

易事特集团股份有限公司	文件编号		文件版本	V1.0
	文件密级	保密	生效日期	2020.09
	制定部门		软件部	

EA900 G4 6-20k

研发版 Modbus 通讯协议

易事特集团股份有限公司	文件编号		文件版本	V1.0
	文件密级	保密	生效日期	2020.09
	制定部门	软件部		

序号	版本	修改内容	修改人	修改时间	备注
1	Ver 1.0	新制定	钟庭欢	2020-9-30	2.7

目 录

一、协议相关说明.....	1
1、协议适用范围.....	1
2、规范性引用文件.....	1
3、协议简介.....	1
4、接口方式.....	1
6、响应信息分类.....	5
7、功能代码.....	7
二、寄存器列表.....	8
1、读输入寄存器（功能码 0x04）.....	8
2、读离散量（功能码0x02）.....	16
3、读保持寄存器（功能码 0x03）.....	19
4、预置寄存器列表（功能码 0x06, 0x10）.....	26
5、历史记录寄存器列表（功能码 0x41, EA900 G4 6-20K 专用）.....	26
三、通信内容.....	27
1、读输入寄存器（功能码0x04）.....	27
2、读离散量（功能码0x02）.....	28
3、读取设备寄存器（功能码 0x03）.....	29
4、预置寄存器（功能码 0x06,0x10）.....	30
5、历史记录寄存器列表（功能码 0x41, EA900 G4 6-20K 专用）.....	30
附录A CRC 校验.....	31
附录B 高低位字节表.....	33

一、协议相关说明

1、协议适用范围

本协议文档规范了设备 EA900 G4 6-20K UPS 通过 RS 232 连接提供的 Modbus 接口需求。

2、规范性引用文件

1. RFC791, 互联网协议, Sep81 DARPA

2. MODBUS 协议参考指南 Rev J,MODICON, 1996 年 6 月, doc#PI_MBUS_300

MODBUS 是一项应用层报文传输协议, 用于在通过不同类型的总线或网络连接的设备之间的客户机/服务器通信。目前, 使用下列情况实现 MODBUS:

1) 以太网上的 TCP/IP。

2) 各种媒体 (有线: EIA/TIA-232-E、EIA/TIA-485-A; 光纤、无线等等) 上的异步串行传输。

3、协议简介

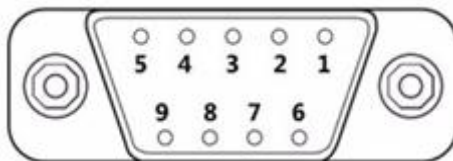
Modbus 协议是应用于控制器上的一种通用语言。通过该协议使控制器经由网络和其他 UPS 设备之间可以进行通信。本通信采用应答方式, 由主机发起请求 (发送遥测、遥信信息), 从机执行请求并且应答。从机需通过地址设置加以区分, 从机可设置的地址范围为 1~247。

4、接口方式

1、接口说明:

RS232 接口:	异步, 半双工
波特率:	默认 9600bps
数据长度:	RTU 模式、8 位
停止位:	1 位

2、接口示意图（根据实际机器接口图和引脚定义，可使用机器本身说明书上的说明）
PC 接线与 UPS 接线规则(采用 9Pin 头母座)



DB9母头（孔）

引脚编号	名称	功能
2	RXD（Receibed Data）	串口数据输入
3	TXD（Trasmitted Data）	串口数据输出
5	GND（Signal Ground）	信号地线

5、Modbus RTU数据帧格式

本协议仅支持Modbus 通信RTU 模式。

控制器以RTU 模式在Modbus 总线上进行通讯时，信息中的每个字节按十六进制。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；
- 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 低字节	CRC 高字节	结束
至少3.5 个字符空闲时间	1 byte	1 byte	2 bytes	2 bytes	1 byte	1 byte	至少3.5 个字符空闲时

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，若无奇偶校验位，则 10 位传输

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 低字节	CRC 高字节	结束
至少3.5 个字符空闲时间	1 byte	1 byte	N bytes	1 byte	1 byte	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 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，寄存器内容为0x1222

请求帧信息：

	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：

	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	0x34	0xFD
字节数	1	1	1	2		2	

6、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。
0x03	不合法数据	查询数据区的值是从机不允许的值。
0x04	从机设备故障	从机执行主机请求的动作时出现不可恢复的错误。
0x08	内存奇偶校验错误	从机读扩展内存中的数据时，发现有奇偶校验错误，主机按从机的要求重新发送数据请求。

【例如】***

RTU 模式：(ASCII 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

7、功能代码

功能码	名称	作用
0x02	读离散量输入	读从机离散量输入寄存器中的二进制数据
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
0x04	读取输入寄存器	在一个或多个输入寄存器取得当前的二进制值
0x06	写单个保持寄存器	写从机上的单个寄存器（可作为设置单个参数的功能码）
0x10	写多个保持寄存器	写从机上的多个寄存器（可作为设置多个参数的功能码）
0x41	读取历史记录	读取一条历史记录

二、寄存器列表（若预留位有值不可用，则备注说明无效位）

1、读输入寄存器（功能码 0x04）

地址		数据内容	数据长度 /格式	类型	单位	系数	备注
HEX	DEC						
0x0000	0	旁路A相电压	2Bytes	SHORT	V	0.1	
0x0001	1	预留2	2Bytes	SHORT	/	0.1	
0x0002	2	预留3	2Bytes	SHORT	/	0.1	
0x0003	3	旁路A相电流	2Bytes	SHORT	A	0.1	
0x0004	4	预留5	2Bytes	SHORT	/	0.1	
0x0005	5	预留6	2Bytes	SHORT	/	0.1	
0x0006	6	旁路频率	2Bytes	SHORT	Hz	0.1	
0x0007	7	预留8	2Bytes	SHORT	/	0.1	
0x0008	8	预留9	2Bytes	SHORT	/	0.1	
0x0009	9	旁路PF	2Bytes	SHORT	/	0.01	
0x000A	10	预留11	2Bytes	SHORT	/	1	
0x000B	11	预留12	2Bytes	SHORT	/	1	
0x000C	12	输入A相电压	2Bytes	SHORT	V	0.1	
0x000D	13	输入B相电压	2Bytes	SHORT	/	0.1	
0x000E	14	输入C相电压	2Bytes	SHORT	/	0.1	
0x000F	15	输入A相电流	2Bytes	SHORT	A	0.1	
0x0010	16	输入B相电流	2Bytes	SHORT	/	0.1	
0x0011	17	输入C相电流	2Bytes	SHORT	/	0.1	
0x0012	18	输入频率	2Bytes	SHORT	Hz	0.1	
0x0013	19	预留20	2Bytes	SHORT	/	0.1	
0x0014	20	预留21	2Bytes	SHORT	/	0.1	
0x0015	21	输入PF	2Bytes	SHORT	/	0.01	
0x0016	22	预留23	2Bytes	SHORT	/	1	
0x0017	23	预留24	2Bytes	SHORT	/	1	
0x0018	24	输出A相电压	2Bytes	SHORT	V	0.1	
0x0019	25	预留26	2Bytes	SHORT	/	0.1	
0x001A	26	预留27	2Bytes	SHORT	/	0.1	
0x001B	27	输出A相电流	2Bytes	SHORT	A	0.1	
0x001C	28	预留29	2Bytes	SHORT	/	0.1	
0x001D	29	预留30	2Bytes	SHORT	/	0.1	
0x001E	30	输出频率	2Bytes	SHORT	Hz	0.1	
0x001F	31	预留32	2Bytes	SHORT	/	0.1	

