

## 易事特 EA99 系列 Modbus 通讯协议

版权: 易事特电源股份有限公司

时间: 2009 年 2 月

## History Revision :

序号	版本	修改内容	修改时间	备注
1	Ver 1.0	确定基本的电气量	2009-02-03	
2	Ver 1.1	细化总线模式时本机的处理方案, 更正了错误帧的分类, 根据协议修改了错误响应信息的格式; 列举了每种功能码的命令信息和响应信息。	2009-02-23	
3	Ver 1.2	更改遥测、遥信电气量数目。详见第二部分	2009-02-27	
4	Ver 1.3	通信电气量序号重新编排	2009-03-11	
5	Ver 1.4	功能代码 0x04 帧格式更改	2009-05-25	

## 一、协议相关说明

### 1、协议简介

Modbus 协议是应用于控制器上的一种通用语言。通过该协议使控制器经由网络和其他 UPS 设备之间可以进行通信。本通信采用应答方式，由主机发起请求（发送遥测、遥信信息），从机执行请求并且应答。从机需通过地址设置加以区分，从机可设置的地址范围为 0x01~0xFF。

### 2、接口方式

RS485 接口：异步，半双工；如图 1 所示为监控板接口示意图，RS485 接口如(c)所示；

波特率：可设置为 1200bps、2400 bps、4800 bps、9600 bps、19200 bps

数据长度：RTU 模式时为 8 位、ASCII 模式时为 7 位

奇偶校验位：可设置为奇校验、偶校验或者无校验

停止位：1 位

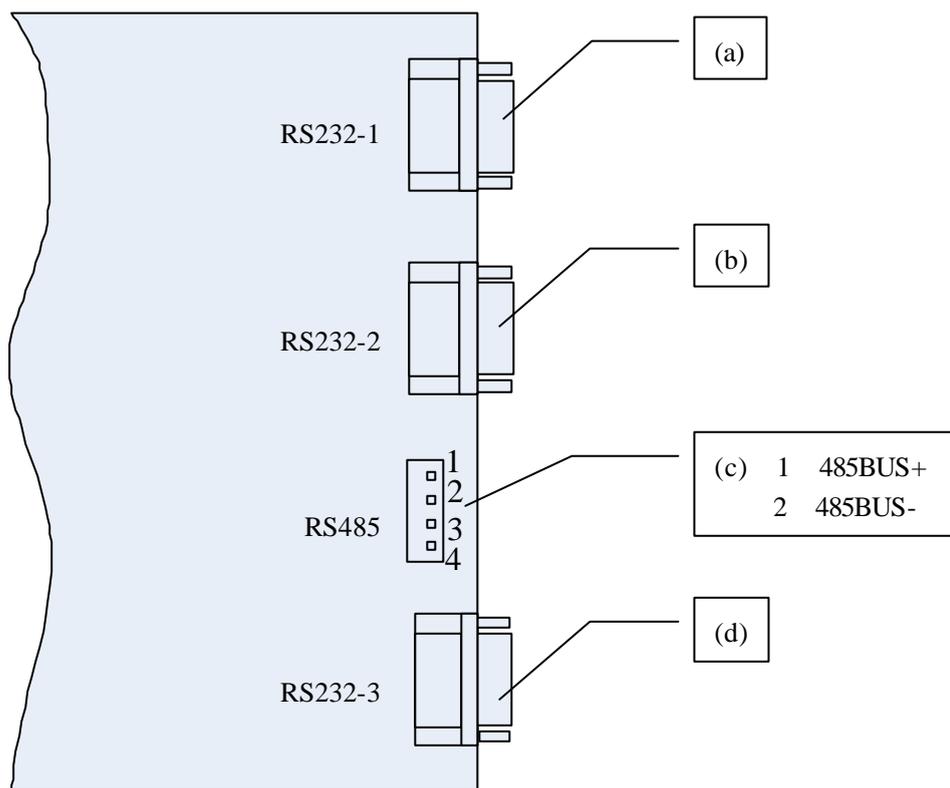


图 1 监控板接口示意图

### 3、协议格式

本协议支持 MODbus 通信模式可选，包括 RTU 模式和 ASCII 模式：

#### 3.1 RTU 模式的帧格式

控制器以 RTU 模式在 Modbus 总线上进行通讯时，信息中的每 8 bit 字节包含 2 个 4 bit 十六

进制的字符。RTU 模式中每个字节的格式为：

- 编码系统 : 8 位二进制；
- 起始位 : 1 位
- 数据位 : 8 位数据位，低位先送；
- 奇/偶校验 : 奇校验或者偶校验时为 1 位；无奇偶校验时该位为 1 位停止位；
- 停止位 : 1 位
- 错误校验区 : 循环冗余校验(CRC)

RTU 模式的请求帧格式为：

起始	设备地址	功能代码	寄存器 起始地址	寄存器 个数	CRC 高字节	CRC 低字节	结束
至少 3.5 个 字符空闲时间	8 bit	8 bit	16 bit	16 bit	8 bit	8 bit	至少 3.5 个 字符空闲时间

其中 RTU 模式字符传输格式采用 11 位传输，其中数据位为 8 位；位序列为：

起始位	1	2	3	4	5	6	7	8	停止位 (奇/偶校验位)	停止位
