# 详解RS232、RS485、RS485、串口&握手

## **一、RS232基础知识**

计算机与计算机或计算机与终端之间的数据传送可以采用串行通讯和并行通讯二种方式。由于串行通讯方式具有使用线路少、成本低，特别是在远程传输时，避免了多条线路特性的不一致而被广泛采用。

在串行通讯时，要求通讯双方都采用一个标准接口，使不同的设备可以方便地连接起来进行通讯。RS-232-C接口（又称EIA RS-232-C）是目前最常用的一种串行通讯接口。

RS-232-C是美国电子工业协会EIA（Electronic Industry Association）制定的一种串行物理接口标准。RS是英文“推荐标准”的缩写，232为标识号，C表示修改次数，代表RS232的最新一次修改（1969），在这之前，有RS232B、RS232A。它是在1970 年由美国电子工业协会（EIA）联合贝尔系统、调制解调器厂家及计算机终端生产厂家共同制定的用于串行通讯的标准。它的全名是“数据终端设备（DTE）和数据通讯设备（DCE）之间串行二进制数据交换接口技术标准”。

### **1、电气特性**

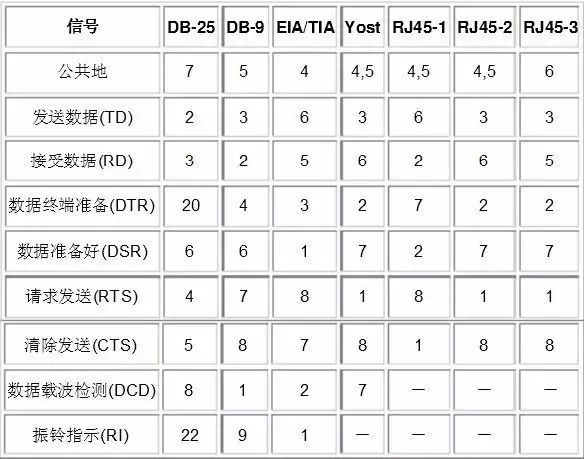
EIA-RS-232C对电器特性、逻辑电平和各种信号线功能都作了规定。 在TxD和RxD上：逻辑1(MARK)=-3V～-15V ，逻辑0(SPACE)=+3～＋15V  在RTS、CTS、DSR、DTR和DCD等控制线上：

信号有效（接通，ON状态，正电压）＝+3V～+15V

信号无效（断开，OFF状态，负电压)=-3V～-15V

根据设备供电电源的不同，+-5、+-10、+-12和+-15这样的电平都是可能的。

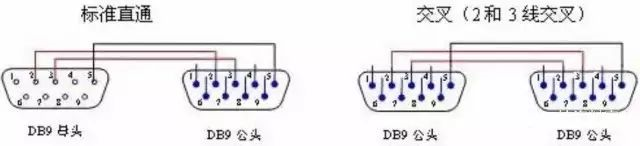
### **2、连接器的机械特性**

由于RS-232C并未定义连接器的物理特性，因此，出现了DB-25、DB-15和DB-9各种类型的连接器，其引脚的定义也各不相同。最近，8管脚的RJ-45型连接器变得越来越普遍，尽管它的管脚分配相差很大。EIA/TIA 561标准规定了一种管脚分配的方法，但是由Dave Yost发明的被广泛使用在Unix计算机上的Yost串连设备配线标准 （"Yost Serial Device Wiring Standard"）以及其他很多设备都没有采用上述任一种连线标准。下表中列出的是被较多使用的RS-232中的信号和管脚分配：

信号的标注是从DTE设备的角度出发的，TD、DTR和RTS信号是由DTE产生的，RD、DSR、CTS、DCD和RI信号是由DCE产生的。

PC 机的RS-232 口为9 芯针插座。一些设备与PC 机连接的RS-232 接口,因为不使用对方的传送控制信号,只需三条接口线,即“发送数据TXD”、“接收数据RXD”和“信号地GND”。

双向接口能够只需要3根线制作是因为RS-232的所有信号都共享一个公共接地。非平衡电路使得RS-232非常的容易受两设备间基点电压偏移的影响。对于信号的上升期和下降期，RS-232也只有相对较差的控制能力，很容易发生串话的问题。RS-232被推荐在短距离（15m以内）间通信。由于非对称电路的关系，RS-232接口电缆通常不是由双绞线制作的。



