

# SIEMENS西门子青海省玉树藏族自治州（授权）一级代理商——西门子伺服电机西北总代理

产品名称	SIEMENS西门子青海省玉树藏族自治州（授权）一级代理商——西门子伺服电机西北总代理
公司名称	广东湘恒智能科技有限公司
价格	.00/件
规格参数	西门子总代理:PLC 西门子一级代:驱动 西门子代理商:伺服电机
公司地址	惠州大亚湾澳头石化大道中480号太东天地花园2栋二单元9层01号房
联系电话	15903418770 15915421161

## 产品详情

一、什么是MODBUS？1.基本概念MODBUS是MODICON公司（现为施耐德电气公司的一个品牌）最先倡导的一种软的通讯规约，经过大多数公司的实际应用，逐渐被认可成为一种标准的通讯规约，只要按照这种规约进行数据通讯或传输，不同的系统就可以通讯。目前，在RS232/RS485通讯过程中，更是广泛采用这种规约。常用的MODBUS通讯规约有二种，一种是MODBUS ASCII，一种是MODBUS RTU。一般来说，通讯数据量少而且主要是文本的通讯则采用MODBUS ASCII规约，通讯数据量大而且是二进制数值时，多采用MODBUS RTU规约。在实际的应用过程中，为了解决某一个特殊问题，人们喜欢自己修改MODBUS规约来满足自己的需要（事实上，人们经常使用自己定义的规约来通讯，这样能解决问题，但不太规范）。更为普通的用法是，少量修改规约，但将规约格式附在软件说明书一起，或直接放在帮助中，这样就方便了用户的通讯。2.MODBUS协议简述AC RXXE系列仪表使用的是MODBUS-RTU通讯协议，MODBUS协议详细定义了校验码、数据序列等，这些都是特定数据交换的必要内容。MODBUS协议在一根通讯线上使用主从应答式连接（半双工），这意味着在一根单独的通讯线上信号沿着相反的两个方向传输。首先，主计算机的信号寻址到一台唯一的终端设备（从机），然后，终端设备发出的应答信号以相反的方向传输给主机。MODBUS协议只允许在主机（PC，PLC等）和终端设备之间通讯，而不允许独立的终端设备之间的数据交换，这样各终端设备不会在它们初始化时占据通讯线路，而jinxian于响应到达本机的查询信号。3.查询—回应周期查询消息中的功能代码告之被选中的从设备要执行何种功能。数据段包含了从设备要执行功能的任何附加信息。例如功能代码03是要求从设备读保持寄存器并返回它们的内容。数据段必须包含要告之从设备的信息：从何寄存器开始读及要读的寄存器数量。错误检测域为从设备提供了一种验证消息内容是否正确的方法。回应如果从设备产生一正常的回应，在回应消息中的功能代码是在查询消息中的功能代码的回应。数据段包括了从设备收集的数据：如寄存器值或状态。如果有错误发生，功能代码将被修改以用于指出回应消息是错误的，同时数据段包含了描述此错误信息的代码。错误检测域允许主设备确认消息内容是否可用。4.传输方式传输方式是指一个数据帧内一系列独立的数据结构以及用于传输数据的有限规则，下面定义了与MODBUS协议—RTU方式相兼容的传输方式。每个字节的位：· 1个起始位 · 8个数据位，最小的有效位先发送 · 无奇偶校验位 · 1个停止位 错误检测(Error

checking) : CRC (循环冗余校验) 5.协议当数据帧到达终端设备时, 它通过一个简单的“端口”进入被寻址到的设备, 该设备去掉数据帧的“信封”(数据头), 读取数据, 如果没有错误, 就执行数据所请求的任务, 然后, 它将自己生成的数据加入到取得的“信封”中, 把数据帧返回给发送者。返回的响应数据中包含了以下内容: 终端从机地址(Address)、被执行了的命令(Function)、执行命令生成的被请求数据(Data)和一个校验码(Check)。发生任何错误都不会有成功的响应, 或者返回一个错误指示帧。6.数据帧格式AddressFunctionDataCheck8-Bits8-BitsN x

