SIEMENS核心供应商6SN1118-0DJ21-0AA2

产品名称	SIEMENS核心供应商6SN1118-0DJ21-0AA2
公司名称	湖南西控自动化设备有限公司
价格	.00/件
规格参数	西门子:西门子授权代理商 备件:核心供货商 德国:现货
公司地址	中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301
联系电话	17838383235 17838383235

产品详情

看大神讲解MODBUS标准协议，想不理解都难！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）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通讯规约有两种，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ACRXXXE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，而于响应到达本机的查询信号。3.查询—回应周期查询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，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ACRXXXE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，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，高位填以0，低位移出并检测。 如果低位为0：重复第三步（下一次移位）；如果低位为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。 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.通讯应用格式祥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 hiDatastartAddr 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。二、特点Modbus具有以下几个特点1、标准、开放，用户可以免费、放心地使用Modbus协议，不需要交纳许可证费，也不会侵犯知识产权。目前，支持Modbus的厂家超过400家，支持Modbus的产品超过600种。 2、Modbus可以支持多种电气接口，如RS-232、RS-485等，还可以

在各种介质上传送，如双绞线、光纤、无线等。 3、Modbus的帧格式简单、紧凑，通俗易懂。用户使用容易，厂商开发简单。三、功能码定义1.ModBus功能码01READ COIL STATUS02READ INPUT STATUS03READ HOLDING REGISTER04READ INPUT REGISTER05WRITE SINGLE COIL06WRITE SINGLE REGISTER15WRITE MULTIPLE COIL16WRITE MULTIPLE REGISTER四、传输方式在ModBus系统中有2种传输模式可选择。这2种传输模式与从机PC通信的能力是同等的。选择时应视所用ModBus主机而定，每个ModBus系统只能使用一种模式，不允许2种模式混用。一种模式是ASCII（美国信息交换码），另一种模式是RTU（远程 终端设备）。用户选择想要的模式，包括 串口通信参数（ 波特率、校验方式等），在配置每个控制器的时候，在一个Modbus网络上的所有设备都必须选择相同的传输模式和串口参数。所选的ASCII或RTU方式仅适用于标准的Modbus网络，它定义了在这些网络上连续传输的消息段的每一位，以及决定怎样将信息打包成消息域和如何解码。在其它网络上（像MAP和Modbus Plus）Modbus消息被转成与串行传输无关的帧。1.传输模式特性ASCII可打印字符便于故障检测，而且对于用语言（如Fortran）编程的主计算机及主PC很适宜。RTU则适用于 机器语言编程的计算机和PC主机。用RTU模式传输的数据是8位二进制字符。如欲转换为ASCII模式，则每个RTU字符首先应分为高位和低位两部分，这两部分各含4位，然后转换成十六进制等量值。