0x0020	32	预留33	2Bytes	SHORT	/	0.1	
0x0021	33	输出PF	2Bytes	SHORT	/	0.01	
0x0022	34	预留35	2Bytes	SHORT	/	1	
0x0023	35	预留21	2Bytes	SHORT	/	1	
0x0024	36	输出视在功率	2Bytes	SHORT	KVA	0.1	
0x0025	37	预留38	2Bytes	SHORT	/	0.1	
0x0026	38	预留39	2Bytes	SHORT	/	0.1	
0x0027	39	输出有功功率	2Bytes	SHORT	KW	0.1	
0x0028	40	预留41	2Bytes	SHORT	/	0.1	
0x0029	41	预留42	2Bytes	SHORT	/	0.1	
0x002A	42	输出无功功率	2Bytes	SHORT	KVAR	0.1	
0x002B	43	预留44	2Bytes	SHORT	/	0.1	
0x002C	44	预留45	2Bytes	SHORT	/	0.1	
0x002D	45	输出负载百分比	2Bytes	SHORT	%	0.1	
0x002E	46	预留47	2Bytes	SHORT	/	1	
0x002F	47	预留48	2Bytes	SHORT	/	1	
0x0030	48	预留49	2Bytes	SHORT	/	1	
0x0031	49	电池电压	2Bytes	SHORT	V	0.1	
0x0032	50	预留51	2Bytes	SHORT	/	1	
0x0033	51	电池电流	2Bytes	SHORT	A	0.1	
0x0034	52	预留53	2Bytes	SHORT	/	1	
0x0035	53	电池温度	2Bytes	SHORT	C	0.1	
0x0036	54	电池剩余放电时间	2Bytes	SHORT	Min	1	
0x0037	55	电池容量	2Bytes	SHORT	%	0.1	
0x0038	56	逆变电流	2Bytes	SHORT	A	0.1	
0x0039	57	整流温度	2Bytes	SHORT	C	0.1	
0x003A	58	逆变温度	2Bytes	SHORT	C	0.1	
0x003B	59	设备运行天数	2Bytes	SHORT	/	1	
0x003C	60	预留61	2Bytes	SHORT	/	1	
0x003D	61	预留62	2Bytes	SHORT	/	1	
0x003E	62	预留63	2Bytes	SHORT	/	1	
0x003F	63	预留64	2Bytes	SHORT	/	1	
0x0040	64	预留65	2Bytes	SHORT	/	1	
0x0041	65	预留66	2Bytes	SHORT	/	1	
0x0042	66	预留67	2Bytes	SHORT	/	1	
0x0043	67	软件版本号1-2	2Bytes	Char	/	1	

0x0044	68	软件版本号3-4	2Bytes	Char	/	1	
0x0045	69	软件版本号5-6	2Bytes	Char	/	1	
0x0046	70	状态字	2Bytes	USHORT	/	1	<p>bit1~bit2 整流状态 0: 关闭 1: 软起 s 2: PFC模式 3: 电池模式 bit3~bit4 逆变状态 0: 关闭 1: 软起 2: 正常运行 3: 预留 bit5~bit7 电池状态 0: 电池未连接 1: 电池待机 (不充电不放电) 2: 电池均充 3: 电池浮充 4: 电池放电 bit8~bit9 旁路状态 0: 无旁路 1: 旁路正常 bit10~bit11 输出供电状态 0: 无输出 1: 旁路输出 2: 逆变输出 3: 它机输出 (并机) bit12~bit16 预留</p>
0x0047	71	系统工作模式	2Bytes	USHORT	/	1	<p>0: 故障清除模式 1: 上电模式 2: 待机模式 3: 旁路模式 4: 市电模式 5: 电池模式 battery mode</p>

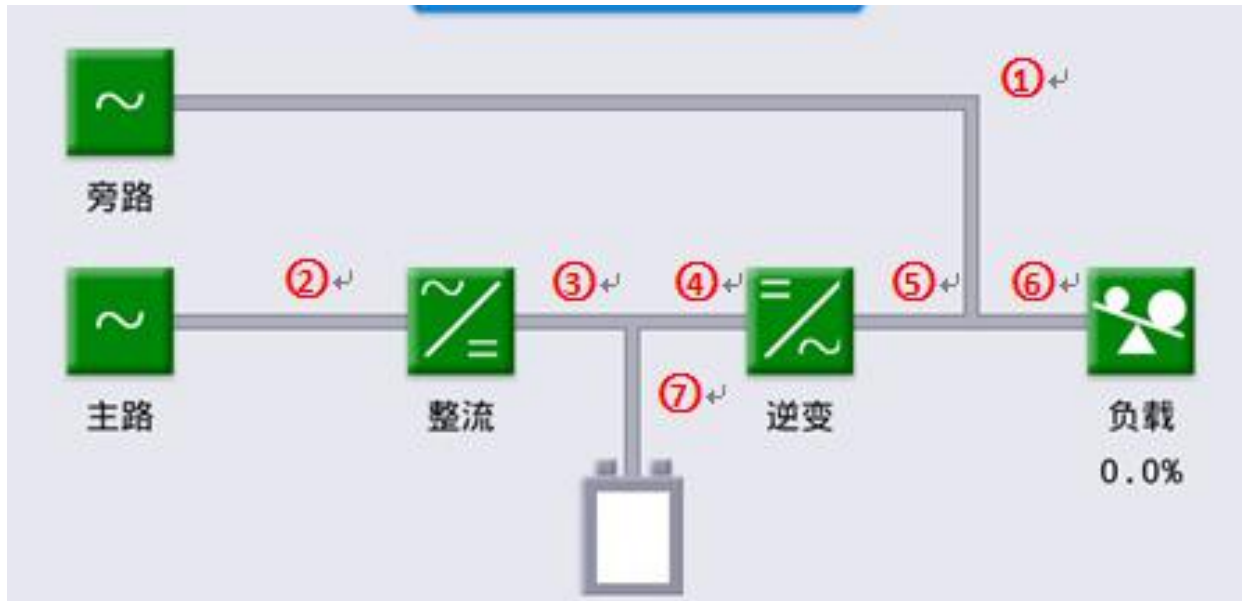
							6: 电池自检模式 7: 故障模式 8: 变频模式 9: ECO模式 10: 关机模式 11: 测试模式 test mode
0x0048	72	能流线状态寄存器1	2Bytes	SHORT	/	1	附图1 bit0~bit1 能流线1状态 0: 无能流 1: 能流向右 其他: 预留 bit2~bit3 能流线2状态 0: 无能流 1: 能流向右 其他: 预留 bit4~bit5 能流线3状态 0: 无能流 1: 能流向右 其他: 预留 bit6~bit7 能流线4状态 0: 无能流 1: 能流向右 其他: 预留 bit8~bit9 能流线5状态 0: 无能流 1: 能流向右 其他: 预留 bit10~bit11 能流线6状态 0: 无能流 1: 能流向右 其他: 预留 bit12~bit13 能流线7状态 0: 无能流 1: 能流向下 2: 能流向上 3: 能流无方向

