The RS232 standard

- Its origins predate modern computers and it contains many features that are not relevant to the modern user.
- It can control very old primitive modems and has many control signals to do this in hardware.
- It is often used without these older control and status lines.
- There are only two common ways of controlling using modern PCs, the original 25 way connector, which is big and expensive has been superceded by a cheaper 9 pin variant.

The Data Format

- Normally used to send 7 or 8 bits of actual data, it is essential to prefix each data word with a start bit which must always be a “zero”.
- Also to allow the receiver to synchronise its timing to an edge, the start bit must have been preceded by a “one”, so the data is always followed by a “stop” bit of a “one” and the lines idle state is a “one”. It is possible to transmit a second character immediately after the first, the stop bit of the first being followed by the start bit of the next. Of course the receiving device must be ready to accept the two characters without losing one.
- The terms “zero” and “one” are not used when discussing RS232. A “Zero” is sent as +12 volts and a “one” as –12 volts, upside logic which can get quite confusing. The reasons for this are over 30 years old, originally telegraphs sent current through lines when in an idle or “mark” state and a startbit was therefore a “space”.
- We often draw diagrams using a logical convention of a ‘1’ at the top and a ‘0’ at the bottom.
- Don’t forget in the RS232 a ‘1’ or ‘MARK’ is actually –12 volts and a ‘0’ or ‘SPACE’ is actually +12 volts.
- We use voltage converter chips to translate from +/-12 to 0 to 5 volts suitable for interfacing to standard logic chips.
- These voltage converters are know as line receivers and line drivers and actually have inverters inside them so the –12 volts/’1’/MARK converts to +5 volts/logical high and the other voltage of +12volts/’0’/SPACE converts to 0 volts/logical low.

Line driver and receiver chips

- Typical line drivers/receivers chips for RS232 are the MAXIM MAX232 or MAX233 chips. (look these up at Http://www.maxim-ic.com) the original specification states that RS232 should drive 50 feet but modern line driver/receivers can manage much better than this. The maximum distance before errors occur is also a function of the type of cable used.

Baud Rate	max distance shielded cable	max distance unshielded cable
110	5000ft	3000ft
300	5000ft	3000ft
1200	3000ft	3000ft
2400	1000ft	500ft
4800	1000ft	250ft
9600	250ft	250ft

- The speed is quoted as the "BAUD" rate. Technically this is the number of signals per second. For simple transmissions this is the same as "Bits Per Second" or BPS. When RS232 is converted to audio whistles for transmission through the telephone this one to one relationship does not always hold. By using 4 different signals we can send 600 changes of signal per second (600 baud) but representing 2400 bits of information per second. This special coding suits the telephone system which has severe bandwidth limitations.

The transmission and reception of serial data

- The transmission and reception of serial data using the RS232 protocol can be undertaken in two ways; hardware and software.
- A special chip or special circuitry within a computer can be designed that uses a number of shift registers and holding latches/registers. Other circuitry can analyse the incoming character and detect errors such as parity, overrun and framing.
- Similarly circuitry can add parity bits and stop/start bits to outgoing transmissions. Provision for generating and clearing interrupts, differing modes of data format, a variety of baud rates and a number of control and handshake lines all mean that a hardware solution may require a large number of gates or silicon area. This can be expensive although if high performance is required then there is little choice.
- Typical hardware chips are the 8250, 8252 , 6850 and 6551 which are quite old and used with 8 bit microprocessors originally, they are still serviceable however.
- Modern PCs may use improved versions of the 8250, these contain extra facilities such as fifo buffers, they are “nearly” compatible with older programs, they are fully compatible if only OS provided services are employed to access the ports.

