Practical RS232 Design Considerations

Introduction

In the 60's a standards committee now known as the Electronic Industries Association developed an interface to connect computer terminals to modems. Over the years this has been updated: the most commonly used version of the standard is RS-232C (sometimes known as EIA232); the most recent is RS-232E. RS stands for Recommended Standard. The standard defines the electrical and mechanical characteristics of the connection - including the function of the signals and handshake pins, the voltage levels and maximum bit rate.

This article aims at providing insight to the basic theory and specification of the RS232 signal format as well as offering sound practical advise applicable to the design and implementation of communication systems

What is RS232?

RS-232 is a single-ended data transmission system, which means that it uses a single wire for data transmission. Signals are processed when they are positive or negative when compared with a ground. A negative voltage signal represents logic '1', and positive voltage represents logic '0'. Because signals traveling this single wire are vulnerable to degradation, RS-232 systems are recommended for communication over short distances.

A typical RS-232 system is made up of two types of device, data communication equipment, DCE, and data terminal equipment, DTE. Typically DTE is defined as the communication source, and DCE is defined as the device that provides a communication channel between two DTE-type devices. DTE has a plug (Male) and a DCE a socket (Female).

However, after all this talk of standards a multitude of interpretations exist. Different manufacturers implement RS-232 communications in different ways - often with very good reasons. For instance, instead of being a DCE device, a measuring instrument might be configured as DTE. In this case you need an adaptor or the RS-232 cable wired differently than normal.

How fast is RS232?

The speed of RS-232 communications is expressed in Baud. In data acquisition, a Baud is equivalent to bits per second. The unit is named after Jean Maurice-Emile Baudot (1845-1903), a French telegraph engineer and the inventor of the first teleprinter. It was proposed at the International Telegraph Conference of 1927.

The maximum speed, according to the standard, is 20000 Baud. However, modern equipment can operate much faster than this. No matter how fast (or slow) your connection - the maximum number of readings per second you can take from your instrument depends on the software.

The length of the cable also plays a part in maximum speed. The longer the cable, the greater the cable's capacitance and the slower the speed you can run at and still obtain accurate results. The capacitance links voltage changes on one signal wire to an adjacent signal wire. Both resistance and capacitance increase with cable length. We generally recommend a maximum distance of 16 meters, but this depends on the type of hardware you are connecting and characteristics of the cable.

Flow control for RS232 devices

Both software and hardware flow control needs software to perform the handshaking task. This makes the term software flow control somewhat misleading. What is meant is that with hardware flow control, additional lines are present in the communication cable, which signal handshaking conditions. With software flow control, which is also known under the name XON-XOFF flow control, bytes are sent to the sender using the standard communication lines.

Using hardware flow control implies, that more lines must be present between the sender and the receiver, leading to a thicker and more expensive cable. Therefore, software flow control is a good alternative if it is not needed to gain maximum performance in communications. Software flow control makes use of the data channel between the two devices, which reduces the bandwidth.

For Software flow control, two bytes have been predefined in the ASCII set to be used with software flow control. These bytes are named XOFF and XON, because they can stop and restart transmitting. The byte value of XOFF is 19, it can be simulated by pressing Ctrl-S on the keyboard. XON has the value 17 assigned which is equivalent to Ctrl-Q. Using software flow control is easy. If sending of characters must be postponed, the character XOFF is sent on the line, to restart the communication again XON is used. Sending the XOFF character only stops the communication in the direction of the device, which issued the XOFF.

This method has a few disadvantages. One is already discussed: using bytes on the communication channel takes up some bandwidth. One other reason is more severe. Handshaking is mostly used to prevent an overrun of the receiver buffer, the buffer in memory used to store the recently received bytes. If an overrun occurs, this affects the way new coming characters on the communication channel are handled. In the worst case where software has been designed badly, these characters are thrown away without checking them. If such a character is XOFF or XON, the flow of communication can be severely damaged. The sender will continuously supply new information if the XOFF is lost, or never send new information if no XON was received.

