

双向 PCS 设备 RS-485 通信协议

协议修订记录

版本	日期	创建人/修改人	修改说明	修改原因
V7	2023.01..06		增加查询版本信息命令	
V6	2022.12.28		增加有功功率上报	
V5	2022.08.15		修改交流电流上报精度，增加风扇转速上报，增加过载状态上报	
V4	2022.07.22		仅保留自动旁路功能相关描述	
V3	2022.07.04		增加旁路功能	
V2	2021.07.19		调整容量计算最小单位为 5mAh/bit	
V1	2020.03.27		1.增加 0x07 充放电功率设置请求命令 2.增加 0x08 充放电功率设置响应命令 3.命令 06 增加 07~10 字节-- PCS 单次充电/放电累计毫安时	
V0	2020.12.14		建档	

一、协议简介:

本协议通过 RS485 通信接口与主机进行通信连接和命令交互，采用主从应答的方式实现多机互联通信的功能，在进行设置操作时，由主机向指定从机发出数据设置请求，将主机发送的数据设置到指定的从机中，在非广播模式下进行设置操作，从机将会将设置是否出错的状态结果回复给主机。若在广播模式下进行设置操作，从机不会向主机回复数据；在进行查询操作时，由主机向指定从机发出数据查询请求，从机收到请求后将特定的数据回复给主机。

二、通信格式和速率:

- 开始位: 1
- 数据位: 8
- 奇偶校验位: 无
- 停止位: 1

通信波特率为 9600Bps，主机两次发送请求数据帧的最小时间间隔不得小于 25ms。

三、通信地址:

通信地址的有效范围从 0 ~ 31，每个模块设置为不同的地址，其中地址 0 作为广播地址保留，非广播模式下的通信地址范围为 1 ~ 31，其他数值为无效地址，不能进行正常数据通信和交互。

注：下述所做的协议阐述若无特殊指明都是针对非广播模式的。

四、协议框架:

● 数据帧格式

	数据内容	字节序号	备注
帧头	0x7E	Byte1	固定不变
命令代码	内容详见协议文本	Byte2	
地址	1 ~ 31	Byte3	
数据	Data1 ~ DataN	Byte4 ~ ByteN	若数据占双字节，高字节在前，低字节在后
CRC 校验	CRC16	ByteN+1 ~ ByteN+2	高字节在前，低字节在后，帧头帧尾和转义数据不参与计算
帧尾	0x7E	ByteN+3	固定不变

● 协议数据转义处理

数据帧中定义 0x7E 为帧头和帧尾的标识符，为了不让数据帧中的其他数据与头和尾标识因相同而产生冲突，所以规定在数据帧中(除了头尾字节外)出现 0x7E 时，需要对数据进行转义处理，具体的处理方式为：将 0x7E 转换为 0x7D 和 0x5E(其中 0x7D 为填充字符，0x5E 为 0x7E 的 bit5 位取反)；在数据帧中出现 0x7D 时，将 0x7D 转换为 0x7D 和 0x5D(其中 0x7D 为填充字符，0x5D 为 0x7D 的 bit5 位取反)。

例如，下面数据帧中除帧头帧尾外的数据中出现了 0x7E 或者 0x7D，需要对下面的数据帧进行转义处理：

0x7E	0xAB	0x7E	0x12	0x7D	0x59	CRC16_H(0x7E)	CRC16_L(0x7D)	0x7E
------	------	------	------	------	------	---------------	---------------	------

将上述数据帧进行转义后为：

0x7E	0xAB	0x7D	0x5E	0x12	0x7D	0x5D	0x59	0x7D	CRC16_H(0x5E)
0x7D	CRC16_L(0x5D)	0x7E							

