

## IRCOMM Ver2.00 软件使用手册

### 1. 软件简介

IRCOMM Ver2.0是由北京异特路智能通讯科技有限公司开发的一款通用单任务串口测试软件。该软件小巧、实用、功能强大，兼具专业性与实用性，无须安装即可运行。同时，该软件是一款自由软件，完全免费，所有用户都可以从异特路公司的网站(www.itrob.com.cn 或 www.itrob.cn)自由下载，且没有期限限制,可以无限期使用。

注：

itRob是北京异特路智能通讯科技有限公司的商标

### 2. 软件相关信息

软件名称：IRCOMM Ver2.00  
 操作系统：Windows2000/XP及以上操作系统  
 软件大小：280KB  
 软件图标：

### 3. 界面说明

运行IRCOMM Ver2.0的可执行文件IRCOMM.exe后弹出窗口如图1所示：（此时串口没有打开）

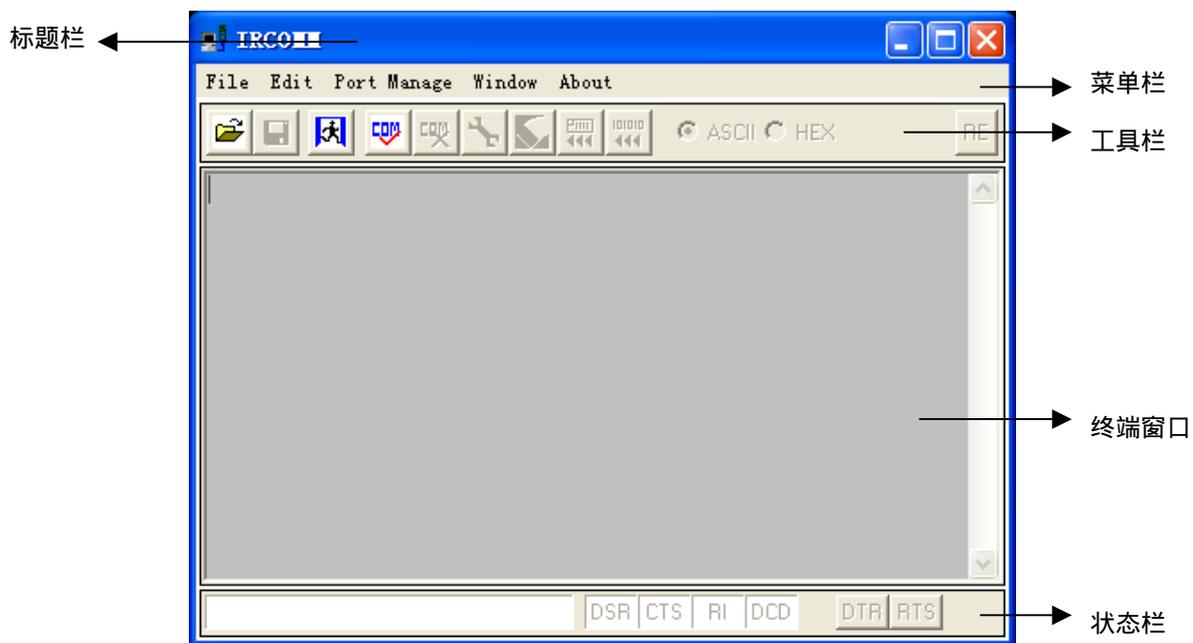


图1 IRCOMM Ver2.0运行后的弹出窗口

各部分功能如下：

标题栏：显示软件名称（IRCOMM）和程序图标。

菜单栏：包含IRCOMM Ver2.0的除终端接收显示允许按钮(RE)和DTR、RTS信号输出状态控制按钮以外的所有功能。

工具栏：提供最常用的串口操作。这些操作全都可以通过菜单栏命令实现，终端接收显示允许按钮（RE）除外。

终端窗口：当串口处于打开状态时显示从串口接收到的数据。

状态栏：当串口成功打开后显示串口的相关信息，这些信息包括：

串口号、波特率、数据位、校验方式、停止位、Modem输入信号状态（DSR、CTS、RI、DCD），以及Modem输出控制信号按钮（DTR、RTS）。

下面对每个部分内容做具体介绍。

## 4. 操作说明

### 4.1 标题栏

显示软件名称“IRCOMM”

### 4.2 菜单栏

主要包括5个菜单，分别是[File]、[Edit]、[Port Manage]、[Window] 和 [About]菜单。

下面对每个菜单的功能做具体介绍。

#### 4.2.1 [File]菜单

包括4个子菜单，分别是[Open csi File]、[Save csi File]、[Save csi File As]、[Exit]。

[Open csi File]子菜单：打开之前已经保存的串口设置信息文件（csi）命令。弹出文件打开对话框，如图2。

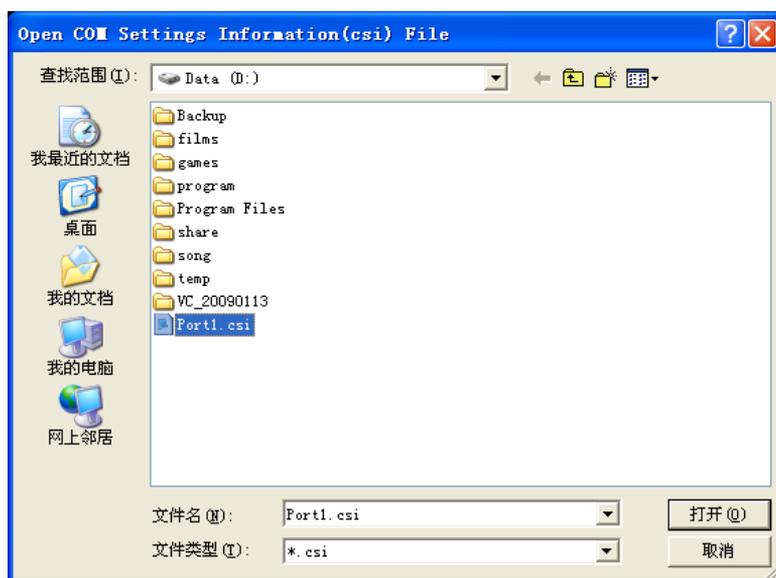


图2 打开csi文件对话框

选择之前已保存的csi文件名，点击打开按钮，IRCOMM会自动载入信息并打开相应串口。

[Save csi File]子菜单：将当前已打开串口的设置信息保存到csi文件中。如果之前已有打开/保存过的文件，则自动将信息保存到该文件中，否则将弹出文件保存对话框如图4所示：

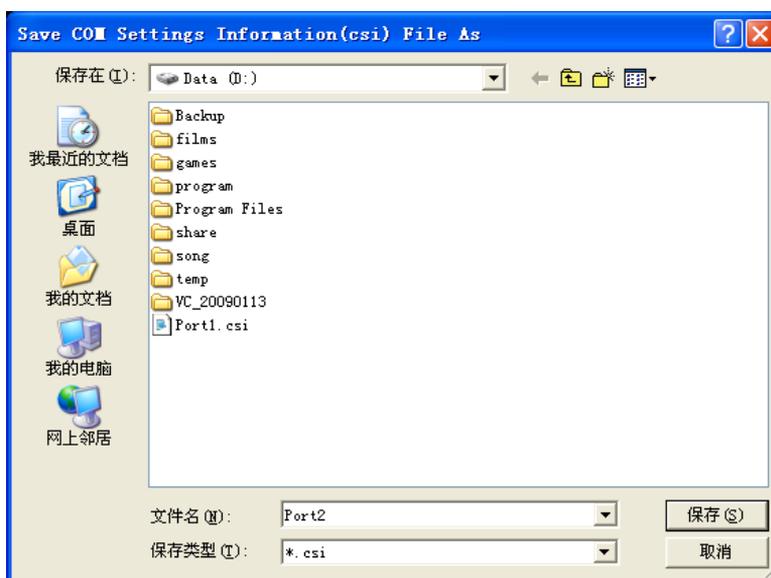


图4 保存csi文件对话框

注意：当用户保存csi文件时，在文件保存对话框中输入文件名后系统会在后面自动添加.csi扩展名。点击保存后，IRCOMM会自动将当前串口的设置信息保存于该文件中。



图3 载入csi文件成功对话框

[Save csi File As]子菜单：将已打开串口的设置信息保存于新的csi文件中。弹出对话框如图3所示。

[Exit]子菜单：退出IRCOMM程序命令。如果当前有正处于打开状态的串口，则IRCOMM会提示是否关闭该串口，如：当前打开的是串口COM1,则执行[Exit]会出现提示对话框如图5所示：

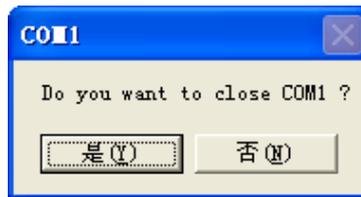


图5

说明：串口设置信息文件（COMPort Settings Information）简称csi文件，是IRCOMM Ver2.0使用的一种信息保存文件，该文件内保存了串口的各种设置信息，这些信息最初是通过IRCOMM的设置对话框进行设置的。这些对话框包括：**COM Set**、**Flow Control Settings**、**Timeouts Settings**、**Terminal Option** 和 **Data Transfer** 5个对话框。

以上5个对话框的设置信息会保存到csi文件中。（以上5个对话框的设置内容和方法将在后面介绍）用户可以通过打开之前已保存的csi文件来自动载入串口的设置信息，并自动打开串口，避免了重复输入设置信息所带来的麻烦，从而给用户带来方便。

#### 4.2.2 [Edit]菜单

包括2个子菜单，分别是[Copy] 和 [Font]。

[Copy]子菜单：将终端窗口中显示的串口接收到的信息选中并复制到系统剪贴板。

[Font]子菜单：设置终端窗口中的字符的字体、字型、大小、和颜色等。

说明：以上两个菜单项只有当串口处于打开状态时才可用。

#### 4.2.3 [Port Manage]菜单

包括9个子菜单，分别是[COM Open]、 [COM Close]、 [COM Set]、 [Flow Control]、 [Timeouts]、 [COM Information]、 [File Transfer]、 [Data Transfer] 和 [Terminal Option]。