This also holds for communication lines where signal quality is bad. What happens if the XOFF or XON message is not received clearly because of noise on the line? Special precaution is also necessary that the information sent does not contain the XON or XOFF characters as information bytes. Therefore, serial communication using software flow control is only acceptable when communication speeds are not to high, and when the probability of buffer overruns or data damage can occur, are minimal.

Hardware flow control is superior compared to software flow control using the XON and XOFF characters. The main problem is, that an extra investment is needed. Extra lines are necessary in the communication cable to carry the handshaking information. Hardware flow control is sometimes referred to as RTS / CTS flow control. This term mentions the extra input and outputs used on the serial device to perform this type of handshaking. RTS / CTS in its original outlook is used for handshaking between a computer and a device connected to it such as a modem.

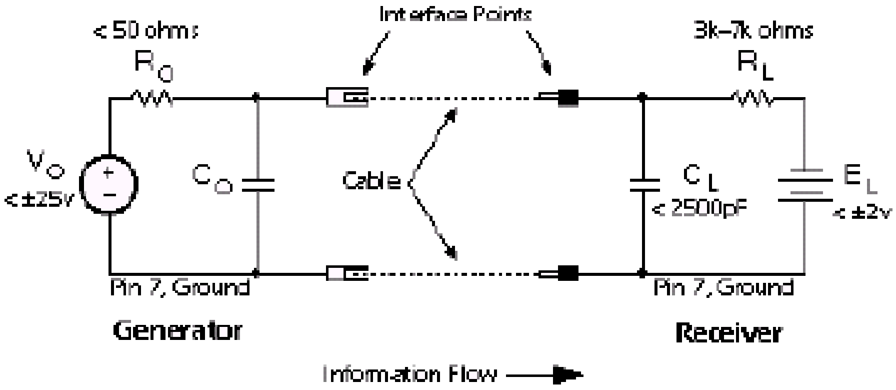
First, the computer sets its RTS line to signal the device that some information is present. The device checks if there is room to receive the information and if so, it sets the CTS line to start the transfer. When using a null modem connection, this is somewhat different. There are two ways to handle this type of handshaking in that situation.

One is, where the RTS of each side is connected with the CTS side of the other. In that way, the communication protocol differs somewhat from the original one. The RTS output of computer A signals computer B that A is capable of receiving information, rather than a request for sending information as in the original configuration. This type of communication can be performed with a null modem cable for full handshaking. Although using this cable is not completely compatible with the original way hardware flow control was designed, if software is properly designed for it, it can achieve the highest possible speed because no overhead is present for requesting on the RTS line and answering on the CTS line.

In the second situation of null modem communication with hardware flow control, the software side looks quite similar to the original use of the handshaking lines. The CTS and RTS lines of one device are connected directly to each other. This means, that the request to send query answers itself. As soon as the RTS output is set, the CTS input will detect a high logical value indicating that sending of information is allowed. This implies that information will always be sent as soon as the RTS output is set even when the DCE is ready or not. To prevent this from happening, two other pins on the connector are used, the data set ready DSR and the data terminal ready DTR. These two lines indicate if the device attached is working properly and willing to accept data. When these lines are cross-connected (as in most null modem cables) flow control can be performed using these lines. A DTR output is set, if that computer accepts incoming characters.

The RS232 electrical equivalent Circuit

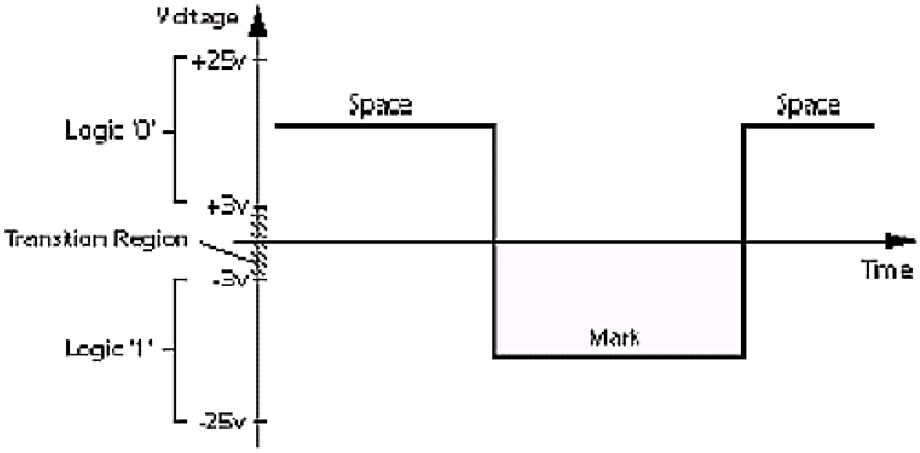
The electrical equivalent circuit shown here may represent all signal lines, regardless of whether they provide data, timing, or control information:



This is the equivalent circuit for an EIA232 signal line and applies to signals originating at either the DTE or DCE side of the connection. "Co" is not specified in the standard, but is assumed to be small and to consist of parasitic elements only. "Ro" and "Vo" are chosen so that the short-circuit current does not exceed 500ma. The cable length is not specified in the standard; acceptable operation is experienced with cables that are less than 8 meters in length.

- Signal State Voltage Assignments

Voltages of -3v to -25v with respect to signal ground (pin 7) are considered logic '1' (the marking condition), whereas voltages of +3v to +25v are considered logic '0' (the spacing condition). The range of voltages between -3v and +3v is considered a transition region for which a signal state is not assigned.



Logic states are assigned to the voltage ranges shown here. Note that this is a "negative logic" convention, which is the reverse of that used in most modern digital designs.

Most contemporary applications will show an open-circuit signal voltage of -8 to -14 volts for logic '1' (mark), and +8 to +14 volts for logic '0' (space). Voltage magnitudes will be slightly less when the generator and receiver are connected (when the DTE and DCE devices are connected with a cable).

> IMPORTANT: If you insert an LED signal tester in an EIA232 circuit to view signal states, the signal voltage may drop in magnitude to very near the minimum values of 3v for logic '1', and +3v for logic '0'. Also note that some inexpensive EIA232 peripherals are powered directly from the signal lines to avoid using a power supply of their own. Although this usually works without problems, keep the cable short, and be aware that noise immunity will be reduced.

- Short-Circuit Tolerance

The generator is designed to withstand an open-circuit (unconnected) condition, or short-circuit condition between its signal conductor and any other signal conductor, including ground, without sustaining damage to itself or causing damage to any associated circuitry. The receiver is also designed to accept any signal voltage within the range of ±25 volts without sustaining damage.

> CAUTION: Inductive loads or magnetically induced voltages resulting from long cables may cause the received voltage to exceed the ±25-volt range momentarily during turn-on transients or other abnormal conditions, possibly causing damage to the generator, receiver, or both. Keep the cable length as short as possible, and avoid running the cable near high-current switching loads like electric motors or relays.

- *Fail-Safe Signals*

Four signals are intended to be fail-safe in that during power-off or cable-disconnected conditions, they default to logic '1' (negative voltage). They are:
Request to Send - Default condition is not asserted.
DTE Ready - Default condition is DTE not ready.
DCE Ready - Default condition is DCE not ready.