### **3、传输电缆**

RS-232-C标准规定的数据传输速率为每秒50、75、 100、150、300、600、1200、2400、4800、9600、19200波特，驱动器允许有2500pF的电容负载，通信距离将受此电容限制。

例如，采用150pF/m的通信电缆时，最大通信距离为15m；若每米电缆的电容量减小，通信距离可以增加。传输距离短的另一原因是RS-232属单端信号传送，存在共地噪声和不能抑制共模干扰等问题，因此一般用于20m以内的通信。

由RS-232C标准规定在码元畸变小于4%的情况下，传输电缆长度应为50 英尺，其实这个4%的码元畸变是很保守的，在实际应用中，约有99%的用户是按码元畸变10-20%的范围工作的，所以实际使用中最大距离会远超过50英尺，美国DEC公司曾规定允许码元畸变为10%而得出下面实验结果。其中1号电缆为屏蔽电缆，型号为DECP.NO.9107723 内有三对双绞线，每对由22# AWG 组成，其外覆以屏蔽网。2号电缆为不带屏蔽的电缆。型号为DECP.NO.9105856-04是22#AWG的四芯电缆。

### **4、链路层**

在RS-232标准中，字符是以一系列位元来一个接一个的传输。最长用的编码格式是异步起停asynchronous start-stop格式，它使用一个起始位后面紧跟7或8 个数据比特，这个可能是奇偶位，然后是两个停止位。所以发送一个字符需要10比特，带来的一个好的效果是使全部的传输速率，发送信号的速率以10分划。

串行通信在软件设置里需要做多项设置，最常见的设置包括波特率、奇偶校验和停止位。 波特率是指从一设备发到另一设备的波特率，即每秒钟多少比特bits per second (bit/s)。典型的波特率是300、1200、2400、9600、19200等bit/s。一般通信两端设备都要设为相同的波特率，但有些设备也可以设置为自动检测波特率。

奇偶校验Parity是用来验证数据的正确性。奇偶校验一般不用，如果使用，那么既可以做奇校验也可以做偶校验。奇偶校验是通过修改每一发送字节（也可以限制发送的字节）来工作的。如果不作奇偶校验，那么数据是不会被改变的。在偶校验中，因为奇偶校验位会被相应的置1或0（一般是最高位或最低位），所以数据会被改变以使得所有传送的数位（含字符的各数位和校验位）中“1”的个数为偶数；在奇校验中，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。奇偶校验可以用于接受方检查传输是否发送生错误——如果某一字节中“1”的个数发生了错误，那么这个字节在传输中一定有错误发生。如果奇偶校验是正确的，那么要么没有发生错误要么发生了偶数个的错误。

停止位是在每个字节传输之后发送的，它用来帮助接受信号方硬件重同步。

在串行通信软件设置中D/P/S是常规的符号表示。8/N/1（非常普遍）表明8bit数据，没有奇偶校验，1bit停止位。数据位可以设置为7、8或者9，奇偶校验位可以设置为无（N）、奇（O）或者偶（E），奇偶校验位可以使用数据中的比特位，所以8/E/1就表示一共8位数据位，其中一位用来做奇偶校验位。停止位可以是1、1.5或者2位的（1.5是用在波特率为60wpm的电传打字机上的）。

### **5、传输控制**

当需要发送握手信号或数据完整性检测时需要制定其他设置。公用的组合有RTS/CTS，DTR/DSR或者XON/XOFF（实际中不使用连接器管脚而在数据流内插入特殊字符）。

接受方把XON/XOFF信号发给发送方来控制发送方何时发送数据，这些信号是与发送数据的传输方向相反的。XON信号告诉发送方接受方准备好接受更多的数据，XOFF信号告诉发送方停止发送数据直到知道接受方再次准备好。XON/XOFF一般不赞成使用，推荐用RTS/CTS控制流来代替它们。

