0006: 错误应答, ota 正在给 BMS 板升级(BMS->柜体) ... 待扩展 分体版, 只支持柜子升级

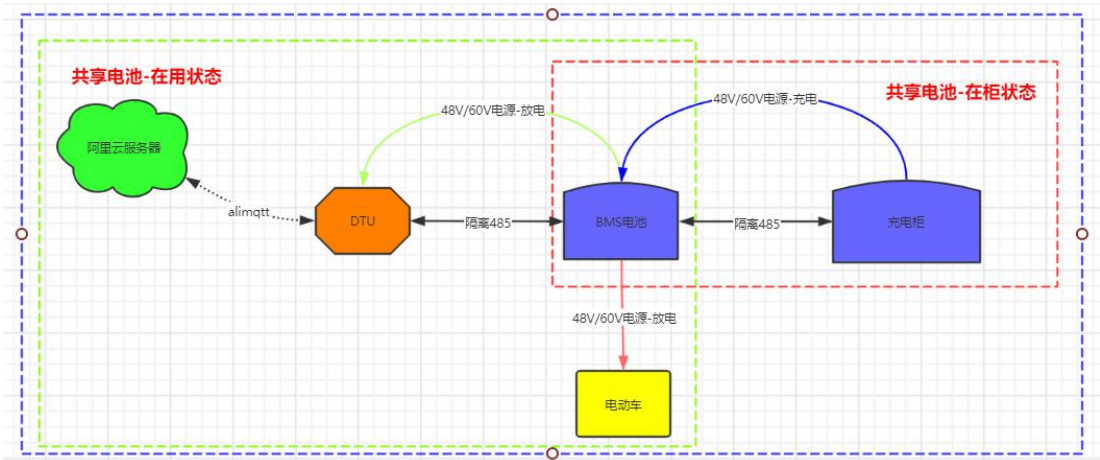
							合体版，既支持柜子升级 也支持通讯板升级
	30646	bmsota 校验口令	2		0	R/W	<p>校验口令的目的为增加数据传输的安全性与正确性。</p> <p>校验口令时，柜体向 BMS 发送一个 random 值，BMS 正确收到该值后取反+1，进行回复。柜体验证通过后，方可进行文件升级。</p> <p>分体版，只支持柜子升级</p> <p>合体版，既支持柜子升级 也支持通讯板升级</p>
	30647	电池包信息上报周期	2	s	0	R/W	1~65535s，默认 180 秒
	30648	远程锁车	2		0	R/W	<p>0：解锁</p> <p>1：锁车</p> <p>当电池拔插接入车辆时才可以锁车</p>
DTU		预留	1		0	R/W	
	30649	语音提示开关	1		0	R/W	<p>默认：1</p> <p>0：关</p> <p>1：开</p>
	30650	语音提示电量阈值	1		0	R/W	<p>例如：默认 30%</p> <p>阈值：30%，低于 30%语音提示一次</p>
		语音提示电量下降增量值	1		0	R/W	<p>默认：10%</p> <p>低于阈值后每下降 10%语音提示一次</p>
	30651~30660	OTA IP 地址	20		0	R/W	<p>例如：</p> <p>“192.168.1.0: 8283”</p>

4.2.表 8 故障码列表

BMS 故障码表		DTU 故障码表	
BMS 故障位	意义	DTU 故障位	意义
Bit0	单体过压保护	Bit0	485 故障与 BMS 连接故障
Bit1	单体欠压保护	Bit1	GPS 定位故障
Bit2	整组过压保护	Bit2	GPRS 联网故障（预留）
Bit3	整组欠压保护	Bit3	联网密钥丢失故障（预留）
Bit4	充电过温保护	Bit4	注网故障（预留）
Bit5	充电低温保护	Bit5	
Bit6	放电过温保护	Bit6	
Bit7	放电低温保护	Bit7	
Bit8	充电过流保护	Bit8	
Bit9	放电过流保护	Bit9	
Bit10	短路保护	Bit10	
Bit11	预留	Bit11	
Bit12	预留	Bit12	
Bit13	压差保护	Bit13	
Bit14	MOS 高温保护	Bit14	
Bit15	温感故障	Bit15	

# 5. 系统间通讯

小鲸换电系统如下图所示：



在用状态如绿色方框所示，DTU 与 BMS 电池通过隔离 485 通讯使用 modbus 协议，电池为 DTU 和电动车供电，此时 DTU 通过 mqtt 连接至阿里云将定位及 BMS 电池数据传至阿里云服务器。