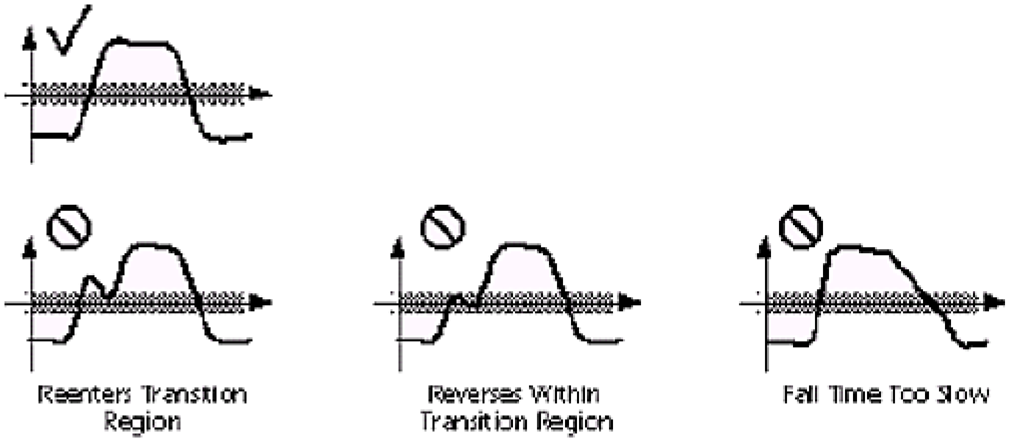
Note specifically that if the cable is connected but the power is off in the generator side, or if the cable is disconnected, there should be adequate bias voltage in the receiver to keep the signal above +3v (logic '0') to ensure that the fail-safe requirement is met. Schmitt triggers or other hysteresis devices may be used to enhance noise immunity in some designs, but should never be adjusted to compromise the fail-safe requirement.

- Signal Timing

Changes in signal state from logic '1' to logic '0' or vice versa must abide by several requirements, as follows:

1 - Signals that enter the transition region during a change of state must move through the transition region to the opposite signal state without reversing direction or reentering.
2 - For control signals, the transit time through the transition region should be less than 1ms.
3 - For Data and Timing signals, the transit time through the transition region should be
  a less than 1ms for bit periods greater than 25ms,
  b 4% of the bit period for bit periods between 25ms and 125µs,
  c less than 5µs for bit periods less than 125µs.

The rise and fall times of data and timing signals ideally should be equal, but in any case vary by no more than a factor of three.

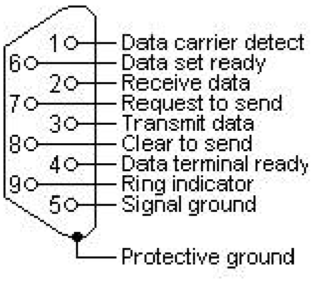


An acceptable pulse (top) moves through the transition region quickly and without hesitation or reversal. Defective pulses (bottom) could cause data errors.

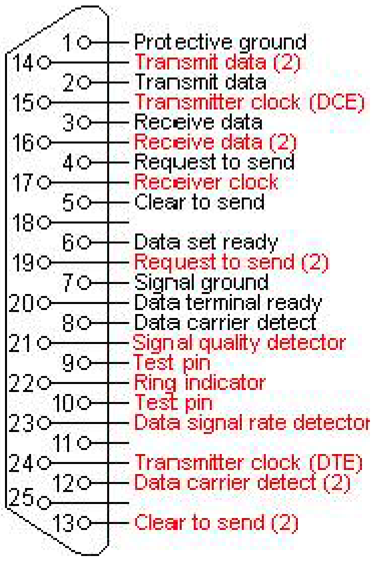
4 - The slope of the rising and falling edges of a transition should not exceed 30v/µS. Rates higher than this may induce cross talk in adjacent conductors of a cable.

RS232 pin assignment

DB 9 (DTE)



25 pin (DTE)



DTR: Data Terminal Ready--Used by a DTE to signal that it is plugged in and available to begin communication.
DSR: Data Set Ready--Sister signal to DTR, it is used by the DCE to indicate it is ready to begin communication.
CTS: Clear to Send--Used by DCE to signal it is available to send data, and used in response to a RTS request for data.
RTS: Request to Send--Used by a DTE to indicate that it wants to send data. Also, in a multi-drop network, used to turn
carrier on the modem on and off.
DCD: Data Carrier Detect--Used by a DCE to indicate to the DTE that it has received a carrier signal from the modem and
that real data is being transmitted.
RI: Ring Indicator--Used by DCE modem to tell the DTE that the phone is ringing and that data will be forthcoming.
TxD: Transmit Data--This wire is used for sending data.
RxD: Receive Data--This line is used for receiving data.
GND: Signal Ground--This pin is the same for DTE and DCE devices, and it provides the return path for both data and hand-shake signals.