

工业串口通信之有话好好说

原创文章，转载请注明出处。

更多实用资料请登录方正智芯官网：www.founderchip.com

作者：北岛李工

前两篇文章我们对工业上的串口通信（RS232\RS422\RS485）进行了简单介绍，知道了通信的甲乙双方如何在物理层面上建立起通信的链路。是不是有了这些条件通信的甲乙双方就可以进行沟通了呢？



其实还是不够的。这就好比两个人打电话，电话线路是通畅的，但是两个人说的不是同一种语言，一个人说中文，一个人说德文，说了半天一句没听懂，“你这是弄啥勒”？啪，电话挂掉了。

两个讲着不同语言的人是无法进行交流的，要想交流，就必须对语言进行统一，比如都用英文，这样两个人就能够进行沟通，愉快的玩耍了。

工业上的通信和日常说话其实是一样一样的，当我们把物理上的线路都正确的连接后，还需要对相互的之间数据的发送接收进行相应的规定，这种规定称为通信协议。

通信协议对数据报文进行了明确的定义，比如：帧的起始信号，结束信号，数据的长度，校验码等等。工业上的通信协议非常多，比如：Modbus总线协议；ProfiBus总线协议，ProfiNet总线协议，S7 通信协议；Interbus总线协议；汽车上用的最广泛的CAN协议；及很多基于CAN协议的如DeviceNet,CanOpen等等。有了这些协议，通信的甲乙双方（总线协议可以有很多通信小伙伴）就可以建立起连接，愉快的交换数据了。

今天我们要着重说的是串口通信的协议，称之为ASCII协议。该协议有这样一些参数：波特率，起始位，数据位，停止位，校验位。

1) 波特率 (baud rate)：是指每秒钟所传输的字符（也称为码元）的个数。注意这里的字符不是字节，不一定是8位。波特率的单位是 “Baud” 。

很多小伙伴容易把“波特率”和“比特率”进行混淆。比特是英文的“bit”的翻译，表示二进制的“位”。比特率是指每秒钟所传输的位的个数，单位为bps(bit per second)。假设一秒钟传输2个字节(一个字节有八个位)，比特率就是 $2*8=16\text{bps}$ 。

再回到波特率，假如波特率中的字符（码元）是这样定义的：1个起始位，1个停止位，8个数据位。那么码元的总位数(bit)= $1+1+8=10$ 。在波特率等于9600的情况下，传输的比特率= $9600*10=96000\text{bps}$ 。

2) 起始位：串行通信在空闲的时候，总线上的电平为高电平（逻辑1），开始传输数据时，要先把总线上的电平拉低一个时间单位（逻辑0），称为1个起始位。时间单位是与波特率相关的，波特率越高，时间单位的宽度越窄。起始位不需要设置，默认为1，是包含在数据位里的。

