# USB和串口(COM)的区别，以及相互转换

1、串口、COM口是指的物理接口形式(硬件)。而TTL、RS-232、RS-485指的是串口的电平标准(电信号)。

2、接线的时候，一般只接GND、RX(接收)、TX(发送)。不会接入Vcc等电源线，避免与目标设备上的供电冲突。

（接线法则：主机的 TX 接目标设备的 RX ，主机的 RX 接目标设备的 TX ，但是很多设计人员为了接线更为直观而故意颠倒标记 RX、TX ，如果有问题可以尝试交换RX、TX，不会烧坏设备 。）

3、PL2303、CP2102、FT232R 芯片是用USB来扩展串口(TTL电平输出)的芯片，需要安装Windows驱动。

（常用于笔记本增加串口，注意其兼容性不如板载串口。优先选择：FT232R > CP2102 > PL2303 ）

4、MAX232芯片是TTL电平与RS232电平的专用双向转换芯片，不同引脚实现TTL转RS-232或RS-232转TTL的功能。

（TTL与RS232转换芯片很多很多，正向、逆向接口数量不同：比如：MAX202、SP213、MAX3232 ）

5、TTL电平标准 是 低电平为0，高电平为1（对地，标准数字电路逻辑）。

RS-232电平标准 是 正电平为0，负电平为1（对地高低，电压具体数值不敏感，甚至可以用高阻态）。

6、RS-485、RS-422 与RS-232类似，但是采用差分信号逻辑，更适合长距离、高速传输。这里略过不讲。

7、台式电脑后边带的D型9针插头（板载串口，公口），是RS-232电平标准的。可以通过MAX232转换为TTL电平。

串口、COM口：

COM口即串行通讯端口，简称串口。这里区别于 USB的“通用串行总线”和“SATA”串行硬盘。

一般我们见到的是两种物理形式。D型9针插头（DB9）和 4针杜邦头 两种，早年的25针串口已被淘汰。

# 串口信号定义和接线方法-5针串口-9针串口-全功能串口

•串口、并口接口定义

　　并行口与串行口的区别是交换信息的方式不同，并行口能同时通过8条数据线传输信息，一次传输一个字节；而串行口只能用1条线传输一位数据，每次传输一个字节的一位。并行口由于同时传输更多的信息，速度明显高于串行口，但串行口可以用于比并行口更远距离的数据传输。

1、25针并行口插口的针脚功能：

　 针脚 功能 　　　　　　　针脚 功能

　　 1 选通 (STROBE低电平) 10 确认 (ACKNLG低电平)

　　 2 数据位0 (DATAO) 11 忙 (BUSY)

　　 3 数据位1 (DATA1) 12 却纸 (PE)

　　 4 数据位2 (DATA2) 13 选择 (SLCT)

　　 5 数据位3 (DATA3) 14 自动换行 (AUTOFEED低电平)

　　 6 数据位4 (DATA4) 15 错误观点(ERROR低电平)

　　 7 数据位5 (DATA5) 16 初始化成(INIT低电平)

　　 8 数据位6 (DATA6) 17 选择输入 (SLCTIN低电平)

　　 9 数据位7 (DATA7) 18-25 地线路(GND)

2.串行口的典型代表是RS-232C及其兼容插口，有9针和25针两类。

25针串行口具有20mA电流环接口功能，用9、11、18、25针来实现。其针脚功能如下：

　针脚 功能 　　　　针脚 功能

　　 1 未用

　　 2 发出数据(TXD) 11 数据发送(一)

　　 3 接受数据(RXD) 12-17 未用

　　 4 请求发送(RTS) 18 数据接收( )

　　 5 清除发送(CTS) 19 未用

　　 6 数据准备好(DSR) 20 数据终端准备好比(DTR)

　　 7 信号地线路 (SG) 21 未用

　　 8 载波检测 (DCD) 22 振铃指示精神 (RI)

　　 9 发送返回( ) 23-24 未用

　　 10 未用 25 接收返回(一)

9针串行口的针脚功能：

　　 针脚 功能 针脚 功能

　　 1 载波检测(DCD) 6 数据准备好(DSR)

　　 2 接受数据(RXD) 7 请求发送(RTS)

　　 3 发出数据(TXD) 8 清除发送(CTS)

　　 4 数据终端准备好(DTR) 9 振铃指示(RI)