五、协议内容：

为了突出协议数据区的内容，以下协议文本的内容中不包含帧头，CRC16 和帧尾的数据信息。

● 数据设置请求命令

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0x01	由主机发送至从机
2	地址	1	1~31	
3	充放电模式选择	1	0x00: 待机 (默认) 0x01 ~ 0x04: 保留 0x05: 自动旁路模式	待机模式(0x00): PCS 不执行任何操作; 自动旁路(0x05): 用自动模式数据查询命令(0x09)可查询状态;
4	PCS 通道选择	1	0x01: CH1 0x02: CH2	单输入输出选择 0x01
5	整流输出电压高字节	1	0.1V/Bit	例: Vout=560=0x0230, 对应的整流输出电压为 56vdc
6	整流输出电压低字节	1		
7	最高充电电流高字节	1	0.1A/Bit	例: Iout=200=0x00C8, 对应的充电电流为 20A。
8	最高充电电流低字节	1		
9	PCS 开启或关闭 AC 旁路端口输出	1	0x01: 开启 0x02: 关闭	处于自动旁路模式(0x05)时, 0x01 表示开启旁路输出 (有市电接入时, 优先旁路输出市电; 无市电接入或者市电不正常时, 开启逆变器输出) 0x02 表示关闭旁路/逆变器输出
10	UPS 电压设置	1	1V/Bit	支持 100V/50Hz,60Hz, // 110V/60Hz, // 120V/60Hz, // 220V/50Hz, // 230V/50Hz, // 240V/60Hz。 0 代表采取上一次发给 PCS 的设定值
11	UPS 频率设置	1	1Hz/Bit	

● 数据设置应答命令

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0x02	由从机回复至主机
2	地址	1	1~31	
3	操作状态码	1	0x00: 设置完成 0x01: 参数越界 0x02: 操作受限或当前条件下不支持此操作 0x03: 其他错误	
4	保留	1		

● 自动模式数据查询请求命令

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0x09	由主机发送至从机
2	地址	1	1~31	
3	保留	1		
4	保留	1		

● 自动模式数据查询应答命令

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0x0A	由从机回复至主机
2	地址	1	1~31	
3	PCS 状态信息	1	状态位定义详见下表	
4	PCS 充放电状态	1	0x01 ~ 0x06: 保留 0x07: 待机 0x08: 整流不带旁路 0x09: 整流带旁路 0x0A: 逆变输出 0x0B: 自动模式就绪	
5	PCS 直流侧电压高字节	1	0.1V/Bit	例: $V_{out} = 560 = 0x0230$, 对应电压为 56Vdc
6	PCS 直流侧电压低字节	1		
7	PCS 直流侧电流高字节	1	0.1A/Bit	例: $I_{out} = 200 = 0x00C8$, 对应电流为 20A
8	PCS 直流侧电流低字节	1		
9	AC 交流输入电压高字节	1	0.1V/Bit	例: $V_{out} = 2200 = 0x0898$, 对应电压为 220Vac
10	AC 交流输入电压低字节	1		
11	PCS 逆变输出电压高字节	1	0.1V/Bit	例: $V_{out} = 2200 = 0x0898$, 对应电压为 220Vac
12	PCS 逆变输出电压低字节	1		
13	PCS 输入交流电流高字节	1	0.01A/Bit	例: $I_{in} = 2000 = 0x07D0$, 对应电流为 20Aac
14	PCS 输入交流电流低字节	1		
15	PCS 逆变/旁路输出交流电流高字节	1	0.01A/Bit	例: $I_{in} = 2000 = 0x07D0$, 对应电流为 20Aac
16	PCS 逆变/旁路输出交流电流低字节	1		
17	PCS 风扇转速 (1~255)	1	0.392%/Bit	例: 60% Speed = 153 = 0x99, 100% Speed = 255 = 0xFF
18	保留	1		

● 自动模式数据查询请求命令(新版)

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0x0D	由主机发送至从机
2	地址	1	1~31	
3	保留	1		
4	保留	1		

● 自动模式数据查询应答命令(新版)