							(电池不放电也不充电) bit14~bit15 预留
0x0049	73	能流线状态寄存器2	2Bytes	SHORT	/	1	预留
0x004A	74	预留73	2Bytes	SHORT	/	1	
0x004B	75	预留74	2Bytes	SHORT	/	1	
0x004C	76	预留75	2Bytes	SHORT	/	1	
0x004D	77	预留76	2Bytes	SHORT	/	1	
0x004E	78	预留77	2Bytes	SHORT	/	1	
0x004F	79	预留78	2Bytes	SHORT	/	1	
0x0050	80	预留79	2Bytes	SHORT	/	1	
0x0051	81	预留80	2Bytes	SHORT	/	1	
0x0052	82	预留81	2Bytes	SHORT	/	1	
0x0053	83	预留82	2Bytes	SHORT	/	1	
0x0054	84	预留83	2Bytes	SHORT	/	1	
0x0055	85	预留84	2Bytes	SHORT	/	1	
0x0056	86	预留85	2Bytes	SHORT	/	1	
0x8000	32768	MODbus Decode Version	2Bytes	USHORT	/	1	0xAAAA(固定值, 短信报警器识别标志, 外置为0xAAAA, 内置为0)
0x8001	32769	设备类型标识符1	2Bytes	USHORT	/	1	设备类型分类第1级①, 例如001
0x8002	32770	设备类型标识符2	2Bytes	USHORT	/	1	设备类型分类第2级②, 例如003
0x8003	32771	设备类型标识符3	2Bytes	USHORT	/	1	设备类型分类第3级③, 例如005
0x8004	32772	设备名称1-2	2Bytes	USHORT	/	1	按小端模式从低到高存储, 例如名称" EAST易事特", 寄存器0x8004值0x4145, 寄存器0x8005值0x5453, 寄存

							器0x8006值 0xD7D2, 寄存器0x8007值 0xC2CA, 寄存器0x8008值 0xD8CC
0x8005	32773	设备名称3-4	2Bytes	USHORT	/	1	
0x8006	32774	设备名称5-6	2Bytes	USHORT	/	1	
0x8007	32775	设备名称7-8	2Bytes	USHORT	/	1	
0x8008	32776	设备名称9-10	2Bytes	USHORT	/	1	
0x8009	32777	设备名称11-12	2Bytes	USHORT	/	1	
0x800A	32778	设备名称13-14	2Bytes	USHORT	/	1	
0x800B	32779	设备名称15-16	2Bytes	USHORT	/	1	
0x800C	32780	设备型号1-2	2Bytes	USHORT	/	1	按小端模式从低到高存储
0x800D	32781	设备型号3-4	2Bytes	USHORT	/	1	
0x800E	32782	设备型号5-6	2Bytes	USHORT	/	1	
0x800F	32783	设备型号7-8	2Bytes	USHORT	/	1	
0x8010	32784	设备型号9-10	2Bytes	USHORT	/	1	
0x8011	32785	设备型号11-12	2Bytes	USHORT	/	1	
0x8012	32786	设备型号13-14	2Bytes	USHORT	/	1	
0x8013	32787	设备型号15-16	2Bytes	USHORT	/	1	
0x8014	32788	设备厂家1-2	2Bytes	USHORT	/	1	按小端模式从低到高存储
0x8015	32789	设备厂家3-4	2Bytes	USHORT	/	1	
0x8016	32790	设备厂家5-6	2Bytes	USHORT	/	1	
0x8017	32791	设备厂家7-8	2Bytes	USHORT	/	1	
0x8018	32792	设备厂家9-10	2Bytes	USHORT	/	1	
0x8019	32793	设备厂家11-12	2Bytes	USHORT	/	1	
0x801A	32794	设备厂家13-14	2Bytes	USHORT	/	1	
0x801B	32795	设备厂家15-16	2Bytes	USHORT	/	1	
0x801C	32796	程序版本/固件版本	2Bytes	USHORT	/	1	例如传输值0x0100,则版本号为256
0x801D	32797	软件版本号 char1	2Bytes	USHORT	/	1	
0x801E	32798	软件版本号 char2	2Bytes	USHORT	/	1	
0x801F	32799	软件版本号 char3	2Bytes	USHORT	/	1	
0x8020	32800	软件版本号	2Bytes	USHORT	/	1	