[COM Open]（打开串口）子菜单：当串口没有打开时，该菜单项有效，串口打开后该菜单项无效。执行该命令后弹出COM Set对话框，如图6所示：

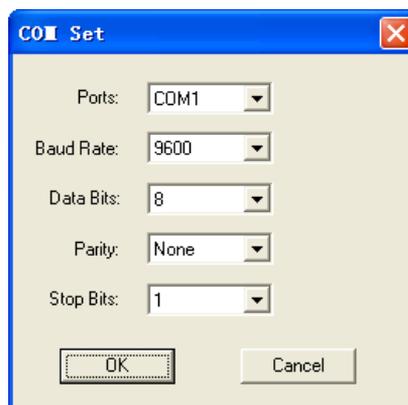


图6

在COM Set对话框中可以设置的项有**Ports**(端口号)、**Baud Rate**(波特率)、**Data Bits**(数据位)、**Parity**(校验方式)和**Stop Bits**(停止位)。

**Ports**(端口号)可以打开的串口范围从COM1~COM256,实际能否打开还要取决于计算机系统已安装的串口。

**Baud Rate**(波特率)下拉列表框中列出了从50~6000000bps范围内的常用波特率值，当列表中的波特率仍不能满足用户要求时，可通过列表最后一项User defin（用户自定义波特率）来直接输入想要的数值。当用户选择User defin列表框后，右边会自动出现  按钮和当前的用户自定义波特率数值，如图7所示。

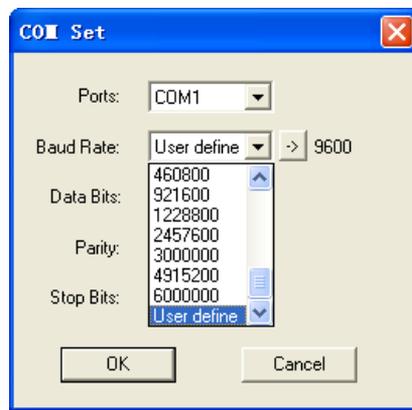


图7

当用户点击  按钮后会弹出User Baudrate Setting(用户自定义波特率对话框)，如图8所示：



图8 用户自定义波特率对话框

用户可直接输入想要的波特率数值（如12345）（在用户的串口硬件支持的前提下），并点击OK按钮，将自定义波特率写入Baud Rate下拉列表。如图9所示：

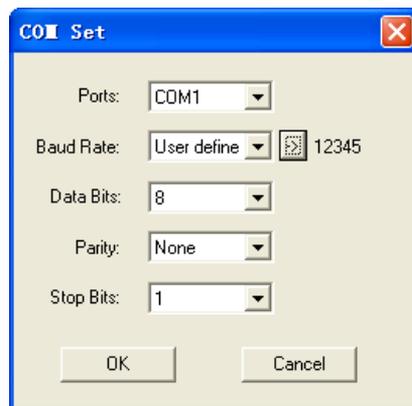


图9

**Data Bits**(数据位)下拉列表中的可指定的数据位有5、6、7、8。**注意！能否指定想要的的数据位还要看用户的计算机所安装的串口是否支持。**几乎所有的串口都支持8个数据位，但5、6、7个数据位有些串口可能不支持。

**Parity**(校验方式)下拉列表中的可选项有None(无校验)、Even(偶校验)、Odd(奇校验)、Mark(标志校验)和Space(空格校验)。

**Stop Bits**(停止位)下拉列表的可选项有1、1.5和2。具体设置能否成功还要看硬件是否支持，但一般情况下大多数串口都支持1位停止位。

当以上5项串口基本信息全都设置完成后，点击OK按钮打开Ports指定的串口。如果成功则程序的工作区（终端窗口）会自动由灰色变成白色。同时，在状态栏中会显示已打开串口的基本信息，如图10所示：

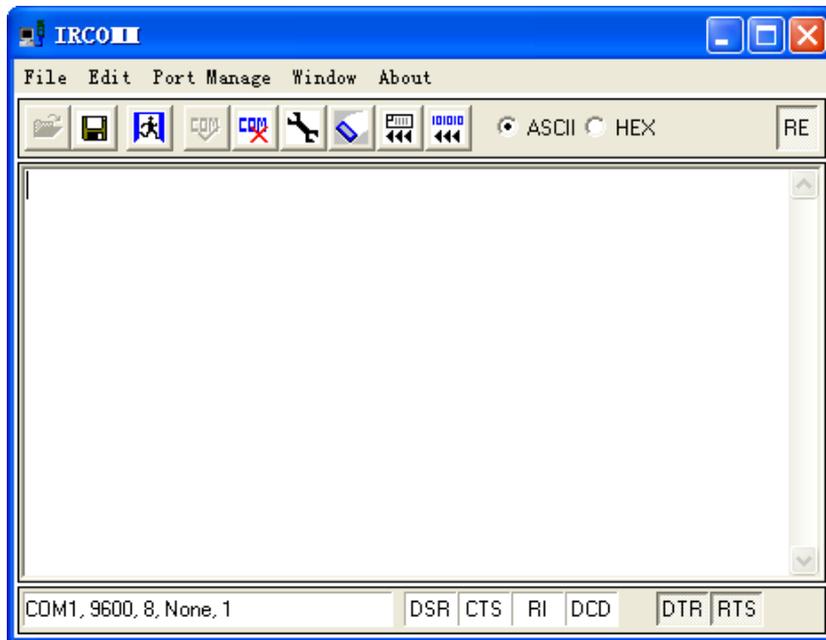


图10 串口成功打开后的主界面

**[COM Close]** (关闭串口)子菜单：关闭当前已打开的串口（该菜单项只有当IRCOMM有串口打开时才有效）。

**[COM Set]** (串口设置)子菜单：对当前已打开的串口的Baud Rate(波特率)、Data Bits(数据位)、Parity(校验方式)、Stop Bits(停止位)参数进行设置。执行该菜单命令将弹出COM Set对话框，和执行**[COM Open]**子菜单命令时所弹出的COM Set对话框相同，只是无法设置串口号(因为串口已经打开)。如图11所示：

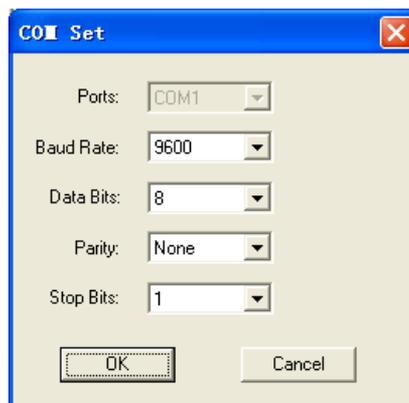


图11

**[Flow Control]** (流控设置)子菜单：对当前已打开串口的硬件流控和软件流控方式进行设置。执行该菜单命令将弹出Flow Control Settings对话框，如图12所示。下面对各设置项进行说明。

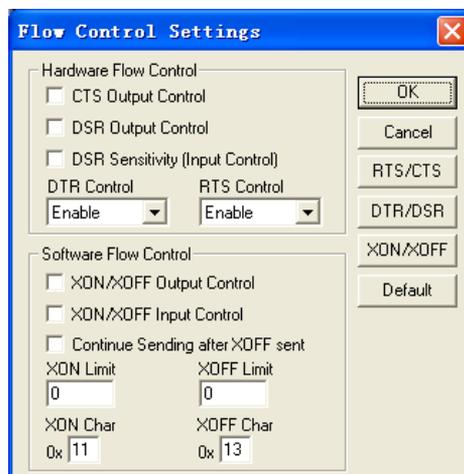


图12

说明：Flow Control Settings对话框中的各设置项直接对应串口DCB结构中的相应成员。关于串口DCB数据结构的详细信息请参见Microsoft公司的MSDN。

Hardware Flow Control(硬件流控)部分有5个设置项，分别为：

**CTS Output Control**(CTS输出流控)选择按钮：设置对象为DCB结构的fOutxCtsFlow成员。当该设置项选中时将使得fOutxCtsFlow=TRUE，表明此时CTS信号将参与发送过程的控制。既，当CTS信号状态为ON时允许发送，当CTS信号为OFF时发送被挂起；当该项未被选中时fOutxCtsFlow=FALSE，CTS信号将不参与发送过程的控制。

**DSR Output Control**(DSR输出流控)选择按钮：设置对象为DCB结构中的fOutxDsrFlow成员。当该设置项选中时将使得fOutxDsrFlow=TRUE，表明此时DSR信号将参与发送过程的控制。既，当DSR信号状态为ON时允许发送，当DSR信号为OFF时发送被挂起；当该项未被选中时fOutxDsrFlow=FALSE，DSR信号将不参与发送过程的控制。

**DSR Sensitivity(Input Control)**(串口通讯程序对DSR信号敏感)(输入流控)选择按钮：设置对象为DCB结构中的fDsrSensitivity成员。当该设置项选中时将使得fDsrSensitivity=TRUE，表明此时驱动程序对DSR信号敏感。既当DSR信号为OFF时，接收到的任何字节都被忽略，当DSR信号为ON时，接收才可正常进行；当该设置项未选中时fDsrSensitivity=FALSE，通讯程序将不具备以上特性。

**DTR Control**(DTR流控)下拉列表：设置对象为DCB结构的fDtrControl成员。该下拉列表共3个选项，分别为Disable、Enable 和 Handshake。

当选择Disable时，fDtrControl=0x00，表示禁止DTR信号，并保持禁止状态(OFF)。选中此项将导致状态栏的DTR按钮无效(变灰)；

当选择Enable时，fDtrControl=0x01，表示允许DTR信号，并保持允许状态(ON)。选中此项将导致状态栏的DTR按钮有效，此时可手动控制DTR信号的状态(通过点击状态栏的DTR按钮)；

当选择Handshake时，fDtrControl=0x02，表示允许DTR硬件握手。选中此项后，DTR的状态将受驱动程序自动控制，同时状态栏的DTR按钮无效(变灰)，既用户不能手动控制DTR信号的状态。