XON/XOFF是一种工作在终端间的带内方法，但是必须两端都支持这个协议，而且在突然启动的时候会有混淆的可能。

XON/XOFF可以工作于3线的接口。RTS/CTS最初是设计为电传打字机和调制解调器半双工协作通信的，每次它只能一方调制解调器发送数据。终端必须发送请求发送信号然后等到调制解调器回应清除发送信号。尽管RTS/CTS是通过硬件达到握手，但它有自己的优势。

**6、RS-232标准的不足**

经过许多年来RS-232 器件以及通信技术的改进，RS-232 的通信距离已经大大增加。由于RS-232 接口标准出现较早，难免有不足之处，主要有以下四点：

（1） 接口的信号电平值较高，易损坏接口电路的芯片，又因为与TTL 电平不兼容故需使用电平转换电路方能与TTL 电路连接。

（2） 传输速率较低，在异步传输时，波特率为20Kbps。现在由于采用新的UART 芯片16C550 等，波特率达到115.2Kbps。

（3） 接口使用一根信号线和一根信号返回线而构成共地的传输形式， 这种共地传输容易产生共模干扰，所以抗噪声干扰性弱。

（4） 传输距离有限，最大传输距离标准值为50米，实际上也只能用在15米左右。

## **二、RS485基础知识**

针对RS-232串口标准的局限性，人们又提出了RS-422,RS-485接口标准。RS-485/422采用平衡发送和差分接收方式实现通信：发送端将串行口的TTL电平信号转换成差分信号A、B两路输出，经过线缆传输之后在接收端将差分信号还原成TTL电平信号。由于传输线通常使用双绞线，又是差分传输，所以有极强的抗共模干扰的能力，总线收发器灵敏度很高，可以检测到低至200mV电压。故传输信号在千米之外都可以恢复。

### **1、RS-485 的电气特性**

驱动器能输出±7V的共模电压

接收器的输入电阻RIN≥12kΩ

输入端的电容≤50pF

在节点数为32个，配置了120Ω的终端电阻的情况下，驱动器至少还能输出电压1.5V（终端电阻的大小与所用双绞线的参数有关）

发送端：逻辑"1"以两线间的电压差为+（2 至6） V 表示；逻辑"0"以两线间的电压差为-（2 至6）V 表示。

接收器的输入灵敏度为200mV（即（V+）-（V-）≥0.2V，表示信号"0"；（V+）-（V-）≤-0.2V，表示信号"1"）

### **2、传输速率与传输距离**

RS-485 的数据最高传输速率为10Mbps，最大的通信距离约为1219M，传输速率与传输距离成反比，在10Kb/S的传输速率下，才可以达到最大的通信距离。

但是由于RS-485 常常要与PC 机的RS-232口通信，所以实际上一般最高115.2Kbps。又由于太高的速率会使RS-485 传输距离减小，所以往往为9600bps 左右或以下。

### **3、网络拓扑**

RS-485 接口是采用平衡驱动器和差分接收器的组合，抗共模干扰能力增强，即抗噪声干扰性好。RS-485采用半双工工作方式，支持多点数据通信。

RS-485总线网络拓扑一般采用终端匹配的总线型结构。即采用一条总线将各个节点串接起来，不支持环形或星型网络。如果需要使用星型结构，就必须使用485中继器或者485集线器才可以。RS-485/422总线一般最大支持32个节点，如果使用特制的485芯片，可以达到128个或者256个节点，最大的可以支持到400个节点。

### 4、连接器

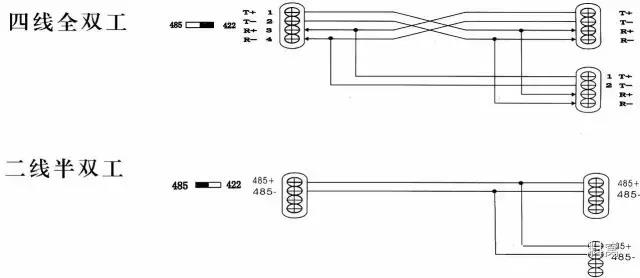
RS-485 的国际标准并没有规定RS485 的接口连接器标准、所以采用接线端子或者DB-9、DB-25 等连接器都可以。