		char4					
0x8021	32801	软件5版本	2Bytes	USHORT	/	1	例如传输值 0x0100,则版本号 为256
0x8022	32802	软件6版本	2Bytes	USHORT	/	1	例如传输值 0x0100,则版本号 为256
0x8023	32803	软件7版本	2Bytes	USHORT	/	1	例如传输值 0x0100,则版本号 为256
0x8024	32804	软件8版本	2Bytes	USHORT	/	1	例如传输值 0x0100,则版本号 为256
0x8025	32805	序列号char1、 char2	2Bytes	USHORT	/	1	bit1-bit8: char1 bit9-bit16: char2
0x8026	32806	序列号char3、 char4	2Bytes	USHORT	/	1	bit1-bit8: char3 bit9-bit16: char4
0x8027	32807	序列号char5、 char6	2Bytes	USHORT	/	1	bit1-bit8: char5 bit9-bit16: char6
0x8028	32808	序列号char7、 char8	2Bytes	USHORT	/	1	bit1-bit8: char7 bit9-bit16: char8
0x8029	32809	序列号char9、 char10	2Bytes	USHORT	/	1	bit1-bit8: char9 bit9-bit16: char10
0x802A	32810	序列号char11、 char12	2Bytes	USHORT	/	1	bit1-bit8: char11 bit9-bit16: char12
0x802B	32811	序列号char13、 char14	2Bytes	USHORT	/	1	bit1-bit8: char13 bit9-bit16: char14
0x802C	32812	序列号char15、 char16	2Bytes	USHORT	/	1	bit1-bit8: char15 bit9-bit16: char16

附图 1. 能流图界面及能流线编号



2、读离散量（功能码0x02）

02 功能码只支持连续 8 的倍数个寄存器读取，开始地址也必须时 8 的倍数，如从地址 0 连续读取 8、16、24.....个寄存器，或者从地址 16 连续读取 8、16、24.....个寄存器等，建议一次性读完，从地址 0 连续读取 96 个寄存器；

地址		告警/故障	数据长度	说明
HEX	DEC			
0x0000	0	BUS高压	1 bit	1: 发生; 0: 消失
0x0001	1	BUS低压	1 bit	1: 发生; 0: 消失
0x0002	2	BUS不平衡	1 bit	1: 发生; 0: 消失
0x0003	3	BUS软启动失败	1 bit	1: 发生; 0: 消失
0x0004	4	逆变软启动失败 fail	1 bit	1: 发生; 0: 消失
0x0005	5	逆变高压	1 bit	1: 发生; 0: 消失
0x0006	6	逆变低压	1 bit	1: 发生; 0: 消失
0x0007	7	逆变禁止	1 bit	1: 发生; 0: 消失
0x0008	8	过温	1 bit	1: 发生; 0: 消失
0x0009	9	输出短路	1 bit	1: 发生; 0: 消失
0x000A	10	过载故障	1 bit	1: 发生; 0: 消失
0x000B	11	负功异常 N	1 bit	1: 发生; 0: 消失
0x000C	12	关机故障	1 bit	1: 发生; 0: 消失
0x000D	13	并机软件版本不一致(未使用)	1 bit	1: 发生; 0: 消失
0x000E	14	同步信号异常(未使用)	1 bit	1: 发生; 0: 消失
0x000F	15	同步脉冲异常	1 bit	1: 发生; 0: 消失
0x0010	16	逆变继电器粘死	1 bit	1: 发生; 0: 消失
0x0011	17	BUS短路	1 bit	1: 发生; 0: 消失
0x0012	18	CAN通讯异常	1 bit	1: 发生; 0: 消失
0x0013	19	物理地址冲突	1 bit	1: 发生; 0: 消失
0x0014	20	并机机型不兼容	1 bit	1: 发生; 0: 消失
0x0015	21	电池节数设置错误	1 bit	1: 发生; 0: 消失
0x0016	22	输入SCR软启动失败	1 bit	1: 发生; 0: 消失
0x0017	23	整流SCR故障(未使用)	1 bit	1: 发生; 0: 消失
0x0018	24	逆变器故障	1 bit	1: 发生; 0: 消失
0x0019	25	并机旁路接线错误	1 bit	1: 发生; 0: 消失
0x001A	26	逆变开路	1 bit	1: 发生; 0: 消失
0x001B	27	PFC故障	1 bit	1: 发生; 0: 消失
0x001C	28	逆变电容异常	1 bit	1: 发生; 0: 消失
0x001D	29	风扇故障	1 bit	1: 发生; 0: 消失
0x001E	30	EPO	1 bit	1: 发生; 0: 消失
0x001F	31	辅助电源异常	1 bit	1: 发生; 0: 消失

0x0020	32	预留1	1 bit	1: 发生; 0: 消失
0x0021	33	预留2	1 bit	1: 发生; 0: 消失
0x0022	34	预留3	1 bit	1: 发生; 0: 消失
0x0023	35	预留4	1 bit	1: 发生; 0: 消失
0x0024	36	预留5	1 bit	1: 发生; 0: 消失
0x0025	37	预留6	1 bit	1: 发生; 0: 消失
0x0026	38	预留7	1 bit	1: 发生; 0: 消失
0x0027	39	预留8	1 bit	1: 发生; 0: 消失
0x0028	40	预留9	1 bit	1: 发生; 0: 消失
0x0029	41	预留10	1 bit	1: 发生; 0: 消失
0x002A	42	预留11	1 bit	1: 发生; 0: 消失
0x002B	43	预留12	1 bit	1: 发生; 0: 消失
0x002C	44	预留13	1 bit	1: 发生; 0: 消失
0x002D	45	预留14	1 bit	1: 发生; 0: 消失
0x002E	46	预留15	1 bit	1: 发生; 0: 消失
0x002F	47	预留16	1 bit	1: 发生; 0: 消失
0x0030	48	并机线连接异常error	1 bit	1: 发生; 0: 消失
0x0031	49	并机地址冲突 conflict	1 bit	1: 发生; 0: 消失
0x0032	50	电池反接	1 bit	1: 发生; 0: 消失
0x0033	51	过载告警	1 bit	1: 发生; 0: 消失
0x0034	52	电池未接	1 bit	1: 发生; 0: 消失
0x0035	53	输入过流	1 bit	1: 发生; 0: 消失
0x0036	54	电池电压高	1 bit	1: 发生; 0: 消失
0x0037	55	开机异常	1 bit	1: 发生; 0: 消失
0x0038	56	充电器故障	1 bit	1: 发生; 0: 消失
0x0039	57	EEPROM异常	1 bit	1: 发生; 0: 消失
0x003A	58	输入过流超时	1 bit	1: 发生; 0: 消失
0x003B	59	电池低压	1 bit	1: 发生; 0: 消失
0x003C	60	UPS维护时间到	1 bit	1: 发生; 0: 消失
0x003D	61	同步脉冲异常loss	1 bit	1: 发生; 0: 消失
0x003E	62	同步信号异常	1 bit	1: 发生; 0: 消失
0x003F	63	CAN通讯异常	1 bit	1: 发生; 0: 消失
0x0040	64	旁路高压	1 bit	1: 发生; 0: 消失
0x0041	65	旁路故障	1 bit	1: 发生; 0: 消失
0x0042	66	市电高压	1 bit	1: 发生; 0: 消失
0x0043	67	市电频率异常	1 bit	1: 发生; 0: 消失
0x0044	68	旁路频率超跟踪	1 bit	1: 发生; 0: 消失
0x0045	69	旁路逆变切换次数超限	1 bit	1: 发生; 0: 消失
0x0046	70	电池放电终止	1 bit	1: 发生; 0: 消失
0x0047	71	电池测试成功	1 bit	1: 发生; 0: 消失