-----	---	---	---	---	---	---	---	---	--------------	-----

RTU 模式的响应帧格式为：

起始	设备地址	功能代码	数据	CRC 高字节	CRC 低字节	结束
至少 3.5 个 字符空闲时间	8 bit	8 bit	8n 个 bit	8 bit	8 bit	至少 3.5 个 字符空闲时间

消息发送至少需要 3.5 个字符时间的停顿间隔开始。在最后一个传输字符之后，需要至少 3.5 个字符时间的停顿来标定消息的结束。一个新的消息可在此停顿后开始。

整个消息帧必须作为一连续的流转输。如果在帧完成之前两个字符间有超过 1.5 个字符空闲的停顿时间，认为帧错误，停止接收，并重新启动接收。也就是要保证两个帧间的间隔至少大于 3.5 个字符的时间，1.5 个字符时间和 3.5 个字符时间与具体的通信波特率有关，计算方法如下：  
如通信波特率为 9600，那么

$$1.5 \text{ 个字符间隔时间} = (1/9600) \times 11 \times 1.5 \times 1000 = 1.72 \text{ ms}$$

$$3.5 \text{ 个字符间隔时间} = (1/9600) \times 11 \times 3.5 \times 1000 = 4.01 \text{ ms}$$

【例如】\*\*\*

请求帧信息：请求 1 号机的数据，位置为：寄存器起始地址 0002，寄存器个数为 1 个

	地址	功能码	寄存器起始地址		寄存器个数		CRC 校验	
数据	0x01	0x03	0x00	0x02	0x00	0x01	0x25	0xCA
字节数	1	1	2		2		2	

响应帧信息：1 号机的响应帧

	地址	功能码	返回数据字节数	数据内容		CRC 校验	
数据	0x01	0x03	0x02	0x12	0x22	0xE9	0x5C
字节数	1	1	1	2		2	

### 3.2 ASCII 模式的帧格式

当控制器以 ASCII 模式在 Modbus 总线上进行通讯时，一个信息中的每 8bit 字节作为 2 个 ASCII 字符传输，ASCII 码每个字节的格式为：

编码系统 : 16 进制，ASCII 字符‘0’-‘9’，‘A’-‘F’  
 起始位 : 1 位  
 数据位 : 7 位数据，低位先送  
 奇/偶校验 : 奇校验或者偶校验时为 1 位；无奇偶校验时该位为 1 位停止位；  
 停止位 : 1 位  
 错误校验区 : 纵向冗余校验 (LRC)

ASCII 模式的请求帧格式为：

起始	设备地址	功能代码	寄存器起始地址	寄存器个数	LRC	结束
: (0x3A)	16 bit	16 bit	32 bit	32 bit	16 bit	CRLF (0x0D0A)