## **三、RS422基础知识**

RS-422 的电气性能与RS-485近似一样。主要的区别在于：

（1）RS-485 有2 根信号线：发送和接收都是A 和B。由于RS-485 的收与发是共用两根线，所以不能够同时收和发（半双工）。

（2）RS-422 有4 根信号线：两根发送（Y、Z）、两根接收（A、B）。由于RS-422 的收与发是分开的，所以可以同时收和发（全双工）。

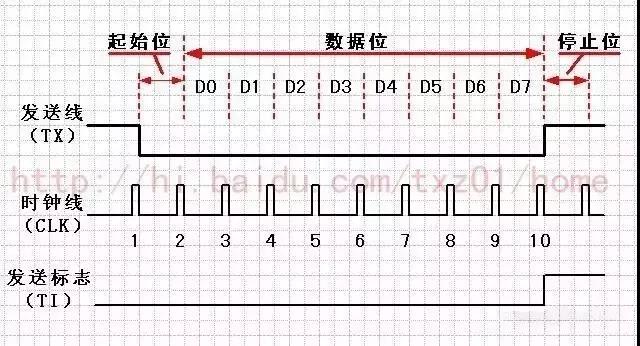
（3）支持多机通信的RS-422将Y-A 短接作为RS-485 的A、将RS-422 的Z-B 短接作为RS-485 的B可以这样简单转换为RS-485。

很多人往往都误认为RS-422串行接口是RS-485串行接口的全双工版本，实际上，它们在电器特性上存在着不少差异，共模电压范围和接收器输入电阻不同使得该两个标准适用于不同的应用领域。RS-485串行接口的驱动器可用于RS-422串行接口的应用中，因为RS-485串行接口满足所有的RS-422串行接口性能参数，反之则不能成立。对于RS-485串行接口的驱动器，共模电压的输出范围是-7V和+12V之间；对于RS-422串行接口的驱动器，该项性能指标仅有±7V。RS-422串行接口接收器的最小输入电阻是4KΩ；而RS-485串行接口接收器的最小输入电阻则是12KΩ。

## **四、串口与握手基础知识**

### **一、串口基础知识**

串口是计算机上一种非常通用设备通信的协议（不要与通用串行总线Universal Serial Bus或者USB混淆）。大多数计算机包含两个基于RS232的串口。串口同时也是仪器仪表设备通用的通信协议；很多GPIB兼容的设备也带有RS-232口。同时，串口通信协议也可以用于获取远程采集设备的数据。

串口通信的概念非常简单，串口按位（bit）发送和接收字节。尽管比按字节（byte）的并行通信慢，但是串口可以在使用一根线发送数据的同时用另一根线接收数据。它很简单并且能够实现远距离通信。比如IEEE488定义并行通行状态时，规定设备线总常不得超过20米，并且任意两个设备间的长度不得超过2米；而对于串口而言，长度可达1200米。

典型地，串口用于ASCII码字符的传输。通信使用3根线完成：（1）地线，（2）发送，（3）接收。由于串口通信是异步的，端口能够在一根线上发送数据同时在另一根线上接收数据。其他线用于握手，但是不是必须的。串口通信最重要的参数是波特率、数据位、停止位和奇偶校验。对于两个进行通行的端口，这些参数必须匹配：

#### 1、波特率

这是一个衡量通信速度的参数。它表示每秒钟传送的bit的个数。例如300波特表示每秒钟发送300个bit。当我们提到时钟周期时，我们就是指波特率例如如果协议需要4800波特率，那么时钟是4800Hz。这意味着串口通信在数据线上的采样率为4800Hz。通常电话线的波特率为14400，28800和36600。波特率可以远远大于这些值，但是波特率和距离成反比。高波特率常常用于放置的很近的仪器间的通信，典型的例子就是GPIB设备的通信。

#### 2、数据位

这是衡量通信中实际数据位的参数。当计算机发送一个信息包，实际的数据不会是8位的，标准的值是5、7和8位。如何设置取决于你想传送的信息。比如，标准的ASCII码是0～127（7位）。扩展的ASCII码是0～255（8位）。如果数据使用简单的文本（标准 ASCII码），那么每个数据包使用7位数据。每个包是指一个字节，包括开始/停止位，数据位和奇偶校验位。由于实际数据位取决于通信协议的选取，术语“包”指任何通信的情况。

