

Embedded System

Covers Embedded system related topics Microcontroller and its peripherals, Basic linux and windows commands, UART,I2C,SPI, CAN etc communication system. Basic differences. Operating system basic fundamentals.

PAGES

[Home](#)
[Differences](#)
[Operating System concepts](#)
[Dos commands](#)
[Microcontroller Bootloader](#)
[Difference among RS 232, RS485 and RS422 and Pin Description](#)
[SVN : Source control Subversion](#)
[Basics of Embedded Linux](#)
[Symmetric multiprocessing](#)
[Embedded linux](#)
[check sum calculation](#)
[An overview of Linux Boot Process for Embedded Systems](#)
[Preprocessor Directives](#)
[The Embedded C++ specification](#)
[SPI and I2C protocol](#)
[Kernel Architecture](#)
[Linux device drivers](#)
[IPv6 address formation and conversion](#)
[Interrupts programmings articles](#)

Difference among RS 232, RS485 and RS422 and Pin Description

General RS232 information:

..... RS232 describes the standard for serial data communication between PCs and peripheral devices, such as modems, mice etc. and between PCs. Nowadays RS232 is replaced by USB more and more, but it is still an important standard, especially in the world of mobile devices, such as palmtop PCs and mobile phones.

There are two kinds of devices (from the viewpoint of RS232 communication): DTE and DCE, DTE stands for Data Terminal Equipment and means the "terminal devices", i.e. the PC or a simple terminal or in our case the palmtop; in general a device with a keyboard and a screen for manual data input and output. Printers are also DTE, because they act similar to a screen.

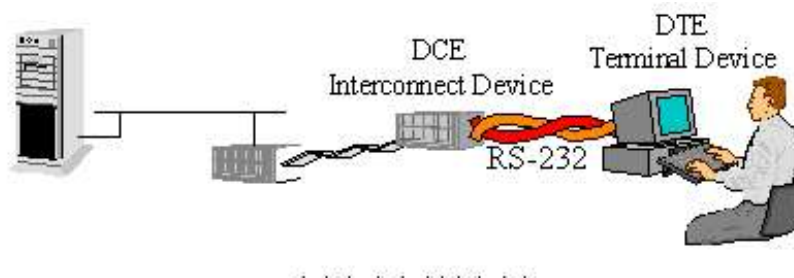
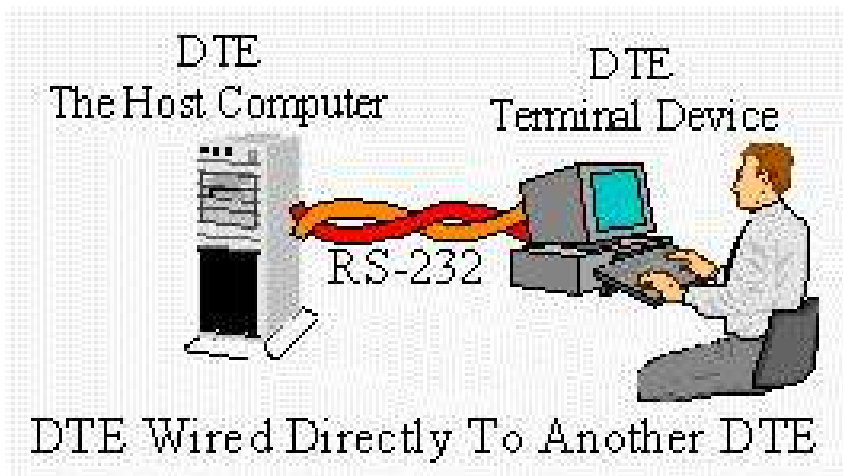
DCE stands for Data Communication Equipment, i.e. modems, mobile phones etc. To be honest, I cannot say for sure if mice are DTE or DCE. Does someone know?

You can connect two devices with RS232 in **three** combinations:

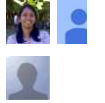
The usual connection is a DTE-DCE connection, e.g. if you connect a modem to your PC. This is done with a straight-through 1:1 serial cable.

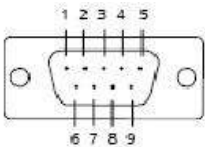
The other kind of common connections is a DTE-DTE connection, for instance, if you connect two PCs in order to exchange data between them. For that kind of connection you need a so-called null modem cable.