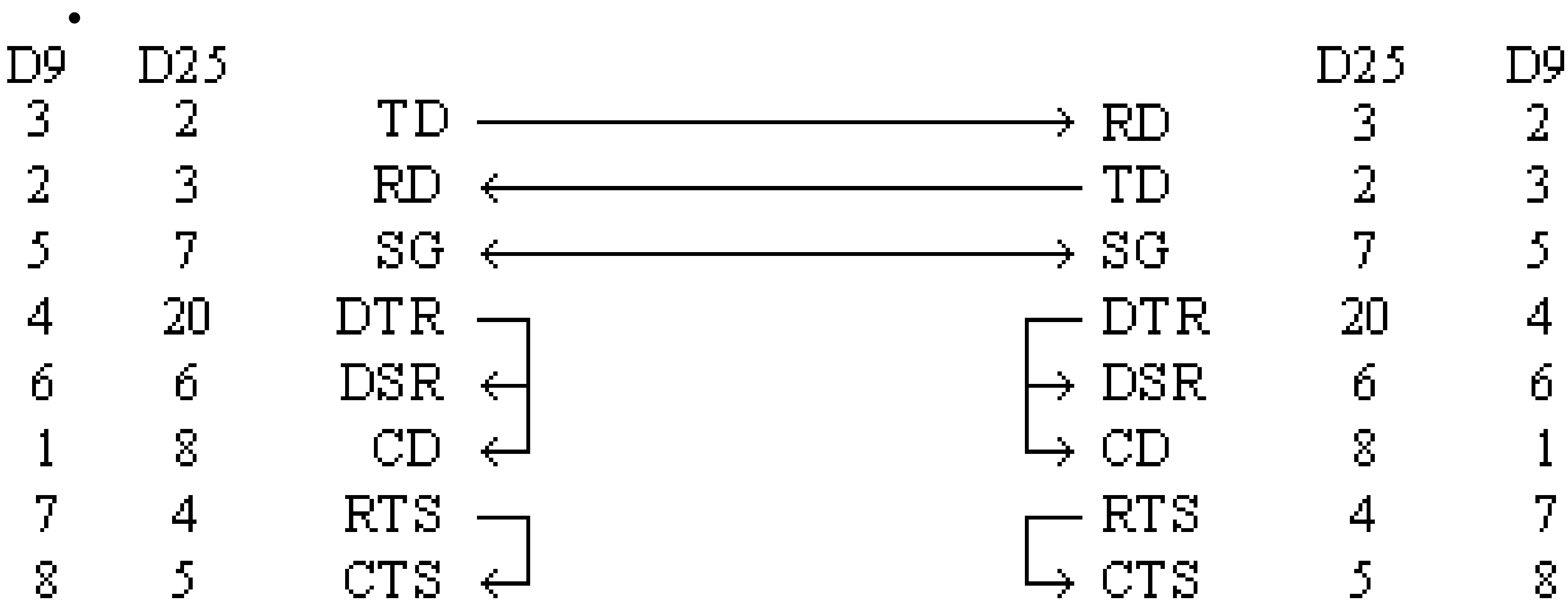
The control and status facilities

- We might need to assert or de-assert other output pins to control the flow of incoming data. Alternatively it might be necessary to monitor a input pin and halt the flow of outgoing data if a handshake input line is asserted.
- Typically these control signals are called RTS, Request to Send and CTS, Clear To Send.
- The other typical handshake/Control lines are DTR/DSR; these are Data Terminal Ready and Data Set Ready. They usually indicate that each end of the link is switched on and connected. RTS/CTS are used for a character by character handshake.
- The other type of data flow control is Xon/Xoff. This uses special characters embedded in the data stream. Control-S and Control-Q are most often used. This is often used when complete buffers of data are sent and each receiver can actually accept data in multiples of one character. A printer can have a 80 character buffer for instance.
- The PC interface provides two outputs RTS and DTR and three inputs CTS, DSR and DCD.