2) 数据位：表示传输的数据的位数，范围是5-8，通常是8位；注意，8位数据由于包含了一个起始位，其有效的数据位数为7位。

3) 校验位：

校验是数据传送时采用的一种校正数据错误的一种方式，通常分为奇校验和偶校验，串行通信中可以有如下几种校验位的选择：

3.1) 奇校验(Odd)：数据在传输过程中，为1的位 (bit) 的总数应为奇数；若不是奇数，则校验位=1；若是，则校验位=0；

3.2) 偶校验(Even)：数据在传输过程中，为1的位 (bit) 的总数应为偶数；若不是偶数，则校验位=1；若是，则校验位=0；

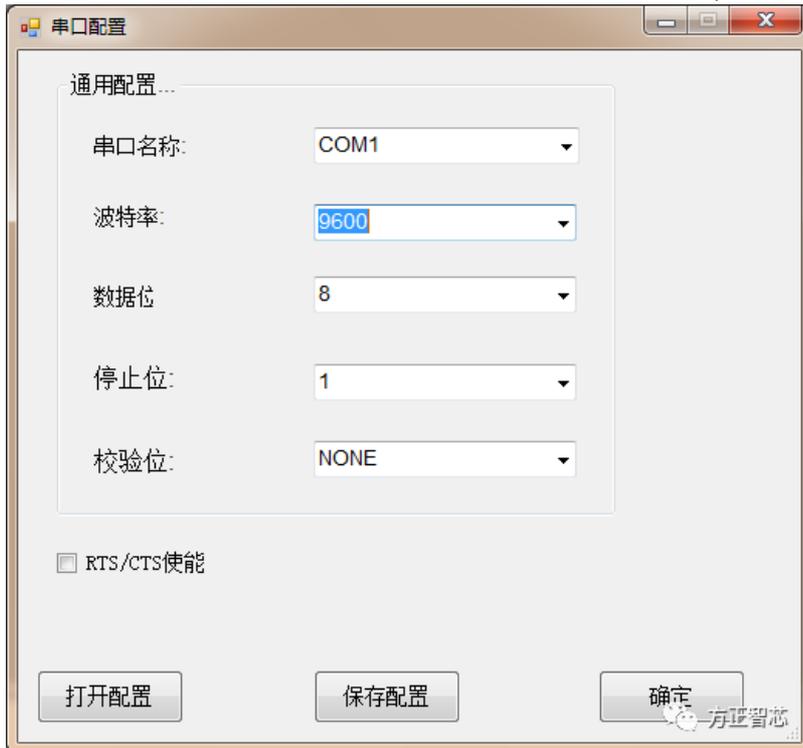
3.3) 空位校验(Space)：校验位=0

3.4) 标记校验(Mark)：校验位=1

3.5) 无校验(no parity)：不使用校验位

4.停止位：当一个字符的数据传输完成后，要把总线的电平拉高。停止位也是以时间长度来衡量的，通常可以选择“1”，“1.5”，“2”。以1.5为例，它表示停止位的高电平要保持1.5个时间单位的长度（取决于波特率）。停止位之后总线就转入空闲状态等待下次的起始信号。

下图就是方正智芯的串口调试软件“Como”的串口配置界面（我们后续文章将会讲到这个软件）。



这里注意的是“串口名称”这一项，计算机上可能有很多串口，COM1，COM2等，在这里选择你要使用的是哪个串口。

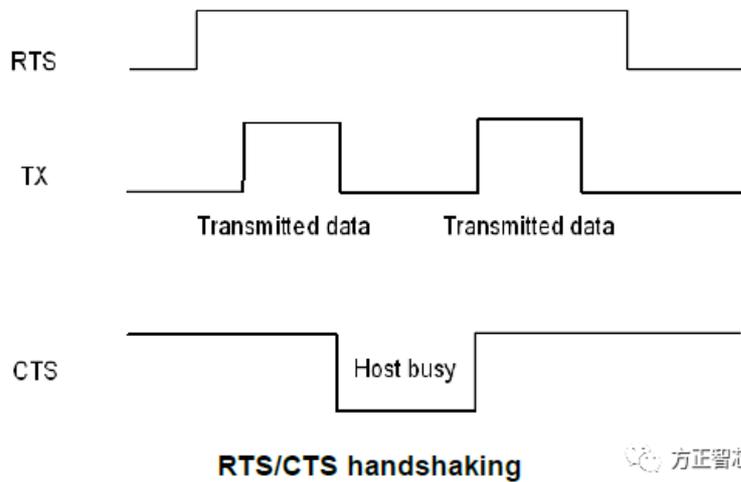
细心的小伙伴可能会发现，在上图的界面中，有个“RTS/CTR使能”的选项，这是什么东东呢？这就要说到串行通信的“流控制”（handshaking）了。