　　 5 信号地线(SG)

• 串口通信基本原理及接线方法

目前较为常用的串口有9针串口(DB9)和25针串口(DB25)，通信距离较近时(<12m)，可以用电缆线直接连接标准RS232端口(RS422,RS485较远)，若距离较远，需附加调制解调器(MODEM)。最为简单且常用的是三线制接法，即地、接收数据和发送数据三脚相连，本文只涉及到最为基本的接法，且直接用RS232相连。

1.DB9和DB25的常用信号脚说明

9针串口(DB9) 25针串口(DB25)

针号 功能说明 缩写 针号 功能说明 缩写

1 数据载波检测 DCD 8 数据载波检测 DCD

2 接收数据 RXD 3 接收数据 RXD

3 发送数据 TXD 2 发送数据 TXD

4 数据终端准备 DTR 20 数据终端准备 DTR

5 信号地 GND 7 信号地 GND

6 数据设备准备好 DSR 6 数据准备好 DSR

7 请求发送 RTS 4 请求发送 RTS

8 清除发送 CTS 5 清除发送 CTS

9 振铃指示 DELL 22 振铃指示 DELL

2.RS232C串口通信接线方法(三线制)

首先，串口传输数据只要有接收数据针脚和发送针脚就能实现：同一个串口的接收脚和发送脚直接用线相连，两个串口相连或一个串口和多个串口相连

· 同一个串口的接收脚和发送脚直接用线相连 对9针串口和25针串口，均是2与3直接相连;

· 两个不同串口(不论是同一台计算机的两个串口或分别是不同计算机的串口)

上面表格是对微机标准串行口而言的，还有许多非标准设备，如接收GPS数据或电子罗盘数据，只要记住一个原则：接收数据针脚(或线)与发送数据针脚(或线)相连，彼此交叉，信号地对应相接，就能百战百胜。

3.串口调试中要注意的几点：

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拨串口，插拨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=45H

D7 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”。

最简单的检错方法是“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 0110，0101

0 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 0100，0101

0 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。

PS:

　　DB9只有9根线，遵循RS232标准。定义如下：

　　DTR,DSR——DTE设备准备好/DCE设备准备好。主流控信号。

　　RTS,CTS——请求发送/清除发送。用于半双工时，收发切换。属于辅助流控信号。半双工的意思是说，发的时候不收，收的时候不发。那么怎么区分收发呢？缺省时是DCE向DTE发送数据，当DTE决定向DCE发数据时，先有效RTS，表示DTE希望向DCE发送，一般DCE不能马上转换收发状态，DTE就通过监测CTS是否有效来判断可否发送，这样避免了DTE在DCE未准备好时发送所导致的数据丢失。全双工时，这两个信号一直有效即可。

　　随着计算机的日益普及，很多非RS232的串口也要接入PC机，如果为每一种新出现的串口都增加一个新的I/O口显然不现实，因为PC后面板位置有限，因此，将RS232串口和非RS232串口都通过RS232口接入是最佳方案。UART的U(通用)指的就是这个意思。早期ROM BIOS和DOS里的通信软件都是为RS232设计的，在没有检测到DCD有效前不会发送数据，因此，就连发送一个字符这样朴素的应用也要给出DCD、DTR、DSR等控制信号。因此，串口接头上要将一些控制线短接，或者干脆绕过系统软件自己写通信程序。

　　到此，UART的涵义就总结为：通用的 异步 (串行) I/O口。

　　就在UART冠以通用二字，准备一统江湖的时候，制造商们不满于它的速度、体积和灵活性(软件可配置)，推出了USB和　1394串口。目前，笔记本上的UART串口有被取消的趋势，因而有网友发出了“没有串口，吾谁与归”的慨叹，古今多少事，都付笑谈中，USB取代UART是后话，暂且不表。

　　话说自从贺氏(Hayes)公司推出了聪明猫(SmartModem)，他们制定的MODEM接口就成了业界标准，自此以后，所有公司制造的兼容猫都符合贺氏标准(连AT指令也兼容)。

　　细观贺氏制定的MODEM串口，与RS232标准大不相同。DTR在整个通信过程中一直保持有效，DSR在MODEM上电后/可以拨号前有效(取决于软件对DSR的理解)，在通信过程的任意时刻，只要DTR/DSR无效，通信过程立即终止。在某种意义上，这也可以算是流控，但肯定不是RS232所指的那种主流控。如果拘泥于RS232，你是不会理解DTR和DSR的用途的。