其中 ASCII 模式字符传输格式，采用 10 位传输，其中 7 位数据位，位序列为：

起始位	1	2	3	4	5	6	7	停止位 (奇偶校验位)	停止位

ASCII 模式的响应帧格式为：

起始	设备地址	功能代码	数据内容	LRC	结束
: (0x3A)	16 bit	16 bit	32n 个 bit	16 bit	CRLF (0x0D0A)

ASCII 模式帧格式的帧头为“0x3A”，帧尾为“0x0D”和“0x0A”。字符之间发送的最大间隔为 1s，若大于 1s，则接收设备认为出现了一个错误。在 ASCII 模式下，数据字节全部以 ASCII 码方式发送，先发送高 4 位，然后发送低 4 位。例如：0x01，会传输 0x30，0x31 两个 ASCII 字符。此时数据采用 LRC 校验，校验涵盖从从机地址到数据的信息部分。校验和等于所有参与校验数据的字符和(舍弃进位位)的补码+1。

【例如】\*\*\*

请求帧信息：请求 1 号机 002 参数的数据帧，数据个数为 1 个：

	起始	地址	功能码	寄存器起始地址		寄存器个数		LRC	结束
字符	:	0x01	0x03	0x00	0x02	0x00	0x01	F9	CRLF
ASCII	0x3A	0x3031	0x3033	0x3030	0x3032	0x3030	0x3031	0x4639	0x0D0A

响应帧信息为：写入 4000(即 0x0FA0)到从机 1 的内部寄存器 0002 如下表：

	起始	地址	功能码	返回数据字节数	数据内容		LRC	结束
字符	:	0x01	0x06	0x02	0x0F	0xA0	0x48	CRLF
ASCII	0x3A	0x3031	0x3036	0x3032	0x3046	0x4130	0x3438	0x0D0A

#### 4、响应信息分类

主机向从机设备发送查询并希望有一个正常响应，主机查询中有可能产生 4 种事件：

(1) 从机接收查询，无通讯错误，正常处理信息，则返回一个正常响应事件。

(2) 由于通讯出错，从机不能接收查询数据，因而不返回响应。此时，主机依靠处理程序判定为查询超时。

(3) 若从机接收查询，发现有 (LRC 或 CRC) 通讯错误，不返回响应，此时依靠主机处理程序判定为查询超时。

(4) 从机接收查询，无通讯错误，但无法处理(如读不存在的寄存器地址或错误的寄存器个数)时，向主机报告错误的性质。

向主机报告错误的响应信息有 2 个与正常响应不相同的区域：

**功能代码区：**正常响应时，从机的响应功能代码区，带原查询的功能代码。所有功能代码的 MSB 为 0(其值低于 80H)。不正常响应时，从机把功能代码的 MSB 置为 1，使功能代码值大于 80H，高于正常响应的值。这样，主机应用程序能识别不正常响应事件，能检查不正常代码的数据区。

**数据区：**正常响应中，数据区含有(按查询要求给出的)数据或统计值，在不正常响应中，数据区为一个不正常代码，它说明从机产生不正常响应的条件和原因。

不正常代码及含义如下表所示：

代码	名称	含义
0x01	不合法功能代码	从机接收的是一种不能执行的功能代码。发出查询命令后，该代码指示无程序功能。
0x02	不合法数据地址	接收的数据地址，是从机不允许的地址；如：寄存器起始地址错误，查询的寄存器个数错误。

#### 【例如】\*\*\*

RTU 模式：(ASC 模式类似)

命令信息：请求 1 号机的数据，位置为：寄存器起始地址 0066，寄存器个数为 2 个

	地址	功能码	寄存器起始地址		寄存器个数		CRC 校验	
数据	0x01	0x03	0x00	0x66	0x00	0x02	0x24	0x14

响应信息：1 号机的响应帧，因为寄存器起始地址错误，因此返回信息为不合法的数据地址

	地址	功能码	数据内容	CRC 校验	
数据	0x01	0x83	0x02	0xC0	0xF1

## 5、功能代码

功能码	名称	作用
0x03	读取保持寄存器	在一个或多个保持寄存器取得当前的二进制值（可作为获取模拟量功能码）
0x04	读输入寄存器	读从机输入寄存器中的二进制数据 （可作为获取告警和状态量功能码）