0x0048	72	禁止开机	1 bit	1: 发生; 0: 消失
0x0049	73	电池测试未通过	1 bit	1: 发生; 0: 消失
0x004A	74	并机均流异常	1 bit	1: 发生; 0: 消失
0x004B	75	逆变锁相异常	1 bit	1: 发生; 0: 消失
0x004C	76	电池维护成功	1 bit	1: 发生; 0: 消失
0x004D	77	电池维护异常	1 bit	1: 发生; 0: 消失
0x004E	78	输入电流不平衡	1 bit	1: 发生; 0: 消失
0x004F	79	系统初始化错误	1 bit	1: 发生; 0: 消失
0x0050	80	PFC异常	1 bit	1: 发生; 0: 消失
0x0051	81	母线低压切电池次数超限	1 bit	1: 发生; 0: 消失
0x0052	82	市电异常	1 bit	1: 发生; 0: 消失
0x0053	83	旁路异常	1 bit	1: 发生; 0: 消失
0x0054	84	旁路频率异常	1 bit	1: 发生; 0: 消失
0x0055	85	逆变输出电压检测不一致	1 bit	1: 发生; 0: 消失
0x0056	86	电池电压异常	1 bit	1: 发生; 0: 消失
0x0057	87	旁路输出电压检测不一致	1 bit	1: 发生; 0: 消失
0x0058	88	逆变电压有效值异常	1 bit	1: 发生; 0: 消失
0x0059	89	维修旁路使能	1 bit	1: 发生; 0: 消失
0x005A	90	过温频繁	1 bit	1: 发生; 0: 消失
0x005B	91	并机电池节数设置不一致	1 bit	1: 发生; 0: 消失
0x005C	92	电池温度补偿异常(未使用)	1 bit	1: 发生; 0: 消失
0x005D	93	旁路电压不稳定(未使用)	1 bit	1: 发生; 0: 消失
0x005E	94	测试模式	1 bit	1: 发生; 0: 消失
0x005F	95	过载频繁	1 bit	1: 发生; 0: 消失

3、读保持寄存器（功能码 0x03）

地址		数据内容	数据长度/格式	类型	单位	系数	备注
HEX	DEC						
0x0000	0	系统工作模式	2bytes	SHORT	/	1	1: 单机 2: 并机 4: ECO 6: 并机ECO 8: 自老化
0x0001	1	并机数量	2bytes	SHORT	/	1	取值范围: 0-10
0x0002	2	设备地址	2bytes	SHORT	/	1	取值范围: 0-10
0x0003	3	输出电压调整	2bytes	SHORT	V	1	范围:额定电压正负10V以内
0x0004	4	频率跟踪限制	2bytes	SHORT	Hz	0.1	范围:0.5-5.0 Hz
0x0005	5	电池节数	2bytes	SHORT	/	1	范围: 铅酸电池 16,18,20,22,24, 锂电池 140-180的偶数
0x0006	6	电池AH	2bytes	SHORT	AH	1	范围:1-30000
0x0007	7	单体浮充电压	2bytes	SHORT	V	0.01	范围跟电池类型有关 铅酸电池:2.10-2.35 锂电池:3.30-4.15
0x0008	8	单体均充电压	2bytes	SHORT	V	0.01	范围跟电池类型有关 铅酸电池:2.20-2.45 锂电池:3.30-4.15
0x0009	9	单体Eod电压0.6c	2bytes	SHORT	V	0.01	范围跟电池类型有关 铅酸电池:1.6-1.85 锂电池:2.60-2.80
0x000A	10	单体Eod电压0.15c	2bytes	SHORT	V	0.01	范围跟电池类型有关 铅酸电池:1.65-1.90 锂电池:2.65-2.85
0x000B	11	充电电流限制	2bytes	SHORT	A	0.01	范围0.5、1.0、2.0、3.0、 4.0、5.0、6.0、7.0、8.0、 9.0、10.0、11.0、12.0
0x000C	12	均充时间限制	2bytes	SHORT	小时	1	范围: 1-48
0x000D	13	均充周期	2bytes	SHORT	小时	1	范围: 1-3000
0x000E	14	电池自动维护周期	2bytes	SHORT	小时	1	范围: 720-30000
0x000F	15	额定容量	2bytes	SHORT	VA	1	范围: 6000、10000、 20000
0x0010	16	使用容量	2bytes	SHORT	VA	1	范围: <=额定容量
0x0011	17	额定输入电压	2bytes	SHORT	V	1	范围: 100、110、120、

							125、200、208、220、230、240
0x0012	18	额定输入频率	2bytes	SHORT	Hz	1	范围：50、60
0x0013	19	额定输出电压	2bytes	SHORT	V	1	范围：100、110、120、125、200、208、220、230、240
0x0014	20	额定输出频率	2bytes	SHORT	Hz	1	范围：50、60
0x0015	21	系统码1	2bytes	SHORT	/	1	<p>Bit1降额使能 1: 允许降额, 0: 禁止降额</p> <p>Bit2单单/三单机型选择 11/31(1) 1: 三单, 0: 单单</p> <p>Bit3电池自动均充使能 (1: 允许自动均充, 0: 禁止自动均充)</p> <p>Bit4电池自动维护使能1: 允许自动维护, 0: 禁止自动维护</p> <p>Bit5额定频率自适应使能 1: 使能, 0: 禁止</p> <p>Bit6: 母线过压允许重启 1: 允许, 0: 禁止</p> <p>Bit7: 风扇故障蜂鸣器告警 1: 允许, 0: 禁止</p> <p>Bit8: 缺相运行 (三单机型使用) 1: 允许, 0: 禁止</p> <p>Bit9: 变频模式 1: 使能, 0: 禁止</p> <p>Bit10: 降低输出功率因数 1: 0.9, 0: 1</p> <p>Bit10 - Bit16: 预留</p> <p>Bit11: LCD显示容量 1: 使能, 0: 禁止</p> <p>Bit12: LM接反告警 1: 禁止, 0: 允许</p> <p>Bit13上电自启动 1: 禁止, 0: 允许</p> <p>Bit14 - Bit16: 预留</p>
0x0016	22	系统码2	2bytes	SHORT	/	1	<p>Bit1-Bit3 旁路电压上限 (研发/用户) 000:+15% 001:+10% 010:+20% 011:+25%</p>