Relativity

- The two ends of an RS232 link used to be called DTE and DCE, standing for Data Terminal Equipment (e.g a computer) and Date Communcation Equipment (e.g modem). Since the outputs and inputs are all uni-directional you can only connect a PC to a modem using a 1:1 cable.
- To interconnect two PCs you have to cross over outputs and inputs, thus the dataout on Pin 3 (of a 9 pin D-type) has to be linked across to Pin 2 of the other PC, the data in pin. The only exception is the ground, or reference pin, pin 5.
- Serial Ports come in two "sizes", There are the D-Type 25 pin connector and the D-Type 9 pin connector both of which are male on the back of the PC, thus you will require a female connector on your device. Below is a table of pin connections for the 9 pin and 25 pin D-Type connectors.
- Serial Pinouts are listed on www.beyondlogic.org/serial/serial.htm but note … for the 9 pin D-type only! pin 3 is the data output on a PC, pin 2 is the data input and pin 5 is ground. Pin 7 is the RTS output and Pin 4 is the DTR output.
- To connect two PCs together we use what is called a “null modem cable”

Null Modem RS232 Lead



Typical meanings of hardware control

- DTR, when output by a PC " I am switched on" it stands for DATA TERMINAL READY
- DSR, an input from a serial peripheral, "I am switched on and initialised; DATA SET READY
- DCD, if a modem then there is a connection, DATA CARRIER DETECT
- CTS, CLEAR TO SEND, an input on the PC, a peripheral will assert this to tell the PC that it may transmit to it, "I am ready to receive"
- RTS, REQUEST TO SEND, an output from a PC, it will go high when the PC wants to send something, a receiver should assert CTS when it sees it.

- Thus there exists the possibility for "flow control" where a receiver of data can throttle back or halt the transmitter of that data, this might get used when interfacing to a serial printer for instance.
- Many alternative "protocols" exist, the above is the closest to the original spec. (which was written for really old mechanical hardware of no relevance now)

Software flow Control

- So if our DTE to DCE speed is several times faster than our DCE to DCE speed the PC can send data to your modem at 115,200 BPS. Sooner or later data is going to get lost as buffers overflow, thus flow control is used. Flow control has two basic varieties, Hardware or Software.
- Software flow control, sometimes expressed as Xon/Xoff uses two characters Xon and Xoff. Xon is normally indicated by the ASCII 17 character where as the ASCII 19 character is used for Xoff. The modem will only have a small buffer so when the computer fills it up the modem sends a Xoff character to tell the computer to stop sending data. Once the modem has room for more data it then sends a Xon character and the computer sends more data. This type of flow control has the advantage that it doesn't require any more wires as the characters are sent via the TD/RD lines. However on slow links each character requires 10 bits which can slow communications down.
- Software flow control can be a disaster on noisy lines, but it is ok to use it when lines are short and you expect perfect communication.

Simple terminal Program

```
/********* snippet of interesting C program ***************/
printf("\nSample Comm's Program. Press ESC to quit \n");
do {
   c = inportb(PORT1 + 5);          /* Check to see if char has been
received. */
   if (c & 1) {
       ch = inportb(PORT1);         /* If so, then get Char */
       printf("%c",ch);
    }                               /* Print Char to Screen */
    if (kbhit()){
       ch = getch();                /* If key pressed, get Char */
       outportb(PORT1, ch);
     }                              /* Send Char to Serial Port */
} while (ch !=27);                 /* Quit when ESC (ASC 27) is pressed */ }
```

RS485/RS422

So, what is the main difference between RS 232 and RS 422 & 485? The RS 232 signals are represented by voltage levels with respect to ground. There is a wire for each signal, together with the ground signal (reference for voltage levels). This interface is useful for point-to-point communication at slow speeds. For example, port COM1 in a PC can be used for a mouse, port COM2 for a modem, etc. This is an example of point-to-point communication: one port, one device. Due to the way the signals are connected, a common ground is required. This implies limited cable length - about 30 to 60 meters maximum. (Main problems are interference and resistance of the cable.) Shortly, RS 232 was designed for communication of local devices, and supports one transmitter and one receiver.

RS 422 & 485 uses a different principle: Each signal uses one twistedpair (TP) line - two wires twisted around themselves. We're talking 'Balanced data transmission', or 'Differential voltage transmission'. Simply, let's label one of the TP wires 'A' and the other one 'B'. Then, the signal is inactive when the voltage at A is negative and the voltage at B is positive. Otherwise, the signal is active, A is positive and B is negative. Of course, the difference between the wires A and B matters. For RS 422 & 485 the cable can be up to 1200 meters (4000 feet) long, and commonly available circuits work at 2.5 MB/s transfer rate.