## 二、通信内容

### 1、遥测量（功能码 0x03）

序号 (寄存器)	名称	DATA 类型 (Hi-Lo)	系数	单位	备注
1	交流输入电压 A	Unsigned int	0.1	伏特 V	
2	交流输入电压 B	Unsigned int	0.1	伏特 V	
3	交流输入电压 C	Unsigned int	0.1	伏特 V	
4	交流输出电压 A	Unsigned int	0.1	伏特 V	
5	交流输出电压 B	Unsigned int	0.1	伏特 V	
6	交流输出电压 C	Unsigned int	0.1	伏特 V	
7	交流输出电流 A	Unsigned int	0.1	安培 A	
8	交流输出电流 B	Unsigned int	0.1	安培 A	
9	交流输出电流 C	Unsigned int	0.1	安培 A	
10	输入直流电压	Unsigned int	0.1	伏特 V	
11	输出频率	Unsigned int	0.01	赫兹 Hz	
12	(保留)				
13	电池电压	Unsigned int	0.1	伏特 V	
14	(保留)				
15	(保留)				
16	(保留)				
17	(保留)				
18	电池温度	Unsigned int	0.1	摄氏度	
19	(保留)				
20	(保留)				
21	(保留)				
22	(保留)				
23	交流输入电流 A	Unsigned int	0.1	安培 A	
24	交流输入电流 B	Unsigned int	0.1	安培 A	
25	交流输入电流 C	Unsigned int	0.1	安培 A	
26	(保留)				
27	(保留)				
28	(保留)				
29	交流输入频率 A	Unsigned int	0.1	赫兹 Hz	
30	交流输入频率 B	Unsigned int	0.1	赫兹 Hz	
31	交流输入频率 C	Unsigned int	0.1	赫兹 Hz	
32	输入功率因数 A	Unsigned int	0.01		
33	输入功率因数 B	Unsigned int	0.01		
34	输入功率因数 C	Unsigned int	0.01		
35	旁路电压 A	Unsigned int	0.1	伏特 V	
36	旁路电压 B	Unsigned int	0.1	伏特 V	
37	旁路电压 C	Unsigned int	0.1	伏特 V	

38	旁路频率 A	Unsigned int	0.1Hz	赫兹 Hz	
39	旁路频率 B	Unsigned int	0.1Hz	赫兹 Hz	
40	旁路频率 C	Unsigned int	0.1Hz	赫兹 Hz	
41	(保留)				
42	(保留)				
43	(保留)				
44	(保留)				
45	(保留)				
46	(保留)				
47	输出功率因数 A	Unsigned int	0.01		
48	输出功率因数 B	Unsigned int	0.01		
49	输出功率因数 C	Unsigned int	0.01		
50	A 相输出视在功率	Unsigned int	0.1	千伏安 kVA	
51	B 相输出视在功率	Unsigned int	0.1	千伏安 kVA	
52	C 相输出视在功率	Unsigned int	0.1	千伏安 kVA	
53	A 相输出有功功率	Unsigned int	0.1	千瓦 kW	
54	B 相输出有功功率	Unsigned int	0.1	千瓦 kW	
55	C 相输出有功功率	Unsigned int	0.1	千瓦 kW	
56	(保留)				
57	(保留)				
58	(保留)				
59	A 相输出负载百分比	Unsigned int	0.1	%	
60	B 相输出负载百分比	Unsigned int	0.1	%	
61	C 相输出负载百分比	Unsigned int	0.1	%	
62	电池电流	int	0.1	安培 A	正数：充电电流； 负数：放电电流
63	电池剩余后备时间	Unsigned int	0.1	分钟 min	
64	电池容量%	Unsigned int	0.1	%	
65	并机 A 相输出视在功率	Unsigned int	0.1	千伏安 kVA	
66	并机 B 相输出视在功率	Unsigned int	0.1	千伏安 kVA	
67	并机 C 相输出视在功率	Unsigned int	0.1	千伏安 kVA	
68	并机 A 相输出有功功率	Unsigned int	0.1	千瓦特 kW	
69	并机 B 相输出有功功率	Unsigned int	0.1	千瓦特 kW	
70	并机 C 相输出有功功率	Unsigned int	0.1	千瓦特 kW	
71	(保留)				
72	(保留)				
73	(保留)				
74	并机 A 相输出负载百分比	Unsigned int	0.1	%	
75	并机 B 相输出负载百分比	Unsigned int	0.1	%	
76	并机 C 相输出负载百分比	Unsigned int	0.1	%	
77	环境温度	Unsigned int	0.1	摄氏度	
78	(保留)				
79	(保留)				