**RTS Control**(RTS流控)下拉列表：设置对象为DCB结构的fRtsControl成员。该下拉列表共4个选项，分别为Disable、Enable、Handshake 和 Toggle。

当选择Disable时，fRtsControl=0x00，表示禁止RTS信号，并保持禁止状态(OFF)。选中此项将导致状态栏的RTS按钮无效(变灰)；

当选择Enable时，fRtsControl=0x01，表示允许RTS信号，并保持允许状态(ON)。选中此项将导致状态栏的RTS按钮有效，此时可手动控制RTS信号的状态(点击状态栏的RTS按钮)；

当选择Handshake时，fRtsControl=0x02，表示允许RTS硬件握手。选中此项后，RTS的状态将受驱动程序自动控制(当接收缓冲区内的字节数少于缓冲区容量的1/2时，RTS将自动置为ON状态；当接收缓冲区内的字节数大于缓冲区容量的3/4时，RTS信号将自动置为OFF状态)，同时状态栏的RTS按钮无效(变灰)，既此时用户不能手动控制RTS信号的状态。

当选择Toggle时，fRtsControl=0x03，表示RTS进入套索模式。选中此项后，RTS信号线的状态将反映发送缓冲区内是否有等待发送的字节。既，当发送缓冲区内有等待发送的字节时，RTS信号置ON；当发送缓冲区内没有等待发送的字节时(发送缓冲区空)，RTS信号置OFF。

Software Flow Control(软件流控)部分有7个设置项，分别为：

**XON/XOFF Output Control**(XON/XOFF输出流控)选择按钮：设置对象为DCB结构中的fOutX成员。当选中该项时fOutX=TRUE，此时串口发送过程将采用软件流控方式，既当接收到XOFF时发送过程停止，当接收到XON字符时发送继续；当该项未选中时fOutX=FALSE，发送过程将不采用软件流控方式。

**XON/XOFF Input Control**(XON/XOFF输入流控)选择按钮：设置对象为DCB结构中的fInX成员。当选中该项时fInX=TRUE，此时串口接收过程将采用软件流控方式，即当接收缓冲区满时自动发送XOFF字符，当接收缓冲区空时自动发送XON字符；当该项未被选中时fInX=FALSE，接收过程将不采用软件流控方式。

**Continue Sending after XOFF Sent** (软件流控下缓冲区满时是否继续发送)选择按钮：设置对象为DCB结构的fTXContinueOnXoff成员。当该设置项选中时，fTXContinueOnXoff=TRUE，此时，当接收缓冲区满，并且已经发送出XOFF字符后，发送过程继续；当该设置项未选中时，fTXContinueOnXoff=FALSE，此时，当接收缓冲区已满，并且已经发送出XOFF字符后，发送过程停止，直到接收缓冲区空，并且再次发送出XON字符后，发送过程才可继续。

**XON Limit** (软流控的接收缓冲区下限)输入框：设置对象为DCB结构的XonLim成员。该输入项的内容指定了接收缓冲区的下限(十进制)。当接收缓冲区内的字节数少于这个设定值时，将发送出XON字符。启动这一动作的前提如下：

- 1、XON/XOFF Input Control选项被选中
- 2、DTR Control下拉列表选择Handshake
- 3、RTS Control下拉列表选择Handshake

**XOFF Limit** (软流控的接收缓冲区上限)输入框：设置对象为DCB结构的XoffLim成员。该输入项的内容指定了接收缓冲区的上限(十进制)。但注意！XOFF Limit的数值并不是这个上限值，实际的上限值计算方法如下：

上限值(字节数)=接收缓冲区的容量(字节数)- XOFF Limit

当接收缓冲区内接收到的字节数大于这个上限值时，驱动程序将自动发送XOFF字符。

**XON Char** (XON 字符)输入框：设置对象为DCB结构的XonChar成员。该输入项的内容将指定XON字符(2位十六进制数)。

**XOFF Char** (XOFF 字符)输入框：设置对象为DCB结构的XoffChar成员。该输入项的内容将指定XOFF字符(2位十六进制数)。

**RTS/CTS** (RTS/CTS硬件流控)按钮：点击该按钮将导致CTS Output Control选择按钮被选中，同时RTS Control下拉列表选中Handshake项。点击此按钮的同时还将导致其它设置项载入缺省值。

**DTR/DSR** (DTR/DSR硬件流控)按钮：点击该按钮将导致DSR Output Control选择按钮被选中，同时DTR Control下拉列表选中Handshake项。点击此按钮的同时还将导致其它设置项载入缺省值。

**XON/XOFF** (XON/XOFF软件流控)按钮：点击此按钮将导致XON/XOFF Output Control 和 XON/XOFF Input Control选择按钮被选中。点击此按钮的同时还将导致其它设置项载入缺省值。

**Default** (载入缺省值)按钮：点击此按钮将使对话框内的所有设置项载入缺省值(默认值)。当用户打开串口后第一次打开Flow Control Settings对话框时，系统载入的就是缺省值。对于一般的应用，选择缺省值就可以了。

**说明**：1、上文中提到的DTR/RTS信号的ON/OFF状态分别表示DTR/RTS信号输出+3V~+15V或输出-3V~-15V。

**[Timeouts]** (超时设置)子菜单：对当前已打开串口的超时结构进行设置。执行该菜单命令将弹出Timeouts Settings对话框，如图13所示。下面对各设置项进行说明。

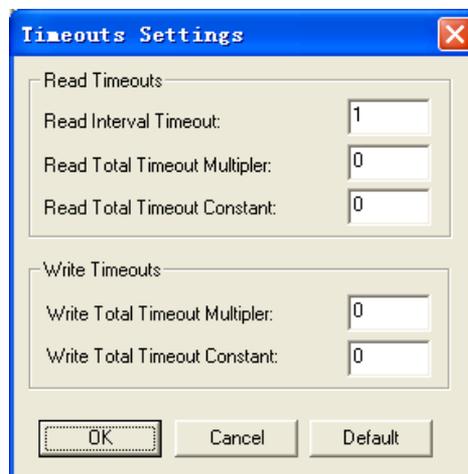


图13

说明：Timeouts Settings对话框中的各设置项直接对应串口COMMTIMEOUTS结构中的相应成员。关于串口COMMTIMEOUTS数据结构的详细信息请参见Microsoft公司的MSDN。

Read Timeouts(接收超时)设置栏中包括3个设置项，分别为：

**Read Interval Timeout**(读间隔超时)输入框：设置对象为COMMTIMEOUTS结构的ReadIntervalTimeout成员。

**Read Total Timeout Multiplier**(读总超时系数)输入框：设置对象为COMMTIMEOUTS结构的ReadTotalTimeoutMultiplier成员。

**Read Total Timeout Constant**(读总超时常量)输入框：设置对象为COMMTIMEOUTS结构的ReadTotalTimeoutConstant成员。

Write Timeouts(发送超时)设置栏中包括2个设置项，分别为：

**Write Total Timeout Multiplier**(读总超时系数)输入框：设置对象为COMMTIMEOUTS结构的WriteTotalTimeoutMultiplier成员。

**Write Total Timeout Constant**(读总超时常量)输入框：设置对象为COMMTIMEOUTS结构的WriteTotalTimeoutConstant成员。

**Default**(载入缺省设置)按钮：用户点击该按钮将使得本对话框的5个参数载入缺省值，具体为：

Read Interval Timeout=1

Read Total Timeout Multiplier=0

Read Total Timeout Constant=0

Write Total Timeout Multiplier=0

Write Total Timeout Constant=0

该缺省设置表示，当接收字符的间隔时间超过1ms时，接收过程立即返回，并且接收过程不使用总超时时间，同时发送过程也不使用总发送时间。

**[COM Information]**(串口信息显示)子菜单：显示当前已打开按钮的DCB、COMMTIMEOUTS 和 COMMPROP结构各成员数据的数值。图14是显示当前已打开串口DCB信息的示例。

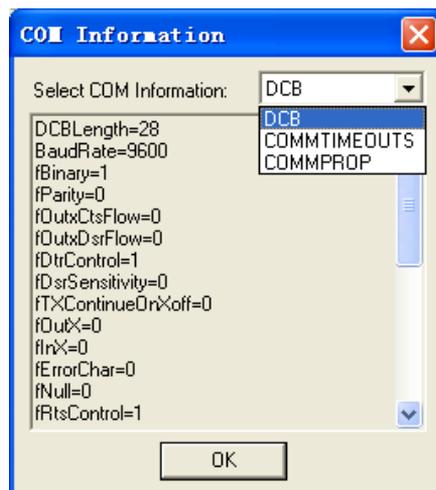


图14

说明：COM Information对话框的主要目的是方便用户查看当前已打开串口的DCB、COMMTIMEOUTS 和 COMMPROP结构各成员数据的数值信息，使用户更详细的了解串口各种信息，从而给调试和编程带来方便。该对话框无须用户做任何设置。

[File Transfer](文件传输)子菜单：执行该菜单命令将弹出File Transfer(文件传输)对话框。该对话框允许用户选择文件的Direction(传送方向),以确定发送文件还是接收文件。 File Transfer对话框如图15所示：

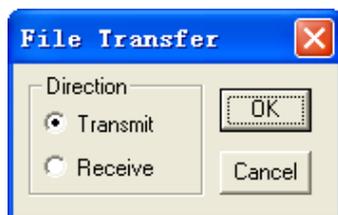


图15

对话框中有2个选择项，分别为：

**Transmit(文件发送)**选择项:当用户选择此项并点击OK按钮后会弹出Open Transmit File(打开发送文件)对话框，如图16所示：

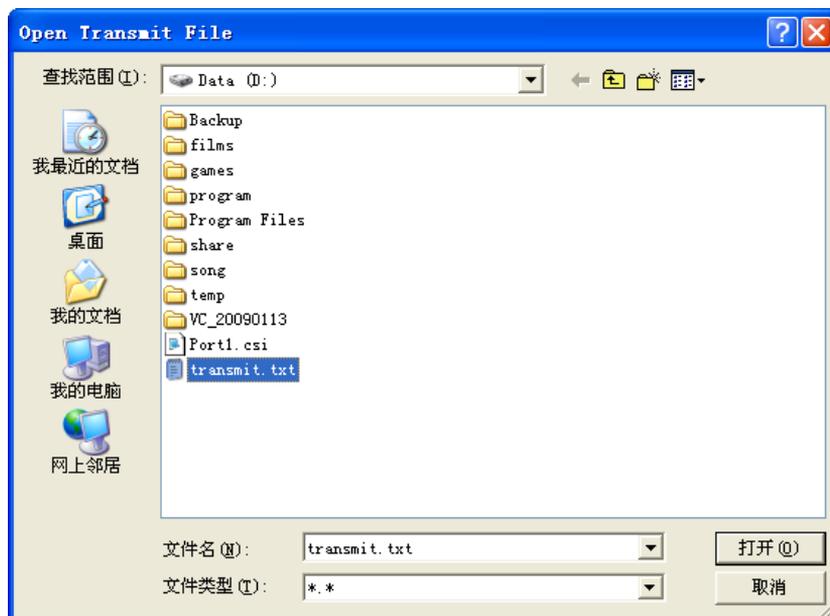


图16

在Open Transmit File对话框中，用户可以选择已存在的文件进行发送。文件的类型没有限制，可以发送任意类型的文件。**但注意！尽量不要发送可执行文件，IRCOMM Ver2.0在发送文件时采用的是直接传输方式，不具备出错重传机制！因此不能保证发送内容的正确性和完整性。**当然，在完全保证串口通讯没有误码的情况下，发送的文件是正确的。另外，发送的文件容量不能超过4G。但由于串口通讯速率的限制，不建议通过串口发送超过1M的文件(时间会很长)。

