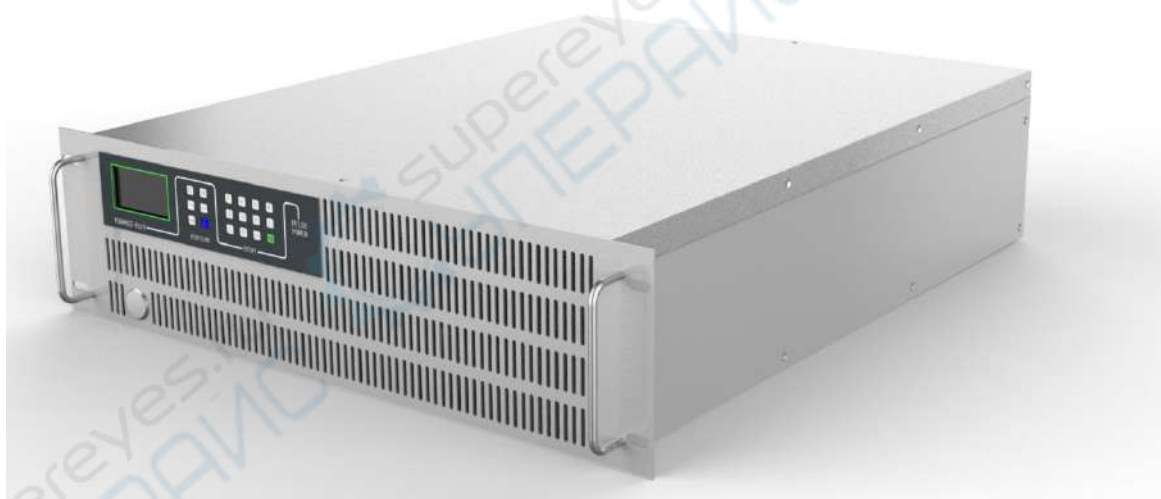


# ModbusRTU 通信协议



版本 V1.0 2019

# 目录

1 简介.....	3
1.1 范围.....	3
1.2 术语说明.....	3
1.3 参考文献.....	3
2 通信协议.....	3
2.1 硬件电气接口.....	3
2.2 传输模式.....	4
2.3 数据帧格式.....	4
2.3.1 地址（Address）域.....	4
2.3.2 功能码（Function Code）域.....	5
2.3.3 数据（Data）域.....	5
2.3.4 错误校验（Check）域.....	5
3 通信命令详解.....	6



# 1 简介

## 1.1 范围

本协议完全符合Modbus协议标准规范，适用于电源与远端监控系统之间的通信。若用户需要通过远端监控来获取电源的运行状态和实时参数，就必须严格遵循此协议。

## 1.2 术语说明

RTU：Remote Terminal Unites，远端终端单元。

## 1.3 参考文献

GBT 19582.1-2008 基于Modbus协议的工业自动化网络规范

GBT 19582.2-2008 基于Modbus协议的工业自动化网络规范

# 2 通信协议

Modbus协议在一根通信线上使用主从应答连接（半双工），这意味着在一根单独的通信线上信号沿着相反的两个方向传输。首先，主机的信号寻址到一台唯一的终端设备(从机)，然后终端设备发出应答信号并以相反的方向传输给主机。

此协议采用Modbus RTU通信传输方式，下面简单说明协议各层的数据格式。

## 2.1 硬件电气接口

TIA/EIA - 485 – A

DB9针型插足	引脚	信号	描述
	1	A	485-A
	2	B	485-B
	3~9	N.C.	未使用

DB9端口定义

## 2.2 传输模式

Modbus数据链路层支持RTU 和ASCII两种传输模式。这里采用RTU传输模式。

RTU传输模式中每个字节的格式为：

编码系统： 8bits 二进制系统

每个字节的位：8个数据位，首先发送最低有效位

1个奇偶校验位或无校验

1个停止位或2个停止位(无校验)

本产品默认设置为8个数据位，无奇偶校验位，1个停止位，波特率为9600bps。

## 2.3 数据帧格式

表2-1，显示数据帧格式

表2-1 数据帧格式

<i>Address</i>	<i>Function Code</i>	<i>Data</i>	<i>Check</i>
8 bits	8 bits	N*8 bits	16 its

【注】本产品可通过上位机修改波特率与通讯地址，具体详询我司售后服务人员。

### 2.3.1 地址（Address）域

地址在数据帧的开始部分，由一个字节8 bits（0～255）组成，这些位标明了用户指定的终端设备的地址，该设备将接收来自与之相连的主机发送的数据。每个终端设备的地址必须是唯一的，仅仅被寻址到的终端会响应包含了该地址的查询。当终端发送回一个响应，响应中的从机地址数据便告诉了主机哪台终端正与之进行通信。

【注】本产品默认地址为：01H。

### 2.3.2 功能码（Function Code）域

功能域代码告诉了被寻址到的终端执行何种功能。表 2-2 列出本设备支持的功能码，以及它们的内容和意义。

表2-2 功能码

<i>Function Code (Dec)</i>	<i>Function Code (Hex)</i>	意义
03	0x03	读取保持寄存器
04	0x04	读取输入寄存器
16	0x10	连续写多个寄存器
	0x83/84/90	错误功能码

