

西门子CPU控制器6ES7511-1CK01-0AB0代理现货

产品名称	西门子CPU控制器6ES7511-1CK01-0AB0代理现货
公司名称	浔之漫智控技术（上海）有限公司
价格	.00/件
规格参数	规格:全新原装 型号:齐全 德国:PLC
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213
联系电话	18717946324 18717946324

产品详情

西门子CPU控制器6ES7511-1CK01-0AB0代理现货

作为西门子PLC控制器的全国代理商，我们的产品在全国范围内有广泛的销售网络和服务网络。我们与西门子公司建立了良好的合作关系，在产品质量、技术支持和配件供应等方面享有优先权。我们还为客户提供专业、快捷和周到的售后服务，在设备故障或问题出现时，我们能够在最短的时间内进行快速响应和解决问题。

串口调试时，准备一个好用的调试工具，如串口调试助手、串口精灵等，有事半功倍之效果；强烈建议不要带电插拔串口，插拔时至少有一端是断电的，否则串口易损坏。单工、半双工和全双工的定义如果在通信过程的任意时刻，信息只能由一方A传到另一方B，则称为单工。如果在任意时刻，信息既可由A传到B，又能由B传A，但只能由一个方向上的传输存在，称为半双工传输。如果在任意时刻，线路上存在A到B和B到A的双向信号传输，则称为全双工。电话线就是二线全双工信道。由于采用了回波抵消技术，双向的传输信号不致混淆不清。双工信道有时也将收、发信道分开，采用分离的线路或频带传输相反方向的信号，如回线传输。奇偶校验串行数据在传输过程中，由于干扰可能引起信息的出错，例如，传输字符‘E’，其各位为：0100，0101=45HD7 D0由于干扰，可能使位变为1，这种情况，我们称为出现了“误码”。我们把如何发现传输中的错误，叫“检错”。发现错误后，如何消除错误，叫“纠错”。的检错方法是“奇偶校验”，即在传送字符的各位之外，再传送1位奇/偶校验位。可采用奇校验或偶校验。奇校验：所有传送的数位（含字符的各数位和校验位）中，“1”的个数为奇数，如：1 0110，01010 0110，0001偶校验：所有传送的数位（含字符的各数位和校验位）中，“1”的个数为偶数，如：1 010 0，01010 0100，0001奇偶校验能够检测出信息传输过程中的部分误码（1位误码能检出，2位及2位以上误码不能检出），同时，它不能纠错。在发现错误后，只能要求重发。但由

于其实现简单，仍得到了广泛使用。有些检错方法，具有自动纠错能力。如循环冗余码（CRC）检错等。串口通讯流控制我们在串行通讯处理中，常常看到RTS/CTS和XON/XOFF这两个选项，这就是两个流控制的选项，目前流控制主要应用于调制解调器的数据通讯中，但对普通RS232编程，了解一点这方面的知识是有好处的。那么，流控制在串行通讯中有何作用，在编制串行通讯程序怎样应用呢？这里我们就来谈谈这个问题。

1.流控制在串行通讯中的作用这里讲到的“流”，当然指的是数据流。数据在两个串口之间传输时，常常会出现丢失数据的现象，或者两台计算机的处理速度不同，如台式机与单片机之间的通讯，接收端数据缓冲区已满，则此时继续发送来的数据就会丢失。现在我们在网络上通过MODEM进行数据传输，这个问题就尤为突出。流控制能解决这个问题，当接收端数据处理不过来时，就发出“不再接收”的信号，发送端就停止发送，直到收到“可以继续发送”的信号再发送数据。因此流控制可以控制数据传输的进程，防止数据的丢失。PC机中常用的两种流控制是硬件流控制（包括RTS/CTS、DTR/CTS等）和软件流控制XON/XOFF（继续/停止），下面分别说明。

2.硬件流控制硬件流控制常用的有RTS/CTS流控制和DTR/DSR（数据终端就绪/数据设置就绪）流控制。硬件流控制必须将相应的电缆线连上，用RTS/CTS（请求发送/清除发送）流控制时，应将通讯两端的RTS、CTS线对应相连，数据终端设备（如计算机）使用RTS来起始调制解调器或其它数据通讯设备的数据流，而数据通讯设备（如调制解调器）则用CTS来起动和暂停来自计算机的数据流。这种硬件握手方式的过程为：我们在编程时根据接收端缓冲区大小设置一个高位标志（可为缓冲区大小的75%）和一个低位标志（可为缓冲区大小的25%），当缓冲区内数据量达到高位时，我们在接收端将CTS线置低电平（送逻辑0），当发送端的程序检测到CTS为低后，就停止发送数据，直到接收端缓冲区的数据量低于低位而将CTS置高电平。RTS则用来标明接收设备有没有准备好接收数据。常用的流控制还有还有DTR/DSR（数据终端就绪/数据设置就绪）。我们在此不再详述。由于流控制的多样性，我个人认为，当软件里用了流控制时，应做详细的说明，如何接线，如何应用。

3.软件流控制由于电缆线的限制，我们在普通的控制通讯中一般不用硬件流控制，而用软件流控制。一般通过XON/XOFF来实现软件流控制。常用方法是：当接收端的输入缓冲区内数据量超过设定的高位时，就向数据发送端发出XOFF字符（十进制的19或Control-S，设备编程说明书应该有详细阐述），发送端收到XOFF字符后就立即停止发送数据；当接收端的输入缓冲区内数据量低于设定的低位时，就向数据发送端发出XON字符（十进制的17或Control-Q），发送端收到XON字符后就立即开始发送数据。一般可以从设备配套源程序中找到发送的是什么字符。应该注意，若传输的是二进制数据，标志字符也有可能出现在数据流中出现而引起误操作，这是软件流控制的缺陷，而硬件流控制不会有这个问题

Delphi是一种具有功能强大、简便易用和代码执行速度快等优点的可视化快速应用开发工具，它在构架企业信息系统方面发挥着越来越重要的作用，许多程序员愿意选择Delphi作为开发工具编制各种应用程序。但是，美中不足之处是Delphi没有自带的串口通信控件，在它的帮助文档里也没有提及串口通信，这就给编制通信程序的开发人员带来许多不便。目前，利用Delphi实现串口通信的常用的方法有3种：一是利用控件，如MSCOMM控件和SPCOMM控件；二是使用API函数；三是调用其他串口通信程序。其中利用API编写串口通信程序较为复杂，需要掌握大量的通信知识。相比较而言，利用SPCOMM控件则相对较简单，并且该控件具有丰富的与串口通信密切相关的属性及事件，提供了对串口的各种操作，而且还支持多线程。下面本文结合实例详细介绍SPCOMM控件的使用。