The third kind of connections is DCE-DCE. You need a so-called tail circuit cable for that, but since this is a very uncommon kind of connection, I won't describe it here.



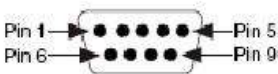
The above fig. represents pin description of RS485 of Db 9 connector

<http://1e>
[Người the](#)

[Theo dõi](#)
[Blog Ar](#)
[▼ 2019](#)
[▼ M: AT](#)
[► 2017](#)
[► 2015](#)
[► 2014](#)
[► 2013](#)
[► 2012](#)
[► 2011](#)



PIN	RS-232	RS-485 (4W)	RS-485 (2W)	RS-422
1	DCD	TxD-(A)	---	TxD-(A)
2	RXD	TxD+(B)	---	TxD+(B)
3	TXD	RxD+(B)	Data+(B)	RxD+(B)
4	DTR	RxD-(A)	Data-(A)	RxD-(A)
5	GND	GND	GND	GND
6	DSR	---	---	---
7	RTS	--	---	---
8	CTS	--	---	---
9	--	--	---	---

DB-9 Male



Pin Functions for RS-485 and RS-422

Data	
TXD+ (pin 8)	Serial Data Output (differential)
TXD- (pin 9)	Serial Data Output(differential)
RXD+ (pin 4)	Serial Data Input(differential)
RXD- (pin 5)	Serial Data Input(differential)
Handshake	
RTS+ (pin 3)	Request to Send (differential)
RTS- (pin 7)	Request to Send (differential)
CTS+ (pin 2)	Clear to Send (differential)
CTS- (pin 6)	Clear to Send (differential)
DSR (pin 6)	Data Set Ready
Ground	
GND (pin 1)	Ground

Serial Protocol Comparisons

Differences between the different serial protocols, RS232, RS423, RS422 and RS485:

RS232 uses single ended TX and RX. This means a common ground wire is shared between TX and RX. Only 3 wires are needed for a data only serial channel: TX, RX, and GND.

Disadvantages of single ended signaling is that it is more susceptible to noise than differential signaling, effective cable distances are shorter and data rates are slower.

The voltage ranges of RS232 signaling is +3V to +15V for a "high" and -3V to -15V for a "low".

RS423 uses single ended TX and differential RX. TX is comprised of 2 wires: TX+, a single ended signal, and TX- (which is really just GND). RX is comprised of RX+ and RX-, which go into a true differential receiver. So, 4 wires are needed for a data only serial channel: TX+, TX- (GND), RX+ and RX-.

The single ended TX+ pin drives the RX pin of an RS232 port, the TX- (GND) just ties to the RS232 GND pin and provides a common GND.

The voltage levels of signaling is +5V for a "high" and -5V for a "low", which is within the RS232 voltage level ranges mentioned above.

RS422 uses differential TX and RX. The TX pair of wires TX+ and TX- either drive TX+ to 5V and TX- to 0V for a "high", or TX+ to 0V and TX- to 5V for a "low". So TX- is not tied to GND as in RS423, instead it drives "opposite" the voltage that is on the TX+ pin. RS422 RX+ and RX- detect differential voltages just as in the RS423 case. 4 wires are needed for a data only serial channel: TX+, TX-, RX+ and RX-.

Since the TX pair of wires does not have a GND wire, RS422/RS485 can not be connected directly to an RS232 port. Attempting to do this (by connecting RS422 TX+ to RS232 RX and RS422 TX- to RS232 GND) would cause the RS422 TX- pin to be held at GND and the RX422 TX+ pin would drive 5V for "high" (which is valid RS232) and 0V for "low" (which is not a valid RS232 signal level).

Since RS422/RS485 are truly differential for both TX and RX, cable distances can be much longer and data rates much higher than RS232. If twisted pair cabling is used, (which makes sense because you end up with a complementary TX+ and TX- pair and RX+ and RX- pair), you reduce EMI emissions because the wire pairs help to cancel external transmissions. These wire pairs also reduce the impact of external electrical noise on the signaling.