贺氏不但改了DTR和DSR，竟然连RTS和CTS的涵义也重新定义了。因此，RTS和CTS已经不具有最开始的意义了。从字面理解RTS和CTS，是用于半双工通信的，当DTE想从收模式改为发模式时，就有效RTS请求发送，DCE收到RTS请求后不能立即完成转换，需要一段时间，然后有效CTS通知DTE：DCE已经转到发模式，DTE可以开始发送了。在全双工时，RTS和CTS都缺省置为有效即可。然而，在贺氏的MODEM串口定义中，RTS和CTS用于硬件流控，和什么全双工/半双工一点关系也没有。 注意，硬件流控是靠软件实现的，之所以强调“硬件”二字，仅仅是因为硬件流控提供了用于流量情况指示的硬件连线，并不是说，你只要把线连上，硬件就能自己流控。如果软件不支持，光连上RTS和CTS是没有用的。

　　RTS和CTS硬件流控的软件算法如下：(RTS有效表示PC机可以收，CTS有效表示MODEM可以收，这两个信号互相独立，分别指示一个方向的流量情况。)

　　PC端处理：

　　 发.　　 当发现（不一定及时发现） CTS (-3v to -15v)无效时，停止发送,

　　　　　　 当发现（不一定及时发现） CTS (3v to 15v)有效时，恢复发送；

　　 收. 　　 当接收buffers中的bytes当接收buffers中的bytes>N 时，给 RTS 无效信号（-3v to -15v);

　　MODEM端处理：

　　同上，但RTS与CTS交换。

　　在RS232中本来CTS 与RTS 有明确的意义，但自从贺氏(HAYES ) 推出了聪明猫(SmartModem)后就有点混淆了。在RS232中RTS 与CTS 是用来半双工模式下的方向切换；HAYES Modem中的RTS ，CTS 是用来进 行硬件流控的。通常UART的RTC、CTS 的含义指后者，即用来做硬流控的。

　　硬流控的RTS 、CTS ：RTS （Require To Send，发送请求）为输出信号，用于指示本设备准备好可接收；CTS（Clear To Send，发送清除）为输入信号，有效时停止发送。假定A、B两设备通信，A设备的RTS 连接B设备的CTS ；A设备的CTS 连接B设备 的RTS 。 前一路信号控制B设备的发送，后一路信号控制A设备的发送。对B设备的发送（A设备接收）来说，如果A设备接收缓冲快满的时发出RTS 信号（意思 通知B设备停止发送），B设备通过CTS 检测到该信号，停止发送；一段时间后A设备接收缓冲有了空余，发出RTS 信号，指示B设备开始发送数据。A设备发（B设备接收） 类似。上述功能也能在数据流中插入Xoff（特殊字符）和Xon（另一个特殊字符）信号来实现。A设备一旦接收到B设备发送过来的Xoff，立刻停止发 送；反之，如接收到B设备发送过来的Xon，则恢复发送数据给B设备。同理，B设备也类似，从而实现收发双方的速度匹配。

　　半双工的方向切换：RS232中使用DTR（Date Terminal Ready，数据终端准备）与DSR（Data Set Ready ，数据设备准备好）进行主流控，类似上述的RTS 与CTS 。对半双工的通信的DTE（Date Terminal Equipment，数据终端设备）与DCE（Data circuit Equipment ）来说，默认的方向是DTE接收，DCE发送。如果DTE要发送数据，必须发出RTS 信号，请求发送数据。DCE收到后如果 空闲则发出CTS 回 应RTS 信 号，表示响应请求，这样通信方向就变为DTE->TCE，同时RTS 与CTS 信号必须一直保持。从这里可以看出，CTS ，TRS虽 然也有点流控的意思（如CTS 没有发出，DTE也不能发送数据），但主要是用来进行方向切换的。

#如果UART只有RX、TX两个信号，要流控的话只能是软流控；如果有RX，TX，CTS ，RTS 四个信号，则多半是支持硬流控的UART；如果有 RX，TX，CTS ，RTS ，DTR，DSR 六个信号的话，RS232标准的可能性比较大。