RS422/485

What is the difference between RS 422 and RS 485? Electrical principle is the same: both use differential transmitters with alternating voltages 0 and 5V. However, RS 422 is intended for point-to-point communications, like RS 232. RS 422 uses two separate TP wires, data can be transferred in both directions simultaneously. RS 422 is often used to extend a RS 232 line, or in industrial environments.

RS 485 is used for multipoint communications: more devices may be connected to a single signal cable - similar to e.g. ETHERNET networks, which use coaxial cable. Most RS 485 systems use Master/Slave architecture, where each slave unit has its unique address and responds only to packets addressed to this unit. These packets are generated by Master (e.g. PC), which periodically polls all connected slave units.

http://www.hw.cz/english/docs/rs485/rs485.html

This article will mainly cover the Master/Slave architecture because it is sufficient for 95% of applications.

In special cases (security systems, ...), an improved version of multiprocessor communication is used. This system uses only a single line for bidirectional communication; however, there is no Master. All units announce a packet transmission of a specified length, and at the same time listen whether the data has been successfully transmitted.

If it's not the case, they stop communicating and listen for what has happened. At this time, urgent packets can be transmitted over the line. This system is ideal for devices, that need to immediately transfer some very important and up-to-date data, without waiting for Master to give them a chance to do so. On the other side, useful data transfer is less effective (about 30% less effective than the first system).

In Master/Slave architecture, slave never starts the communication.

It is critical for Master to send correct addresses.

RS 485 exists in two versions: 1 TwistedPair or 2 TwistedPairs.

- Single TwistedPair RS 485

In this version, all devices are connected to a single TwistedPair. Thus, all of them must have drivers with tri-state outputs (including the Master). Communication goes over the single line in both directions. It is important to prevent more devices from transmitting at once (software problem).

- Double TwistedPair RS 485

Here, Master does not have to have tri-state output, since Slave devices transmit over the second twistedpair, which is intended for sending data from Slave to Master. This solution often allows to implement multipoint communication in systems, which were originally designed (HW as well as SW) for RS232. Of course, Master software needs to be modified, so that Master periodically sends query packets to all Slave devices. Increased data throughput is evident in large volumes.

Sometimes you can see a RS 485 system in a point-to-point system. It is virtually identical to RS 422; the high impedance state of the RS 485 output driver is not used. The only difference in hardware of the RS 485 and RS 422 circuits is the ability to set the output to high impedance state.

Advantages of differential signals

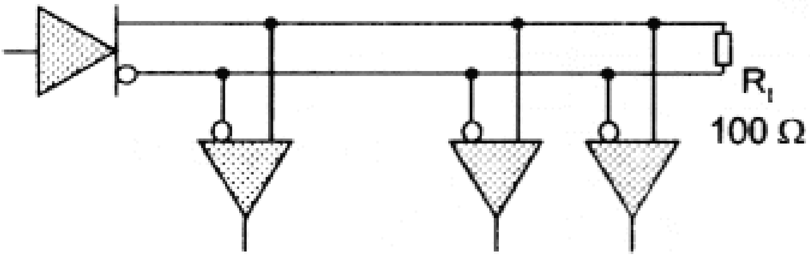
First, let's talk about advantages and disadvantages of RS 422/485. For a basic RS 422/485 system, we need an I/O driver with differential outputs and an I/O receiver with differential inputs. Noise and interference is introduced into the line; however, since the signal is transferred via a twisted pair of wires, the voltage difference (between A and B) of this interference is almost zero. Due to the differential function of the RS 422/485 input amplifier of the receiver, this interference is eliminated. The same is true for crosstalk from neighbouring lines, as well as for any other source of interference, as long as the absolute maximum voltage ratings of the receiver circuits are not exceeded.

Differential inputs ignore different earth potentials of the transmitter and the receiver. This is very important for communications of diverse systems, where great problems would otherwise arise - e.g. different power sources, etc. TwistedPair cables, together with correct terminations (to eliminate reflections), allow data transfer rate of over 10Mbit/s with cables up to 1 km long.