错误功能码为相应功能码+0x80，例如表中的：0x03 对应 0x83；0x06 对应 0x86。

### 2.3.3 数据（Data）域

数据域包含了终端执行特定功能所需要的数据或者终端响应查询时采集到的数据。这些数据的内容可能是数值、参考地址或者设置值。例如：功能码告诉终端读取一个寄存器，数据域则需要指明从哪个寄存器开始及读取多少个数据。

### 2.3.4 错误校验（Check）域

该域允许主机和终端检查传输过程中的错误。有时，由于电噪声和其它干扰，一组数据在从一个设备传输到另一个设备时在线路上可能会发生一些改变，错误校验能够保证主机或者终端不去响应那些传输过程中发生了改变的数据，这就提高了系统的安全性和效率，错误校验使用了16 位循环冗余的方法（CRC16）。低字节先发，高字节后发。

表2-3 错误校验

<i>Check</i>	
Low 8bits	High 8bits

【注】发送序列总是相同的— 地址、功能码、数据和错误校验。

### 3 通信命令详解

#### 1) 功能码 0x03, 读取保持寄存器

地址与数据对应表:

寄存器地址(Hex)	名称	请求字数(Hex)	回复字节数(Hex)	单位
0x0040	电压设置值	0x0002	0x04	0.01V
0x0041	电流设置值	0x0002	0x04	0.01A
0x0042	机器状态	0x0002	0x04	0-关机; 1-开机
0x0043	频率设置值	0x0002	0x04	Hz
0x0044	占空比设置值	0x0002	0x04	%

举例说明:

##### ◆ 读电压设置值:

主机发送: 01 03 00 40 00 02 C5 DF

从机回复: 01 03 04 00 00 3A 98 E9 39 (当前设置值 150V, 单位 0.01V)

##### ◆ 读电流设置值:

主机发送: 01 03 00 41 00 02 94 1F

从机回复: 01 03 04 00 00 03 E8 FA 8D (当前设置值 10A, 单位 0.01A)

##### ◆ 读机器状态:

主机发送: 01 03 00 42 00 02 64 1F

从机回复: 01 03 04 00 00 00 01 3B F3 (当前机器处于开机状态)

##### ◆ 读频率设置值:

主机发送: 01 03 00 43 00 02 35 DF

从机回复: 01 03 04 00 01 86 A0 C9 EB (当前设置值 100000Hz)

##### ◆ 读占空比设置值:

主机发送: 01 03 00 44 00 02 84 1E

从机回复: 01 03 04 00 00 00 32 7B E6 (当前设置值 50%)

#### 2) 功能码 0x04, 读取输入寄存器

地址与数据对应表:

寄存器地址(Hex)	名称	请求字数(Hex)	回复字节数(Hex)	单位
0x0000	输出电压值	0x0001	0x02	0.01V
0x0001	输出电流值	0x0001	0x02	0.01A

举例说明:

##### ◆ 读输出电压值:

主机发送: 01 04 00 00 00 01 31 CA

从机回复: 01 04 02 8C 98 DC 5A (当前输出电压值 359.92V, 单位 0.01V)

##### ◆ 读输出电流值:

主机发送: 01 04 00 01 00 01 60 0A

从机回复: 01 04 02 03 35 79 D7 (当前输出电流值 8.21A, 单位 0.01A)

### 3) 功能码 0x10, 连续写多个寄存器

地址与数据对应表:

寄存器地址(Hex)	名称	下发字数(Hex)	下发字节数(Hex)	单位
0x0040	电压设置值	0x0002	0x04	0.01V
0x0041	电流设置值	0x0002	0x04	0.01A
0x0042	机器状态	0x0002	0x04	0-关机; 1-开机
0x0043	频率设置值	0x0002	0x04	Hz
0x0044	占空比设置值	0x0002	0x04	%

举例说明:

◆ 远程设置输出电压值:

主机发送: 01 10 00 40 00 02 04 00 00 4E 20 C3 E7 (远程设置电压 200V, 单位 0.01V)

从机回复: 01 10 00 40 00 02 40 1C

◆ 远程设置输出电流值:

主机发送: 01 10 00 41 00 02 04 00 00 04 B0 35 27 (远程设置电流 12A, 单位 0.01A)

从机回复: 01 10 00 41 00 02 11 DC

◆ 远程开关机:

主机发送: 01 10 00 42 00 02 04 00 00 00 00 76 46 (远程关机)

从机回复: 01 10 00 42 00 02 E1 DC

◆ 远程设置频率值:

主机发送: 01 10 00 43 00 02 04 00 01 5F 90 DF D6 (远程设置频率 90000Hz)

从机回复: 01 10 00 43 00 02 B0 1C

◆ 远程设置占空比值:

主机发送: 01 10 00 44 00 02 04 00 00 00 3C F6 7D (远程设置占空比 60%)

从机回复: 01 10 00 44 00 02 01 DD