顺便提一下：

DCD（ Data Carrier Detect， 数据载波检测）：DCE向DTE指示，线路上检测到载波。

RI（Ring Indicator，振铃指示）：DCE向DTE指示，有呼叫接入。

• 串口通信基本接线方法

１.串口通信基本接线方法，目前较为常用的串口有9针串口（DB9）和25针串口（DB25），通信距离较近时( <12m)，可以用电缆线直接连接标准RS232端口(RS422,RS485较远)，若距离较远，需附加调制解调器（MODEM）。最为简单且常用的是三线制接法，即地、接收数据和发送数据三脚相连，本文只涉及到最为基本的接法，且直接用RS232相连。

2.RS232C串口通信接线方法（三线制） 首先，串口传输数据只要有接收数据针脚和发送针脚就能实现：同一个串口的接收脚和发送脚直接用线相连，两个串口相连或一个串口和多个串口相连 同一个串口的接收脚和发送脚直接用线相连 对9针串口和25针串口，均是2与3直接相连； 两个不同串口（不论是同一台计算机的两个串口或分别是不同计算机的串口） 上面表格是对微机标准串行口而言的，还有许多非标准设备，如接收GPS数据或电子罗盘数据，只要记住一个原则：接收数据针脚（或线）与发送数据针脚（或线）相连，彼些交叉，信号地对应相接，就能百战百胜。

3.串口调试中要注意的几点： 不同编码机制不能混接，如RS232C不能直接与RS422接口相连，市面上专门的各种转换器卖，必须通过转换器才能连接； 线路焊接要牢固，不然程序没问题，却因为接线问题误事； 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拨串口，插拨时至少有一端是断电的，否则串口易损坏

• 制作RS232串口连接线

9针串口功能一览表

　　 针脚 功能1 载波检测 （DCD） 2 接收数据 （RXD） 3 发送数据（TXD）

　　 4 数据终端准备好 （DTR） 5 信号地 （GND） 6 数据准备好 （DSR）

　　 7 发送请求 （RTS） 8 发送清除 （CTS） 9 振铃指示 （RI）

　　 串口联机线的连接方法

　　 串口联机线主要用于直接把两台电脑的com口连接。比较早一点的AT架构的电脑的串口有为9针,和25针两种,现在的ATX架构的电脑两个串口全部是9针。于是联机线就分为3种（9针对9针串口联机线,9针对25针串口联机线,25针对25针串口联机线）这些直接电缆连接线可以互换的连线方法如下表:

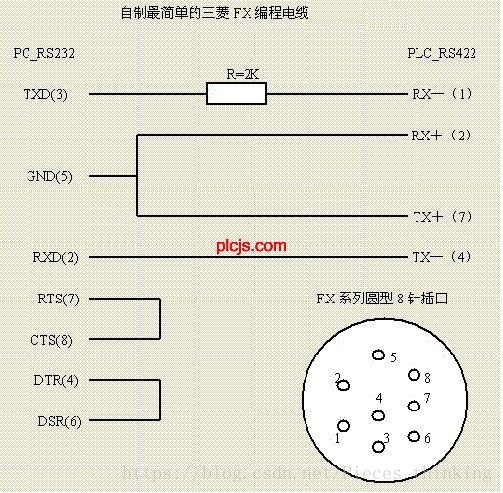
　　 串口连机线一览

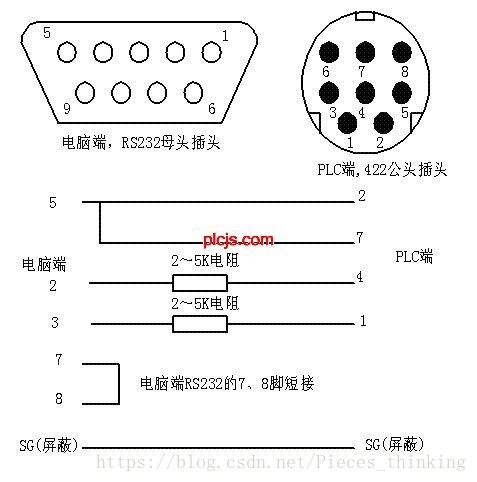
　　 9针对9针串口连接

　　 9针母头 9针母头

　　 2 —— 3 3 —— 2 4 —— 6 5 —— 5 6 —— 4 7 —— 8 8 —— 7

如果需要做其他用途，连接 针脚 功能 2 接收数据 （RXD） 3 发送数据（TXD）5 信号地 （GND）





# 串口 COM口 TTL RS-232 RS-485 区别

**Point：**

1、串口、COM口是指的物理接口形式(硬件)。而TTL、RS-232、RS-485是指的电平标准(电信号)。

2、接设备的时候，一般只接GND RX TX。不会接Vcc或者+3.3v的电源线，避免与目标设备上的供电冲突。

3、PL2303、CP2102芯片是 USB 转 TTL串口 的芯片，用USB来扩展串口(TTL电平)。

4、MAX232芯片是 TTL电平与RS232电平的专用双向转换芯片，可以TTL转RS-232，也可以RS-232转TTL。

5、TTL标准是低电平为0，高电平为1（+5V电平）。RS-232标准是正电平为0，负电平为1（±15V电平）。

6、RS-485与RS-232类似，但是采用差分信号负逻辑。这里略过不讲。

**串口、COM口：**

COM口即串行通讯端口，简称串口。这里区别于USB的“通用串行总线”和硬盘的“SATA”。

一般我们见到的是两种物理标准。**D型9针插头**，和 **4针杜邦头** 两种。

这是常见的**4针串口**，在电路板上常见，经常上边还带有杜邦插针。还有时候有第五根针，3.3V电源端。

由于是预留在电路板上的，协议可以有很多种，要看具体设备。

http://s13.sinaimg.cn/middle/6566538dt9b5fccf2db43&690

下面这个就是**D型9针串口**(通俗说法)。在台式电脑后边都可以看到。

记住，这种接口的协议只有两种：RS-232和RS-485。不会是TTL电平的(除非特殊应用)。

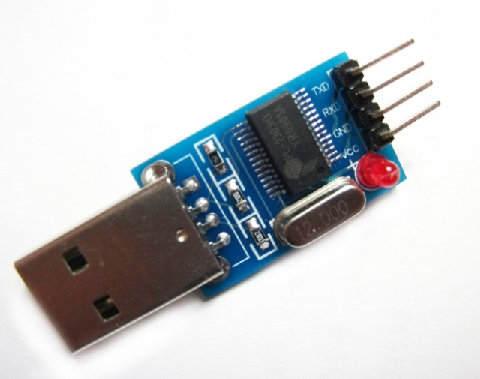
9针串口的定义可以参考这里：<http://wenku.baidu.com/view/5c170c6925c52cc58bd6be6e.html>

我们一般只接出RXD TXD两针，外加GND。



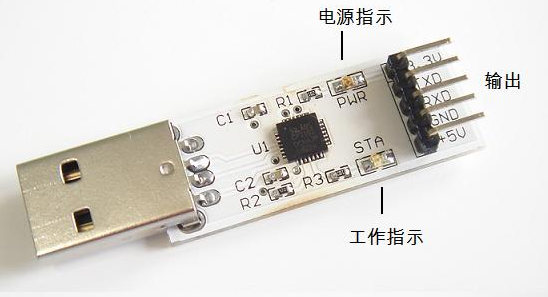
下图是个USB转TTL串口的小板，可以用USB扩展出一个串口。芯片为PL2303HX。