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0x0E	由从机回复至主机
2	地址	1	1~31	
3	PCS 状态信息	1	状态位定义详见下表	
4	PCS 充放电状态	1	0x01 ~ 0x06: 保留 0x07: 待机 0x08: 整流不带旁路 0x09: 整流带旁路 0x0A: 逆变输出 0x0B: 自动模式就绪	
5	PCS 直流侧电压高字节	1	0.1V/Bit	例: $V_{out} = 560 = 0x0230$, 对应电压为 56Vdc
6	PCS 直流侧电压低字节	1		
7	PCS 直流侧电流高字节	1	0.1A/Bit	例: $I_{out} = 200 = 0x00C8$, 对应电流为 20A
8	PCS 直流侧电流低字节	1		
9	AC 交流输入电压高字节	1	0.1V/Bit	例: $V_{out} = 2200 = 0x0898$, 对应电压为 220Vac
10	AC 交流输入电压低字节	1		
11	PCS 逆变输出电压高字节	1	0.1V/Bit	例: $V_{out} = 2200 = 0x0898$, 对应电压为 220Vac
12	PCS 逆变输出电压低字节	1		
13	PCS 输入交流电流高字节	1	0.01A/Bit	例: $I_{in} = 2000 = 0x07D0$, 对应电流为 20Aac
14	PCS 输入交流电流低字节	1		
15	PCS 逆变/旁路输出交流电流高字节	1	0.01A/Bit	例: $I_{in} = 2000 = 0x07D0$, 对应电流为 20Aac
16	PCS 逆变/旁路输出交流电流低字节	1		
17	PCS 风扇转速 (1~255)	1	0.392%/Bit	例: 60% Speed = 153 = 0x99, 100% Speed = 255 = 0xFF
18	AC 有功功率上报高字节	1	1W/Bit	例: 1500=0x05DC, 对应的有功功率为 1500W
19	AC 有功功率上报低字节	1		
20	保留	1		

21	保留	1		
22	保留	1		
23	保留	1		
24	保留	1		
25	保留	1		

● PCS 状态信息位定义

PCS 状态信息位	含义	描述
Bit0	硬件故障	0: 正常 1: 硬件故障
Bit1	PCS 过温故障	0: 正常 1: PCS 温度过高保护
Bit2	AC 输入电压状态	0: AC 输入电压正常 1: 无 AC 电压 或者 AC 过/欠压
Bit3	PCS 逆变输出过载状态	0: PCS 逆变输出正常 1: PCS 逆变输出过载, 进入限功率输出状态
Bit4	保留	
Bit5	保留	
Bit6	保留	
Bit7	保留	

● 版本信息查询请求命令

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0xF0	由主机发送至从机
2	地址	1	1~31	
3	保留	1		
4	保留	1		

● 版本信息查询应答命令

字节序号	数据项	长度 (Byte)	内容	描述
1	命令码	1	0xF1	由从机回复至主机
2	地址	1	1~31	
3	DC 产品型号	1	产品型号及年份	例: 0x15F3, 其中 15 表示 1500W 机型, F 对应 G 系列型号, 3 表示 2023 年发布的软件版本
4		1		
5	DC 版本日期	1	日期版次	例: 0xC31A, 其中 C 表示 12 月份, 31 表示 31 日, A 表示 A 版本 (版次), 即 12 月 31 日 A 版本
6		1		
7	PFC 产品型号	1	产品型号及年份	例: 0x15F3, 其中 15 表示 1500W 机型, F 对应 G 系列型号, 3 表示 2023 年发布的软件版本
8		1		
9	PFC 版本日期	1	日期版次	例: 0x104A, 其中 1 表示 1 月份, 04 表示 4 日, A 表示 A 版本 (版次), 即 1 月 4 日 A 版本
10		1		