Appletalk uses a variation of RS422. TX+ and TX- are "turned off" into a high impedance state where they are not driving a valid differential voltage. By doing this they enable multidrop capabilities where multiple parties can share a common set of serial lines. This is somewhat similar to RS485 described below, however Appletalk typically also includes an isolation transformer between the RS422 driver and the serial cable.

RS485 uses differential TX and RX as in RS422, however RS485 is a superset of RS422. It adds a slightly wider input voltage range on its RX pair, to make it less susceptible to noise and GND potential differences among RS485 devices.

RS485 also adds multidrop capability. There can be up to 32 RS485 devices on a single set of serial lines. This is achieved by having an "output enable" on the TX ports and allowing only one RS485 device to transmit at a time on a given pair of wires. No isolation transformer is needed as in Appletalk described above.

RS485 serial "networks" can be wired for 2 or 4 wire mode. In 2-wire mode, only one RS485 device can transmit at a time, this is known as half-duplex communication. An advantage to this topology is that you can save on cabling cost since the cable can be several thousand feet in length.

In 4-wire mode, a single RS485 device is designated as the "master" device. Its TX pair is connected to all the other "slave" devices' RX pairs, and all the slave's TX pairs are tied to the master's RX pair. This has the advantage of enabling the master device to transmit to the slaves while a single slave device may also transmit back to the master simultaneously. This is known as full-duplex communication. Another advantage of this topology is that the slaves are unable to inadvertently transmit to each other's RX pairs because they are not physically connected. This may make it easier to run slaves of different protocols on the same serial network.

In 4-wire mode, The MSS485's RJ45 serial port or screw terminal port can be connected directly to another RS422/RS485 port. The RS485 port cannot be connected directly to an RS232 port without an RS485-to-RS232 adapter for reasons described above in the RS422 section.

Since, like RS422, RS485 is differential it has similar cable length, data rate and noise immunity characteristics to RS422 described above.

RS 232 supports 2 types of flow control(handshaking).

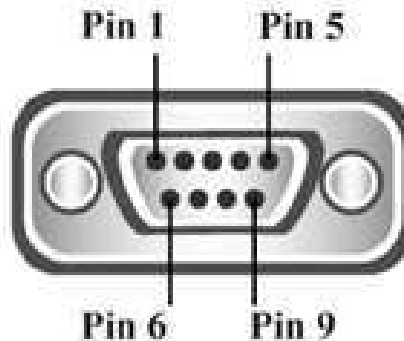
RS 485 -half duplex supports no flow control(handshaking).

RS 485 full duplex supports only software flow control(handshaking).

RS422/485

Pin 1	TXD-
Pin 2	TXD+
Pin 3	RTS-
Pin 4	RTS+
Pin 5	GND
Pin 6	RXD-
Pin 7	RXD+
Pin 8	CTS
Pin 9	CTS+

RS422/485 Pinout (9 Pin)



Handshaking

The method of exchanging signals for data flow control between computers and data sets is called handshaking. The most popular and most often used handshaking variant is called XON/XOFF; it's done by software, while other methods are hardware-based.

XON/XOFF

Two bytes that are not mapped to normal characters in the ASCII charset are called **XON** (DC1, Ctrl-Q, ASCII 17) and **XOFF** (DC3, Ctrl-S, ASCII 19). Whenever either one of the sides wants to interrupt the data flow from the other (eg. full buffers), it sends an **XOFF** *Transmission Off*. When its buffers have been purged again, it sends an **XON** *Transmission On* to signal that data can be sent again. With some implementations, this can be any character.

Xon/Xoff: This method can be used if no RTS/CTS lines are available. It is simple: If the receiver cannot handle more data (due to a buffer overflow, for example), it sends the Xoff character (Ctrl-S, ASCII 0x13) to the sender, so the sender knows it has to wait with further transmissions until the receiver sends the Xon character. If the receiver is ready again, it sends Xon (Ctrl-Q, ASCII 0x11) and the sender continues to send data.

XON/XOFF is of course limited to text transmission. It cannot be used with binary data since binary files tend to contain every single one of the 256 characters. That's why hardware handshaking is normally used with modems, while XON/XOFF is often used with printers and plotters and terminals.

DTR/DSR

The *Data Terminal Ready* and *Data Set Ready* signals of the serial port can be used for handshaking purposes, too. Their names express what they do: the computer signals with DTR that it's ready to send and receive data, while the data set sets DSR. With most modems, the meaning of these signals is slightly different: DTR is ignored or causes the modem to hang up if it is dropped, while DSR signals that a connection has been established.