However, all of these advantages come at a cost. RS 422/485 circuits are more complex, and thus more expensive. Higher data transfer speeds require correctly connected and matched terminations, which can be a problem in systems where the number of connected devices changes. And, of course, TwistedPair cables are required.

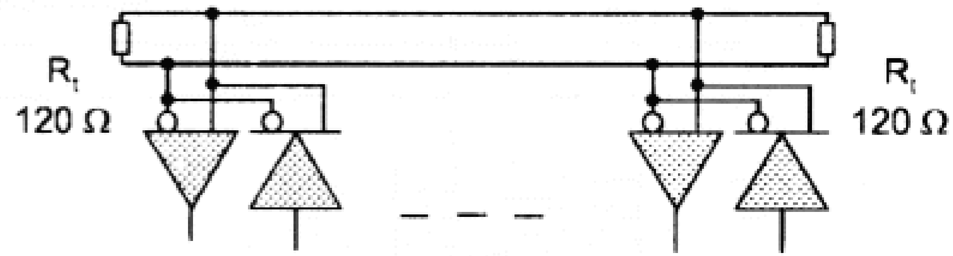
Comparison RS422 - RS485

RS-422



1 generator
up to 10 receiver
simplex operation

RS-485



up to 32 generators / receivers
Half duplex operation

RS-422		RS-485
-7V to +7V	Max common mode voltage	-7 V to +12 V
4 kΩ	Receiver input impedance	12 kΩ
100 Ω	Minimum generator load	60 Ω
<150 mA to GND	Generator short circuit current	<250mA to -7 V/+12 V

Terminations,cable lengths, data transfer speed

RS 422/485 line termination is essential, especially for faster data transfer rates and long cables.

Main reasons for correct termination are reflections at the ends of the line, and the minimum transmitter load requirement.

For RS 422, the termination is fairly simple (see picture comparing RS 422 and RS 485). A terminating resistor of 100Ohm is connected to the end of the line.

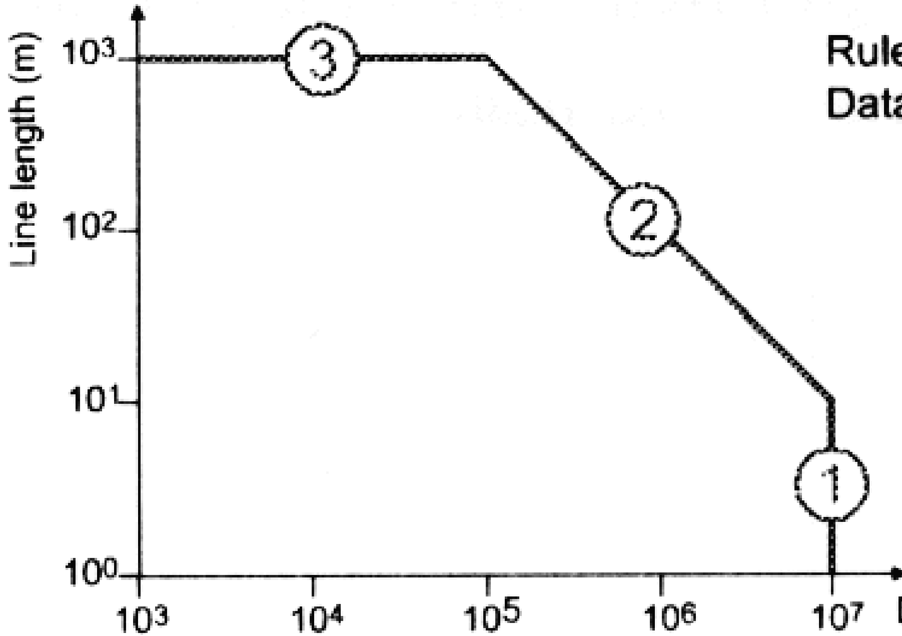
If there are more RS 422 receivers connected to the line, the resistor can be a little bigger. The value can be calculated since the input impedance of the receivers is known.

There are more sophisticated methods that will “bias” the line into a known state if ALL transmitters are disabled, this can be very important to avoid detection of false start bits.