							<p>Bit4-Bit6 旁路电压下限 (研发/用户) 000: -20% 001: -10% 010: -15% 011: -30% 100: -40%</p> <p>Bit7-Bit8 旁路频率范围 (研发/用户) 00: +-5Hz 01: +-1Hz 10: +-3Hz 11: 无限制</p> <p>Bit9-Bit11 自老化电流百分比 (研发) 000: 30% 001: 40% 010: 50% 011: 60% 100: 70% 101: 80% 110: 90% 111: 100%</p> <p>Bit12-Bit16 预留</p>
0x0017	23	电池低压点倍数	2bytes	SHORT	/	0.01	范围: 1.05-1.25 (如倍数为1.05, 则电池电压低判定点电压为: 1.05*EOD点电压)
0x0018	24	用户设置码	2bytes	SHORT	/	1	<p>Bit1切逆变使能1: 锁旁路超时时允许切逆变, 0: 锁旁路超时时禁止切逆变</p> <p>Bit2禁止旁路 1: 禁止旁路工作, 0: 允许旁路工作</p> <p>Bit3EPO功能使能 1: 允许使用EPO, 0: 禁止EPO功能</p> <p>Bit4输入反相运行 1: 允许, 0: 禁止</p> <p>Bit5输出接变压器 1: 已接, 0: 未接</p> <p>Bit6: 输入接发电机 1: 已接, 0: 未接</p> <p>Bit7: 电池类型 1: 锂电池, 0: 铅酸电池</p> <p>Bit8-Bit16 预留</p>
0x0019	25	显示屏背光灯点亮时间	2bytes	SHORT	Min	1	范围: 1-30
0x001A	26	电池维护周期	2bytes	SHORT	天	1	范围: 0-3000

0x001B	27	预留0	2bytes	SHORT		1	范围:
0x001C	28	逆变电压值校正	2bytes	SHORT	V	0.1	范围: 实测逆变电压有效值, 读取此寄存器时, 为逆变电压有效值采样
0x001D	29	预留1	2bytes	SHORT		1	范围:
0x001E	30	预留2	2bytes	SHORT		1	范围:
0x001F	31	旁路电压值校正	2bytes	SHORT	V	0.1	范围: 实测旁路电压有效值, 读取此寄存器时, 为旁路电压有效值采样
0x0020	32	预留3	2bytes	SHORT		1	范围:
0x0021	33	预留4	2bytes	SHORT		1	范围:
0x0022	34	逆变直流量校正	2bytes	SHORT	V	0.01	范围: 实测逆变电压直流量, 只写入, 读取无效
0x0023	35	预留5	2bytes	SHORT		1	范围:
0x0024	36	预留6	2bytes	SHORT		1	范围:
0x0025	37	输入电压A相值校正	2bytes	SHORT	V	0.1	范围: 实测输入电压A相有效值, 读取此寄存器时, 为输入A相电压有效值采样
0x0026	38	输入电压B相值校正	2bytes	SHORT	V	0.1	范围: 实测输入电压B相有效值, 读取此寄存器时, 为B相输入电压有效值采样
0x0027	39	输入电压C相值校正	2bytes	SHORT	V	0.1	范围: 实测输入电压C相有效值, 读取此寄存器时, 为C相输入电压有效值采样
0x0028	40	电池电压值校正	2bytes	SHORT	V	0.1	范围: 实测电池电压值
0x0029	41	预留7	2bytes	SHORT		1	
0x002A	42	输入带变压器电压校正1	2bytes	SHORT	V	0.1	范围: 输入带变压器实测10%负载时变压器原边电压有效值, 只写入, 读取无效
0x002B	43	输入带变压器电压校正2	2bytes	SHORT	V	0.1	范围: 输入带变压器实测90%负载时变压器原边电压有效值, 只写入, 读取无效
0x002C	44	输出带变压器电压校正1	2bytes	SHORT	V	0.1	范围: 输出带变压器实测10%负载时变压器副边电压有效值, 只写入, 读取无效
0x002D	45	输出带变压器电压	2bytes	SHORT	V	0.1	范围: 输出带变压器实测

		校正2					90%负载时变压器副边电压有效值，只写入，读取无效
0x002E	46	SNMP卡槽通讯协议设置	2bytes	SHORT			0:modbus Ascii 1:Modbus Rtu 2:MegaTec
0x002F	47	变压器线圈扎数比	2bytes	SHORT	1	0.1	输出端变压器匝数比
0x0030	48	预留	2bytes	SHORT			
0x0031	49	ups维护周期	2bytes	SHORT	天	1	
0x0032	50	设备型号word1	2bytes	SHORT		1	bit1-bit8: char1 bit9-bit16: char2
0x0033	51	设备型号word2	2bytes	SHORT		1	bit1-bit8: char3 bit9-bit16: char4
0x0034	52	设备型号word3	2bytes	SHORT		1	bit1-bit8: char5 bit9-bit16: char6
0x0035	53	设备型号word4	2bytes	SHORT		1	bit1-bit8: char7 bit9-bit16: char8
0x0036	54	设备型号word5	2bytes	SHORT		1	bit1-bit8: char9 bit9-bit16: char10
0x0037	55	公司名字word1	2bytes	SHORT		1	bit1-bit8: char1 bit9-bit16: char2
0x0038	56	公司名字word2	2bytes	SHORT		1	bit1-bit8: char3 bit9-bit16: char4
0x0039	57	公司名字word3	2bytes	SHORT		1	bit1-bit8: char5 bit9-bit16: char6
0x003A	58	公司名字word4	2bytes	SHORT		1	bit1-bit8: char7 bit9-bit16: char8
0x003B	59	公司名字word5	2bytes	SHORT		1	bit1-bit8: char9 bit9-bit16: char10
0x003C	60	公司名字word6	2bytes	SHORT		1	bit1-bit8: char11 bit9-bit16: char12
0x003D	61	公司名字word7	2bytes	SHORT		1	bit1-bit8: char13 bit9-bit16: char14
0x003E	62	预留9	2bytes	SHORT			
0x003F	63	预留10	2bytes	SHORT			
0x0040	64	预留11	2bytes	SHORT			
0x0041	65	预留12	2bytes	SHORT			
0x0042	66	输入电压高告警点	2bytes	SHORT	V	1	范围：220-300
0x0043	67	输入电压低告警点	2bytes	SHORT	V	1	范围：500-700
0x0044	68	输入频率高告警点	2bytes	SHORT	Hz	0.1	范围：300-600
0x0045	69	输入频率低告警点	2bytes	SHORT	Hz	0.1	范围：720-300
0x0046	70	输出电压高告警点	2bytes	SHORT	V	1	范围：220-300
0x0047	71	输出电压低告警点	2bytes	SHORT	V	1	范围：100-220