用以构成报文的ASCII字符都是十六进制字符。ASCII模式使用的 字符虽是RTU模式的两倍，但ASCII数据的译码和处理更为容易一些，此外，用RTU模式时报文字符必须以连续数据流的形式传送，用ASCII模式，字符之间可产生长达1s的间隔，以适应速度较慢的机器。控制器能设置为两种传输模式（ASCII或RTU）中的任何一种在标准的Modbus网络通信。2.ASCII模式当控制器设为在Modbus网络上以ASCII（美国标准信息交换代码）模式通信，一个信息中的每8位字节作为2个ASCII字符传输，如数值63H用ASCII方式时，需发送两个字节，即ASCII “ 6"（0110110）和ASCII ” 3 “ （0110011），ASCII字符占用的位数有7位和8位，国际通用7位为多。这种方式的主要优点是字符发送的时间间隔可达到1秒而不产生错误。代码系统 十六进制，ASCII 字符0...9,A...F消息中的每个ASCII字符都是一个十六进制字符组成 每个字节的位1个起始位 7个 数据位，小的有效位先发送 1个 奇偶校验位，无校验则无 1个停止位（有校验时），2个Bit（无校验时） 错误检测域 LRC(纵向冗长检测)3.RTU模式当控制器设为在Modbus网络上以RTU模式通信，在消息中的每个8Bit字节按照原值传送，不做处理，如63H，RTU将直接发送01100011。这种方式的主要优点是：数据帧传送之间没有间隔，相同波特率下传输数据的密度要比ASCII高，传输速度更快。 代码系统 8位二进制， 十六进制数0...9，A...F 消息中的每个8位域都是一或两个十六进制字符组成 每个字节的位 1个起始位 8个 数据位，小的有效位先发送 1个 奇偶校验位，无校验则无 1个停止位（有校验时），2个Bit（无校验时）五、数据校验方式1.CRCCRC域是两个字节，包含一16位的二进制值。它由传输设备计算后加入到消息中。接收设备重新计算收到消息的CRC，并与接收到的CRC域中的值比较，如果两值不同，则有误。 CRC是先调入一值是全 “ 1 ” 的16位寄存器，然后调用一过程将消息中连续的8位字节和当前寄存器中的值进行处理。仅每个字符中的8Bit数据对CRC有效，起始位和停止位以及 奇偶校验位均无效。 CRC产生过程中，每个8位字符都单独和寄存器内容相异或（XOR），结果向低有效位方向移动，高有效位以0填充。LSB被提取出来检测，如果LSB为1，寄存器单独和预置的值或一下，如果LSB为0，则不进行。整个过程要重复8次。在后一位（第8位）完成后，下一个8位字节又单独和寄存器的当前值相异或（XOR）。终寄存器中的值，是消息中所有的字节都执行之后的CRC值。 CRC添加到消息中时，低字节先加入，然后高字节。 CRC-16错误校验程序如下：报文（此处只涉及 数据位，不指起始位、停止位和任选的 奇偶校验位）被看作是一个连续的二进制，其高有效位（MSB）发送。报文先与X↑16相乘（左移16位），然后看X↑16+X↑15+X↑2+1除，X↑16+X↑15+X↑2+1可以表示为二进制数11000，0000，0000，0101。整数商位忽略不记，16位余数加入该报文（MSB先发送），成为2个CRC校验字节。余数中的1全部初始化，以免所有的零成为一条报文被接收。经上述处理而含有CRC字节的报文，若无错误，到接收设备后再被同一多项式（X↑16+X↑15+X↑2+1）除，会得到一个零余数（接收设备核验这个CRC字节，并将其与被传送的CRC比较）。全部运算以2为模（无进位）。 习惯于成串发送数据的设备会送出字符的右位（LSB-低有效位）。而在生成CRC情况下，发送首位应是被除数的高有效位MSB。由于在运算中不用进位，为便于操作起见，计算CRC时设MSB在右位。生成多项式的位序也必须反过来，以保持一致。多项式的MSB略去不记，因其只对商有影响而不影响余数。 生成CRC-16校验字节的步骤如下： ①装如一个16位寄存器，所有数位均为1。 ②该16位寄存器的高位 字节与开始8位字节进行 “ 异或 ” 运算。运算结果放入这个16位寄存器。 ③把这个16寄存器向右移一位。 ④若向右（标记位）移出的数位是1，则生成多项式10，1000，000，0000，001和这个寄存器进行 “ 异或 ” 运算；若向右移出的数位是0，则返回③。 ⑤重复③和④，直至移出8位。 ⑥另外8位与该十六位寄存器进行 “ 异或 ” 运算。 ⑦重复③~⑥，直至