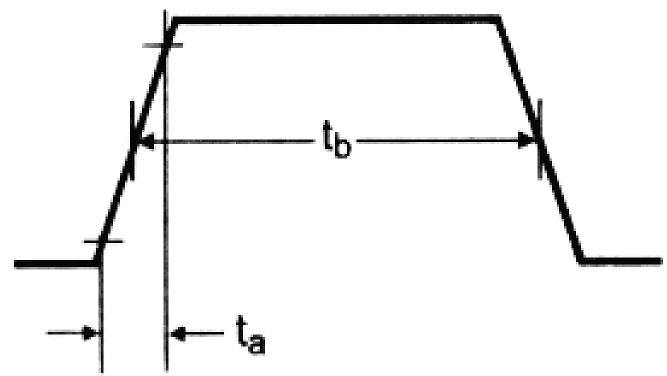
Also the power consumption of the terminators can be reduced, see the national semiconductor website for application notes (www.natsemi.com)

Speed vs distance

Maximum Data Rate in a Balanced Interface



Rule of thumb:
Data rate (Bit/s) × line length (m) ≤ 10^8



1. Data rate is limited by the rise time of the output signal of the generator (ITU-T V.11: $t_r \le 0.1 \times t_b$; ISO8482: $t_r \le 0{,}3 \times t_b$).
2. Data rate is limited due to signal distortion caused by losses on the transmission line (e.g. skin effect).
3. Data rate is limited due to the attenuation caused by the ohmic resistance of the wires (a 1000 m twisted pair cable with wires of 0.6 mm diameter has a resistance of ≈ 100 Ω).

Protocols

From a network point of view, the RS 485 incorporates a bus topology. Since Slave stations have no means of starting the communication without a risk of collision, they need to be assigned a 'right to transmit' by the Master station.

Assignment is done centrally via pooling, where the central (Master) station periodically asks all Slaves whether they have data to transmit. If so, the questioned station sends the data immediately; otherwise, it replies with a confirmation packet only, or does not reply at all.

This method is good for Multipoint systems with smaller number of Slave stations (approx. up to 100). For more stations, the reply would become too slow. Of course, individual system requirements need to be considered.

Mentioned 100 stations is for an "on-line" system, where stations have to interactively react to user requests, thus the reply delay needs to be less than 0.5 sec (considered for 115200bd data transfer rate, which is seldom available in industrial environments).

Protocols continued

Of course, in systems where the Master has no priority function, or due to other factors (e.g. large number of stations with low frequency of data transfers), different access methods may be used.

For example, the random access method ALOHA. Here, any station sends its data regardless of the transfer channel status. If a collision occurs, the station does not receive a confirmation, and repeats transmission. However, this method utilises on average only about 18% of available bandwidth, and with larger volumes of data the throughput decreases rapidly due to larger number of collisions.

With RS 485, where transmitters can at the same time "listen" for the channel status, the ALOHA method can be improved by a "carrier" (data activity) detection. In this case, stations begin transmission only if the channel is idle. Both methods essentially require a transfer protocol with error detection.

Final notes on RS485…

Commonly available parts for RS 422/485. MAXIM; they make about ten versions - differences are in speed (0.25 or 2.5 Mbit/s), operation type (half/full duplex), number of devices (32, 128), and in several other parameters. They are labeled MAX481, 485, 487, 491 and so on. (www.maxim-ic.com)

Another manufacturer is National Semiconductors, label DS3695A. I assume that every major manufacturer offers some circuits for RS 422/485; however, I don't know any details. (www.natsemi.com)

Some info in this article comes from excellent publication by Texas Instruments, who make a whole family of RS 422/485 circuits labelled SN 7517X, where X is one- or two-digit type specification. (www.ti.com)

Also Linear Technology make 485 line drivers, receivers and transceivers.

All of these manufacturers have good application notes that will help you design RS485 networks.

Several hints:

- Don't know which wire is A and B? When idle, B is more positive than A.
- You don't always have to use TP cables. For small distances and low speeds, common telephone cables are good enough.
- Termination is not critical for small distances and low speeds - works fine with MAX circuits.