这里所说的“流”，是指数据流。数据在两个串口之间进行传输时，有时候会出现数据丢失的情况。因为虽然我们配置好了波特率，数据位等相关的信息，但是两个串口各自的数据处理速度很多时候是不一样的。比如台式机与单片机的通信，当单片机来不及处理台式机发来的数据时，就会造成数据缓存区满的情况。如果此时台式机继续发送数据，单片机就无法接收这些数据，从而造成数据丢失。

为了解决串行通信中甲乙双方数据处理速度不同造成的数据丢失的情况，聪明的工程师们提出了“流控制”的概念。流控制有两种：一种是硬件流控制，一种是软件流控制。

1) 硬件流控制。所谓硬件流控制，是指利用硬件线路的电气信号进行发送和接收的控制，硬件流控制必须用电缆将相应的控制引脚进行连接。例如当使用RTS\CTS这一对硬件流控制，首先要将通信甲乙双方的RTS和CTS进行连接（如果你不清楚RTS和CTS是什么含义，请参考文章：工业串口通信之掀起串口的盖头）。

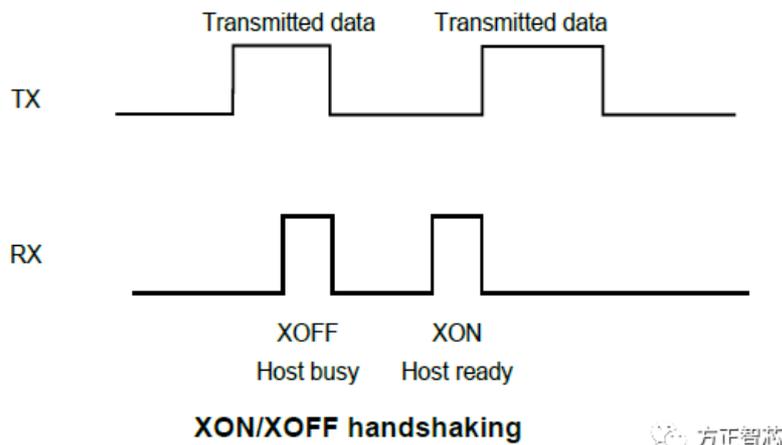
下图是Datalog 扫码枪硬件流控制（RTS\CTS）的示意图：



当扫码枪要发送数据时，先将RTS的电平拉高，然后检测CTS的电平，如果CTS为高电平，则启动数据的发送。如果CTS为低电平，则暂停数据的发送。当数据发送完毕时，将RTS的电平拉低。
 硬件流控制还可以使用DTR和DSR这一对连线的信号，和RTS\CTS类似，不再赘述。

2) 软件流控制

硬件流控制由于需要相应的电缆接线，会增加项目的成本。在某些对控制要求不是很严格的场合，可以使用软件进行流控制。软件流控制一般通过XOFF和XON来实现。还以Datalog的扫码枪为例，如下图：



当上位机或PLC (Host) 的输入缓冲区内的数据超过设定的上限值时 (比如缓冲区大小的75%) ，上位机发送XOFF字符 (13 Hex, 表示16进制的13) 。扫码枪收到XOFF字符即停止数据的发送。当上位机或PLC (Host) 的输入缓冲区的数据恢复到正常范围时，上位机发送XON字符 (11 Hex, 表示16进制的11) ，扫码枪收到XON字符时，就继续发送数据。

软件流控制可以减少布线的成本，但是在设计程序时必须谨慎使用0x13(13 Hex) 和0x11(11 Hex)这两个字符作为数据包的内容，否则很有可能引起软件的误操作。

这篇文章我们详细介绍了工业串口通信中各通信参数的含义及如何配置，并对通信过程的“流”控制进行了讲解，相信到目前为止，你已经对串口通信有了不错的理解了。下一篇文章我们将讲解PLC是如何接收发送串行数据的，敬请期待“工业串口通信之PLC的串口模块”

长按扫描下面的二维码或登录www.founderchip.com，关注“方正智芯”的原创文章，提供工业控制领域的技术和经验的分享，持续关注，持续进步。



方正智芯