80	(保留)				
81	(保留)				
82	(保留)				
83	(保留)				
84	(保留)				
85	(保留)				
86	(保留)				
87	(保留)				
88	(保留)				
89	(保留)				
90	(保留)				
91	(保留)				
92	(保留)				
93	(保留)				
94	(保留)				
95	(保留)				
96	(保留)				
97	(保留)				
98	(保留)				
99	(保留)				
100	(保留)				

注：

**Unsigned int** : 为无符号 16bit 整型。

**Int** : 为有符号 16bit 整型。

【例如】\*\*\*

假设 UPS 设备地址设置为 0x18，查询寄存器起始地址值为 0x0008，寄存器个数为 2 个，即查询“交流输出电流 B”和“交流输出电流 C”的值；假设此时“交流输出电流 B”的值为 89.2A，交流输出电流 C”的值为 88.9A，根据该值的系数为 0.1，那么：

寄存器 0x0008 的值为： $(892)_D = (037C)_H$

寄存器 0x0009 的值为： $(889)_D = (0379)_H$

则返回数据的字节数为 4 个，RTU 模式时，对数据查询的请求帧信息和响应帧信息为：

请求帧信息为：

	地址	功能码	寄存器起始地址	寄存器个数	CRC 校验
数据	0x18	0x03	0x0008	0x0002	0xC047

响应帧信息为：

	地址	功能码	返回数据字节数	数据内容	CRC 校验
数据	0x18	0x03	0x04	0x037C 0x0379	0x7C72

对上述情况采用 ASCII 模式时，对数据查询的请求帧信息和响应帧信息为：

请求帧信息为：

	起始	地址	功能码	寄存器起始地址		寄存器个数		LRC	结束
数据	:	0x18	0x03	0x0008		0x0002		0xDB	CRLF
ASCII	0x3A	0x3138	0x3033	0x3030	0x3038	0x3030	0x3032	0x4442	0x0D0A

响应帧信息为：

	起始	地址	功能码	返回数据 字节数	数据内容				LRC	结束
数据	:	0x18	0x03	0x04	0x037C		0x0379		0xE6	CRLF
ASCII	0x3A	0x3138	0x3033	0x3034	0x3033	0x3743	0x3033	0x0D0A	0x4536	0x0D0A

