

The Virtual Terminal Model

Overview



The VSM Virtual Terminal enables you to use the keyboard and screen of your PC to send and receive RS232 asynchronous serial data to and from a simulated microprocessor system. It is especially useful in debugging where you can use it to display debug/trace messages generated by the software which you are developing.

The Virtual Terminal is specified as follows:

- Fully bi-directional - received serial data is displayed as ASCII characters whilst key presses are transmitted as serial ASCII data.
- Simple two wire serial data interface: RXD for received data and TXD for transmitted data.
- Simple two wire hardware handshake interface: RTS for ready-to-send and CTS for clear-to-send.
- Baud rate from 300 to 57,600 baud.
- 7 or 8 data bits.
- Odd, even or no parity.
- 0, 1 or 2 stop bits.
- XON/XOFF software handshaking in addition to hardware handshaking.
- Normal or inverted polarity for both RX/TX and RTS/CTS signals.

Using the Virtual Terminal

To attach a teletype terminal to your simulated target system:

1. Select the *Multimeter* icon, pick the VIRTUAL TERMINAL and place one on the schematic.
2. Wire the RX and TX pins to the transmitting and receiving lines of the system under test. RX is an input and TX is an output.
3. If the target system uses hardware handshaking, wire the RTS and CTS pins to the appropriate flow control lines. RTS is an output, signalling that the virtual terminal is ready to receive data, whilst CTS is an input which must be high (or floating) before the virtual terminal will transmit.
4. Edit the virtual terminal component to select the required baud rate, word length, parity, flow control and polarity settings.
5. Start the simulation in the usual way. The terminal will display incoming data as soon as it is received; to transmit characters into the system, make sure the terminal popup has the focus and then type the required text on the PC keyboard.
6. Once the simulation is started, further functionality is available from a context menu which may be displayed by clicking right on the popup window. This menu enables you to pause the display, and to copy and paste text to and from the clipboard.

Notes:



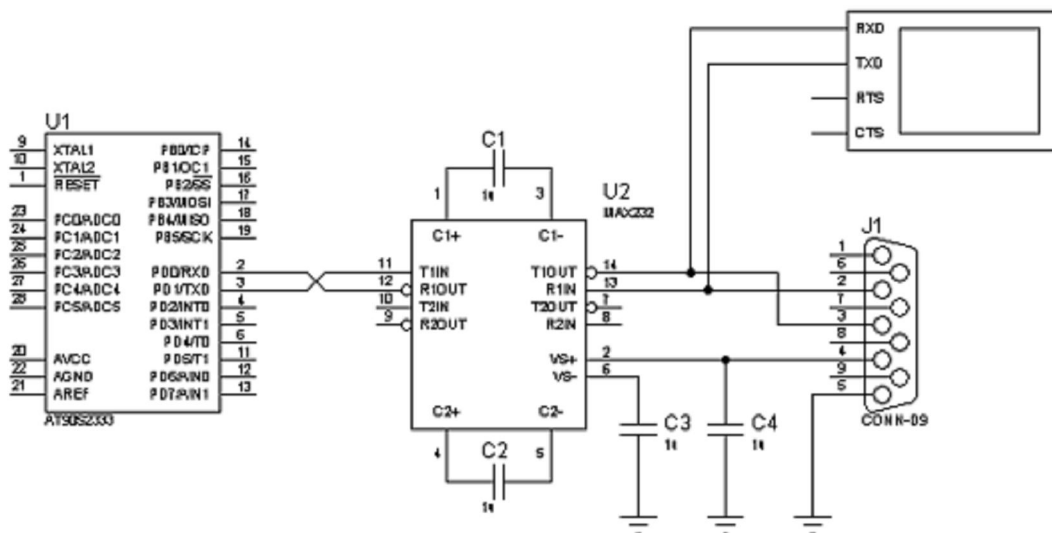
- i** The Virtual Terminal supports the ASCII control codes CR (0Dh), BS (0x08h) and BEL (0x07h). All other codes including LF (0x0A) are ignored.
- i** The Virtual Terminal is a purely digital model and as such does not require any particular voltage levels on its pins. The assumption is that it will normally be wired directly to a CPU or UART rather than to the output side of an RS232 driver such as the MAX232 which contains logic inverters.
- i** The RX and TX pins default to active high. Thus the idling state is high, the start bit is low and the stop bit high. Data bits appear as logic high for '1' and logic low for '0'. This is directly compatible with the on board UARTs present in many microcontrollers, and also with external UARTs such as the 6850 and 8250. If this is not the case (typically because you are connecting the terminal on the output side of an RS232 driver) you will need to set the *RX/TX polarity* to be *inverted*.
- i** The RTS and CTS pins are also active high. If you wish to connect them to inverted control lines (i.e. RTS/CTS bar), you should set the *RTS/CTS polarity* to be *inverted*.
- i** By default, the terminal does **not** display (echo) the characters you type; the assumption is that the host system to which it is connected will do this. If you wish the terminal to display typed characters itself, you should select *Echo Typed Characters* from the context menu.
- i** You can pre-define text to be transmitted into your circuit at start up using the **TEXT** property.
e.g.

TEXT="Hello World"

will transmit the text "Hello World" into the circuit. The script text is transmitted one character time after start up, unless CTS is inactive.

MAX232 Model

Proteus VSM includes a model for the MAX232 RS232 driver and it is thus possible to connect the virtual terminal to a target CPU as shown below:



However, it is vital to note that the MAX232 contains logic inverters and so the RX/TX polarity of the virtual terminal must be changed to inverted if this circuit is to simulate properly.

Please note also that the MAX232 is modelled at the digital level only - no attempt is made to simulate the operation of its internal voltage converters etc. Simulating these features would incur a huge performance overhead for little purpose. The same performance hit will also be incurred if you connect any analogue components (resistors, capacitors, oscilloscope etc.) to the TxOut/RxIn pins.