选择要发送的文件并点击**打开**按钮，将弹出File Transmit(文件发送)对话框，文件发送开始，如图17所示。

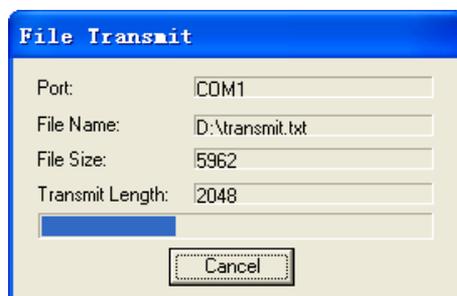


图17

在File Transmit对话框中将显示当前正在发送的文件的相关信息，包括Port(当前串口号)、File Name(发送文件的全路径名称)、File Size(文件大小(字节数))、Transmit Length(已发送的字节数)以及描述发送过程完成情况的进度条。如果用户想在文件还没发送完成前终止发送过程，可以点击**Cancel(撤消)**按钮，终止文件的发送。

当文件发送完成后将弹出消息框提示文件发送完成，如图18所示：



图18

**Receive(文件接收)**选择项:当用户选择此项并点击OK按钮后，会弹出Save Receive File(保存接收文件对话框)，如图19所示：

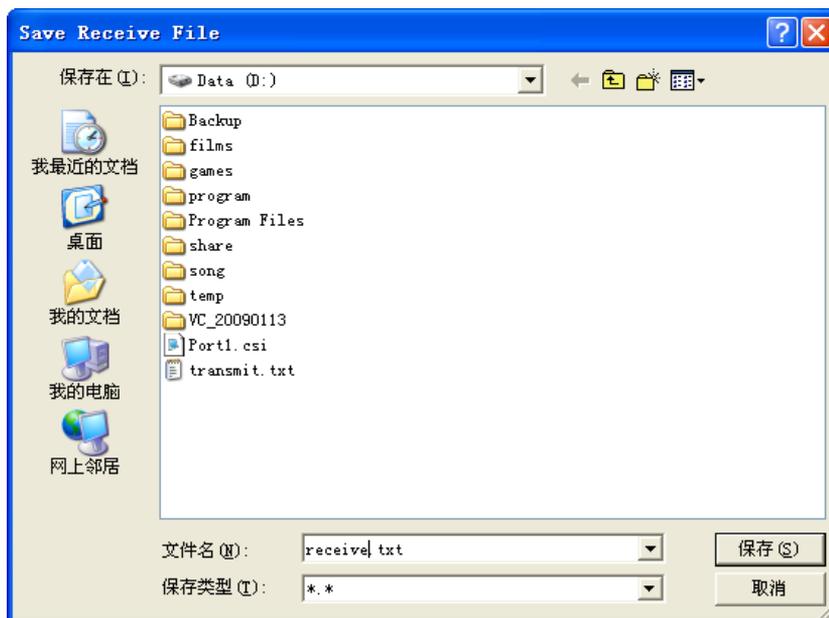


图19

用户可在**文件名(N)**输入框内输入要保存的接收文件的文件名，或选择一个已存在的文件名(此时，原有的文件内容会被清零，并会被接收内容覆盖)，然后选择**保存**按钮，此时会弹出File Receive(文件接收信息)对话框，如图20所示：

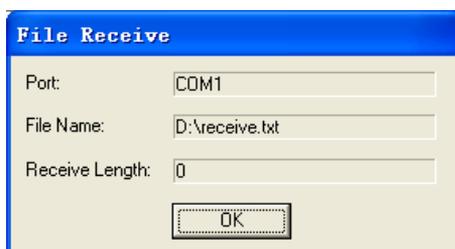


图20

File Receive对话框中显示当前文件接收过程的相关信息，包括Port(当前串口号)、FileName(保存接收数据的文件的全路径名称)、Receive Length(已接收的字节数)。

当弹出File Receive对话框后，文件接收过程启动，此时IRCOMM会将当前已打开串口接收到的数据保存到用户刚才指定的文件中。

**注意！如果串口超过30秒没有接收到任何数据(字节)，则文件接收过程将自动终止，并保存已接收的数据到文件中。**

说明：实现在两台计算机间传输文件的操作方法为接收方先启动文件接收过程，同时发送方在接收方启动接收过程的30秒内启动发送过程。

[Data Transfer](数据发送)子菜单：执行该菜单命令将弹出Data Transfer(数据发送)对话框，该对话框可指定要发送的数据内容、重复发送方式和发送间隔时间。如图21所示：

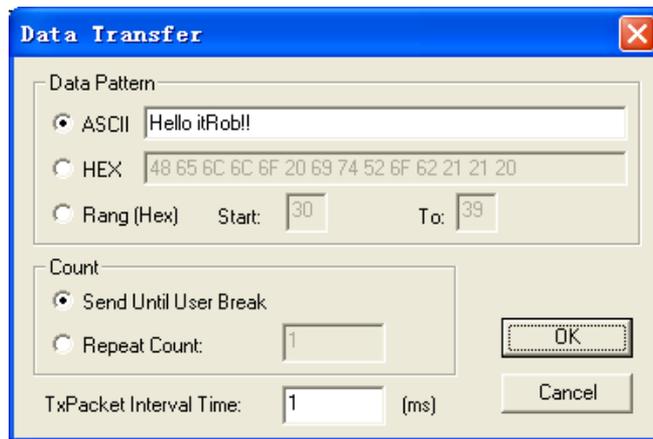


图21

Data Pattern(要发送的数据内容)部分主要包括3个选项，用户可通过这3个选项来选择3种格式之一来输入要发送的数据，分别为：

**ASCII**(ASCII格式输入)选项：选择该项可使用户以ASCII格式输入要发送的数据。对应的输入框内用户可直接输入ASCII格式的发送数据，也可以直接输入汉字。该框内的缺省内容为“Hello itRob!!”

**HEX**(十六进制格式输入)选项：选择该项将使用户以十六进制格式输入要发送的数据。

这里规定的十六进制格式包括2个要求：

一、字符要求，每个十六进制数包括数字1~9、小写字母a~f或大写字母A~F。

二、格式要求，每个字节必须用2位十六进制数表示(00~ff)，如ASCII码的‘a’的十六进制数为61、‘m’的十六进制数为6D或6d、空字符的十六进制数为00(不能写成0)，如果要表示多个字节则将每个字节对应的十六进制数依次表示，之间可以加1个或多个空格也可以不加。如“12345”的十六进制表示应为31 32 33 34 35(带空格)，也可以写成3132333435(不带空格)。当用户在HEX框中输入的HEX数据字符不符合要求或格式错误时，在点击OK按钮后IRCOMM会提示用户输入内容错误。

例1：12 348a 6B是正确的，313233 34 aB Ac是正确的。

例2：123 A4是错误的，AG 32是错误的，A 2D 4C是错误的。

HEX输入框的缺省内容为48 65 6C 6C 6F 20 69 74 52 6F 62 21 21 20，对应的ASCII字符串为“Hello itRob!!”。

**Rang(Hex)**(区域字节发送)选项：该选项可以让用户指定要发送的连续字节串的范围。其中Start为起始字节的十六进制数，To为终止字节的十六进制数。

例如用户想发送从00开始到FF结束的共256个连续字节串，则可以在Start框中输入00，同时在To框中输入ff即可。这里对Start和To输入框的输入内容也有要求，其中每个输入框内只能输入表示1个字节的2位十六进制数，同时Start输入框的内容不能大于To输入框的内容。否则，当用户点击OK按钮时IRCOMM会提示输入错误。

Count(重复方式)部分主要包括2个选项：用户可通过这两个选项来决定数据的重复发送方式。

**Send Until User Break**(连续发送直到用户终止为止)选项：该选项将使串口不间断的发送用户数据，直到用户终止发送为止。

**Repeat Count**(重复计数发送)：用户选择该选项后可输入想要重复发送的次数(十进制数)，当程序发送完指定次数的数据包后，发送过程自动停止。该输入框的内容不能为0，否则点击OK按钮后IRCOMM会提示输入错误。

**TxPacket Interval Time**(数据包间隔时间)：用户可在此输入框内输入要发送的数据包的间隔时间，单位为毫秒(ms)。这里要注意！如果每个数据包之间的时间间隔为T，发送数据包的时间为Tp，数据包的间隔时间为Ti，那么有如下关系式： $T=Tp+Ti$ 。如图22所示。该输入框的缺省内容为1ms。