They are only important at the beginning and the end of a data transmission. This is also called "handshaking", but doesn't mean the handshaking during the data transmission. See below.

DTR/DSR: These control lines assure that there are really two devices present which want to communicate with each other: First, the DTE sets DTR "high", then the DCE answers setting DSR "high".

These two lines are kept high during the whole transmission process. If the transmission has to be interrupted (because of a modem hang-up for example), the DCE drops the DSR to "low" so the DTE knows the transfer is finished.

Note: DTR and DSR are modem control signals.

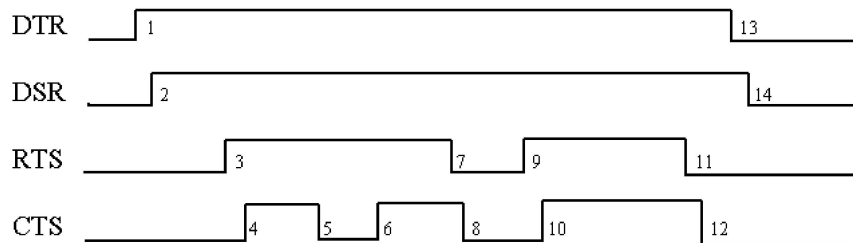
The DCD line is only used by modems. It is set to "high" by the modem to let the computer know it has detected the carrier tone of the modem on the other side (i.e. it has a connection to the other side).

RTS/CTS

While DTR and DSR are mostly used to establish a connection, RTS and CTS have been specially designed for data flow control.

RTS/CTS: These are the actual flow control lines: The sender sets RTS "high" (ready to send), the receiver answers setting CTS (clear to send) to "high". If the receiver needs a pause (buffer overflow, for example), it drops CTS to "low", so the sender knows it has to wait with further transmission until the receiver has set CTS "high" again. This way it is assured that no data is lost.

A normal handshaking protocol between a computer and a modem looks like this:

**Null Modem:**

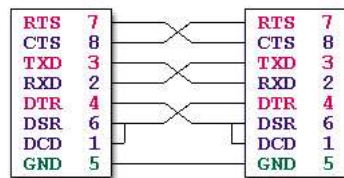
To connect two computers together, a special cable is required which mimics the behaviour of a modem. This specially wired cable, with two female connectors, is known as a null-modem cable.

A null modem cable swaps:

CTS with RTS

TXD with RXD

DSR and DCD with DTR.

**3 comments:**

Unknown April 17, 2013 at 4:51 AM

Before reading this post i was confused about [wireless RS 422](#), RS485 and RS232. But you really explain the concept very well and clear my doubt. Keep posting such a informative post.

[Reply](#)



Daccess Technologies August 11, 2016 at 1:21 AM

These [Guard tour patrol system in pune](#), monitor watch frameworks incorporate I-catch and RFID watching peruser.

[Reply](#)**hemmymission** August 11, 2017 at 8:51 PM

This blog is really helpful regarding all educational knowledge I earned. It covered a great area of subject which can assist a lot of needy people. Everything mentioned here is clear and very useful.

[belden](#)[Reply](#)

To leave a comment, click the button below to sign in with Blogger.

[SIGN IN WITH BLOGGER](#)[Home](#)

Subscribe to: [Posts \(Atom\)](#)

ELECTRONICS AND EMBEDDED OPEN SOURCE

[89c51](#)[ARM programming with GNU tool chains](#)[Computer concepts](#)[Development tools for Embedded](#)[Developments of different applications summary](#)[Directories concept in linux](#)[Eeherald Embedded Tutorials](#)[Embeddec C explore](#)[Embedded linux tutorials](#)[Embedded News](#)[Embedded System](#)[Embedded system information & articles](#)[File concept in linux](#)[Linux basic view](#)[Linux Device drivers](#)[Linux help](#)[Linux journal](#)[Linux tutorials](#)[Microcontroller](#)[Operaing system concepts](#)[PIC24 tutorials](#)[Webinars](#)[Wifi tutorials](#)

by Lokesh, Embedded Engineer. Ethereal theme. Powered by [Blogger](#).