0x0048	72	输入电压高保护点	2bytes	SHORT	V	1	范围：220-300
0x0049	73	输入电压低保护点	2bytes	SHORT	V	1	范围：100-220
0x004A	74	输出电压高保护点	2bytes	SHORT	V	1	范围：220-300
0x004B	75	输出电压低保护点	2bytes	SHORT	V	1	范围：100-220
0x004C	76	序列号char1、char2	2bytes	char	/		bit1-bit8: char1 bit9-bit16: char2
0x004D	77	序列号char3、char4	2bytes	char	/		bit1-bit8: char3 bit9-bit16: char4
0x004E	78	序列号char5、char6	2bytes	char	/		bit1-bit8: char5 bit9-bit16: char6
0x004F	79	序列号char7、char8	2bytes	char	/		bit1-bit8: char7 bit9-bit16: char8
0x0050	80	序列号char9、char10	2bytes	char	/		bit1-bit8: char9 bit9-bit16: char10
0x0051	81	序列号char11、char12	2bytes	char	/		bit1-bit8: char11 bit9-bit16: char12
0x0052	82	序列号char13、char14	2bytes	char	/		bit1-bit8: char13 bit9-bit16: char14
0x0053	83	序列号char15、char16	2bytes	char	/		bit1-bit8: char15 bit9-bit16: char16
0x0054	84	预留13	2bytes	SHORT			
0x0055	85	预留14	2bytes	SHORT			
0x0056	86	预留15	2bytes	SHORT			
0x0057	87	预留16	2bytes	SHORT			
0x0058	88	预留13	2bytes	SHORT			
0x0059	89	预留14	2bytes	SHORT			
0x8000	32768	清除故障	2Bytes	SHORT	/	1	1: 清除故障, 0: 忽略
0x8001	32769	清除历史记录	2Bytes	SHORT	/	1	1: 清除历史记录, 0: 忽略
0x8002	32770	蜂鸣器消音	2Bytes	SHORT	/	1	1: 蜂鸣器消音, 0: 蜂鸣器正常工作
0x8003	32771	手动切旁路	2Bytes	SHORT	/	1	1: 切旁路(逆变供电状态下且旁路正常时有效), 0: 忽略
0x8004	32772	手动切逆变	2Bytes	SHORT	/	1	1: 切逆变(手动切至旁路供电后, 逆变正常时有效), 0: 忽略
0x8005	32773	程序升级	2Bytes	SHORT	/	1	1: 升级程序, 0: 忽略
0x8006	32774	电池测试	2Bytes	SHORT	/	1	测试类型: 1: 电池检测(放电20s) 2: 电池维护(放电至电池

							低压)
0x8007	32775	终止电池测试	2Bytes	SHORT	/	1	1: 终止电池自检, 0: 忽略
0x8008	32776	开关机	2Bytes	SHORT	/	1	只写。 1: 关机/关机断输出; 2: 关机转旁路; 3: 开机。
0x8009	32777	设置使能密码 (ups识别到正确的 设置使能密码后, 才可对设置寄存器 进行写入操作, 读 取不受影响)	2Bytes	SHORT	/	1	密码计算方式为: type 识别3个寄存器地址对应的数值累加和 + 当前设备内序列号(16个字符) ASCII 值累加和的总和(Uint16)作为”使能码“ eg. type_1 =2 type_2=2 type_3=1 sn:123456789ABCDEF 使能码 = 2+2+1+ (0x30 +0x31 +0x32 +0x33 + ...+0x56)
0x800A	32778	预留3	2Bytes	SHORT	/	1	
0x800B	32779	预留4	2Bytes	SHORT	/	1	
0x800C	32780	预留5	2Bytes	SHORT	/	1	
0x800D	32781	预留6	2Bytes	SHORT	/	1	
0x800E	32782	预留7	2Byte	SHORT	/	1	

4、预置寄存器列表（功能码 0x06, 0x10）

（寄存器定义与0x03功能码一致）

5、历史记录寄存器列表（功能码 0x41, EA900 G4 6-20K 专用）

寄存器地址	数据名称	数据长度	数据类型	单位	系数	备注
0	第1条历史记录	2Bytes	SHORT	/	1	逐条查询，每次查询一条历史记录，寄存器起始地址只能为5的整数倍（一条记录占用5个地址），根据返回的历史记录调数决定时候继续查询
5	第2条历史记录	2Bytes	SHORT	/	1	
10	第3条历史记录	2Bytes	SHORT	/	1	
15	……	2Bytes	SHORT	/	1	
20	……	2Bytes	SHORT	/	1	

三、通信内容

1、读输入寄存器（功能码0x04）

【举例】

假设UPS 设备地址设置为0x18，查询寄存器起始地址值为0x0010，寄存器个数为2 个，即查询“R 相输出电流”和“S 相输出电流”的值；假设此时“R 相输出电流”的值为89.2A，“S 相输出电流”的值为88.9A，根据该值的系数为0.1，那么：

寄存器0x0010 的值为： $(892)_D = (037C)_H$

寄存器0x0011 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	0x04	0x00	0x10	0x00	0x02	0x72	0x07
字节数	1	1	2		2		2	

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	0x04	0x037C	0x0379	0x73	0xCB
字节数	1	1	1	4		2	

2、读离散量（功能码0x02）

【举例】

假设UPS 设备地址设置为0x18，查询寄存器起始地址值为51，即0x0033，寄存器个数为1个，即查询“UPS 过载状态”；假设此时“UPS 已过载”；即该值为1。返回数据时，在该字节中由低位向高位排列，直至8个位为止。下一个字节中的8个输入位也是从低位到高位排列。若返回的输入位数不是8的倍数，则在最后的数据字节中的剩余位直至字节的最高位全部填零。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	起始地址		离散量个数		CRC 校验	
数据	0x18	0x02	0x00	0x33	0x00	0x01	0x4B	0xCC
字节数	1	1	2		2		2	