#### 3、停止位

用于表示单个包的最后一位。典型的值为1，1.5和2位。由于数据是在传输线上定时的，并且每一个设备有其自己的时钟，很可能在通信中两台设备间出现了小小的不同步。因此停止位不仅仅是表示传输的结束，并且提供计算机校正时钟同步的机会。适用于停止位的位数越多，不同时钟同步的容忍程度越大，但是数据传输率同时也越慢。

#### 4、奇偶校验位

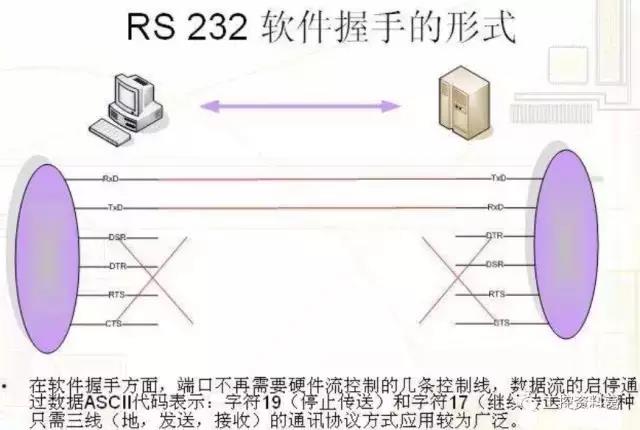
在串口通信中一种简单的检错方式。有四种检错方式：偶、奇、高和低。当然没有校验位也是可以的。对于偶和奇校验的情况，串口会设置校验位（数据位后面的一位），用一个值确保传输的数据有偶个或者奇个逻辑高位。例如，如果数据是011，那么对于偶校验，校验位为0，保证逻辑高的位数是偶数个。如果是奇校验，校验位位1，这样就有3个逻辑高位。高位和低位不真正的检查数据，简单置位逻辑高或者逻辑低校验。这样使得接收设备能够知道一个位的状态，有机会判断是否有噪声干扰了通信或者是否传输和接收数据是否不同步。

### **二、握手基础知识**

RS-232通行方式允许简单连接三线：Tx、Rx和地线。但是对于数据传输，双方必须对数据定时采用使用相同的波特率。尽管这种方法对于大多数应用已经足够，但是对于接收方过载的情况这种使用受到限制。这时需要串口的握手功能。在这一部分，我们讨论三种最常用的RS-232握手形式：软件握手、硬件握手和Xmodem。

#### 1、软件握手

我们讨论的第一种握手是软件握手。通常用在实际数据是控制字符的情况，类似于GPIB使用命令字符串的方式。必须的线仍然是三根：Tx、Rx和地线，因为控制字符在传输线上和普通字符没有区别，函数SetXModem允许用户使能或者禁止用户使用两个控制字符XON和OXFF。这些字符在通信中由接收方发送，使发送方暂停。



例如：假设发送方以高波特率发送数据。在传输中，接收方发现由于CPU忙于其他工作，输入buffer已经满了。为了暂时停止传输，接收方发送XOFF，典型的值是十进制19，即十六进制13，直到输入buffer空了。一旦接收方准备好接收，它发送XON，典型的值是十进制17，即十六进制11，继续通信。输入buffer半满时，LabWindows发送XOFF。此外，如果XOFF传输被打断，LabWindows会在buffer达到75％和90％时发送XOFF。显然，发送方必须遵循此守则以保证传输继续。

#### 2、硬件握手

第二种是使用硬件线握手。和Tx和Rx线一样，RTS/CTS和DTR/DSR一起工作，一个作为输出，另一个作为输入。第一组线是RTS （Request to Send）和CTS（Clear to Send）。当接收方准备好接收数据，它置高RTS线表示它准备好了，如果发送方也就绪，它置高CTS，表示它即将发送数据。另一组线是DTR（Data Terminal Ready）和DSR（Data Set Ready）。这些现主要用于Modem通信。使得串口和Modem通信他们的状态。例如：当Modem已经准备好接收来自PC的数据，它置高DTR线，表示和电话线的连接已经建立。读取DSR线置高，PC机开始发送数据。一个简单的规则是DTR/DSR用于表示系统通信就绪，而RTS/CTS用于单个数据包的传输。