## 2、遥信量（功能码 0x04）

序号(寄存器)	名称	类型	备注
101	供电方式	Unsigned int	0x0001：UPS供电 0x0002：旁路供电 0x0080：均不供电 0x0081：邻机供电
102	(保留)		
103	电池状态	Unsigned int	0x0000：电池充电器不工作 0x0001：电池自检状态 0x0002：电池均充 0x0004：电池浮充 0x0008：电池放电
104	整流器状态	Unsigned int	0x0000：整流器OFF 0x0001：整流器软启动 0x0002：整流器正常工作
105	输入空开状态	Unsigned int	0x0001：输入空开闭合 0x0000：输入空开断开
106	开机允许状态	Unsigned int	0x0001：开机允许 0x0000：开机禁止
107	电池接触器状态	Unsigned int	0x0001：接触器闭合 0x0000：接触器断开
108	电池有无状态	Unsigned int	0x0001：电池已连接 0x0000：电池未连接
109	输出空开状态	Unsigned int	0x0001：输出空开闭合 0x0000：输出空开断开
110	旁路可供电状态	Unsigned int	0x0001：旁路可供电 0x0000：旁路不可供电
111	逆变器状态	Unsigned int	0x0000：逆变器OFF 0x0001：逆变器软启动 0x0002：逆变器正常工作
112	逆变供电状态	Unsigned int	0x0000：逆变不可供电 0x0001：逆变可供电，但未供电 0x0002：逆变正供电
113	维修旁路空开状态	Unsigned int	0x0001：维修旁路空开闭合 0x0000：维修旁路空开断开
114	旁路空开状态	Unsigned int	0x0001：旁路空开闭合 0x0000：旁路空开断开
115	紧急关机提示	Unsigned int	0x0001：紧急关机提示 0x0000：紧急关机无提示
116	维修模式	Unsigned int	0x0001：维修模式 0x0000：非维修模式
117	启机容量不足， 无法承担当前负载	Unsigned int	0x0001：无法承担当前负载 0x0000：正常

118	负载大于单机容量， 无法完成间断切换	Unsigned int	0x0001：无法完成间断切换 0x0000：正常
119	发电机接入	Unsigned int	0x0001：已连接 0x0000：未连接
120	电池接触器驱动信号状态	Unsigned int	0x0001：脱扣 0x0000：无
121	电池BCB接入	Unsigned int	0x0001：已连接 0x0000：未连接
122	电池BCB状态	Unsigned int	0x0001：闭合 0x0000：断开
123	(保留)		
124	(保留)		
125	(保留)		
126	(保留)		
127	(保留)		
128	(保留)		
129	(保留)		
130	(保留)		
131	(保留)		
132	(保留)		
133	(保留)		
134	(保留)		
135	(保留)		
136	(保留)		
137	(保留)		
138	(保留)		
139	(保留)		
140	(保留)		
141	(保留)		
142	(保留)		
143	(保留)		
144	(保留)		
145	(保留)		
146	(保留)		
147	(保留)		
148	(保留)		
149	(保留)		
150	(保留)		
151	同步/不同步状态	Unsigned int	0x0000：同步      0x00F0：不同步
152	市电故障	Unsigned int	0x0000：正常      0x00F0：故障
153	整流器故障	Unsigned int	0x0000：正常      0x00F0：故障
154	逆变器故障	Unsigned int	0x0000：正常      0x00F0：故障
155	旁路故障	Unsigned int	0x0000：正常      0x00F0：故障

156	电池电压故障	Unsigned int	0x0000 : 正常      0x00F0 : 故障
157	(保留)		
158	(保留)		
159	(保留)		
160	(保留)		
161	(保留)		
162	(保留)		
163	电池温度	Unsigned int	0x0000 : 正常      0x00F0 : 故障
164	(保留)		
165	(保留)		
166	(保留)		
167	(保留)		
168	交流输入A相电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
169	交流输入B相电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
170	交流输入C相电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
171	交流输入A相电流	Unsigned int	0x0000 : 正常      0x00F0 : 故障
172	交流输入B相电流	Unsigned int	0x0000 : 正常      0x00F0 : 故障
173	交流输入C相电流	Unsigned int	0x0000 : 正常      0x00F0 : 故障
174	交流输出A相电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
175	交流输出B相电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
176	交流输出C相电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
177	交流输出A相电流	Unsigned int	0x0000 : 正常      0x00F0 : 故障
178	交流输出B相电流	Unsigned int	0x0000 : 正常      0x00F0 : 故障
179	交流输出C相电流	Unsigned int	0x0000 : 正常      0x00F0 : 故障
180	直流电压	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
181	输出频率	Unsigned int	0x0000 : 正常 0x0001 : 低于下限 0x0002 : 高于上限
182	市电电压异常	Unsigned int	0x0000 : 正常      0x00F0 : 故障
183	市电频率异常	Unsigned int	0x0000 : 正常      0x00F0 : 故障