8-Bits16-Bits7.地址(Address)域地址域在帧的开始部分, 由一个字节(8位二进制码)组成, 十进制为0~255, 在我们的系统中只使用1~247, 其它地址保留。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。8.功能(Function)域功能域代码告诉了被寻址到的终端执行何种功能。下表列出了该系列仪表用到的功能码, 以及它们的意义和功能。代码意义行为03读数据寄存器获得一个或多个寄存器的当前二进制值16预置多寄存器设定二进制值到一系列多寄存器中(不对ACRXXE开放)9.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: 功能域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, 内嵌的地址和数据依照类型和从机之间的不同内容而有所不同。10.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, 出错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)。11.错误检测的方法错误校验(CRC)域占用两个字节, 包含了一个16位的二进制值。CRC值由传输设备计算出来, 然后附加到数据帧上, 接收设备在接收数据时重新计算CRC值, 然后与接收到的CRC域中的值进行比较, 如果这两个值不相等, 就发生了错误。CRC运算时, 首先将一个16位的寄存器预置为全1, 然后连续把数据帧中的每个字节中的8位与该寄存器的当前值进行运算, 仅仅每个字节的8个数据位参与生成CRC, 起始位和终止位以及可能使用的奇偶位都不影响CRC。在生成CRC时, 每个字节的8位与寄存器中的内容进行异或, 然后将结果向低位移位, 高位则用“0”补充, 最低位(LSB)移出并检测, 如果是1, 该寄存器就与一个预设的固定值(0A001H)进行一次异或运算, 如果最低位为0, 不作任何处理。上述处理重复进行, 直到执行完了8次移位操作, 当最后一位(第8位)移完以后, 下一个8位字节与寄存器的当前值进行异或运算, 同样进行上述的另一个8次移位异或操作, 当数据帧中的所有字节都作了处理, 生成的最终值就是CRC值。12.生成一个CRC的流程为

预置一个16位寄存器为0FFFFH(全1), 称之为CRC寄存器。

把数据帧中的第一个字节的8位与CRC寄存器中的低字节进行异或运算, 结果存回CRC寄存器。

将CRC寄存器向右移一位, 最高位填以0, 最低位移出并检测。如果最低位为0: 重复第三步(下一次移位); 如果最低位为1: 将CRC寄存器与一个预设的固定值(0A001H)进行异或运算。

重复第三步和第四步直到8次移位。这样处理完了一个完整的八位。

重复第2步到第5步来处理下一个八位, 直到所有的字节处理结束。最终CRC寄存器的值就是CRC的值

。此外还有一种利用预设的表格计算CRC的方法, 它的主要特点是计算速度快, 但是表格需要较大的存储空间, 该方法此处不再赘述, 请参阅相关资料。13.通讯应用格式详解本节所举实例将尽可能的使用如图

所示的格式, (数字为16进制)。AddrFunData start reg hiData start reg loData #of regs hiData #of regs

loCRC16 loCRC16hi01H03H00H00H00H03H05HCBHAddr: 从机地址Fun: 功能码Data start reg

hi: 数据起始地址 寄存器高字节Data start reg lo: 数据起始地址 寄存器低字节Data #of reg

hi: 数据读取个数 寄存器高字节Data #of reg lo: 数据读取个数 寄存器低字节CRC16 Hi: 循环冗余校验

高字节CRC16 Lo: 循环冗余校验 低字节 14.读数据(功能码03)查询数据帧此功能允许用户获得设备采集与记录的数据及系统参数。主机一次请求的数据个数没有限制, 但不能超出定义

的地址范围。下面的例子是从01号从机读3个采集到的基本数据(数据帧中每个地址占用2个字节)UA、UB、UC, 其中UA的地址为0025H, UB的地址为0026H, UC的地址为0027H。Addr FunData startAddr hiData startAddr

loData#ofregs hiData #ofregs loCRC16 loCRC16 hi01H03H00H25H00H03H14H00H响应数据帧响应包含从机

地址、功能码、数据的数量和CRC错误校验。下面的例子是读取UA、UB、UC

(UA=082CH, UB=082AH, UC=082CH)的响应。AddrFunByte countData1 hiData1 loData2 hiData2 loData3

hiData3 loCRC16 loCRC16 hi01H03H06H08H2CH08H2AH08H2CH94H4EH错误指示码如果主机请求的地址不存在则返回错误指示码: FFH。