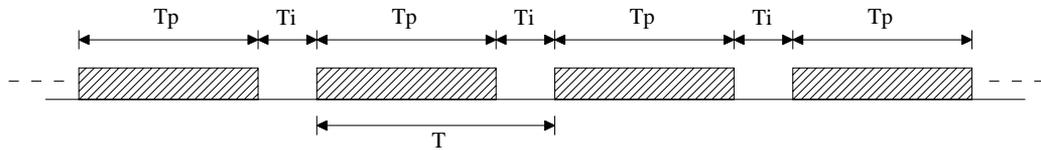


图22

比如：当串口发送一个数据包的时间 $T_p=20\text{ms}$ ，每个数据包之间的间隙时间为 $T_i=10\text{ms}$ ，那么每个数据包之间的实际间隔时间为 $T$ 为 $20+10=30\text{ms}$ 。TxPacket Interval Time输入框内输入的数值为包间隙时间 $T_i$ ，而发送每个数据包的时间 $T_p$ 则与当前串口的通讯速率(波特率)、数据位、校验方式、停止位数、以及数据包的大小(字节数)等因素有关。

**说明：**Data Transfer对话框集成了对用户输入信息进行错误检查的功能，如果用户输入的信息有错误时(如十六进制格式错误)，则当用户点击OK按钮时会弹出错误提示消息框，提示用户发生错误的原因。

当用户正确地输入了Data Transfer对话框内的所有信息并点击OK按钮后，发送过程启动，同时弹出Data Transmit Information(数据发送信息对话框)。在该对话框内将显示当前发送数据的过程信息，如图23所示：

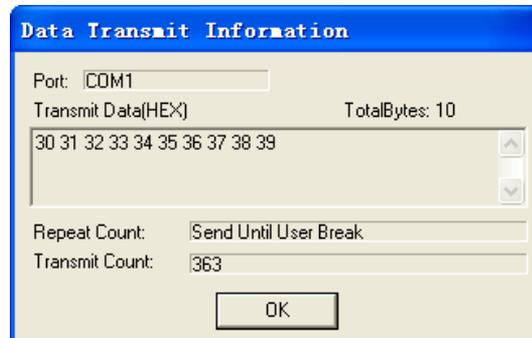


图23

Data Transmit Information对话框内显示的信息如下：

Port(当前串口号):显示当前已打开的串口号

Transmit Data(HEX)(发送的数据包的内容):以十六进制方式显示用户发送的数据包的内容，每个字节以2位十六进制数显示。

TotalBytes(发送数据包的字节数):用户在[Data Transfer]对话框内的Data Pattern部分输入的数据包(字符串)的大小(字节数)。

Repeat Count(重复发送方式):用户在Data Transfer对话框内的Count部分指定的重复发送方式。如果用户指定的是Send Until User Break选项，则Repeat Count框内显示“Send Until User Break”；如果用户选择的是Repeat Count选项，则Repeat Count框内显示用户指定的重复发送次数。

Transmit Count(已发送数据包计数):显示当前已经发送出去的数据包的数量。

当用户点击OK按钮或在重复计数发送方式下已发送数据包的数量达到用户指定的计数值时，发送过程终止，同时Data Transmit Information对话框关闭。

**说明：**Data Transmit Information对话框只是将当前发送过程的信息显示出来，而无需用户输入任何信息。

[Terminal Option](终端操作)子菜单：执行该菜单命令将弹出Terminal Option(终端操作)对话框，该对话框可指定终端发送(键盘发送)和接收显示方式(以ASCII/HEX方式显示接收到的数据以及回车字符的显示方式)。如图24所示：

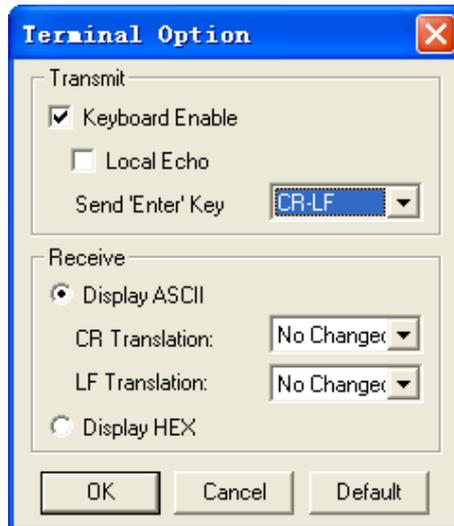


图24

Transmit(终端发送)部分包括3个设置项，分别为：

**Keyboard Enable**(键盘发送允许)选择按钮：用户选择该选项将使键盘发送有效，此时当用户敲击键盘时，可以将键盘按键所对应的字符从串口发送出去。若该选项未被选中，则键盘发送无效。

**Local Echo**(本地环回)选择按钮：用户选择此项，则当敲击键盘时，在从串口发送键盘对应的字符的同时，该字符也在终端窗口显示出来。若此选项未被选中，则终端窗口没有自环功能。当**Keyboard Enable**选择按钮未被选中时，该设置项无效。

**Send 'Enter' Key**(回车键发送方式下拉列表)：该列表允许用户选择Enter键(回车键)的发送方式。有3个选项，分别为：CR-LF、CR、LF。其中CR(回行符)的十六进制表示为0x0d，LF(换行符)的十六进制表示为0x0a。

CR-LF：当用户选择此项时表示当用户按下Enter键时，将先后从串口发送出0x0d和0x0a两个字符。

CR：当用户选择此项时表示当用户按下Enter键时，将从串口发送出0x0d字符。

LF：当用户选择此项时表示当用户按下Enter键时，将从串口发送出0x0a字符。

(当**Keyboard Enable**选择按钮未被选中时，该下拉列表框项无效)

Receive(终端接收)部分包括2个选项，分别为**Display ASCII**和**Display HEX**。

**Display ASCII**(以ASCII方式显示接收到的数据)：选择此项将使终端窗口以ASCII方式显示接收到的数据。

该选项下包括2个下拉列表框，分别为：

**CR Translation**(对CR字符的解释)：有3个选项CR-LF、LF和No Change。

当选择CR-LF时，当接收到CR(0x0d)时，程序将该字符转换成0x0d和0x0a两个字符并插入到终端显示缓冲区中。

当选择LF时，当接收到CR(0x0d)时，程序将该字符转换成0x0a字符并插入到终端显示缓冲区中。

当选择No Change时，当接收到CR(0x0d)时，程序将不对其作任何转换直接插入到终端显示缓冲区中。

(当**Display ASCII**选择按钮未被选中时，**CR Translation**下拉列表框项无效)

**LF Translation**(对LF字符的解释)：有3个选项CR-LF、CR和No Change。

当选择CR-LF时，当接收到LF(0x0a)时，程序将该字符转换成0x0d和0x0a两个字符并插入到终端显示缓冲区中。

当选择CR时，当接收到LF(0x0a)时，程序将该字符转换成0x0d字符并插入到终端显示缓冲区中。

当选择No Change时，当接收到LF(0x0a)时，程序将不对其作任何转换直接插入到终端显示缓冲区中。

(当**Display ASCII**选择按钮未被选中时，**LF Translation**下拉列表框项无效)

**Display HEX**(以十六进制方式显示接收到的数据)：选择此项将使终端窗口以十六进制的方式显示接收到的数据。该显示方式下将不对接收到的任何字节作任何转换。此方式下用户可以最全面的了解串口接收到的数据。

说明：当用户选择**Display ASCII**选项时，当串口接收到NULL(0x00)空字符时将其直接丢弃，不会将其放入终端显示缓冲区中。

**Default** (载入缺省设置按钮)：用户点击此按钮会使Terminal Option对话框载入缺省(默认)设置，如图24所示。

#### 4.2.4 [Window]菜单

包括2个子菜单，分别是[**Toolbar**]和[**Clear Screen**]。

[**Toolbar**] (显示工具栏)子菜单：用户选择该菜单项将使IRCOMM的工具栏显示或隐藏。

[**Clear Screen**] (终端窗口清空)子菜单：选择该菜单项将使终端窗口中显示的内容(从串口接收的数据)清空。

#### 4.2.5 [About]菜单

用户选择此菜单将直接弹出About (关于)对话框，如图25所示。



图25

About对话框中将显示IRCOMM软件的相关信息，包括：

软件图标：(如图25)

公司LOGO：**itRob**

软件名称及版本信息：IRCOMM Ver2.00

公司名称：itRob Technologies Co.,Ltd

版权及设计人信息：Copyright© 2008 DB X.Q.

公司网址信息：[www.itrob.com.cn](http://www.itrob.com.cn) 和 [www.itrob.cn](http://www.itrob.cn)

#### 4.3 工具栏

工具栏上的所有按钮可以实现的功能全都可以通过菜单栏实现，除了**RE**(接收允许)按钮。

串口未打开时的工具栏如图26所示：



图26

串口打开后的工具栏如图27所示：



图27

下面依次介绍各按钮功能：

 打开之前已经保存的串口设置信息文件(csi)命令按钮：功能相当于[**Open csi File**]子菜单。

 将当前已打开串口的设置信息保存为csi文件命令按钮：功能相当于[**Save csi File**]子菜单。

 退出IRCOMM命令按钮：功能相当于[Exit]子菜单。

 打开串口命令按钮：功能相当于[COM Open]子菜单。

 关闭串口命令按钮：功能相当于[COM Close]子菜单。

 串口设置命令按钮：功能相当于[COM Set]子菜单。

 清空终端窗口命令按钮：功能相当于[Clear Screen]子菜单。

 文件传输命令按钮：功能相当于[File Transfer]子菜单。

 数据发送命令按钮：功能相当于[Data Transfer]子菜单。

 终端窗口数据接收允许(Receive Enable)命令选择按钮：该按钮是工具栏中唯一一个无法通过菜单实现其功能的按钮。该按钮处于选中状态  时终端窗口显示新接收到的数据；当该按钮处于未选中状态  时，终端窗口将停止显示新接收到的数据。

ASCII 和 HEX单选按钮：设置终端窗口以ASCII还是HEX方式显示接收到的数据。这两个单选按钮直接对应Terminal Option对话框中的Display ASCII 和 Display HEX单选按钮。

#### 4.4 终端窗口

终端窗口的主要作用就是显示串口接收到的数据。关于终端窗口的相关设置见Terminal Option设置对话框部分。

除了Terminal Option设置对话框的相关设置外，通过[Font]子菜还可以设置终端窗口内数据的显示字体、字型、大小和颜色等信息。如图28所示，字体为宋体、字型为常规、大小为10(磅)、效果无、颜色为红色，同时显示格式为HEX(十六进制)，串口接收的数据为从00到FF共256个字节时的终端窗口显示效果。