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	0x02	0x01	0x01	0x67	0x14
字节数	1	1	1	1	2	

3、读取设备寄存器（功能码 0x03）

假设 UPS 设备地址设置为 0x18，请求读取寄存器 43-44 的内容：

寄存器 0x0043 的值为：(541)D = (021D)H

寄存器 0x0044 的值为：(309)D = (0135)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	0x00	0x43	0x00	0x02	0x37	0xD6
字节数	1	1	2		2		2	

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	0x021D	0x0135	0x22	0xCB
字节数	1	1	1	4		2	

4、预置寄存器（功能码 0x06,0x10）

【举例】

假设UPS 设备地址设置为0x18，预置寄存器起始地址值为1，寄存器个数为1 个，即电池测试10S。寄存器内容被预置后返回正常响应；预置单个寄存器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		预置数据		CRC 校验	
数据	0x18	0x06	0x00	0x01	0xFF	0xFF	0xDB	0xB3
字节数	1	1	2		2		2	

响应帧信息为：

	地址	功能码	寄存器地址		预置成功的数据		CRC 校验	
数据	0x18	0x06	0x00	0x01	0xFF	0xFF	0xDB	0xB3
字节数	1	1	2		2		2	

5、历史记录寄存器列表（功能码 0x41，EA900 G4 6-20K 专用）

RTU 模式请求帧信息

	地址	功能码	寄存器起始地址		寄存器个数		CRC 校验	
数据	0x00	0x41	0x0*	0x**	0x00	0x05	0XX	0XX
字节数	1	1	2		2		2	

响应帧信息为：

	地址	功能码	寄存器地址	历史记录数据				CRC 校验	
数据	0x**	0x41	0x0A	word1	word2	word3	word1	word1	0xDB 0xB3
字节数	1	1	1	10				2	

历史记录数据内容格式

字	word1	word2	word3	word4	word5
含义	历史记录 ID (bit1-bit8: ID; bit9-bit12: 记录产生时的系统工作模式; bit13-bit14: 历史记录状态, 1: 发生; 0: 消失, bit15-bit16: 预留)	历史记录产生的时间 (天)	历史记录产生的时间 (秒, 低 16 位)	历史记录产生的时间 (秒, 高 16 位)	当前历史记录的条数

附录A CRC 校验

CRC 循环冗余校验

循环冗余校验 (CRC) 域为两个字节，包含一个二进制 16 位值。附加在报文后面的 CRC 的值由发送设备计算。接收设备在接收报文时重新计算 CRC 的值，并将计算结果于实际接收到的 CRC 值相比较。如果两个值不相等，则为错误。

CRC 的计算, 开始对一个 16 位寄存器预装全 1. 然后将报文中的连续的 8 位子节对其进行后续的计算。只有字符中的 8 个数据位参与生成 CRC 的运算，起始位，停止位和校验位不参与 CRC 计算。CRC 的生成过程中，每个 8-位字符与寄存器中的值异或。然后结果向最低有效位 (LSB) 方向移动(Shift) 1 位，而最高有效位 (MSB) 位置充零。然后提取并检查 LSB：如果 LSB 为 1，则寄存器中的值与一个固定的预置值异或；如果 LSB 为 0，则不进行异或操作。这个过程将重复直到执行完 8 次移位。完成最后一次（第 8 次）移位及相关操作后，下一个 8 位字节与寄存器的当前值异或，然后又同上面描述过的一样重复 8 次。当所有报文中子节都运算之后得到的寄存器中的最终值，就是 CRC。

生成 CRC 的过程为:

1. 将一个 16 位寄存器装入十六进制 FFFF (全 1). 将之称作 CRC 寄存器.
2. 将报文的第一个 8 位字节与 16 位 CRC 寄存器的低字节异或，结果置于 CRC 寄存器.
3. 将 CRC 寄存器右移 1 位 (向 LSB 方向)，MSB 充零. 提取并检测 LSB.
4. (如果 LSB 为 0): 重复步骤 3 (另一次移位).
- (如果 LSB 为 1): 对 CRC 寄存器异或多项式值 0xA001 (1010 0000 0000 0001).
5. 重复步骤 3 和 4，直到完成 8 次移位。当做完此操作后，将完成对 8 位字节的完整操作。
6. 对报文中的下一个字节重复步骤 2 到 5，继续此操作直至所有报文被处理完毕。
7. CRC 寄存器中的最终内容为 CRC 值.
8. 当放置 CRC 值于报文时，如下面描述的那样，高低字节必须交换。

将 CRC 放置于报文当 16 位 CRC (2 个 8 位字节) 在报文中传送时，低位字节首先发送，然后是高位字节。

例:执行 CRC 生成的 C 语言的函数在下面示出。所有的可能的 CRC 值都被预装在两个数组中，当计算报文内容时可以简单的索引即可。一个数组含有 16 位 CRC 域的所有 256 个可能的高位字节，另一个数组含有低位字节的值。这种索引访问 CRC 的方式提供了比对报文缓冲区的每个新字

符都计算新的 CRC 更快的方法。

注意: 此函数内部执行高/低 CRC 字节的交换。此函数返回的是已经经过交换的 CRC 值。

也就是说, 从该函数返回的 CRC 值可以直接放置于报文用于发送。

函数使用两个参数:

unsigned char *puchMsg; 指向含有用于生成 CRC 的二进制数据报文缓冲区的指针

unsigned short usDataLen; 报文缓冲区的字节数.

CRC 生成函数

unsigned short CRC16 (puchMsg, usDataLen) /* 函数以 unsigned short 类型返回 CRC */

unsigned char *puchMsg ; /* 用于计算 CRC 的报文 */

unsigned short usDataLen ; /* 报文中的字节数 */

```
{
    unsigned char uchCRCHi = 0xFF ; /* CRC 的高字节初始化 */
    unsigned char uchCRCLo = 0xFF ; /* CRC 的低字节初始化 */
    unsigned uIndex ;          /* CRC 查询表索引 */
    while (usDataLen--)      /* 完成整个报文缓冲区 */
    {
        uIndex = uchCRCLo ^ *puchMsgg++ ; /* 计算 CRC */
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex] ;
        uchCRCHi = auchCRCLo[uIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}
```

附录B 高低位字节表

高字节表

/* 高位字节的 CRC 值 */

```
static unsigned char 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, 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, 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
};

```

低字节表

/* 低位字节的 CRC 值 */

```

static char 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
};