

How RS232 Works

RS232 is a method (or protocol) that defines how to transfer data between two devices using a few wires.

It uses a serial transmission method where bytes of data are sent out, one bit at a time, onto a single wire.

Data is only transmitted in one direction for each wire so for bi-directional communication (two directions) you will need two wires. These two wires, along with a ground reference, make up the minimum configuration that is acceptable.

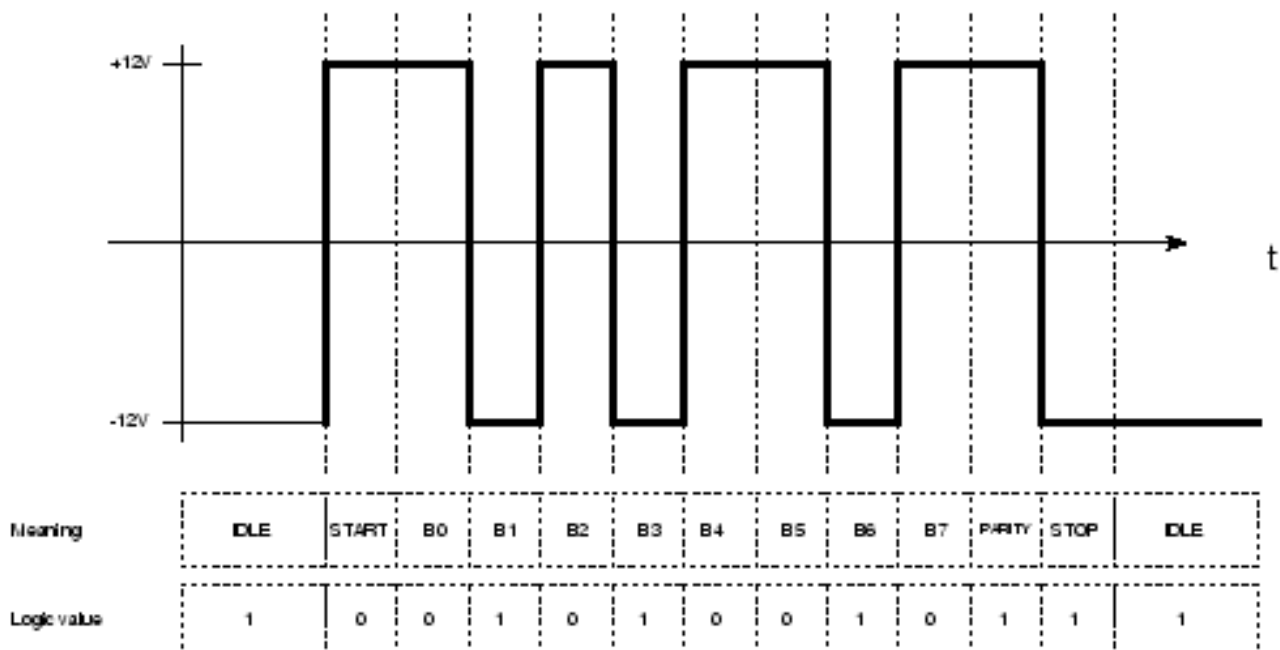
Basically, RS232 can transfer a single byte of data over a serial cable having between 3 to 22 signals and running at speeds from 100 to 20k baud. Common baud rates used are 2.4k, 9.6k, 19.2k; the cable length can be up to 50ft. Higher baud rates are used but not covered by the standard they still work though e.g. 38400, 57600 Baud (bits/s).

To transfer a block of data individual bytes are transmitted one after another.

Data is transmitted serially in one direction over a pair of wires. Data going out is labelled Tx (indicating transmission) while data coming in is labelled Rx (indicating reception). Crossing over Tx & Rx between the two systems lets each unit talk to the other.

Each byte can be transmitted at any time (as long as the previous byte has been transmitted). The transmitted byte is not synchronized to the receiver - it is an asynchronous protocol i.e. there is no clock signal. For this reason, software at each end of the communication link must be set up in exactly the same way so that each serial decoder chip can decode the serial data stream.

RS232 Transmission of the letter 'J'



Note: The signal level inversion (logic 1 is -12V and logic 0 is +12V).

Baud

This is simply the transmission speed measured in bits per second. It defines the frequency of each bit period.

For a baud rate of 2400 (2400 bps) the frequency is 2400Hz and the bit period is 1/2400 or 416.6us. This is the information that a receiver uses to recover the bits from the data stream.

Voltage levels

Transmitter

In order to work over long cables, high voltages are sent from each transmitter. Due to cable resistance the voltage reduces the further the signal has to travel. The output voltage specification is from +5V to +25V (transmitting a logical zero) and -5V to -25V (transmitting a logical one).

All signals in the cable have to generate the same voltage levels e.g. DTR, DSR, RTS, CTS. So you need a lot of level translator chips for a full interface but for very short distances you only need Tx and Rx and ground.

The receiver can accept minimum signal levels of $\pm 3V$ and a maximum voltage of $\pm 25V$, however most systems use $\pm 12V$

Receiver

At the receiver the input voltage levels are defined as $\pm 3V$ i.e. to receive a logic zero the voltage must be greater than +3V and to receive a logic one the voltage must be smaller than -3V. This allows for voltage drop and any spurious noise.

Bitstream

At the beginning of each transmission a “start bit” is transmitted indicating to the receiver that a byte of data is about to follow. The start bit lets the receiver synchronize to the data bits. Once the start bit is found the receiver knows where the following bits will be as it is given the sample period (derived from the baud rate) as part of the initialization process.

Data bits follow the start bit. There will be seven or eight data bits with the lsb transmitted first. You can choose between seven or eight bits as ASCII is made up of the alphabet within the first seven bits. The eighth bit (optional) extends the character set for graphical symbols.

Note: If you use RS232 to transmit raw data (binary data) then you will need 8 data bits.

The “parity bit” is a crude error detection mechanism. You can use either odd parity or even parity or none at all (in this case no parity bit is transmitted). It simply evaluates all the data bits and for odd parity returns a logic one if there is an odd number of data bits that are set. For even parity an even number of data bits that are set, sets the parity bit. At the receiver the parity bit is used to tell if an error occurred during transmission.

The “stop bit” gives a period of time before the next start bit can be transmitted. It is the opposite sense to the start bit and because of this allows the start bit to be seen. The stop bit can be set choosing from 1, 1.5, or 2 bit periods.

Typical Settings

Typical settings for use on the desktop e.g. between a microcontroller and HyperTerminal:

Typical Baud rate settings:

| | |
|-----------|------|
| Baud | 9600 |
| Data bits | 8 |
| Parity | None |
| Stop bits | 1 |

Hardware Connections 3 (Rx,Tx,GND) - Rx and Tx crossed over.



HEAD OFFICE AND FACTORY
Aquavision
Unit 2 Grosvenor Business Park
Horsfield Way, Stockport
Cheshire, SK6 2SU
United Kingdom
+44 [0]161 711 1900

AMERICAN OFFICE
Aquavision, Inc.
#197
5 Southside Drive, Suite 11
Clifton Park, NY 12065
America
+1 800 524 1804



sales@aquavision.tv
sales@aquavision.us

www.aquavision.tv
www.aquavision.us