红色方框为电池在柜状态，BMS 电池与柜体通过隔离 485 通讯，充电柜根据电池状态给充电进行充电，并将电池数据传至服务器。

## 5.1. 与 BMS 系统通讯

电池接口，上端采用 2+4 接口，下端采用圆形接口，两个口属于并联接口功能相同，都支持进行充放电及通讯功能。

4 个信号线为：485A、485B、短接点 A、短接点 B。

BMS 系统需要支持第二章中全部 BMS 域寄存器的读写功能，即第二章的 modbus-rtu 命令。

在使用状态下：

- 1、BMS 初次接入 DTU 车辆上，DTU 读取全部 BMS 相关寄存器。
- 2、定时 3s 读取 BMS 状态寄存器。
- 3、根据服务器命令 DTU 读写 BMS 相关寄存器。

在柜状态下：

- 1、BMS 初次接入 DTU 车辆上，DTU 读取全部 BMS 相关寄存器。
- 2、定时 300ms 读取 BMS 状态寄存器。

## 5.2. 与阿里云服务器通讯

DTU 通过阿里云 mqtt 协议与服务器通讯，数据内容采用 json 包的格式，包含 modbus-tcp 协议仓储有效数据内容，协议内容见第三章。例如：{"message": "modbus-tcp 内容"}

上报情况如下：

- 1、入网时，电池在就上报所有信息，否则只上报 DTU 信息
- 2、正常工作时，出现电池通信故障时，上报 DTU 故障信息寄存器
- 3、网络正常，检测到电池上线，只上报电池信息
- 4、定时上报时，如果电池在线且无通信故障时，上报 DTU+BMS 状态信息；电池离线只上报 DTU 信息
- 5、状态信息变化时引起的上报同第 4 点，上报条件如下：
  - 1) soc 下降超过 1%
  - 2) 充放电状态发生变化
  - 3) BMS 故障发生变化
  - 4) DTU 故障发生变化
  - 5) 电池包信息上报周期时间到
  - 6) 仓储模式发生变化，合体版使用



## 6. OTA 服务

### 6.1. 设备请求升级

#### 1. 获取 ota json 包信息

请求方法为: GET

请求 URL:

52.80.105.198:8283/ota/dtu?dtu=LSDDTU01012003030001

http 协议版本: HTTP/1.1

ota json 包内容如下:

```
{  
  "ota_dev": "DTU",  
  "corever": 38,  
  "hardver": 1,  
  "softver": 1,  
  "ota_url": "URL",  
  "productType": "LSDDTU01"  
}
```

#### 2. 获取升级固件

请求方法为: GET

请求 URL:

使用"ota\_url": "URL"的 url 来获取固件

http 协议版本: HTTP/1.1

例:

```
GET https://hdg-firmware.oss-cn-beijing.aliyuncs.com/LSDDTU01_38_01_02.bin HTTP/1.1
```

### 6.2. 服务器推送升级

#### 1. 下发 ota json 包信息

服务器下发 json 格式命令如下, ota\_dev 指明是哪个设备, 例如 DTU 或者 BMS, ota\_url 指明对应 url 地址。服务器推送的固件必须满足型号、内核版本、硬件版本于设备一致, 软件版本比设备版本高。

```
{  
  "ota_dev": "DTU",
```

```
"corever": 38,  
"hardver": 1,  
"softver": 1,  
"ota_url": "URL",  
"productType": "LSDDTU01"  
}
```

## 2. 获取升级固件

请求方法为: GET

请求 URL:

使用"ota\_url": "URL"的 url 来获取固件

http 协议版本: HTTP/1.1

例:

GET url HTTP/1.1

## 6.3. 文件名命名规则:

OTA 文件命名规则为:

DTU: 硬件型号+内核版本+硬件版本+软件版本

其他: 硬件型号+硬件版本+软件版本

一个硬件型号+硬件版本只能对应一个板子, 软件版本只升级到高版本。

如下所示:

### 1、通讯板

LSDDTU01\_38\_01\_02.bin

型号: LSDDTU01

内核版本: 38

硬件版本: 01

软件版本: 02

### 2、BMS

LSDBMS01\_01\_02.bin

型号: LSDBMS01

硬件版本: 01

软件版本: 02