



Serial Handshaking in general

RS232 handshaking was defined when the targets were mainly modems and terminals, so terms are defined in that context. The Data Terminal Equipment (or Host, or the computer) sends data to and receives data from the Data Circuit-terminating Equipment (the serial device).

<u>Abbr.</u>	<u>Signal</u> <u>Full Name</u>	<u>Asserted by</u>	<u>Meaning</u>
DTR	Data Terminal Ready	COM port	The computer is ready for serial operation
DSR	Data Set Ready	Device	The serial peripheral device is ready to operate
RTS	Request to send	COM port	The computer is ready to receive (requesting <i>the device</i> to send)
CTS	Clear to send	Device	The device is ready to receive (the COM port is clear to send data)
Tx	Transmit data	COM port	Computer's transmit line
Rx	Receive data	Device	Device's transmit/computer's receive line
Ground			
DCD	Data Carrier Detect		Not applicable to most general serial devices
RI	Ring Indicator		Not applicable to most general serial devices

DTR is asserted by the computer COM port when the COM port is ready to operate.

DSR is asserted by the serial device when the device is ready to operate.

RTS and CTS are the actual handshaking lines.

RTS is asserted by the computer COM port whenever it can receive data; RTS is driven low when the COM port is busy sending or receiving data.

CTS is asserted by the serial device whenever it can receive data; CTS is driven low by the device when the device is busy sending or receiving data.

Elo handshake particulars are on the next page.



Elo Handshaking

Applies only when handshaking is enabled (single-touch driver by default, multi-touch driver only when so configured by the eloptions.ini file).

In the following discussion, "asserted" means "driven high".

The DSR line is asserted when the Elo controller is powered up (the Data Set is Ready).

The DTR line is asserted by the comport when it has been initialized. The Elo controller cannot transmit unless DTR is asserted, indicating that the Data Terminal is Ready. (*)

The Elo controller keeps CTS low while it is running the Power On Self-Test; after POST completes, the Elo controller asserts CTS (saying, "Data terminal, I am ready to receive data, you are Clear to Send").

By default, if the COM port does not detect DSR and CTS asserted, the single-touch driver (5-series) will not load; it generates a "No touchscreen controller detected" message. By default, the multi-touch (6-series) driver ignores DSR and CTS and loads the driver.

On receipt of CTS at the COM Port, the software driver instructs the computer to send a diagnostics query (literally, transmits 'd0000000', the SmartSet diagnostics query command, on the TxD line). The computer hasn't received data from the Elo controller yet, so the computer is ready to receive and so indicates by having RTS asserted (Request to *device* To Send a reply).

When the Elo controller receives the command and verifies that it is a valid command, it de-asserts CTS to tell the computer not to send further data until it has processed the command. Since RTS is asserted, the Elo controller sends the response packet on the RxD line (then reasserts CTS); the driver interprets the response and displays the POST results in the driver's Properties section.

For all intents and purposes, handshaking is required only when the driver loads. The sequence that matters (for the single-touch driver) is:

DSR	Asserted by Elo controller on power up
DTR	Asserted by COM port on power up/initialization (or float)
CTS	Asserted by Elo controller after POST
RTS	Asserted by COM port when RxD line is inactive (or float) (COM port ready to receive).

(*) DTR, strictly speaking, doesn't HAVE to be asserted; as long as neither DTR or RTS is pulled low (meaning, both float), the controller can transmit. Continued on the following page.



In reality, if neither DTR or RTS is pulled low - that is, if they are not connected at all - the only handshaking signals needed are:

DSR Asserted by Elo controller on power up
and
CTS Asserted by Elo controller after POST.

This is because both DTR and RTS are connected through diodes to a pullup resistor in the Elo controller. If either line is pulled to ground, the UART on the Elo controller will see a logic low (the 0.7 volt drop across the diode); otherwise, both DTR and RTS will be seen as high.

Normally a serial peripheral transmits data to the COM port only when RTS is asserted. But note that the multi-touch Elo driver ignores handshaking by default and the single-touch driver can be silently installed to ignore handshaking. In these cases, a simple 2-wire connection can be used (RxD and ground). How can Elo get by with that? The answer is that something like a modem, which handles a lot of bidirectional transmission, very much needs to rely on handshaking to make sure that only the modem or the COM port is talking at any given time (both the device and the COM port have a line they can pull low to say, “don’t send, I’m sending data now”). A pokey touchscreen, however, is pretty much talking only to the COM port; the COM port doesn’t have to talk back. And 9600 BAUD, the default transmission rate, is plenty fast to get the data sent before another touch packet becomes ready to send. It’s conceivable that a program or an operator could try to send a command to the Elo controller while someone was touching the touchscreen, and that would result in corrupted data. But it’s not a common problem. If anyone is really worried about it, the multi-touch driver can be forced to require handshaking (an option in the `eloptions.ini` file).