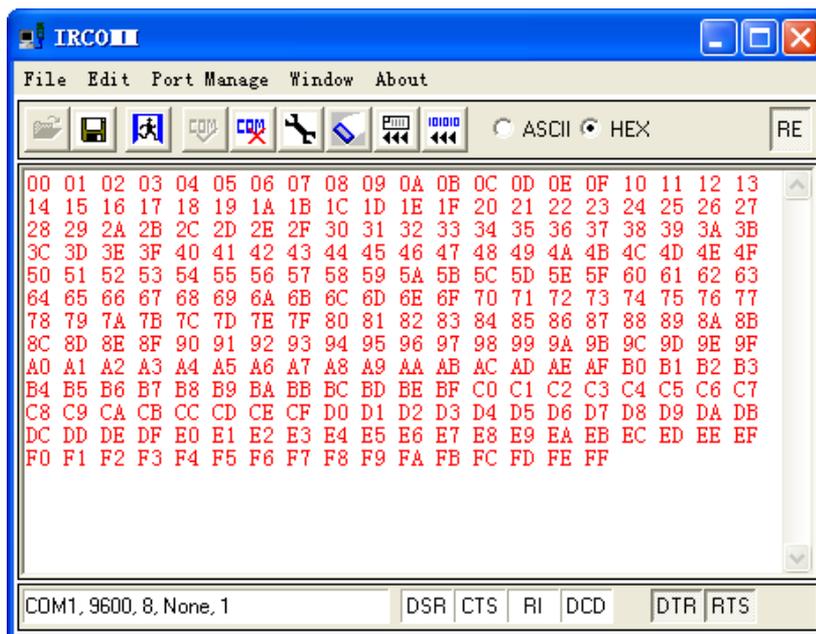


图28

说明：IRCOMM Ver2.0的终端窗口的显示缓冲区为2048个字节，当显示数据所需要的字节数超过2048时，终端窗口内的数据会滚动显示，每次滚动60个字符，之前接收到的数据将被丢弃。因此在ASCII格式下显示的数据数量要多于HEX格式下显示的数据数量。

终端窗口的另一个重要功能就是键盘发送功能，当键盘使能开关打开(Terminal Option对话框中的Keyboard Enable按钮为选中状态)时，用户可通过敲击键盘以实现单个字符的发送(先用鼠标单击终端窗口，然后敲击键盘)。如敲击键盘上的‘a’键，则串口会发送出0x61字符。

#### 4.5 状态栏

状态栏位于主窗口的最下端(如图1所示)。状态栏的主要作用是显示当前已打开串口的参数信息,也显示串口硬件握手信号的状态(DSR、CTS)以及Modem信号(RI、DCD)状态,同时用户可通过状态栏的DTR和RTS选择按钮来手动控制DTR和RTS信号的输出状态。举例:如图29所示



图29

如图29可知,当前已打开串口为COM1、波特率9600bps、8位数据位、无校验位、1位停止位,串口的输入硬件握手信号状态为DSR为OFF、CTS为ON,串口的Modem信号状态为RI为OFF、DCD为OFF,串口的输出硬件握手信号状态为DTR为OFF、RTS为ON。

说明:当串口处于打开状态时,DTR和RTS按钮是否可用还受到Flow Control Settings对话框内硬件流控的设置方式影响,具体内容见[Flow Control](流控设置)子菜单部分。

## 5 应用举例

### 5.1 RS-232自环测试

RS-232自环测试是最常使用的一种测试方法,其主要目的是测试当前的RS-232串口收发是否正常,从而确定该串口的基本收发能力。自环测试的原理是将计算机RS-232口的第2针RxD(接收)和第三针TxD(发送)短接,运行串口测试软件从该串口发数据,同时查看该串口是否同时接收到相同的数据,从而判断该串口的发送和接收功能是否正常。

步骤一:准备一台普通台式计算机(含至少1个标准的RS-232接口)。

这里先对标准计算机的RS-232串口的物理接口及针脚定义作一简单介绍,计算机的RS-232的最常见的物理接口为DB-9M(针)接口,如图30所示,各针脚定义如表1所示。

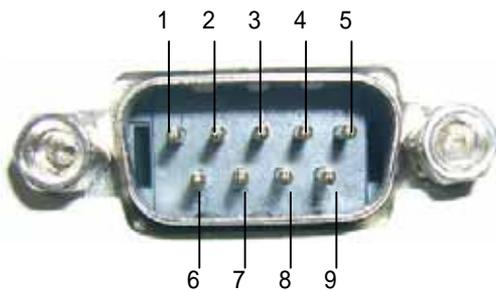


图30 DB-9M(针)接口

表1 DB-9M(针脚定义)

针脚编号	定义	说明
1	DCD	数字载波检测(输入)
2	RxD	接收(输入)
3	TxD	发送(输出)
4	DTR	数字终端准备就绪(输出)
5	GND	信号地
6	DSR	数字设备准备就绪(输入)
7	RTS	请求发送(输出)
8	CTS	清除发送(输入)
9	RI	振铃指示(输入)

步骤二:制作自环线。准备一个DB-9F(孔)直插头,焊接一根短路线在其对应的2脚和3脚之间,这样RS-232自环线就制作完成了,如图31所示。(为了简单也可以在串口发送期间用导体(电线或金属片)临时将计算机上的DB-9M(针)口的2针和3针短路的方法,但这样做容易导致短路,还是建议用户采用自制自环线)

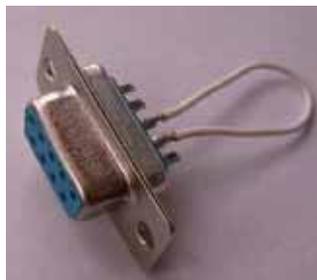


图31 自制自环线

步骤三：将制作好的自环线直接插入计算机的RS-232口。(假设这个RS-232口为COM1)

步骤四：启动测试软件IRCOMM，打开COM1口，设置参数使用缺省值即可，如图6所示：

步骤五：发送数据。点击IRCOMM工具栏中的  按钮，弹出Data Transfer对话框，当前要发送数据的缺省内容为“Hello itRob!!”，直接点击OK按钮开始连续发送。

如果串口正常，那么IRCOMM的终端窗口会源源不断的接收到与发送相同的数据，如图32所示：

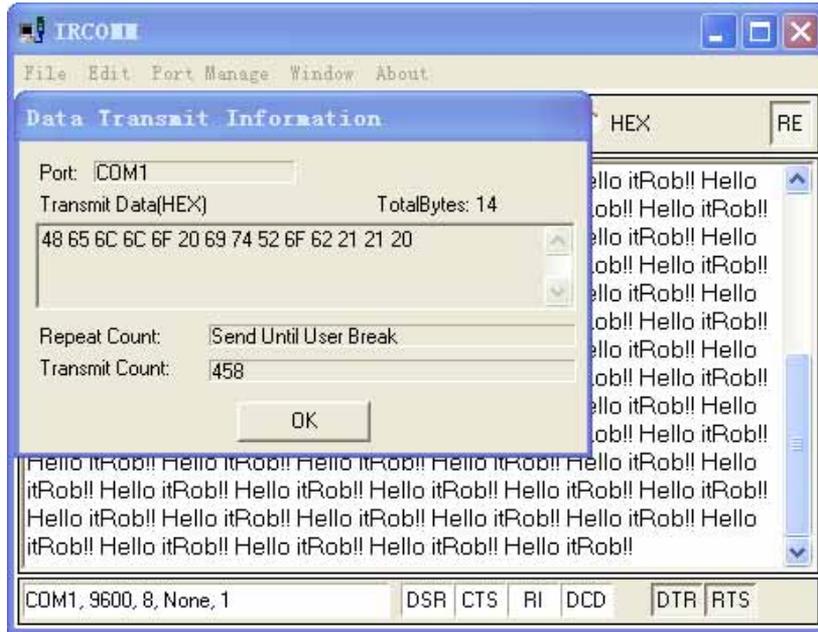


图32

### 5.2 RS-232硬件流控测试

这里只举例说明RS-232的RTS/CTS硬件流控的测试方法。(假设测试计算机有两个RS-232串口COM1和COM2)

步骤一：首先自制一条带RTS/CTS流控信号的交叉线两端都是DB-9F插头，将该交叉线连接于计算机的COM1和COM2口之间，使得COM1与COM2有如图33所示的连接。(图中没有画线的信号在这里不用)

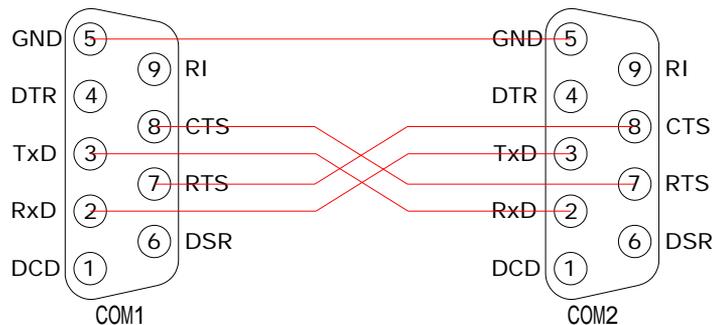


图33

步骤二：运行IRCOMM。由于每个IRCOMM程序进程只能打开一个串口，所以要同时测试COM1和COM2两个串口就需要运行2个IRCOMM程序。在两个IRCOMM中分别打开COM1和COM2(波特率、数据位、校验方式和停止位都采用缺省设置)。

步骤三：设置COM1的流控方式为RTS/CTS流控(直接点击Flow Control Settings对话框的RTS/CTS按钮即可，如图35所示)，同时，COM2的流控方式不用设置(采用缺省设置，既没有流控)。