在LabWindows，函数SetCTSMode使能或者禁止使用硬件握手。如果CTS模式使能，LabWindows使用如下规则：当PC发送数据：RS-232库必须检测CTS线高后才能发送数据。

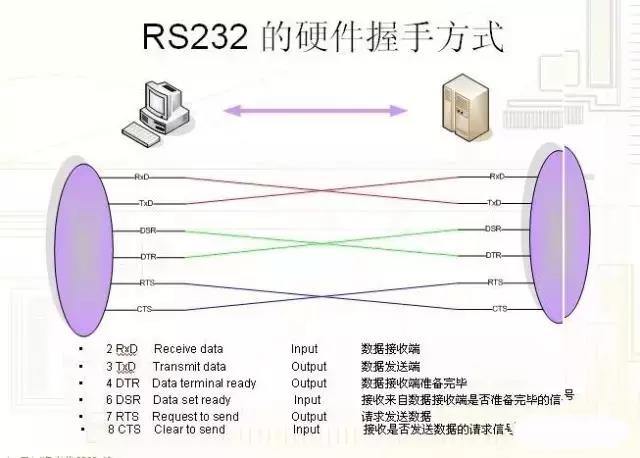
当PC接收数据：

如果端口打开，且输入队列有空接收数据，库函数置高RTS和DTR。

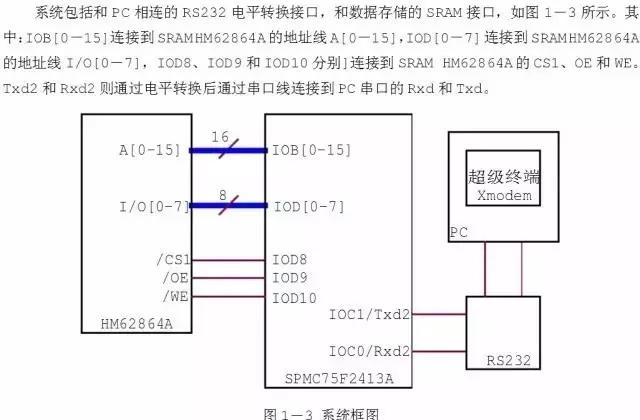
如果输入队列90％满，库函数置低RTS，但使DTR维持高电平。

如果端口队列近乎空了，库函数置高RTS，但使DRT维持高电平。

如果端口关闭，库函数置低RTS和DTR。



#### 3、XModem握手

最后讨论的握手叫做XModem文件传输协议。这个协议在Modem通信中非常通用。尽管它通常使用在Modem通信中，XModem协议能够直接在其他遵循这个协议的设备通信中使用。在LabWindows中，实际的XModem应用对用户隐藏了。只要PC和其他设备使用XModem协议，在文件传输中就使用LabWindows的XModem函数。函数是XModemConfig，XModemSend和XModemReceive。

XModem使用介于如下参数的协议：start\_of\_data、end\_of\_data、neg\_ack、wait\_delay、 start\_delay、max\_tries、packet\_size。这些参数需要通信双方认定，标准的XModem有一个标准的定义：然而，可以通过 XModemConfig函数修改，以满足具体需要。这些参数的使用方法由接收方发送的字符neg\_ack确定。这通知发送方其准备接收数据。它开始尝试发送，有一个超时参数start\_delay；当超时的尝试超过max\_ties次数，或者收到接收方发送的start\_of\_data，发送方停止尝试。如果从发送方收到start\_of\_data，接收方将读取后继信息数据包。包中含有包的数目、包数目的补码作为错误校验、packet\_size字节大小的实际数据包，和进一步错误检查的求和校验值。在读取数据后，接收方会调用wait\_delay，然后想发送方发送响应。如果发送方没有收到响应，它会重新发送数据包，直到收到响应或者超过重发次数的最大值max\_tries。如果一直没有收到响应，发送方通知用户传输数据失败。

由于数据必须以pack\_size个字节按包发送，当最后一个数据包发送时，如果数据不够放满一个数据包，后面会填充ASCII码NULL（0）字节。这导致接收的数据比原数据多。在XModem情况下一定不要使用XON/XOFF，因为XModem发送方发出包的数目很可能增加到XON/OFF控制字符的值，从而导致通信故障。