**电源通讯协议（Modbus RTU）**

**V1.00**

# 1．字节格式

字节传输时包含1个起始位，8个数据位，无校验位，1个停止位。字节传输时序如下表所示。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 起始位 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | 停止位 |

# 2．数据传送方式

|  |  |  |  |
| --- | --- | --- | --- |
| 数据类型 | 寄存器数 | 字节数 | 说 明 |
| 字节数据 | 1 | 1 |  |
| 整形数据 | 1 | 2 | 高字节在前，低字节在后 |
| 长整形数 | 2 | 4 | 高字在前，低字在后 |
| 浮点数据 |

# 3．帧格式

## 3.1读取保持寄存器（功能码 03H）

### 3.1.1上位机发送的帧格式

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 通讯地址（1-127） |
| 2 | 03H | 功能码 |
| 3 | 起始寄存器地址高字节 | 寄存器起始地址 |
| 4 | 起始寄存器地址低字节 |
| 5 | 寄存器个数高字节 | 连续寄存器个数 |
| 6 | 寄存器个数低字节 |
| 7 | CRC16低字节 | CRC 校验数据 |
| 8 | CRC16高字节 |

### 3.1.2正常应答帧格式

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 通讯地址（1-127） |
| 2 | 03H | 功能码 |
| 3 | 数据域字节数(N) |  |
| 4 | 字节1 |  |
| **……** | …… |  |
|  | 字节N |  |
| N+4 | CRC16低字节 |  |
| N+5 | CRC16高字节 |  |

### 3.1.3异常应答

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 仪表的通讯地址（1-127） |
| 2 | 83H | 功能码 |
| 3 |  | 错误代码  01H:寄存器地址或个数错误；  02H:数据错误；  05H:IO错误； |
| 4 | CRC16低字节 |  |
| 5 | CRC16高字节 |  |

## 3.2 连续写保持寄存器（功能码 10H）

### 3.2 .1上位机发送的帧格式

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 通讯地址（1-127） |
| 2 | 10H | 功能码 |
| 3 | 寄存器起始地址高字节 | 起始寄存器地址 |
| 4 | 寄存器起始地址低字节 |
| **5** | 寄存器个数高字节 | 寄存器个数 |
| 6 | 寄存器个数低字节 |
| 7 | 字节数（M） |  |
| 8 | 字节1 |  |
|  | 字节2 |
|  | 字节3 |
|  | …… |
|  | 字节N |  |
| M+8 | CRC16低字节 | CRC校验数据 |
| M+9 | CRC16高字节 |

### 3.2.2正常应答

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 通讯地址（1-127） |
| 2 | 10H | 功能码 |
| 3 | 起始地址高字节 | 寄存器起始地址 |
| 4 | 起始地址低字节 |
| 5 | 寄存器个数高字节 |  |
| 6 | 寄存器个数低字节 |
| 7 | CRC16低字节 | CRC校验数据 |
| 8 | CRC16高字节 |

### 3.2.3异常应答

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 仪表地址 | 仪表的通讯地址（1-255之间） |
| 2 | 90H | 功能码 |
| 3 |  | 错误代码  01H:寄存器地址或个数错误；  02H:数据错误；  05H:IO错误； |
| 4 | CRC16低字节 |  |
| 5 | CRC16高字节 |  |

## 3.3读取只读寄存器（功能码 04H）

### 3.3.1上位机发送的帧格式

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 通讯地址（1-127） |
| 2 | 04H | 功能码 |
| 3 | 起始寄存器地址高字节 | 寄存器起始地址 |
| 4 | 起始寄存器地址低字节 |
| 5 | 寄存器个数高字节 | 连续寄存器个数 |
| 6 | 寄存器个数低字节 |
| 7 | CRC16低字节 | CRC 校验数据 |
| 8 | CRC16高字节 |

### 3.3.2正常应答帧格式

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 通讯地址（1-127） |
| 2 | 04H | 功能码 |
| 3 | 数据域字节数(N) |  |
| 4 | 字节1 |  |
| **……** | …… |  |
|  | 字节N |  |
| N+4 | CRC16低字节 |  |
| N+5 | CRC16高字节 |  |

### 3.3.3异常应答

|  |  |  |
| --- | --- | --- |
| 顺序 | 代 码 | 说 明 |
| 1 | 地址 | 仪表的通讯地址（1-127） |
| 2 | 84H | 功能码 |
| 3 |  | 错误代码  01H:寄存器地址或个数错误；  02H:数据错误；  05H:IO错误； |
| 4 | CRC16低字节 |  |
| 5 | CRC16高字节 |  |

# 4．通讯波特率

可以在电源菜单中设置通讯波特率，具体方法请阅读电源手册相关篇章。

|  |  |
| --- | --- |
|  | 可选波特率 |
| 1 | 2400bps |
| 2 | 4800bps |
| 3 | 9600bps |
| 4 | 19200bps |
| 5 | 38400bps |

# 5．仪表地址

仪表地址即通讯地址，可以在电源菜单中进行设置，具体方法见说明书相关篇章。

# 6．通讯功能码

10H: 写设置参数、控制电源输出停止

03H: 读设置参数

04H: 读测量数据、状态

# 7．通讯数据CRC校验

static const unsigned char aucCRCHi[] = {

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, 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

};

static const unsigned char aucCRCLo[] = {

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

};

Unsigned short

usMBCRC16( unsigned char \* pucFrame, unsigned short usLen )

{

unsigned char ucCRCHi = 0xFF;

unsigned char ucCRCLo = 0xFF;

int iIndex;

while( usLen-- )

{

iIndex = ucCRCLo ^ \*( pucFrame++ );

ucCRCLo = ( unsigned char )( ucCRCHi ^ aucCRCHi[iIndex] );

ucCRCHi = aucCRCLo[iIndex];

}

return ( unsigned short )( ucCRCHi << 8 | ucCRCLo );

}

# 8．数据寄存器地址

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 寄存器地址（十进制） | 说明 | 单位 | 数据范围 | 功能码 | 数据格式 |
| 41001 | 电压设置 | 电压值\*1000 |  | 10H | S32 |
| 41003 | 电流设置 | 电流值\*1000 |  | S32 |
| 41007 | 频率设置 | 频率值\*1000 |  | S32 |
| 41009 | 档位设置 | N/A | 0:高档  1:低档 | U16 |
| 46001 | 输出 | N/A | 任意值 | 10H | U16 |
| 46002 | 停止 | N/A | 任意值 | U16 |
|  | | | | | |
| 36000 | 测量VRMS值 | 电压值/1000 | S32 | 04H | S32 |
| 36002 | 测量IRMS值 | 电流值/1000 | S32 | S32 |
| 36004 | 测量有功功率值 | 功率值/1000 | S32 | S32 |
| 36006 | 测量PF值 | PF值/1000 | S32 | S32 |
| 36008 | 测量视在功率值 | 功率值/1000 | S32 | S32 |
| 36010 | 测量VPEAK值 | VPEAK值/1000 | S32 | S32 |
| 36012 | 测量IPEAK值 | IPEAK值/1000 | S32 | S32 |
| 35999 | 状态字 | N/A | 见附录 | U16 |

注：

S32 有符号长整形

U16 无符号短整形

**附录1 状态字说明**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BIT15~BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| 保留 | 0 | 0 | 1:有报警  0:无报警 | 1:输出  0:停止 |