由于COM1的流控方式设置为RTS/CTS流控，因此用户不能手动控制COM1的RTS输出状态(状态栏的RTS按钮不可用)。COM1的状态栏如图34所示：



图34

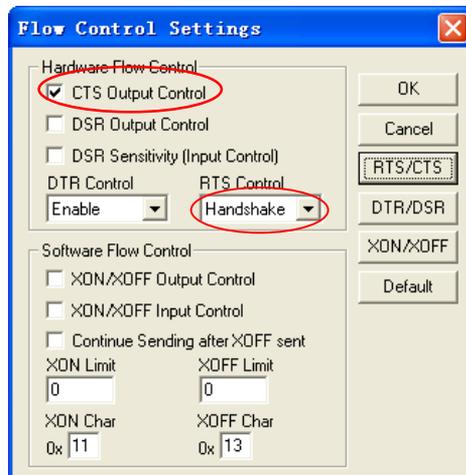
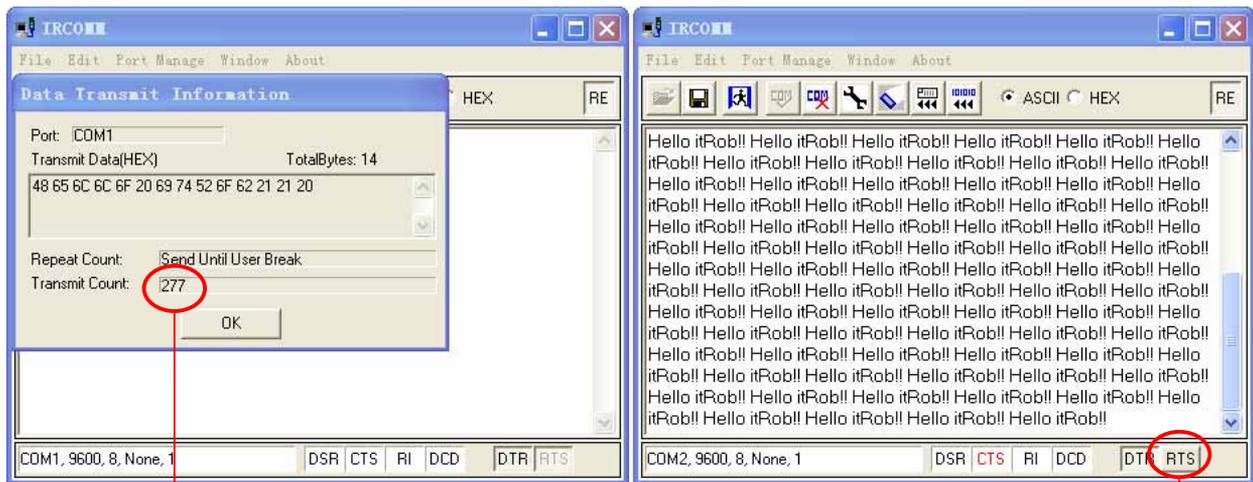


图35 COM1的硬件流控设置

步骤四：启动COM1发送。点击COM1所在IRCOMM窗口的  按钮，弹出Data Transfer对话框，直接点击OK按钮，开始连续发送。

如果正常，则COM2所在的IRCOMM窗口会源源不断的收到COM1发来的数据。此时，点击COM2所在IRCOMM窗口的状态栏上的RTS选择按钮以手动设置RTS输出信号的状态(ON/OFF)，看看会发生什么！

实际情况是，当用户手动设置COM2的RTS信号，使其输出ON状态时，COM1发出的数据会被COM2接收到，并源源不断地被COM2所在IRCOMM窗口显示出来。而当用户手动设置COM2的RTS信号，使其输出OFF状态时，COM2所在IRCOMM窗口将接收不到从COM1发来的数据，与此同时，COM1的Data Transmit Information(数据发送信息)对话框内的Transmit Count(发送计数)也停止了。这说明，不是COM2停止接收，而是COM1停止发送了，而导致COM1停止发送的原因正是COM2的RTS信号的ON/OFF状态。具体来说就是，当COM2的RTS信号输出ON时(导致COM1的CTS信号输入ON)COM1发送启动，而当COM2的RTS信号输出OFF时(导致COM1的CTS信号输入OFF)，COM1的发送停止。



当COM2的RTS信号输出OFF时，COM1发送停止。此时Transmit Count计数停止。

用户点击该按钮置COM2的RTS信号为OFF状态

图36

用户可调换COM1和COM2的流控设置重复上述测试步骤，也可以采用DTR/DSR硬件流控，其测试方法与RTS/CTS硬件流控类似。

说明：串口的流控动作并不是由用户程序实现的，而是由系统(串口驱动程序)自动完成的，用户程序所能做的只是改变串口的参数设置，一旦设置成功，之后的流控动作就由系统自动完成。

5.3 RS-232软件流控测试

简单说来，软件流控就是通过XON或XOFF字符来控制发送或接收过程的一种流控方式。由于软件流控在实际应用中很少使用，这里只简单举例说明如何通过XON/XOFF字符控制串口的发送(启动/停止)。

(假设测试计算机有COM1和COM2两个串口)

步骤一：自制一条三线交叉线，将该交叉线连接于计算机的COM1和COM2口之间，使得COM1与COM2有如图37所示的连接。(图中没有画线的信号在这里不用)

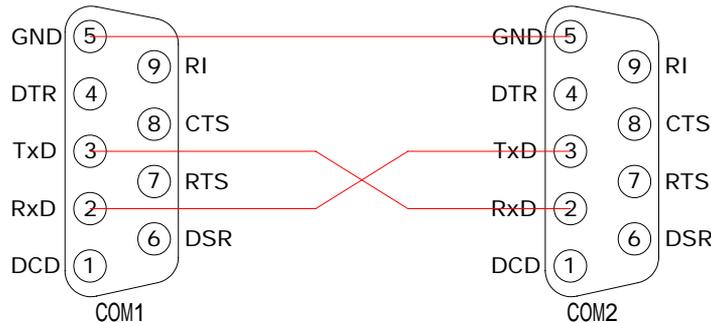


图37

步骤二：启动两个IRCOMM程序分别打开COM1和COM2这两个串口(先都使用缺省设置)。并且设置COM1的终端显示格式为HEX，这样做的目的是显示从COM2发来的XON和XOFF字符(实际情况是这两个字符不会显示出来)。

步骤三：执行COM1的Flow Control菜单项，弹出Flow Control Settings对话框中的XON/XOFF Output Control前选中该选项(如图38所示)，以使COM1的输出为软流控的工作方式，点击OK按钮完成设置。

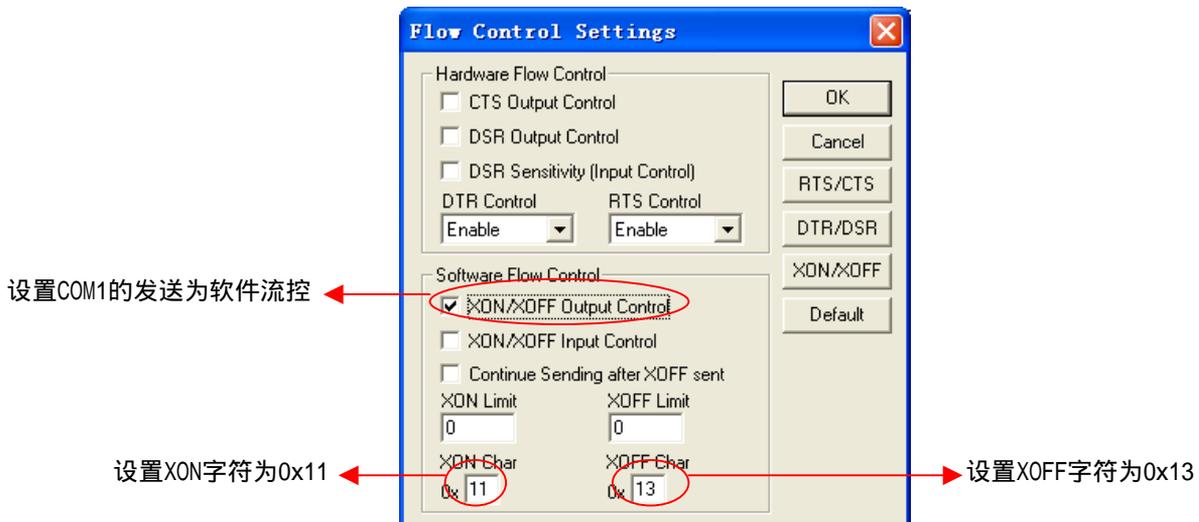


图38 COM1的流控设置

步骤四：启动COM1发送。点击COM1所在IRCOMM窗口的 按钮，弹出Data Transfer对话框，直接点击OK按钮，开始连续发送。如果正常，则COM2的终端窗口会源源不断的收到COM1发来的数据。此时，点击COM2的发送按钮 ，在弹出的Data Transfer对话框中点击HEX单选按钮，并在右侧输入框内输入13(就是XOFF字符)，同时再选择Repeat Count单选按钮，并在右侧输入框内输入1(只发送1次)，如图39所示，点击OK按钮。

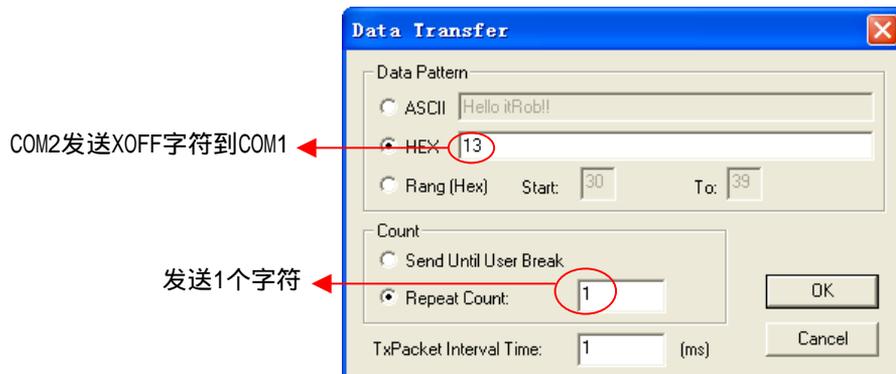


图39 COM2的数据发送

步骤五：现在看看COM1会发生什么！

实际情况是，当COM2发送出XOFF字符后，COM1的发送过程立刻停止了，COM1的Data Transmit Information(数据发送信息对话框)中的Transmit Count(发送计数)也停止计数了。如果要使COM1恢复发送，则需要COM2向COM1发送一个XON字符(0x11)。

以上事实说明，当COM1的发送设置为软件流控(XON/XOFF流控)时，COM1的发送动作要受从COM2发出的XON/XOFF字符控制。当COM1收到XOFF字符时发送过程停止(挂起)；当COM1收到XON字符时发送过程恢复。