184	主路输入相序接反故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
185	输入软启动失败	Unsigned int	0x0000 : 正常	0x00F0 : 故障
186	整流IGBT过流故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
187	输入电感过温	Unsigned int	0x0000 : 正常	0x00F0 : 故障
188	整流器过温	Unsigned int	0x0000 : 正常	0x00F0 : 故障
189	母线过压	Unsigned int	0x0000 : 正常	0x00F0 : 故障
190	风扇故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
191	主路反灌保护	Unsigned int	0x0000 : 正常	0x00F0 : 故障
192	平衡电路故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
193	母线电压低关机	Unsigned int	0x0000 : 正常	0x00F0 : 故障
194	电池接反	Unsigned int	0x0000 : 正常	0x00F0 : 故障
195	电池漏电保护	Unsigned int	0x0000 : 正常	0x00F0 : 故障
196	电池冷启动失败	Unsigned int	0x0000 : 正常	0x00F0 : 故障
197	充电器过压	Unsigned int	0x0000 : 正常	0x00F0 : 故障
198	电池过温	Unsigned int	0x0000 : 正常	0x00F0 : 故障
199	充电器故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
200	电池电压低	Unsigned int	0x0000 : 正常	0x00F0 : 故障
201	电池EOD	Unsigned int	0x0000 : 正常	0x00F0 : 故障
202	整流通讯故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
203	旁路相序反	Unsigned int	0x0000 : 正常	0x00F0 : 故障
204	旁路电压异常	Unsigned int	0x0000 : 正常	0x00F0 : 故障
205	旁路晶闸管故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
206	旁路过载	Unsigned int	0x0000 : 正常	0x00F0 : 故障
207	旁路过载延时到	Unsigned int	0x0000 : 正常	0x00F0 : 故障
208	旁路频率超跟踪	Unsigned int	0x0000 : 正常	0x00F0 : 故障
209	旁路反灌保护	Unsigned int	0x0000 : 正常	0x00F0 : 故障
210	负载冲击转旁路	Unsigned int	0x0000 : 正常	0x00F0 : 故障
211	本机过载超时	Unsigned int	0x0000 : 正常	0x00F0 : 故障
212	逆变切旁路时,旁路异常关机	Unsigned int	0x0000 : 正常	0x00F0 : 故障
213	辅助电源掉电	Unsigned int	0x0000 : 正常	0x00F0 : 故障
214	辅助电源故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
215	邻机请求转旁路	Unsigned int	0x0000 : 正常	0x00F0 : 故障
216	本小时切换次数限制	Unsigned int	0x0000 : 正常	0x00F0 : 故障
217	(保留)			
218	(保留)			
219	(保留)			
220	(保留)			
221	(保留)			
222	逆变电感过温	Unsigned int	0x0000 : 正常	0x00F0 : 故障
223	逆变器过温	Unsigned int	0x0000 : 正常	0x00F0 : 故障
224	逆变IGBT过流	Unsigned int	0x0000 : 正常	0x00F0 : 故障
225	输出熔断器故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
226	本机输出过载	Unsigned int	0x0000 : 正常	0x00F0 : 故障

227	逆变晶闸管故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
228	逆变通讯故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
229	手动开机失败	Unsigned int	0x0000 : 正常	0x00F0 : 故障
230	并机线连接故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
231	并机均流故障	Unsigned int	0x0000 : 正常	0x00F0 : 故障
232	用户操作错误	Unsigned int	0x0000 : 正常	0x00F0 : 故障
233	输出短路	Unsigned int	0x0000 : 正常	0x00F0 : 故障
234	电池手动自检测试	Unsigned int	0x0000 : 无意义 0x0001 : 测试成功 0x0002 : 测试失败	
235	电池容量测试	Unsigned int	0x0000 : 无意义 0x0001 : 测试成功 0x0002 : 测试失败	
236	系统测试	Unsigned int	0x0000 : 无意义 0x0001 : 测试成功 0x0002 : 测试失败	
237	(保留)			
238	(保留)			
239	(保留)			
240	(保留)			
241	(保留)			
242	(保留)			
243	(保留)			
244	(保留)			
245	(保留)			
246	(保留)			
247	(保留)			
248	(保留)			
249	(保留)			
250	(保留)			
251	(保留)			
252	(保留)			
253	(保留)			
254	(保留)			
255	(保留)			
256	(保留)			
257	(保留)			
258	(保留)			
259	(保留)			
260	(保留)			