网上经常混淆各种串口，但是这个确实是可以给STC单片机下载程序的。



这是另一种，CP2102芯片的，也是USB转TTL串口。

据说比PL2303的好，实际使用中没感觉出来。这个小板就多了+3.3V电源端，以适应不同的目标电路。

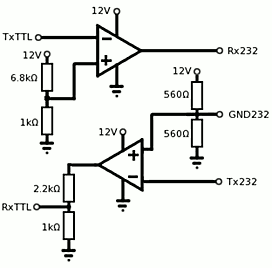


上边介绍的都是USB转TTL串口，如果目标设备上是RS-232串口(D型9针接口)咋弄？

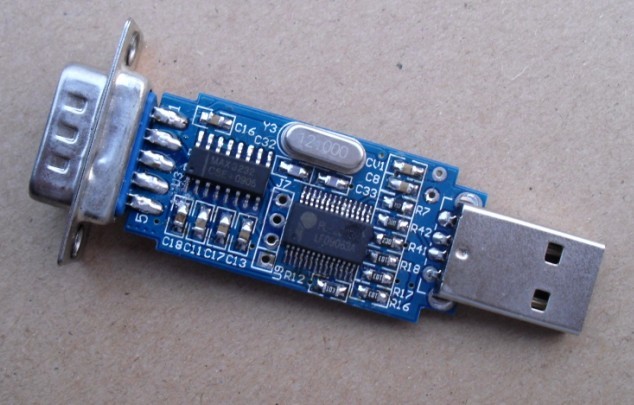
再接一片MAX232转换一下就行。

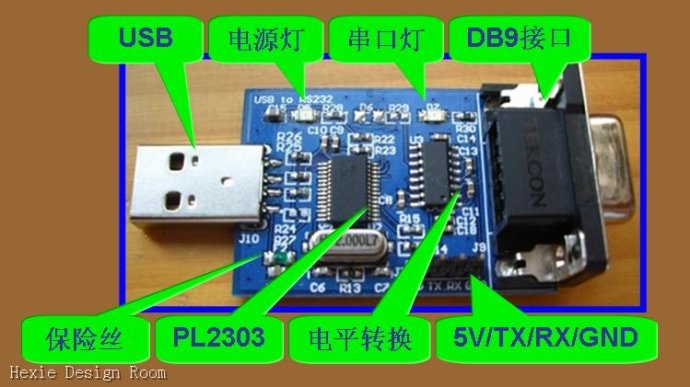
你也可以搭一个简单的比较器电路，来实现TTL转RS-232的功能，如下图。

RS-232转TTL咋弄？这就需要你动点脑子咯

[](http://photo.blog.sina.com.cn/showpic.html#blogid=6566538d0100r7p8&url=http://s16.sinaimg.cn/orignal/6566538dt9b606e9d320f)  
当然，早有人想到了做一款成品。

仔细看下图，USB经过PL2303转成了TTL串口，中间那四个窟窿可以引出，再经由MAX232转换为RS-232电平，9针串口引出。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=6566538d0100r7p8&url=http://s6.sinaimg.cn/orignal/6566538dt9b5fc04f7465)

下面这是另一款：电平转换依旧用的是MAX232。  
[](http://photo.blog.sina.com.cn/showpic.html#blogid=6566538d0100r7p8&url=http://s3.sinaimg.cn/orignal/6566538dt9b5fc0404732)  
  
  
  
你或许会买到这种：看起来里边只有一个牛屎芯片的。

但是记住一点，只要是 D型9针串口，不会是TTL电平的，没特殊说明就默认是RS-232。

所以这根线，不管里边构造怎样的，是 USB转RS-232串口 的线。  
[](http://photo.blog.sina.com.cn/showpic.html#blogid=6566538d0100r7p8&url=http://s8.sinaimg.cn/orignal/6566538dt9b5fc06df457)  
  
**我想要强调的是，一个设备的串口可以根据引出的串口线来判断是TTL还是RS-232，从而决定链接方式以及是否需要转换电路。**

## **详解RS232、RS485、RS485、串口&握手**

[EDN电子技术设计](javascript:void(0);) *前天*

**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，所以又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是可以恢复。   
  
**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。

**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。

END

**推荐阅读：**

* [深度分析华为鸿蒙系统两大核心技术！](http://mp.weixin.qq.com/s?__biz=MjM5MzUxMTAwMg==&mid=2649734923&idx=1&sn=aa76d743f60b547303b2d199da7fe43f&chksm=be8efc3e89f9752870abf2838cf58caa8452732f8b3f0aa4755975714112c861ac848a4635a0&scene=21#wechat_redirect)
* [一文读懂PID控制算法（抛弃公式，从原理上真正理解PID控制）](http://mp.weixin.qq.com/s?__biz=MjM5MzUxMTAwMg==&mid=2649734802&idx=2&sn=b8a896404ebaf1dbe275ba4a281c3dcd&chksm=be8efda789f974b115ec328f16c2d662e4d9e233877480d7e81d66b7caba8f9fd6189ca47720&scene=21#wechat_redirect)
* [18650电池将退出汽车领域？将被哪种电池取代？](http://mp.weixin.qq.com/s?__biz=MjM5MzUxMTAwMg==&mid=2649734923&idx=2&sn=bec60cadb8e13ca28a56bb8ae256d0b9&chksm=be8efc3e89f97528bc5c27e813129ed2a328dc3f1a3fb88a897db0f40ac94185f05186c85425&scene=21#wechat_redirect)

阅读 1213

 在看8

[写留言](javascript:;)