

MODBUS通信协议(2018年7月4日启用)

1 串行通信设置:

本协议采用的RS232 或 RS485异步通信的数据传送格式:

1 采用通信MODBUS的RTU方式

10 设置9600, n, 8, 1波特率9600, 无校验, 8位, 1位停止位, 可选波特率 (9600, 4800) 建议 (9600)。

11 校验方式采用CRC校验 (从地址字节开始到校验字节之前的字节)

2 发送的格式:

0x03、0x04命令应答

(1) 上位计算机:

地址字节~命令字节~开始地址高字节~开始地址低字节~读地址个数高字节~读地址个数低字节~校验低字节~校验高字节

0x01(地址可变)~0X03~0X00~0X00~0X00~0X06~CRCLO~CRCHI

0x01(地址可变)~0X04~0X00~0X00~0X00~0X06~CRCLO~CRCHI

上面的命令功能是从零号地址字开始读六个数据字。

支持的方式可以是:

开始地址字 (大于等于0, 小于等于5) 与地址个数字 (大于等于1, 小于等于6) 的和小于等于6。

开始地址字的内容为要读出的起始字在下位机的存储地址, 读地址个数字的内容为从开始地址字开始要连续读出字的个数。其具体的对应表格应根据具体的需要来确定。可以参见后面的设定表格。

数据高字节~数据低字节为要发送字数据的分离。采用了十六进制的数据。

字数据=高位字节*256+低位字节

请根据具体的设计要求发送正确范围的数据。

(2) 下位单片机返回的数据格式:

命令正确返回的数据格式

地址字节~命令字节~返回字节数 (字数乘2) ~开始数据高字节~开始数据低字节~高~低~...~高~低~校验低字节~校验高字节

0x01命令应答

(1) 上位计算机:

地址字节~命令字节~开始地址高字节~开始地址低字节~读地址个数高字节~读地址个数低字节~校验低字节~校验高字节

0x01(地址可变)~0X01~0X00~0X00~0X00~0X01~CRCLO~CRCHI

上面的命令功能是从零号地址字开始读1个数据字。

支持的方式可以是:

开始地址 (大于等于0, 小于等于5) 与地址个数 (大于等于1, 小于等于6) 的和小于等于6。

开始地址的内容为要读出的起始字在下位机的存储地址, 读地址个数的内容为从开始地址字开始要连续读出的个数。其具体的对应表格应根据具体的需要来确定。可以参见后面的设定表格。

数据高字节~数据低字节为要发送字数据的分离。采用了十六进制的数据。

字数据=高位字节*256+低位字节

请根据具体的设计要求发送正确范围的数据。

(2) 下位单片机返回的数据格式:

命令正确返回的数据格式

地址字节~命令字节~0x02~0X00~状态字节~校验低字节~校验高字节

错误命令应答

错误命令返回的数据格式

1 命令错 (非0X03或0x04命令): 地址字节~命令字节~0X01~校验低字节~校验高字节

- 2 功能错 (0X03或0x04命令但开始地址和要求的的数据长度错,CRC校验错): 地址字节~命令字节~0X02~校验低字节~校验高字节

3数据在下位机的存储地址

0x03/0x04命令

地址 (字)	内容	备注
0	状态字	高字节 (0x00) ~低字节 (0bxx 54 32 10)
1	历史最大温度值字	单位为0.1摄氏度
2	A相温度字	单位为0.1摄氏度
3	B相温度字	单位为0.1摄氏度
4	C相温度字	单位为0.1摄氏度
5	D相温度字	单位为0.1摄氏度

0x03命令~~地址5~~状态字低字节译码

地址 (位)	内容	备注
0	FJ	若置位1, 表示下位机已经启动风机。
1	CW	若置位1, 表示下位机已经超温。
2	GZ	若置位1, 表示下位机发生故障。
3	TZ	若置位1, 表示下位机已经跳闸。
4	FJ2	若置位1, 表示下位机已经启动风机2。
5	MLS	若置位1, 表示下位机已经启动门联锁继电器。

注, 假如值为0x8000表示该相温度超上限而开路;

其他情况正温度用正数表示, 负温度用补码表示。

0x01命令 (高字节 (0x00) 状态字低字节 (高位) xx 54 32 10 (低位))

地址 (位)	内容	备注
0	FJ	若置位1, 表示下位机已经启动风机。
1	CW	若置位1, 表示下位机已经超温。
2	GZ	若置位1, 表示下位机发生故障。
3	TZ	若置位1, 表示下位机已经跳闸。
4	FJ2	若置位1, 表示下位机已经启动风机2。
5	MLS	若置位1, 表示下位机已经启动门联锁继电器。

4. 静止时间要求

发送数据前要求数据总线静止时间即无数据发送时间大于 (5ms):近似为9600波特率发送五个字节的时间。

附件: CRC校验算法程序

```
function CalcCRC16(str: string): Word;
procedure CRC16(Data: Byte);
var
  i: Integer;
begin
  Result := Result xor Data;
  for i:=1 to 8 do
  begin
    if ((Result and 1)=1) then
      Result := (Result shr 1) XOR $A001
    else
      Result := Result shr 1;
    end;
  end;
end;
```

```
var
  i: Integer;
begin
  Result := $FFFF;
  for i:=1 to Length(str) do
    CRC16(Byte(str[i]));
end;
```