注：

Unsigned int : 为无符号整型。

## 【例如】\*\*\*

假设 UPS 设备地址设置为 0x18，查询寄存器起始地址值为 103，即 0x0067，寄存器个数为 1 个，即查询“电池状态”；假设此时“电池状态”为放电状态，即 0x08。

则返回数据的字节数为 1 个，RTU 模式时，对状态查询的请求帧信息和响应帧信息为：

请求帧信息为：

	地址	功能码	寄存器起始地址	寄存器个数	CRC 校验
数据	0x18	0x04	0x0067	0x0001	0x1C82

响应帧信息为：

	地址	功能码	返回数据字节数	数据内容	CRC 校验
数据	0x18	0x04	0x02	0x0008	0x1347

对上述情况采用 ASCII 模式时，对数据查询的请求帧信息和响应帧信息为：

请求帧信息为：

	起始	地址	功能码	寄存器起始地址		寄存器个数		LRC	结束
数据	:	0x18	0x04	0x0067		0x0001		0x7C	CRLF
ASCII	0x3A	0x3138	0x3034	0x3030	0x3637	0x3030	0x3031	0x3743	0x0D0A

响应帧信息为：

	起始	地址	功能码	返回数据字节数	数据内容		LRC	结束
数据	:	0x18	0x04	0x02	0x0008		0xDB	CRLF
ASCII	0x3A	0x3138	0x3034	0x3032	0x3030	0x3038	0x4442	0x0D0A

## 附录 A LRC/CRC 校验

- LRC 校验
- CRC 校验

### LRC 纵向冗余校验

LRC 域是一个包含一个 8 位二进制值的字节。LRC 值由传输设备来计算并放到消息帧中，接收设备在接收消息的过程中计算 LRC，并将它和接收到消息中 LRC 域中的值比较，如果两值不等，说明有错误。

LRC 校验比较简单，它在 ASCII 协议中使用，检测了消息域中除开始的冒号及结束的回车换行号外的内容。它仅仅是把每一个需要传输的数据按字节叠加后取反加 1 即可。

### CRC 循环冗余校验

循环冗余校验 CRC 区为 2 字节，含一个 16 位二进制数据。由发送设备计算 CRC 值，并把计算值附在信息中，接收设备在接收信息时，重新计算 CRC 值，并把计算值与接收的在 CRC 区中实际值进行比较，若两者不相同，则产生一个错误。

CRC 开始时先把寄存器的 16 位全部置成“1”，然后把相邻 2 个 8 位字节的数据放入当前寄存器中，只有每个字符的 8 位数据用作产生 CRC，起始位、停止位和奇偶校验位不加入到 CRC 中。

产生 CRC 期间，每 8 位数据与寄存器中值进行异或运算，其结果向右移一位(向 LSB 方向)，并用“0”填入 MSB，检测 LSB，若 LSB 为“1”则与预置的固定值异或，若 LSB 为“0”则不作异或运算。

重复上述过程，直至移位 8 次，完成第 8 次移位后，下一个 8 位数据，与该寄存器的当前值异或，在所有信息处理完后，寄存器中的最终值为 CRC 值。

## 附录 B 高低位字节表

### 高位字节表

/\* Table of CRC values for high-order byte \*/

static unsigned int auchCRCHi[] = {

```
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40 } ;
```

### 低位字节表

/\* Table of CRC values for low-order byte \*/

static unsigned int auchCRCLo[] = {

```
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 } ;
```