**说明：串口设置成XON/XOFF软件流控时，被设置串口将忽略接收到的XON/XOFF字符。以上实验中，COM1的终端窗口并不显示从COM2发来的XON/XOFF字符，而其它字符则可以正常显示。此外，串口的XON/XOFF字符可以由用户在Flow Control Settings设置对话框中自行设定，如图38所示。**

本例只简单介绍RS-232串口输出软件流控的测试方法，输入软件流控的测试方法与之类似，但输入软件流控涉及串口接收缓冲区的操作问题，详细内容请参考Microsoft公司的MSDN。

#### 5.4 高速串口测试

IRCOMM Ver2.0的一项重要功能就是可以由用户任意指定想要的波特率(在硬件设备支持的前提下)，而不只是通过下拉列表选择，这意味着该软件可以测试高速串口设备，如：IR-1401XH系列产品，而这也正是异特路公司开发IRCOMM Ver2.0的初衷。

表2 异特路公司的IR-1401系列产品如下表所示：

产品型号	功能	隔离	速率
IR-1401A	USB转串口RS-232转换器	(无)	<1Mbps
IR-1401B	USB转串口RS-485/422转换器	(无)	<115.2kbps(常规)
IR-1401BG	USB转串口RS-485/422转换器	1500V光电隔离	<115.2kbps(常规)
IR-1401BH	USB转串口RS-485/422转换器	(无)	<3Mbps(高速)
IR-1401BHG	USB转串口RS-485/422转换器	1500V光电隔离	<3Mbps(高速)
IR-1401C	USB转串口RS-485转换器	(无)	<115.2kbps(常规)
IR-1401CG	USB转串口RS-485转换器	1500V光电隔离	<115.2kbps(常规)
IR-1401CH	USB转串口RS-485转换器	(无)	<3Mbps(高速)
IR-1401CHG	USB转串口RS-485转换器	1500V光电隔离	<3Mbps(高速)
IR-1401D	USB转串口RS-485/422/232转换器	3000V光电隔离	<115.2kbps(常规)

下面以异特路公司最有代表性的产品IR-1401BHG为例，介绍如何实现3Mbps的高速RS-485和RS-422通讯。

说明：由于要测试的RS-485的速率为3Mbps，而普通计算机自带串口的波特率最高为115.2kbps，因此无法用普通计算机串口转RS-485的方式测试。这里我们使用2个IR-1401BHG进行测试。

(假设测试计算机具备至少2个USB接口，同时已经安装好了IR-1401BHG的驱动程序)

步骤一：将两个IR-1401BHG插入计算机的USB接口，同时将两个IR-1401BHG的RS-485接口相连，如图40所示。

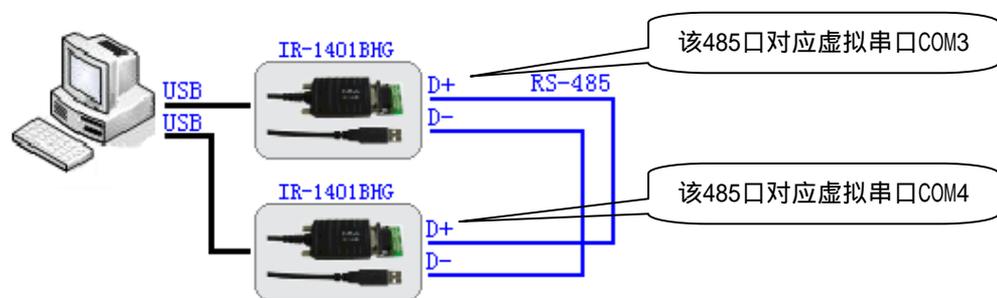


图40

步骤二：进入计算机的设备管理器，查看两个IR-1401BHG对应的虚拟串口编号，如图41所示。

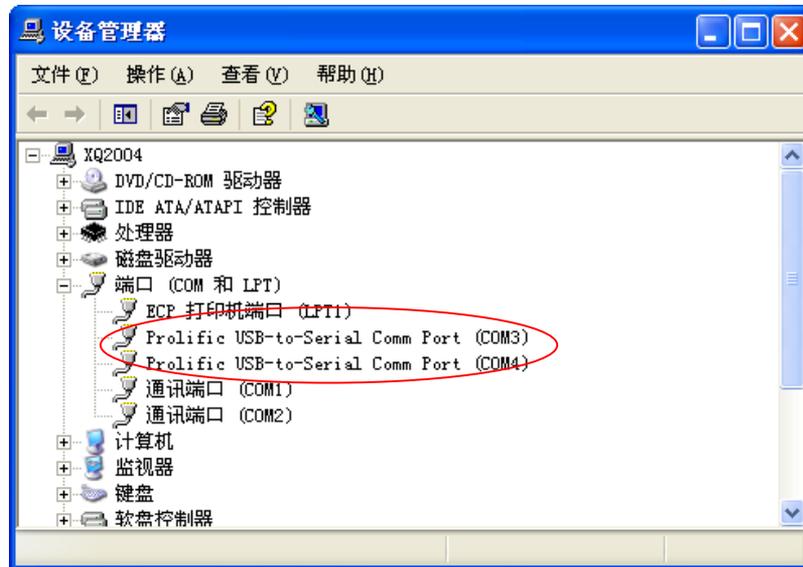


图41

由图41的设备管理器可知，两个转换器IR-1401BHGRS-485口对应计算机的虚拟串口号分别为COM3和COM4。

步骤三：运行IRCOMM分别打开COM3和COM4，同时将两个串口的波特率都设为3000000bps（既3Mbps），如图42所示。

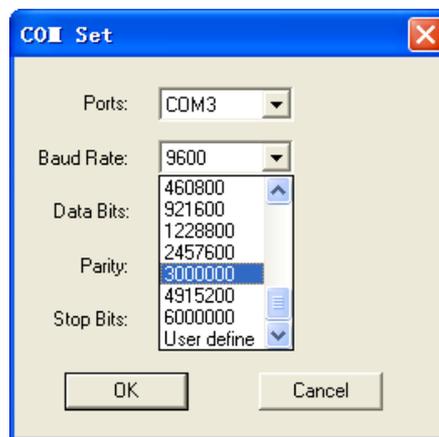


图42

步骤四：点击COM3的  按钮，弹出data Transfer对话框后直接点击OK按钮连续发送数据。如果正常，COM4口将源源不断地收到COM3发来的数据，并且无误码。如图43所示。

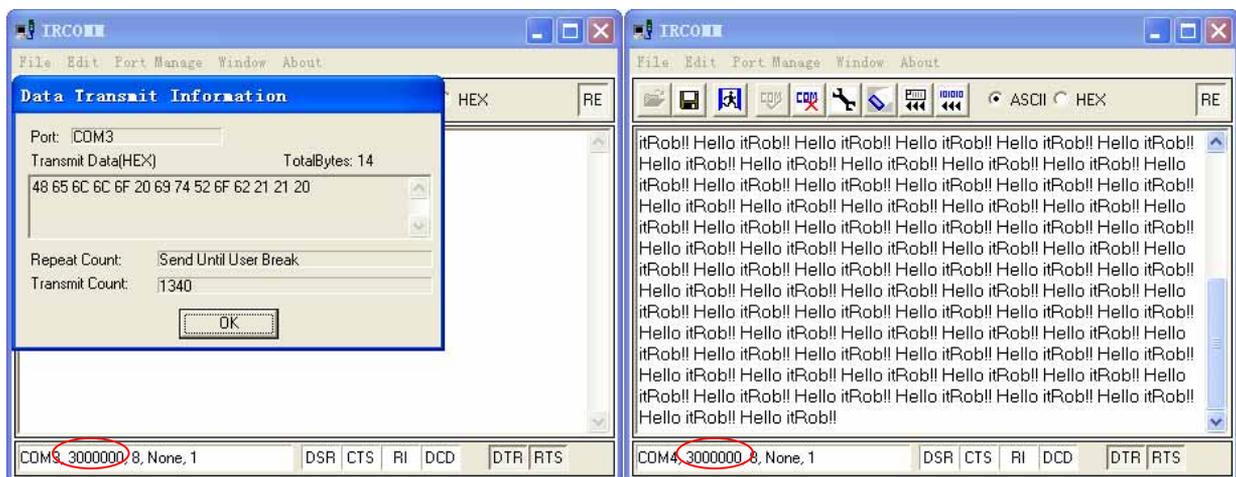


图43

改变数据发送方向，由COM4向COM3发送数据，同样，COM3会源源不断地收到COM4发来的数据，且无误码。

说明：

- 1、本测试中对虚拟串口COM3和COM4的波特率设置(3Mbps)是直接来自Baud Rate下拉列表中选择，也可以选择下拉列表最后一项User Def in项从而可以直接指定想要的波特率，如图8所示。
- 2、每个IR-1401BHG转换器对应的虚拟串口号可以在设备管理器里进行修改，可指定的串口号从0到256。
- 3、由于本测试是两线RS-485通讯，而两线RS-485的通讯是半双工的，当COM3向COM4发送数据的同时，COM4不能向COM3发送数据，只有当COM3停止发送后，COM4才可以向COM3发送数据(既收发不能同时进行)。否则RS-485总线会发生冲突，而且这种冲突是破坏性的(短路)，长时间工作会对RS-485接口造成损害(短时间的表现形式为通讯误码)建议用户尽量不要测试两线RS-485的全双工通讯。而RS-232和RS-422以及四线RS-485是全双工的。
- 4、本示例只测试了IR-1401BHG的RS-485通讯，没有测试RS-422通讯。关于IR-1401BHG的RS-422接线方式请参见IR-1401系列产品说明书。
- 5、关于异特路公司IR-1401系列USB转RS-485/422/232转换器产品的详细信息，请登陆异特路公司的网站www.itrob.cn或www.itrob.com.cn 获取。

**DO NOT COPY!!**

ITROB TECHNOLOGY DEPARTMENT

DT-CAAI-AD-BH DB X.Q.

Designed By: 

北 京 异 特 路 智 能 通 讯 科 技 有 限 公 司

电话：010-62977213 传真：010-62977237

WEB：www.itrob.cn 或 www.itrob.com.cn

E-mail: itrob@sina.com