六、CRC 数据表格

为了统一通信双方计算 CRC16 校验数据，可直接应用下面的函数得到 CRC16 的计算结果，若计算完整一帧数据的 CRC16 的初始参数 usCRCinit 传入 0 即可，若对一个数据帧分多次计算 CRC16 时，计算首个字节时传入 0，后续计算依次将前一次计算结果作为参数传入直到一帧数据计算完成。

```
unsigned short CRC16_CCITT(Uchar *puchMsg, unsigned short usCRCinit, unsigned short usDataLen)
{
    unsigned short wCRCin = usCRCinit, i = 0;
    unsigned short wCPoly = 0x1021;
    unsigned char wchar = 0;

    while(usDataLen--)
    {
        wchar = *(puchMsg++);
        wCRCin ^= (wchar << 8);
        for(i = 0; i < 8; i++)
        {
            if(wCRCin & 0x8000)
                wCRCin = (wCRCin << 1) ^ wCPoly;
            else
                wCRCin = wCRCin << 1;
        }
    }
    return wCRCin;
}
```

或者结合查表的方式计算得到 CRC16:

```
const unsigned short CRC16Table[256] = {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7, //注意本行的 0x1021
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
    0xdbfd, 0xcdbc, 0xfbff, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
```

```

0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};

```

利用下面的函数查表计算完整一帧数据的 CRC16 的初始参数 usCRCInit 传入 0 即可：

```

unsigned short CRC_16( unsigned char * pucData, unsigned short usCRCInit, unsigned short usSize)
{
    unsigned short nAccum = usCRCInit, i = 0;

    for ( i = 0; i < usSize; i++ )
        nAccum = ( nAccum << 8 ) ^ ( unsigned short )CRC16Table[( nAccum >> 8 ) ^ *pucData++];

    return nAccum;
}

```

七、示例说明

● 自动旁路输出开启：

7E	01	01	05	01	01	E0	00	64	01	DC	32	56	02	7E
帧头	命令	地址	模式	通道	整流输出电压高	整流输出电压低	整流输出电流高	整流输出电流低	旁路打开	UPS输出电压	UPS频率	CRC16高字节	CRC16低字节	帧尾

以上数据表明：地址为 01，通道为 01，自动旁路模式，直流输出电压为 48V，电流设置 10A，旁路输出打开，UPS 输出电压为 220Vac，频率为 50Hz；其中 CSH，CSL 分别表示校验数据的高，低字节

● 自动旁路输出关闭：

7E	01	01	05	01	01	E0	00	64	02	DC	32	0F	52	7E
帧头	命令	地址	模式	通道	整流输出电压高	整流输出电压低	整流输出电流高	整流输出电流低	旁路关闭	UPS输出电压	UPS频率	CRC16高字节	CRC16低字节	帧尾

以上数据表明：地址为 01，通道为 01，自动旁路模式，直流输出电压为 48V，电流设置 10A，旁路输出关闭，UPS 输出电压为 220Vac，频率为 50Hz；其中 CSH，CSL 分别表示校验数据的高，低字节

● 设置为待机模式：

7E	01	01	00	01	01	E0	00	64	02	00	36	31	7E
帧头	命令	地址	模式	通道	整流输出电压高	整流输出电压低	整流输出电流高	整流输出电流低	输出关闭	保留	CRC16高字节	CRC16低字节	帧尾

以上数据表明：地址为 01，通道为 01，待机模式，直流输出电压为 48V，电流设置 10A，输出关闭，其中 CSH，CSL 分别表示校验数据的高，低字节

CSH，CSL 校验数据可以根据上面所述的计算方法计算得到。初期为了调试方便可以用 0xAA 0x55 替代，在正式版本中要务必采用计算得到的 CRC 验证数据的正确性。

八、常见问题说明

● 当处于自动旁路模式时，电池充满如何停止充电？

1. 切换成待机模式：此种操作适合没有旁路输出的场合。
2. 将输出电压设置为低于当前电池电压：此种操作适合接入市电，并开启旁路功能的场合。