该 报文所有字节均与16位 寄存器进行 “ 异或 ” 运算，并移位8次。 ⑧这个16位寄存器的内容即2字节CRC错误校验，被加到报文的高有效位。 另外，在某些非ModBus通信协议中也经常使用CRC16作为校验手段，而且产生了一些CRC16的变种，他们是使用CRC16多项式X↑16+X↑15+X↑2+1，单装入的16位寄存器为0000；使用CRC16的反序X↑16+X↑14+X↑1+1，装入寄存器值为0000或FFFFH。2.LRCLRC错误校验用于ASCII模式。这个错误校验是一个8位二进制数，可作为2个ASCII十六进制字节传送。把十六进制字符转换成二进制，加上无循环进位的二进制字符和二进制补码结果生成LRC错误校验（参见图）。这个LRC在接收设备进行核验，并与被传送的LRC进行比较，冒号（：）、回车符号（CR）、换行字符（LF）和置入的其他任何非ASCII十六进制字符在运算时忽略不计。六、协议比较Modbus的协议内容是完全公开的，内容是简单滴，实现起来是非常容易滴，单片机、PLC、DCS统统都能轻易实现。 Profibus则要复杂一些，关键是需要专用芯片进行二次开发，并且需要得到上级组织的认证，开发成本肯定高不少 。 当然从性能上讲，基于串口的modbus rtu/ASCII通讯性能肯定比不过profibus dp，但是就一些仪表级的简单通讯或者控制器级别的小数据量通讯，modbus是足以胜任的。说白了，就是modbus是*丝，profibus是高富帅！Modbus支持的功能码功能码名称作用01读取线圈状态取得一组逻辑线圈的当前状态（ON/OFF)02读取输入状态取得一组开关输入的当前状态（ON/OFF)03读取保持寄存器在一个或多个保持寄存器中取得当前的二进制值04读取输入寄存器在一个或多个输入寄存器中取得当前的二进制值05强置单线圈强置一个逻辑线圈的通断状态06预置单寄存器把具体二进值装入一个保持寄存器07读取异常状态取得8个内部线圈的通断状态，这8个线圈的地址由控制器决定08回送诊断校验把诊断校验报文送从机，以对通信处理进行评鉴09编程（只用于484）使主机模拟编程器作用，修改PC从机逻辑10控询（只用于484）可使主机与一台正在执行长程序任务从机通信，探询该从机是否已完成其操作任务，仅在含有功能码9的报文发送后，本功能码才发送11读取事件计数可使主机发出单询问，并随即判定操作是否成功，尤其是该命令或其他应答产生通信错误时12读取通信事件记录可是主机检索每台从机的ModBus事务处理通信事件记录。如果某项事务处理完成，记录会给出有关错误13编程（184/384 484 584）可使主机模拟编程器功能修改PC从机逻辑14探询（184/384 484 584）可使主机与正在执行任务的从机通信，定期控询该从机是否已完成其程序操作，仅在含有功能13的报文发送后，本功能码才得发送15强置多线圈强置一串连续逻辑线圈的通断16预置多寄存器把具体的二进制值装入一串连续的保持寄存器17报告从机标识可使主机判断编址从机的类型及该从机运行指示灯的状态18（884和MICRO 84）可使主机模拟编程功能，修改PC状态逻辑19重置通信链路发生非可修改错误后，是从机复位于已知状态，可重置顺序字节20读取通用参数（584L）显示扩展存储器文件中的数据信息21写入通用参数（584）把通用参数写入扩展存储文件，或修改之22～64保留作扩展功能备用65～72保留以备用户功能所用留作用户功能的扩展编码73～119非法功能120～127保留留作内部作用128～255保留用于异常应答功能码命令详解在这些功能码中较长使用的是1、2、3、4、5、6号功能码，使用它们即可实现对下位机的数字量和模拟量的读写操作。1、01号命令，读可读写数字量寄存器（线圈状态）：计算机发送命令：[设备地址] [命令号01] [起始寄存器地址高8位] [低8位] [读取的寄存器数高8位] [低8位] [CRC校验的低8位] [CRC校验的高8位] 例：[11][01][00][13][00][25][CRC低][CRC高] 意义如下：